



HAL
open science

Algorithmique et contrôle en vision par ordinateur

Augustin Lux

► **To cite this version:**

Augustin Lux. Algorithmique et contrôle en vision par ordinateur. Interface homme-machine [cs.HC]. Institut National Polytechnique de Grenoble - INPG; Université Joseph-Fourier - Grenoble I, 1985. Français. NNT: . tel-00010610

HAL Id: tel-00010610

<https://theses.hal.science/tel-00010610>

Submitted on 13 Oct 2005

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.

TU 8084

THESE

présentée à

l' Université Scientifique et Médicale de Grenoble
et à
l' Institut National Polytechnique de Grenoble

pour obtenir le grade de
DOCTEUR ES SCIENCES
« Informatique »

par

Augustin LUX



ALGORITHMIQUE ET CONTROLE EN VISION PAR ORDINATEUR.



Thèse soutenue le 20 septembre 1985 devant la commission d'examen.

G. VEILLON	} Président
O.D. FAUGERAS	
J.C. LATOMBE	} Rapporteurs
R. MOHR	
J. L. CROWLEY	} Examineurs
J. MOSSIERE	

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

Année universitaire 1982-1983

Président de l'Université : M. TANCHE

MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.

(RANG A)

SAUF ENSEIGNANTS EN MEDECINE ET PHARMACIE

PROFESSEURS DE 1ère CLASSE

ARNAUD Paul	Chimie organique
ARVIEU Robert	Physique nucléaire I.S.N.
AUBERT Guy	Physique C.N.R.S.
AYANT Yves	Physique approfondie
BARBIER Marie-Jeanne	Electrochimie
BARBIER Jean-Claude	Physique expérimentale C.N.R.S. (labo de magnétisme)
BARJON Robert	Physique nucléaire I.S.N.
BARNOUD Fernand	Biosynthèse de la cellulose-Biologie
BARRA Jean-René	Statistiques - Mathématiques appliquées
BELORISKY Elie	Physique
BENZAKEN Claude (M.)	Mathématiques pures
BERNARD Alain	Mathématiques pures
BERTRANDIAS Françoise	Mathématiques pures
BERTRANDIAS Jean-Paul	Mathématiques pures
BILLET Jean	Géographie
BONNIER Jean-Marie	Chimie générale
BOUCHEZ Robert	Physique nucléaire I.S.N.
BRAVARD Yves	Géographie
CARLIER Georges	Biologie végétale
CAUQUIS Georges	Chimie organique
CHIBON Pierre	Biologie animale
COLIN DE VERDIERE Yves	Mathématiques pures
CRABBE Pierre (détaché)	C.E.R.M.O.
CYROT Michel	Physique du solide
DAUMAS Max	Géographie
DEBELMAS Jacques	Géologie générale
DEGRANGE Charles	Zoologie
DELOBEL Claude (M.)	M.I.A.G. Mathématiques appliquées
DEPORTES Charles	Chimie minérale
DESRE Pierre	Electrochimie
DOLIQUE Jean-Michel	Physique des plasmas
DUCROS Pierre	Cristallographie
FONTAINE Jean-Marc	Mathématiques pures
GAGNAIRE Didier	Chimie physique

.../...

GASTINEL Noël	Analyse numérique - Mathématiques appliquées
GERBER Robert	Mathématiques pures
GERMAIN Jean-Pierre	Mécanique
GIRAUD Pierre	Géologie
IDELMAN Simon	Physiologie animale
JANIN Bernard	Géographie
JOLY Jean-René	Mathématiques pures
JULLIEN Pierre	Mathématiques appliquées
KAHANE André (détaché DAFCO)	Physique
KAHANE Josette	Physique
KOSZUL Jean-Louis	Mathématiques pures
KRAKOWIAK Sacha	Mathématiques appliquées
KUPTA Yvon	Mathématiques pures
LACAZE Albert	Thermodynamique
LAJZEROWICZ Jeannine	Physique
LAJZEROWICZ Joseph	Physique
LAURENT Pierre	Mathématiques appliquées
DE LEIRIS Joël	Biologie
LLIBOUTRY Louis	Géophysique
LOISEAUX Jean-Marie	Sciences nucléaires I.S.N.
LOUP Jean	Géographie
MACHE Régis	Physiologie végétale
MAYNARD Roger	Physique du solide
MICHEL Robert	Minéralogie et pétrographie (géologie)
MOZIERES Philippe	Spectrométrie - Physique
OMONT Alain	Astrophysique
OZENDA Paul	Botanique (biologie végétale)
PAYAN Jean-Jacques (détaché)	Mathématiques pures
PEBAY PEYROULA Jean-Claude	Physique
PERRIAUX Jacques	Géologie
PERRIER Guy	Géophysique
PIERRARD Jean-Marie	Mécanique
RASSAT André	Chimie systématique
RENARD Michel	Thermodynamique
RICHARD Lucien	Biologie végétale
RINAUDO Marguerite	Chimie CERMAV
SENGEL Philippe	Biologie animale
SERGERAERT Francis	Mathématiques pures
SOUTIF Michel	Physique
VAILLANT François	Zoologie
VALENTIN Jacques	Physique nucléaire I.S.N.
VAN CUTSEN Bernard	Mathématiques appliquées
VAUQUOIS Bernard	Mathématiques appliquées
VIALON Pierre	Géologie

PROFESSEURS DE 2^{ème} CLASSE

ADIBA Michel	Mathématiques pures
ARMAND Gilbert	Géographie

.../...

AURIAULT Jean-Louis	Mécanique
BEGUIN Claude (M.)	Chimie organique
BOEHLER Jean-Paul	Mécanique
BOITET Christian	Mathématiques appliquées
BORNAREL Jean	Physique
BRUN Gilbert	Biologie
CASTAING Bernard	Physique
CHARDON Michel	Géographie
COHENADDAD Jean-Pierre	Physique
DENEUVILLE Alain	Physique
DEPASSEL Roger	Mécanique des fluides
DOUCE Roland	Physiologie végétale
DUFRESNOY Alain	Mathématiques pures
GASPARD François	Physique
GAUTRON René	Chimie
GIDON Maurice	Géologie
GIGNOUX Claude (M.)	Sciences nucléaires I.S.N.
GUITTON Jacques	Chimie
HACQUES Gérard	Mathématiques appliquées
HERBIN Jacky	Géographie
HICTER Pierre	Chimie
JOSELEAU Jean-Paul	Biochimie
KERCKOVE Claude (M.)	Géologie
LE BRETON Alain	Mathématiques appliquées
LONGEQUEUE Nicole	Sciences nucléaires I.S.N.
LUCAS Robert	Physiques
LUNA Domingo	Mathématiques pures
MASCLE Georges	Géologie
NEMOZ Alain	Thermodynamique (CNRS - CRTBT)
OUDET Bruno	Mathématiques appliquées
PELMONT Jean	Biochimie
PERRIN Claude (M.)	Sciences nucléaires I.S.N.
PFISTER Jean-Claude (détaché)	Physique du solide
PIBOULE Michel	Géologie
PIERRE Jean-Louis	Chimie organique
RAYNAUD Hervé	Mathématiques appliquées
ROBERT Gilles	Mathématiques pures
ROBERT Jean-Bernard	Chimie physique
ROSSI André	Physiologie végétale
SAKAROVITCH Michel	Mathématiques appliquées
SARROT REYNAUD Jean	Géologie
SAXOD Raymond	Biologie animale
SOUTIF Jeanne	Physique
SCHOOL Pierre-Claude	Mathématiques appliquées
STUTZ Pierre	Mécanique
SUBRA Robert	Chimie
VIDAL Michel	Chimie organique
VIVIAN Robert	Géographie

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Président : Daniel BLOCH
Vice-Présidents : B. BAUDELET
H. CHERADAME
R. CARRE
J.M. PIERRARD

Année universitaire 1984-1985

Professeurs des Universités

E.N.S.E.E.G.

BESSON	Jean	LOUCHET	François
BONNETAIN	Lucien	PARIAUD	Jean-Charles
BONNIER	Etienne	RAMEAU	Jean-Jacques
DURAND	François	SOHM	Jean-Claude
GUYOT	Pierre	SOUQUET	Jean-Louis

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

BARIBAUD	Michel	GENTY	Pierre
BLIMAN	Samuel	GUERIN	Bernard
BUYLE BODIN	Maurice	POUPOT	Christian
CHENEVIER	Pierre	SERMET	Pierre
COHEN	Joseph	ZADWORNÝ	François
COUMES	André		

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

BARRAUD	Alain	JOUBERT	Jean-Claude
BAUDELET	Bernard	JOURDAIN	Geneviève
BLOCH	Daniel	LACOUME	Jean-Louis
BRISSONNEAU	Pierre	LONGEQUEUE	Jean-Pierre
CAVAIGNAC	Jean-François	MASSELOT	Christian
CHARTIER	Germain	MORET	Roger
CHERUY	Arlette	PAUTHENET	René
DURAND	Jean-Louis	PERRET	René
FELICI	Noël	PERRET	Robert
FOULARD	Claude	POLOUJADOFF	Michel
GAUBERT	Claude	SABONNADIÈRE	Jean-Claude
IVANES	Marcel	SCHLENKER	Claire
JALINIER	Jean-Michel	SCHLENKER	Michel
JAUSSAUD	Pierre		

E.N.S.H.G.

BOIS	Philippe	LESPINARD	Georges
BOUVARD	Maurice	MOREAU	René
LESIEUR	Marcel	PIAU	Jean-Michel

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

ANCEAU	François	MOSSIERE	Jacques
FONLUPT	Jean	ROBERT	François
LATOMBE	Jean-Claude	SAUCIER	Gabrielle
MAZARE	Guy	VEILLON	Gérard

U.E.R.M.C.P.P.

CHERADAME	Hervé	RENAUD	Maurice
CHIAVERINA	Jean	ROBERT	André
GANDINI	Alessandro	SILVY	Jacques

Professeurs Associés

BLACKWELDER	Ronald	ENSHG
HAYASHI	Hirashi	ENSIEG
PURDY	Gary	ENSEEG

Professeurs à l'Université des Sciences Sociales (Grenoble II)

BOLLIET	Louis
CHATELIN	Françoise

Chercheurs du C.N.R.S.

Directeurs de recherche :

CARRE	René
FRUCHARD	Robert
JORRAND	Philippe
VACHAUD	Georges

Maître de recherche :

ALLIBERT	Michel	JOUD	Jean-Charles
ANSARA	Ibrahim	KAMARINOS	Georges
ARMAND	Michel	KLEITZ	Michel
BINDER	Gilbert	LANDAU	Ioan-Dore
BORNARD	Guy	LASJAUNIAS	Jean-Claude
DAVID	René	MERMET	Jean
DESPORTES	Jacques	MUNIER	Jacques
DRIOLE	Jean	PIAU	Monique
GIGNOUX	Damien	PORTESEIL	Jean-Louis
GIVORD	Dominique	THOLENCE	Jean-Louis
GUELIN	Pierre	VERDILLON	André
HOPFINGER	Emile	SUERY	Michel

Personnalités habilitées à diriger des travaux de recherche
(Décision du conseil scientifique)

E.N.S.E.E.G.

ALLIBERT	Colette	HAMMOU	Abdelkader
BERNARD	Claude	MALMEJAC	Yves (CENG)
BONNET	Roland	MARTIN GARIN	Régina
CAILLET	Marcel	NGUYEN TRUONG	Bernadette
CHATILLON	Catherine	RAVAINE	Denis
CHATILLON	Christian	SAINFORT	(CENG)
COULON	Michel	SARRAZIN	Pierre
DIARD	Jean-Paul	SIMON	Jean-Paul
EUSTATHOPOULOS	Nicolas	TOUZAIN	Philippe
FOSTER	Panayotis	URBAIN	Georges(ODEILLO)
GALERIE	Alain		

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

BARIBAUD	Michel	DOLMAZON	Jean-Marc
BOREL	Joseph	HERAULT	Jeanny
CHOVET	Alain	MONLLOR	Christian
CHEHIKIAN	Alain		

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

BORNARD	Guy	LEJEUNE	Gérard
DESCHIZEAUX	Pierre	MAZUER	Jean
GLANGEAUD	François	PERARD	Jacques
KOFMAN	Walter	REINISCH	Raymond

E.N.S.H.G.

ALEMANY	Antoine	OBLED	Charles
BOIS	Daniel	ROWE	Alain
DARVE	Félix	VAUCLIN	Michel
MICHEL	Jean-Marie	WACK	Bernard

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

BERT	Didier	DELLA DORA	Jean
CALMET	Jacques	FONLUPT	Jean
COURTIN	Jacques	SIFAKIS	Joseph
COURTOIS	Bernard		

U.E.R.M.C.P.P.

CHARUEL	Robert
---------	--------

C.E.N.G.

CADET	Jean	NIFENECKER	Hervé
COEURE	Philippe (LETI)	PERROUD	Paul
DELHAYE	Jean-Marc (STT)	PEUZIN	Jean-Claude(LETI)
DUPUY	Michel (LETI)	TAIEB	Maurice
JOUVE	Hubert (LETI)	VINCENDON	Marc
NICOLAU	Yvan (LETI)		

Laboratoires extérieurs

C.N.E.T.

DEMOULIN
DEVINE
GERBER

Eric
R.A.B.
Roland

MERCKEL
PAULEAU

Gérard
Yves

I.N.S.A. Lyon

GAUBERT

C.

ECOLE NATIONALE SUPERIEURE DES MINES DE SAINT-ETIENNE

Directeur : M. M. MERMET
Directeur des Etudes et de la formation : M. J. LEVASSEUR
Directeur des Recherches : M. J. LEVY
Secrétaire Général : M^{le} M. CLERGUE

Professeurs de la 1ère Catégorie

COINDE	Alexandre	Gestion
GOUX	Claude	Metallurgie
LEVY	Jacques	Métallurgie
LOWYS	Jean-Pierre	Physique
MATHON	Albert	Gestion
RIEU	Jean	Mécanique-Résistance des matériaux
SOUSTELLE	Michel	Chimie
FORMERY	Philippe	Mathématiques Appliquées

Professeurs de 2ème catégorie

HABIB	Michel	Informatique
PERRIN	Michel	Géologie
VERCHERY	Georges	Matériaux
TOUCHARD	Bernard	Physique industrielle

Directeur de recherche

LESBATS	Pierre	Métallurgie
---------	--------	-------------

Maîtres de recherche

BISCONDI	Michel	Métallurgie
DAVOINE	Philippe	Géologie
FOURDEUX	Angeline	Métallurgie
KOBYLANSKI	André	Métallurgie
LALAUZE	René	Chimie
LANCELOT	Francis	Chimie
LE COZE	Jean	Métallurgie
THEVENOT	François	Chimie
TRAN MINH	Canh	Chimie

Personnalités habilitées à diriger des travaux de recherche

DRIVER	Julian	Métallurgie
GUILHOT	Bernard	Chimie
THOMAS	Gérard	Chimie

Professeur à l'UER de Sciences de Saint-Etienne

VERGNAUD	Jean-Maurice	Chimie des Matériaux & chimie industrielle
----------	--------------	--

Je tiens à remercier:

Monsieur Gérard Veillon, Directeur de l'ENSIMAG, de s'être intéressé à mon travail, et de m'avoir fait l'honneur de présider le jury.

Monsieur Jean-Claude Latombe, Professeur à l'ENSIMAG, qui a su créer l'environnement scientifique, matériel, et humain, dans lequel cette recherche a pu se dérouler, et qui m'a orienté et influencé tout au long de ce travail.

Messieurs Olivier Faugeras, Directeur scientifique à l'INRIA, et Roger Mohr, Professeur à l'INPL, d'avoir bien voulu juger ce travail.

Messieurs James L. Crowley, Professeur à l'ENSIMAG, et Jacques Mossière, Professeur à l'ENSIMAG, d'avoir accepté de faire partie du jury.

C'est avec le plus grand plaisir que je remercie les membres de l'équipe pour l'enthousiasme, le génie, et l'atmosphère chaleureuse qu'ils créent tous les jours, dans des conditions souvent difficiles; et ceux qui, dans le groupe vision, partagent mes intérêts: Paul-Louis Borianne, Yves Demazeau, Jose-Luis Gordillo, Gérard Mézin, Thierry Mondot, Viviane Morille.

Je remercie l'équipe du service de reproduction pour son travail soigné et efficace.

Augustin Lux

PLAN

INTRODUCTION

PARTIE I:

LA VISION PAR ORDINATEUR - PRESENTATION ET REFLEXIONS

1 VISION PAR ORDINATEUR ET INTELLIGENCE ARTIFICIELLE - PREMIERES DEFINITIONS

- 1.1 Définition intuitive de la VO
- 1.2 La vision dans le contexte image et ordinateur
- 1.3 Vision réflexe
 - 1.3.1 Rôle de la vision réflexe
 - 1.3.2 Comparaison avec la vision générale
- 1.4 Présentation de l'IA
 - 1.4.1 Le rôle de l'IA
 - 1.4.2 Structure type d'un système d'IA
 - 1.4.3 Comparaison avec la programmation algorithmique
- 1.5 IA et VO

2 STRUCTURES ET SYSTEMES POUR LA VO - UN TOUR D'HORIZON

- 2.1 Le debut de la vision et le monde des blocs
 - 2.1.1 Préhistoire: Roberts
 - 2.1.2 Le travail de Guzmán
 - 2.1.3 Problèmes d'étiquetage
 - 2.1.3.1 Les étiquettes de Huffman et Clowes

- 2.1.3.2 La généralisation de Waltz
- 2.2 Approches fondamentales - structuration
 - 2.2.1 Le paradigme "segmentation - interprétation"
 - 2.2.2 La théorie de D.Marr
 - 2.2.3 L'approche système selon T.Binford
- 2.3 Description de quelques systèmes
 - 2.3.1 Le travail de Shirai
 - 2.3.2 Analyse de régions: MSYS et ARGOS
 - 2.3.3 VISIONS
 - 2.3.4 ACRONYM
 - 2.3.5 Le système de Rochester
 - 2.3.6 Le travail de Tsuji et Nakano
 - 2.3.7 Le travail de Nagao et Matsuyama
 - 2.3.8 Systèmes industriels
 - 2.3.8.1 Le travail de Perkins
 - 2.3.8.2 Le travail de Yachida et Tsuji
 - 2.3.8.3 La méthode LFF
- 2.4 Résumé et conclusion

3 CONCEPTS DE BASE POUR NOTRE TRAVAIL

- 3.1 Problèmes d'architecture d'un système de VO
- 3.2 Structures de données intermédiaires
- 3.3 Problèmes algorithmiques fermés
- 3.4 Structures de contrôle

PARTIE II:
DE LA CONSTRUCTION D'INDICES VISUELS

1 INTRODUCTION

- 1.1 Approche et motivation
- 1.2 La généralité de la notion de ligne
- 1.3 Géométrie digitale

2 EXTRACTION DE LIGNES DE CONTRASTE

- 2.1 Image gradient
 - 2.1.1 Position du problème
 - 2.1.2 Intérêt de l'image gradient
 - 2.1.3 Calcul de l'image gradient
 - 2.1.4 Interprétation de l'image gradient, profils
 - 2.1.5 Relation avec d'autres travaux
 - Marr et Hildreth
 - Canny
- 2.2 Suivi de lignes de contraste

3 ANALYSE DE LIGNES

- 3.1 Motivation
- 3.2 Courbure discrète
- 3.3 Une étude algorithmique: Recherche de segments de droites
 - 3.3.1 Présentation du problème
 - 3.3.2 L'algorithme de Roberts
 - 3.3.3 Approximation aux moindres carrés
 - 3.3.4 Découpage récursif
 - 3.3.5 Reconnaissance syntaxique
 - 3.3.6 Découpage itératif
 - 3.3.7 Analyse de courbure

- 3.3.8 La méthode de Hough
- 3.3.9 Applications
- 3.3.10 Conclusion

4 LE SYSTEME CAIMAN: REALISATION ET APPLICATIONS

- 4.1 Présentation
- 4.2 Composants du système interactif
- 4.3 L'aspect "machine spécialisée"
- 4.4 Niveaux d'interprétation dans le monde des blocs
- 4.5 Localisation d'objets dans un montage complexe
- 4.6 Réglage d'un terminal à écran cathodique

PARTIE III: **SUR LA RECONNAISSANCE D'OBJETS**

1 UNE FORMALISATION DE LA NOTION DE RECONNAISSANCE

- 1.1 Aspects intuitifs du problème
 - 1.1.1 Présentation du problème
 - 1.1.2 L'aspect "visibilité partielle"
 - 1.1.3 Problèmes liés à la visibilité partielle
- 1.2 Définitions préliminaires
 - 1.2.1 Indices visuels
 - 1.2.2 Modèles
 - 1.2.3 Indices image
 - 1.2.4 Correspondances
- 1.3 Hypothèses
 - 1.3.1 Définition
 - 1.3.2 Reconnaissance d'un objet
 - 1.3.3 Transformations géométriques

- 1.4 Espaces de recherche
 - 1.4.1 Le graphe des hypothèses
 - 1.4.2 Hypothèses équivalentes
 - 1.4.3 Le graphe réduit
- 1.5 Algorithmes de reconnaissance
 - 1.5.1 Recherche dans le graphe d'hypothèses
 - 1.5.2 Recherche par prédiction
- 1.6 Généralisations
 - 1.6.1 Différents types d'IVs
 - 1.6.2 Cas de plusieurs objets

2 MISE EN OEUVRE - APPLICATIONS

- 2.2.1 Correspondance approximative
- 2.2 L'interaction interprétation - description
 - 2.2.1 Motivation
 - 2.2.2 La prédiction - vérification
- 2.3 Premier exemple: le système PVV
 - 2.3.1 Description
 - 2.3.2 Le module de description et le dialogue PVV - CAIMAN
 - 2.3.3 Evaluation
- 2.4 Comparaison avec d'autres systèmes
- 2.5 Deuxième exemple: interprétation de données 3-D
- 2.6 Comparaison avec d'autres algorithmes de reconnaissance
 - 2.6.1 Recherche de cliques maximales
 - 2.6.2 La méthode de Hough généralisée
 - 2.6.3 Propagation de contraintes

3 UTILISATION DE CONNAISSANCES: LA GENERATION DE PLANS

- 3.1 Motivation
- 3.2 Exemple: Le système de Garvey
 - 3.2.1 Vue d'ensemble
 - 3.2.2 Modélisation

- 3.2.3 Description de l'image
- 3.2.4 Génération et exécution de plans
 - 3.2.4.1 Vue d'ensemble
 - 3.2.4.2 Génération de plans
 - 3.2.4.3 Valuation d'un plan
 - 3.2.4.4 Exécution d'un plan
 - 3.2.4.5 Exemple: Recherche d'une chaise
- 3.3 Discussion
 - 3.3.1 Comparaison avec la prédiction - vérification
 - 3.3.2 Comparaison avec la génération de plans d'actions
 - 3.3.3 Généralisation

4 CONCLUSION SUR LA RECONNAISSANCE D'OBJETS

- 4.1 Adéquation au problème physique
- 4.2 L'identification est un problème d'IA

CONCLUSION GENERALE

RESUME

Cette thèse présente un travail dans les domaines de la Vision par Ordinateur et de l'Intelligence Artificielle mené dans l'équipe Intelligence Artificielle et Robotique du Laboratoire LIFIA (auparavant Laboratoire IMAG).

L'exposé est composé de trois parties.

La première partie donne un aperçu des approches théoriques et des réalisations existantes en VO intégrant des concepts de l'Intelligence Artificielle. Nous constatons à travers cet aperçu que la VO est une discipline jeune, fourmillante d'activités, mais sans véritables bases théorique et pratique bien établies. Ce constat d'anarchie relative est le point de départ de notre travail personnel, exposé dans les parties II et III, dans lequel nous proposons une structuration d'une partie du domaine de la VO et une architecture pour un système de VO, de façon à créer une base exploitable pour diverses applications et recherches. L'originalité de l'architecture proposée tient à deux éléments:

** L'intégration du concept de structure de contrôle, venant de l'IA, dans un système dans lequel les connaissances sont essentiellement représentées par un ensemble de procédures organisées hiérarchiquement.*

** L'étude systématique des problèmes algorithmiques liés au sous-ensemble de la vision considéré.*

Nous avons donc combiné une approche par l'algorithmique classique et une approche par l'IA, approches développées dans les deux parties suivantes de l'exposé.

La deuxième partie traite de problèmes de vision très proches de l'image, souvent appelés "vision de bas niveau". Nous présentons l'architecture d'un système de vision de bas niveau et l'étude algorithmique détaillée des problèmes soulevés, en particulier du problème de découpage d'une ligne digitale en segments de droites. L'aspect expérimental de ce travail a été important. Il a amené à la définition d'un système logiciel de base pour la VO, le système CAIMAN. Quelques applications caractéristiques, réalisées par des chercheurs du laboratoire, sont décrites pour étayer la validité de notre approche.

La troisième partie étudie les problèmes posés par l'interprétation d'indices visuels en termes de modèles symboliques. Nous donnons une formalisation de la notion de reconnaissance d'objets et nous analysons sa complexité algorithmique. Cette formalisation amène naturellement à l'introduction des concepts de l'Intelligence Artificielle: l'identification d'un objet partiellement visible fait apparaître le besoin d'un raisonnement symbolique sophistiqué, et la complexité exponentielle du problème doit être maîtrisée par des structures de contrôle adaptées. Dans ce cadre nous étudions les stratégies de prédiction - vérification et de génération de plan.

SYNOPTIQUE

PARTIE I: LA VISION PAR ORDINATEUR - PRESENTATION ET REFLEXIONS

CHAPITRE 1

VISION PAR ORDINATEUR ET INTELLIGENCE ARTIFICIELLE - PREMIERES DEFINITIONS

Ce chapitre introduit la Vision par Ordinateur et l'Intelligence Artificielle en tant que domaines scientifiques, en précisant leurs objectifs, et en esquissant les principes de base.

Les deux domaines sont présentés pour des raisons bien différentes.

- La VO est une discipline jeune aux contours mal perçus. Notre introduction, donnée dans les paragraphes 1.1 à 1.3, cherche d'abord à fixer un cadre général et à situer la VO par rapport à des disciplines voisines, qui sont le Traitement d'Images, la Reconnaissance de Formes, le Graphique. Ensuite nous faisons allusion à la vision dans le contexte de l'automatisme avec la vision réflexe, puis nous définissons les liens généraux entre VO et IA.

- L'IA est un domaine de recherche de grande actualité sur lequel il existe de nombreux ouvrages scientifiques. Cependant, aussi paradoxal que cela puisse paraître, il n'existe pas de définition généralement admise de ce domaine, et le terme IA couvre des notions très variées selon les auteurs et le contexte (recherche - industrie, logiciel - matériel, théorie - applications). Pour cette raison nous développons au §1.4 notre point de vue des bases de l'IA, tel qu'il émerge actuellement dans l'équipe au terme de quinze ans d'activité dans ce domaine. Les notions introduites dans ce paragraphe seront utilisées dans la suite à divers endroits de la première partie, et surtout dans la troisième partie de ce rapport.

Le dernier paragraphe de ce chapitre intègre la VO et l'IA dans un seul schéma fonctionnel, mettant en évidence les relations au plus haut niveau entre ces deux domaines.

CHAPITRE 2

STRUCTURES ET SYSTEMES POUR LA VO - UN TOUR D'HORIZON

Nous ne savons pas - du moins à l'heure actuelle - dériver les spécifications détaillées d'un système de vision automatique à partir des principes généraux décrits au premier chapitre. Ce deuxième chapitre approche la VO de façon complémentaire: par un tour d'horizon de quelques systèmes de VO existants et qui ont, d'une façon ou d'une autre, influencé notre travail. Ce tour d'horizon montre la grande variété des travaux en VO. Nous remontons d'abord aux origines de la VO avec la vision dans le monde des blocs (§2.1). Sur la base de ces systèmes nous abordons la difficile discussion sur l'architecture générale d'un système de vision, dans laquelle nous évoquons notamment la proposition de D.Marr et la classification de T.Binford. Le gros du chapitre est ensuite constitué par la description de nombreux systèmes que nous avons classés en trois groupes selon leur finalité: recherche d'une faculté de vision générale (§2.3.1-§2.3.4), applications spécialisées de pointe (§2.3.5-§2.3.7), et systèmes à vocation industrielle (§2.3.8). Dans un dernier paragraphe nous suggérons une classification de ces groupes par rapport aux critères de généralité, expertise, et performance.

CHAPITRE 3

CONCEPTS DE BASE POUR NOTRE TRAVAIL

Le chapitre 3 amorce la jonction entre l'étude générale de la première partie et le travail personnel exposé dans la deuxième partie de cette rédaction, en présentant des concepts pour l'architecture d'un système de vision à partir d'une réflexion (1) sur les systèmes décrits précédemment, (2) sur les relations entre l'IA et l'algorithmique, et (3) sur une première expérience avec un système de vision. Nous identifions trois dimensions selon lesquelles l'architecture d'un système doit être optimisée, et que nous appelons structures de données (§3.2), problèmes algorithmiques fermés (§3.3), et structures de contrôle (§3.4). Nous introduisons une forme de coroutines, les algorithmes incrémentaux, qui combinent une grande efficacité au sens de l'algorithmique (faible complexité) avec la possibilité d'un contrôle explicite au sens de l'IA. La plupart des exemples utilisés dans ce chapitre ont un rapport direct avec l'étude détaillée de la deuxième partie.

PARTIE II:

DE LA CONSTRUCTION D'INDICES VISUELS

La partie II développe les éléments d'un système de description d'images. Le rôle d'un système de description d'image est de construire une représentation de l'image pouvant être directement mise en correspondance avec des modèles par un module d'interprétation. Cette représentation, dans laquelle nous nous limitons à décrire l'image en tant que structure à deux dimensions, est construite en passant par des structures de données intermédiaires. Nous avons choisi les lignes comme éléments de base de la description, en particulier les lignes de contraste. Les problèmes que nous traitons sont donc la construction de lignes à partir d'une image, et l'analyse de lignes en vue de produire des "indices visuels", qui seront, par exemple, des segments de droites ou des angles.

CHAPITRE 1

INTRODUCTION

Le chapitre 1 justifie d'abord notre choix des lignes de contraste comme structure de données intermédiaire d'un système de VO; le paragraphe 1.1 fait ainsi suite au paragraphe 3.2 de la première partie. Le paragraphe 1.3 présente quelques aspects généraux de la géométrie digitale - la géométrie sur une grille discrète de points. Comparée à la géométrie habituelle dans un espace continu, la géométrie digitale possède des caractéristiques particulières qui ont des conséquences parfois surprenantes, et qui marquent les problèmes de vision de bas niveau d'un cachet original.

CHAPITRE 2

EXTRACTION DE LIGNES DE CONTRASTE

Le chapitre 2 décrit l'extraction de lignes de contraste. Nous essayons d'être concis et constructif à propos de ce sujet (qui a déjà reçu trop d'attention par rapport à son importance à l'intérieur d'un système de vision). Nous définissons une ligne de contraste comme une crête dans l'image gradient, ce qui amène à des algorithmes simples et satisfaisants, compte tenu des possibilités de notre matériel. Quelques photographies illustrent le travail d'expérimentation, et nous établissons brièvement le rapport entre nos travaux et quelques résultats récents venant d'une approche mathématiquement plus fouillée.

CHAPITRE 3

ANALYSE DE LIGNES

Le chapitre 3 étudie le découpage d'une ligne digitale en entités géométriques simples. C'est un problème de géométrie digitale, pour lequel notre objectif principal est la recherche d'algorithmes réalistes plutôt que l'étude dans l'abstrait des problèmes de géométrie. Nous résumons brièvement notre expérience avec la courbure discrète permettant, en particulier, de trouver des angles et des arcs de cercle (§3.2). L'essentiel du chapitre est ensuite consacré à l'étude détaillée des problèmes liés au type d'indice visuel le plus simple, le segment de droite (§3.3). De nombreux algorithmes avec divers variantes sont présentés, leurs propriétés géométriques et algorithmiques sont analysées. Ce paragraphe n'est pas seulement un "survey" mais contient aussi de nombreux éléments originaux. En particulier, nous caractérisons par plusieurs propriétés l'algorithme bien connu du découpage récursif (§3.3.4), et nous présentons un algorithme nouveau, le découpage itératif (§3.3.6). Le paragraphe 3.3.8 analyse l'intérêt de ces algorithmes pour trois problèmes d'application, et nous terminons par quelques remarques bibliographiques.

CHAPITRE 4

LE SYSTEME CAIMAN - REALISATION ET APPLICATIONS

Le chapitre 4 présente un système logiciel nommé CAIMAN - système Conversationnel d'Analyse d'IMAGES Numérisées - qui intègre l'ensemble des algorithmes que nous avons présentés dans les chapitres précédents dans un environnement complet pour la VO. CAIMAN est un système de vision de bas niveau avec une structure de contrôle flexible et pouvant dialoguer avec un programme d'interprétation. Les principes généraux et le contenu algorithmique du système étant décrits dans les chapitres précédents, la description que nous donnons dans les paragraphes 4.1 à 4.3 reste brève et concerne les aspects importants de la réalisation du système. Une description détaillée des différentes versions est consignée dans les manuels d'utilisation et les manuels de programmation. L'ambition du système CAIMAN étant l'utilité pratique, nous décrivons dans les paragraphes 4.4 à 4.6 trois projets d'application qu'il a permis de réaliser dans notre équipe. CAIMAN est également à la base du système PVV présenté au paragraphe III-2.3.

PARTIE III

SUR LA RECONNAISSANCE D'OBJETS

La partie III concerne le passage entre les niveaux "indice visuel" et "objet" - la reconnaissance d'objets. Comme la notion intuitive de reconnaissance d'objets recouvre une grande variété de phénomènes mentaux nous présentons d'abord une formalisation générale de la reconnaissance en termes de correspondances entre ensembles, ce qui nous permet de situer dans un cadre formel des notions comme reconnaissance, objet, visibilité partielle, indice visuel, modèle, ... Cette formalisation fournit la spécification du problème informatique de reconnaissance. Nous analysons les problèmes posés par cette spécification sur le plan algorithmique (chapitre 2) et sur le plan de l'IA (chapitre 3).

CHAPITRE 1

UNE FORMALISATION DE LA NOTION DE RECONNAISSANCE

Dans ce chapitre nous présentons le problème de reconnaissance d'objets partiellement visibles du point de vue de la complexité algorithmique; nous nous intéressons principalement à la combinatoire des structures de données, ainsi qu'au coût en temps et en place des algorithmes.

Afin de ne pas alourdir l'exposé, nous traitons d'abord le problème sous une forme épurée: reconnaissance d'objets d'un seul type d'objet, employant un seul type d'indice visuel, le segment de droite, et des correspondances géométriques exactes. Toutefois, nous n'émettons pas d'hypothèse concernant la dimension de l'univers (2D ou 3D). Il s'avère que, d'un point de vue combinatoire, cette forme simplifiée est le cas le plus défavorable. Elle permet donc de mettre en évidence les difficultés fondamentales soulevées.

Quelques généralisations simples sont discutées au paragraphe 1.6: prise en compte de différents types d'indices et de différents types d'objets.

CHAPITRE 2

MISE EN OEUVRE - APPLICATIONS

Le chapitre 2 analyse l'utilisation du formalisme de reconnaissance dans un système de VO. Il devient alors important de considérer deux aspects nouveaux - la correspondance approximative, nécessitée par la nature bruitée et imparfaite des images, et l'interaction interprétation -

description, qui donne la possibilité d'introduire des structures de contrôle ayant le potentiel de réduire de façon significative le coût des calculs. L'analyse débouche sur la définition de la stratégie de prédiction - vérification (§2.2.3). Ces développements sont illustrés à l'aide de quelques systèmes de vision existants, notamment le système PVV issu de notre équipe, puis comparés à d'autres stratégies de reconnaissance.

CHAPITRE 3

UTILISATION DE CONNAISSANCES: LA GENERATION DE PLANS

L'approche algorithmique développée dans les chapitres 1 et 2 fait apparaître une combinatoire du problème de reconnaissance, qui semble être en contradiction avec notre intuition. Nous expliquons cette contradiction par le fait que la vision humaine intègre dans la reconnaissance d'innombrables connaissances concernant la structure du monde, qui se traduisent par des contraintes pour le processus de reconnaissance. Nous considérons donc l'aspect IA de la vision. Sans proposer un modèle général pour un tel mécanisme, ce chapitre décrit des idées existantes, et les relie aux développements précédents.

CHAPITRE 4

CONCLUSION SUR LA RECONNAISSANCE D'OBJETS

Ce chapitre situe notre approche à la reconnaissance par rapport à la nature physique du problème, et par rapport à la définition de la VO donnée au premier chapitre de la partie I, puis débouche sur une proposition originale sur les rôles respectifs de la Vision par Ordinateur et de l'Intelligence Artificielle.

INTRODUCTION

LE THEME

Notre travail concerne la **Vision par Ordinateur** - la VO - et sa relation avec l'**Intelligence Artificielle** - l'IA. Il est motivé par l'intérêt de cette relation pour les deux domaines.

- Pour l'IA, car un système d'IA manipule des connaissances sur son environnement, qui proviennent en dernier ressort des capteurs. La vision constitue, par le potentiel de l'information qu'elle fournit, un capteur privilégié, que l'on doit pouvoir parfaitement intégrer dans la structure d'ensemble d'un système d'IA.

- Pour la VO, car son problème central est l'organisation sous forme de connaissances de l'énorme quantité d'informations implicites dans le signal des images. La VO trouve son plein sens seulement dans le contexte d'un système d'IA complet avec ses finalités; à un niveau plus technique, la difficulté de ce problème constitue une incitation pour l'aborder sous l'angle de l'IA.

L'APPROCHE

Nous considérons la VO comme un domaine scientifique et une discipline technique à part entière. Il nous paraît nécessaire d'insister sur ce point pour deux raisons: d'une part, à l'heure actuelle, les problèmes de vision sont encore trop souvent perçus à travers une application particulière, où les impératifs techniques empêchent une prise de conscience de la généralité et de la difficulté des problèmes. D'autre part, si des éléments de solution d'un problème de vision peuvent être apportés par plusieurs disciplines différentes, comme le traitement du signal, ou la reconnaissance de formes,

ces approches, pour autant qu'elles soient utiles et nécessaires, restent toujours partielles, et doivent être considérées dans le cadre d'une discipline plus globale.

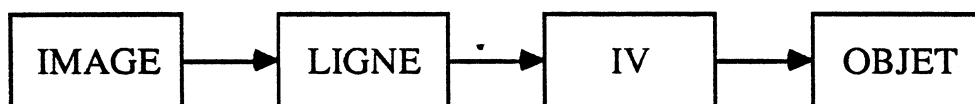
Pourtant la diversité des systèmes de VO semble contredire l'existence d'une discipline scientifique cohérente. Nous ne voyons là que la manifestation du manque de maturité de la VO. C'est une discipline jeune qu'il est intéressant de mettre en contraste avec l'analyse syntaxique et la construction d'un compilateur. Pour l'écriture d'un compilateur on dispose aujourd'hui de bases solides, tant sur le plan théorique, que sur le plan pratique, avec la théorie des langages, divers algorithmes d'analyse syntaxique, et il existe une méthodologie de la compilation. Grâce à ce savoir, un compilateur peut aujourd'hui faire l'objet d'un projet d'étudiant, alors que c'était un sujet de thèse il y a vingt ans. La VO, qui est certainement confrontée à des problèmes beaucoup plus complexes que la compilation, se trouve actuellement dans un état comparable à l'état de la compilation de cette époque: les recherches et les réalisations sont très nombreuses, mais sans véritable base commune. Chaque système est conçu et construit à partir de zéro, et cherche un point de compromis entre trois critères qu'il est impossible de satisfaire simultanément - généralité, expertise, et efficacité.

Ce constat d'anarchie relative en VO et notre conviction personnelle que la VO deviendra une discipline cohérente sont les bases du développement de notre travail. Nous étudions un système de vision selon deux aspects: **l'aspect algorithmique** et **l'aspect IA**.

* **L'aspect algorithmique** concerne d'abord la décomposition du problème global de la vision en une succession d'étapes

intermédiaires. En effet, comme un objet ne se définit pas naturellement en termes du signal capteur, mais en termes de volumes, de surfaces, de couleurs, etc., il est nécessaire d'introduire des structures de données intermédiaires entre la donnée initiale de l'image, qui est une matrice de pixels, et le résultat d'une interprétation qui est une description de la scène en termes d'objets, et de relations spatiales entre ces objets. Il est clair que le choix de cette décomposition conditionne fatalement le potentiel de l'ensemble du système en ce qui concerne et les possibilités d'application et les possibilités de réalisation. Cependant, la nature de ces structures intermédiaires, et même leur nombre, varie d'un système de VO à un autre: l'architecture globale d'un système de VO reste un problème ouvert.

Nous proposons une architecture dans laquelle nous utilisons les lignes de contraste comme structure intermédiaire charnière. Ce choix est motivé par le fait que les lignes de contraste sont les indices visuels les plus fiables au niveau d'une image (rapport signal sur bruit maximal dans l'image gradient). Le système que nous proposons construit des lignes de contraste à partir desquelles différents "indices visuels" - IV - sont élaborés. Les IVs forment le vocabulaire pour la définition de la grammaire des objets. Les règles de grammaire définissant un objet constituent le "modèle" de l'objet. Les grandes étapes de l'interprétation d'une image sont alors



* L'aspect IA concerne la structuration du système, dans laquelle des "connaissances", à caractère déclaratif, sont combinées à l'aide de "règles d'inférences" selon une "structure de contrôle". L'introduction de ce paradigme de l'IA dans le schéma ci-dessus pose alors une difficulté fondamentale. Les techniques de représentation déclarative de connaissances s'appliquent bien aux objets, à la description de propriétés d'objets - comme aux structures de haut niveau, à droite dans le schéma, mais elles s'appliquent mal aux structures plus proches de l'image, pour lesquelles une représentation procédurale est plus naturelle.

Cette difficulté se ressent bien dans les systèmes décrits auparavant, dans lesquels on trouve soit une séparation nette entre une partie IA et une partie procédurale, soit des systèmes entièrement procéduraux dans lesquels l'aspect IA est peu prononcé.

Nous surmontons cette difficulté par l'introduction d'algorithmes incrémentaux, implantés comme des co-routines. L'activation d'une telle co-routine peut être considérée comme l'application d'une règle d'inférence à des connaissances procédurales. Le séquençement des activations peut alors être soumis à une structure de contrôle par le moyen d'un mécanisme de propagation de buts dans une décomposition hiérarchique de problème.

Cette architecture nous permet donc de combiner une structure de contrôle indépendante des connaissances avec une représentation procédurale de connaissances. De ce fait, l'aspect structure de contrôle n'est plus limité aux parties de haut niveau d'un système de vision (comme cela est souvent le cas dans des systèmes antérieurs), mais englobe la totalité des niveaux, ce qui est d'autant plus important que le bas

niveau représente une grande quantité de calculs. De cette approche découle également une contrainte majeure pour l'aspect algorithmique: il nous amène à chercher des algorithmes de type incrémental.

LA DEMARCHE

Notre travail est à situer dans le contexte de l'équipe IA et Robotique du LIFIA dont le point focal, dans une perspective à long terme, est la réalisation d'un "robot intelligent". L'objectif global de nos recherches en vision est alors la construction d'un système de VO pour la robotique et intégrant des principes de l'IA. Conscients de l'impossibilité de construire un système général, et ne voulant pas créer encore un autre système trop particulier, nous avons cherché, en suivant les principes énoncés ci-dessus, à rassembler des éléments de base d'un système de VO à l'intérieur d'un environnement informatique complet. Ce système, appelé CAIMAN, a alors permis le développement d'une grande variété d'applications (tout comme un générateur automatique d'analyseurs syntaxiques facilite le développement de différents compilateurs). Grâce à une structure interne flexible il joue différents rôles dans les recherches en VO de l'équipe, et dans le développement d'applications de type industriel.

- Il fournit un environnement interactif pour le développement d'algorithmes. Un système de VO contient un grand nombre d'algorithmes formant le contexte nécessaire au développement, à la mise au point, et à l'évaluation d'un nouvel algorithme. (Pour travailler avec un algorithme particulier isolément, il faudrait simuler ses entrées - or la réalité de la vision est trop

complexe pour être simulée.) En particulier, les algorithmes contenus dans le système CAIMAN lui-même sont le résultat d'une évolution constante, longue de plusieurs années, dans ce contexte.

- Il fournit une bibliothèque de programmes pour la mise en oeuvre rapide d'une application. Pour une application donnée, on peut extraire un sous-ensemble figé du système doté seulement des fonctionnalités requises par cette application.

- Il sert de machine spécialisée dans l'analyse d'images de bas niveau. Le système CAIMAN peut être placé sous le contrôle d'un programme situé sur un autre ordinateur, permettant de réaliser des systèmes de VO repartis.

LE RAPPORT

Ce rapport est composé de trois parties.

La partie I présente la VO en tant que domaine scientifique, en donnant une vue globale du problème, et par la description d'un certain nombre de systèmes de VO, représentatifs de l'état de l'art. Cette partie débouche sur l'énoncé des principes qui ont guidé notre travail.

La partie II concerne la construction des niveaux intermédiaires à partir d'une image, la partie d'un système de VO qui est souvent appelée "vision de bas niveau". Elle comporte une analyse algorithmique détaillée des problèmes posés par la construction et l'analyse de lignes de contraste, et la description du système logiciel CAIMAN mettant en oeuvre les solutions proposées.

La partie III concerne le passage entre les niveaux "indice visuel" et "objet" - la reconnaissance d'objets. Comme la notion intuitive de reconnaissance d'objets recouvre une grande variété de phénomènes mentaux nous présentons d'abord une formalisation générale de la reconnaissance en termes de correspondances entre ensembles, ce qui nous permet de situer dans un cadre formel des notions comme reconnaissance, objet, visibilité partielle, indice visuel, modèle, ... L'étude de la complexité algorithmique de la reconnaissance ainsi définie fait apparaître une forte combinatoire dans un espace de recherche discret. Nous avons donc les conditions classiques pour faire appel aux techniques de l'IA. Nous étudions l'impact de deux concepts venant de l'IA, les stratégies de prédiction - vérification et de génération de plans.

LA LECTURE

La lecture de ce rapport ne se fait pas nécessairement de façon séquentielle de la première page à la dernière (pas plus qu'il n'a été rédigé ainsi). L'unité du sujet se situant avant tout sur le plan conceptuel, que nous avons essayé de faire ressortir dans cette introduction, les trois parties sont relativement indépendantes sur le plan technique, et les liens importants sont notés explicitement. En tournant les pages, le lecteur familier avec le monde de la VO reconnaitra les paysages, et pourra suivre à son gré quelques méandres de l'exposé. La table des matières, et le synoptique composé à partir des têtes de chapitre, faciliteront l'orientation.

PARTIE I
LA VISION PAR ORDINATEUR
PRESENTATION ET REFLEXIONS

CHAPITRE 1

VISION PAR ORDINATEUR ET INTELLIGENCE ARTIFICIELLE - PREMIERES DEFINITIONS

Ce chapitre introduit la Vision par Ordinateur et l'Intelligence Artificielle en tant que domaines scientifiques, en précisant leurs objectifs, et en esquissant les principes de base. Les deux domaines sont présentés pour des raisons bien différentes.

- La VO est une discipline jeune aux contours mal perçus. Notre introduction, donnée dans les paragraphes 1.1 à 1.3, cherche d'abord à fixer un cadre général et à situer la VO par rapport à des disciplines voisines, qui sont le traitement d'images, la reconnaissance de formes, le graphique. Ensuite nous faisons allusion à la vision dans le contexte de l'automatisme avec la vision réflexe, puis nous définissons les liens généraux entre VO et IA.

- L'IA est un domaine de recherche de grande actualité sur lequel il existe de nombreux ouvrages scientifiques. Cependant, aussi paradoxal que cela puisse paraître, il n'existe pas de définition généralement admise de ce domaine, et le terme IA couvre des notions très variées selon les auteurs et leur contexte (recherche - industrie, logiciel - matériel, théorie - applications). Pour cette raison nous développons au paragraphe 1.4 notre point de vue des bases de l'IA, tel qu'il émerge actuellement dans l'équipe au terme de quinze ans d'activité dans ce domaine. Les notions introduites dans ce paragraphe seront utilisées dans la suite à divers endroits de la première partie, et surtout dans la troisième partie de ce rapport.

Le dernier paragraphe de ce chapitre intègre la VO et l'IA dans un seul schéma fonctionnel, mettant en évidence les relations au plus haut niveau entre ces deux domaines.

1.1 DEFINITION INTUITIVE DE LA VO

La vision par ordinateur - VO - a pour objectif la construction d'un capteur visuel capable de remplir la fonction de l'oeil humain dans un système artificiel.

Cet énoncé est peu orthodoxe en tant que définition d'un domaine scientifique - il ne précise ni le type de problèmes traités en VO, ni les outils utilisés. En ce sens c'est une définition pré-scientifique faisant état de l'ambition des chercheurs.

Etant donné que la vue est d'une importance capitale pour l'intelligence humaine - l'oeil et le cerveau sont intimement liés au point qu'il est impossible de séparer la vision des autres activités intellectuelles - nous considérons que la VO doit construire un ensemble matériel - logiciel s'intégrant dans une "intelligence artificielle".

Nous allons donc préciser notre définition de la VO dans la terminologie de l'IA: le but d'un système de vision est la construction, à partir d'une image, d'une "description symbolique" de la scène représentée. La description symbolique est composée à partir des connaissances du système, et peut être conditionnée par les préoccupations courantes ("objectif") du système. Nous appelons "interprétation d'image" la construction d'une telle description.

D'où le schéma de la figure I-1

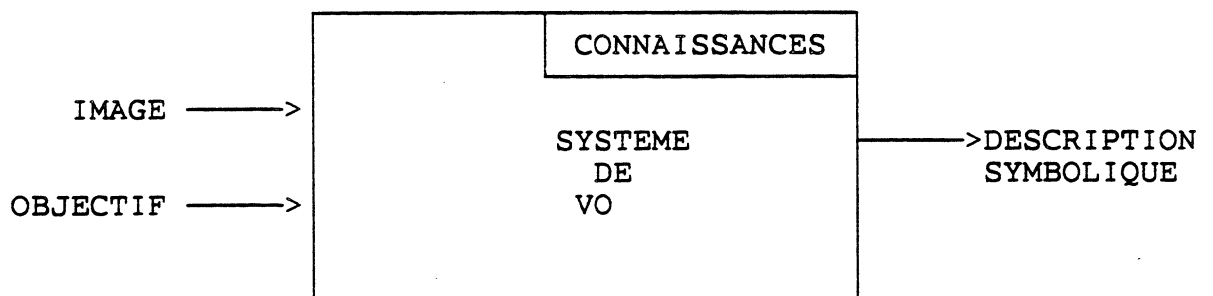


Figure I-1

1.2 LA VISION DANS LE CONTEXTE IMAGE ET ORDINATEUR

Si la définition de la VO doit être donnée dans le cadre de l'IA, la réalisation d'un système de VO dépasse largement le cadre de l'IA; elle nécessite de solides connaissances dans de nombreuses spécialités scientifiques. D'autre part, des problèmes d'interprétation d'images se posent dans un grand nombre d'applications. Pour ces raisons, et pour des raisons historiques, plusieurs branches scientifiques ont évolué dans le contexte "image et ordinateur", que nous caractérisons brièvement ci-dessous.

* Le traitement d'images.

Le traitement d'images a pour objectif la transformation d'une image en d'autres images, mettant mieux en évidence, pour l'observateur humain, certaines informations recherchées, ou tout simplement plus esthétiques, selon le schéma de la figure I-2.

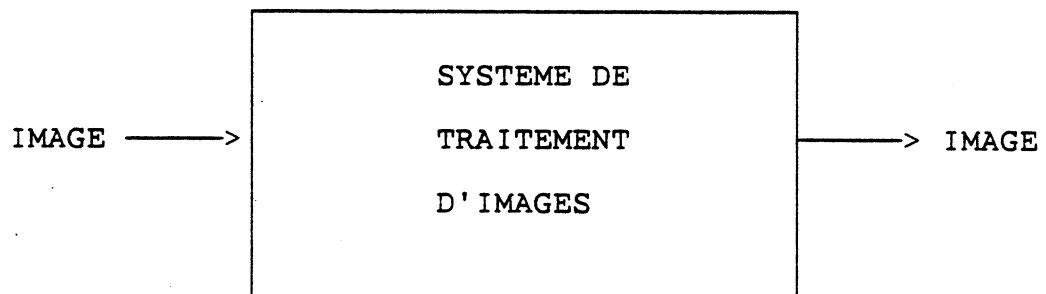


Figure I-2

On peut considérer le traitement d'images comme l'approche de l'image par les outils de traitement du signal.

Exemples:

- La mise en forme d'images aériennes avec des fausses couleurs.
- Le traitement d'images de satellite en vue d'enlever un voile de nuages, ou de redresser la perspective.
- Suppression du bruit dans des images documentaires

* La reconnaissance de formes.

Prise au sens large, la reconnaissance de formes inclut la VO. Dans un sens plus strict, la reconnaissance de formes cherche à caractériser des "formes" à l'aide d'outils mathématiques, le problème type étant celui de la reconnaissance de caractères, et correspond au schéma de la figure I-3.

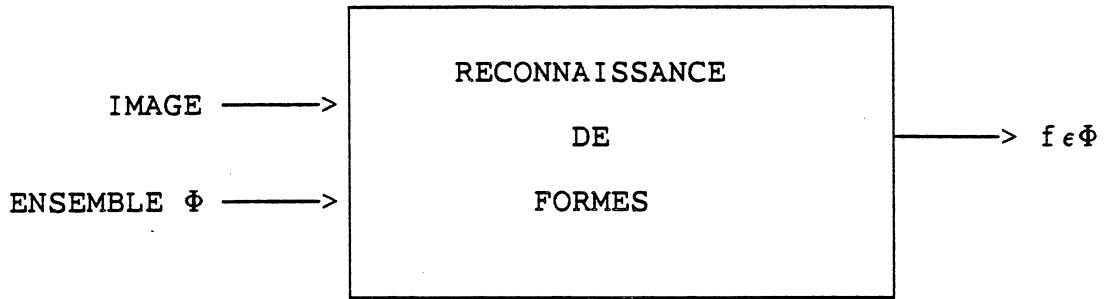


Figure I-3

Exemples:

- La classification de chromosomes.
- La reconnaissance de caractères d'imprimerie ou manuscrits.
- La classification de surfaces cultivées dans des images de satellite, selon les caractéristiques spectrales.

*** Le graphique.**

L'objectif d'un système graphique est la synthèse d'une image à partir d'informations symboliques. C'est donc l'inverse de la VO.

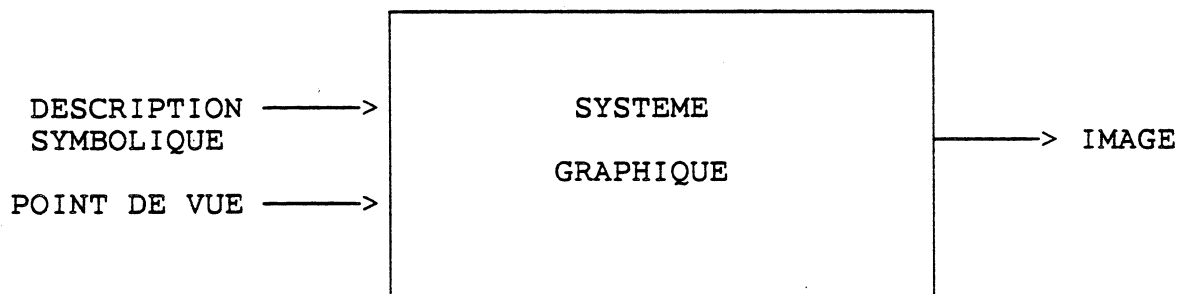


Figure I-4

Il est regrettable que la VO et le graphique soient deux milieux tout à fait séparés; ils ont plusieurs points importants en commun, en particulier le problème de représentation d'une description symbolique de scènes, et des outils mathématiques pour la modélisation géométrique.

1.3 LA VISION REFLEXE

1.3.1 Le rôle de vision réflexe

Le schéma que nous avons donné pour la VO avec la figure I-1 présente un défaut sérieux. En situant la vision en dehors de toute finalité, il nous prive de critères de jugement. Sans finalité, la description d'une scène relève de l'arbitraire - lorsqu'on demande à deux personnes différentes de décrire une image, leurs descriptions risquent d'avoir peu de choses en commun. Un point de vue radicalement différent est donné par l'approche à la vision dans le contexte de l'automatique, où le capteur vision doit fournir une information d'asservissement en boucle fermée.

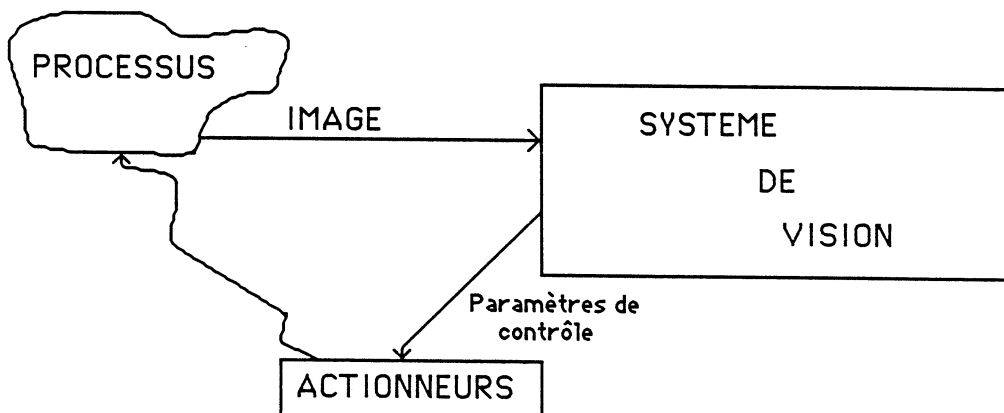


Figure I-5

Dans ce cas, le rôle de la vision consiste à extraire un certain nombre de paramètres de la scène qui sont importants pour le processus à contrôler. La construction d'une description symbolique plus générale n'est pas nécessaire. Ce type de vision existe également dans la nature, et se trouve apparemment chez des insectes.

Nous citons en exemple le modèle tout à fait remarquable du contrôle par la vision du vol de la mouche domestique, élaboré pendant une recherche longue de 14 années par l'équipe de W.E.Reichardt à Tübingen. [Reichardt et Poggio 76].

La vision de la mouche sert à élaborer les différents paramètres contrôlant le vol de l'insecte. Elle peut être décomposé en 5 sous-systèmes indépendants. Par

exemple, lorsque la mouche poursuit un "petit objet volant à proximité", les paramètres de position de l'objet poursuivi sont transformés en commande de vol "vertical" et en commande "changement de direction horizontale". La commande de direction est décrite par la formule

$$D(\Psi) + r(\Psi) \cdot \dot{\Psi}$$

où Ψ est la direction de l'objet poursuivi, et les fonctions D et r sont données par la figure I-6.

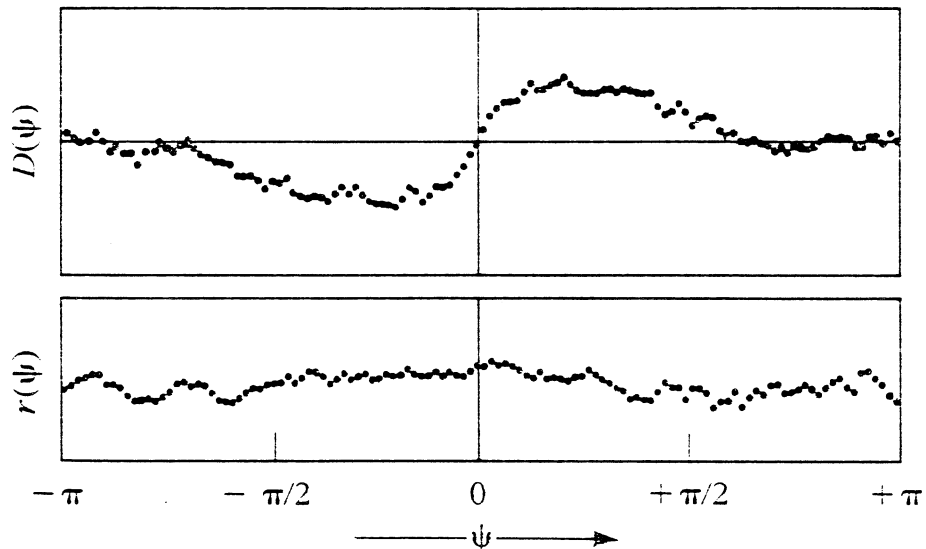


Figure I-6. Les fonctions D et r intervenant dans le réflexe visuel "poursuite d'un objet" de la mouche. ([Marr82, Reichardt76])

Nous appelons ce type de vision la "vision réflexe".

1.3.2 Comparaison avec la vision générale

Du point de vue de l'ingénieur, la vision réflexe possède plusieurs attraits. En effet, étant donné qu'elle doit répondre à un objectif technique précis pouvant être mis en équation, elle donne lieu à une réalisation spécialisée à haute performance. Dans le cas d'une mouche, le système visuel produit des réflexes rapidement - entre l'arrivée d'un signal capteur et la production de la commande angulaire correspondante se déroulent seulement 21ms - et avec un matériel d'une complexité remarquablement réduite. A l'opposé, le système de vision général que l'on trouve chez l'homme est d'une complexité immense, et les temps de réponse

au niveau symbolique sont de l'ordre de la seconde.

Toutefois, à regarder de plus près, la vision humaine s'intègre également dans un dessin plus complet que celui suggéré par la figure I-1.

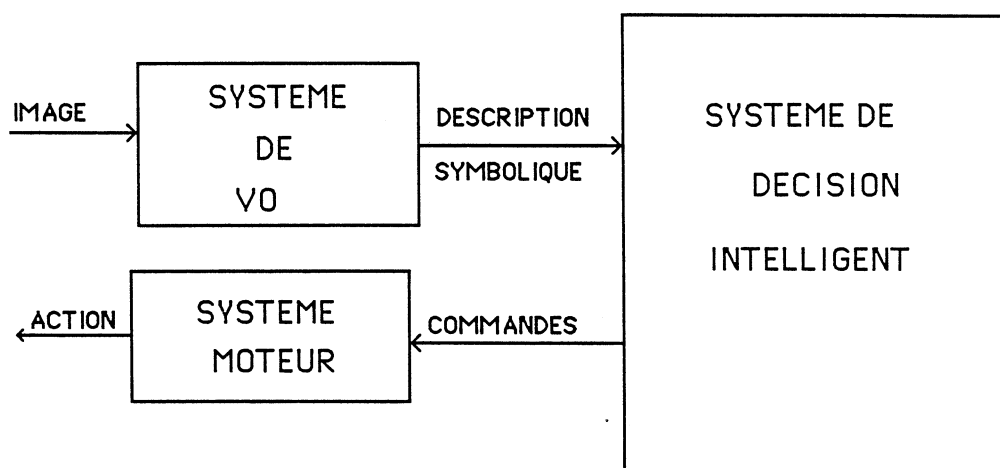


Figure I-7

La description symbolique produite par la VO sert d'entrée à un "système de décision" produisant éventuellement des actions sur l'environnement. D'un point de vue purement formel, les schémas I-5 et I-7 ne sont pas très différents (la boîte "vision" du schéma I-5 a été coupée en deux), mais nous voyons une différence qualitative importante: la vision réflexe sert dans le contexte d'une tâche pour laquelle elle a été explicitement conçue, alors que la vision générale doit servir de capteur visuel pour un ensemble de tâches vaste et ouvert, qui ne sont pas toutes "explicitement prévues" dans le système de vision. Ainsi, le système visuel humain permet d'asservir les gestes lors d'un saut d'une branche d'un arbre à un autre - tâche que l'on peut considérer comme explicitement prévue par la construction du système - et aussi les gestes pour la conduite d'une voiture.

La différence entre les paramètres fournis par la vision réflexe et la description symbolique d'une vision générale apparaît alors clairement: les paramètres doivent être pertinents pour une tâche connue, alors que la description symbolique doit fournir un maximum d'information "objective" sur la scène, de façon à permettre la solution du plus grand nombre de problèmes non connus à l'avance. Les termes "objectif" et "problème" n'ont pas de définition antérieure à

ce contexte mais ils sont définis par ce contexte. Ils peuvent être combinés dans l "axiome" suivant:

"Une information est objective dans la mesure où sa connaissance permet de résoudre des problèmes."

qui exprime le fait que le formalisme dans lequel est représentée une description symbolique définit implicitement les problèmes que le système pourra résoudre. La vision doit donc construire une description symbolique aussi riche que possible.

De façon générale, les descriptions symboliques qui nous intéressent décrivent une scène en termes plutôt qualitatifs - l'important est de connaître la nature et l'identité des objets présents, la nature de leurs mouvements et/ou actions, et des relations entre eux. Par contre, des informations plus quantitatives et numériques - position absolue, vitesse, etc. - jouent un rôle mineur ou peuvent carrément être absentes.

Il s'ensuit une autre observation importante: contrairement aux paramètres de la vision réflexe, les éléments d'une description symbolique ne rentrent pas directement dans un "algorithme d'asservissement". - ils ne peuvent pas servir simplement de données à un algorithme. Dans la mesure où il y a à résoudre un problème nouveau et non prévu, le système de décision doit faire "jouer son intelligence" pour trouver une façon d'exploiter la description symbolique pour ce problème. Nous tenons là la raison profonde qui lie entre elles les disciplines de la VO et de l'IA.

Nous avons distingué, dans ce paragraphe, la vision réflexe et la vision générale. Il n'est cependant pas notre propos de les opposer. On se convainc assez facilement que le système visuel humain ne construit pas systématiquement une description symbolique complète avant de produire une réaction. Le réflexe visuel existe aussi dans le comportement humain. Par exemple, lorsque l'on voit un objet arriver vers soi, on réalise un mouvement pour l'éviter, et cela en moins d'une seconde.

Il est facile, en effet, d'imaginer une structure pour un système de vision permettant de combiner les deux types de vision. Dans la mesure où une description symbolique représente un fort degré d'abstraction par rapport aux données capteur, il paraît impossible d'élaborer une telle structure directement à partir des données capteur. Une description symbolique est construite en plusieurs étapes, et chacune de ces étapes pourrait être "déviiée" pour servir directement à un réflexe.

1.4 PRESENTATION DE L'IA

1.4.1 Le rôle de l'IA

Nous avons évoqué à plusieurs reprises l'IA. Dans ce paragraphe nous allons préciser notre conception de l'IA par un bref développement de ses aspects principaux. Malheureusement, nous ne pouvons pas nous référer à un ouvrage existant - malgré son développement rapide et une attention particulière des médias, l'IA est une discipline mal définie, et même les références le plus citées [Barr et Feigenbaum 81, Nilsson 80, Winston 84] se contentent d'une vague paraphrase à la place d'une définition.

Comme pour la vision, la motivation originelle de l'IA vient du rêve de créer des machines à l'image de l'homme. L'"Intelligence Artificielle" doit alors amener à la création de machines dotées d'une intelligence telle qu'on la trouve chez l'homme. Nous avons ici une situation analogue à celle posée par la première définition "pré-scientifique" de la vision - l'introduction d'un domaine scientifique (ou du moins supposé tel) par une référence à une faculté humaine. Cependant, si la référence à la vision est satisfaisante intuitivement, parce que la vision humaine est une expérience constamment vécue, il n'en va pas de même pour l'intelligence: l'intelligence humaine n'est pas une faculté bien circonscrite, et elle ne correspond pas à une expérience quotidienne remplaçant une définition rigoureuse. Quelques propositions dans ce sens sont données par [Newell 82] et [Marr 77]. En attendant une telle définition nous nous limitons à une description.

Dans la discussion de l'IA il convient de distinguer trois niveaux d'abstraction que nous appelons le domaine conceptuel, l'étude formelle, et le niveau réalisation.

1) Domaine conceptuel.

Dans le domaine conceptuel il s'agit de proposer et de justifier des modèles formels de l'intelligence. Un tel modèle doit réunir dans un cadre cohérent un certain nombre de notions intuitives du langage courant liées à l'intelligence, telles que connaissance, savoir, savoir-faire, raisonnement, apprentissage, abstraction, intuition, conscience, idée, pensée, concept, mémoire.

De même que les modèles élaborés en physique incluent des appareils de mesure liant une théorie à la réalité sensorielle, les modèles de l'IA doivent contenir des références à des notions objectives et mesurables, permettant d'interpréter la théorie en termes de notre expérience vécue.

A ce niveau, l'IA se recouvre avec de nombreux domaines scientifiques: philosophie, en particulier l'épistémologie, la logique, la psychologie, la cybernétique - on parle souvent de sciences cognitives. La spécificité de l'IA parmi ces disciplines vient du fait qu'elle cherche les fondations de l'intelligence dans le cadre déterminé par l'informatique.

2) Etude formelle.

A ce niveau il s'agit d'étudier formellement, notamment avec les outils de l'algorithmique, les modèles proposés au niveau du domaine conceptuel. Ceci est le domaine le plus activement développé de l'IA. Parmi les modèles étudiés on trouve par exemple les systèmes à règles de production, les réseaux sémantiques, la programmation logique. L'étude de formalismes est ici une fin en soi, la question de son adéquation à la solution d'une application ne peut être posée à ce niveau.

3) Niveau réalisation.

Ce niveau se préoccupe de la réalisation technique des systèmes modélisés, essentiellement au moyen d'outils informatiques (au sens large). A ce niveau se place par exemple l'implantation de langages comme LISP ou PROLOG, qui a amené à des réalisations de machines spécialisées très sophistiquées.

Le non-respect d'une stricte séparation de ces trois niveaux entraîne l'apparition d'énoncés incohérents, de même que le non-respect de la séparation entre langage objet et métalangage en logique entraîne l'apparition de paradoxes. (Par exemple "Où dans un programme se situe l'intelligence?").

Grâce à son lien avec l'informatique, l'IA comporte un aspect expérimental important qui introduit des possibilités tout à fait nouvelles par rapport aux autres disciplines "cognitives" que nous avons citées. Cette relation entre IA et informatique a été bénéfique aussi pour l'informatique. L'objectif (au niveau 1) de construire des programmes possédant certains aspects de l'intelligence a induit (au niveau 2) l'étude de structures de programmes originales, pour donner ainsi naissance à une "technologie logicielle" permettant de construire des applications informatiques dans des domaines nouveaux - c'est l'apparition des "systèmes experts".

La méthodologie de la recherche en IA n'étant pas descendante - il n'y a pas, dans un premier temps, développement des concepts, puis étude de formalismes, et ensuite réalisation - il n'existe pas, dans l'état de l'art actuel, un modèle conceptuel complet satisfaisant. Comme dans les sciences physiques, il faudra une quantité suffisante d'expérimentation pour disposer d'une base à la formation des concepts.

1.4.2 Structure type d'un système d'IA

Nous développons maintenant quelques éléments d'un modèle conceptuel qui constitue un bon cadre a posteriori pour un grand nombre de systèmes d'IA existants.

L'intelligence est une qualité du comportement d'un système complexe. L'activité d'un "système intelligent" - SI - consiste à résoudre des problèmes par l'application de ses connaissances (figure I-8). Un SI possède trois composants principaux: une base de connaissances K, un système d'inférences I, et une structure de contrôle C.

$$SI = K + I + C$$

La base K contient les connaissances que le système possède sur le monde. Ces connaissances sont représentées sous une forme déclarative, c'est à dire chaque connaissance est représentée explicitement, elle peut être acquise et transmise individuellement, et être analysée par le système. L'énoncé d'une connaissance est indépendant d'un problème spécifique, il est polyvalent et peut donc servir à des fins variées.

Les règles du système d'inférence permettent de déduire des connaissances nouvelles à partir des connaissances courantes. Connaissances et inférences définissent ainsi implicitement un graphe, l'espace des connaissances accessibles, qui délimite le potentiel du système. La solution d'un problème est donnée par un noeud ou par un chemin dans ce graphe. Pendant la recherche d'une solution, la structure de contrôle doit guider le processus d'inférence, de nature combinatoire et peu organisé, vers une solution dans les meilleures conditions, en tenant compte en particulier des limites de temps d'exécution et de place de mémoire propres à chaque système.

Un exemple de structure de système correspondant bien à ce schéma se trouve avec la déduction logique dans le cadre du calcul de prédicats de premier ordre. L'ensemble de connaissances est alors un ensemble de formules, le système d'inférence utilise par exemple la règle de résolution, la structure de contrôle peut utiliser la stratégie du support.

Cette structure a été adaptée et généralisée de différentes façons dans les systèmes à base de règles de production, dont la majorité des systèmes experts de nos jours. Les connaissances - l'expertise - sont représentées par des règles, sur lesquelles opèrent des procédures d'inférence spécialisées à un type d'application donnée. La structure de contrôle est adaptée à un domaine général, comme la génération de plans d'actions, la conception/fabrication par ordinateur, ou à un domaine plus spécifique comme l'analyse de structures moléculaires. Par rapport aux systèmes entièrement fondés sur la logique, ces systèmes présentent l'avantage d'une plus

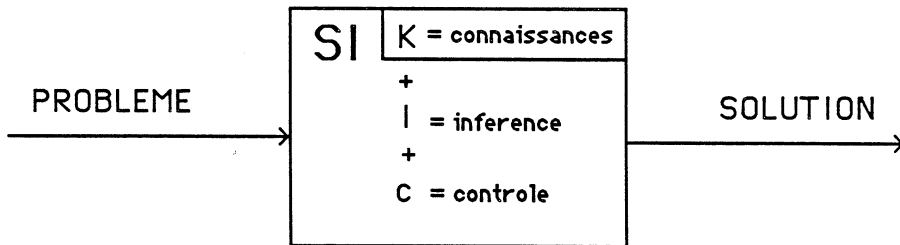


Figure I-8-a. Structure de base d'un SI.

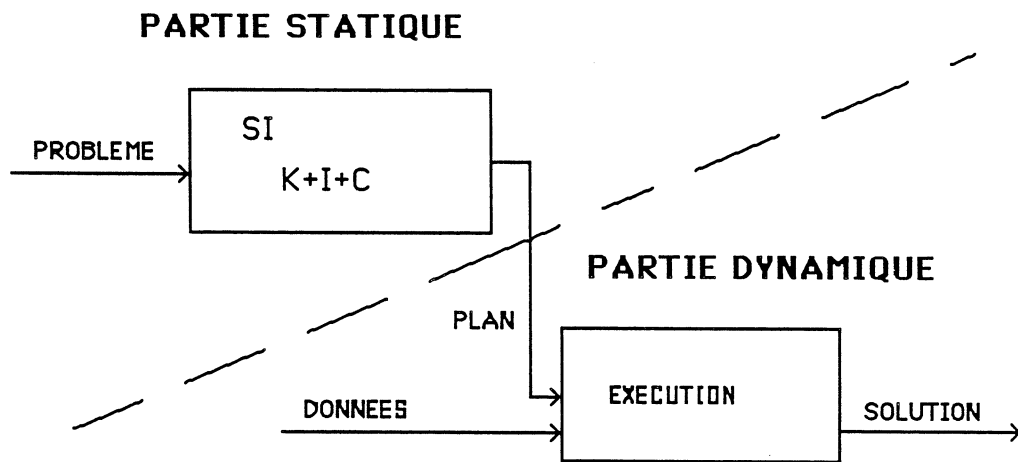


Figure I-8-b. Structure d'un SI à deux phases.

grande flexibilité et efficacité. En contrepartie, la correction et la complétude ne sont pas garanties par une théorie générale.

1.4.3 Comparaison avec l'algorithmique

Bien entendu, un système possédant la structure que nous venons de décrire est aussi un programme au sens de l'algorithmique, caractérisé par la formule

$$\text{Programme} = \text{Algorithme} + \text{Données}$$

On peut considérer la partie C comme un interpréteur, K et I comme une fonction, ou comme un programme. Ce point de vue est même tout à fait adapté dans le contexte de la programmation logique. Cependant, par rapport à la programmation algorithmique, les composants K et I possèdent des propriétés inhabituelles.

- * Les connaissances ne décrivent pas explicitement des algorithmes. Un système peut résoudre des problèmes pour lesquels l'utilisateur ne connaît pas un algorithme de solution spécifique.
- * L'expression des connaissances reste proche de la langue naturelle. La sémantique est celle du domaine d'application, tel que le connaît l'utilisateur, et non dominée par des notions informatiques.
- * Le choix des connaissances pertinentes, la façon de les appliquer, et l'ordre d'application sont décidés par le contrôle en fonction du problème à résoudre, et non prédéterminés par la structure d'un algorithme.
- * La structure de contrôle possédant un caractère général, l'utilisateur peut améliorer le système, ou augmenter son domaine de compétence par l'introduction de nouvelles connaissances (au lieu de mettre à jour des programmes).

On peut également considérer le contrôle comme un algorithme de solution d'une classe de problèmes, K et I comme des données. La généralité d'un SI implique que la classe de problèmes en question est N-P complète, et qu'elle contient de nombreux problèmes indécidables.

Le point de vue qui considère K et I comme un programme permet une classification très générale des SIs en analogie avec les notions d'interprétation et de compilation: un SI peut soit directement produire la solution d'un problème (figure I-8-a), soit produire une structure intermédiaire dont l'exécution

produira la solution. Cela est en particulier le cas de la génération de plans d'actions (figure I-8-b), dans laquelle le plan joue un rôle analogue au code objet. Cette structure, qui introduit une distinction entre génération et exécution, peut posséder des avantages importants.

- * Dans le cas d'un problème répétitif, le plan peut être exécuté plusieurs fois, alors qu'il suffit de le générer une fois.
- * La génération d'un plan peut être complexe et faire intervenir des connaissances nombreuses, ce qui demande un matériel informatique plus coûteux, alors que le plan peut s'exécuter sur un matériel simplifié. Cet aspect est particulièrement important dans une perspective temps réel.
- * Si le plan représente un algorithme, sa complexité peut être analysée. On s'efforcera alors à confiner les problèmes de N-P complétude et d'indécidabilité à la phase de génération de plans, afin de garantir la bonne fin de l'exécution du plan.

Remarques.

- L'utilisation de la génération de plans en vision sera discutée au III-4.

- Nous n'avons pas défini la génération de plans, nous l'avons seulement présentée comme exemple d'un système d'IA avec une structure de type compilation - exécution. On trouvera un traitement plus approfondi dans [Nilsson 81] et [Bessière 84].

1.5 IA et VO

La caractérisation d'un SI au paragraphe 1.4.2 est insuffisante sur au moins un point. Un SI doit vivre en interaction constante avec son environnement, alors que notre modèle est purement "cérébral". En combinant les structures des figures I-7 et I-8 nous obtenons une structure plus satisfaisante (figure I-9): le SI comprend des capteurs et des actionneurs.

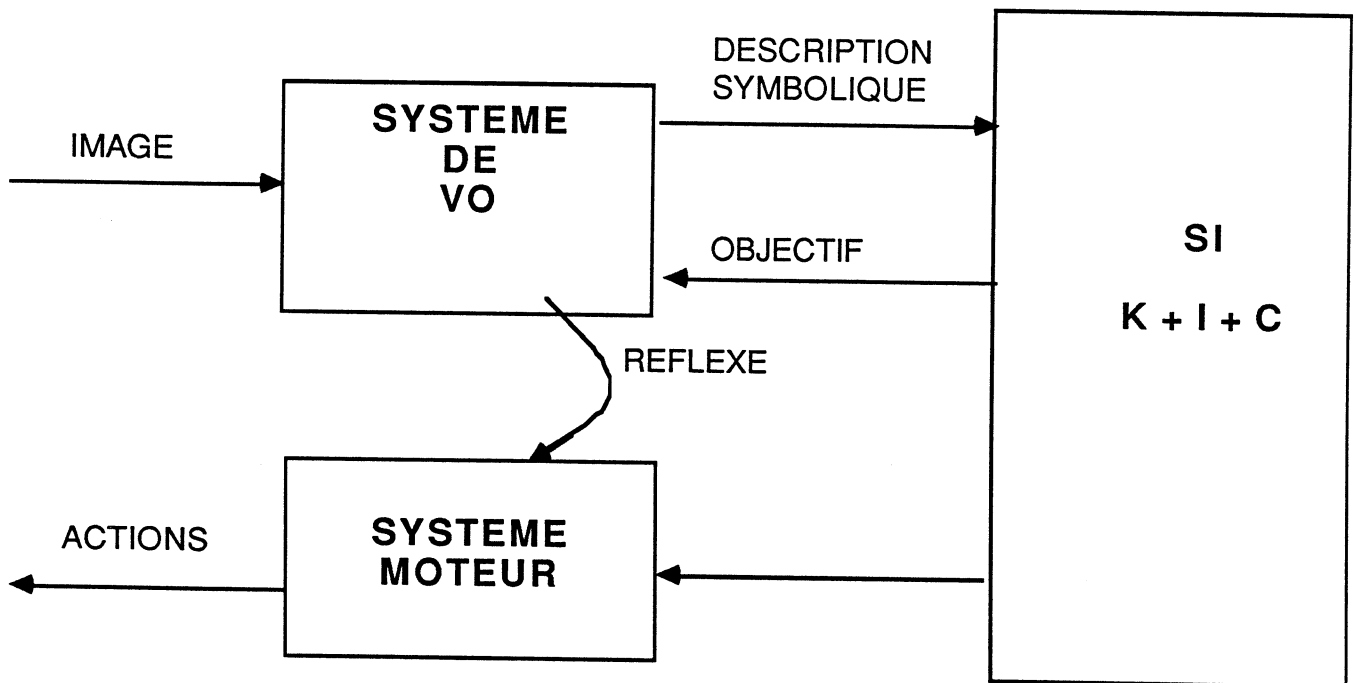


Figure I-9: Composants d'un système intelligent

Les relations entre IA et VO se placent alors à plusieurs niveaux.

1) Relation fonctionnelle au niveau 1.

En tant que capteur, la vision est une source de connaissances pour le SI. Le format d'une description symbolique est donc très généralement celui des connaissances.

2) Relation pratique au niveau 2.

Les problèmes de vision étant d'une grande complexité, il est intéressant de considérer la structure K+I+C pour un système de vision lui-même.

Dans l'abstrait, ces deux relations paraissent incontestables. Cependant, leur réalisation concrète se heurte à de nombreuses difficultés que nous présentons dans les paragraphes suivants à travers l'étude de quelques systèmes de VO existants.

CHAPITRE 2

STRUCTURES ET SYSTEMES POUR LA VO - UN TOUR D'HORIZON

Nous ne savons pas - du moins à l'heure actuelle - dériver les spécifications détaillées d'un système de vision automatique à partir des principes généraux décrits au premier chapitre. Ce deuxième chapitre approche la VO de façon complémentaire: par un tour d'horizon de quelques systèmes de VO existants et qui ont, d'une façon ou d'une autre, influencé notre travail. Ce tour d'horizon montre la grande variété des travaux en VO. Nous remontons d'abord aux origines de la VO avec la vision dans le monde des blocs (§2.1). Sur la base de ces systèmes nous abordons la difficile discussion sur l'architecture générale d'un système de vision, dans laquelle nous évoquons notamment la proposition de D.Marr et la classification de T.Binford. Le gros du chapitre est ensuite constitué par le paragraphe 2.3 dans lequel nous décrivons de nombreux systèmes que nous avons classés en trois groupes selon leurs finalités: recherche d'une faculté de vision générale (§2.3.1-§2.3.4), applications spécialisées de pointe (§2.3.5-§2.3.7), et systèmes à vocation industrielle (§2.3.8). Dans un dernier paragraphe nous suggérons une classification de ces groupes par rapport aux critères de généralité, expertise, et performance.

2.1 LE DEBUT DE LA VISION ET LE MONDE DES BLOCS

D'un point de vue historique, l'idée d'utiliser des images électroniques a été expérimentée dans des laboratoires de recherche dès le début des années 60 dans le contexte de la commande d'un robot. Les premiers travaux avaient un caractère exploratoire. Il s'agissait essentiellement de réunir les possibilités offertes par l'électronique et l'informatique de l'époque.

Comme pour beaucoup de problèmes d'IA, personne ne soupçonnait la difficulté des problèmes soulevés par une telle ambition, puisque pour l'homme, la vision est une expérience permanente

d'une facilité totale. Le premier résultat important de la recherche en VO fut ainsi une prise de conscience: que la facilité de la vision humaine est une apparence, venant d'un système très perfectionné, mais aussi très complexe; et que le fonctionnement de ce système pouvait être compris seulement au terme d'une longue recherche.

En ces débuts, le "monde des blocs" a été le domaine d'expérimentation privilégié pour la robotique, la génération de plans d'actions, l'apprentissage, et pour la vision. Dans le monde des blocs, on cherche à réduire la généralité des problèmes par une restriction sur les objets considérés. Le problème de la vision est alors la reconstruction des volumes représentés dans une image à plusieurs niveaux de gris d'une scène comportant des objets polyédriques (cubes, briques, pyramides, etc.). La figure I-10 montre quelques exemples de scènes dans le monde des blocs.

Il y a une longue lignée de systèmes de ce type caractérisée notamment par les noms de Roberts, Guzmán, Falk, Huffman, Clowes, Waltz, Shirai, Mackworth, Kanade. (Pour une description plus détaillée de ces travaux, voir le volume III du "Handbook of Artificial Intelligence" [Cohen 82]).

2.1.1 Préhistoire: Roberts

Roberts fut le premier à s'intéresser au problème de reconnaissance d'objets à trois dimensions [Roberts 65]. Il était insatisfait des travaux précédents sur la reconnaissance de formes, essentiellement la reconnaissance de caractères. Il ne pensait pas que ces travaux pouvaient ultérieurement être généralisés à la vision 3-D, mais plutôt que "l'étude de formes 2-D abstraites nous éloigne des techniques nécessaires pour la reconnaissance 3-D, au lieu de nous en rapprocher". Et il poursuit "la perception d'objets dans l'espace est un processus que l'on peut définir à partir des propriétés des transformations à trois dimensions et des lois de la nature. En analysant soigneusement ces propriétés, nous avons développé une procédure qui non seulement identifie les objets, mais détermine également leur orientation et leur position dans l'espace". Aujourd'hui encore, on ne saurait mieux définir l'objectif de la VO!

Le travail de Roberts était ambitieux, même par rapport aux critères actuels. Se limitant aux images du monde des blocs, son système opère en trois phases:

- L'image initiale est transformée en un dessin de traits. Dans cette phase, il s'agit d'extraire les lignes de contraste de l'image et de les rapprocher par des lignes brisées.

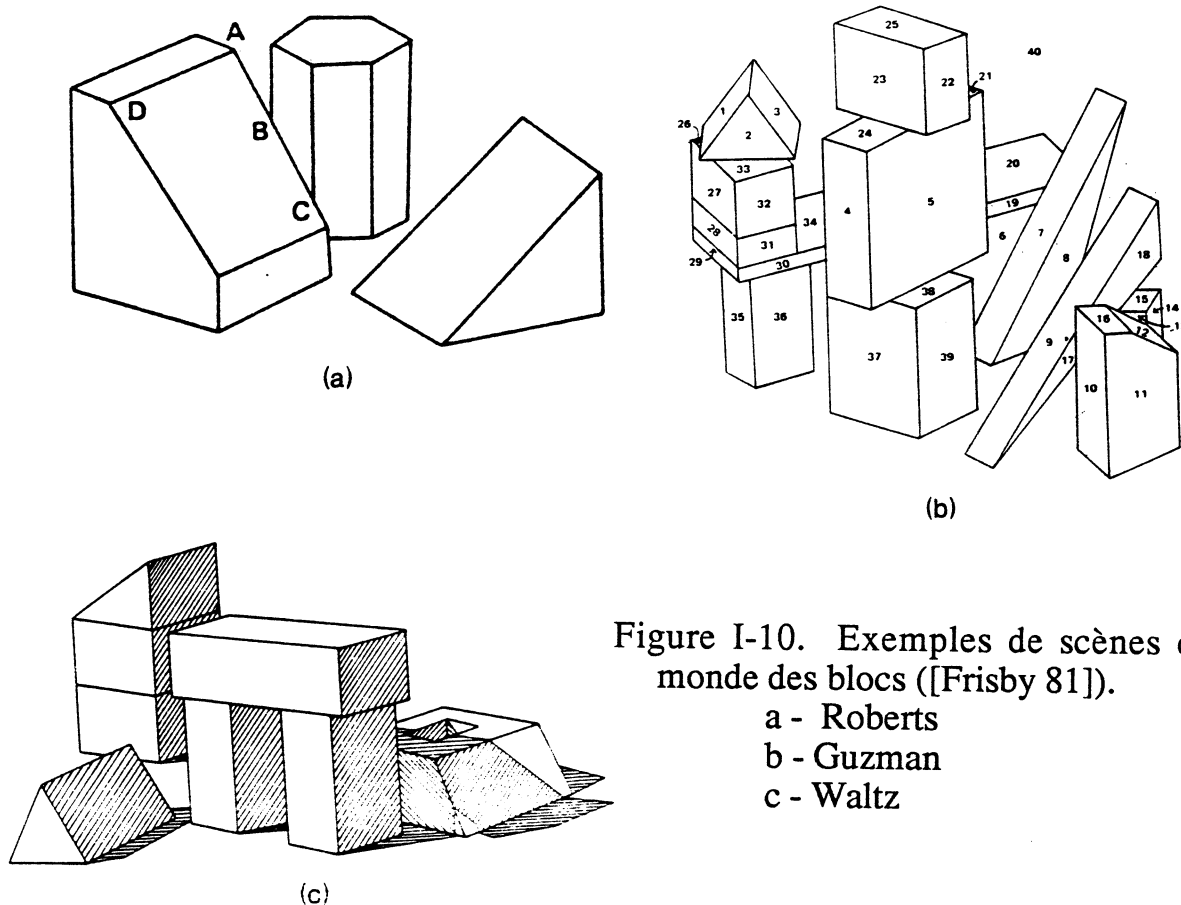


Figure I-10. Exemples de scènes dans le monde des blocs ([Frisby 81]).

a - Roberts
 b - Guzman
 c - Waltz

- Le dessin de traits est mis en correspondance avec des modèles 3-D, et une description quantitative de la scène est produite (identité et position des objets). Les modèles utilisés ne spécifient pas la taille absolue des objets. Par exemple, le modèle du cube représente tous les cubes, quel que soit leur dimension.
- La scène 3-D peut être visualisée avec un point de vue arbitraire, montrant ainsi que le système a bien construit un modèle complet de la scène.

La figure I-10a montre l'exemple d'une scène analysée.

Le trait caractéristique du travail de Roberts tient à l'approche purement numérique au problème de reconnaissance (cf. III-1). Son travail est traditionnellement cité dans la littérature sur la VO, mais en dehors du "Robert's cross operator" pour l'extraction de points de contraste (cf. §II-2.1 et figure II-6-a), aucun résultat tangible n'a été repris dans des recherches ultérieures.

2.1.2 Le travail de Guzmán

Le travail de [Guzmán 68] définit une approche à la vision très différente de celle de Roberts, une approche généralement considérée "typiquement IA" qui a inspiré de nombreuses recherches par la suite.

Guzmán s'intéresse seulement à un aspect très partiel du problème de la vision dans le monde des blocs, qui correspond à une partie de la deuxième phase du système de Roberts. Son programme SEER prend comme entrée essentiellement une liste de régions sous la forme de contours en lignes brisées - des listes d'arêtes et de noeuds - et produit en sortie une partition des régions en objets, c'est à dire le programme décide quelles régions sont à grouper dans un même objet. Cette partition se fait sans modèles d'objets, avec la seule hypothèse que le monde est un monde de blocs à sommets triédriques. Par contre, SEER n'identifie pas les objets; il ne donne ni nom, ni position. Un exemple d'une scène traitée est donné par la figure I-10b.

L'idée importante introduite par le travail de Guzmán est l'analyse "symbolique" des noeuds dans une image. La forme d'un noeud-image permet d'énoncer des contraintes locales concernant les structures 3-D physiquement possibles, qu'il faut intégrer dans une interprétation globale de façon cohérente. Guzmán introduit cette idée de manière purement ad-hoc. Son programme constitue une collection (impressionnante) d'heuristiques de tous genres dont la seule justification vient du fait qu'elles fournissent le "résultat escompté" dans un grand nombre de cas. Malheureusement, la notion de "résultat escompté" est ambiguë parce que le problème, tel qu'il est posé, admet plusieurs solutions. Le résultat escompté est alors l'accord avec l'interprétation humaine, qui peut changer d'une personne à l'autre.

2.1.3 Le problème de l'étiquetage

2.1.3.1 Les étiquettes de Huffman et Clowes

L'aspect de l'interprétation d'images dans le monde des blocs qui a reçu le plus d'attention a été introduit par Guzmán: interpréter les lignes et les noeuds d'un graphe à deux dimensions, fournis par la segmentation, en tant qu'arêtes et sommets d'objets triédriques. Un aspect important de ce problème est l'étiquetage des lignes introduit par Huffman [Huffman 71] et Clowes [Clowes 71]. Les étiquettes

classifient les lignes d'une image suivant les propriétés géométriques des arêtes de la scène:

- > Ligne sur le contour occultant d'un objet. L'objet se trouve à droite de la flèche.
- + Ligne intérieure convexe.
- Ligne intérieure concave.

Un exemple d'étiquetage est donné avec la figure I-11.

Les noeuds de l'image sont classés en quatre catégories:

L -noeud à deux lignes

Y -noeud à trois lignes ne formant aucun angle supérieur à π

Flèche -noeud à trois lignes formant un angle supérieur à π

T -noeud à 3 lignes formant un angle égal à π . Un noeud T correspond le plus souvent non pas à un sommet physique, mais à l'occultation d'un objet par un autre.

Chaque ligne pouvant être étiquetée de quatre façons différentes, un noeud à trois lignes admet $3^4=81$ configurations différentes. Une simple étude géométrique montre que seules quelques unes de ces configurations sont physiquement possibles (figure I-12). Ainsi, une flèche et un Y peuvent donner lieu à 3 configurations, un T à 4, un L à 6. Un étiquetage cohérent d'un graphe doit alors satisfaire à deux contraintes d'intégrité:

- * Chaque noeud doit porter une configuration valide en fonction de son type. C'est une contrainte locale.
- * Chaque ligne porte la même étiquette aux deux extrémités. C'est une contrainte à caractère plus global.

Malgré le nombre limité de configurations possibles pour chaque noeud, l'étiquetage d'un graphe de quelques dizaines de noeuds est déjà un problème combinatoire important, et la recherche dans le graphe d'états est une opération coûteuse. Ce type d'étiquetage pose aussi quelques problèmes de fond. Certains étiquetages cohérents selon les critères ci-dessus peuvent être géométriquement impossibles; et des scènes impossibles peuvent être étiquetées [Huffman 71].

2.1.3.2 La généralisation de Waltz

[Waltz 72] a enrichi cette méthode d'étiquetage, et en même temps trouvé un algorithme efficace et élégant. Il introduit des étiquettes supplémentaires pour les lignes:

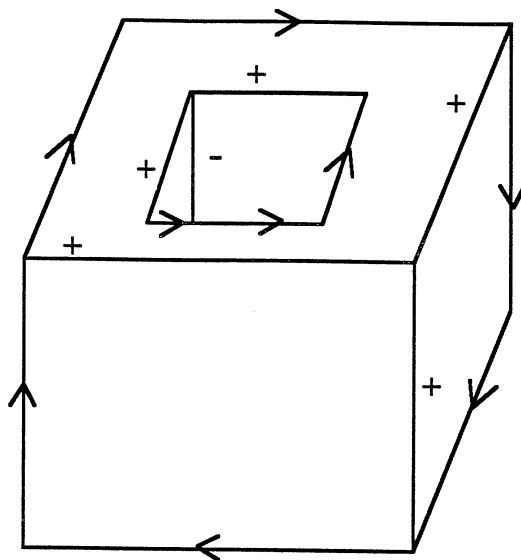


Figure I-11. Une scène étiquetée selon Huffman et Clowes (d'après [Huffman 71]).

Visible surfaces Octants filed	3	2	1	0
1				-
3				-
5			-	-
7		-	-	-
Occlusion				

Figure I-12. Les types de noeuds considérés par Huffman et Clowes ([Ballard 82]).












	Approximate number of combinatorially possible labelings	Approximate number of physically possible labelings
	3,249	92
	185,000	86
	185,000	826
	185,000	623
	11×10^6	10
	11×10^6	435
	11×10^6	213
	11×10^6	128
	11×10^6	160
	600×10^6	20

Figure I-13. Les types de noeuds considérés par Waltz ([Winston 75]).

- Ligne d'ombre. La flèche pointe vers l'intérieur de l'ombre.
-  Deux objets alignés, l'objet occultant étant à droite.
- ⇒ Arête concave séparable (deux objets). Une arête est séparable si elle correspond à deux (ou trois) arêtes physiques appartenant à deux (ou trois) objets juxtaposés. Une séparation des objets entraîne l'apparition de deux arêtes.
- ⇒⇒ Arête concave séparable (trois objets).

et de nouveaux types de noeuds (figure I-13). La figure I-14 montre une scène simple ainsi étiquetée. Chaque région de l'image porte en plus un attribut d'éclairage indiquant si elle est directement éclairée (visible depuis la source de lumière), invisible de la source de lumière et dans sa propre ombre, ou dans l'ombre portée d'un autre objet.

Ces extensions augmentent considérablement la combinatoire du problème. Waltz fit alors deux découvertes remarquables:

* Le nombre de configurations d'étiquettes physiquement possibles pour un noeud est très réduit (figure I-13). Le cas

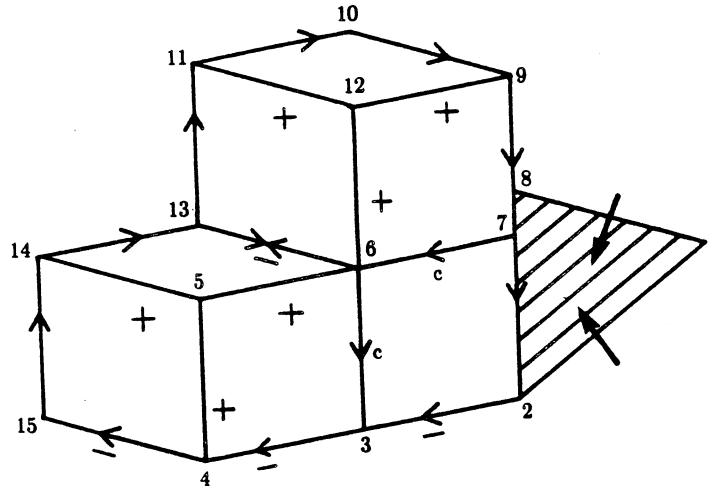


Figure I-14. Une image avec les étiquettes de Waltz.

extrême est celui du "multi-K"



pour lequel il y a environ $6 \cdot 10^6$ configurations au sens combinatoire, dont 20 seulement sont physiquement possibles.

* Un algorithme de propagation de contraintes permet de trouver rapidement le(s) étiquetage(s) cohérent(s) d'un graphe, alors que l'espace de recherche du problème est considérable.

Le principe de cet algorithme est le suivant:

- 1) Initialisation: A chaque noeud attacher toutes les configurations possibles.
- 2) Itération: Pour chaque noeud, éliminer les configurations incompatibles avec un noeud voisin. (Il y a incompatibilité si une ligne reçoit deux étiquettes différentes à ses extrémités).

L'étape d'itération est répétée jusqu'à ce que l'étiquetage ne soit plus modifié.

En principe un tel algorithme est efficace si les conditions de cohérence imposent des contraintes fortes, ce qui s'avère être le cas dans l'étiquetage des scènes considérées par Waltz.

2.1.3.3 Evaluation

Waltz a posé un problème ayant un graphe de recherche très vaste. Il a montré que la physique de notre univers et en particulier du processus de formation d'images imposent des contraintes qui réduisent cette combinatoire et qui permettent de trouver une solution de façon élégante et efficace. La propagation de contraintes est devenue une technique de programmation importante qui a trouvé d'autres applications en IA.

Bien que ce travail n'ait pu être généralisé directement à d'autres domaines (comme la plupart des travaux dans le monde des blocs), il a stimulé la recherche de contraintes analogues pour des scènes plus générales.

Un mérite de Waltz souvent oublié est d'avoir analysé la notion de configuration physiquement possible et d'avoir écrit un programme qui les construit dans les cas considérés.

2.2 APPROCHES FONDAMENTALES - STRUCTURATION

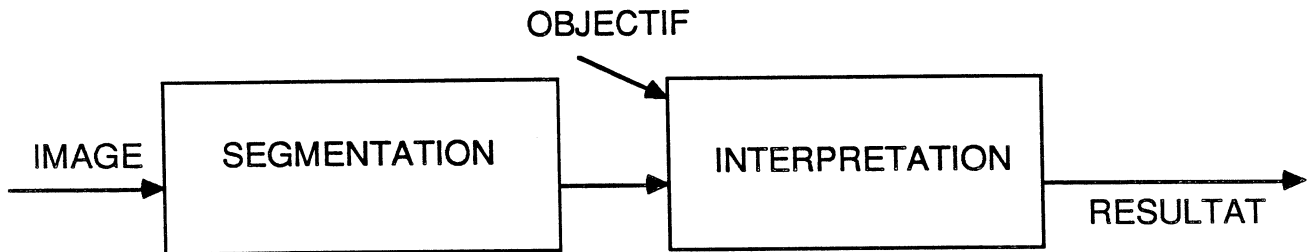
2.2.1 Le paradigme "segmentation - interprétation"

Les systèmes décrits aux paragraphes précédents illustrent un aspect important concernant la conception globale d'un système de vision, prévalant au début des recherches en VO. Il était généralement admis qu'un système de vision était composé de deux parties:

- la **segmentation**, ou vision de bas niveau, qui détecte des "indices visuels" dans l'image.
- l'**interprétation** qui établit la correspondance entre les indices visuels et des modèles d'objets.

Les indices visuels utilisés étaient essentiellement des lignes (lignes de contraste) ou des régions (régions homogènes dans l'image, cf. §2.3.2 ci-après). Il était sous-entendu que la segmentation était bonne lorsque chaque indice visuel correspondait directement à une partie modélisée d'un objet. Par exemple, une ligne correspond à un contour ou une arête interne d'un objet; une région correspond à une face d'un objet. Le schéma utilisé au début était donc purement

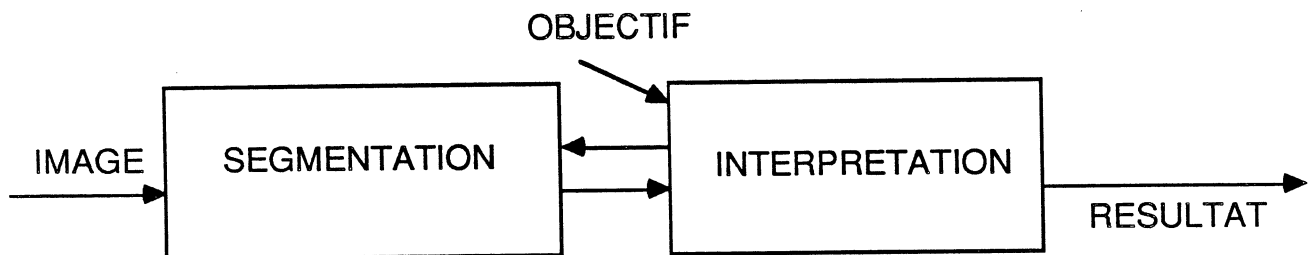
séquentiel:



Cette approche se heurta à des difficultés :

- * une bonne segmentation apparut impossible, sauf dans des domaines d'application restreints (vision binaire industrielle par exemple).
- * l'interprétation était fortement combinatoire.

Une première idée émise pour surmonter ces difficultés fut celle des "systèmes hétérarchiques" (opposé à la hiérarchie du schéma précédant) dans lesquels la segmentation reçoit des informations de l'interprétation, ce qui correspond à l'ajout d'une liaison dans le schéma précédent:



Par exemple, le programme de [Shirai 73], travaillant dans le monde des blocs, extrait dans un premier temps seulement les contours extérieurs de la scène, pour lesquels le contraste est élevé. Ensuite, il utilise un certain nombre de règles heuristiques pour lancer la segmentation à la recherche d'autres lignes: à l'intérieur d'un L en vue de constituer une flèche, en prolongement d'un sommet pendant, ou aux alentours d'un L intérieur en vue de trouver un Y. La figure I-15 illustre les étapes de progression du programme. La recherche des lignes est ainsi parfaitement imbriquée avec l'analyse de la scène.

On espérait qu'avec une telle structure les faiblesses inhérentes à la segmentation seraient compensées par "l'intelligence" globale du système, qui permettrait, à partir de connaissances du monde, d'imposer de nombreuses contraintes à la segmentation. Le courant principal de ces recherches concernait donc l'aspect prédictif à base de connaissances générales.

Dès 1975 cette "impossibilité de segmentation" a été contestée, notamment par D.Marr (cf. paragraphe suivant). L'argument peut être résumé ainsi ([Barrow 78], [Binford 82]): le système visuel humain fournit une perception très vive, même pour des images ne contenant pas d'objets connus, comme la figure I-16, où nous voyons immédiatement des surfaces dans l'espace à trois dimensions; les programmes devraient en être capables aussi. L'apparence d'impossibilité de segmentation vient d'un mauvais énoncé et/ou d'une mauvaise structuration du problème.

Par la suite, le paradigme segmentation - interprétation a progressivement laissé place à des structures de systèmes plus complexes, dans lesquelles on trouve, au lieu d'un unique module de segmentation, la construction d'une succession de structures de données intermédiaires.

2.2.2 La théorie de Marr

La proposition la plus complète pour une théorie de la vision a été élaborée par David Marr et son équipe au MIT. Elle est exposée dans [Marr 82]; on trouve un approfondissement de certaines parties dans [Ullman 78], [Grimson 81] et [Hildreth 83]. La théorie de Marr postule trois structures de données amenant à une description d'une image en termes de volumes dans l'espace 3-D:

- * Le **croquis élémentaire** (primal sketch) rend explicites les changements d'intensité dans l'image (lignes de contraste avec leurs extrémités), leur distribution et organisation géométrique (régularités).
- * Le **croquis 2 1/2-D** rend explicites l'orientation et la profondeur approximative des surfaces visibles, et les contours de discontinuité de ces grandeurs. Le repère utilisé par cette structure est lié à l'observateur.
- * La **description 3-D** décrit les formes et leur organisation spatiale dans un repère propre à chaque objet. Elle utilise une représentation hiérarchique et modulaire à base de primitives volumétriques et de primitives de surface (cylindres généralisés, cf. figure I-23).

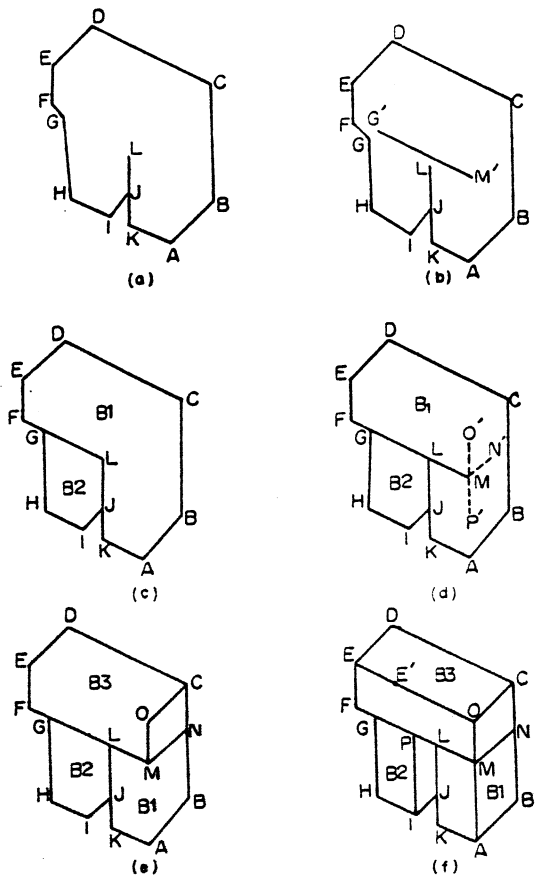


Figure I-15. Etapes d'analyse d'un programme "hétéarchique" d'analyse de scènes ([Shirai 73]).



Figure I-16. Dans cette image nous percevons naturellement des structures à 3 dimensions sans posséder des informations de haut niveau pour guider la perception.

L'utilité, voire la nécessité de ces structures intermédiaires dérive d'une analyse rigoureuse du problème de la vision, présentée dans le livre de Marr [Marr 82]. La démarche intellectuelle étant descendante - Marr part de la question "qu'est-ce que la vision?" pour déduire les éléments d'un système de vision - cette théorie n'est pas entièrement développée jusqu'au niveau de l'implantation. Les parties étudiées en détail concernent le croquis élémentaire, la stéréoscopie, le mouvement, et quelques autres modules contribuant à la construction du croquis 2 1/2-D.

Il est important de noter que l'approche de Marr est une approche fondamentale dont l'objectif est un modèle de la vision humaine; il ne s'est intéressé ni aux possibilités d'une implantation efficace sur un matériel disponible aujourd'hui, en vue d'applications, ni à l'interaction de sa théorie avec l'Intelligence Artificielle.

Ce qui caractérise cette approche vis à vis de la plupart des recherches menées en VO est

- * Le caractère ascendant: une description 3-D de la scène est construite sans utilisation de connaissances sémantiques spécifiques de la scène.
- * La reconnaissance d'objets par mise en correspondance avec les modèles est effectuée sur des structures 3-D. (Cette partie n'a pas été implantée, et sa faisabilité reste contestée).

2.2.3 L'approche système

D'autres propositions pour la structure globale d'un système de VO sont plus ou moins implicites dans plusieurs systèmes présentés dans les paragraphes suivants, notamment: ACRONYM, VISIONS, et le système de l'Université de Rochester. Par rapport à la théorie de Marr, ces propositions diffèrent en plusieurs points:

- * Un système existant montre la faisabilité globale de l'approche. Cependant, les réalisations ne sont pas toujours décrites avec beaucoup de précision, et leur compétence effective n'est pas clairement énoncée.
- * Les composants de ces systèmes ne sont pas justifiés par des considérations théoriques, mais proposés sans discussion. Cette structure est en fait motivée par la flexibilité informatique et reflète l'état de l'art en IA.

- * La correspondance scène - modèle que Marr propose d'effectuer en termes de structures 3-D, est effectuée au niveau 2-D.

Par exemple, dans le système ACRONYM (cf.§2.3.4) les structures de données principales sont constituées par cinq graphes: le graphe d'objets et le graphe des restrictions contiennent les modèles 3-D; le graphe des prédictions fournit des modèles 2-D; le graphe image est une description de l'image; le graphe d'interprétation fournit le résultat du système.

Les modules de traitement sont au nombre de quatre:

- * Le module de modélisation décrit les scènes à reconnaître en termes d'objets dans l'espace 3-D et de relations entre ces objets. Les primitives volumiques utilisées sont les cônes généralisés, identiques aux cylindres généralisés proposés par Marr.
- * Le module d'analyse statique déduit, à partir des modèles, des indices visuels 2-D permettant la reconnaissance d'un objet (graphe des prédictions). Chaque indice visuel est plus ou moins spécifique à un point de vue, mais l'ensemble des indices visuels couvre tous les points de vue.
- * Le module de description extrait d'une image les indices visuels présents (en fonction des IVs du graphe des prédictions).
- * Le module d'interprétation d'image construit un modèle de la scène en établissant une correspondance entre les IVs-image et les résultats de l'analyse statique des modèles.

Cette structuration est intéressante d'un point de vue général, car elle permet de classifier et de comparer de nombreux systèmes de vision (cf.[Binford82]). Cependant elle n'est pas directement comparable à la théorie de Marr qui propose une approche de la description et en partie de la modélisation, sans se préoccuper des questions de réalisation, alors qu'ici on s'intéresse plutôt à la réalisation, sans une théorie complète de la vision.

2.3 DESCRIPTION DE QUELQUES SYSTEMES

Dans les paragraphes suivants nous présentons un certain nombre de systèmes de vision existants que nous avons repartis en trois groupes en fonction des motivations de leurs auteurs.

Dans le premier groupe sont présentés des systèmes qui cherchent à construire une vision générale, à l'intérieur d'un système d'IA, prenant la VO au sens large que nous lui avons donné au premier chapitre. L'intérêt de ces systèmes vient du fait qu'ils mettent en oeuvre des concepts généraux, dont nous avons décrit les grandes lignes dans le paragraphe précédent. Cependant, ces systèmes ne sont pas (pas encore) capables de résoudre des problèmes réalistes en termes d'applications.

Or, de nombreuses applications de la VO doivent également être étudiées, sans attendre l'avènement des systèmes de VO généraux. Nous avons classé dans un deuxième groupe des systèmes de recherches ayant pour objectif de résoudre effectivement un problème de vision difficile et économiquement intéressant. Nous trouvons en particulier l'analyse d'images aériennes et radiographiques. Par rapport aux systèmes du premier groupe, ces systèmes excellent par une expertise spécifique au détriment de la généralité. Lorsqu'il s'agit de recherches en cours, on peut s'attendre à une amélioration des résultats dans un avenir proche.

Dans un troisième groupe nous avons classé les "systèmes industriels". Les systèmes de vision industriels doivent répondre à un cahier des charges très précis, et exécuter une tâche spécifiée avec une performance (vitesse, coût du matériel) imposée et prioritaire. Ces systèmes souffrent alors d'un manque de généralité: Les possibilités de description d'objets sont très réduites, il y a limitation à des scènes 2-D, le point de vue est imposé, ils nécessitent un éclairage spécial. Dans le cas extrême, le système peut ne présenter aucun intérêt pour l'étude scientifique de la vision.

La description de chaque système se décompose en trois parties:

- a) Motivation: nous précisons les aspects du problème de la vision qui ont été particulièrement visés.
- b) Description proprement dite.
- c) Evaluation: nous mentionnons brièvement les aspects importants du travail, et les lacunes les plus nettes.

2.3.1 Le travail de Shirai

Motivation

[Shirai 78] a cherché à écrire un système de vision ayant une compétence dans un environnement moins artificiel que le monde des blocs. Il a choisi des images d'une salle de bureau (office scene). Il analyse des scènes 3-D avec des modèles d'objets. L'effort a été porté sur la compétence: arriver à la reconnaissance de quelques types d'objets dans des images assez générales.

Description

Le système de Shirai reconnaît un certain nombre d'objets dans une image à plusieurs niveaux de gris. Les exemples traités (figure I-17-a) concernent des scènes de bureau; les objets modélisés sont une lampe, un rayon à livres, une tasse, un téléphone, un crayon (figure I-18).

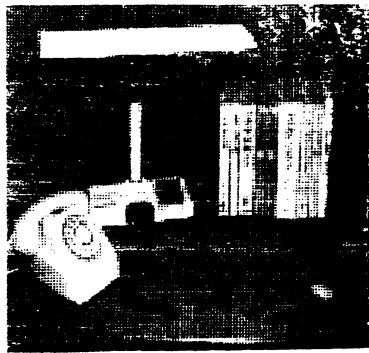
Lors de l'interprétation, le système parcourt une ou plusieurs fois un cycle segmentation - interprétation jusqu'à ce qu'un nombre suffisant d'objets soit trouvé, ou que la segmentation ait été exhaustive.

La segmentation recherche toutes les lignes de contraste dans l'image, en fonction d'un seuil de gradient (qui varie à chaque passage). Les lignes sont ensuite découpées en segments de droites et arcs d'ellipses. L'ensemble des segments constitue la description de l'image.

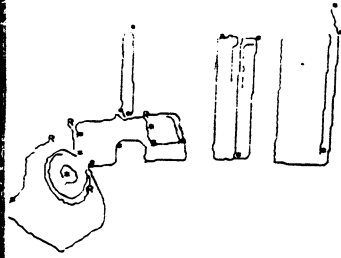
Les modèles d'objet ont une structure hiérarchique, dans laquelle figurent les caractéristiques principales utilisées pour la construction initiale d'une hypothèse, et les caractéristiques secondaires utiles à la vérification. Les caractéristiques peuvent être des lignes fermées ou non. Les segments de l'image sont comparés aux caractéristiques principales pour construire des hypothèses; il s'en suit une phase de prédiction-vérification. La figure I-17 montre les étapes successives du traitement d'un exemple.

Evaluation

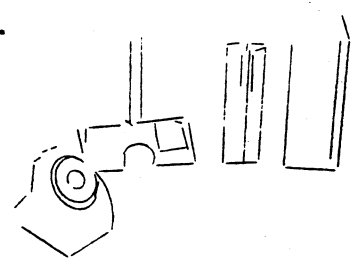
Ce système, qui a marqué à son époque un progrès important relativement au passage du monde des blocs à un environnement plus général, possède aussi de nombreuses lacunes: la modélisation est trop ad-hoc, et de façon générale, aucun des aspects fondamentaux évoqués au §2.2 n'est approfondi.



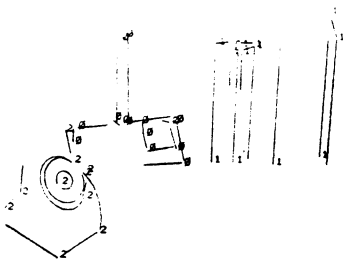
(a) Original scene



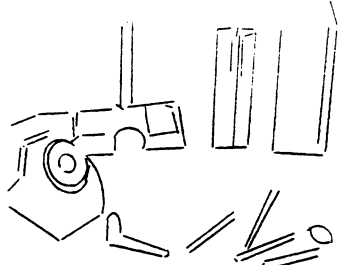
(b) Edges found in cycle 1



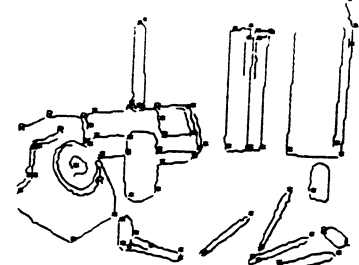
(c) Description



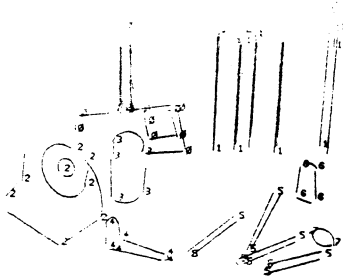
(d) Recognition



(e) Description in cycle 2



(f) Edges found in cycle 4



(g) Recognition

- 0: lamp
- 1: book stand
- 2: telephone
- 3: cup
- 4: pipe
- 5: pen, pencil, ball-pointpen or felt pen
- 6: ink pot
- 7: eraser

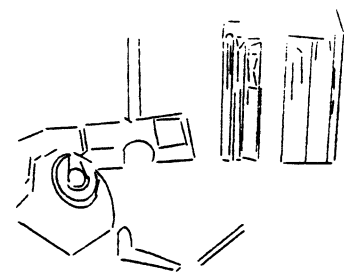


Figure 10. Description of edges found without feedback from the recognition process.

Figure 9. An example of the recognition process.

Figure I-17. Fonctionnement du programme de [Shirai 78].

2.3.2 Analyse de régions: MSYS et ARGOS

Motivation

Les systèmes présentés jusqu'à présent utilisent comme élément de base de description de l'image les lignes de contraste. Une autre approche est l'utilisation d'une partition de l'image en régions homogènes, chaque région étant définie comme un ensemble connexe de pixels satisfaisant un certain prédicat. Dans le cas d'une image à plusieurs niveaux de gris, il peut s'agir de pixels ayant le même niveau de gris, ou appartenant à une plage de valeurs.

Ces deux approches sont duales en ce sens qu'une région définit une ligne par son contour, et qu'une ligne fermée définit une région. Elles amènent cependant à des algorithmes sensiblement différents et ne fournissent pas en général les mêmes résultats.

Description

Le système MSYS [Tenenbaum 77] suit le paradigme segmentation - interprétation avec une analyse de régions. Il a été expérimenté avec des images en couleur d'un paysage, d'un compresseur d'air, et d'une scène de bureau.

La segmentation produit la partition initiale de l'image en régions. L'interprétation doit:

- * Combiner des régions voisines appartenant à un même objet (fusion de régions).
- * Etiqueter chaque région avec un nom d'objet.

La difficulté du problème vient du fait que la segmentation initiale produit un très grand nombre de régions, et que la seule donnée de la couleur ne permet pas d'identifier un objet. MSYS exploite alors des contraintes contextuelles dans une image en utilisant un algorithme de propagation de contraintes.

Un exemple du fonctionnement de cet algorithme est donné par la figure I-19.

Il s'agit d'une scène de bureau où les objets peuvent être la porte (D), le mur (W), le sol (F), une peinture (P), une plinthe (B), ou une poignée de porte (K). Les contraintes prises en compte sont

- à l'intérieur de
- au dessus de
- à coté de

Object	Main feature	Secondary feature
Lamp	Contour of bright strip region corresponding to lamp shade.	Pair of vertical edges of opposite directions corresponding to trunk, and contour of base under trunk.
Book Stand	Long vertical lines clustered in a rectangular region.	Lines connected to main features.
Cup	Pair of vertical edges of opposite directions corresponding to side of cup.	Contour of cup connected to main features.
Telephone	Ellipse corresponding to contour of dial or outer circle surrounding number characters.	Contour of case surrounding main feature.
Small objects (pipe, pen, etc.)	Shape and size of contour.	Details of shape and light intensity changes.

Figure I-18. Modèles d'objets dans [Shirai 78]. Noter le choix ad-hoc des indices visuels pour chaque objet (pas de modélisation générale).

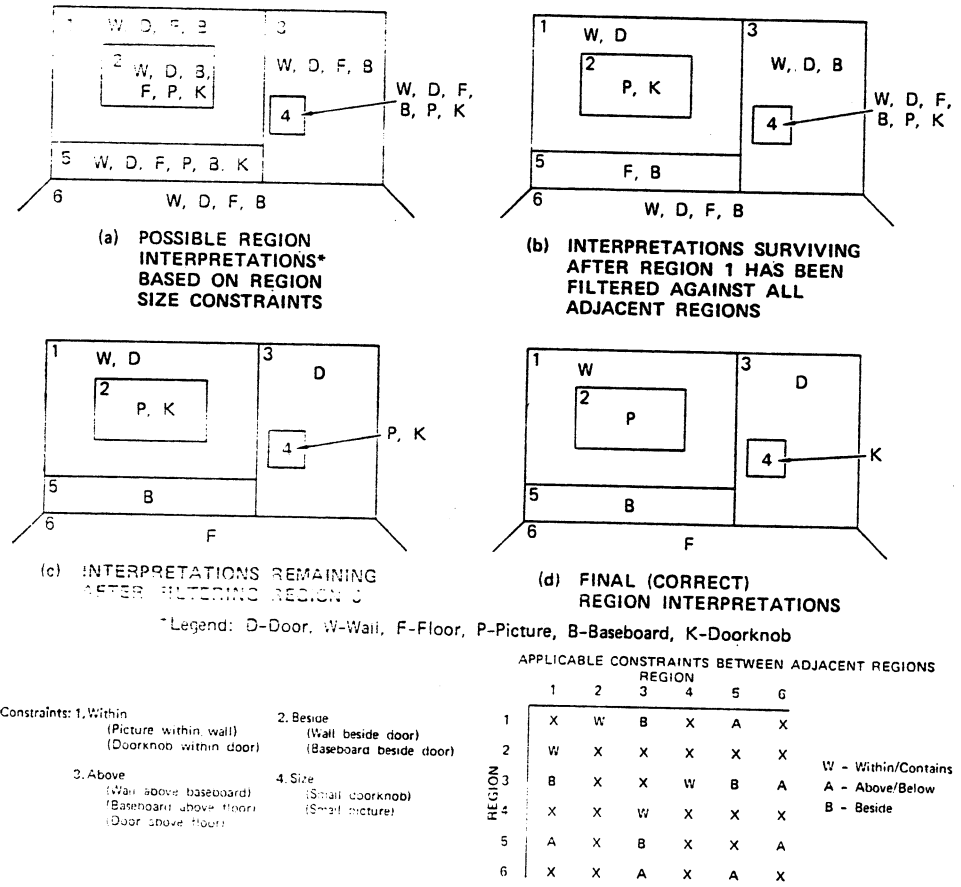


Figure I-19. Illustration de MSYS: propagation de contraintes dans l'analyse de régions [Tenenbaum 77].

et sont données en haut à gauche de la figure I-19. Initialement, toutes les régions reçoivent toutes les étiquettes. Dans le cas de l'exemple, les contraintes ne laissent qu'une seule étiquette par région au bout de 4 itérations.

Une méthode très différente d'étiquetage de régions a été proposée par Rubin avec le système ARGOS [Rubin 80]. Il le traite comme un problème de recherche combinatoire et utilise un algorithme en deux passages sans retour arrière appelé "locus search". Le premier passage est ascendant et construit un graphe contenant les hypothèses les plus plausibles. Le deuxième passage est descendant et choisit une seule solution. ARGOS a été expérimenté avec des vues de la ville de Pittsburgh; il est capable de déterminer approximativement la direction de la prise de vue (à 40 degrés près).

Evaluation

Ce type de système a une structure assez simple qui permet une implantation efficace. L'étiquetage initial des régions sur la base de propriétés des pixels pose cependant des problèmes en cas de présence de régions fortement texturées, et/ou de présence de bruit (apparition de nombreuses petites régions de caractéristiques différentes).

On remarque que les relations servant à la propagation de contraintes sont des relations dans l'image 2-D, non pas des relations dans la scène.

D'autre part, ces systèmes ne tiennent pas compte de caractéristiques globales des régions (par ex. surface, forme) qui permettent souvent une identification.

2.3.3 VISIONS

Motivation

VISIONS [Hanson 78a, Hanson 78b] est un système de vision mettant en oeuvre le principe du "tableau noir" introduit avec le système HEARSAY II [Erman 75] pour la compréhension de la parole afin de proposer une structure claire permettant d'intégrer les nombreux niveaux de représentation de la parole. L'importance de ce principe pour la vision a été suggérée par [Reddy 73]. Le système VISIONS met en oeuvre cette structure dans le domaine de la vision. Il se présente comme un ensemble de processeurs appelés "sources de connaissances", dont chacun est spécialiste d'un certain type d'analyse d'image, les échanges des résultats passant par le tableau noir. L'accent global est mis sur la structure du système - les problèmes de contrôle, de communication.

Description

La structure globale du système VISIONS est donnée par la figure I-20. Son élément central est le "tableau noir" - marqué REPRESENTATION dans la figure I-20 - qui centralise les informations concernant l'interprétation d'une image provenant de différentes "sources de connaissance" (KS - knowledge source). Une KS est un programme spécialisé dans un certain type d'analyse.

Le tableau noir de VISIONS est une description à plusieurs niveaux d'une image dont un exemple est donné avec la figure I-21. Les niveaux inférieurs représentent des régions, des segments de lignes, et des sommets, et forment ainsi un graphe, appelé graphe RSV (regions, segments, vertices). Ce graphe est construit par le sous-système de segmentation (moitié gauche de la figure I-20). Les deux niveaux immédiatement supérieurs du tableau noir décrivent des surfaces et des volumes. Ces niveaux servent donc à reconstituer la configuration 3-D de la scène. Les deux niveaux au sommet de la hiérarchie du tableau noir représentent des objets et des schémas. Ces niveaux établissent le lien entre la scène d'une image et les modèles connus du système, stockés dans une "mémoire à long terme". Un schéma définit un type de scène avec un ensemble d'objets-type et des contraintes (équivalent du "frame" de [Minsky 75]).

Etant donnée la faiblesse des KSs effectivement implantées pour la construction des surfaces et des volumes, les niveaux supérieurs travaillent de façon prédictive, la correspondance avec l'image étant effectuée au niveau du graphe RSV.

La structure de contrôle de VISIONS suit le principe de prédiction-vérification.

- Choix d'un élément non développé d'une hypothèse, c'est à dire choix d'un noeud à partir duquel on calculera une nouvelle prédiction.
- Expansion de cet élément: génération de nouvelles hypothèses par appel d'une KS.
- Vérification des nouvelles hypothèses.

Le choix de l'élément à développer se fait en plusieurs étapes: choix de l'hypothèse au niveau schéma, choix du niveau de représentation à élaborer, choix du noeud. Pour chaque hypothèse le système gère une évaluation de vraisemblance, qui est en particulier fonction de la vraisemblance des hypothèses concurrentes, et du rôle d'une hypothèse pour les hypothèses du niveau supérieur.

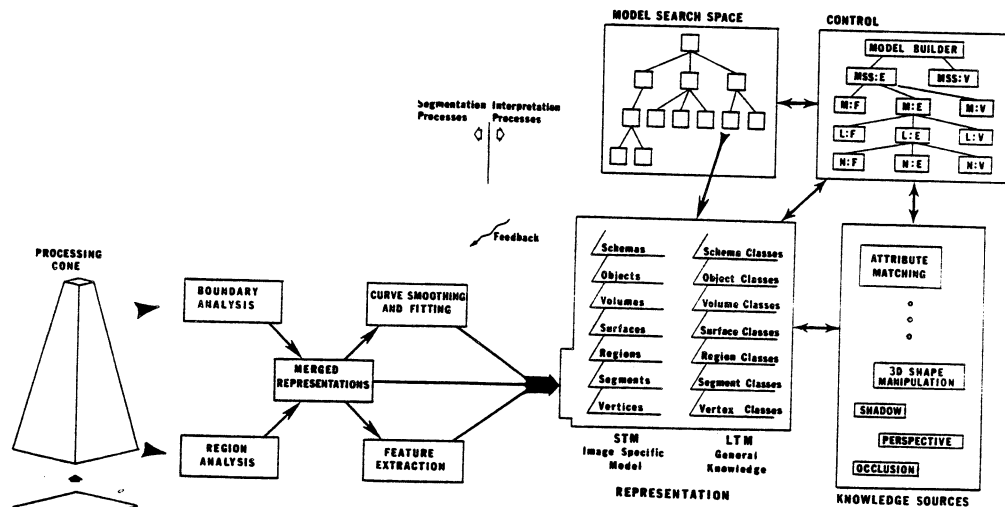
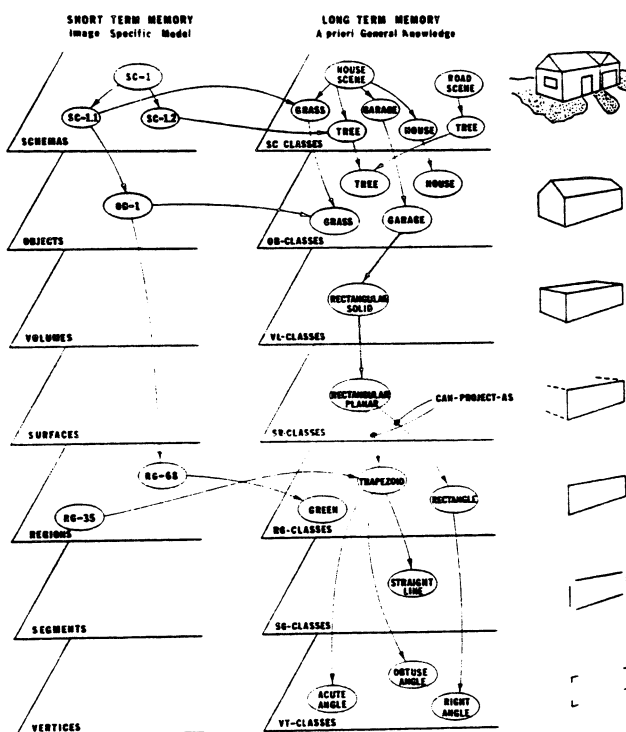


Figure I-20. Structure globale du système VISIONS [Hanson 78a].

La partie gauche du dessin représente l'extraction d'indices visuels, la partie droite concerne la construction d'une description symbolique de la scène. Cette partie interprétation comprend les modèles de représentation des connaissances, des "sources de connaissances" (KSs) pour la formation d'hypothèses, et un arbre de recherche dans l'espace des modèles.

Figure I-21. Structures de données dans le système VISIONS.



Cette figure montre les niveaux de représentation utilisés dans l'interprétation. Les niveaux, qui définissent les niveaux d'abstraction nécessaires dans un système général de perception visuelle, traversent deux structures différentes: 1) une mémoire à long terme (LTM, au milieu) avec des connaissances générales, et 2) une mémoire à court terme (STM, à gauche) pour l'interprétation d'une image (à droite).

Evaluation

Avec ACRONYM, VISIONS constitue l'un des projets de vision les plus ambitieux. Il est seul à utiliser un formalisme de représentation de connaissances sophistiqué au sens de l'IA - les schémas - pour la description symbolique de la scène. Toutefois, cette idée en est à son premier essai; on voit mal comment on peut représenter toutes les scènes possibles par des schémas statiques et des descripteurs géométriques simples. En l'absence d'une application concrète il paraît cependant difficile de prononcer un jugement précis.

2.3.4 ACRONYM

Motivation

L'ambition du système ACRONYM [Brooks 81a] est d'être un système de reconnaissance d'objets indépendant du domaine d'application. Il est guidé par la connaissance de modèles génériques d'objets, et effectue des raisonnements géométriques pour les identifier dans une image. L'accent dans l'interprétation est mis sur des calculs géométriques à base d'inégalités. Notre description d'ACRONYM est inspirée de l'article de Brooks dans [Cohen et Feigenbaum 82].

Description

ACRONYM essaie d'identifier et de classifier des instances de modèles. Il extrait de l'image des informations de forme et de structure 3-D, et détermine l'emplacement et l'orientation dans l'espace des objets. Le système a été testé avec des vues aériennes d'un aéroport, et avec des images d'un poste de travail automatisé présentant des pièces industrielles.

La structure globale du système est présentée dans la figure I-22. Nous avons introduit au §2.2.3 les quatre parties principales qui sont la modélisation, la prédiction, la description, et l'interprétation.

Remarque: le terme "prédiction" a une signification particulière dans le contexte d'ACRONYM que nous avons appelé "analyse statique des modèles" ailleurs.

Dans toutes ses phases, en dehors de la description, deux modules de raisonnement symbolique sont utilisés:

- * Un module géométrique capable de raisonner sur des produits complexes de transformations - typiquement entre 10 et 20 transformations avec plus de 10 variables

libres.

- * Un module d'algèbre capable d'analyser des ensembles d'inégalités symboliques algébriques non-linéaires. Ce module produit des bornes pour les valeurs satisfaisant ces inégalités. Il a été développé à partir d'une méthode de raisonnement pour inégalités linéaires dans les entiers, élaborée par Bledsoe dans le cadre des preuves de programmes [Bledsoe 75], puis étendue par [Shostak 79] aux nombres réels.

Modélisation

Les modèles donnés par l'utilisateur sont de deux sortes: les modèles d'objets et de leurs relations spatiales, et les modèles de classes d'objets et des relations de sous-classes. Les premiers définissent l'aspect géométrique du problème, représenté dans le "graphe d'objets"; les seconds définissent l'aspect algébrique, représenté dans le "graphe de restrictions".

Les objets sont modélisés par les volumes occupés et par des transformations entre systèmes de coordonnées locaux de ces volumes. Les classes (et les sous-classes) sont déterminées par des ensembles d'inégalités (contraintes) dans les paramètres du modèle géométrique. Les noeuds du graphe d'objets dénotent des objets et des sous-objets; ils se réfèrent à des volumes primitifs représentés par des cônes généralisés. La figure I-23 montre des exemples de cônes généralisés, dans lesquels le bloc d'un moteur électrique est représenté par un simple cylindre droit.

Dans le graphe d'objets on trouve deux types d'arcs orientés.

- * Les **arcs composants** définissent la hiérarchie des composants d'un objet complexe. Le moteur électrique pourrait par exemple avoir comme composants une plaque de support, des flasques, un arbre de transmission.
- * Les **arcs de liaison** (affixment) décrivent les relations spatiales entre composants. Par exemple, l'axe de l'arbre de transmission doit être colinéaire avec l'axe du bloc moteur.

Les paramètres nécessaires à la description d'un cône généralisé, d'un arc composant, ou d'un arc de liaison peuvent être donnés par des valeurs numériques précises, spécifiant un objet particulier, ou être laissés comme variables libres, fournissant des contraintes sur les valeurs admises (inégalités algébriques quelconques). Si par exemple on veut décrire une classe de petits moteurs électriques avec des formes et des structures différentes on peut avoir deux

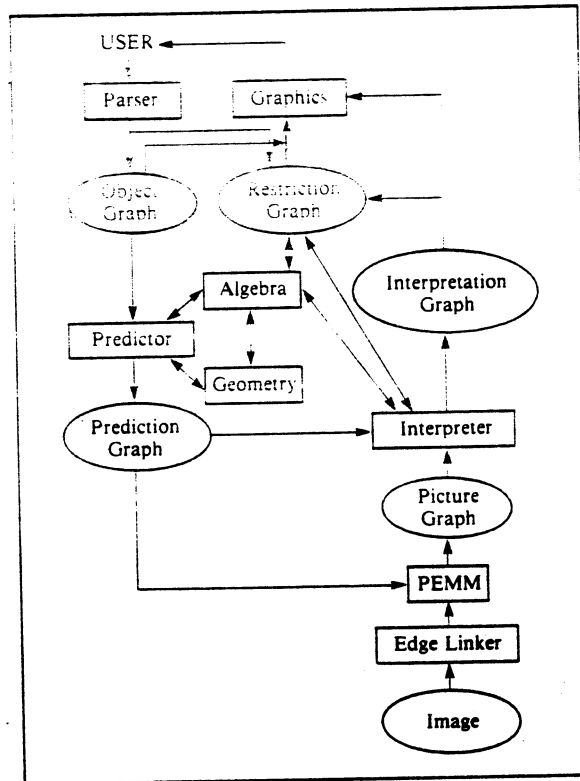


Figure I-22. Structure générale du système ACRONYM (dans [Cohen et Feigenbaum 82]).

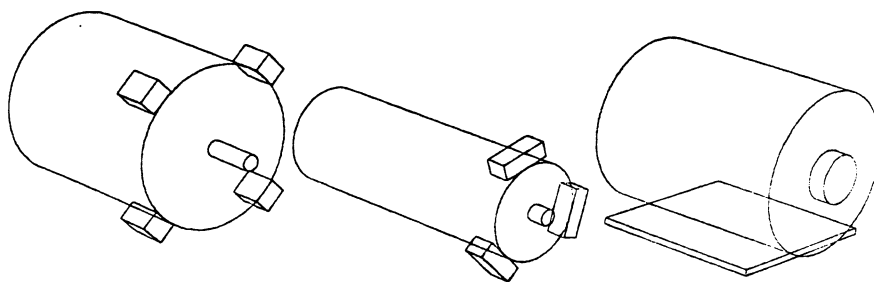


Figure I-23. Trois modèles de moteurs électriques sous forme de cônes généralisés. Il s'agit de trois spécialisations d'un modèle générique. [Brooks 81a]

paramètres LONGUEUR_MOTEUR et RAYON_MOTEUR, et spécifier

$$70.0 \leq \text{LONGUEUR_MOTEUR} * \text{RAYON_MOTEUR}^2 \leq 160.0$$

avec les contraintes supplémentaires

$$6.0 \leq \text{LONGUEUR_MOTEUR} \leq 9.0$$

$$2.0 \leq \text{RAYON_MOTEUR} \leq 3.0$$

De cette façon on peut représenter des variations de grandeur et de forme par des contraintes sur les paramètres des noeuds du graphe des objets.

De façon analogue on peut utiliser des contraintes sur les paramètres des arcs composants pour représenter des variations dans la structure d'un objet, et des contraintes sur les paramètres des arcs de liaison pour représenter des variations dans les relations spatiales. Dans l'exemple du moteur électrique, le nombre de flasques pourrait prendre soit la valeur 0, soit une valeur entre 3 et 6 ce qui imposerait une contrainte de structure pour le moteur, et des paramètres dans les arcs de liaison des flasques au bloc moteur permettraient d'imposer un espacement régulier de chaque coté du moteur.

Les ensembles de contraintes sur les paramètres des modèles forment les noeuds du graphe des restrictions, organisés en treillis par des arcs orientés de "spécialisation". Par exemple, un noeud peut contenir toutes les contraintes propres à une classe générique de moteurs électriques. Ce noeud peut admettre deux spécialisations, dont chacune est un ensemble plus restrictif de contraintes, l'une la sous-classe des moteurs sans plaque de support ni flasques, l'autre la sous-classe des moteurs avec plaque de support et sans flasques. Chacune d'elles peut connaître des spécialisations plus fines. La figure I-23 montre trois instances de tels modèles paramétrés, toutes définies par des valeurs numériques fixes.

Lors de la modélisation, l'utilisateur peut définir explicitement des parties du graphe des restrictions. Pendant l'interprétation, ACRONYM ajoute d'autres noeuds correspondant à des hypothèses d'objets. L'interprétation cherchera ensuite le noeud de restriction le plus spécialisé compatible avec les contraintes dues à l'image - c'est l'identification de sous-classes d'objets.

Le système de modélisation d'ACRONYM est aussi utilisé en dehors de la vision. Il a été appliqué par [Soroka 1980] à la simulation en temps réel pour la programmation de robots, et par [Brooks 81b] à la planification d'assemblages automatiques.

Prédiction

Etant donné un ensemble de modèles et de relations, des techniques de raisonnement géométrique sont utilisées pour prédire des indices visuels invariants, c'est à dire des indices apparaissant dans une image indépendamment de la taille, des variations de structure, des relations spatiales. On prédit également des relations dans l'image entre ces indices. Cela nécessite l'analyse des plages de valeurs admises par les classes de modèles d'objets. Le système ne prédit pas l'apparence complète d'un objet sous tous les angles: il se limite à trouver des indices visuels permettant d'identifier un objet, et de déterminer son orientation et sa position.

Les indices visuels image prédits sont des formes représentées par des bandes (l'équivalent 2-D des cônes généralisés) et des ellipses. On prédit également des relations entre formes dans l'image, comme exclusivité, colinéarité, connexité, angle relatif, distance.

Le résultat essentiel de la prédiction est le graphe des prédictions. Les noeuds de ce graphe sont des prédictions d'indices visuels image, les arcs spécifient des relations attendues entre ces indices. Pendant l'interprétation, les prédictions jouent un rôle double. D'une part, elles constituent un premier filtre pour établir des hypothèses liant des indices visuels image à des indices modèle, d'autre part, elles indiquent comment à partir de la mesure bruitée d'un indice image on peut déduire des informations 3-D concernant l'objet auquel l'indice est lié par l'hypothèse.

Le module de prédiction est implémenté sous forme d'un ensemble d'environ 280 règles de production. Elles sont exécutées avec une structure de contrôle par chaînage arrière, avec la possibilité de définir des sous-buts sur un tableau noir. Le tableau noir est scruté par des méta-règles, qui détectent et résolvent des conflits, et activent la recherche des sous-buts retenus. La sélection des règles se fait par unification de la partie gauche avec un but.

Module de description

Le module de description décrit une image comme un graphe d'image de même format que le graphe des prédictions, c'est à dire en termes de bandes et d'ellipses. Il procède en deux phases, la première ascendante, la deuxième descendante. La première phase, développée à partir du programme de [Nevatia et Babu 81] décrit une image à niveaux de gris par une collection de segments de droites reliés. Il produit typiquement 1000 lignes pour les exemples traités. La figure I-24-a montre les lignes trouvées dans l'image aérienne d'un avion.

Dans la deuxième phase, un programme de correspondance de segments extrait les bandes compatibles avec le graphe des prédictions et produit le graphe image. Le graphe image n'explique aucune information 3-D. La figure I-24-b montre les bandes du graphe image produit à partir de la figure I-24-a, alors que le système cherchait un gros avion sans connaître l'échelle de l'image.

Interprétation

L'interprétation d'une image consiste en la recherche d'un isomorphisme de sous-graphes entre le graphe image et le graphe des prédictions.

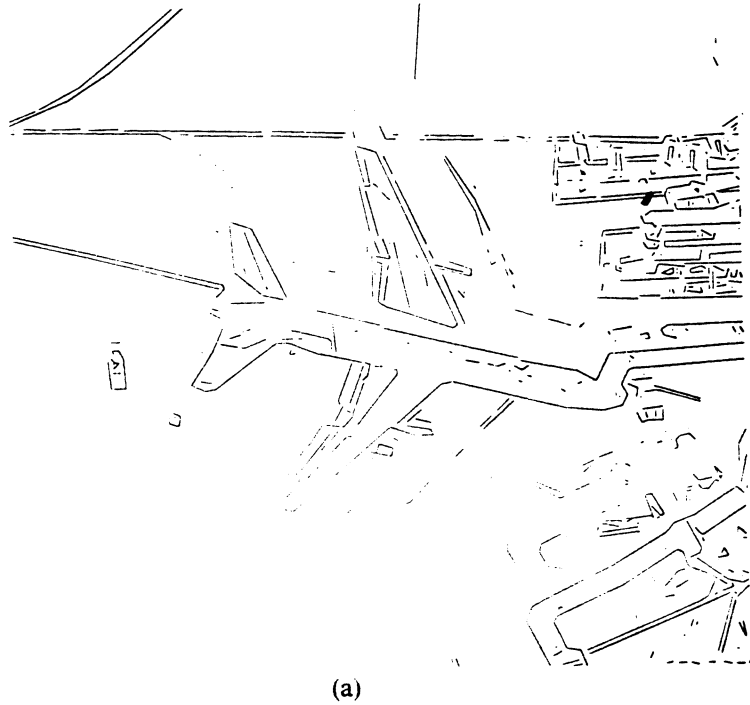
Les indices visuels image doivent non seulement correspondre qualitativement aux prédictions, mais l'ensemble des contraintes numériques engendrées localement doit être globalement cohérent. L'utilisation d'un algorithme de correspondance de sous-graphes permet une bonne interprétation même en cas de visibilité partielle et/ou en cas d'échec des modules de description.

La figure I-25 montre deux spécialisations d'un modèle générique d'un avion civil, utilisées pour l'interprétation de la figure I-24-a (Boeing 747 et Lockheed L-1101). Le modèle géométrique contient environ 30 paramètres liés par des inégalités, exprimant par exemple le fait que l'envergure d'un tel avion est à peu près proportionnelle à la longueur du fuselage. Des inégalités plus spécifiques précisent des sous-classes. Etant donnée une altitude de la caméra entre 1000m et 12000m pour l'image I-24-a, un rapport focal de 20 et une vue légèrement oblique, le système arrive au résultat de la figure I-26. En plus, un grand nombre d'inégalités sur les paramètres sont déterminées. Ceux-ci étant compatibles avec le modèle générique et le modèle du L-1011, mais non pas avec le modèle du B-747, le système conclut qu'il s'agit d'un L-1011. Notons que cette conclusion n'est pas liée à la taille apparente de l'avion dans l'image, mais provient uniquement des relations entre les différentes parties de l'avion, comme le rapport longueur-largeur, et l'angle entre l'aile et le fuselage.

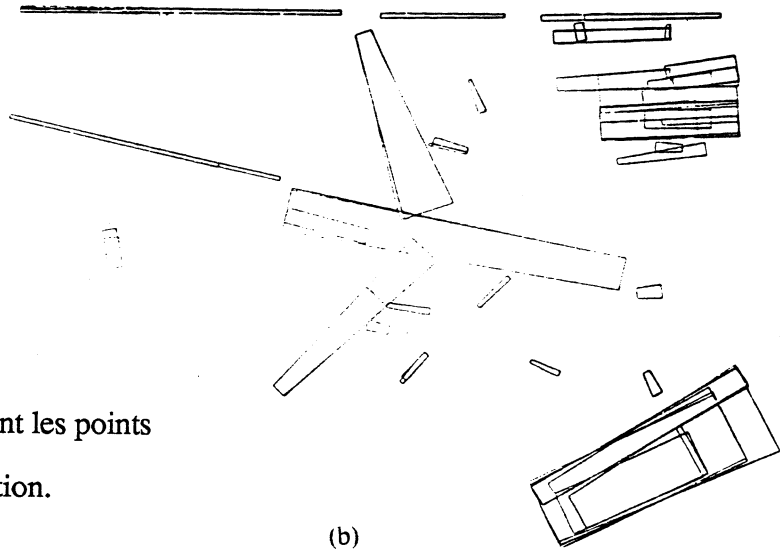
Evaluation

ACRONYM est le premier système de vision qui cherche à combiner des facilités générales de modélisation et des images "naturelles" sans contrainte de position ou d'éclairage. Il présente des résultats significatifs pour ce problème.

Nous lui reprochons de trop fonder l'interprétation sur une représentation par inégalités numériques. D'autre part, l'état exact de la réalisation n'est pas bien précisé. Il



(a)



(b)

a) segments de droites approchant les points de contraste.
 b) Bandes du graphe de description.

Figure I-24. Exemple d'une image traitée par ACRONYM ([Brooks 81a]).

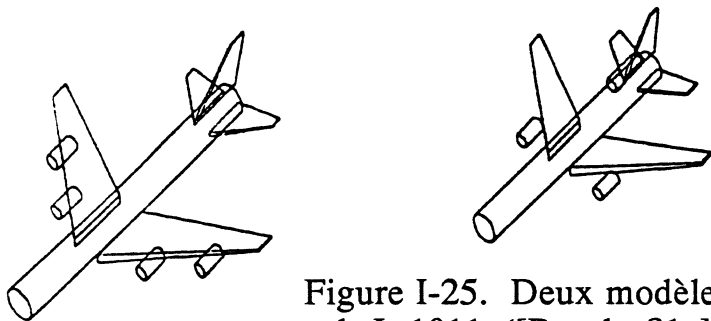


Figure I-25. Deux modèles d'avion: le Boeing 747 et le L-1011 ([Brooks 81a]).

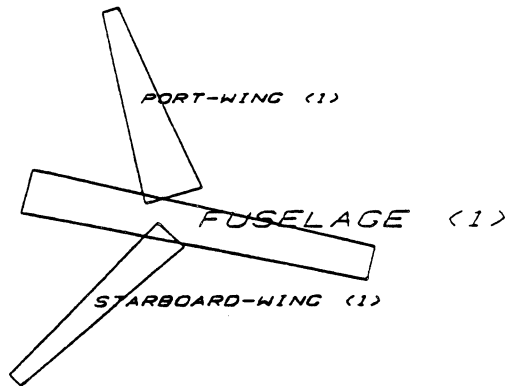


Figure I-26. Avion identifié dans la figure 24 ([Brooks 81a]).

n'est pas clair si tous les résultats cités sont obtenus avec une version intégrée du système ou des modules séparés, et quelle a été l'étendue de l'expérimentation.

2.3.5 Le système de Rochester

Motivation

Ce système [Ballard 77] [Ballard 78] cherche à limiter l'analyse d'une image au minimum nécessaire pour répondre à une question, et à créer un environnement dans lequel il est facile d'intégrer des nouveaux programmes d'application.

Description

Le système développé par Ballard, Brown et Feldman à l'Université de Rochester a été conçu pour répondre à des questions concernant une image. Par exemple: "trouver un bateau dans l'image aérienne d'un port". Dans ce contexte, la segmentation globale de l'image n'est plus nécessaire: elle est limitée au minimum requis pour la réponse. Les éléments de segmentation sont conservés d'une question à l'autre (pas de duplication d'effort). Le système se limite à des scènes 2-D. Les exemples traités sont: l'analyse d'une image aérienne d'un port (cf. figure I-27), et la recherche des côtes dans une image radiographique de la poitrine.



Figure 4.1a Aerial Mapping Photograph

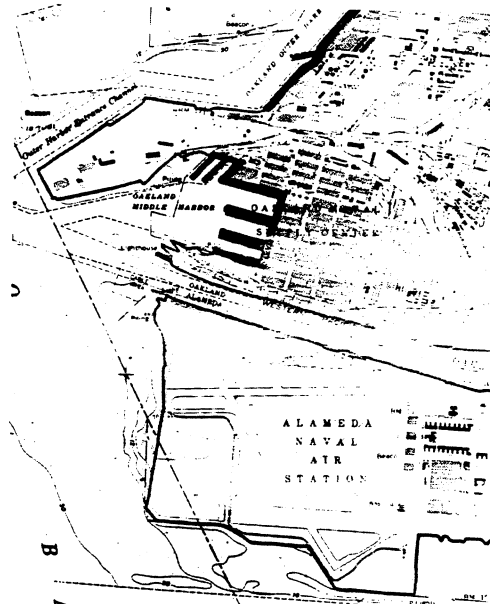


Figure 4.1b Topographic Map of Area in 4.1a.

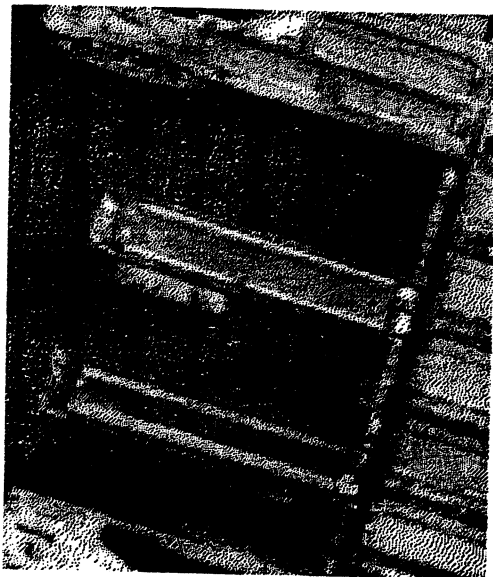


Figure 4.1c Halftoned Representation of Window from Digitized Version of 4.1a.

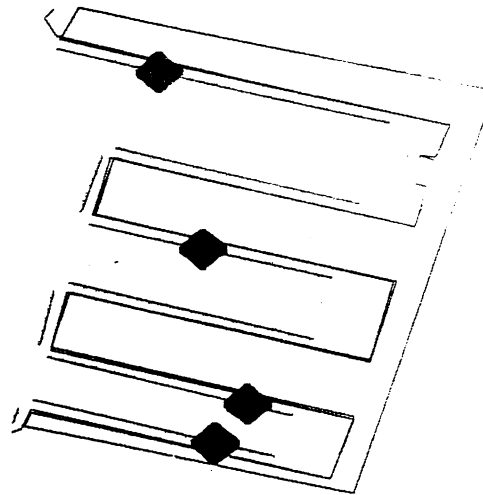


Figure 4.1d Coastline, Dock Area, Loci of Possible Ship Centers, Points of Application of Ship Template and Location of Ships.

Figure I-27. Exemple de fonctionnement du système de Rochester ([Ballard 78]).

Le système gère trois niveaux d'informations:

- 1) Le niveau image contient l'image et les résultats de la segmentation: lignes de contraste, régions, texture.
- 2) Les modèles constituent un réseau sémantique représentant l'apparence des objets, et des relations entre objets. Les relations sont matérialisées par des arcs de contraintes dans le réseau.
- 3) Le niveau correspondance est un réseau sémantique dont chaque noeud établit un lien entre des éléments du niveau image et un noeud-objet du niveau modèle.

En fin d'interprétation le niveau correspondance inclut la réponse du système.

Les questions posées au système ne peuvent pas être quelconques. Elles doivent appartenir à certaines classes de questions. Chaque classe de questions définit (ou est définie par) une procédure d'exécution qui est une procédure spécialisée d'interprétation d'images, écrite dans un langage de haut niveau. Elle impose une stratégie de recherche en fonction des procédures de correspondance attachées aux modèles. Une procédure de correspondance décrit algorithmiquement une façon de trouver l'objet dans l'image. Elle possède une précondition, une postcondition, un coefficient de fiabilité, et un coût. La procédure d'exécution choisit une procédure de correspondance en fonction de ces propriétés.

Voici l'exemple simplifié d'une procédure d'exécution pour la recherche d'une côte dans d'une radiographie:

```
PROCEDURE MatchRib(Node)
  BEGIN
    if there is an INSTANCE of Node then
      BEGIN
        print(Node,"already matched");
        return
      END
    else BEGIN
      Find x such that
        x is a RIB-PROCEDURE of Node,
        and RELIABILITY of x is acceptable,
        and COST of x is no greater than that of any other
          reliable RIB-PROCEDURE of Node;
      ApplyProc(x,Node);
      For each v such that
        v is a RIB,
        and v is a NEIGHBOR of Node
        do MatchRib(v)
      END
    END
  END
```

Evaluation

Ce système a été le premier à utiliser une représentation procédurale des connaissances dans un programme de vision réaliste. Cependant on peut se demander si le système reste maitrisable dans le cas d'une application plus complexe.

2.3.6 Le travail de Tsuji et Nakano

Motivation

Le système d'interprétation de ciné-angiogrammes de Tsuji et Nakano [Tsuji 81] tranche nettement avec les autres par la nature même de l'application. Contrairement à une image comme la figure I-16, dans laquelle le système visuel humain perçoit naturellement une structure riche à trois dimensions, notre oeil propose peu d'interprétations pour les radiographies du coeur telle que la figure I-28. Il fournit simplement une description à deux dimensions de type "réseau dans un plan". L'interprétation de ces images est une activité de raisonnement par des médecins spécialisés. La nature de ce problème explique l'approche par un système expert.

Description

Les ciné-angiogrammes sont des images d'un film radiographique du système artériel de la région du coeur. Les images d'une séquence varient d'une part à cause du battement du coeur, d'autre part à cause d'un changement de position volontaire de la caméra.

Les difficultés de l'analyse d'une séquence de ciné-angiogrammes sont nombreuses:

- le contraste est faible, les images sont très bruitées,
- le système artériel est complexe,
- l'interprétation nécessite souvent l'utilisation de plusieurs images,
- les images d'une séquence peuvent varier très rapidement avec un léger changement du point de vue.

De plus, il est impensable de construire un modèle géométrique complet du système artériel; et les médecins ne peuvent fournir une description algorithmique de leur activité. La solution choisie par Tsuji et Nakano est l'interprétation de lignes de contraste par un système expert.

On suppose qu'un "prétraitement" extrait de l'image les lignes de contraste. Chaque ligne (correspondant potentiellement à un vaisseau) est définie par une séquence de points et une épaisseur moyenne. La tâche du système est l'identification d'un certain nombre de vaisseaux importants en fonction d'un modèle implanté sous forme d'un ensemble de règles. La figure I-29 montre le modèle sous-jacent et donne quelques noms d'artères.

Le système contient plusieurs ensembles de règles de production:

- * Les **règles d'identification** servent à l'interprétation des lignes. Elles sont affectées d'un facteur de plausibilité. Voici un exemple d'une telle règle:

SI une ligne candidate

- 1) bifurque à 0%- 20% de l'artère coronaire descendante gauche (LAD)
- 2) bifurque de LAD avec un angle de 20-90 degrés
- 3) est plus courte que l'artère circonflexe gauche
- 4) est plus courte que LAD

ALORS

il est très plausible (0.9) qu'il s'agit d'une branche diagonale.

- * Les **règles de jugement** servent à comparer les interprétations de deux images différentes, et à effectuer un choix en cas de contradiction.

- * Les **règles de contrôle** définissent à chaque instant les sous-ensembles des règles activables. Ces règles représentent l'ordre dans lequel il faut procéder à l'identification.

L'analyse démarre systématiquement à partir de la sonde (qui a injecté un liquide rendant visibles les artères) et cherche d'abord l'artère antérieure descendante gauche (LAD). La figure II-21 montre les résultats obtenus pour la figure I-30. Le prototype du système contient 50 règles. Les premiers résultats sont encourageants, mais le travail de recherche n'est pas encore terminé.

Evaluation

Ce système représente un des rares cas de système expert en VO et donne des résultats significatifs pour un problème difficile. Cependant, l'intérêt de cette approche est lié à la nature très technique du domaine.



Figure I-28. Un ciné-angiogramme ([Tsuji 81]).

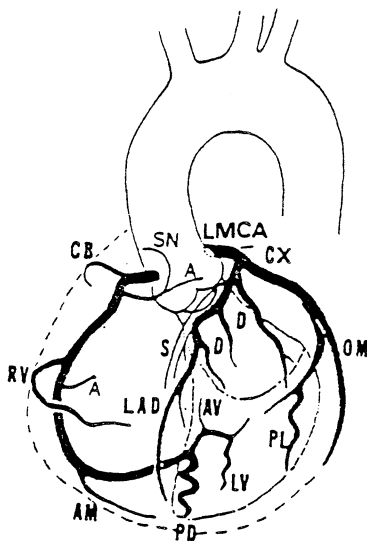


Diagram of right and left coronary arteries as views in the left anterior oblique projection. The abbreviations of branches of the left coronary artery in this picture are as follows:

- LMCA=left main coronary artery
- LAD =left anterior descending branch
- CX =circumflex
- D =diagonal branch
- S =septal branch
- OM =obtuse marginal branch
- PL =posterolateral branch
- CB =conus branch

Figure I-29. Modèle du système artériel.

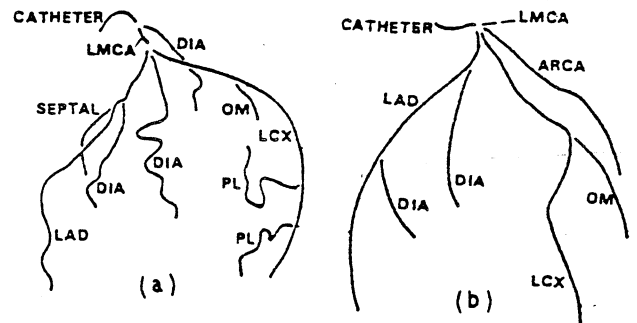


Figure I-30. Les deux meilleures interprétations de I-28.

2.3.7 Le travail de Nagao et Matsuyama

Motivation

Le système de Nagao et Matsuyama [Nagao 80] interprète des images aériennes multispectrales de régions habitées. Il réunit un grand nombre d'opérateurs d'analyse d'images avec des procédures sophistiquées d'interprétation contextuelle au sein d'une structure "sources de connaissances multiples avec tableau noir".

Description

Au plus haut niveau, le système de Nagao et Matsuyama est composé d'un ensemble de procédures spécialisées (KSs) communiquant par un tableau noir (figure I-31). Cette terminologie est la même que pour le système VISIONS. Le tableau noir contient la description et l'interprétation de l'image en plusieurs niveaux (figure I-32). Ces niveaux sont

- région élémentaire
- région caractéristique
- objet
- catégorie d'objet.

Les modèles d'objets sont bi-dimensionnels, ce qui est adapté aux vues aériennes.

La première phase d'exécution lisse l'image et construit les régions élémentaires en fonction de propriétés des pixels. Ensuite, une analyse de chaque région calcule les caractéristiques des régions. Une telle région peut être

- large homogène
- ombre
- porteuse d'ombre
- allongée
- végétation
- fortement contrastée

Toutes les informations concernant les régions sont stockées dans la table des propriétés, faisant partie du tableau noir. En fonction des caractéristiques, un exécutif central active différentes KSs pour certaines régions. Lorsqu'une KS est activée, elle analyse la région ayant provoqué son appel, et s'il existe à l'intérieur de celle-ci une région élémentaire non encore classifiée, essaie de la classifier. En cas d'échec, cette région est marquée "rejetée", et sera analysée à nouveau par la même KS seulement si des informations nouvelles sont trouvées dans le voisinage.

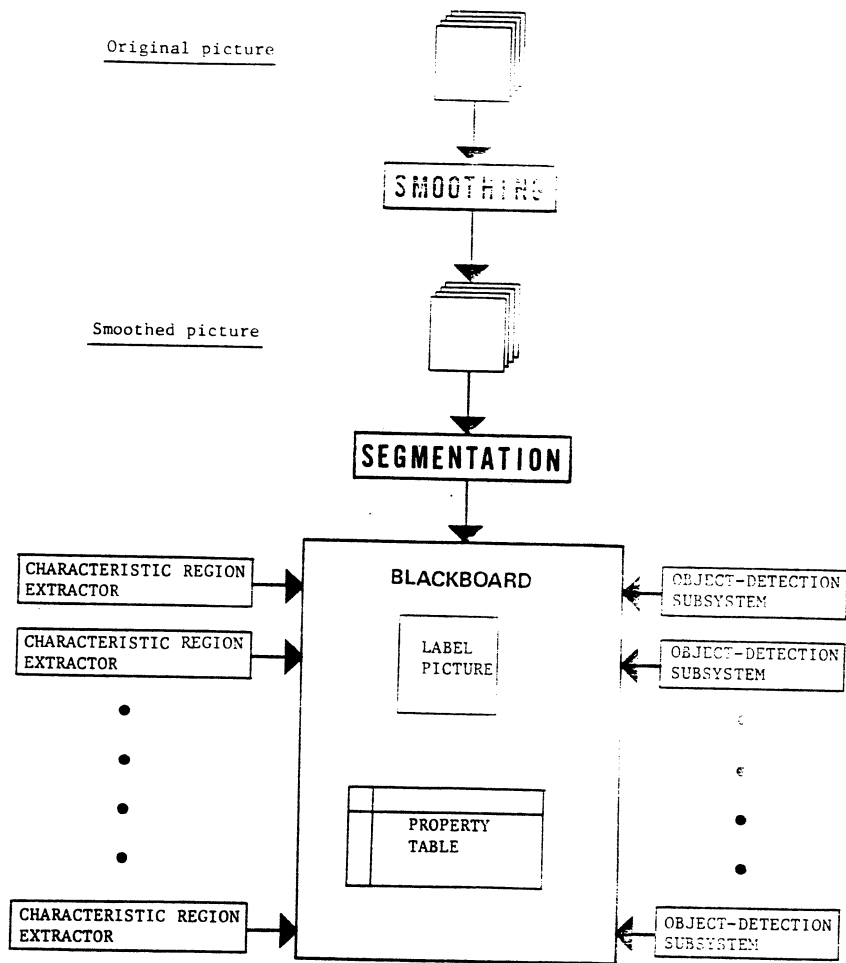
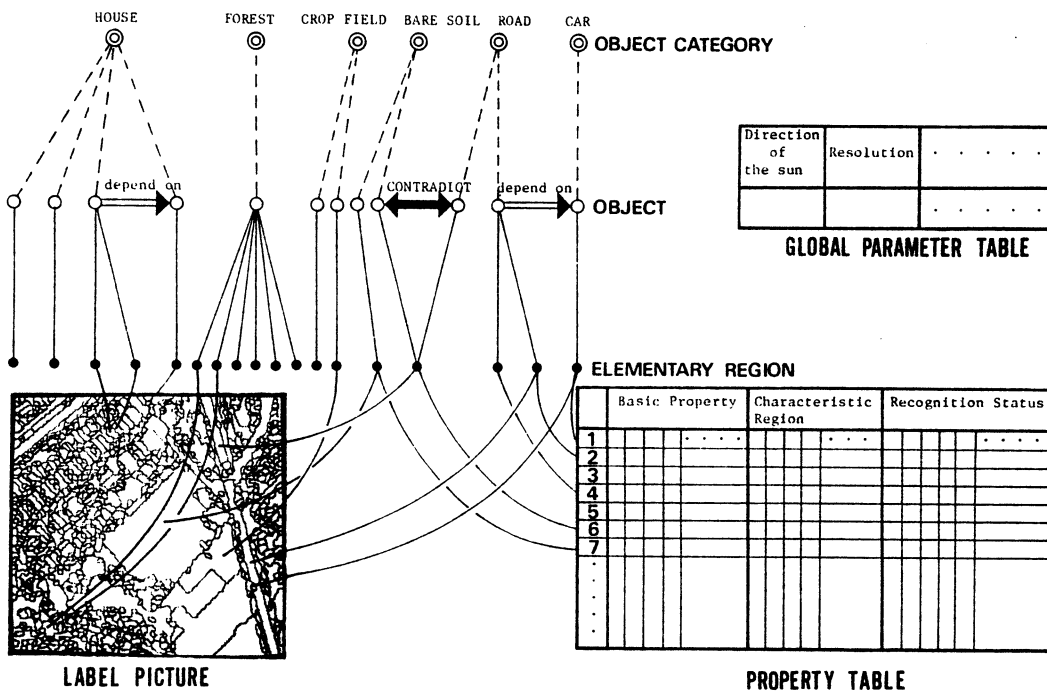


Figure I-31. Structure globale du système de Nagao et Matsuyama.



- ELEMENTARY REGION NODE
- OBJECT NODE
- ◎ OBJECT CATEGORY NODE
- IS-A RELATION
- PART-WHOLE RELATION
- ====> DEPENDENCY RELATION

Figure I-32. Structure du tableau noir.

Une région n'admet qu'une interprétation. En cas de contradiction, le système efface l'interprétation la moins vraisemblable, et recommence l'interprétation de la région; le contexte ayant changé, l'analyse sera différente. Dans certains cas de contradiction, le système peut décider de couper en deux une région élémentaire.

Evaluation

Ce système possède une structure globale claire et des performances remarquables. Cependant, dans la mesure où les KSs sont des programmes relativement complexes et tout de même interdépendants, la modifiabilité et l'extensibilité du système restent à prouver.

2.3.8 Systèmes de vision industriels

Parmi les systèmes de type industriel on trouve les systèmes effectuant une analyse de connexité dans des images binaires. Un objet est alors caractérisé par les propriétés de sa projection 2-D (surface apparente, périmètre, nombre de trous, moments d'inertie,...). Ces systèmes, commercialisés depuis 1979 (BBC, Machine Intelligence Corp., Automatix), ne font guère plus aujourd'hui l'objet de recherches et ne seront pas décrits ici.

Une autre classe de systèmes visant les applications industrielles est fournie par les "systèmes de vision 2-D" pouvant traiter des vues partielles d'objets connus. Une scène est dite "à 2 dimensions" si la position d'un objet est définie par une transformation dans un plan (translation, rotation, homothétie), souvent perpendiculaire à l'axe de la rétine de la caméra. La forme géométrique de la vue d'un objet est alors indépendante de la prise de vue.

Ce créneau revêt un intérêt commercial considérable, car il combine un grand potentiel d'applications industrielles et des possibilités d'implantation rapide à un prix compétitif. Les problèmes traités sont plus simples que dans le cas des systèmes des paragraphes précédents, mais ils ont l'avantage de nécessiter la mise en oeuvre de structures informatiques bien maîtrisables, aboutissant à des systèmes opérationnels. Ci-dessous nous présentons trois systèmes. Malgré la ressemblance des spécifications externes, ces systèmes sont de conceptions fort différentes, et offrent des possibilités d'applications différentes. Il est donc intéressant de noter qu'aucun de ces systèmes n'est plus général que les autres. On

peut facilement trouver des problèmes que seulement un de ces systèmes saurait résoudre. Le "plus petit commun multiple", capable de traiter des vues partielles de plusieurs objets dans une image à plusieurs niveaux de gris, avec des indices visuels variés et une bonne précision n'a pas encore vu le jour.

2.3.8.1 Le travail de Perkins

Motivation

Ce système [Perkins 78] a été conçu à l'origine pour la recherche d'un ou de plusieurs objets entièrement visibles dans une image à plusieurs niveaux de gris. Il a ensuite été adapté à des vues partielles d'un objet, puis à des vues partielles de plusieurs objets. L'accent dans la réalisation est mis sur la compétence des opérateurs de description de l'image.

Description

Etant donnée la taille d'une image, une analyse ascendante est forcément une opération très combinatoire, donc coûteuse en place mémoire et en temps de calcul. Il est donc surprenant de constater l'existence d'un système à caractère industriel qui procède de façon essentiellement ascendante: c'est le programme de PERKINS réalisé chez General Motors. Ce programme possède une structure à passages multiples, construisant successivement les niveaux d'une solution. Il utilise les niveaux suivants: pixel, point de contraste, ligne de contraste, segment polyligne, objet.

Son fonctionnement est le suivant: A l'image initiale (256 x 256, 64 niveaux de gris) est appliqué un opérateur de détection de lignes (opérateur de Hueckel). Les points ainsi trouvés sont chaînés pour former des lignes de contraste. Les lignes de contraste sont découpées en segments, un segment pouvant être un segment de droite ou un arc de cercle. La séquence de segments résultant d'une ligne de contraste est appelée "polyligne" (figure I-33).

Les polylignes sont comparées aux modèles d'objets (en essayant différentes rotations). Pour les hypothèses retenues, on vérifie la cohérence avec les points de contraste trouvés dans la première phase (figure I-34).

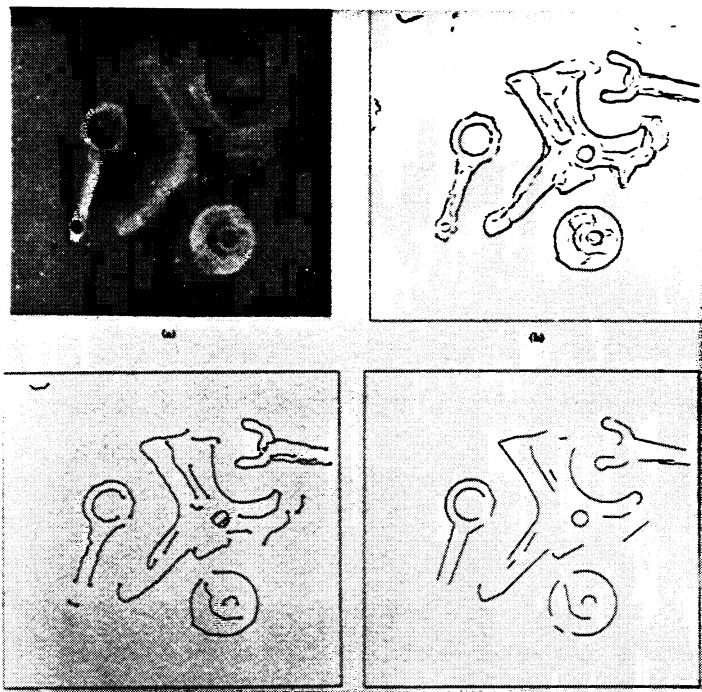


Figure I-33. Analyse d'image dans [Perkins 78]

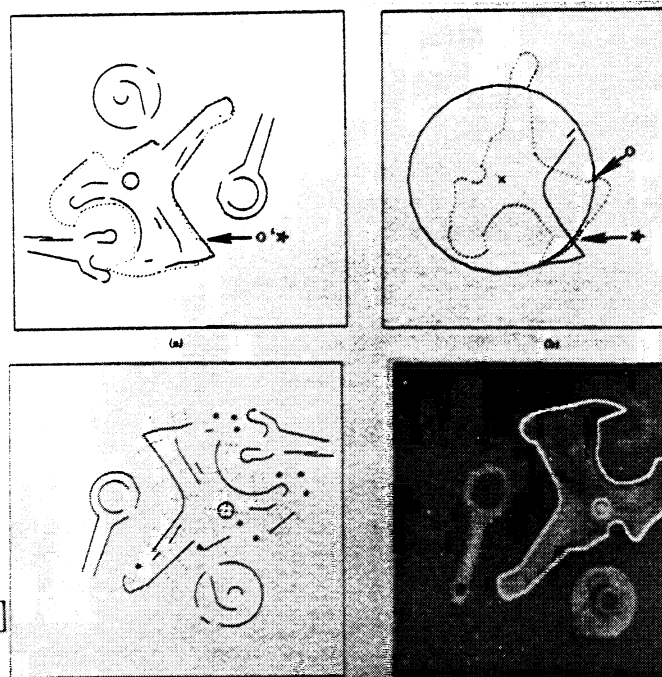


Figure I-34. Identification et localisation d'un objet

L'intérêt d'une analyse ascendante est clair:

- * Jusqu'au niveau polyligne l'analyse est complètement déterministe.
- * Pour des paramètres de formation de lignes de contraste donnés, il n'y a qu'une solution. C'est donc la situation idéale pour l'analyse ascendante, alors qu'une analyse descendante souffrirait d'un nombre quasi illimité de configurations image possibles.
- * Une analyse ascendante complète garantit que toute l'information contenue dans l'image a été prise en compte, puisque chaque pixel est utilisé au moins une fois.

L'analyse devient ambiguë seulement lorsqu'on établit des hypothèses d'objets, mais la combinatoire est faible dans le cas de la recherche d'un objet entièrement visible, vu la structure des modèles et leur nombre réduit. Cette phase commence par un tri des possibilités, puis le programme construit ces hypothèses de façon ascendante, cherchant un ensemble de polylignes compatibles avec un modèle.

Le seul aspect prédictif du système se situe dans la phase finale, où l'hypothèse est globalement vérifiée.

Evaluation

L'intérêt de ce programme réside dans les algorithmes de calcul d'indices visuels de l'image, plus sophistiqués que ceux des programmes décrits dans les paragraphes ci-après. Par contre, les stratégies de contrôle n'ont pas été travaillées. La structure à passages multiples et fortement ascendante, qui est simple sur le plan réalisation, implique beaucoup de calculs non-utilisés par la suite. Aussi le temps d'exécution est-il très long (plusieurs dizaines de secondes sur un IBM-370/168 pour une image 256 x 256). L'utilisation des polygones pour l'identification et la localisation des objets donne des résultats imprécis, et les algorithmes deviennent très combinatoires dans le cas de vues partielles.

2.3.8.2 Le travail de Yachida et Tsuji

Motivation

Ce système [Yachida 77] traite l'image à plusieurs niveaux de gris d'un objet entièrement visible. Il identifie l'objet et donne sa position avec une bonne précision. Il emploie une stratégie de reconnaissance optimale. La création des modèles se fait de façon interactive.

Description

Le système de Yachida et Tsuji sert à la reconnaissance et à la localisation d'objets à partir d'une image en noir et blanc (64 niveaux) montrant un ou plusieurs objets sans recouvrement.

Dans la première phase, le système travaille avec une image binaire à résolution réduite (128 x 128), dans laquelle les objets sont détectés par une analyse de connexité (figure I-35). Chaque connexe donne lieu à la création d'un ensemble d'hypothèses (figure I-36). Une image binaire est donc utilisée pour dégrossir le problème.

Dans la deuxième phase, une image à résolution complète est utilisée. Plus précisément, on prend un quart d'une image 256 x 256. Cette astuce permet de se contenter d'une mémoire

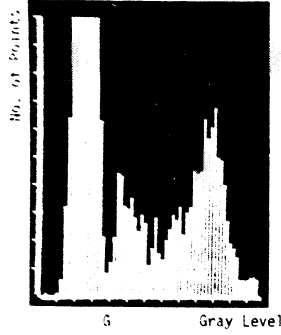


Fig. 4. An example of a gray-level histogram.

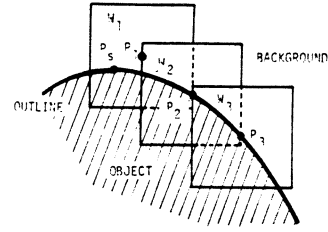
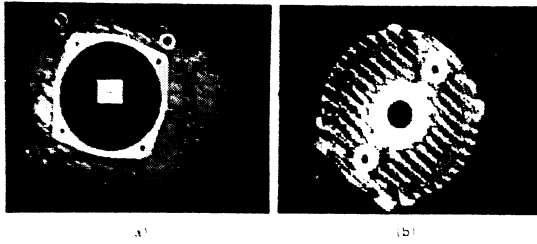


Fig. 6. Sequence of applying local windows. P_1 : The starting point by the preprocessor and the center of the first window W_1 . P_2 , last point detected in the current window and the center of the next window.



5. An example of an input image taken in the high-resolution picture.

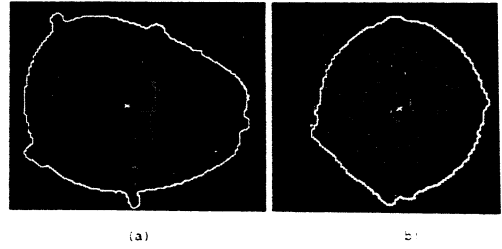


Fig. 7. An outline detected by the first-stage processor for the object shown in Fig. 5. The symbol x shows the centroid of the object.

Figure I-35. Analyse de l'image par le programme de Yachida et Tsuji. En bas à droite le résultat de la première phase.

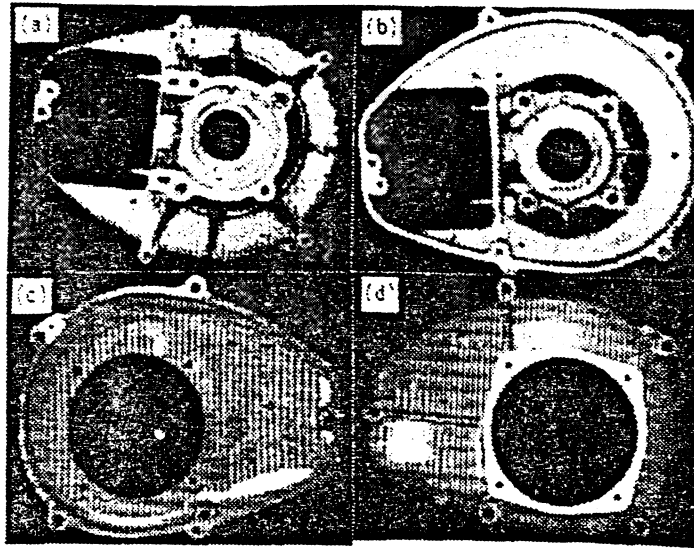


Figure I-36. Objets à contours semblables qui doivent être distingués pendant la deuxième phase.

d'image de 128 x 128, mais un contour complet doit donc tenir dans une fenêtre 128 x 128. Dans la deuxième phase, le système applique différents opérateurs de recherche d'indices visuels, en fonction de l'état courant de l'analyse. Le choix du prochain indice visuel à chercher tient compte des critères suivants:

- indice appartenant aux objets les plus plausibles
- indice discriminant les hypothèses les plus plausibles
- fiabilité des opérateurs
- coût des opérateurs.

Le programme de reconnaissance s'arrête lorsqu'une hypothèse devient plus plausible que les autres, ou lorsque tous les indices possibles ont été recherchés.

Dans une troisième phase, après identification, le système peut encore chercher des indices utiles pour la localisation ou l'inspection précise de l'objet (par exemple, des trous pour placer une vis).

Structures de contrôle utilisées

La première phase effectue une analyse ascendante afin d'établir des hypothèses d'objets. La deuxième phase possède un caractère nettement prédictif. Dans le contexte de ce système, la stratégie de prédiction-vérification montre des avantages importants:

- * Les "opérateurs spécialisés" de recherche d'indices visuels ne sont appliqués qu'à des endroits très limités de l'image. Il en résulte une optimisation considérable du temps d'exécution. Autrement dit, le système évite de chercher des informations qui lui seraient de toute façon inutiles.
- * La prédiction supplée l'incompétence des opérateurs spécialisés. Contrairement aux opérateurs de Perkins, les opérateurs utilisés ici ne sont pas capables de segmenter une ligne de contraste, c'est à dire de décider à quels endroits la découper en segments primitifs. Ces opérateurs ne permettraient donc pas, même dans le principe, d'effectuer une analyse ascendante. Cette faiblesse est compensée par la stratégie de prédiction-vérification. Bien entendu, cette "faiblesse" permet d'utiliser des méthodes plus simples.

Evaluation

Le point fort est le calcul dynamique du prochain indice visuel à chercher. Un deuxième point fort concerne la réalisation en FORTRAN sur PDP-8, avec des temps d'exécution allant de 20 à 90 secondes. La précision géométrique des résultats n'est pas indiquée. La publication de 1977 annonce le développement d'un matériel spécialisé permettant de réduire très fortement ces temps.

Le point faible du système vient du fait qu'il ne peut traiter des vues partielles, et cela pour plusieurs raisons:

- * La construction des hypothèses initiales à partir de contours dans une image binarisée.
- * Dans la deuxième phase, le type d'un indice recherché est imposé par la prédiction; si l'on cherche un segment de droite, on acceptera comme tel n'importe quelle ligne de contraste.

D'autre part, l'utilisation d'une image binaire dans la première phase expose le système à toutes les difficultés inhérentes à ce type d'images (éclairage, contraste).

2.3.8.3 La méthode LFF

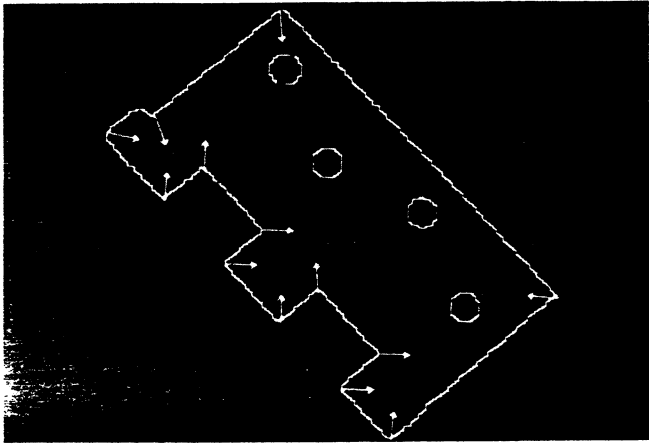
Motivation

Ce système [Bolles 82] identifie et localise un ou plusieurs objets partiellement visibles dans une image binaire. La reconnaissance se fait suivant une stratégie calculée à l'avance. Il fait partie d'une "station de vision" complète incluant des facilités de création de modèles d'objets.

Description

Autour de la méthode LFF ("Local Feature Focus method") Bolles et Cain ont construit un système d'analyse de scènes à vocation industrielle. Ce système identifie et localise des objets partiellement visibles dans une image binaire. Il utilise des combinaisons d'indices visuels locaux pour l'identification. Les indices actuellement implémentés sont les coins et les trous.

Une caractéristique essentielle et originale de ce système est l'analyse statique de l'ensemble des modèles d'objets. Cette



1. hole	2. hole	3. hole	4. hole
5. B	6. B	7. B	8. A
9. A	10. B	11. B	12. A
13. A	14. B	15. B	16. B

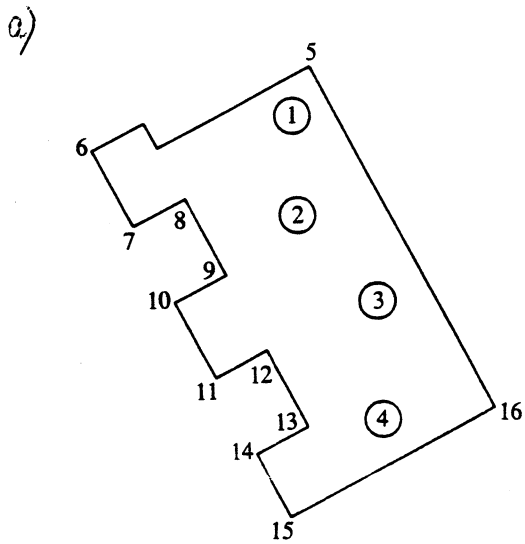
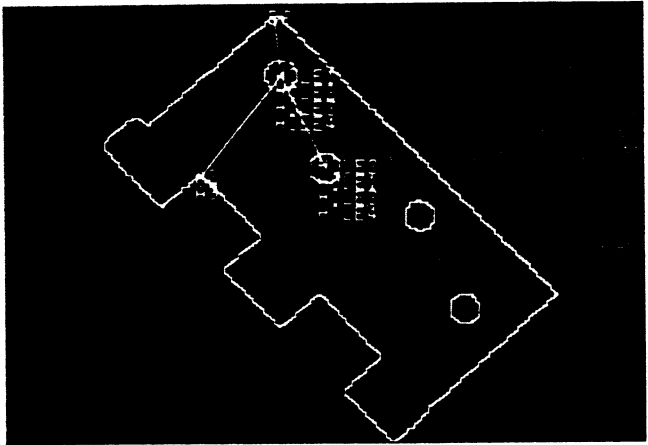
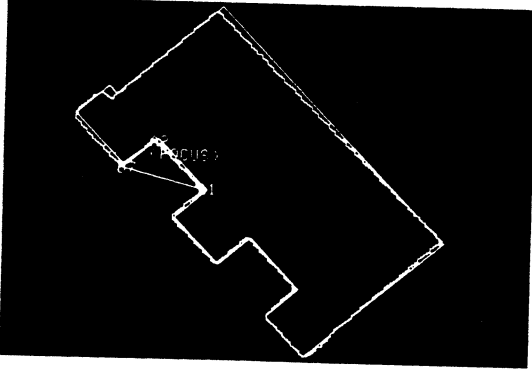
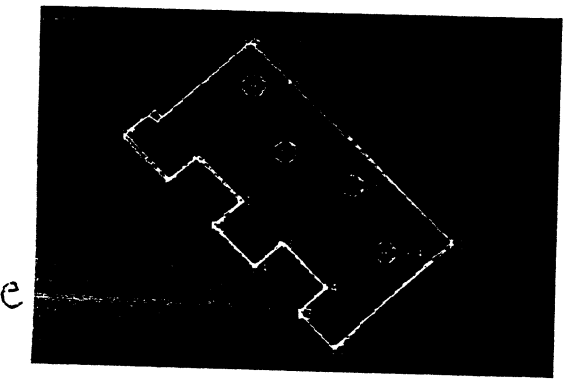
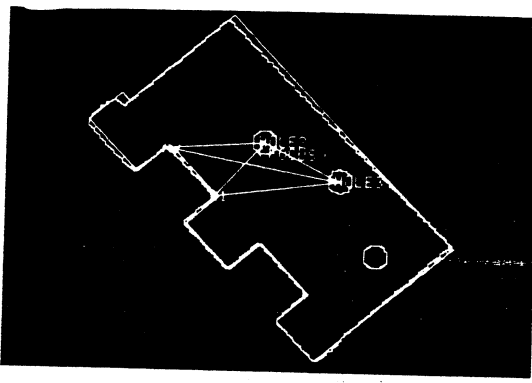
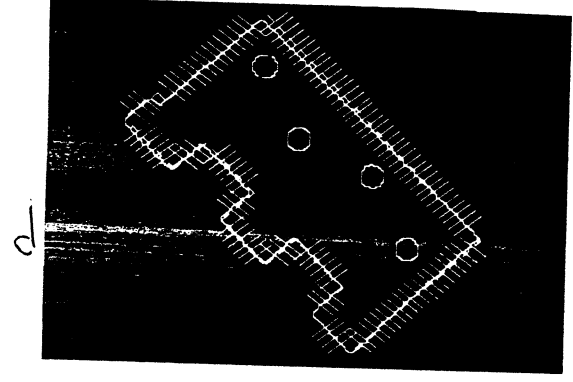


Figure II-37. Identification d'un objet par la méthode LFF.

- a) coins et trous détectés dans une image.
- b) indice central et indices voisins.
- c) hypothèse d'objet
- d) vérification de l'hypothèse.
- e) résultat final.



analyse est comparable à une compilation de programme. Elle a lieu une seule fois avant la phase de reconnaissance, et son but est la transformation des modèles en une forme facilement exploitable par le programme de reconnaissance. Notons que l'analyse est spécifique à un ensemble d'objets. Cela a deux conséquences

- Lors d'une modification de l'ensemble des objets considérés pour une application, il faut recompiler.

- La présence d'un objet inconnu (donc imprévu) peut saboter la reconnaissance de façon non-intuitive, même si cet objet ne ressemble pas aux autres.

Le résultat de l'analyse de modèles est un ensemble de stratégies de reconnaissance pour chaque objet, à base de groupements locaux d'indices visuels (local feature clusters). Les stratégies concernant un objet sont ordonnées suivant leur coût d'exécution. Chaque groupement possède un indice central, par rapport auquel il est indexé dans la base de données, et un ensemble d'indices voisins. Dans le cas idéal, l'ensemble des indices d'un groupement admet une seule hypothèse par rapport aux modèles considérés, même si certains indices ne sont pas visibles. Dans la phase d'analyse de modèles, le système construit de façon interactive tous les groupements locaux ayant ces propriétés.

Lors de l'analyse d'une scène, le programme procède en plusieurs étapes: d'abord tous les indices visuels sont extraits d'une image binarisée (figure I-37-a). Cela est effectué par un matériel spécialisé, donc rapide. L'indice central le moins coûteux (du modèle recherché) est choisi, et le programme essaie de trouver d'autres indices d'un groupement aux alentours (figure I-37-b).

En appliquant un algorithme de recherche de cliques maximales au graphe de toutes les affectations possibles entre indices image et indices du groupement modèle, une hypothèse d'objet est établie (figure I-37-c). Cette hypothèse est confirmée par une série de prédictions-vérifications. Finalement la présence du contour de l'objet est vérifiée dans l'image pour un certain nombre de points.

En cas d'échec lors d'une de ces étapes il y a retour-arrière systématique.

Evaluation

Le point fort du système est la synthèse d'une stratégie de reconnaissance par compilation d'objets. Les points faibles sont:

- * La segmentation complète fournit beaucoup de données même si elle est rapide.
- * Les stratégies compilées peuvent résulter dans un comportement non-intuitif du système, puisqu'elles éliminent (par définition) beaucoup d'informations des modèles.

2.4 RESUME ET CONCLUSION

A l'examen des systèmes de vision décrits dans les paragraphes précédents on est frappé par l'absence de base commune entre ces systèmes. Chaque réalisation "démarré à zéro" - l'ensemble des algorithmes mis en jeu, du premier traitement de l'image jusqu'à la décision finale, a entièrement été conçu et programmé par les auteurs de chaque système. Nous voyons là le symptôme d'un manque de maturité d'une science jeune qui ne dispose pas d'un cadre théorique bien développé, ni d'un ensemble d'outils bien accordés. Il reste un long chemin à parcourir avant d'arriver à l'état de maturité de l'analyse syntaxique - pour citer un exemple concret - qui dispose de la théorie des langages et des automates d'une part, et de nombreux algorithmes d'analyse syntaxique d'autre part, aux propriétés parfaitement étudiées.

En prenant plus de recul il est cependant possible de situer tous ces systèmes par rapport à l'état de l'art, et d'identifier leur contribution respective au progrès en la matière.

Le cahier de charges d'un système de VO universel et idéal comporte trois exigences qui semblent contradictoires (du moins à l'heure actuelle). Chacun des trois groupes dans lesquels nous avons classé les systèmes correspond à un compromis particulier entre ces trois critères:

- * La **généralité** du système par rapport aux tâches qu'il est capable d'effectuer.
- * L'**expertise** relative à une tâche précise, c'est à dire la quantité de connaissances spécifiques intelligemment utilisées au niveau de la vision.
- * La **performance** en termes de temps d'exécution et de coût du matériel mis en oeuvre.

Nous pouvons alors résumer les priorités pour chacun des trois groupes dans le tableau suivant.

	GENERALITE	EXPERTISE	PERFORMANCE
<p>Systèmes généraux (§2.3.3, 2.3.4)</p>	<p>classes de tâches assez large</p>	<p>forte, à caractère général déclarative</p>	<p>coût important, gros ordinateur, minutes de calcul</p>
<p>Systèmes dédiés (§2.3.5-2.3.7)</p>	<p>tâche spécifique</p>	<p>forte, à caractère spécifique plus ou moins procédurale</p>	<p>coût important, gros ordinateur</p>
<p>Systèmes industriels (§2.3.8)</p>	<p>tâche réduite</p>	<p>très faible</p>	<p>rapide (quelques secondes maxi), matériel assez bon marché (micro- ordinateur, électronique spéci- alisée</p>

CHAPITRE 3

CONCEPTS DE BASE POUR NOTRE TRAVAIL

Ce bref chapitre amorce la transition entre l'étude générale de la première partie et le travail personnel exposé dans les parties suivantes. Il présente des concepts pour l'architecture d'un système de vision à partir d'une réflexion (1) sur les systèmes décrits précédemment, (2) sur notre conception de l'IA et de l'algorithmique, et (3) sur une première expérience avec un système de vision. Nous identifions trois dimensions selon lesquelles l'architecture d'un système doit être optimisée, et que nous appelons structures de données (§3.2), problèmes algorithmiques fermés (§3.3), et structures de contrôle (§3.4). Nous introduisons une forme de coroutines, les algorithmes incrémentaux, qui combinent une grande efficacité au sens de l'algorithmique (faible complexité) avec la possibilité d'un contrôle explicite au sens de l'IA.

La plupart des exemples utilisés dans ce chapitre ont un rapport direct avec l'étude détaillée de la deuxième partie.

3.1 ARCHITECTURE D'UN SYSTEME DE VISION

Un système de vision peut être analysé suivant différents points de vue. Il est à la fois un capteur devant fournir des informations sur le monde extérieur, un système informatique contenant de nombreux algorithmes, et un élément d'un système de décision, d'un "robot intelligent" au sens de l'IA. La conception d'un système de vision doit intégrer ces trois points de vue dans une architecture cohérente. Cela nous amène à définir trois critères suivant lesquels il s'agit d'optimiser l'architecture d'un système de vision.

- * Les différentes structures de données intermédiaires utilisées pour représenter une scène doivent être significatives par rapport au monde réel.

- * La génération de ces structures de données doit être résolue par des algorithmes efficaces.
- * La structure de contrôle doit permettre d'adapter le comportement du système en fonction d'un objectif global.

Le tableau de la figure I-38 indique les structures de données et les structures de contrôle des systèmes que nous avons présentés dans les paragraphes précédents. Les problèmes algorithmiques n'ont pas reçu, en général, une attention particulière dans ces systèmes. En fait, les problèmes algorithmiques liés à la vision font l'objet de nombreuses recherches, mais se déroulant en dehors des contraintes sévères imposées par le contexte d'un système complet. Le livre de [Ballard et Brown 82] donne une excellente présentation d'une grande partie de ces recherches.

Dans ce chapitre, nous discutons les trois critères ci-dessus. Nous illustrons leur importance par des critiques à l'égard des systèmes décrits précédemment, et nous introduisons les choix à la base de notre travail.

3.2 STRUCTURES DE DONNEES INTERMEDIAIRES

Comme nous l'avons souligné dans les premiers chapitres, il n'existe pas de règles bien établies pour la décomposition du problème global de la vision en étapes successives entre l'image et l'interprétation finale. La définition des structures de données formant ces niveaux intermédiaires est pourtant un choix stratégique qui conditionnera le potentiel du système.

- * Il conditionne le potentiel d'application.
Plus la description de l'image est riche (types variés), et plus ses éléments correspondent à des éléments physiquement significatifs, plus il sera facile de résoudre un problème de vision à l'aide du système. Dans une bonne décomposition doivent apparaître des structures de données intermédiaires "utiles", c.a.d. correspondant à une réalité physique en vue, éventuellement, de la réalisation d'un réflexe visuel, ou significatives dans le dialogue avec un utilisateur.
- * Il conditionne les possibilités de réalisation.
La difficulté des problèmes algorithmiques posés par la construction des niveaux successifs conditionne les possibilités de réalisation et l'efficacité de l'ensemble. Une bonne décomposition est telle que le problème de construction d'un niveau de données à partir du niveau précédent admet des solutions algorithmiques efficaces.

Ces deux aspects - adéquation de la description au problème physique de la vision, adéquation aux moyens de réalisation - sont tout à fait indépendants, et rien ne prouve a priori qu'ils soient conciliables: l'existence d'un formalisme de description d'image satisfaisant les deux critères est encore à prouver. Ainsi, par exemple, la description 3-D proposée par David Marr, qui est une description de la scène en termes de surfaces et de volumes à base de cylindres généralisés, est certainement un formalisme de description puissant, mais il n'existe, à l'heure actuelle, aucune méthode générale pour la construire.

Nous illustrons ces deux aspects à l'aide d'un sous-problème important en vision: l'extraction de lignes de contraste, que nous traiterons en détail au II-2. Une approche qui a fréquemment été utilisée procède de la façon suivante:

On applique à l'image un opérateur de gradient, puis un seuil sur le module du gradient afin de binariser l'image. Ces opérations font apparaître des lignes épaisses et parfois interrompues. On cherche alors à éviter les interruptions de lignes par le bon choix de l'opérateur gradient et du seuil sur le module, puis on amincit par un algorithme de squeletisation de lignes.

Globalement, on décompose donc en trois structures intermédiaires:

S1: Image gradient
S2: Traits épais
S3: Squelette

Le problème de cette approche vient du fait que la deuxième structure intermédiaire - les traits épais - ne correspond pas à une réalité physique dans l'image ou dans la scène. Comme c'est généralement le cas pour une opération de seuillage, la forme de ces traits dépend fortement du bruit présent dans l'image, en particulier du bruit d'échantillonnage, donc des paramètres internes du capteur. Le seul intérêt de cette approche tient au fait que la binarisation amène à une importante réduction de l'information de l'image. Par contre, avec la binarisation on perd toute information concernant la direction du gradient, sans laquelle la squelettisation des traits devient un problème algorithmique difficile et coûteux.

Nous avons choisi une autre approche. On définit une ligne de contraste comme étant une "crête" dans le gradient de l'image. Cette approche amène à l'étude des sous-problèmes suivants:

S1: Image gradient
S4: Points de crête

	Niveaux de description	Modèles	Structures de contrôle
Guzman	lignes de contraste faces objets		ascendant avec heuristiques
Waltz	lignes et noeuds étiquetage		propagation de contraintes
Shirai	lignes de contraste segments de droites et d'ellipses	collection de modèles 2-D	génération d'hypothèses prédiction-vérification
MSYS	plusieurs niveaux de régions	caractéristiques des régions contraintes locales	propagation de contraintes
ARGOS	image multi-spectrale régions	caractéristiques des régions	"locus search" double parcours heuristique d'un graphe
VISIONS	graphe RSV surfaces volumes	cônes généralisés surfaces 3-D réseau sémantique	procédural prédiction-vérification
ACRONYM	segments de droites bandes 2-D	cônes généralisés compilation en caractéristiques 2-D graphe d'objets graphe des prédictions	homomorphisme de graphes système d'inégalités numériques prédiction-vérification

Univ. Rochester	procédures spécialisés	réseau sémantique 2-D	procédural
Nagao et Matsuyama	lignes de contraste	implicites dans règles de correspondance	règles de contrôle
Tsuji et Nakano	image multi-spectrale régions élémentaires régions caractéristiques	procédures propriétés des régions	ascendant
Perkins	segments de droites et de cercles polygones	2-D: polygones	ascendant
Yachida et Tsuji	silhouette dans image binaire segment de droite, de cercle, texture, petit trou	collection de caractéristiques	ascendant, puis arbre de décision calculé dynamiquement
LFF	angles petits trous groupement d'IVs	modèles 2-D compilation en groupements	recherche de cliques maximales
Garvey (chap.III-3)	prédicats à base de pixels	prédicats à base de pixels contexte pour les objets	génération de plans par décomposition du problème

Figure I-38. Tableau comparatif des caractéristiques principales des systèmes présentés.

Nous retrouvons ici le sous-problème - assez difficile - du calcul de l'image gradient. Ensuite, le passage direct aux points de crête présente l'avantage d'une définition simple et intuitivement satisfaisante de la notion de ligne de contraste. Les points de crête sont physiquement bien définis, puisqu'ils correspondent aux endroits de l'image gradient où le rapport signal sur bruit est maximal, et peuvent être obtenus par un calcul local sans difficulté.

3.3 PROBLEMES ALGORITHMIQUES "FERMES"

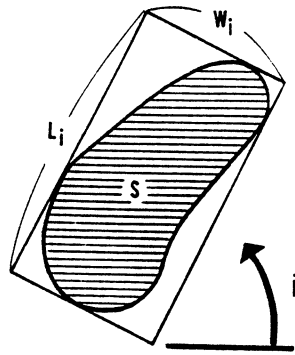
De par la diversité des structures de données manipulées la VO est une source riche en problèmes algorithmiques. Bien souvent, ces problèmes paraissent faciles à résoudre et sont considérés marginaux par rapport aux objectifs de la recherche en cours. Il y a alors une tendance naturelle à implanter la première solution qui vient à l'esprit, et de programmer vite, sans approfondir. Or, sous l'apparence de facilité, bon nombre de ces problèmes cachent des embûches et des subtilités qui se manifesteront petit à petit. Dans les cas bénins cela provoquera une perte de temps. Dans d'autres cas, toute la conception d'un système pourra être remise en cause.

Un problème caractéristique de ce genre est la recherche de segments de droites dans une ligne (traité en détail au paragraphe II-3.3). Il existe une variété étonnante d'algorithmes pour ce problème, qui possèdent des propriétés très différentes. La difficulté vient du fait qu'un segment de droite est défini dans un espace réel, alors qu'une séquence de points est définie dans un espace discret. L'adaptation de la définition d'un segment de droite à ce cas peut se faire de différentes façons dont aucune n'est complètement satisfaisante.

Bien entendu, les problèmes rencontrés en vision ne sont pas toujours aussi espiègles. Cependant, même pour des problèmes plus faciles, on trouve fréquemment des solutions choquantes du point de vue algorithmique.

Par exemple, [Nagao et Matsuyama 80, p.97] utilisent le "rectangle minimal encadrant" pour caractériser la forme d'une région (figure I-39). Ce rectangle, défini comme "le rectangle encadrant la région et ayant la plus petite surface", est obtenu à l'aide de l'algorithme suivant:

- 1) Calculer les 9 rectangles encadrants avec les positions 0° , 10° , ... 80° dans l'image. (Il suffit de considérer les angles entre 0° et 90°).
- b) Choisir celui qui a la plus petite surface.



$$F_i = \frac{L_i \times W_i}{S} ;$$

$$i = 0^\circ, 10^\circ, 20^\circ, \dots, 80^\circ$$

S = area of a region

$$\text{FIT} = \max_i F_i$$

$$\text{ELONG} = L / W$$

DIREC = direction of L

(L and W denote the length of the long side and short side of the MBR.)

Figure I-39. Calcul du rectangle encadrant minimal d'une région dans [Nagao et Matsuyama 80].

Cet algorithme a deux inconvénients majeurs:

- a) Il comporte neuf calculs de rectangle encadrant, d'où un facteur neuf dans le temps de calcul.
- b) La précision directionnelle du résultat est de $\pm 5^\circ$. Une amélioration de la précision implique une augmentation proportionnelle du temps de calcul.

L'objectif du calcul, la caractérisation d'une forme indépendante de sa position, peut également être atteint avec une définition légèrement différente: à la place du rectangle encadrant minimal on peut utiliser le rectangle encadrant parallèle aux axes d'inertie de la région. Cela implique un algorithme parcourant deux fois les points de la région:

- 1) calcul des axes d'inertie
- 2) calcul du rectangle encadrant.

L'avantage de cette solution est d'éviter le facteur 9, et surtout, de donner un résultat dont la précision est limitée seulement par la résolution spatiale de l'image, et non par le principe même de l'algorithme. Notons également qu'il est intéressant d'utiliser dans tous ces calculs (pour les deux algorithmes) seulement les points du bord de la région: les résultats seront aussi significatifs, mais au lieu d'être d'ordre N - nombre de points de la région - les calculs sont d'ordre \sqrt{N} .

Notre propos ici n'est pas de critiquer le système de Nagao et Matsuyama (dont nous avons dit beaucoup de bien ailleurs).

Nous voulons seulement illustrer le fait qu'un système de vision complet doit combiner des dizaines, voir des centaines d'algorithmes, et qu'en l'absence d'une étude algorithmique bien documentée et accessible, toute réalisation contiendra toujours quelques "mauvais" algorithmes.

3.4 STRUCTURES DE CONTROLE

3.4.1 Nécessité d'un contrôle

Un système de vision peut utiliser une bonne structuration en niveaux intermédiaires, employer des algorithmes efficaces pour les passages entre niveaux, et tout de même montrer un comportement global inacceptable et inefficace. Nous illustrons ce point par l'exemple de la structure initiale du système CAIMAN (présenté dans la deuxième partie).

Considérons les trois modules suivants:

- * suivi d'une ligne de contraste
- * analyse d'une ligne: approximation polygonale
- * analyse d'une ligne: recherche de points d'inflexion

La façon la plus simple de combiner ces modules est l'enchaînement séquentiel:

- 1) Itération sur les points d'une ligne, stockage des points dans un vecteur VP.
- 2) Approximation polygonale des points dans VP, stockage des résultats dans un vecteur de segments VS.
- 3) Recherche des points d'inflexion à partir de VP, stockage des résultats dans un vecteur de points d'inflexion VF.

Cette structure est parfaitement adéquate pour l'étude et la mise au point des algorithmes. Mais lorsqu'on l'exploite pour la solution de problèmes de vision précis on constate qu'elle manque de flexibilité.

- 1) Supposons que l'on ait l'objectif de trouver un point d'inflexion situé entre la deuxième et la troisième droite d'une ligne. Avec la structure séquentielle on exécute les actions suivantes:
 - a) Acquisition de tous les points de la ligne dans un vecteur VP.
 - b) Approximation polygonale de l'ensemble de la ligne avec un vecteur résultat VS.
 - c) Calcul des points d'inflexion dans un vecteur VF.

d) Comparaison des tableaux VS et VF.

Il est évident que, si la ligne comporte plus de trois segments de droites, tous les calculs concernant la ligne au delà du début du troisième segment de droite sont inutiles.

- 2) Il faut également noter un problème d'apparence plus technique. Il est difficile de fixer la dimension des vecteurs VP, VS, VF. La longueur des lignes de contraste varie d'une image à l'autre et n'admet pas de borne supérieure évidente. Dimensionner tous les tableaux d'un système en fonction du cas le plus défavorable amènerait à un gâchis de place mémoire qui risque de limiter le potentiel du système par ailleurs, alors que cette place sera rarement utilisée complètement. Une allocation sous forme de listes alourdit la programmation (surtout dans un langage comme FORTRAN).

3.4.2 Contrôle incrémental

Le manque de flexibilité que nous venons de décrire vient du fait que, avec la structure séquentielle du paragraphe précédent, l'ordre des opérations est complètement figé, alors qu'il serait utile, en présence d'un objectif concret, de limiter les opérations à celles qui sont nécessaires pour atteindre cet objectif. Ce rôle est joué, dans le cadre d'un système d'IA, par la structure de contrôle. Dans le contexte algorithmique imposé par la vision de bas niveau, nous introduisons cet aspect par le biais d'une structure incrémentale implémentée par un ensemble de coroutines.

Pour l'exemple traité, cela nous donne la structure suivante: au lieu d'être appelés une seule fois à la fin du suivi de ligne, les programmes d'analyse (approximation polygonale et calcul des points d'inflexion) sont appelés pour chaque point, et calculent leurs résultats au fur et à mesure. Les programmes d'analyse de lignes peuvent alors être considérés comme des automates ayant comme bande d'entrée la séquence des points trouvés par le suivi, et produisant une bande de

sortie. Ces automates sont "accrochés" à l'intérieur de la boucle de suivi, comme le montre la figure I-40. On peut également décrire cette structure comme un ensemble de processus communicants, recevant à chaque itération du suivi un message indiquant le point courant du suivi (avec d'autres paramètres éventuellement). Ce point de vue amène naturellement à une programmation en termes d'objets qui nous semble une voie d'avenir intéressante dans ce contexte.

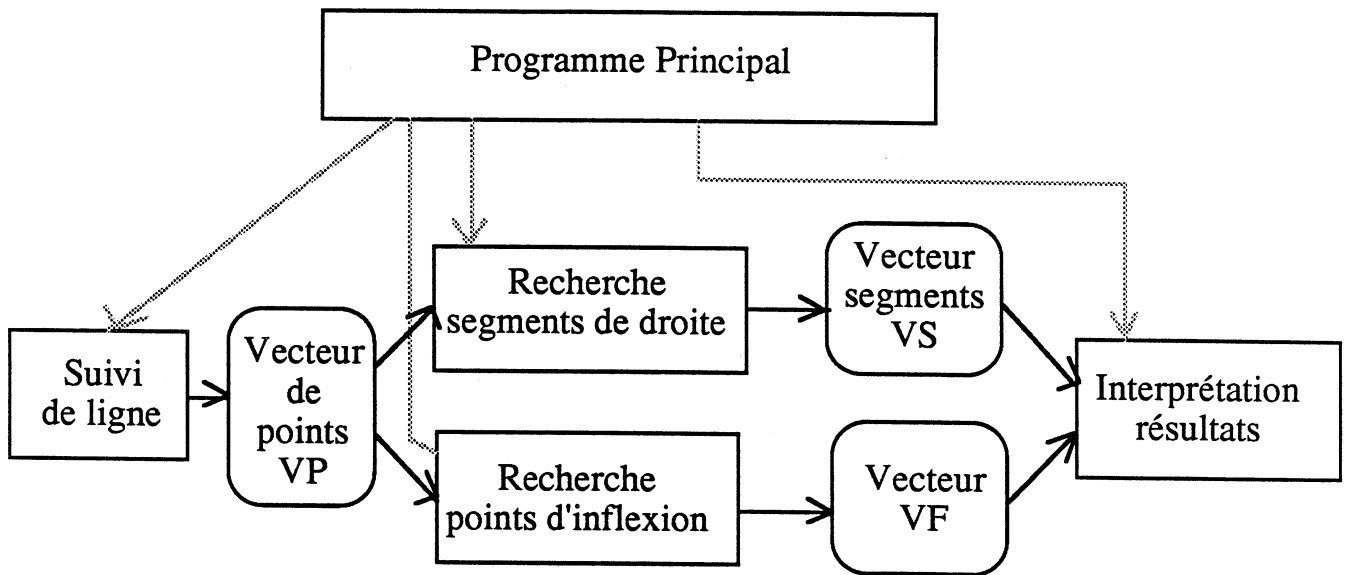
Cette structure présente alors les avantages suivants:

- * Possibilité d'un "arrêt sur but".
Etant donné que les résultats de chaque module sont produits au fur et à mesure, et peuvent être interprétés immédiatement, l'ensemble du système pourra bien s'arrêter, dans notre exemple, au premier point d'inflexion entre la deuxième et la troisième droite, ou à la troisième droite si un tel point n'existe pas. L'exécution des algorithmes se déroulant en quelque sorte en parallèle, il faudra seulement ajouter un mécanisme de communication arrêtant la boucle de suivi au bon moment (propagation de buts).
- * Place mémoire limitée. Le vecteur VP disparaît; il n'est plus nécessaire de stocker les points, puisqu'ils sont utilisés immédiatement - nous dirons qu'ils sont "consommés" - par les deux automates. De même, les vecteurs VS et VF deviennent inutiles, la comparaison des résultats pouvant se faire au fur et à mesure. Il y a donc gain de place, et le problème délicat du choix de la taille des vecteurs ne se pose plus.

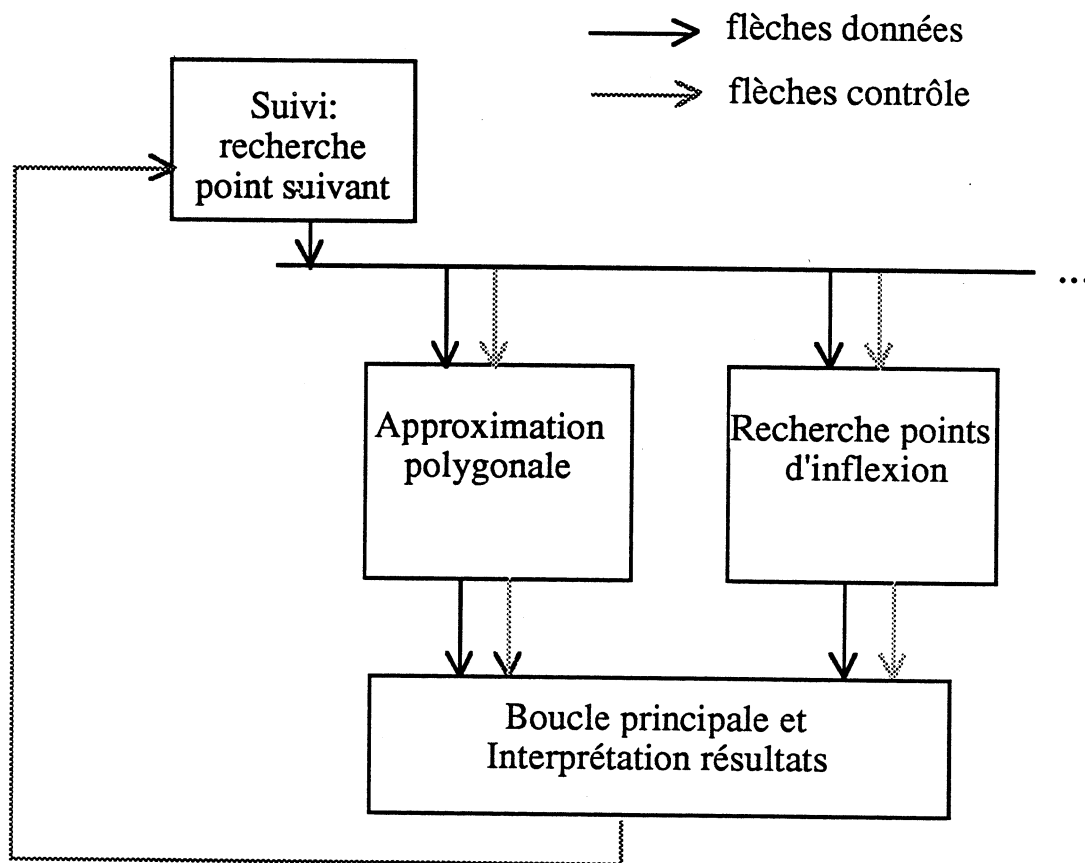
3.4.3 Algorithmes incrémentaux

L'intérêt d'une structure de contrôle incrémentale est à analyser au cas par cas, car son avantage par rapport à une structure séquentielle est lié à une propriété des algorithmes employés que nous appelons incrémentalité.

Si on utilise des algorithmes non incrémentaux, cet avantage disparaît. Dans le cas de notre exemple, si le calcul d'approximation polygonale avait besoin de l'ensemble des points pour son déroulement, cet algorithme serait bien obligé de constituer le vecteur VP et d'attendre la fin du suivi avant de pouvoir conclure à propos de l'objectif. Pire: si le calcul des points d'inflexion en avait également besoin, il constituerait lui aussi une copie d'un tel vecteur. L'arrêt sur but serait impossible, et cette structure ne présenterait aucun avantage.



(a) Enchaînement séquentiel.



(b) Structure incrémentale avec coroutines.

Figure I-40. Deux structures pour un programme d'analyse d'une ligne.

L'incrémentalité représente le plus faible degré de complexité algorithmique. En effet, un algorithme à taille mémoire finie et transformant une séquence d'entrée en une séquence de sortie a forcément un temps de réponse borné (s'il ne boucle pas). Ainsi, un algorithme incrémental est caractérisé par deux propriétés.

- * Il utilise une place mémoire fixe et connue. Il permet donc en particulier de traiter une séquence d'entrée de longueur non-bornée. Cela est intéressant, par exemple, dans le cas des images "infinies" obtenues par saisie ligne par ligne d'une scène mobile (tapis roulant, fabrication continue).
- * Le temps d'exécution à chaque pas est borné par une constante K . Par rapport à la séquence d'entrée nous avons un temps linéaire: le traitement d'une séquence de longueur n se fait en un temps d'ordre $n \cdot K$.

Il faut noter que, d'un point de vue puriste, notre notion d'algorithme incrémental n'est pas tout à fait équivalente à celle d'un automate d'états finis. Nous supposons en effet qu'il n'y a pas de problème de débordement arithmétique dans les calculs. En ce sens, un algorithme incrémental peut posséder un ensemble d'états non borné. Le même argument s'applique au temps de calcul: nous supposons que le temps d'une opération numérique est borné par une constante indépendante des arguments. D'un point de vue pratique, cela signifie qu'il doit y avoir cohérence entre la taille de l'image et la taille du mot de l'ordinateur utilisé. Dans certains cas cela demande une attention particulière, comme dans les images infinies.

Bien entendu, tout problème n'admet pas forcément une solution incrémentale. A l'opposé, un algorithme global nécessite la totalité des données d'entrée avant de produire des résultats. Cela signifie que pour une séquence d'entrée de longueur N , il a un besoin en place mémoire au moins d'ordre N (pour stocker les données) ce qui peut poser les problèmes mentionnés avant. La plupart des algorithmes numériques d'approximation et d'optimisation sont globaux.

PARTIE II

DE LA CONSTRUCTION D'INDICES VISUELS

Cette deuxième partie développe les éléments d'un système de description d'images. Le rôle d'un tel système est de construire une représentation de l'image pouvant être directement mise en correspondance avec des modèles par un module d'interprétation. Cette représentation, dans laquelle nous nous limitons à décrire l'image en tant que structure à deux dimensions, est construite en passant par des structures de données intermédiaires. Nous avons choisi les lignes comme éléments de base de la description, en particulier les lignes de contraste. Les problèmes que nous traitons sont donc la construction de lignes à partir d'une image, et l'analyse de lignes en vue de produire des "indices visuels", qui seront, par exemple, des segments de droites ou des angles.

Le chapitre 2 de cette partie traite l'extraction de lignes de contraste, le chapitre 3 l'analyse de lignes. Le chapitre 4 décrit une réalisation dans laquelle l'ensemble des algorithmes décrits a été mis en œuvre, et quelques applications.

CHAPITRE 1

INTRODUCTION

Le chapitre 1 justifie d'abord notre choix des lignes de contraste comme structure de données intermédiaire; le §1.1 fait ainsi suite au §3.2 de la première partie. Le §1.3 présente quelques aspects généraux de la géométrie digitale - la géométrie sur une grille discrète de points. Comparée à la géométrie habituelle dans un espace continu, la géométrie digitale possède des caractéristiques particulières qui ont des conséquences parfois surprenantes, et qui marquent les problèmes de vision de bas niveau d'un cachet original.

1.1 APPROCHE ET MOTIVATION

Par définition, une ligne de contraste est un lieu de forte variation de l'intensité dans l'image, ou une crête dans l'image gradient. Cette définition est en bon accord avec l'intuition, et implique trois avantages importants par rapport aux critères discutés aux paragraphes I-3.1,3.2:

- * Elle est indépendante du niveau d'éclairage.
- * Elle est indépendante des variations faibles ou systématiques de l'éclairage (gradient constant ou faible).
- * Elle minimise l'influence du bruit du capteur. Les crêtes sont précisément les endroits où l'image gradient possède le meilleur rapport signal/bruit. (Si l'on suppose que le bruit est uniforme et indépendant du niveau du signal).

D'autre part, une ligne étant une entité géométrique à une dimension, sa représentation informatique est simple et amène naturellement à des algorithmes de traitement à nature séquentielle et de faible complexité.

Alternativement, nous aurions pu choisir la région homogène comme primitive de base, comme c'est le cas, par exemple, dans le système MSYS. Cependant, la notion de région, définie comme "ensemble connexe et maximal de pixels homogènes" n'a pas les propriétés que nous venons de citer pour les lignes. Au contraire, la segmentation en régions souffre très généralement des problèmes posés par le seuillage d'une image, à savoir qu'elle dépend fortement des conditions d'éclairage et du bruit. Plus précisément, en dehors des zones à forte variation d'intensité, les frontières des régions homogènes dépendent des aléas de l'échantillonnage, elles sont donc déterminées par les paramètres internes du capteur et non par la "réalité physique".

Nous pensons que la construction de régions est utile en analyse d'images, mais seulement dans un deuxième temps, après une première segmentation à l'aide de critères plus stables, comme la construction de lignes de contraste. La recherche de régions correspond alors à la construction de lignes fermées.

Il existe un troisième type de primitive de description: à partir d'un dégradé dans une image - une région dans laquelle l'intensité varie d'une façon continue et cohérente - on peut, par l'intermédiaire d'un système à dérivées partielles, construire directement des éléments de surfaces à trois dimensions (cf. "shape from shading" [Horn 77, Ikeuchi et Horn 81]). Nous n'avons pas suivi cette possibilité, car elle nécessite une forte puissance de calcul et un capteur à forte résolution (par exemple 256 niveaux de gris).

1.2 LA GENERICITE DE LA NOTION DE LIGNE

Un deuxième aspect intéressant de l'utilisation de lignes vient de la genericité de cette notion par rapport au type d'image considérée (nous pensons ici aux images intrinsèques de [Barrow 78]). Une "crête dans le gradient" correspond à des propriétés différentes selon la nature de l'image.

- * Dans une image en noir et blanc une "ligne de contraste" est liée à une discontinuité dans l'intensité, pouvant être produite par un contour d'objet, une ligne interne, une ligne d'ombre, ou un reflêt.
- * Dans une image en couleur, une "ligne de couleur" est produite par un contraste dans une combinaison des trois plans chromatiques, donnant une information plus fine que le contraste en noir et blanc.

- * Dans une image de distances, une discontinuité de distances définit une arête du contour extérieur d'un objet.
- * Dans une image d'orientations, une discontinuité définit une arête interne.

Dans tous ces cas, une ligne peut aussi être définie à partir d'une configuration plus complexe du gradient. Par exemple, une double-crête dans le gradient, ce qui correspond à un trait fin, comme l'image d'un fil.

La notion de ligne recouvre donc une grande variété de phénomènes, ce qui augmente d'autant l'intérêt des algorithmes analysés dans les chapitres suivants.

1.3 GEOMETRIE DIGITALE

Le traitement informatique d'une image doit faire face à un problème fondamental du traitement digital du signal: la donnée du signal est une représentation échantillonnée d'un phénomène physique pour lequel nous possédons des modèles continus et la majeure partie des caractéristiques du signal auxquelles nous nous intéressons sont définies en termes de ce modèle continue (comme par exemple les dérivées).

Dans le cas d'une image en noir et blanc l'image est formée à partir de valeurs d'intensité lumineuse, quantité physique continue, définie dans le plan de l'image. La seule information que nous ayons concernant cette quantité physique est sa représentation par la matrice de pixels, résultant d'un échantillonnage suivant quatre dimensions: échantillonnage dans le temps, en intensité, et dans deux dimensions de l'espace.

"Valeur échantillonnée" signifie: valeur entière (tronquée ou éventuellement arrondie) dans un système de mesure propre au capteur. Nous ne connaissons donc pas les valeurs exactes de la luminosité. D'autre part, le capteur introduit un bruit d'échantillonnage. Il est raisonnable de considérer que ce bruit est distribué de façon uniforme dans l'intervalle $[-0.5, +0.5]$, ou $[0,1]$. Par les lois de la physique, les valeurs mesurées sont également affectées d'un bruit interne au capteur. On peut estimer que ce bruit est gaussien et de moyenne zéro.

Notre connaissance du signal est donc forcément incomplète. Toutefois nous pouvons chercher à l'améliorer à l'aide d'un capteur plus sophistiqué, avec une meilleure résolution spatiale, un échantillonnage plus fin. Mais ces améliorations ne sauraient changer le fond du problème, car la nature impose

ses limites. Comme la plupart des grandeurs physiques, la luminosité n'est continue qu'au niveau macroscopique. Cette continuité vient de l'intégration, dans le temps et dans l'espace, d'un phénomène discret et aléatoire qui est la nature corpusculaire de la lumière. Si nous poussons trop loin la finesse de notre capteur caméra digitale, le modèle continu ne s'appliquera plus et nous ne saurons plus interpréter les mesures.

Devant le problème de calculer des propriétés d'un signal échantillonné nous avons le choix entre deux approches:

- a) Dans une première étape, recouvrer explicitement la fonction continue sous-jacente, et effectuer les calculs à partir de cette fonction.
- b) Donner une définition des propriétés recherchées en termes de valeurs échantillonnées, et effectuer les calculs directement à partir des données. Cette approche est appelé **géométrie digitale**.

Bien entendu, à la limite ces deux approches se confondent. Cependant, quelques problèmes sont plus spécifiquement liés à la première approche.

- * Avec un lissage de l'image on perd éventuellement des informations de détail qu'une méthode prédictive ou d'accumulation de plausibilités pourrait exploiter.
- * On utilise un modèle du bruit de l'image, qu'il est en général difficile de justifier.
- * La quantité de calcul, surtout si elle dépend de la taille de l'image, peut être importante.

La géométrie digitale est un développement récent sur lequel on trouve de nombreuses publications dans des revues comme *Computer Graphics and Digital Image Processing*.

CHAPITRE 2

EXTRACTION DE LIGNES DE CONTRASTE

Le chapitre 2 décrit un travail relativement ancien, puisqu'il remonte aux débuts de nos activités en VO (cf. [Lux81a,81b]): l'extraction de lignes de contraste. Nous essayons d'être concis et constructif à propos de ce sujet qui a déjà reçu trop d'attention par rapport à son importance au sein d'un système de vision. La définition d'une ligne de contraste comme une crête dans l'image gradient amène à des algorithmes simples et efficaces, compte tenu du matériel dont nous disposons. Quelques photographies illustrent notre travail d'expérimentation, et nous établissons brièvement le rapport entre nos travaux et quelques résultats récents venant d'une approche mathématiquement plus fouillée.

2.1 L'IMAGE GRADIENT

2.1.1 Position du problème

Une "ligne de contraste" dans une image est une ligne - une séquence de pixels voisins - le long de laquelle l'intensité lumineuse change brusquement. La notion de contraste est donc étroitement liée à la notion mathématique de gradient d'une fonction à deux variables. Nous commençons ce paragraphe par une étude du gradient, puis nous définirons les points de contraste et les lignes de contraste à partir du gradient.

Les recherches sur l'extraction de ligne de contraste ont été très nombreuses, et un grand nombre de méthodes a été étudié que l'on peut répartir en trois classes:

- 1) Correspondance avec des masques (template matching), par exemple les masques de Kirsch.
- 2) Approximation du gradient au sens mathématique, par exemple l'opérateur de Sobel.

- 3) Correspondance avec un modèle paramétrique de lignes, par exemple l'opérateur de Hueckel.

En fait, les bases mathématiques sont les mêmes dans les trois cas; seules les présentations diffèrent. Dans les paragraphes suivants, nous présentons une approche de la classe 2.

2.1.2 Intérêt de l'image gradient

Pour une fonction réelle à deux variables $I(x,y)$, le gradient est la fonction vectorielle donnée par les deux dérivées partielles. Ces deux composantes sont dépendantes d'un système d'axes arbitraire, elles n'ont donc pas de signification physique particulière. Le gradient peut être caractérisé de façon indépendante du système d'axe par son module M et sa direction D . Le module M est défini par

$$M(x_0, y_0) = \sqrt{\frac{\partial I^2(x_0, y_0)}{\partial x} + \frac{\partial I^2(x_0, y_0)}{\partial y}}$$

et la direction D par

$$D(x_0, y_0) = \text{ATAN2}\left(\frac{\partial I(x_0, y_0)}{\partial x}, \frac{\partial I(x_0, y_0)}{\partial y}\right)$$

$\text{ATAN2}(x,y)$ étant $\text{ATAN}(x/y)$ comme en FORTRAN.

Le module mesure l'allure à laquelle varie la fonction I en un point - l'inclinaison de la pente; la direction D est celle de la plus forte variation - c'est la direction de la pente, perpendiculaire à la direction d'une éventuelle ligne de contraste.

On appelle "image gradient" d'une image d'intensités lumineuses la matrice de pixels dont les valeurs sont les gradients de l'image d'intensités. La figure II-1 montre une image d'intensités, les figures II-3-a et II-4-a des images gradient associées, dans lesquelles le module du gradient est affiché sous forme d'une intensité (noir = gradient fort). Les figures II-3-a et II-4-a sont obtenus avec des calculs légèrement différents (cf. ci-après).

Le gradient s'interprète alors comme suit:

- a) Gradient nul - zone homogène.
Exemple: Parties blanches du fond de la figure II-4-a.

- b) Gradient non-nul dans une région étendue.
- b-1) Valeurs faibles et assez continues - dégradé dans l'image originale, permettant d'obtenir des informations de surface 3-D par des méthodes numériques sophistiquées.
Exemple: Bord du briquet dans la figure II-3-a.
 - b-2) Valeurs chaotiques - région texturée.
Exemple: Fond de la figure II-3-a.
- c) Gradient non-nul dans une bande assez étroite - ligne de contraste. Les valeurs de direction de gradient sont alors cohérentes et perpendiculaires à celle de la ligne de contraste.
Exemple: Contours des objets dans les figures II-3-a et II-4-a.

L'image gradient fournit donc des informations précieuses, ce qui motive son étude.

2.1.3 Calcul de l'image gradient

Le calcul du gradient dans une image digitalisée est une opération délicate, et cela pour deux raisons. D'une part, l'image n'est pas une fonction réelle, connue en tout point du plan, mais une fonction échantillonnée. D'autre part, le gradient est une dérivée, et les opérations de dérivation augmentent les hautes fréquences, donc en particulier le bruit dans un signal. Le calcul du gradient d'une image doit donc répondre à deux problèmes spécifiques: en se plaçant dans la géométrie digitale, il doit adapter la définition donnée au paragraphe précédent au cas d'une fonction échantillonnée à valeurs entières, et introduire des éléments de lissage afin de réduire l'influence du bruit.

Principe du calcul du gradient

Nous adaptons la définition du gradient au cas d'une fonction discrète en remplaçant les dérivées partielles par des différences de la valeur de la fonction.

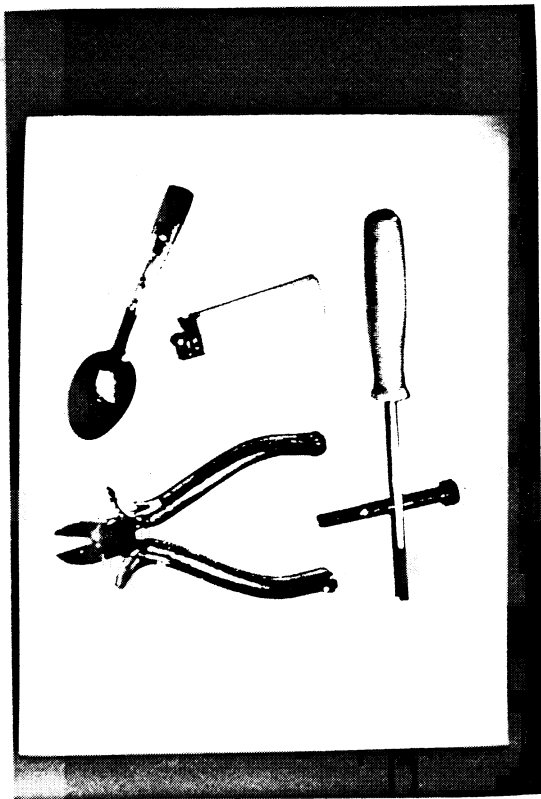


Figure II-1. Une image digitalisée (16 niveaux de gris) ...

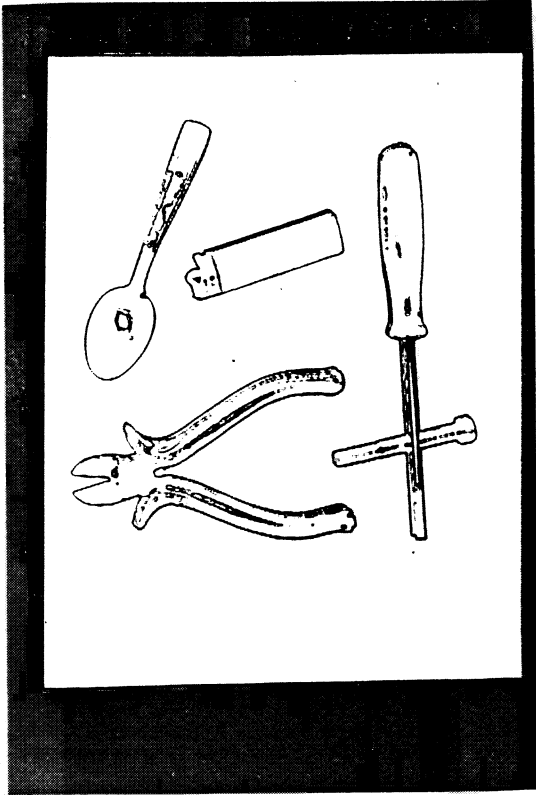
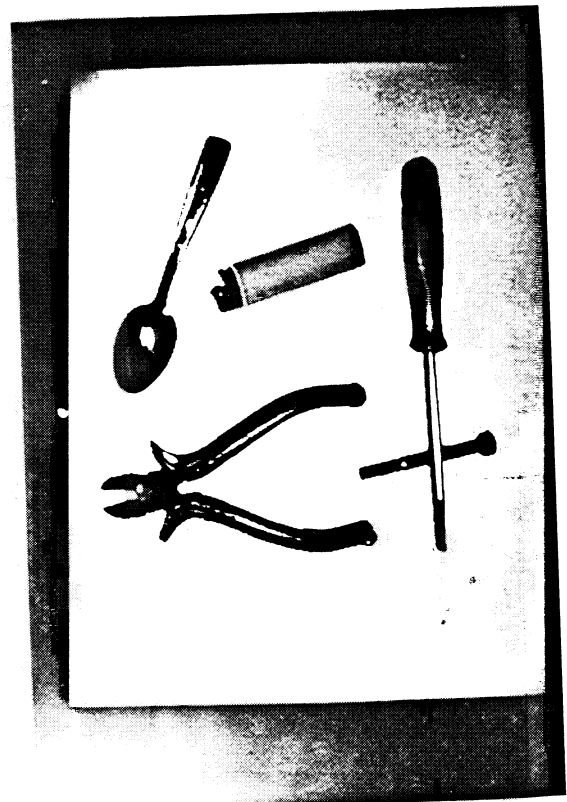


Figure II-3-a. Module d'un gradient 3x3.



II-2

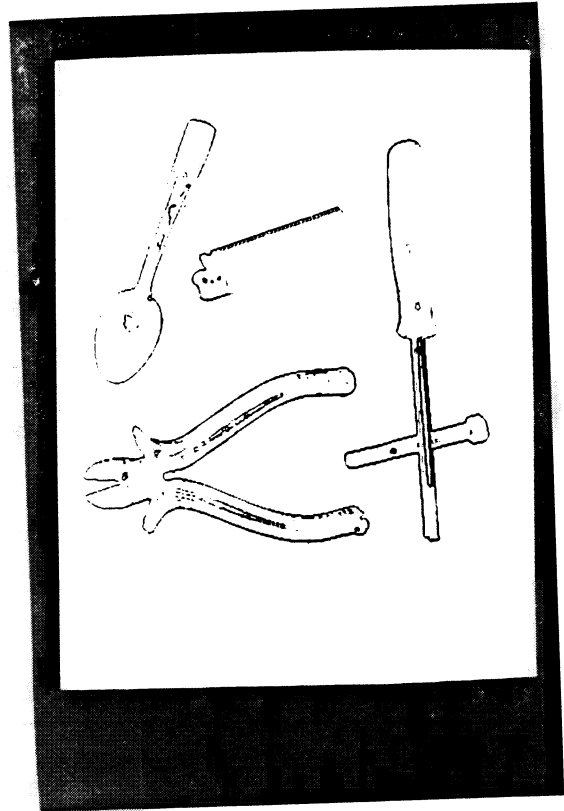


Figure II-3-b. Gradient 3x3 seuillé.

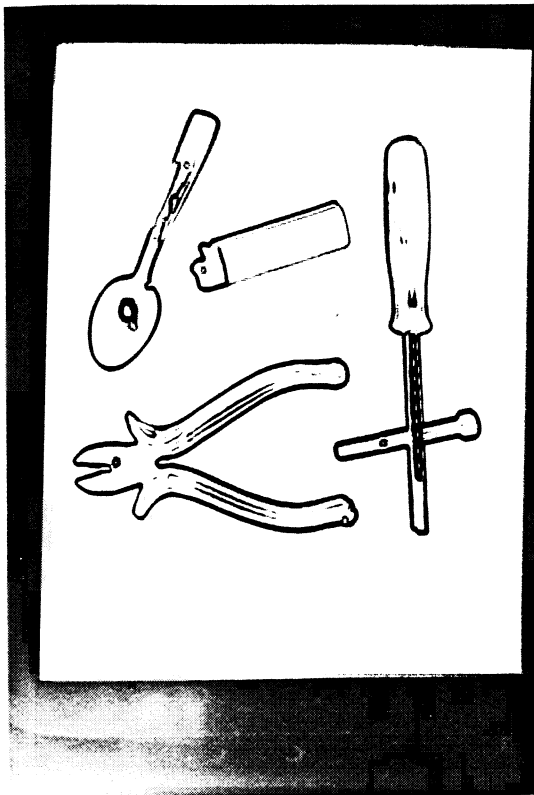


Figure II-4-a. Module d'un gradient 7x7.

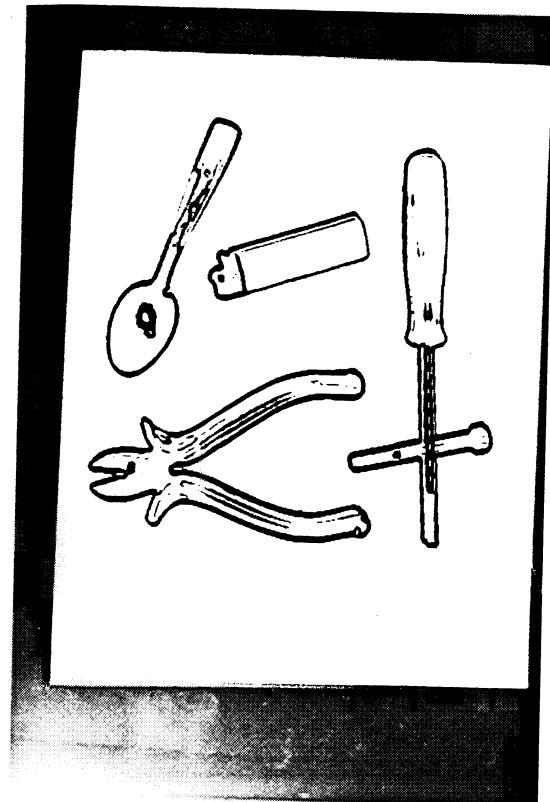


Figure II-4-b. Gradient 7x7 seuillé.

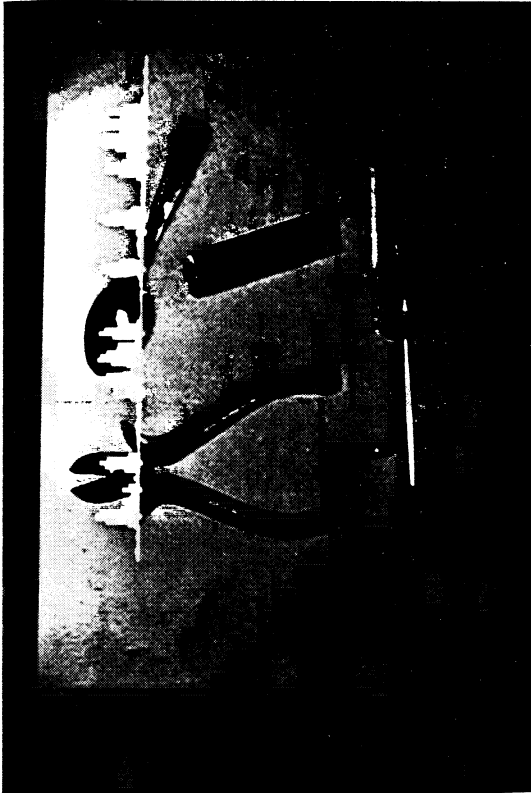


Figure II-5-a. Profil de contraste 5x5 sur un carré autour du noeud formé par l'intersection tournevis - vis.

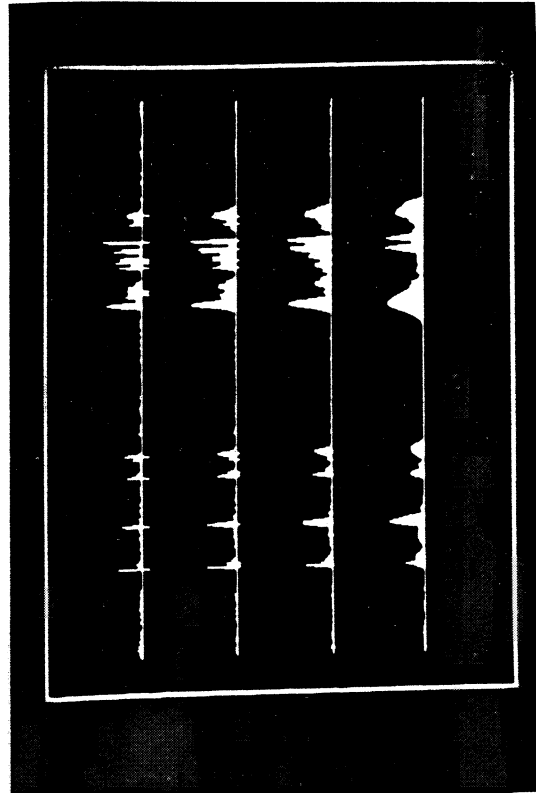


Figure II-5-b. Profils de contraste le long du vecteur (500,10)(10,500) - 2° diagonale - avec des masques 3x3, 5x5, 7x7, et 13x13.

Dans le cas le plus simple, on remplace

$$(a) \quad \frac{\partial I(x_0, y_0)}{\partial x} \quad \text{par} \quad (I(x_0+1, y_0) - I(x_0-1, y_0))/2$$

et

$$(b) \quad \frac{\partial I(x_0, y_0)}{\partial y} \quad \text{par} \quad (I(x_0, y_0+1) - I(x_0, y_0-1))/2$$

Ce calcul est représenté de façon intuitive par deux "masques de convolution"

-1	0	+1
----	---	----

pour (a)

et

+1
0
-1

pour (b)

Plus généralement, un masque est défini par un ensemble C de coefficients

$$C = \{c_{i,j}\} \quad i=i_0, i_k, \quad j=j_0, j_m$$

et le produit du masque avec l'image au point (x_0, y_0) par

$$P(I, x_0, y_0, C) = \sum (c_{i,j} * I(x_0+i, y_0+j)) / \sum |c_{i,j}|$$

Dans l'exemple (a) ci-dessus, $c_{-1,0} = -1, c_{1,0} = 1$.

Il est clair que le calcul d'une différence entre pixels, comme en (a) et (b), dégrade énormément le rapport signal/bruit. On est donc amené à calculer une différence de valeurs moyennes, par exemple avec les masques

0	1
-1	0

1	0
0	-1

Opérateur de Roberts en 2x2

0	0	1
0	0	0
-1	0	0

1	0	0
0	0	0
0	0	-1

Opérateur de Roberts en 3x3

0	1	1
-1	0	1
-1	-1	0

1	1	0
1	0	-1
0	-1	-1

Opérateur de Méré-Vassy en 3x3

-1	0	1
-1	0	1
-1	0	1

1	1	1
0	0	0
-1	-1	-1

Opérateur de Prewitt

2	1	0
1	0	-1
0	-1	-2

0	1	2
-1	0	1
-2	-1	0

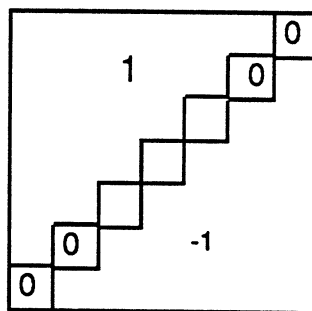
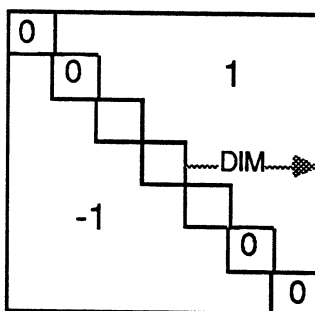
Opérateur de Sobel: NO-SE et NE-SO

1	0	-1
2	0	-2
1	0	-1

1	2	1
0	0	0
-1	-2	-1

Opérateur de Sobel: E-O et N-S

(a) Quelques masques usuels.



(b) Les masques de dérivation utilisés dans le système CAIMAN. La valeur de DIM peut être changée dynamiquement.

Figure II-6. Masques usuels pour le calcul des dérivés partielles en un point d'une image.

+1	+1	+1
0	0	0
-1	-1	-1

-1	0	+1
-1	0	+1
-1	0	+1

Un grand nombre de masques différents a été utilisé pour les calculs des dérivées d'une image par différents chercheurs, dont les plus importants sont rassemblés dans la figure II-6-a.

Nous avons choisi des masques diagonaux de type [Merø et Vassy 75] décrits par la figure II-6-b. Les masques diagonaux sont intéressants dans le cas de pixels de forme non-carrée (cas du standard TV), parce que, contrairement aux masques alignés avec les axes, ils calculent les deux dérivées partielles avec des masques parfaitement symétriques (figure II-6-b).

Généralement, le choix du bon masque fait intervenir plusieurs critères. En ce qui concerne l'extraction de lignes de contraste, l'expérimentation nous a amenés à considérer un seul paramètre comme réellement significatif: la taille des masques, c'est à dire l'importance du lissage avant le calcul de la différence. Plus le masque est grand, plus l'image gradient sera lissée - les détails fins disparaissent, le bruit diminue - et plus le calcul sera onéreux.

Le meilleur choix d'opérateur est aussi fonction de la qualité du matériel (résolution de la camera, optique). Nous avons expérimenté avec des masques de différentes tailles, et obtenu les meilleurs résultats avec des masques 7x7 et 5x5. La comparaison de la figure II-3-a, obtenue avec des masques 3x3, avec la figure II-4-a, obtenue avec des masques 7x7, illustre nettement ce compromis.

2.1.4 Interprétation de l'image gradient - profils

Contrairement aux apparences, l'image gradient ne fournit pas directement les lignes de contraste. La situation est illustrée par les figures II-3 et II-4. Comme nous l'avons mentionné, les lignes de contraste d'une image produisent des régions à fort gradient (figures II-3-a et II-3-b). Le seuillage du gradient ne fournit pas des lignes, mais des traits épais.

L'allure précise des traits dépend du seuil de contraste, et du calcul des dérivées: un seuil élevé donne des traits fins mais souvent interrompus (figure II-3-b), un seuil faible fournit des traits assez complets, mais aussi plus épais, et plus de lignes "parasites" apparaissent. Il y a une relation du même ordre avec la taille des masques du calcul de gradient: une faible taille - figure II-3 - donne des traits plus fins et moins continus, une taille plus importante - figure II-4 des traits plus épais et plus continus.

Cette situation est expliquée par l'analyse de profils de lignes. Le "profil d'intensités" d'une ligne est le tracé de l'intensité des pixels sur un chemin perpendiculaire à la direction de la ligne (donc en direction du gradient). Le "profil de contraste" est le tracé du module du gradient, ou d'une autre mesure du contraste, suivant ce même chemin.

La figure II-7 donne les profils théoriques pour deux cas importants.

* Dans le cas d'une ligne parfaitement nette, le profil d'intensité change brusquement. Il y a discontinuité, et la dérivée au sens continu ne serait pas définie. Le calcul de gradient discret fait apparaître une pyramide dont la largeur est la taille des masques utilisés. Le maximum de contraste se situe précisément sur la rupture d'intensité.

* Dans le cas d'une ligne floue, le profil d'intensité change progressivement sur une largeur de w pixels, la dérivée au sens continu a une forme rectangulaire. Suivant la taille t des masques, les profils de contraste sont sensiblement différents:

- 1) Si $t < w$, le profil présente un plateau de largeur $w-(t-1)$, situé dans la transition des intensités.
- 2) Si $t \geq w$, le profil de contraste a la même allure que pour une ligne nette, c'est à dire un maximum unique, situé au milieu de la transition.

Moyennant un bon choix du paramètre t il est donc parfaitement possible de localiser des lignes floues.

La forme pyramidale des profils de contraste explique les résultats du seuillage de gradient: le seuillage laisse persister une partie du haut de la pyramide, et l'épaisseur des traits variera avec la hauteur absolue des pyramides, variable suivant les endroits de l'image.

La figure II-8 présente des profils analogues pour le cas d'un trait fin dans l'image d'intensités, caractérisé par deux changements d'intensités inverses rapprochés (image d'un fil, d'une écriture, etc.).

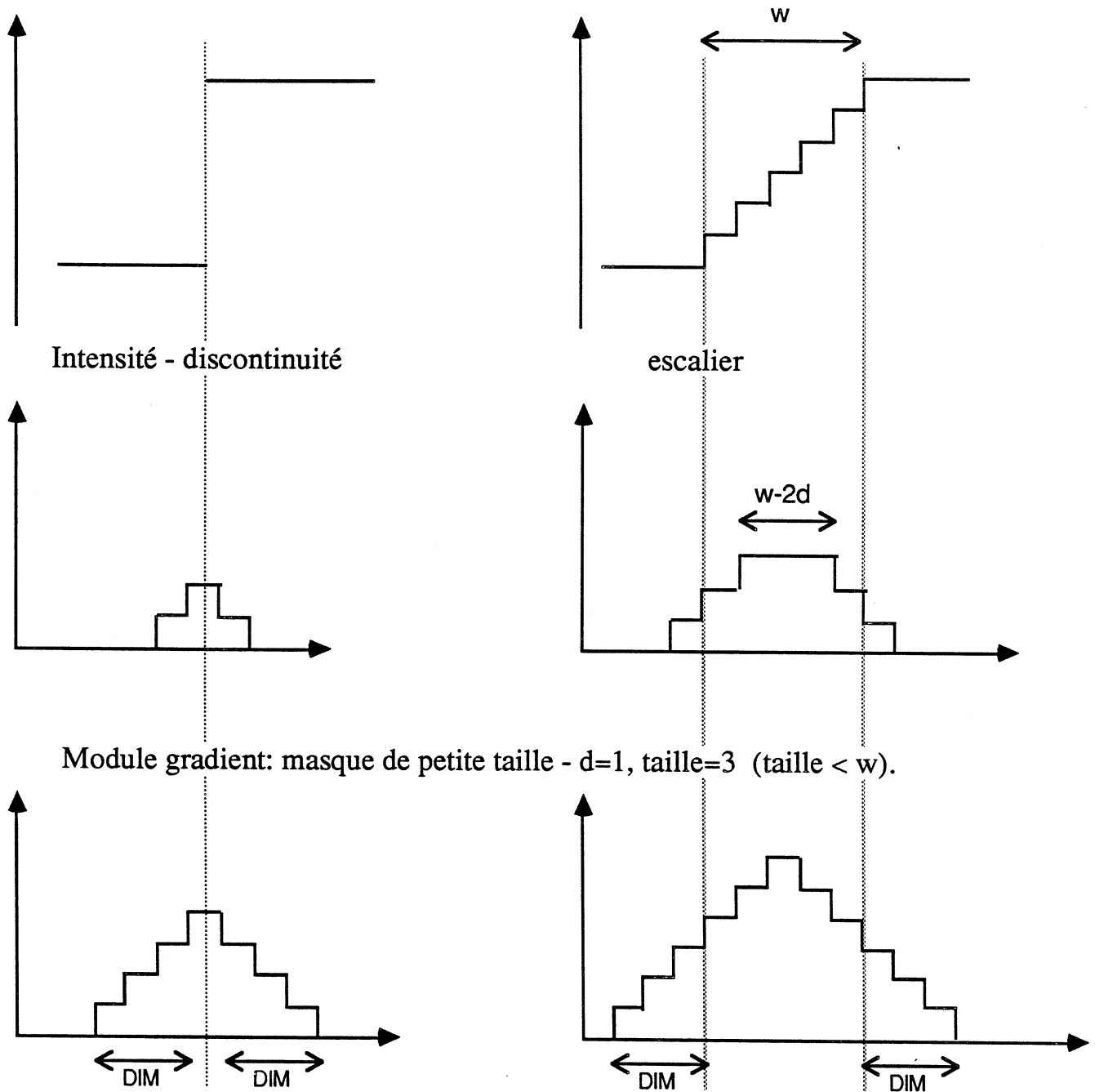


Figure II-7. Profils d'intensité et profils de gradient d'une ligne de contraste.

(a) Le cas d'une ligne nette (colonne de gauche).

(b) Ligne floue, transition de largeur $w=3$ (colonne de droite).

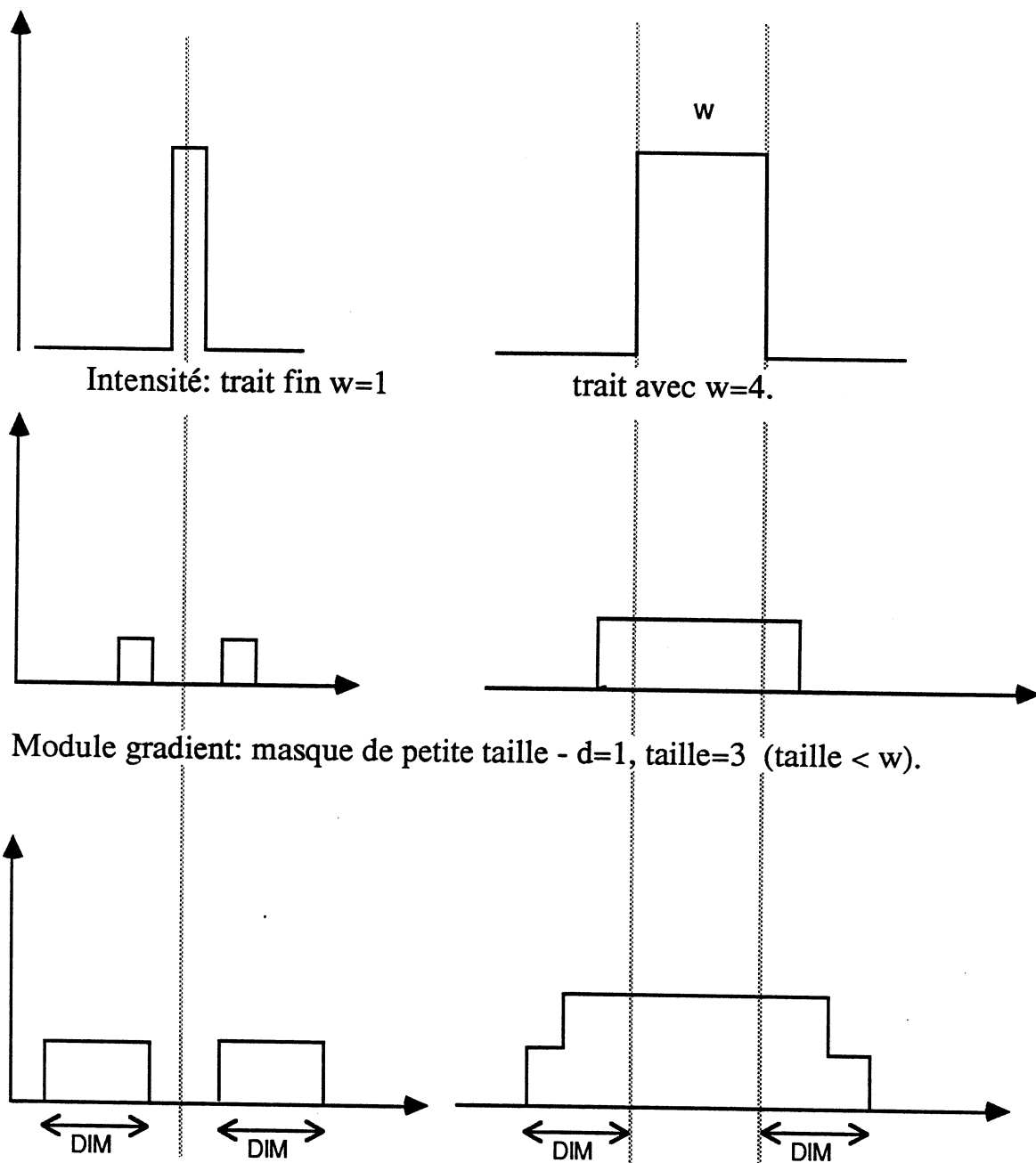


Figure II-8. Profils d'intensité et profils de gradient d'un trait.

(a) Trait très fin, largeur $w=1$ pixel (colonne de gauche).

(b) Trait de largeur 4 (colonne de droite).

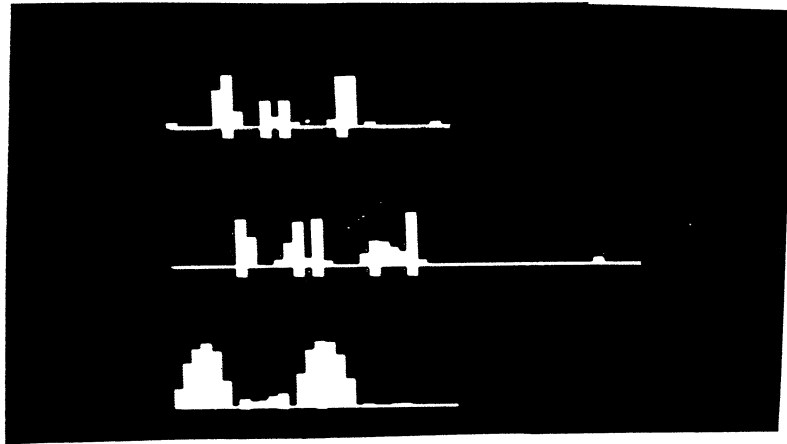


Figure II-9. Profils de lignes de contraste dans l'image II-1.
De haut en bas: masques 3x3,5x5,7x7.
Les pixels maxima de contraste sont également affichés
(traversée du manche droit de la pince).

La figure II-9 montre des profils de contraste "réels", obtenus à partir de l'image II-1. Ces profils, qui font apparaître plusieurs lignes, sont bien en accord avec les profils théoriques des figures II-7 et 8.

2.1.5 Relation avec d'autres travaux

Travaux de Marr et Hildreth

Selon la théorie de D. Marr, le croquis élémentaire met en évidence les changements d'intensité significatifs dans une image. Ces changements peuvent se produire à différentes échelles, c'est à dire les lignes peuvent être plus ou moins flous. Marr propose alors d'appliquer plusieurs opérateurs de détection de contours dont les tailles correspondent à des fréquences spatiales différentes. Cette proposition a été approfondie par [Hildreth 80].

Par définition, sur une ligne de contraste le gradient de l'image est maximal, ou, ce qui est équivalent, la dérivée seconde s'annule. L'opérateur proposé est composé de deux filtres successifs:

- * Un premier filtre Gaussien assure un lissage. La taille de ce filtre détermine l'échelle à laquelle travaille

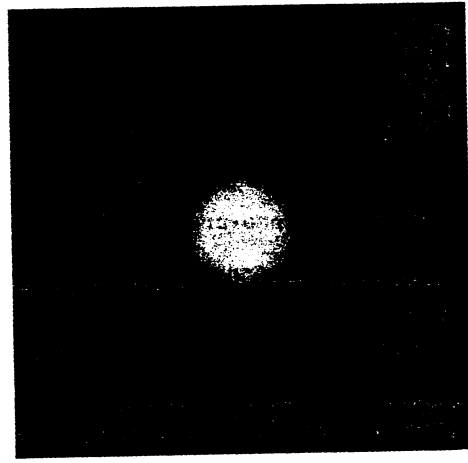
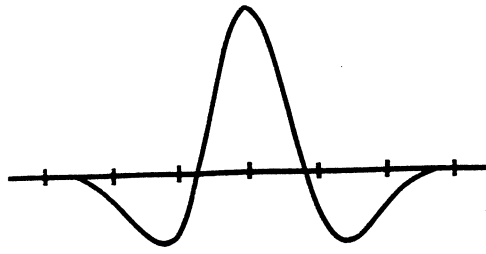


Figure II-10. Le "chapeau mexicain" à une et à deux dimensions.

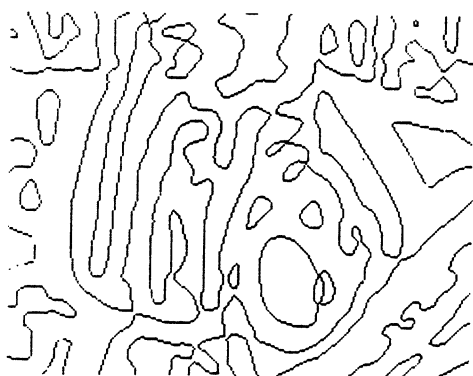
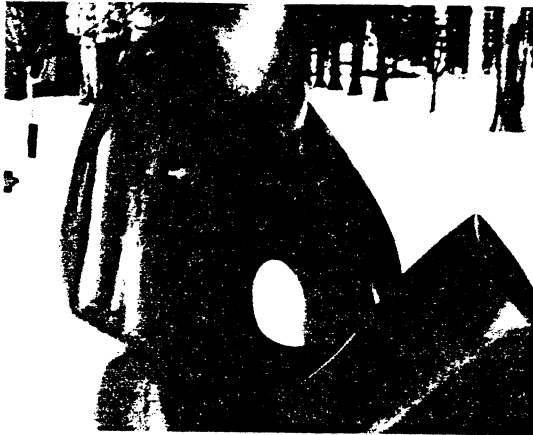


Figure II-11. Passages par zéro de l'opérateur $\nabla^2 G$ avec différentes tailles de w ($w=6, 12, 24$).

l'opérateur.

- * A l'image lissée on applique un filtre Laplacien. Les points de contraste sont alors caractérisés par une valeur nulle de ce Laplacien. L'intérêt du Laplacien vient du fait que c'est l'unique opérateur de dérivée seconde linéaire et indépendant de la direction des lignes de contraste à détecter.

La combinaison de ces deux filtres donne un opérateur sous forme d'un "chapeau mexicain" (figure II-10) dont la taille est caractérisée par le diamètre w de la partie positive, lié à la constante σ du Gaussien par la relation

$$\sigma = w/2*\sqrt{2}$$

La définition analytique de cet opérateur est donnée par la formule

$$\nabla^2 G = (2-r^2/\sigma^2) * \exp(-r^2/(2*\sigma^2))$$

Les résultats de cet opérateur sont illustrés dans la figure II-11. Cette figure montre:

- * une image digitalisée
- * les passages par zéro de l'opérateur pour différentes valeurs de w .

La définition de cet opérateur est le résultat d'une longue recherche, à la fois théorique et expérimentale. Il possède trois propriétés importantes:

- * Il est localisé dans l'espace.
- * Il est localisé en fréquence, c'est à dire il a une bande passante limitée.
- * Il est indépendant de l'orientation des lignes de contraste.

L'importance de ces propriétés devient évidente lorsqu'on expérimente des opérateurs semblables qui ne les possèdent pas. Ces expériences sont illustrées dans la figure II-12.

- * La localisation spatiale est violée par un filtre passe-bande idéal (fonction SINC). Les lobes latéraux de ce filtre provoquent alors une forme de "écho" - le long d'une ligne de contraste apparaissent des lignes parallèles qui ne correspondent à aucune réalité dans l'image.
- * L'utilisation d'une approximation "carrée" de la dérivée seconde produit un opérateur non localisé en fréquence. On constate que, à taille égale, la résolution d'un tel opérateur est plus faible que celle du chapeau, c.a.d. il détecte moins de lignes. Pour les signaux à une dimension, il a été montré [Logan 77] qu'une augmentation de la bande

passante, à fréquence centrale constante, réduit le nombre de passage par zéro.

- * Le problème spécifique d'un opérateur spécialisé dans la détection de lignes d'une direction donnée vient de sa "réponse à distance". Lorsqu'une ligne s'arrête, un tel opérateur donne une réponse positive plusieurs pixels au-delà de la fin de la ligne. Il a donc tendance à voir une ligne plus longue qu'elle ne l'est.

Travail de Canny

[Canny 84] pose le problème d'extraction de points de contraste en termes mathématiques, et le résoud par des méthodes de calcul variationnel. La figure II-13 illustre cette démarche sur l'exemple d'un signal unidimensionnel. La ligne de contraste est modélisée par une transition de type "marche" (step edge) dégradée par un bruit blanc Gaussien. On cherche un opérateur de convolution dont le résultat accuse un maximum à l'emplacement du saut d'intensité du signal. On définit trois critères de qualité pour un tel opérateur.

- * Bonne détection.
L'opérateur doit maximiser la probabilité de détecter les vrais points de contraste et minimiser la probabilité d'une fausse réponse, ce qui revient à maximiser le rapport signal sur bruit en sortie.
- * Localisation.
L'opérateur doit accuser le maximum le plus proche possible du point de contraste.

Notons que, contrairement à la situation rencontrée avec une image réelle, les notions de "vrai point de contraste" et "vrai emplacement" ont un sens, puisque la donnée est définie explicitement par un "vrai" signal et du bruit.

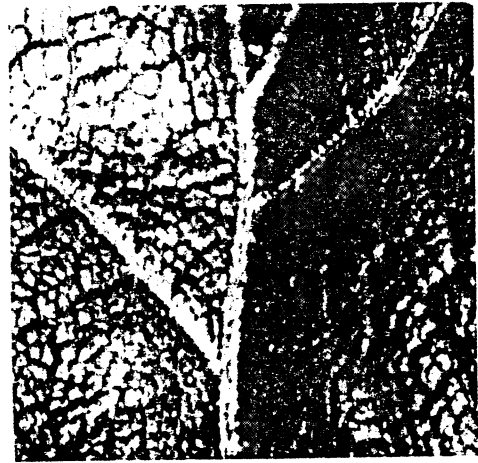
Ces deux critères sont mis en équation. On constate qu'ils ne peuvent être optimisés indépendamment; l'amélioration de l'un entraîne toujours une dégradation de l'autre, le produit étant constant pour une classe d'opérateurs donnée.

L'opérateur optimal pour ces critères (figure II-13-b) est relativement complexe, mais en fait très proche de la dérivée d'un opérateur Gaussien. Dans une partie plus pratique, Canny développe en détail des techniques efficaces pour l'implantation de convolutions avec de tels opérateurs.

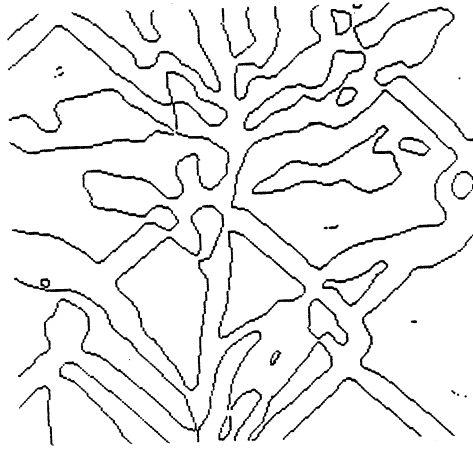
La figure II-14 compare l'opérateur de Canny avec celui de Marr et Hildreth et avec un simple opérateur rectangulaire sur l'exemple d'une ligne de contraste synthétique fortement bruitée (rapport signal/bruit 0.2, image 256 par 256). Cet



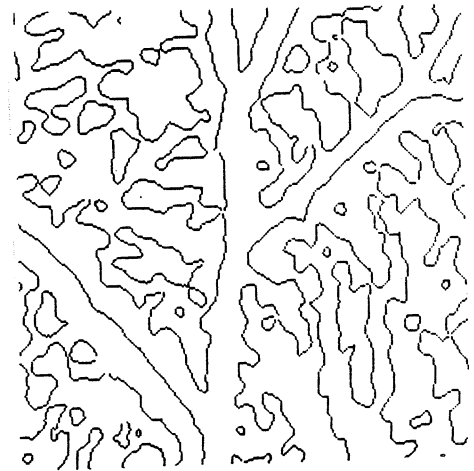
a) image



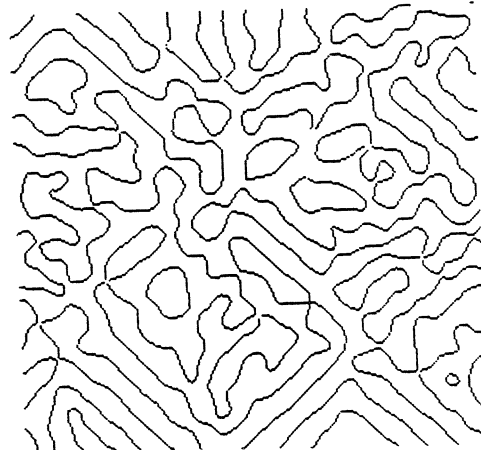
d) image



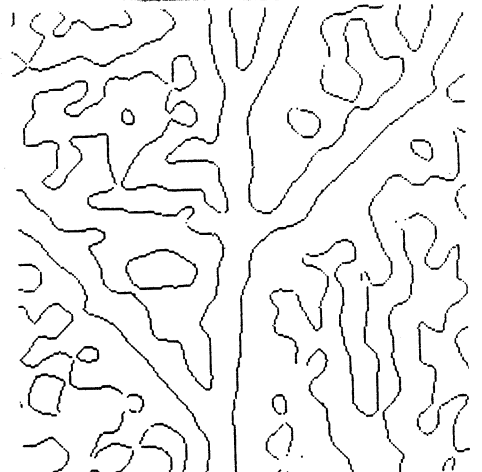
b) passages par zéro de $\nabla^2 G$



e) passages par zéro de $\nabla^2 G$



c) passages par zéro d'un filtre passe-bande idéal (noter l'écho).



f) passages par zéro d'un opérateur carré

Figure II-12. Comparaison de l'opérateur de Marr-Hildreth avec d'autres opérateurs [Hildreth 80].

opérateur donne également des résultats excellents avec des images réelles.

Conclusion

La comparaison de la figure II-13 avec la figure II-7 montre clairement la différence entre les hypothèses de travail de Canny et les nôtres: nous supposons en effet que le bruit d'échantillonnage est prédominant, alors que Canny considère un signal dégradé par un fort bruit Gaussien. Les applications que nous avons traitées, et aussi l'exemple de la figure II-9, montrent que notre hypothèse de travail est vérifiée dans un grand nombre de cas pratiques, ce qui justifie notre choix d'un opérateur simple et peu coûteux en calculs. Nous concluons de la discussion de Marr et Hildreth (cf. figure II-12-c) que notre opérateur donnera des résultats sous-optimaux pour des réseaux de lignes très denses; dans ce cas il nous reste cependant la possibilité de réduire la taille de la fenêtre utilisée. Une évaluation quantitative des opérateurs gradient simples est donnée dans [Crowley et Parker 80].

2.2 SUIVI DE LIGNES DE CONTRASTE

L'analyse du paragraphe précédent nous amène aux définitions suivantes:

DEFINITION: Un **point de contraste** est un maximum strict dans un profil de contraste.

DEFINITION: Une **ligne de contraste** est une séquence de points de contraste $(P_0 \dots P_n)$, chaque point P_i étant voisin (au sens du 8-voisinage) du point précédent P_{i-1} ($i=1, \dots, n$).

Avec l'hypothèse (raisonnable) que le gradient d'une image est à peu près continu, cela définit effectivement les crêtes du gradient.

La recherche d'une ligne de contraste peut alors s'effectuer de la façon suivante:

- 1) Recherche d'un point initial P_0 .
A cet effet on peut analyser le gradient sur un chemin approprié de l'image. Les figures II-5-a et II-5-b illustrent cette étape. La figure II-5-a montre, dans l'image de la figure II-1, le profil de contraste autour d'un "noeud"; la figure II-5-b montre les différents

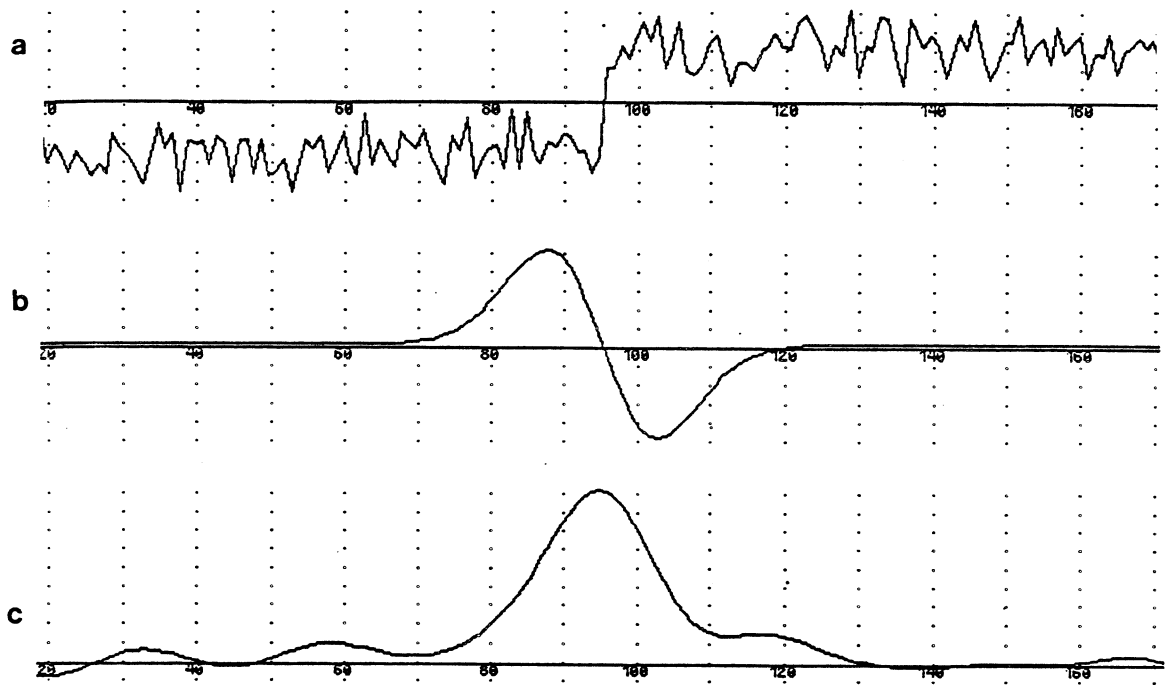


Figure II-13. L'opérateur de [Canny 83].

- a) un contraste bruité
- b) l'opérateur
- c) a) convolué avec b)

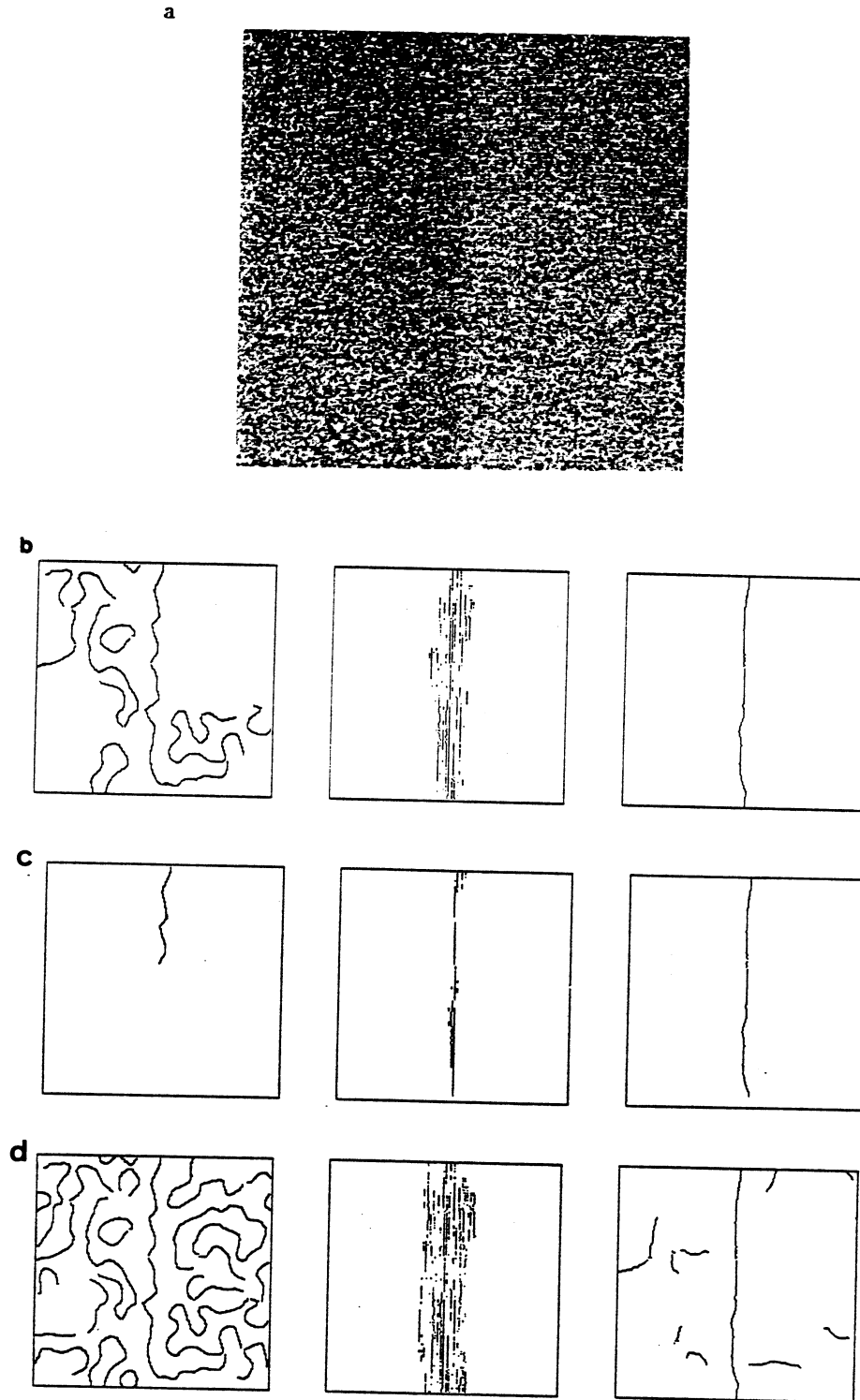


Figure II-14. Performances comparées pour un contraste dégradé par un bruit Gaussien [Canny 83].

a) ligne de contraste

b) lignes trouvées par trois opérateurs:
Laplacien d'un Gaussien (Hildreth, à gauche)
opérateur carré (au milieu)
dérivé d'un Gaussien (Canny, à droite)

c) lignes trouvées en augmentant le seuil des opérateurs de 50%

d) lignes trouvées en réduisant le seuil des opérateurs de 50%

profils obtenus le long du vecteur (500,10)(10,500) avec des masques 3x3, 5x5, 7x7, et 13x13 (noter le comportement du bruit).

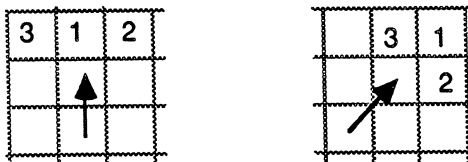
Chaque maximum bien formé dans ce profil est un point de contraste, donc un point initial potentiel.

2) Suivi.

A partir d'un point courant P_i , chercher un nouveau point P_{i+1} . Ce nouveau point doit être

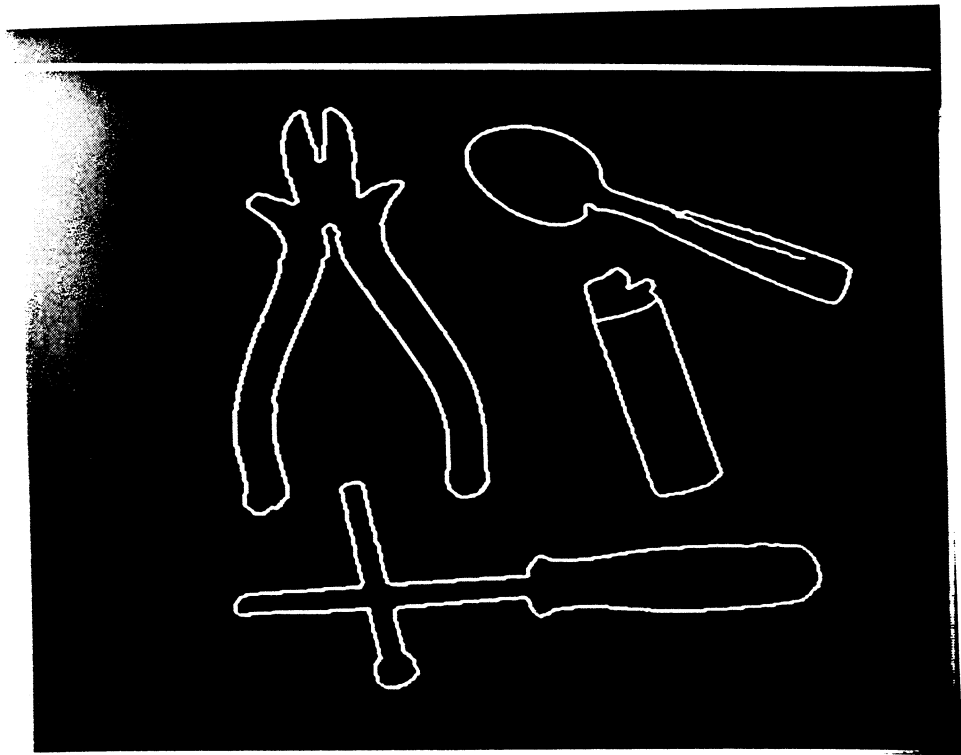
- point de contraste
- voisin de P_i
- différent de P_{i-1} ($i \geq 1$)

L'algorithme effectivement implémenté prend quelques libertés avec cette définition (c'est en cela qu'il diffère de l'algorithme de [Nevatia et Babu 80]): en raison de la continuité du gradient, il n'analyse pas tous les voisins de P_i , mais seulement les quelques points dans la direction de la ligne, esquissés dans le schéma suivant



pour en choisir celui au gradient le plus fort. Nous avons ici *in ovo* le principe du contrôle par prédiction - vérification, grâce auquel trois calculs de gradient sont suffisants, au lieu de sept pour l'ensemble des voisins (excepté P_{i-1}). Les figures II-2 et II-15 montrent quelques lignes de contraste ainsi construites à partir de différents points initiaux.

Notons que certains points d'une telle ligne peuvent ne pas être des points de contraste selon la définition ci-dessus; cela arrive notamment dans les angles et sur les plateaux du gradient. L'acceptation de ces points s'avère comme un avantage de l'algorithme, parce qu'elle permet de constituer des lignes plus longues. Nous nous sommes rendu compte de cet avantage dans la comparaison avec les résultats d'une application stricte des définitions (comme c'est fait par la première version du GTR) - en regardant de près, on constate de nombreuses interruptions et quelques bifurcations de 1 pixel dans les lignes, ce qui complique l'interprétation ultérieure.



CHAPITRE 3

ANALYSE DE LIGNES

Le chapitre 3 étudie le découpage d'une ligne digitale en entités géométriques simples. C'est un problème de géométrie digitale, pour lequel notre objectif principal est la recherche d'algorithmes réalistes plutôt que l'étude dans l'abstrait des problèmes de géométrie. Nous résumons brièvement notre expérience avec la courbure discrète permettant, en particulier, de trouver des angles et des arcs de cercles (§3.2). L'essentiel du chapitre est ensuite consacré à l'étude détaillée des problèmes liés au type d'indice visuel le plus simple, le segment de droite. De nombreux algorithmes avec divers variantes sont présentés, leurs propriétés géométriques et algorithmiques sont analysées. Ce paragraphe n'est pas seulement un "survey" mais contient aussi de nombreux éléments originaux. En particulier, nous caractérisons par plusieurs propriétés l'algorithme bien connu de découpage récursif (§3.3.4), et nous présentons un algorithme nouveau, le découpage itératif (§3.3.6). Le §3.3.8 analyse l'intérêt de ces algorithmes pour trois problèmes d'application, et nous terminons par quelques remarques bibliographiques.

3.1 Position du problème

Une ligne de contraste dans une image n'est pas un élément de description suffisamment abstrait en vue d'une interprétation:

- * Elle peut traverser l' image en sautant d'un objet à un autre, passer par des ombres et des reflêts. Son interprétation en termes d'éléments de la scène peut donc varier le long de la ligne.
- * Etant une séquence de points à coordonnées image, une ligne de contraste est représentée sous une forme dépendant du pas d'échantillonnage, donc aussi du bruit du capteur.

Une ligne de contraste n'est donc pas un bon IV (cf. aussi les critères énoncés aux paragraphes III-1.2.1 et III-1.6). Pour cette raison nous considérons le problème de l'analyse de lignes, qui doit extraire des caractéristiques n'ayant pas ces défauts, comme des segments de droites, des arcs de cercles, des points d'inflexion, des angles, etc.

Dans le cas où le premier défaut ci-dessus peut être éliminé par les conditions de prise de vue (objets isolés, bon éclairage) on peut envisager une comparaison directe entre une ligne-image et un modèle. Cette approche a été utilisée par [Dessimoz 80], et nous l'avons proposée pour le positionnement de pièces de carrosserie [Lux 80]. Elle se heurte cependant fondamentalement au deuxième problème - le bruit, en particulier le bruit d'échantillonnage. La présence de bruit dans une ligne de contraste a une conséquence simple et surprenante: comme la somme des bruits peut ne pas s'annuler dans le calcul de longueur d'une ligne (la figure II-16 illustre le cas d'un segment de droite), toute propriété d'une ligne liée à un calcul de longueur est faussée par le bruit. Or, une mise en correspondance globale entre deux lignes (par exemple une corrélation) dépend crucialement de l'égalité des longueurs.

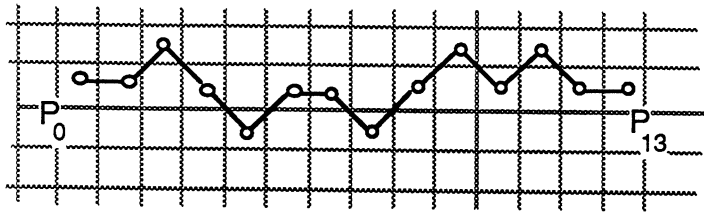
A notre avis, le calcul de longueur d'une ligne digitale suppose au préalable le découpage de la ligne en segments analytiques, tel que nous le présenterons dans les paragraphes suivants.

3.2 La courbure discrète

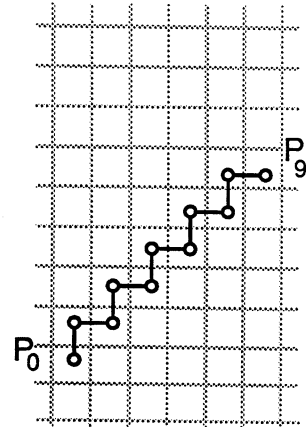
La courbure permet de caractériser très facilement un certain nombre d'IV qui nous intéressent dans une ligne, en particulier

- * Segment de droite - section à courbure nulle
- * Arc de cercle - section à courbure constante et non-nulle
- * Point d'inflexion - changement de signe
- * Angle - point à courbure infinie

On peut légalement déduire de ces exemples que le calcul de courbure à partir d'une ligne de contraste doit être une opération délicate. En effet, la courbure étant une dérivée seconde, elle a tendance à augmenter l'influence du bruit. Nous nous trouvons devant un problème analogue à celui du calcul du gradient, et notre démarche est tout à fait la même: nous appliquons un opérateur différence combiné avec un lissage. Nous appelons le résultat "courbure discrète", et non simplement courbure, parce qu'il possède quelques propriétés particulières, que ne possède pas la courbure classique.



(a) $\overline{P_0 P_{13}} = 13$ si on considère qu'il s'agit d'un segment de droite, $3+10\sqrt{2}$ si on tient compte du "bruit".



(b) $\overline{P_0 P_9} = 5\sqrt{2} \approx 7$ ou 10 suivant le même principe.

Figure II-16. Le bruit peut fausser la longueur d'une ligne digitale.

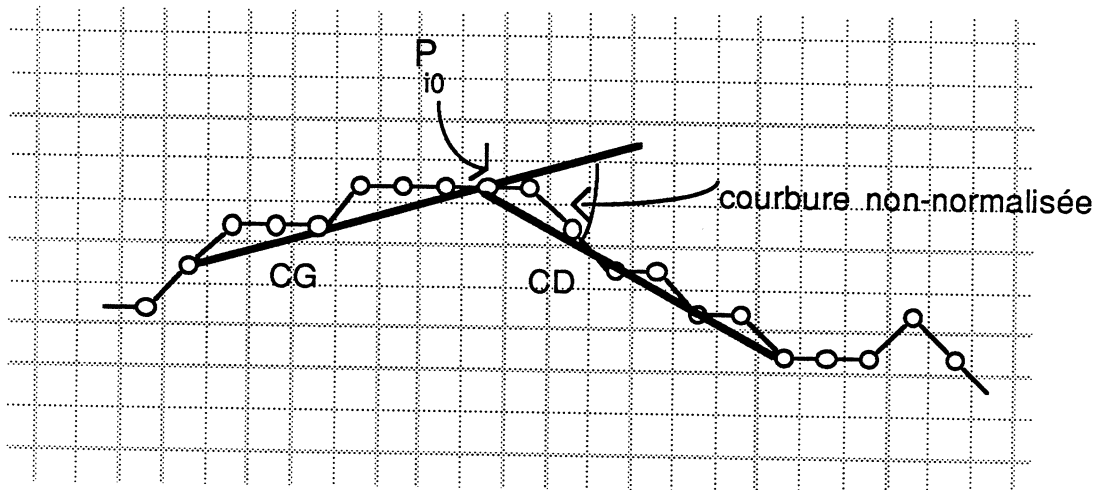


Figure II-17. Courbure discrète avec $k=7$.

DEFINITIONS

Les définitions suivantes sont illustrées par la figure II-17.

Soit une ligne de contraste $(P_0, P_2, \dots P_i \dots)$.

La **corde gauche** CG de longueur k au point i_0 , avec $i_0 \geq k$, est le segment de droite défini par les points P_{i_0-k} et P_{i_0} .

La **corde droite** CD de longueur k au point i_0 , avec $i_0 \geq 0$, est le segment de droite défini par les points P_{i_0} et P_{i_0+k} .

La **courbure discrète non-normalisée** au point P_{i_0} , avec $i_0 \geq k$, est définie par

$$(\text{angle CD}) - (\text{angle CG})$$

La **courbure discrète** au point P_{i_0} est définie par

$$(\text{courbure non-normalisée}) / (\text{longueur CG} + \text{longueur CD})$$

La figure 18 indique le comportement de la courbure discrète non-normalisée dans deux cas simples et importants.

* Segment de droite.

Le long d'un segment de droite, la courbure non-normalisée n'est pas en général nulle partout, mais comporte des bosses positives et négatives, disposées de façon régulière. L'espacement et la hauteur des bosses dépendent de l'orientation de la ligne. Dans le cas le plus défavorable, la hauteur des bosses est de

$$1 / k$$

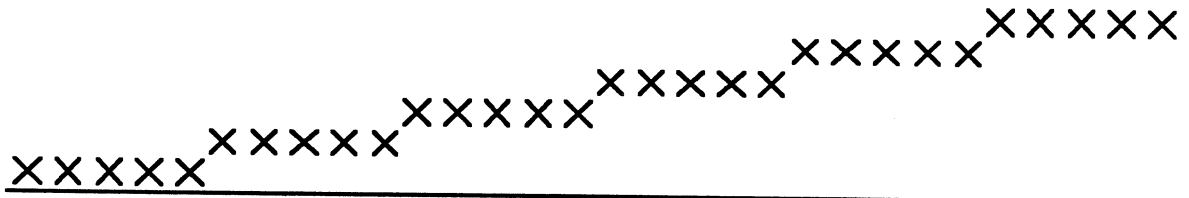
Dans une première approximation, nous pouvons donc considérer toute valeur de courbure non - normalisée inférieure à $1/k$ (en valeur absolue) comme "bruit autour de zéro".

* Angle.

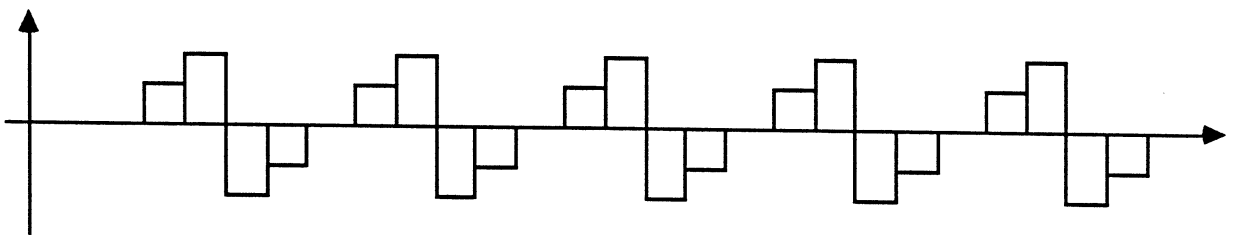
Un angle entre deux segments de droites donne lieu à une pyramide de largeur $2k-1$, la hauteur étant définie par la valeur de l'angle. Une comparaison avec les profils de gradient montre que le même algorithme pourra servir à la localisation des points de contraste dans un profil de gradient et à la recherche d'angles dans la courbure.

Nous avons obtenu d'excellents résultats pour la recherche d'angles dans la courbure non-normalisée (la normalisation de la courbure fausse les profils des angles). Pour l'extraction de segments de droites la courbure est une approche possible qui n'est pas la meilleure (cf. paragraphe suivant); pour l'extraction d'arcs de cercles la courbure discrète donne des résultats intéressants, mais en raison d'une forte influence

du bruit de discrétisation il est difficile d'obtenir une bonne précision.



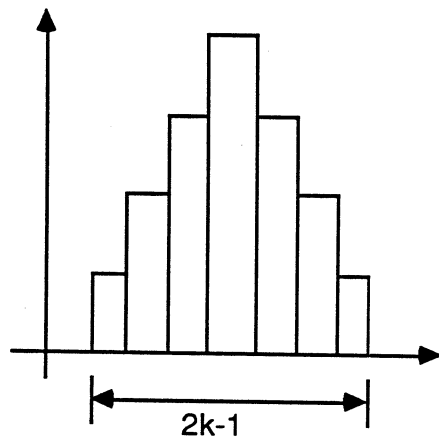
(a) Segment de droite digitale



(b) Courbure discrète (non à l'échelle), $k=2$.



(c) Un angle parfait.



(d) Courbure discrète de l'angle.

Figure II-18. Comportement de la courbure discrète

3.3 Recherche de segments de droites

Symboles utilisés dans le paragraphe 3.3

$[P_0 \dots P_i \dots P_n]$ ligne de contraste à analyser

D segment de droite proposé par un algorithme

$[P_{i_0} \dots P_{i_k}]$ partie de la ligne approché par D

$\text{DIST}(P_i, D)$ distance du point P_i à la droite D

EPSD epsilon largeur

MINLO longueur minimale d'une droite

$\overline{P_i P_j}$ segment de droite reliant les points P_i et P_j

$X_0 \dots X_k$ train polygonal de k segments défini par les
points extrémités

L_i longueur du i° segment

A_i angle au point X_i

critère MC: La droite D approche $[P_{i_0} \dots P_{i_k}]$ au sens des
moindres carrés

critère P ("proximité"): $\text{DIST}(P_i, D) < \text{EPSD}$ pour $i=i_0, \dots, k$

critère M ("maximalité"):

3.3.1 Enoncé et approche naive

Nous considérons le problème suivant.

Etant donné une ligne de contraste $[P_0 \dots P_n]$, déterminer les segments de droites présents dans cette ligne.

C'est un problème de géométrie digitale dont les difficultés apparaissent si on l'approche naïvement. Il admet une solution exacte seulement dans le cas d'un alignement parfait, par exemple

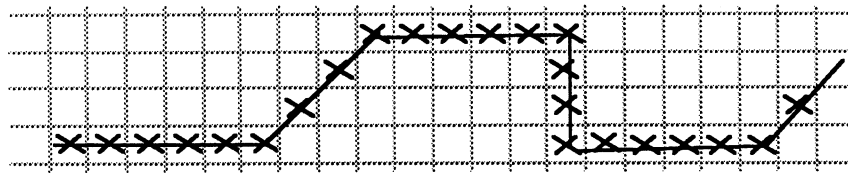


figure II-19

Ou il admet une solution triviale qui consiste à définir un segment de droite chaque fois qu'il y a un écart:

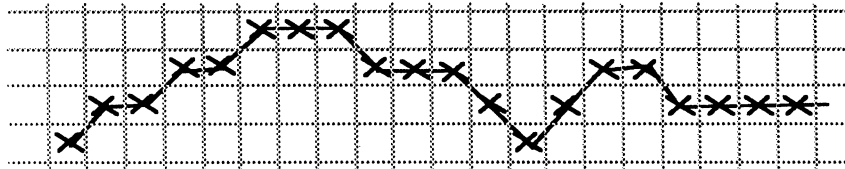


figure II-20

Le premier cas restera toujours exceptionnel, la deuxième solution n'est pas satisfaisante. L'énoncé initial n'est donc pas précis, mais à prendre dans un sens encore à définir. Comme pour la notion de gradient, nous devons adapter la définition de segment de droite, puis chercher des algorithmes pour les détecter.

Une idée de ce genre vient rapidement à l'esprit: une droite approchant les points d'une ligne, en particulier l'approximation au sens des moindres carrés. C'est effectivement une idée intéressante, mais insuffisante. En effet, tout nuage de points admet une droite l'approchant au

sens des moindres carrés, par exemple

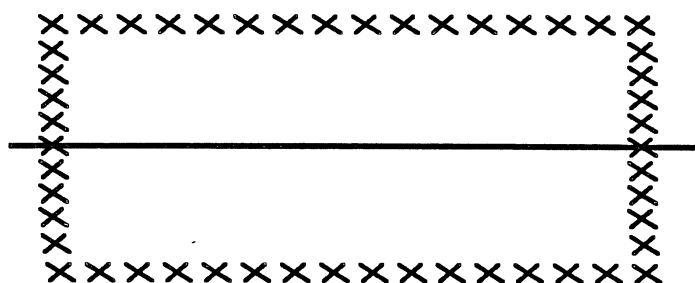


figure II-21

Il convient de diviser le problème en deux sous-problèmes:

- A) Déterminer les "points de cassure", les extrémités des segments.
- B) Déterminer les équations des droites.

Le sous-problème B est effectivement résolu par une méthode aux moindres carrés. Par contre, le sous-problème A est plus délicat, d'autant plus qu'il est en général souhaitable d'éliminer les parties d'une ligne qui ne sont pas droites:

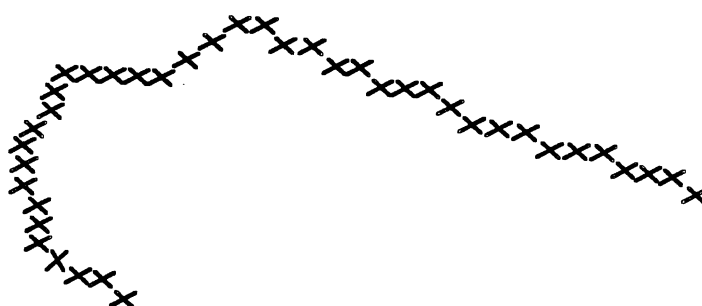


figure II-22

Le problème de recherche de segments de droites dans une séquence de points est non seulement un problème de base important pour l'interprétation d'images - c'est à la fois l'IV le plus simple et le plus utilisé - mais il est aussi un problème algorithmique exemplaire, comparable au problème de tri. Comme pour le problème de tri, il existe de nombreux algorithmes résolvant ce découpage en utilisant des principes

fondamentalement différents. Par une coïncidence remarquable, un de ces algorithmes possède même la structure d'un algorithme de tri bien connu, le tri par segmentation. Cependant, il s'agit seulement d'une coïncidence, il n'y a aucun lien profond entre les deux problèmes, étant donné que le problème de recherche de segments de droites admet des solutions d'ordre n , alors que la complexité du problème de tri est intrinsèquement d'ordre $n \cdot \log(n)$.

Les paragraphes suivants exposent différentes approches au problème du découpage; ces approches sont comparées et discutées ensuite, puis nous donnerons quelques éléments de généralisation à d'autres types de segments de courbes.

3.3.2 L'algorithme de Roberts

Le problème de segmentation d'une ligne s'est posé dès le début de la VO. Aussi commençons-nous par une brève présentation d'un algorithme simple décrit par Roberts dont l'analyse donnera un premier éclairage des difficultés.

Le principe de l'algorithme est le suivant: pour chaque point P_i on calcule l'écart de la meilleure droite approchant l'ensemble des points depuis le dernier point de cassure. Si l'écart dépasse un certain epsilon, on crée un nouveau point de cassure.

```

I0 := 0;    {I0 indice du dernier point de cassure}
E := 0.0;
pour J:=0 à N faire
  debut
    D := droite approchant au sens des MC les points
        (Pi0 ... Pj)
    E := E + DIST(P(J),D)**2;
    si E > ε alors
      debut
        {création résultat: (Pi0 ... Pj-1)
         approché par D}
        I0 := J;
        E := 0.0
      fin
  fin
fin

```

Les segments de droites produits par cet algorithme ont la propriété d'approcher de façon optimale un segment de points, et d'être de longueur maximale garantissant une certaine qualité de l'approximation. Chaque segment satisfait deux critères importants et utiles par la suite.

- (1) Le **critère MC**, le critère d'approximation au sens des moindres carrés.
Chaque segment D approche au sens des moindres carrés "son" morceau de la ligne en minimisant

$$\sum_{i=i_0, \dots, i_k} \text{DIST}^2(P_i, D)$$

- (2) Le **critère MD**, le critère de maximalité à droite.
Nous avons en effet

$$\sum_{i=i_0}^{i_k-1} \text{DIST}^2(P_i, D) < \epsilon$$

et

$$\sum_{i=i_0}^{i_k} \text{DIST}^2(P_i, D) \geq \epsilon$$

Par contre, si le segment est maximal à droite, il ne l'est pas forcément à gauche - on ne sait pas si

$$\sum_{i_0-1}^{i_k-1} \text{DIST}^2(P_i, D) \geq \epsilon$$

L'utilisation du test (2) pour définir les points de cassure a des conséquences indésirables.

* A la fin d'un "bon" segment de droite il aura tendance à situer le point de cassure au-delà de l'angle (cf. figure II-23) parce qu'il faut un certain nombre de points pour dégrader l'écart E.

* Un segment de droite long mais bruité sera coupé au milieu d'une façon tout à fait arbitraire.

Si on utilise l'écart moyen (comme Roberts l'a effectivement fait)

$$(2') \sum \text{DIST}(P_i, D) / (i_k - i_0 + 1) < \epsilon$$

on résoud le deuxième problème, mais on aggrave le premier.

En ce qui concerne l'efficacité, cet algorithme est relativement lent, en raison des calculs d'approximation répétés. Une programmation naïve amène à un temps de calcul de l'ordre n^2 . Nous verrons au paragraphe suivant qu'on peut le ramener à l'ordre n .

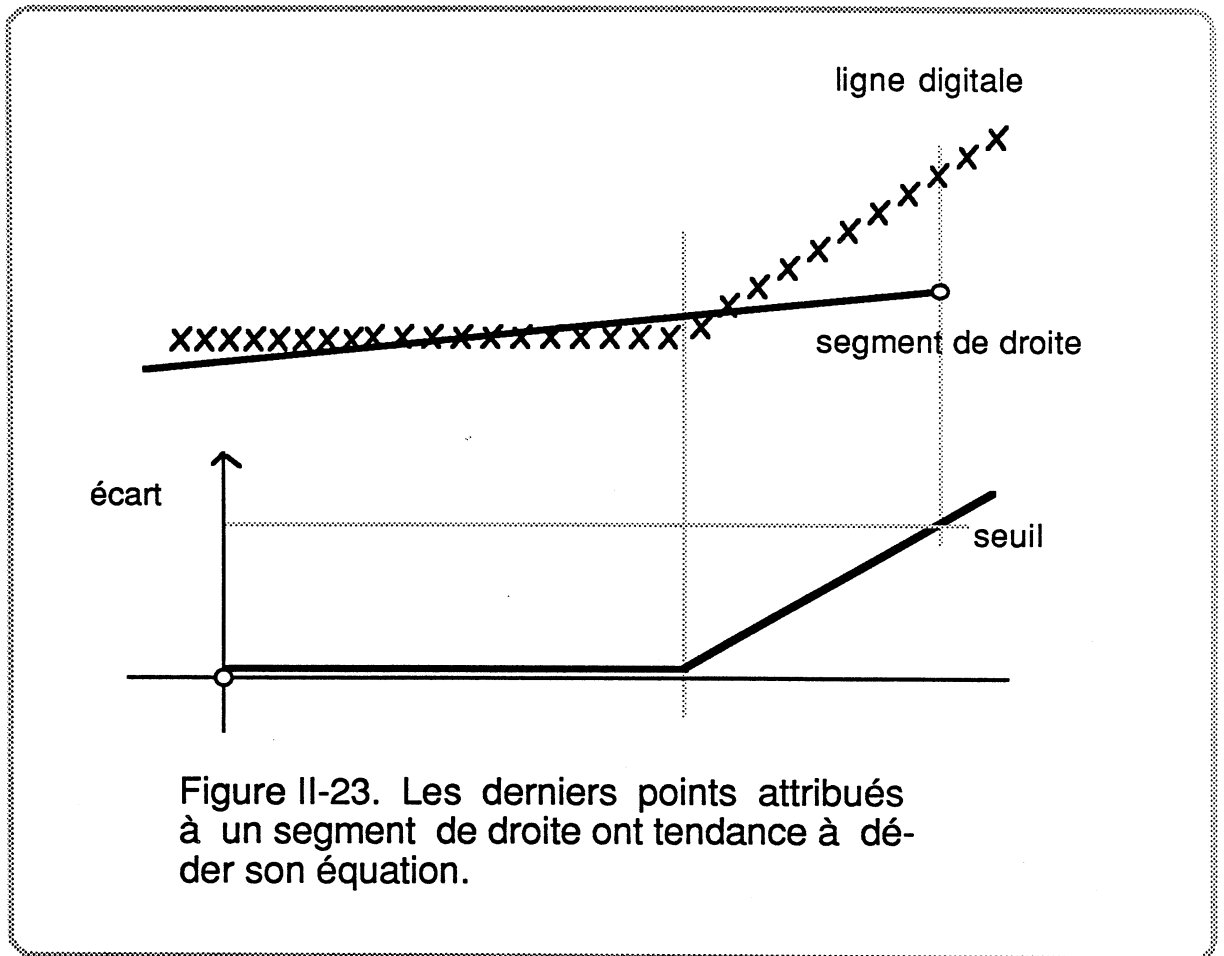


Figure II-23. Les derniers points attribués à un segment de droite ont tendance à dé-der son équation.

3.3.3 Approximation aux moindres carrés

Les formules d'une droite approchant un nuage de points sont bien connues pour la représentation

$$y = a \cdot x + b$$

Dans notre cas, cette représentation est mal adaptée; puisqu'elle n'est pas valable pour les droites verticales. De plus, l'erreur minimisée n'est pas la distance des points à la droite, mais la "distance verticale", ce qui peut donner des résultats paradoxaux. Il est préférable de représenter les droites par la forme normale

$$x \cdot \cos(\varphi) + y \cdot \sin(\varphi) + r = 0$$

Nous dérivons l'équation de la meilleure droite approchant un nuage de points avec cette forme, en suivant la démarche élégante proposé dans [Duda et Hart73]. Une dérivation plus "terre à terre" du même résultat se trouve dans [Horn 72].

On utilise une notation vectorielle; l'équation de la droite devient alors

$$P.V + r = 0$$

où V est le vecteur unitaire normal à la droite.

Etant donné un nuage de n points $\{P_i\}$ nous cherchons à minimiser

$$e^2 = \sum (P_i.V + r)^2$$

En posant la dérivée par rapport à r égale à 0 on obtient

$$r = - P_c . V$$

P_c étant le centre du nuage:

$$P_c = \frac{1}{n} * \sum P_i$$

Le droite recherchée passe donc par le centre de gravité du nuage. On opère un changement de coordonnées afin d'amener ce point dans l'origine:

$$P'_i = P_i - P_c$$

Dans ce nouveau système

$$r'=0$$

pour l'équation de notre droite, ce qui simplifie bien les formules pour la détermination du vecteur V. Etant donné qu'il faut minimiser

$$e^2 = \sum_{i=0}^n (P'_i . V)^2$$

le vecteur directeur de la droite recherchée est vecteur propre associé à la plus grande valeur propre de la matrice de covariance des points

$$M = \sum (P'_i . P'_i) = \begin{pmatrix} X & Z \\ Z & Y \end{pmatrix}$$

Cette formulation a l'intérêt d'être indépendante de la dimension de l'image. Elle s'applique à la recherche de droite dans une image 2-D aussi bien qu'à la recherche d'un

plan dans une image 3-D, et par là à la recherche de droites dans une image 3-D.

Dans le cas 2-D on trouve la valeur propre

$$VP = \frac{X+Y}{2} + \sqrt{\left(\frac{X-Y}{2}\right)^2 + Z^2}$$

Deux vecteurs propres associés sont

$$\begin{pmatrix} Z \\ \frac{VP-X}{1} \\ 1 \end{pmatrix} \quad \begin{pmatrix} 1 \\ Z \\ \frac{VP-Y}{1} \end{pmatrix}$$

Un au moins de ces deux vecteurs est toujours défini.

En ce qui concerne la complexité algorithmique de ces calculs, une observation simple est d'une grande importance: ces calculs peuvent être effectués de façon incrémentale. En effet, les valeurs utilisées sont facilement obtenues à partir des cinq valeurs

$$\sum x_i \quad \sum y_i \quad \sum x_i^2 \quad \sum y_i^2 \quad \sum (x_i * y_i)$$

en observant que

$$X = \sum x_i'^2 = \sum x_i^2 - x_c * \sum x_i$$

$$Y = \sum y_i'^2 = \sum y_i^2 - y_c * \sum y_i$$

$$Z = \sum (x_i' * y_i') = \sum (x_i * y_i) - y_c * \sum x_i$$

(Ces formules se trouvent déjà dans [Horn 72]!).

Dans le cas des segments de droites digitales il faut cependant se méfier d'une illusion de trop grande précision de la "droite optimale". La donnée d'une séquence finie de points ne nous permet pas de déduire une droite unique. Ce point sera discuté de façon plus détaillée avec l'évaluation de l'algorithme de reconnaissance syntaxique.

3.3.4 Découpage récursif

Nous avons critiqué dans l'algorithme de Roberts le fait que la détermination des points de cassure se fait de façon trop indirecte. Le sous-problème B est en quelque sorte traité comme un simple corollaire du sous-problème A. Le découpage récursif procède par ordre inverse - il traite la détermination des équations de droites comme un corollaire du découpage.

Le principe est le suivant: soit VP un vecteur contenant les points de la ligne à découper. On considère la droite passant par le premier et le dernier point; soit Pmax le point le plus éloigné de cette droite (figure II-24), à une distance Dx. Deux cas peuvent se présenter.

- a) Dx est inférieur à une certaine "tolérance latérale" EPSD. Tous les points sont donc situés dans une bande de largeur $2 \cdot \text{EPSD}$ le long de la droite, et on considère que la ligne est composée d'un seul segment de droite.
- b) Dx est supérieur à EPSD. On considère alors Pmax comme un point de cassure. Il faut donc découper séparément la première partie de la ligne, allant jusqu'à Pmax, et la deuxième partie commençant à Pmax.

La terminaison des appels récursifs est garantie, car le nombre de points à traiter diminue strictement à chaque appel imbriqué. Par contre, si la valeur de EPSD est faible, cela peut amener à une forte fragmentation de la ligne, comme dans la figure II-20. Pour cette raison il est intéressant d'introduire un deuxième paramètre, MINLO, correspondant à un nombre de points minimum par segment (MINLO est donc supérieur à 2).

3.3.4.1 Description de l'algorithme

Découpage récursif d'une ligne de contraste fournie dans le vecteur de points VP. Les NBD droites du résultat sont placées dans le vecteur VRD.

On utilise deux fonctions auxiliaires au profil d'appel suivant:

```
function DPP(p1,p2:POINT):DROITE  
  Construction d'une droite à partir de deux points.
```

```
function DISTPD(p:POINT,d:DROITE):real  
  Distance d'un point à la droite
```

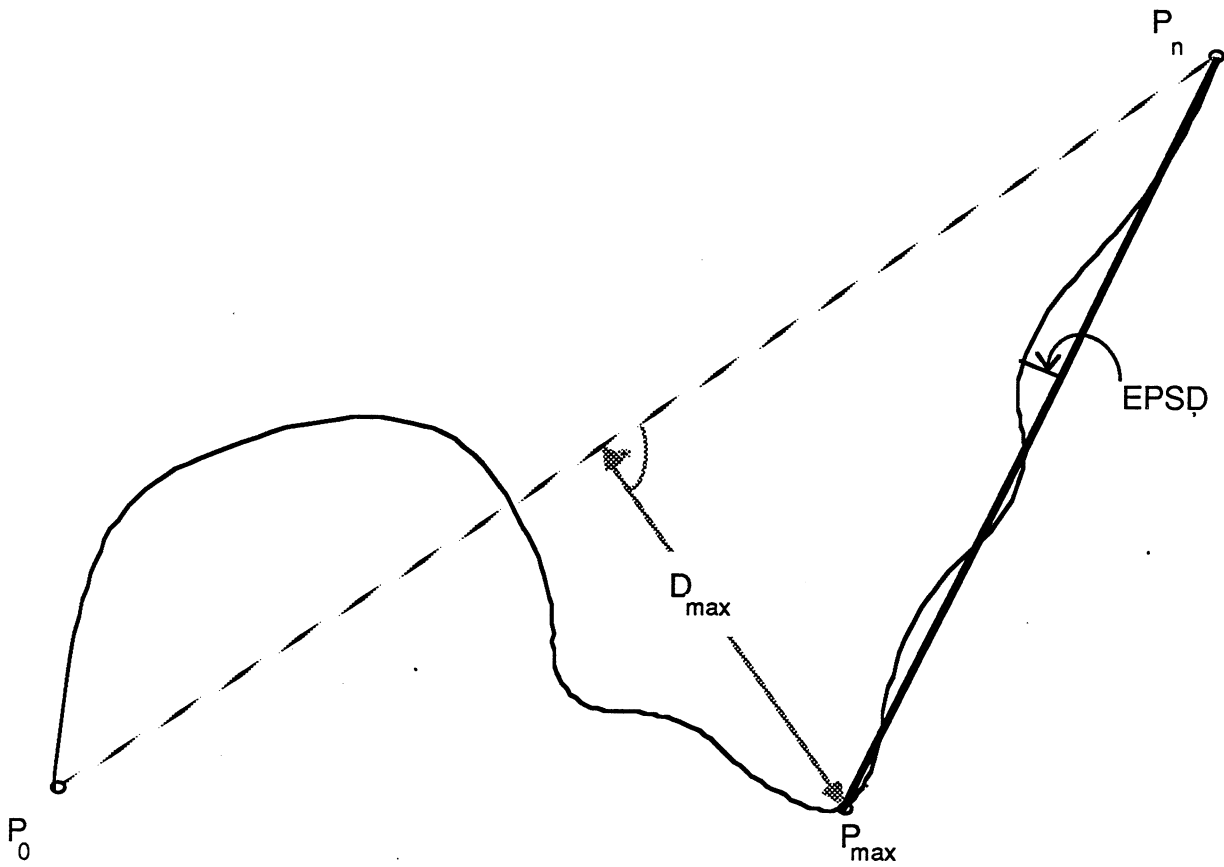


Figure II-24. Découpage récursif: premier point de cassure.
 Le segment $\overline{P_{max}P_n}$ satisfait au critère P et sera donc retenu,
 alors que la partie $\overline{P_0P_{max}}$ sera encore subdivisée.

Paramètres globaux:
 EPSD tolérance latérale
 MINLO longueur minimale des droites (valeur entière en pixels)

Déclarations

```
VP : array[1..N] of POINT;
VRD : array[1..K] of DROITE;
NBD : integer;
```

```
procedure DECOUP(I1,I2:integer);
var
  D : DROITE;
  ECART : real;
  IX : integer;
```

```
begin
  if I2-I1 > MINLO then
    begin
      D := DPP(VP(I1),VP(I2));
      ECART := 0.0;
      for j := I1+1 until I2-1 do
        if DISTPD(VP(j),D) > ECART then
          begin
            IX := j;
            ECART := DISTPD(VP(j),D)
          end;
        if ECART < EPSD then {création segment résultat}
          begin
            NBD := NBD+1;
            VRD(NBD) := D
          end
        else
          begin
            DECOUP(I1,IX);
            DECOUP(IX,I2)
          end
        end
      end
    end
  end
end
```

```
{programme principal}
NBD := 0;
DECOUP(1,N);
```

En cas de gestion explicite de la recursion (en FORTRAN par exemple), il suffit d'empiler les valeurs de IX. On remonte alors avec

```
  I1 := I2; I2 := depiler;
```

La taille de la pile nécessaire est de $N/\text{MINLO}+1$ (cf. évaluation algorithmique).

3.3.4.2 Evaluation des résultats

C'est un algorithme élégant; connu depuis longtemps, il est notamment décrit dans [Duda et Hart 1973] et utilisé dans de nombreuses applications, par exemple [Nevatia et Babu 80, Ayache 83, Herman et Kanade 84]. D'un point de vue pratique, il donne des résultats très satisfaisants que l'on peut, grâce aux deux paramètres EPSD et MINLO, adapter aux caractéristiques de la ligne analysée. Cependant, les propriétés géométriques des résultats ne sont pas évidentes à déduire de l'algorithme. Nous allons les caractériser dans ce paragraphe; en particulier, nous nous intéressons aux points de cassure et à leur stabilité, c'est à dire à leur dépendance vis à vis de facteurs fortuits. Par exemple, étant donné un point de cassure P_x , trouvé à l'occasion d'un appel initial DECOUP(S,T), est-on sûr de trouver ce même point de cassure en découpant une ligne englobante avec un appel DECOUP(S',T'), ($S' < S, T' > T$)?

A - Propriétés simples

La construction même de l'algorithme nous garantit une propriété simple et utile, que nous appelons le **critère de proximité** ou **critère P**: une bande de largeur $2*EPSD$ le long d'un segment contient les points $(P_{i_0} P_{i_0+1} \dots P_{i_k})$, autrement dit, chaque point de la ligne à analyser se trouve à une distance inférieure à EPSD d'un segment de droite du résultat.

$$(1) \text{ critère P: } \text{DIST}(P_i, D) < \text{EPSD} \quad i=i_0, \dots, i_k$$

Mais cette propriété n'est pas poussée très loin. L'algorithme ne minimise pas

$$\max_D \text{DIST}(P_i, D)$$

et ne satisfait ni au critère d'approximation MC, ni au critère de maximalité.

Notons aussi deux autres propriétés du résultat:

- (2) Les parties de la ligne correspondant à deux segments différents sont disjointes, mais la réunion des segments du résultat ne forme pas nécessairement la ligne entière.
- (3) Les segments de droites du résultat sont obtenus et rangés dans le vecteur VD dans l'ordre de composition de la ligne.

B - Les cas problématiques

En essayant de prouver le bon fonctionnement de l'algorithme nous trouvons d'abord un certain nombre de cas dans lesquels il ne donne pas les résultats qu'on pourrait espérer. La mise en œuvre du critère P dans cet algorithme entraîne des problèmes de stabilité. Un changement minime et accidentel des données du problème peut changer radicalement l'allure des résultats, contrairement à ce qu'on attend d'une bonne segmentation en indices visuels. Différents cas d'instabilité sont illustrés par les figures II-25 à II-28.

- * Le premier cas concerne les "zig-zags" (figure II-25 a et b). Dans ce cas le nombre de segments trouvés peut varier entre 1 et une valeur k (qui dépend des détails de l'exemple, mais pour toute valeur voulue de k on peut construire un exemple). Ce problème est lié à la présence de hautes fréquences spatiales dans la ligne; pour l'éviter il faut harmoniser les valeurs de EPSD et MINLO, en choisissant

$$\text{MINLO} \gg \text{EPSD}$$

(cf. figure II-25-c).

- * Le deuxième cas, illustré par la figure II-26 correspond à un problème plus fondamental et inévitable. Si le point Pmax se trouve sur un segment parallèle à $\overline{P_{i1}P_{i2}}$, il est toujours possible qu'un bruit découpe ce segment en deux morceaux. Autrement dit, on ne peut garantir que le nombre de droites produites par le découpage soit indépendant du bruit.
- * Le troisième cas concerne les lignes à courbure relativement constante, comme un arc de cercle. Dans ce cas, les points de cassure sont arbitraires, dépendant complètement du choix des points initiaux. La courbure doit cependant être assez faible - une corde de longueur MINLO doit s'éloigner de moins de EPSD de la circonférence (cf. figure II-42-a):

$$1/c = r \geq (\text{MINLO}^2 + 4*\text{EPSD}^2)/2*\text{EPSD}$$

- * Le quatrième problème concerne les configurations en forme de Z (figure II-28) qui peuvent, dans certaines circonstances, disparaître au découpage.

C - Découpage d'un train polygonal

Etant donné ces cas pathologiques il est surprenant de constater que cet algorithme a été utilisé dans de nombreuses applications avec des résultats tout à fait satisfaisants. En particulier [Ayache 82] l'utilise pour la modélisation et la

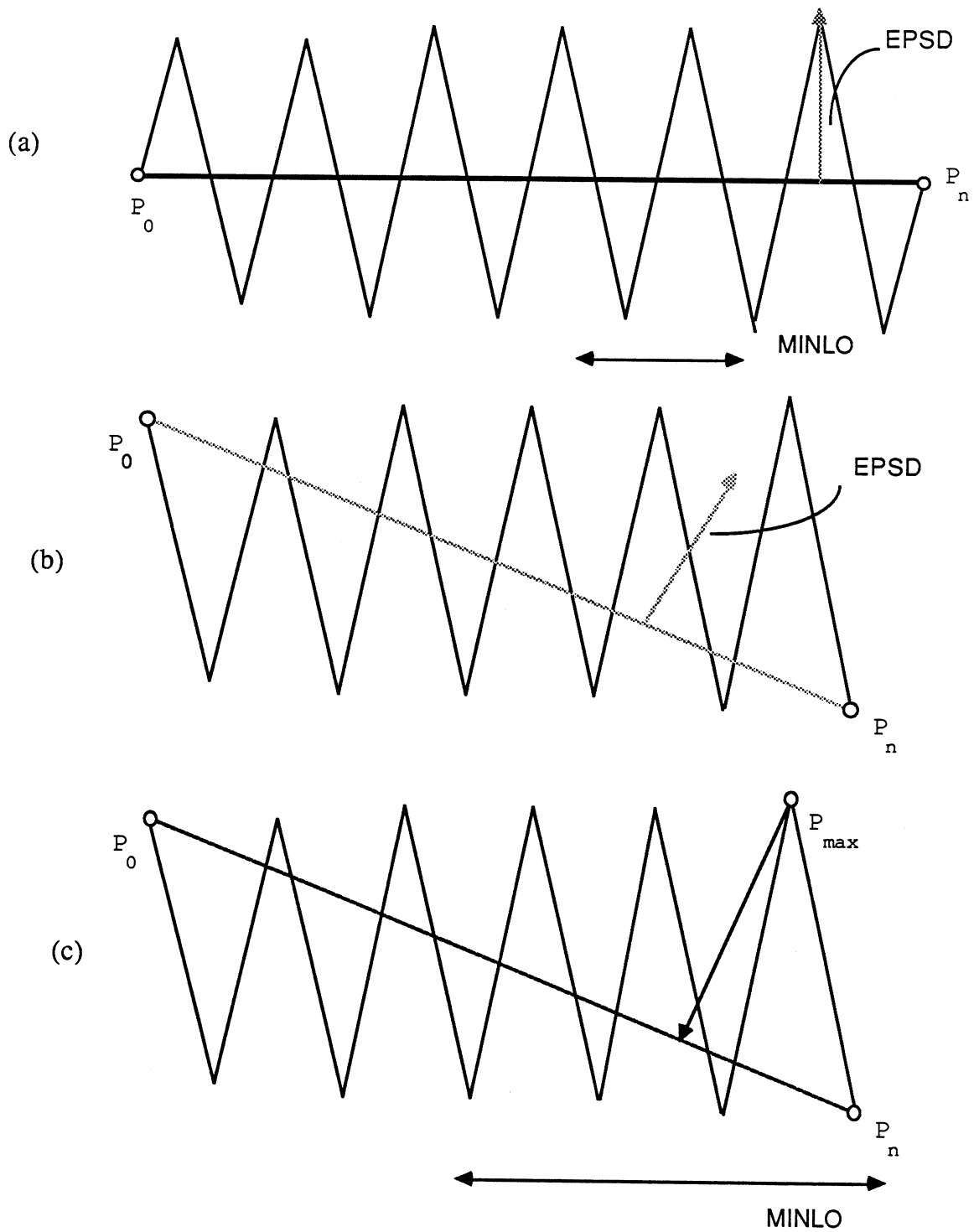


Figure II-25. Instabilité du découpage récursif - le zig-zag. En fonction des emplacements des extrémités, on obtient tantôt un seul segment de droite (a), tantôt un nombre de segments élevé dans la figure (b). Dans les mêmes conditions, mais avec $MINLO \gg EPSD$, aucun des segments n'est retenu (c).

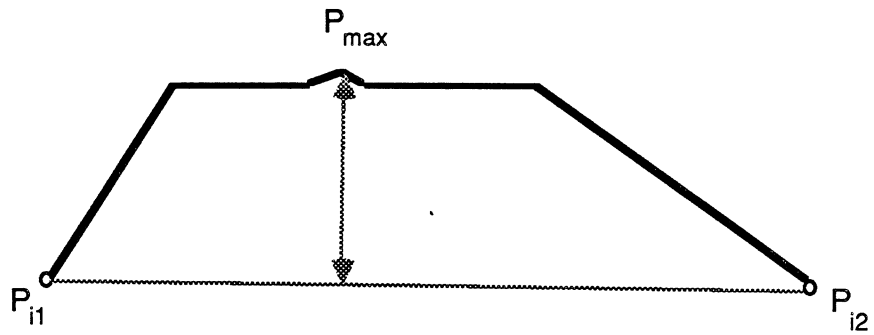


Figure II-26: Instabilité du découpage récursif. Un segment parallèle à $\overline{P_{i1}P_{i2}}$ peut être cassé en raison du moindre bruit.

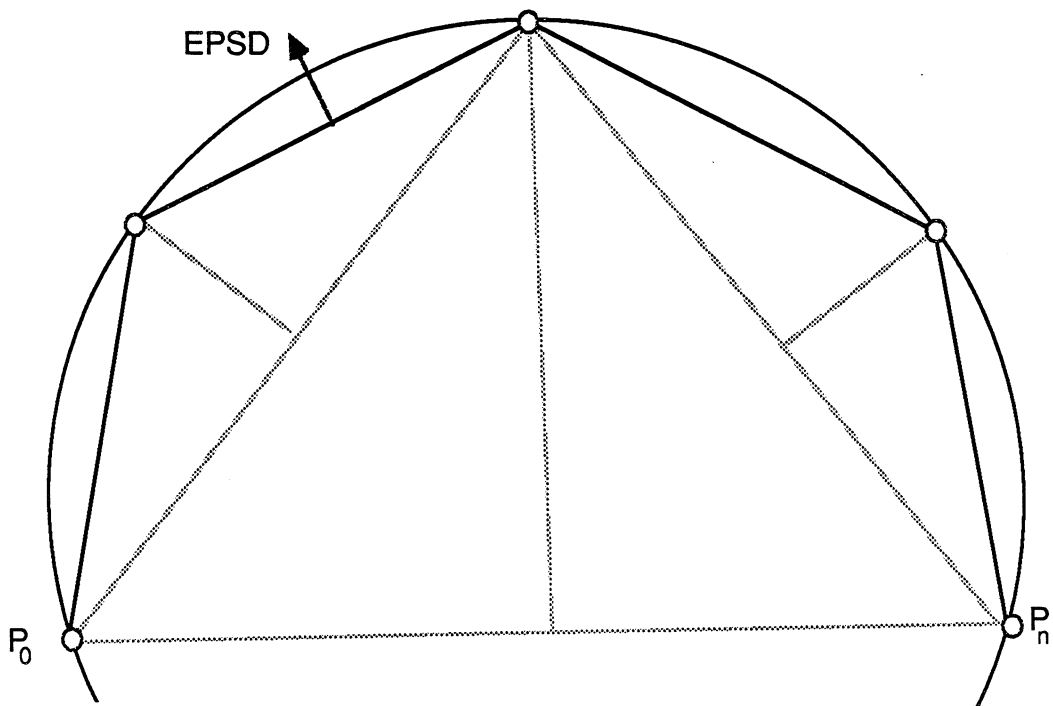
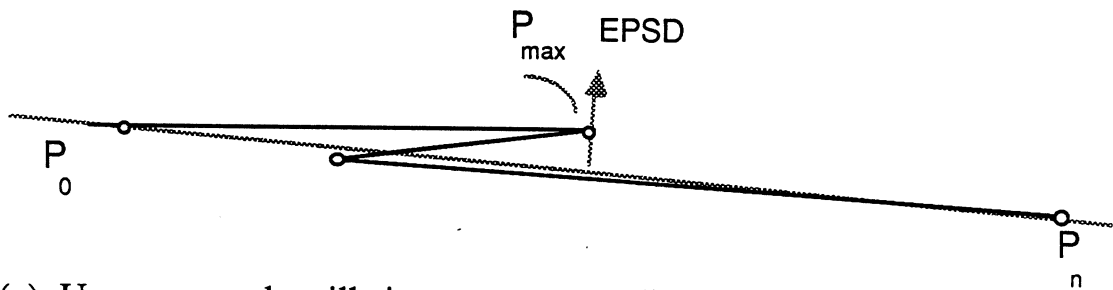
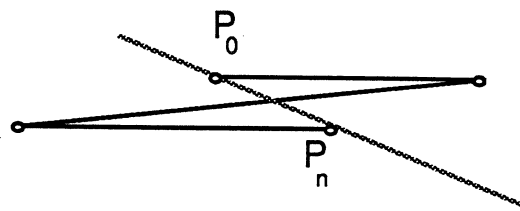


Figure II-27. Un arc de cercle est découpé de façon arbitraire en cordes de longueurs égales, et telles qu'elles s'éloignent de moins de EPSD de la circonférence du cercle.



(a) Un morceau de taille importante peut disparaître au découpage.



(b) Le segment $\overline{P_0 P_n}$ représente l'ensemble de la ligne!

Figure II-28. Instabilité du découpage récursif: le problème du Z.

reconnaissance d'objets de forme irrégulière, dont un exemple est donné par la figure II-29. Bien entendu, dans ce contexte la stabilité du découpage est d'une importance capitale, même si l'algorithme d'interprétation peut rattraper quelques imperfections.

Nous allons donc chercher des conditions garantissant la stabilité des résultats.

Nous nous intéressons d'abord au cas où les données $(P_0 \dots P_1 \dots P_n)$ constituent un train polygonal; celui-ci est entièrement défini par les extrémités des segments $[X_0=P_0, X_1, \dots, X_k=P_n]$. Dans ce cas, le bon fonctionnement de l'algorithme est facile à définir: les points de cassure doivent être exactement les points extrémités, l'algorithme doit "casser la ligne aux angles".

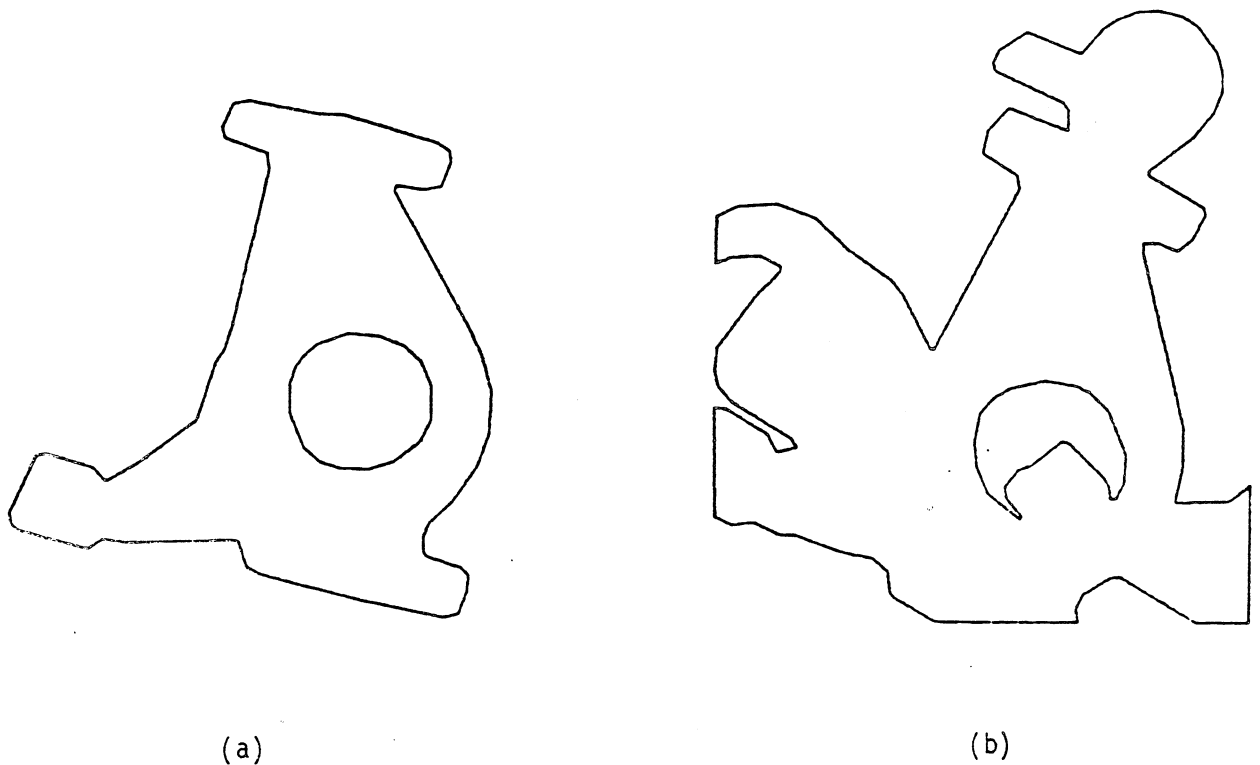


Figure II-29. Approximation polygonale dans [Ayache 83].
 a) un modèle constitué de 66 segments
 b) une scène composé de 117 segments

Nous notons L_i la longueur du i° segment, dont les extrémités sont (X_{i-1}, X_i) , et A_i l'angle associé au point X_i .

Une première condition s'impose trivialement: la longueur L_i de chaque segment doit être supérieure à MINLO. Nous sous-entendrons cette condition par la suite.

Nous donnons d'abord une condition suffisante qui garantit que les points de cassure font partie des extrémités X_i ; pour cela il faut éviter le deuxième cas à problème ci-dessus.

PROPRIETE 1. Si dans l'analyse d'un train polygonal

(a) pour deux extrémités quelconque X_s, X_t , aucun segment $\overline{X_i X_{i+1}}$ n'est parallèle au segment $\overline{X_s X_t}$

alors

les points de cassure déterminés par le découpage récursif sont des points extrémités X_i .

Il faut aussi une condition garantissant que chaque point X_i est effectivement trouvé comme point de cassure. Elle se trouve facilement dans le cas d'une ligne convexe.

PROPRIETE 2. Si le train polygonal est convexe et qu'en plus de la condition (a)

(b) pour chaque extrémité X_i ($i=1, \dots, k-1$)

$$\text{DIST}(X_i, \overline{X_{i-1} X_{i+1}}) > \text{EPSD}$$

alors

les points de cassure du découpage récursif sont exactement les points extrémités $X_0 \dots X_k$.

La condition de convexité élimine les problèmes de la figure II-28, et la condition (b) assure que chaque sommet se détache suffisamment de son entourage pour qu'il apparaisse comme point de cassure.

Si le train polygonal n'est pas convexe, nous avons besoin de conditions plus fortes pour nous prémunir contre les deux dernières configurations pathologiques.

PROPRIETE 3. Si en plus de la condition (a)

(c) pour chaque séquence de trois points X_{i-1}, X_i, X_{i+1}

$$\text{c-1) } \text{DIST}(X_{i-1}, \overline{X_i X_{i+1}}) = L_{i-1} * \sin(A_i) > 2 * \text{EPSD}$$

$$\text{c-2) } \text{DIST}(X_i, \overline{X_{i-1} X_{i+1}}) > 2 * \text{EPSD}$$

$$\text{c-3) } \text{DIST}(X_{i+1}, \overline{X_{i-1} X_i}) = L_i * \sin(A_i) > 2 * \text{EPSD}$$

alors

les points de cassure du découpage récursif sont exactement les points extrémités $X_0 \dots X_k$.

Cette propriété se démontre assez simplement (figure II-30): les conditions (c) assurent que quelque soit la droite D dans le plan

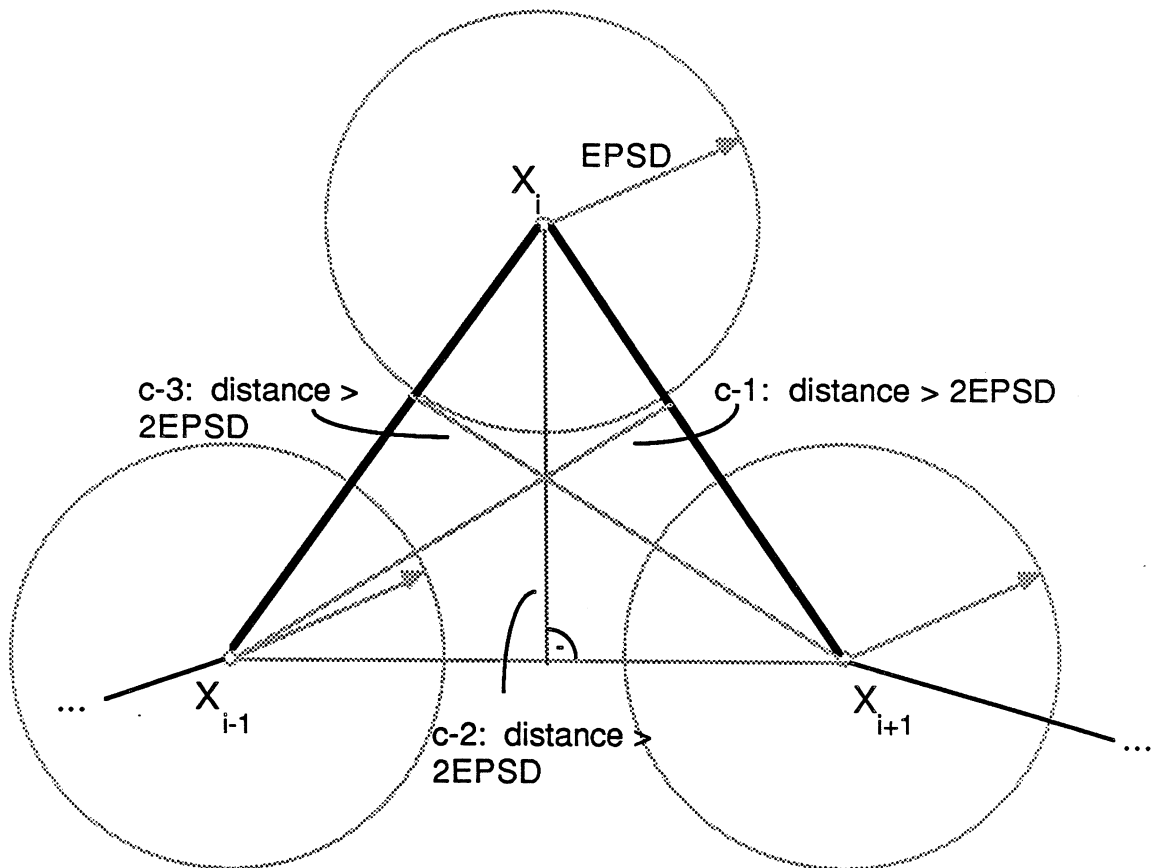


Figure II-30. Les conditions (c) garantissent que les cercles de rayon EPSD autour des trois points X_i sont disjoints. Aucune droite dans le plan ne peut donc intersecter les trois cercles autour de trois points X_i consécutifs.

$$\max\{\text{DIST}(X_{i-1}, D), \text{DIST}(X_i, D), \text{DIST}(X_{i+1}, D)\} > \text{EPSD}$$

ce qui implique que le découpage ne peut s'arrêter avant d'avoir établi les trois points de cassure.

D - Considérations pratiques

Est-il raisonnable d'imposer les conditions (a,b,c) à une application? La condition (a) paraît la plus gênante, d'autant plus qu'elle n'est pas nécessaire (cf. figure II-31-a). Surtout pour des objets manufacturés, on ne peut exclure des lignes et des cordes parallèles. (Il ne semble cependant pas possible de trouver une condition suffisante plus faible qui soit simple).

Nous voyons deux possibilités raisonnables sur le plan pratique pour éviter la condition (a): on peut modifier l'algorithme de façon à détecter les points de cassure qui ne sont pas parmi les X_i (c'est la variante avec "fusion de droites" décrite au paragraphe 3.3.4.4), ou on peut carrément accepter le fait que certains segments peuvent être coupés en deux et en tenir compte dans les autres modules du système de vision.

La condition (b) dans le cas convexe, ainsi que les conditions (c) dans le cas général, imposent des restrictions plus raisonnables à la forme des lignes - essentiellement elles excluent des angles trop pointus ou trop obtus - et peuvent être facilement vérifiées. Si ces conditions ne sont pas strictement nécessaires, comme le montre la figure II-31, on peut cependant les considérer comme "localement nécessaires": tout triplet de points extrémités ne satisfaisant pas aux conditions (c) peut être facilement étendu en un train polygonal à 5 sommets que le découpage récursif ne cassera pas aux angles.

Dans de nombreux cas, pour une forme ne satisfaisant pas aux conditions (c), une analyse globale plus poussée est possible. Il faut alors faire intervenir les valeurs de

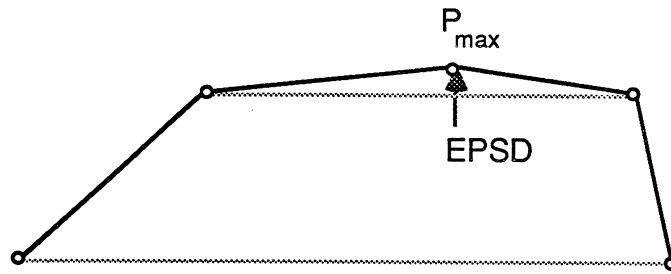
$$\text{DIST}(X_i, \overline{X_s X_t})$$

et étudier l'ordre dans lequel les points de cassure peuvent être établis, en tenant éventuellement compte des possibilités de visibilité partielle. Dans certains cas particuliers, comme celui d'un filet de vis, une telle analyse globale est facile, et permet même de remplacer les conditions (c) par une condition nécessaire et suffisante.

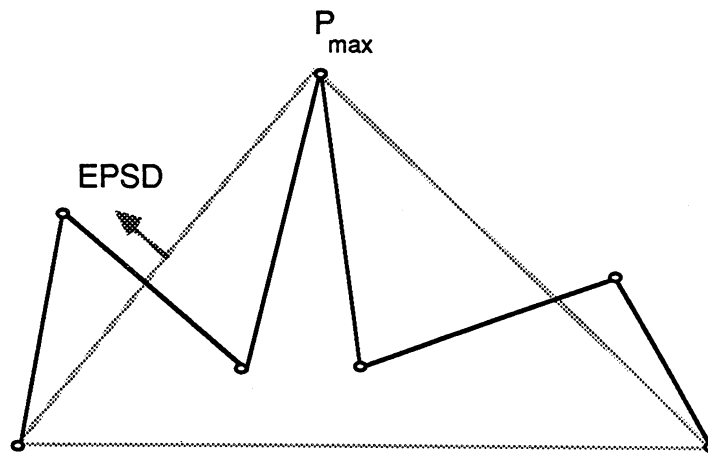
E - Découpage d'une ligne quelconque

L'analyse du découpage d'un train polygonal permet de caractériser et d'expliquer, quoique de façon très qualitative, le découpage d'une ligne quelconque.

Seulement dans le cas d'une ligne à courbure relativement constante le résultat sera totalement arbitraire (cf. figure II-27). Dans les autres cas, les angles, ou les endroits à



(a) Condition (b)



(b) conditions (c)

Figure II-31. Les conditions (b) dans le cas convexe, les conditions (c) dans le cas général ne sont pas nécessaires: les deux trains polygonaux de la figure sont découpés correctement. Cependant, on ne peut plus le garantir en cas de visibilité partielle.

forte courbure (forcément très courts) auront tendance à émerger comme les premiers points de cassure; une fois encadrées par ces points angulaires, les parties rondes seront découpées d'une façon arbitraire, mais fixe. Notons cependant que les résultats du découpage récursif risquent de varier fortement avec l'échelle, à moins d'adapter les paramètres MINLO et EPSD.

3.3.4.3 Evaluation algorithmique

Le découpage récursif a la même structure algorithmique que le tri par segmentation dont on peut reprendre l'évaluation [cf. Berlioux et Bizard 83].

En ce qui concerne la place mémoire, il faut stocker l'ensemble des n points, et il faut une pile pour la gestion des appels récursifs. Dans le cas le plus défavorable (figures II-25 et II-32), la taille de la pile doit être de l'ordre du nombre de droites du résultat, elle varie comme n/MINLO . (Il suffit d'empiler une seule valeur, celle de IX , indice de P_{\max} dans la table).

Une légère variation de l'algorithme - l'inversion de l'ordre des appels récursifs si $I2-IX < IX-I1$ - permet de réduire la taille de la pile à l'ordre

$$\text{LOG}(n/\text{MINLO})$$

En contrepartie, il faut empiler deux valeurs par appel récursif, et les droites ne sont pas construites dans l'ordre d'apparition dans la ligne.

Remarque. Si on gère explicitement la recursion, la taille de la pile n'est pas une quantité négligeable, puisqu'une ligne peut comporter des milliers de points.

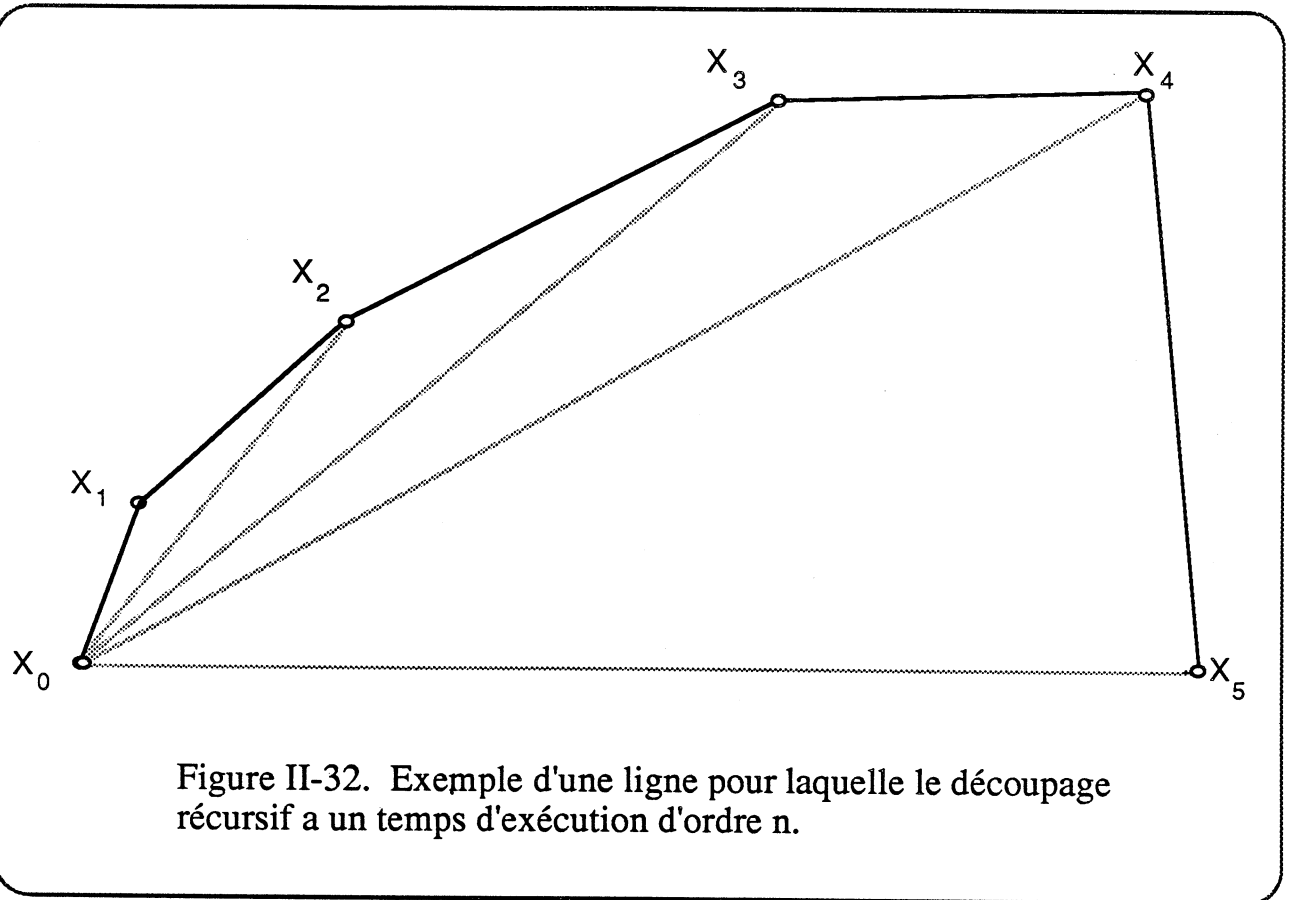
Pour le temps d'exécution, nous nous concentrons sur le nombre de calculs de distance entre un point et une droite. Ce nombre est de l'ordre n^2 dans le cas le plus défavorable, et n dans le cas le plus favorable (celui d'un seul segment de droite). Pour la suite, nous aurons besoin d'une évaluation plus fine en fonction du nombre de droites du résultat. En supposant qu'il y a ND segments de droites de même longueur, on obtient, pour le cas le plus défavorable, un temps proportionnel à

$$n * (ND-1)/2$$

et pour le cas le plus favorable

$$n * \log_2(ND)$$

Bien que l'on ne puisse pas définir un comportement moyen sans référence à un ensemble de formes précises, on peut dans la majorité des cas s'attendre à des temps plus proches du cas le plus favorable, car le cas le plus défavorable correspond à un nombre limité de formes géométriques très caractéristiques (dents de scie, quart d'ellipse, spirale).



3.3.4.4 Variantes de l'algorithme

A - Qualités d'approximation

Si les segments trouvés représentent effectivement une approximation de la ligne digitale, ils n'ont pas un caractère optimal. Il n'y a cependant aucun inconvénient à compléter l'algorithme par une approximation aux moindres carrés, même s'il est difficile de satisfaire les critères P et MC simultanément.

- * A la création de chaque segment résultat on peut effectuer le calcul d'approximation. En contrepartie, cette nouvelle droite ne garantit plus le critère P.
- * Alternativement, on peut combiner les critères P et MC, et à la place de la fonction DPP (qui se contente de calculer la corde entre deux points) utiliser une fonction fournissant une droite optimale. Cela a deux conséquences: d'une part on produit un découpage différent de la ligne; d'autre part, les segments

résultat n'ont plus de raison d'être jointif. Si l'on veut construire un train polygonal à partir des résultats, il faut prolonger les segments aux extrémités, et les points ainsi ajoutés peuvent ne pas satisfaire le critère P.

Dans les deux cas, le temps d'exécution de l'algorithme sera sensiblement augmenté.

B - Fusion de segments

Dans de nombreuses applications on a affaire à des contours contenant des segments parallèles. La condition (a) de la propriété P1 n'est donc pas vérifiée, et le moindre bruit peut casser en deux un segment (figure II-26). Si cela est gênant pour l'algorithme d'interprétation, on peut compléter le découpage par une étape de fusion de segments.

Dans ce cas, on vérifie deux conditions avant la création d'un nouveau segment:

- * le nouveau segment est-il proche du segment à sa gauche (le dernier créé)?
- * l'ensemble des points associés aux deux segments satisfait-il au critère P (proximité) pour un seul segment de droite?

Si ces deux conditions sont vérifiées il suffit d'ajuster le segment précédent au lieu d'en créer un nouveau.

Dans la mesure où le premier test, qui demande peu de calculs, sera en général décisif, le coût de cette variation est négligeable.

C - Réduction du nombre de comparaisons

A l'opposé des éléments d'un vecteur à trier, les points d'une ligne de contraste ne sont pas des données indépendantes - entre deux points consécutifs la distance varie aux plus de $\sqrt{2}$ - ce qui permet de réduire le nombre de points à soumettre au test

$$\text{DISTPD}(\text{VP}(j), D) > \text{ECART}$$

(l'analogie du nombre de comparaisons du tri et élément critique pour le coût), et cela de deux façons:

- a) Si le point VP(j) est proche de la droite en comparaison avec la valeur de ECART, les points suivants - VP(j+1), ... - ne peuvent avoir un écart supérieur. Plus précisément, le prochain à tester aura l'indice

$$j + \lfloor (\text{ECART} - \text{DISTPD}(\text{VP}(j), D)) / \sqrt{2} \rfloor$$

b) Le même raisonnement s'applique à propos de la valeur EPSD, en dessous de laquelle on considère que les points font partie de la droite D. L'indice du prochain point utile à tester ne sera pas inférieur à

$$j + \lfloor (\text{EPSD} - \text{DISTPD}(\text{VP}(j), D)) / \sqrt{2} \rfloor$$

Ce deuxième cas est moins intéressant dans la mesure où EPSD aura en général une valeur faible (typiquement entre 1 et 3).

Cette variation a cependant une valeur toute relative - le gain de temps qui peut en résulter dépend entièrement des données, et à la limite le surcoût de calcul occasionné par les formules ci-dessus peut même résulter dans une dégradation de la performance.

3.3.5 Reconnaissance syntaxique

3.3.5.1 Description

La méthode syntaxique exploite pour la reconnaissance un modèle génératif de droite digitale proposé initialement par [Rosenfeld 75]. Le modèle part d'une image idéalisée dans laquelle un pixel est représenté par un carré du plan. Un pixel de coordonnées entières (X,Y) couvre l'ensemble des points (x,y) à coordonnées réelles tels que

$$X \leq x < X+1, Y \leq y < Y+1$$

Nous considérons qu'un pixel fait partie de "l'image digitale" de la droite si et seulement si l'intersection entre le pixel et la droite est non-vide; de plus, on élimine les angles droits au profit de diagonales. Cette image constitue une ligne digitale. La figure II-33 en montre un exemple, avec une droite passant par l'origine et de pente m faible - $0 < m < 1$. Dans ce cas, la ligne est composée des pixels (X,Y) tels que

$$Y = \lfloor m * X \rfloor$$

On constate que cette ligne est formée d'une suite de "barrettes", et que la longueur des barrettes varie au plus de 1. Cela s'explique facilement. La longueur L de la première barrette à partir de l'origine est telle que

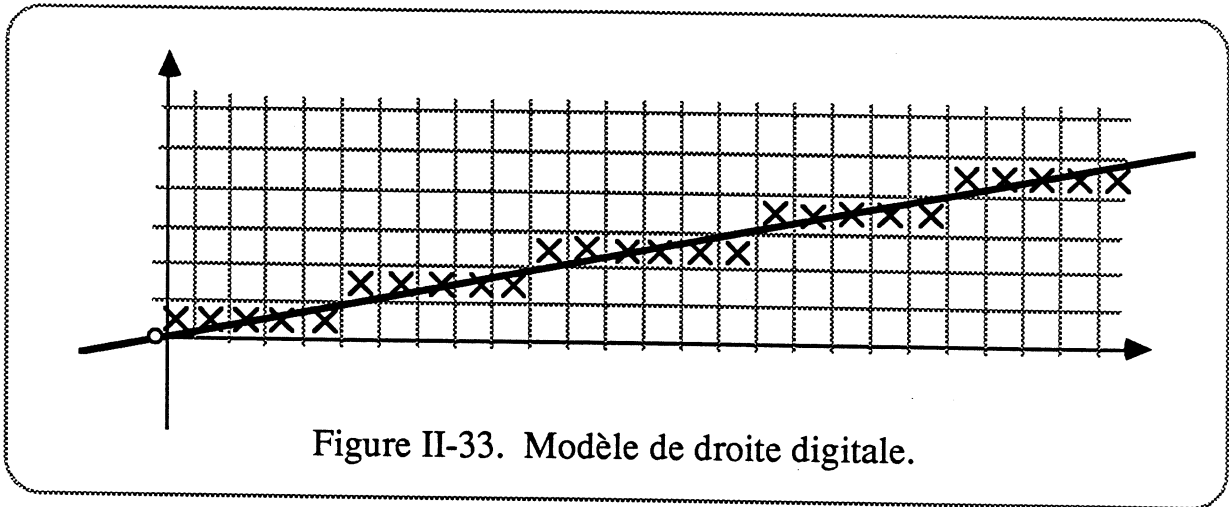


Figure II-33. Modèle de droite digitale.

$$(1) \quad L*m < 1 \leq (L+1)*m$$

c'est à dire L est le quotient entier $L = 1 \text{ DIV } m$. Avec l'hypothèse de récurrence d'une ligne de n barrettes de longueur L_i

$$(2) \quad \sum_{i=1}^n (L_i * m) = m * \sum_{i=1}^n L_i < n < m * (1 + \sum_{i=1}^n L_i)$$

on additionne l'inégalité (1)

$$L*m + \sum (L_i * m) < n + 1 \leq m * (2 + L + \sum L_i)$$

La deuxième inégalité implique $L_n = L$ ou $L_n = L+1$.

Une étude plus détaillée (mais sans difficulté, étant donné les cas de symétrie) montre que les régularités rencontrées dans le cas $0 < m < 1$ correspondent au cas général. Elles nous permettent de caractériser une droite digitale par la grammaire des codes direction (codes de Freeman):

PROPRIETE. La suite S des codes direction d'une droite digitalisée est de la forme

$$(3) \quad S = \{D_p^{+1} D_s\}^*$$

Au plus deux codes différents figurent dans S , celui de la direction principale D_p et celui de la direction secondaire D_s . Ces codes correspondent à deux directions voisines

$$D_s = (D_p \pm 1) \text{ mod } 8$$

et seul le code principal peut se répéter. La longueur d'une répétition peut prendre deux valeurs différentes,

soit L ou L+1.

L'image d'un segment d'une droite est un mot de ce langage, sauf pour la première et la dernière barrette qui peuvent être tronquées, ou alignées avec une barrette d'un segment voisin.

Remarque: on peut caractériser S plus finement. Avec un raisonnement analogue au précédent on montre que les barrettes de longueur L et L+1 sont disposées en fonction du quotient

$$r_1 = 1 \text{ DIV } r$$

où r est le reste de la division 1 DIV m; et ainsi de suite. Seulement si m est rationnel on obtient effectivement un langage régulier. Cependant, le bruit dû au capteur rend illusoire l'exploitation pratique de cette structure précise.

La reconnaissance d'un segment de droite suivant ces principes constitue donc l'analyse d'un langage d'états finis dans les codes direction. L'algorithme comporte deux phases:

1) Acquisition des paramètres.

Pour trouver C1 et C2 il suffit d'analyser une barrette plus un code direction supplémentaire. On vérifie alors que C1 et C2 sont des codes voisins. La longueur k de la barrette fournit la valeur de L ou L+1 (ce qui se précisera par la suite).

2) Fonctionnement en reconnaissance suivant la formule (3).

Il faut également tenir compte de deux cas particuliers. Un segment peut être formée d'une seule barrette; et toutes les barrettes d'un segment peuvent être d'une même longueur.

3.3.5.2 Evaluation

Comparée au découpage récursif, la reconnaissance syntaxique possède des propriétés diagonalement opposées. On peut caractériser de façon beaucoup plus rigoureuse la qualité des résultats, la complexité algorithmique est définitivement plus simple - mais son utilisation pratique donne des résultats decevants.

A - Evaluation algorithmique

La reconnaissance syntaxique fonctionne comme un automate d'états finis avec une bande d'entrée contenant les points de la ligne, et une bande de sortie avec les segments du résultat. L'algorithme n'a pas besoin de stocker les points de la ligne dans un vecteur, et fournit les résultats au fur et à mesure du parcours de la ligne. C'est donc un algorithme incrémental dont l'utilisation pratique fait preuve d'une grande efficacité. Le temps d'exécution est d'ordre n non seulement en théorie, mais le nombre d'instructions effectivement nécessaire pour le traitement de chaque point est très faible.

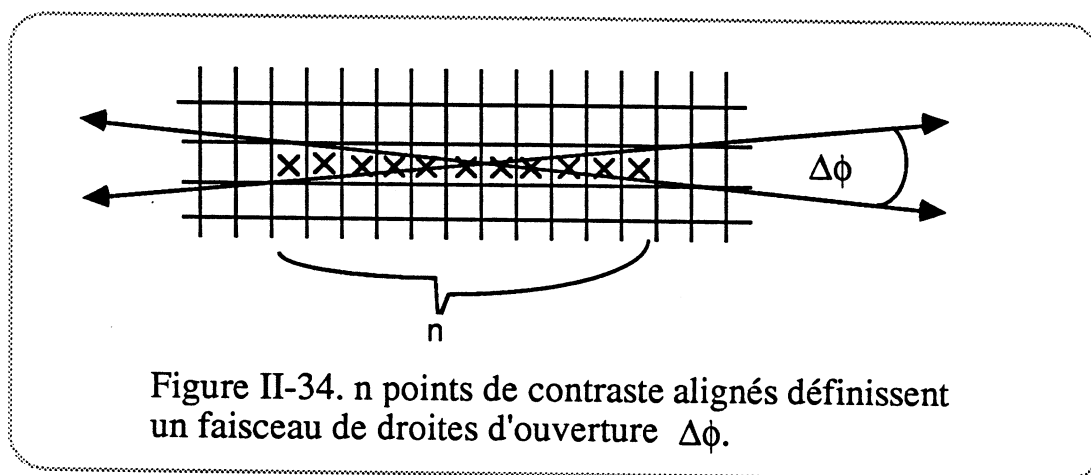


Figure II-34. n points de contraste alignés définissent un faisceau de droites d'ouverture $\Delta\phi$.

B - Evaluation des résultats

La reconnaissance syntaxique résout seulement le sous-problème A - la recherche des point de cassure. Elle s'appuie sur un modèle mathématique, les résultats ont donc des propriétés bien définies. Ce modèle nous permet également de trouver les équations des segments de droites. Cependant, une observation fondamentale s'impose: une ligne digitale finie ne définit pas un segment de droite unique, mais un faisceau de segments. Le cas le plus simple est illustré par la figure II-34 à partir de laquelle nous dérivons la marge possible des paramètres de l'équation de droite associée.

$$\Delta\phi \approx \pm 1/n$$
$$\Delta r \approx \pm 0.5$$

Cela traduit la règle générale qui veut qu'un échantillon fini (ici d'une droite) permet d'identifier un phénomène (ici l'équation de droite) avec une précision augmentant avec la longueur de l'échantillon. Il est important de noter

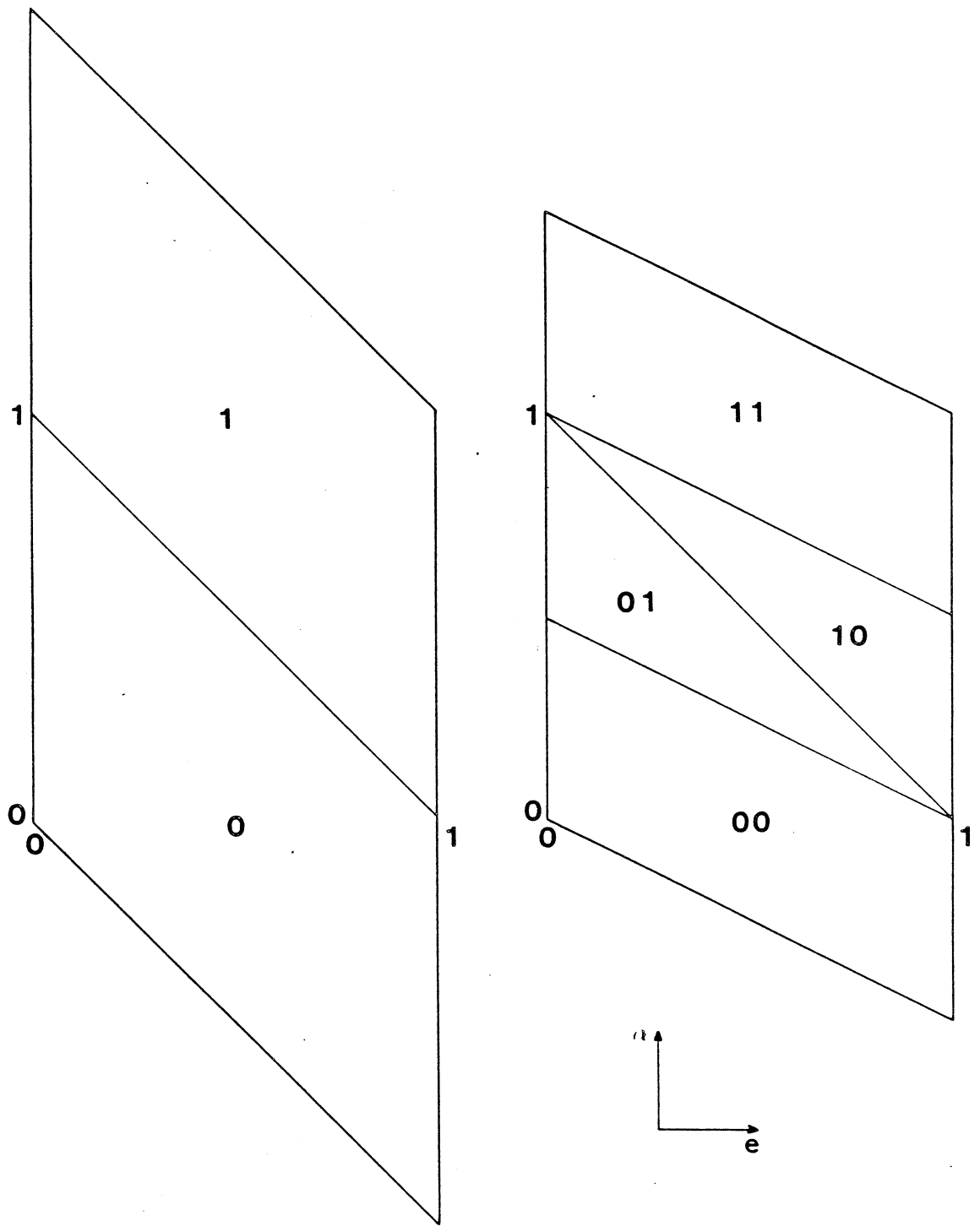


Figure II-35. Repartition de l'espace (e,α) par les suites de codes direction de longueur 1 et 2 (Dorst et Smeulders 85).

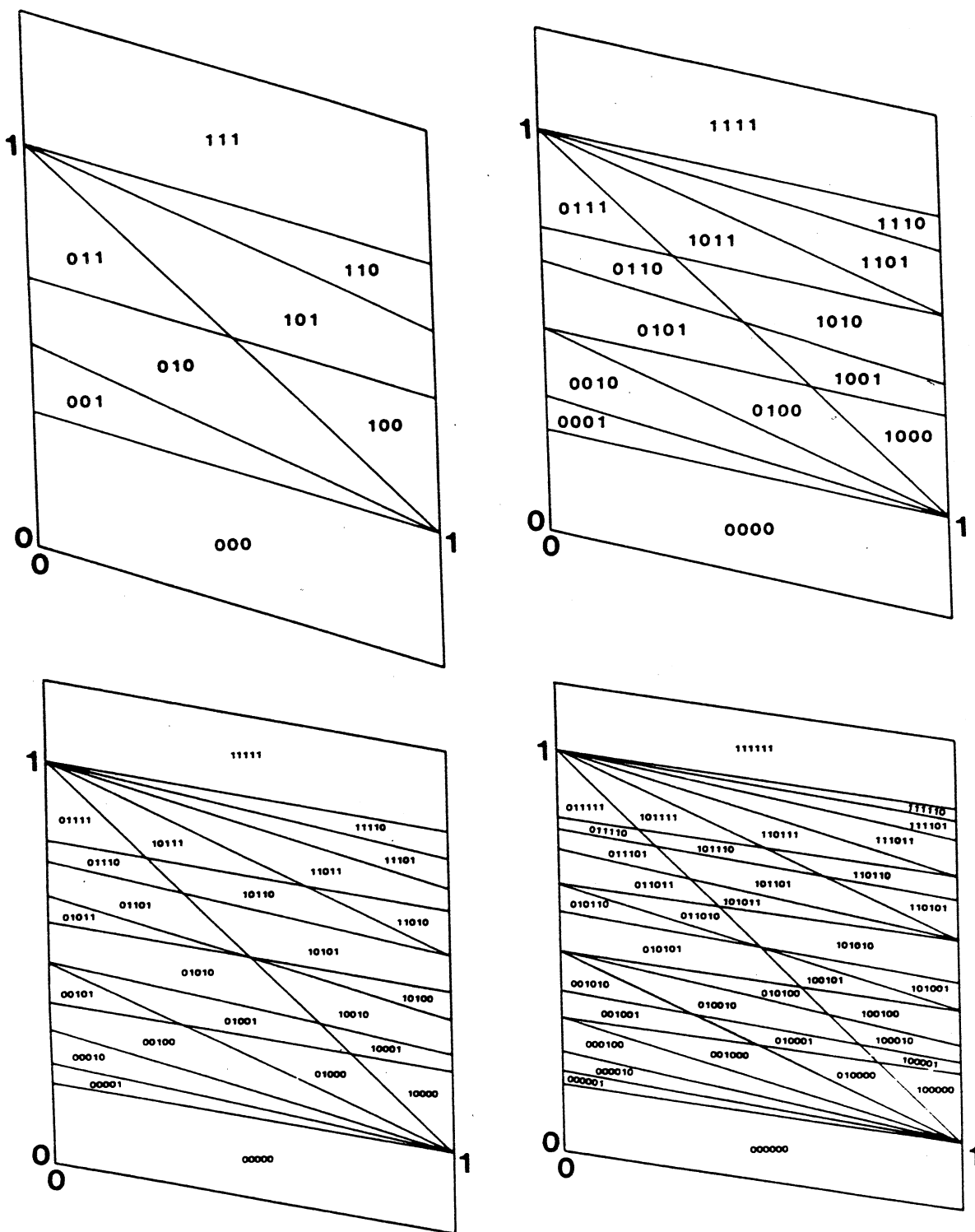


Figure II-35 (suite). Repartition de l'espace (e, α) par les suites de codes direction de longueur 3 à 6 (Dorst et Smeulders 85).

qu'aucune de ces droites n'est plus probable qu'une autre. L'utilisation d'une approximation "optimale", comme au sens des moindres carrés, donne une illusion de précision qui n'est pas toujours bien fondée, d'autant plus que dans certains cas particuliers, notamment pour des lignes courtes, le bruit d'échantillonnage n'est pas un bruit blanc.

Dans une étude fort intéressante sur les propriétés des segments de droites digitales, [Dorst et Smeulders83] dérivent les formules exactes pour l'ensemble des droites pouvant produire un mot donné dans les codes direction. Il s'avère que cet ensemble forme soit un triangle, soit un quadrilatère dans l'espace des paramètres (pente a , abscisse à l'origine e). La répartition de cet espace par les mots de un à six codes direction est reproduite dans la figure II-35.

Cependant, une telle analyse est trop fine par rapport aux qualités du matériel de prise de vue donc nous disposons. Au bruit d'échantillonnage se superpose alors un bruit d'origine physique qui rend illusoire les détails de notre analyse. Cela signifie que les images de droites physiques (d'un bord de table par exemple) ne sont en général pas conforme au modèle présenté, ce qui résulte dans une fragmentation excessive des lignes par la reconnaissance syntaxique. Il paraît donc nécessaire d'augmenter la stabilité de l'algorithme par rapport au bruit. Malheureusement il s'avère difficile d'absorber certaines formes de bruit sans sacrifier les qualités des résultats. Etant donné la vue très locale que possède cet algorithme, il se laisserait facilement piéger par des courbes à grand rayon de courbure. Cependant, nous n'avons pas mené l'étude jusqu'au bout. Nous pensons que l'utilisation simultanée de la courbure discrète permettrait de trouver des solutions intéressantes (cf. figure II-18-a et b, et paragraphe 3.3.7).

Un autre inconvénient de cet algorithme vient du fait que, considérant de nombreux cas particuliers, il y a certaines lourdeurs dans les programmes (par exemple pour la détermination des extrémités des segments).

3.3.6 Découpage itératif

3.3.6.1 Description

Les deux méthodes que nous venons de présenter de façon détaillée - le découpage récursif et la reconnaissance syntaxique - ont des qualités et des défauts complémentaires. En dehors de quelques cas exceptionnels, le découpage récursif

est peu sensible au bruit (à condition de bien choisir ses paramètres). Son inconvénient majeur vient d'un besoin en place mémoire et en temps d'exécution non-linéaire. La reconnaissance syntaxique a un caractère incrémental, donc une bonne efficacité. Par contre, elle accepte mal le bruit dû au capteur.

Nous avons donc cherché à combiner les avantages des deux méthodes et proposons l'algorithme de découpage itératif suivant qui utilise les mêmes paramètres globaux que le découpage récursif.

1) Test de linéarité.

Sur les MINLO premiers points de la ligne, on calcule la distance à la droite passant par le premier et le dernier point, comme pour le découpage récursif. Si $D_x > \text{EPSD}$ on itère sur ce test avec la ligne ($P_{\text{max}} \dots P_n$). A la sortie de cette itération on dispose d'un segment ($P_i \dots P_{i+\text{MINLO}-1}$) à partir duquel on calcule la droite D l'approchant au sens des moindres carrés.

2) Vérification.

Si les points suivants - $P_{i+\text{MINLO}} \dots$ se trouvent à une distance inférieure à EPSD de la droite D , ils font partie du segment.

La structure de cet algorithme - détermination des paramètres, puis vérification - est celle de la reconnaissance syntaxique, la première étape étant empruntée au découpage récursif. Il est incrémental, ce qui implique en particulier que la longueur des droites pouvant être détectées n'est pas bornée, et qu'il peut y avoir arrêt en fonction d'un but ou d'une contrainte. L'utilisation d'une approximation au sens des moindres carrés enlève au résultat le caractère aléatoire d'un choix des extrémités parmi les P_i .

3.3.6.2 Evaluation

L'incrémentalité de cet algorithme lui assure une complexité d'ordre n . La place nécessaire est essentiellement occupée par le vecteur des MINLO points utilisés pour le test de linéarité; grâce à ce vecteur, le découpage itératif a une vue plus globale de la ligne que la reconnaissance syntaxique, ce qui lui permet d'établir une bonne hypothèse pour la droite D . Par le choix de la taille de ce vecteur, qui définit une borne supérieure à MINLO, l'implémenteur peut sélectionner un compromis entre la globalité de l'algorithme et le coût en place.

Par construction, chaque segment de droite D trouvé par le découpage itératif a les propriétés suivantes:

- (0) D approche un segment de points ($P_{i_0} \dots P_{i_k}$)
- (1) $i_k - i_0 \geq \text{MINLO}$
- (2) D approche au sens des moindres carrés le début du segment de points ($P_{i_0} \dots P_{i_0 + \text{MINLO} - 1}$)
- (3) $\text{DIST}(P_i, D) < \text{EPSD}$ pour $i = i_0 + \text{MINLO} \dots i_k$
- (4) $\text{DIST}(P_{i_{k+1}}, D) \geq \text{EPSD}$

Il se produit donc un mélange des critères vus précédemment. Le critère P n'est pas strictement vérifié pour le début du segment, puisqu'on utilise la droite définie par les deux points ($P_{i_0}, P_{i_0 + \text{MINLO}}$) dans le test de linéarité, est non la droite D finalement retenue. Par contre, il est vrai "à un epsilon près", l'epsilon étant égal à $2 * \text{EPSD}$.

Les variantes de l'algorithme présentées dans le paragraphe suivant fournissent des segments avec des propriétés légèrement différentes.

3.3.6.3 Variantes de l'algorithme

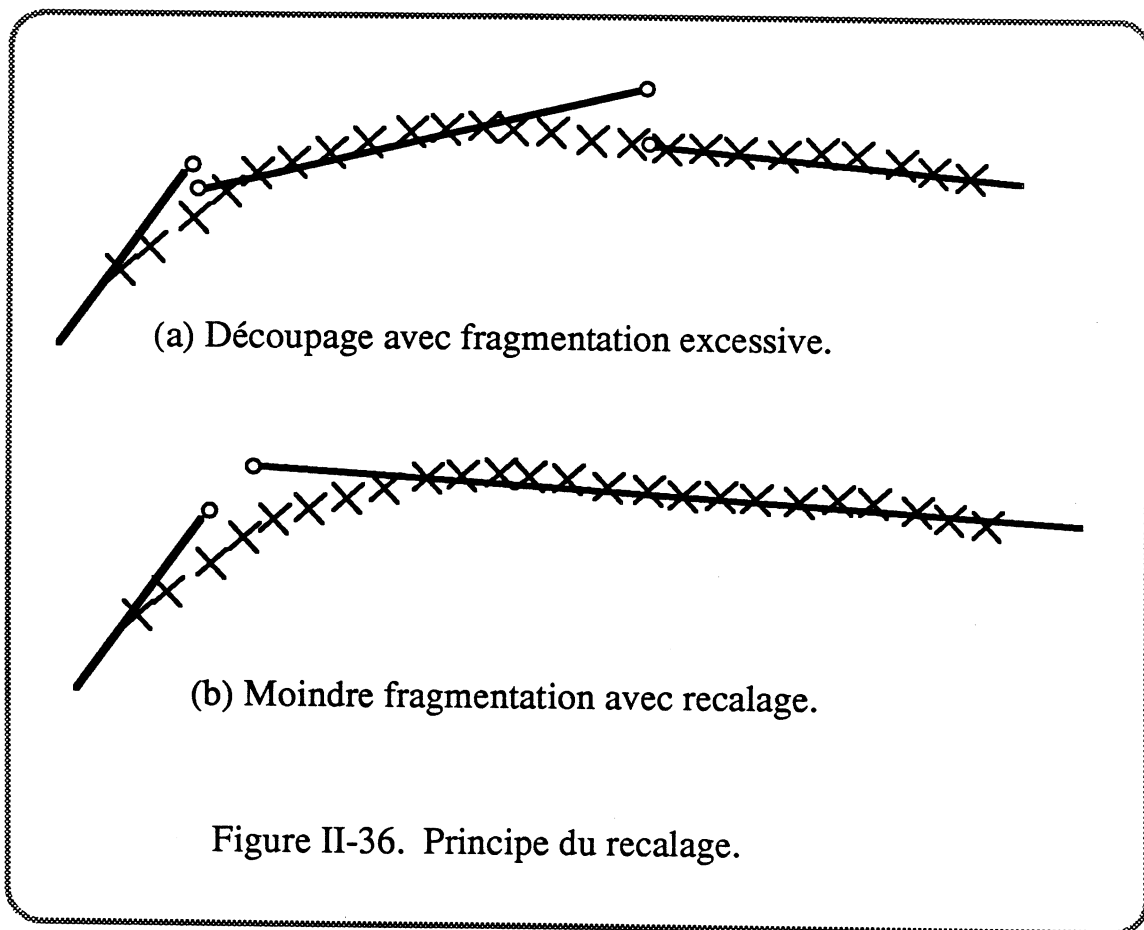
D'un point de vue pratique, le découpage itératif est intéressant parce qu'il combine les avantages des deux algorithmes précédents (découpage récursif et reconnaissance syntaxique) sans avoir leurs inconvénients:

- * Il est peu sensible au bruit et donne des résultats avec des propriétés comparables au découpage récursif.
- * Il est incrémental et possède des performances comparables à la reconnaissance syntaxique.

Au fil de l'expérimentation cet algorithme a été affiné et complété par plusieurs paramètres globaux permettant de l'ajuster plus finement aux caractéristiques d'un problème. Nous décrivons ci-après les variantes appelées recalage, centrage, et demi-tour.

A - Recalage

Si les segments de droites sont longs on constate que la droite proposée par l'étape 1) est parfois légèrement décalée par rapport à l'idéal (intuitif), ce qui amène à une fragmentation gênante des lignes (cf. figure II-36-a). Ce problème est lié à plusieurs facteurs.



* La non-linéarité d'un tube vidicon.

Avec un tube vidicon l'image d'une droite est légèrement déformée. La droite D approche bien le début de la ligne, et s'en écarte progressivement.

* La valeur de MINLO.

Si on choisit une valeur faible pour MINLO, afin de trouver également des segments courts, le calcul de D est fondé sur un nombre insuffisant de points.

Nous avons alors introduit une variation de la phase 2 de l'algorithme (vérification).

2') Recalage/vérification.

Soit le point courant P_j . On calcule la droite D' qui approche au sens des moindres carrés la totalité des points $(P_{i_0} \dots P_j)$. Si

$$\text{DIST}(P_j, D') < \text{EPSD}$$

alors P_j fait partie du segment courant.

Remarque: il suffit de recalculer D seulement lorsque P_j est trop éloigné de la droite courante.

La droite D' trouvée de cette façon possède alors des propriétés différentes de la version initiale. Notamment la propriété (2) ci-dessus est remplacée par une propriété nettement plus satisfaisante.

(2') D' approche l'ensemble des points $(P_{i_0} \dots P_{i_k})$

Par contre, on ne peut plus garantir la propriété (3) - les points du début de la ligne peuvent s'éloigner de la droite lors du recalage. Nous avons cependant mis un "garde-fou"; le test (3) est vérifié pour le point P_{i_0} après chaque recalage. Dans la pratique, ce problème ne s'est jamais présenté, et cette variante donne d'excellents résultats. En particulier, grâce à la propriété (2') les résultats deviennent plus indépendants du sens de parcours, et leur qualité indépendante de la longueur des segments. Nous remarquons que malgré l'optimalité globale du résultat l'algorithme conserve son incrémentalité.

B - Centrage

La maximalité de chaque segment est une propriété intuitivement souhaitable pour le découpage d'une ligne, parce qu'elle garantit qu'un "vrai" segment de droite n'est pas découpé en morceaux. Cependant, le critère (4) ne résoud pas vraiment le problème. Garantissant localement la maximalité à droite, il introduit un défaut que nous avons déjà constaté pour l'algorithme de Roberts: il a tendance à dépasser les bons points de cassure. En fait, la maximalité à droite implique la non-maximalité à gauche du segment suivant. La figure II-37 montre le cas le plus simple. Si un segment se termine dans un angle droit, les EPSD points au-delà de l'angle sont inclus dans le segment. Dans certaines applications, l'inclusion de ces points dans les calculs d'approximation peut sensiblement dégrader les résultats. De plus, les résultats dépendent du sens du parcours. Une analyse détaillée de cette situation paraissant difficile à justifier, nous avons décidé d'ignorer, pour les calculs d'approximation, un nombre fixe de points aux extrémités des segments, précisé par un paramètre global LIM. La formule (2') obtenue dans la variante avec recalage devient alors

(2'') D' approche $(P_{i_0+LIM} \dots P_{i_k-LIM})$

Il faut bien entendu que $\text{MINLO} > 2 * \text{LIM} + 2$.

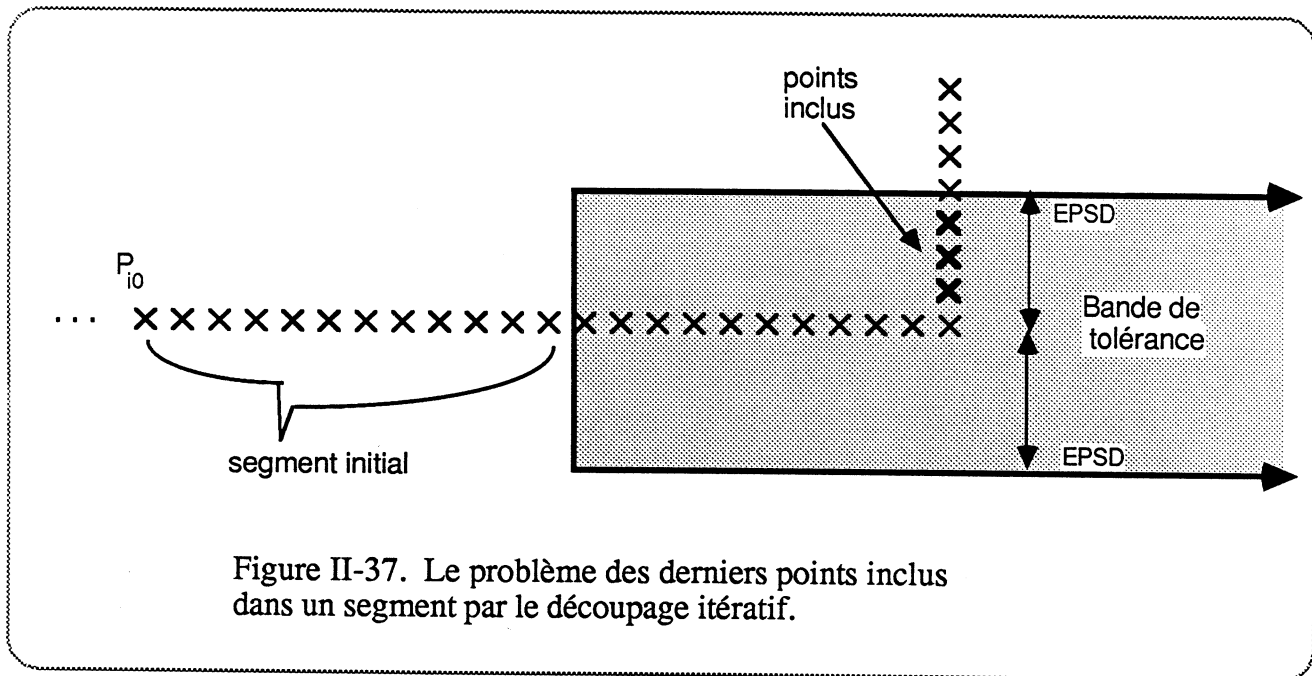


Figure II-37. Le problème des derniers points inclus dans un segment par le découpage itératif.

Le centrage soulève un problème annexe qui se présente parfois dans le contexte de l'extraction de segments de droite: la localisation précise des extrémités. Ce problème est résolu d'une façon satisfaisante par le découpage récursif, puisque nous avons pu montrer qu'il casse un train polygonal aux angles dans des conditions assez générales. La situation est plus complexe lorsqu'on utilise un calcul d'approximation pour trouver les équations des droites. Dans ce cas ni le point P_{i_0} ni le point P_{i_k} ne se trouvent en général sur la droite. Si les données représentent un train polygonal, la solution la plus simple est aussi la meilleure: définir les extrémités par les intersections de deux droites consécutives. Dans ce cas il est préférable de choisir une valeur assez forte pour LIM.

C - Demi-tour

Dans la variante avec demi-tour, la phase 2 de l'algorithme se présente de la façon suivante.

2a) Vérification à droite.

Si les points $P_{i+MINLO}, \dots$ se trouvent à une distance inférieure à EPSD de la droite D , ils font partie du segment. C'est la phase 2 de l'algorithme de base.

2b) Vérification à gauche.

Si les points P_{i-1}, P_{i-2}, \dots se trouvent à une distance inférieure à EPSD de la droite D , ils font partie du segment.

Si la recherche de segments de droite se déroule en co-routine avec le suivi de ligne, l'étape 2b correspond à un abandon du suivi de l'étape 2a, afin de suivre "dans l'autre sens" à partir du point initial.

Cette variante est intéressante dans différentes conditions:

- * lorsque le point initial d'un suivi peut tomber au milieu d'un segment de droite, celui-ci peut être recouvert en entier. Ce cas se présente par exemple dans le projet sur le monde des blocs décrit au paragraphe 4.4.
- * lorsqu'on recherche un segment de droite dans le voisinage d'un arc à faible courbure, on peut effectuer le test de linéarité (phase 1) avec une valeur très sévère de EPSD, afin de localiser une partie de la droite avec certitude, puis la prolonger des deux côtés (avec éventuellement une autre valeur de EPSD). Une application de ce type est présentée au paragraphe 3.2.9.3.
- * la possibilité d'extension des deux cotés permet d'envisager une extraction systématique de droites à partir de points de contraste dans un balayage de type vidéo de l'image. Cette possibilité est intéressante dans la perspective d'un traitement d'image en temps réel avec une structure en pipeline, dans laquelle un processeur "extraction systématique de droites" fonctionne en aval d'un processeur de type GTR.

3.3.7 Analyse de courbure

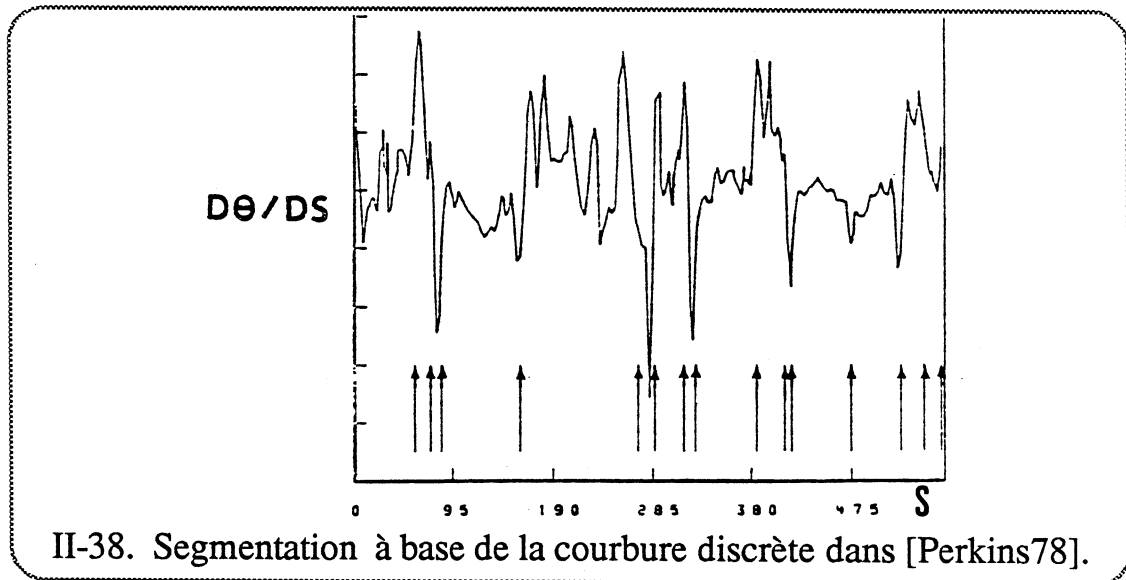
Avec la discussion de la courbure discrète (§3.2) nous avons mentionné la possibilité de l'utiliser pour l'extraction de segments de droites. D'après la figure II-18, on peut construire un algorithme selon les principes de la reconnaissance syntaxique, ou tout simplement procéder à un seuillage de la courbure discrète. Ces algorithmes ont l'avantage d'être incrémentaux. Le seuillage accepterait cependant des arcs à faible courbure, et présente plus généralement le défaut de ne pas garantir le critère P, autrement dit on ne peut pas facilement borner

$$\max_i \text{DIST}(P_i, D).$$

Ce défaut est intrinsèque à l'approche par la courbure, sa correction demande la prise en compte d'autres informations, comme la donnée des points P_i .

En ce qui concerne la courbure discrète, ce type d'algorithme se justifie seulement si on dispose de la courbure pour d'autres raisons, sinon son calcul est trop coûteux en comparaison avec le découpage itératif.

Un exemple intéressant est donné par le système de Perkins, décrit au paragraphe I-2.3.8.1, qui utilise une autre forme de courbure pour la recherche de segments de droite et d'arcs de cercle. Perkins a procédé comme suit: utilisant l'opérateur de Hueckel, il dispose de la direction de la tangente en chaque point. Il définit la courbure comme étant le changement de tangente d'un point à l'autre. Les pics dans cette courbure correspondent alors aux points de cassure (problème A), les droites se distinguent par une tangente à peu près constante (figure II-38).



Le sous-problème A étant résolu, les équations de droites peuvent être basées simplement sur les points de cassure, ou obtenues par une approximation au sens des moindres carrés.

3.3.8 La méthode de Hough

La méthode de Hough traite un problème sensiblement plus général que celui que nous nous sommes posés: l'ordre des points n'est pas pris en compte, les P_i ne constituent pas nécessairement une ligne de contraste. La méthode de Hough permet donc en particulier d'identifier des droites "en pointillé", dont quelques points sont visibles à des endroits dispersés.

Principe

On considère que chaque point P_i définit un ensemble de droites E_i passant par ce point. Si plusieurs points P_i sont colinéaires, les ensembles E_i ont une intersection non vide, qui est la droite passant par ces points. La méthode de Hough construit explicitement tous les ensembles E_i pour les points $(P_0 \dots P_n)$, et cherche les intersections non-vides des E_i .

La réalisation de cette méthode est compliquée par le fait que l'ensemble E_i est infini - il faut le discrétiser. Pour cela on définit un "tableau d'accumulation", dont les dimensions correspondent aux deux paramètres d'une droite passant par un point (x_i, y_i)

$$x_i \cos(\varphi) + y_i \sin(\varphi) + r = 0$$

Les cases du tableau sont toutes initialisées à 0. Pour un point donné, l'ensemble des valeurs (φ, r) définissant E_i correspond à une courbe sinusoidale dans le tableau d'accumulation. Toutes ces cases sont alors incrémentées de 1 - le point P_i "vote" pour ses droites. Après traitement de tous les points, une case portant la valeur m témoigne de m points colinéaires; l'équation de la droite est donnée par les coordonnées de la case dans le tableau d'accumulation.

Evaluation

Cette méthode possède des propriétés très différentes des algorithmes précédents. Elle présente aussi un certain nombre de problèmes majeurs, liés essentiellement à la discrétisation de l'espace des paramètres au moyen du tableau d'accumulation

- 1) La précision avec laquelle une droite est définie dépend du système de coordonnées et de la taille de l'image. En effet, à une case du tableau d'accumulation correspond une famille de droites

$$x \cos(\varphi + \Delta\varphi) + y \sin(\varphi + \Delta\varphi) + (r + \Delta r) = 0$$

qui a la forme d'un faisceau. La distance maximale d'un point à la droite pour laquelle il a voté est donnée par la plus grande ouverture du faisceau. Cette valeur est donnée par la formule

$$\Delta\varphi * \text{distance de l'origine}$$

qui est minimale si l'origine du repère est au milieu de l'image. De plus, cette distance n'est pas uniforme dans toute l'image, elle se dégrade vers les bords.

- 2) La précision des résultats augmente avec la taille du tableau d'accumulation, alors que la plausibilité des droites résultats diminue avec cette taille: d'après la première remarque, si l'on veut améliorer la précision des résultats, il faut augmenter la taille du tableau d'accumulation. Dans ce cas, le poids accumulé dans chaque case sera plus faible, l'évidence pour une droite moins bonne. C'est une forme de relation d'incertitude qui impose des limites à la qualité des résultats.
- 3) L'algorithme est onéreux en place, d'autant plus qu'il peut être utile de garder une référence de chaque point ayant voté pour une case pour des calculs ultérieurs. Il est également onéreux en temps, puisque pour chaque point il faut calculer l'ensemble des cases du tableau d'accumulation concernées.

Les avantages de la méthode de Hough sont également nombreux.

- 1) En cumulant l'évidence pour une droite, elle permet de détecter des droites dans des conditions difficiles, où elles n'apparaissent nettement nulle part.
- 2) Les points P_i étant traités indépendamment les uns des autres, elle permet un fort degré de parallélisme que l'on peut exploiter avec un matériel spécialisé.
- 3) Elle se généralise directement à d'autres types de courbes paramétrisées.
- 4) Elle permet de tenir compte de contraintes sur les paramètres, qui se traduisent par une réduction de la taille du tableau d'accumulation, et du temps d'exécution.

Les propriétés de la transformée de Hough sont complexes, comme le montre le travail de [Horain 84]. La mise en œuvre de cette méthode demande une étude soignée; pour cette raison son utilisation a donné lieu à de nombreuses recherches qui visent à trouver un compromis intéressant parmi toutes ces propriétés.

3.3.9 Applications

Nous allons par une brève analyse de trois exemples choisis dans des applications industrielles illustrer l'utilité des discussions des paragraphes précédents.

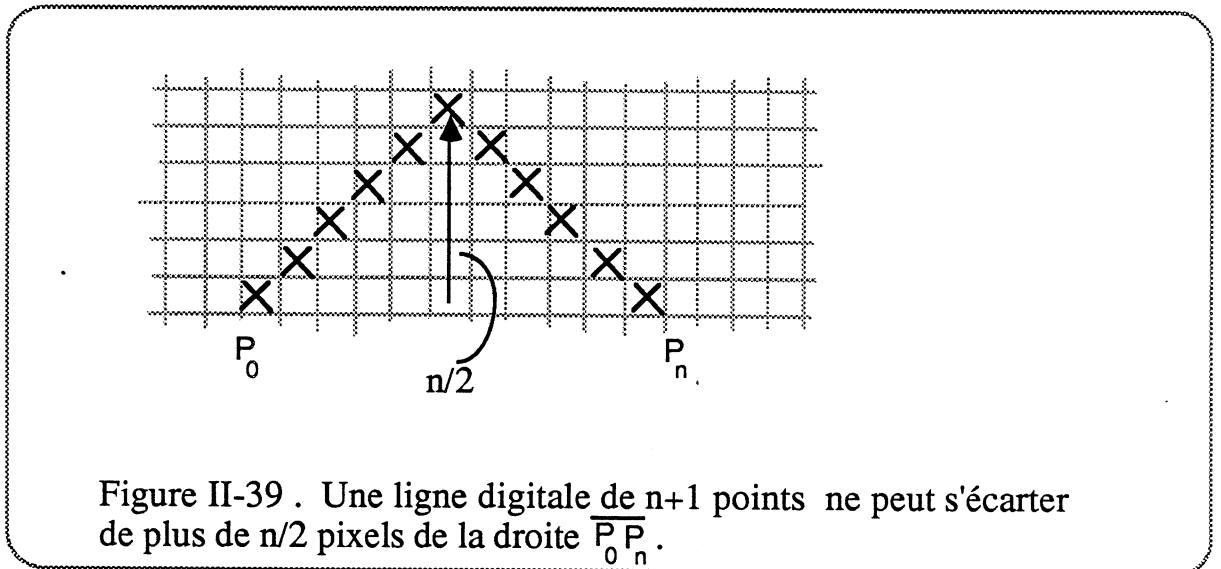
3.3.9.1 Modélisation de formes

Dans de nombreux domaines dans lesquels il faut découper une matière première (textile, cuir, bois) se pose le problème d'enregistrer une forme bidimensionnelle (le contour d'un patron) qui doit approcher la forme initiale selon le critère P. Par rapport à l'enregistrement point par point des coordonnées du contour, cette méthode présente plusieurs avantages:

- * Les données sont directement utilisables pour la commandes d'une machine de découpe.
- * Il y a lissage par rapport au données brutes des points du contour.
- * On réalise une réduction sensible de la quantité d'informations.

Ce problème peut être résolu par l'algorithme de découpage itératif aussi bien que par l'algorithme de découpage récursif; pour ce dernier il faut cependant que ne se pose pas le problème du Z. Pour les deux algorithmes, le choix du paramètre EPSD est imposé par l'énoncé du problème, et le paramètre MINLO s'en déduit, étant donné que toute partie de la ligne doit être approchée, c.a.d. il ne faut pas qu'il y ait des trous dans le découpage. Pour cela il suffit de poser (cf. figure II-39)

$$\text{MINLO} = 2 * \text{EPSD} + 1$$



Nous remarquons que dans ces conditions les zig-zags peuvent poser un problème (figure II-25); les formes modélisées ne doivent donc pas faire apparaître des trop hautes fréquences spatiales.

Par contre, les autres algorithmes que nous avons présentés sont mal adaptés à ce problème: la reconnaissance syntaxique ne fonctionne que pour les contours polygonaux, ni un découpage basé sur la courbure ni l'algorithme de Hough ne garantit le critère P.

3.3.9.2 Inspection de vis

L'inspection de vis a notamment fait l'objet d'une étude par Patrice Horaud [Horaud 81]. Nous utilisons cet exemple pour montrer l'intérêt d'une analyse des temps d'exécution.

La tâche se présente comme suit: dans l'image du filet d'une vis compter le nombre de pas de vis. Elle sera résolue par l'analyse d'une approximation polygonale du filet. Le découpage en segments de droites du contour du filet est l'étape critique du traitement.

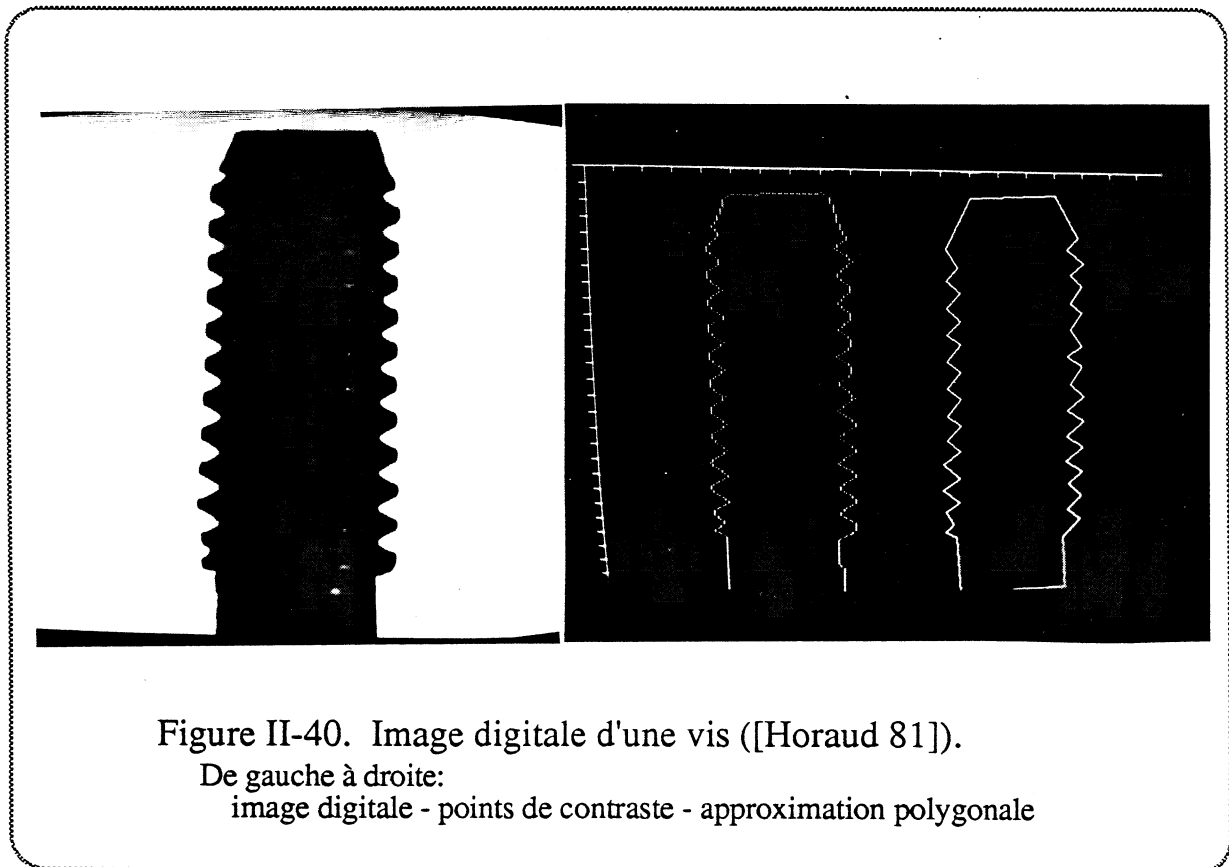


Figure II-40. Image digitale d'une vis ([Horaud 81]).

De gauche à droite:

image digitale - points de contraste - approximation polygonale

Nous allons comparer les découpages récursif et itératif par rapport à ce problème. Le choix des paramètres - notamment de EPSD, MINLO - ne présente pas de difficulté particulière, et

les deux algorithmes délivrent des résultats acceptables. Nous nous intéressons alors plus spécialement aux temps d'exécution, sachant que l'ordre de grandeur fourni par l'évaluation générale des paragraphes précédents correspond à un comportement asymptotique et ne permet pas forcément de conclure dans un cas concret, où il faut tenir compte de divers coefficients constants.

Comme nous avons vu (§3.3.4.3), les dents de scie de chaque moitié du filet correspondent au cas le plus défavorable du découpage récursif, avec un temps d'exécution de la forme

$$T_r = r_1 + r_2 * ND + r_3 * (n * (ND - 1) / 2)$$

où ND est le nombre de segments, et les trois constantes r_1, r_2, r_3 s'interprètent comme suit:

- r_1 Coût d'appel et d'initialisation de la fonction
- r_2 Coût de l'initialisation d'un segment de droite
- r_3 Coût du test de distance pour chaque point.

Pour l'algorithme de découpage itératif nous avons

$$T_i = i_1 + i_2 * ND + i_3 * n$$

Il est raisonnable de considérer que r_1, i_1 et r_2, i_2 sont sensiblement égaux, et que leurs contributions au temps total sont faibles, puisque n devrait être supérieur à 100 dans le cas d'une image 256 par 256. Nous négligerons donc les deux premiers termes de T_r et T_i .

En ce qui concerne le troisième coefficient, si nous utilisons le découpage itératif avec une approximation aux moindres carrés, la valeur de i_3 inclut, en plus d'un calcul de distance (valant 2 multiplications et trois additions, et le contrôle de l'itération), la mise à jour des formules de la fin du paragraphe 3.3.3, soit 3 multiplications et 5 additions, et nous supposons (cf. remarque à la fin de ce paragraphe)

$$i_3 \approx 2 * r_3$$

Dans le cas des vis traitées par Horaud (figure II-40), le nombre de segments de droites par demi-filet est $ND=26$, et on obtient

$$\begin{aligned} T_r &\approx 12.5 * r_3 * n \\ T_i &\approx 2 * r_3 * n \end{aligned}$$

Pour l'ensemble du filet de la vis, considérant que le premier point de cassure correspond à la pointe, on trouve

$$\begin{aligned} T_r' &= r_3 * (2 * n) + 2 * T_r \approx 27 * r_3 * n \\ T_i' &= 2 * T_i \approx 4 * r_3 * n \end{aligned}$$

On constate donc que malgré des calculs plus précis le découpage itératif est environ six fois plus rapide que le découpage récursif.

Bien entendu, un tel résultat doit être vu en relation avec l'ensemble des programmes d'une application. Si le suivi de ligne effectue des calculs de gradient par logiciel, la différence globale sera minime! Toutefois, en milieu industriel, les calculs des points de contraste devraient être réalisés par un matériel spécialisé, et la différence entre les deux algorithmes sera sensible.

Sachant que quelques variations faibles dans les hauteurs respectives des dents du filet suffisent à "apprivoiser" le temps défavorable du découpage récursif il est aussi intéressant d'analyser le cas le plus favorable. On obtient alors les temps

$$\begin{aligned} T_r &\approx r_3 * n * \log_2(ND) \approx 5 * r_3 * n \\ T'_r &\approx 12 * r_3 * n. \end{aligned}$$

L'avantage du découpage itératif est plus faible, mais toujours réel.

Finalement, nous remarquons que le surcoût en temps de calcul dû à l'approximation des moindres carrés fournit en contrepartie une qualité supplémentaire des résultats qui nous semble utile. Il serait plus juste pour une pure comparaison des temps d'exécution, de considérer $i_3 \approx r_3$, ce qui ajouterait un facteur deux en faveur du découpage itératif.

3.3.9.3 Le disque avec méplat

Le repérage de position d'une plaque de silicium découpée d'un côté nous fournit l'énoncé d'un problème d'une simplicité exemplaire (figure II-41): localiser un unique segment de droite formant le "méplat" d'un cercle. La difficulté vient de ce que les dimensions de la figure dans l'image donnent une grande influence au bruit de discrétisation.

Pour que le méplat se distingue dans l'image digitale, il faut qu'il soit plus long que les "faux méplats" provenant de l'échantillonnage spatial du cercle (figure II-42). Etant donné les dimensions du disque cela implique

$$LM = r/3 \gg 2*\sqrt{2*r} \quad (1)$$

ou mieux, si on admet $h=2$ dans les formules de la figure II-42

$$LM = r/3 \gg 2*\sqrt{4*r} \quad (2)$$

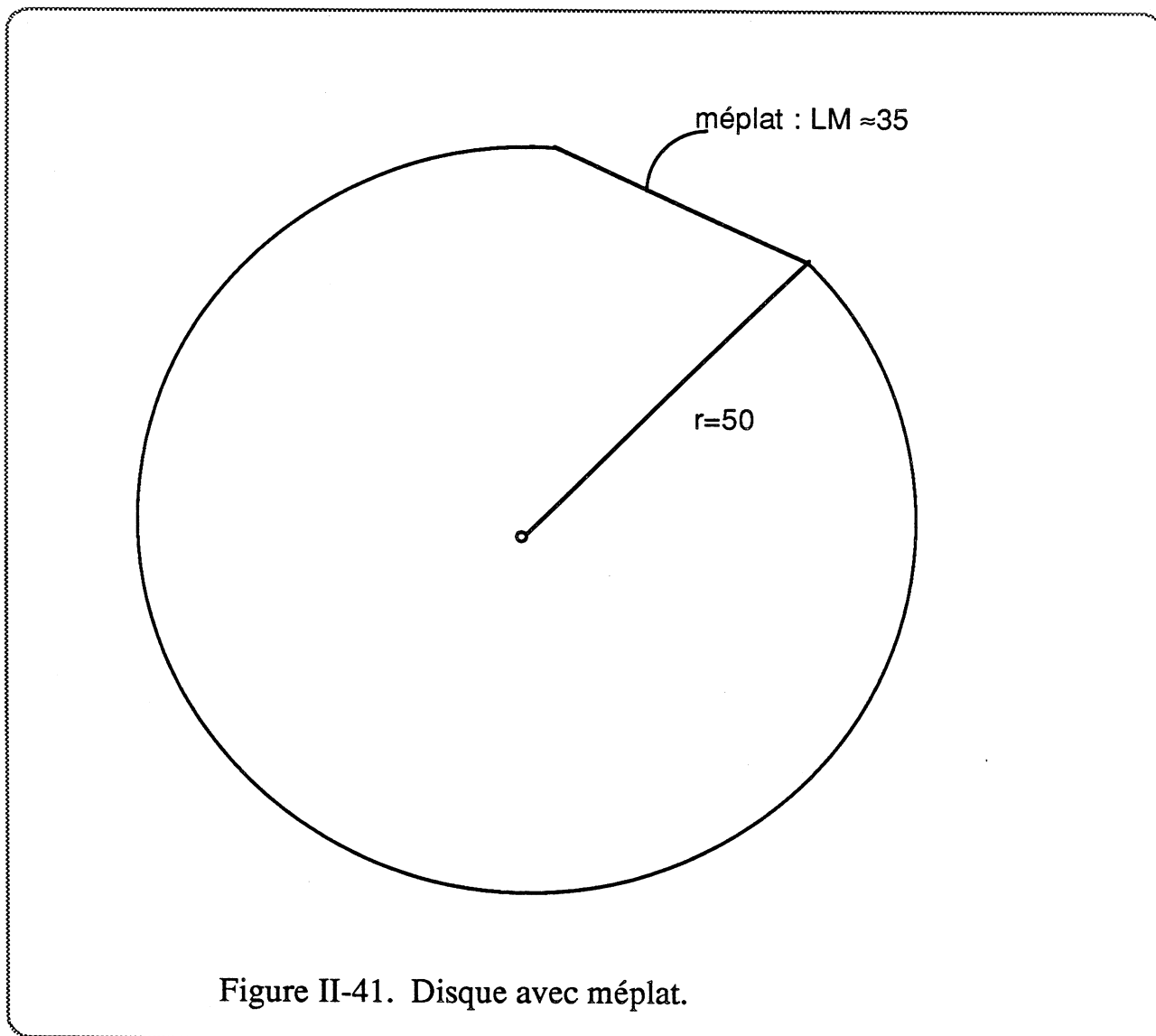


Figure II-41. Disque avec méplat.

La formule (1) est équivalente à

$$r*(r-72) \gg 0$$

La fonction $r*(r-72)$ étant strictement croissante avec r (ce qui signifie que plus le pas d'échantillonnage d'un cercle est fin, plus les "faux méplats" paraîtront courts par rapport aux vrais), cela nous définit la résolution minimale permettant de résoudre le problème. Il faut que

$$r \gg 72 \text{ pixels}$$

Partant de la formule (2), plus réaliste, on obtient

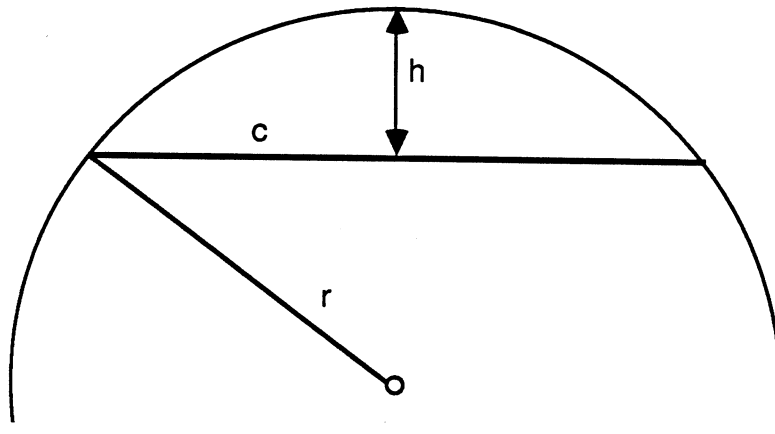


Figure II-42-a. Longueur c d'une corde de hauteur h dans un cercle ($h < r$):

$$c = 2 \sqrt{2hr - h^2}$$

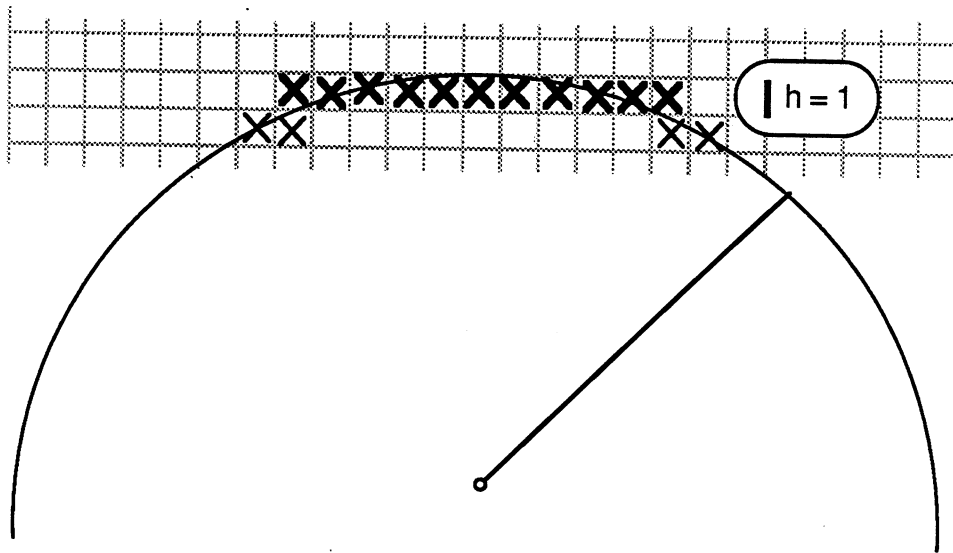


Figure II-42-b. Avec les principes du §3.3.5, la digitalisation d'un cercle donne des faux méplats de longueur $2 \sqrt{2r}$. Le découpage récursif avec $EP_{SD}=1$ risque de donner des droites de longueur $2\sqrt{4r}$, si $r \gg h$. (La figure montre un seul des faux méplats).

r >> 144 pixels.

La résolution standard pour les applications industrielles étant 256 par 256, nous avons forcément $r < 128$.

Quoique les formules de la figure II-42 puissent sembler pessimistes par rapport aux résultats trouvés avec une bonne caméra (ce que notre expérimentation dément), il est clair qu'avec une résolution de 256 par 256 on se trouve à la limite des possibilités du signal. De ce fait quelques simplifications que nous avons jusqu'à maintenant passées sous silence ne sont plus acceptables et doivent être revues.

- * Nous avons exprimé la longueur MINLO en pixels, alors qu'un nombre donné de pixels colinéaires peut correspondre à des distances différentes suivant l'orientation.
- * Les calculs de distance sont biaisés par le fait que pour la plupart des capteurs le repère image n'est pas orthonormé. Par exemple, le pixel du standard télévision possède une forme rectangulaire de proportion largeur/hauteur = 4/3.

Nous avons trouvé que la variante de l'algorithme de découpage itératif avec demi-tour et centrage représente une solution acceptable de ce problème, et donne en pratique des résultats satisfaisants. On choisit une valeur très stricte pour EPSD - 1 - afin d'assurer que le test de linéarité trouve un segment entièrement localisé sur le méplat; cette condition détermine également la valeur de MINLO = LM/2. Les deux phases de vérification garantissent la symétrie gauche/droite de la recherche, et les faux méplats sont éliminés en fonction de la longueur du segment final.

Cependant, nous pensons que sur le plan du principe l'algorithme de reconnaissance syntaxique est mieux adapté. Dans la mesure où il tient explicitement compte du bruit d'échantillonnage il exploite mieux que les autres algorithmes l'information contenue dans la ligne à découper. Toutefois ce potentiel est réalisable seulement en absence d'autres bruits, ce qui est à vérifier avec chaque nouveau matériel, et demanderait un important effort pour la mise en œuvre.

3.10 CONCLUSION

Dans ce paragraphe nous avons cherché à montrer l'importance d'une analyse rigoureuse des résultats fournis par un algorithme de découpage d'une ligne, et nous avons identifié plusieurs critères utiles à une telle analyse.

D'autres critères sont possibles:

- * [Wall et Danielsson84] minimisent l'aire entre chaque segment de droite et la ligne $[P_{i_0} \dots P_{i_k}]$. Il s'avère que cette aire peut être calculée de façon incrémentale par un algorithme proche de celui de Roberts, avec une très faible quantité de calculs - seulement six multiplications, et trois dans une version simplifiée. C'est l'algorithme d'extraction de segments de droites le plus rapide. Il a comme inconvénient de dépasser les angles (il coupe selon un critère cumulatif), et de fournir des segments ayant des points P_i comme extrémités.
- * [Pavlidis et Horowitz74] et [Kurozumi et Davis81] proposent de minimiser globalement le nombre de segments de droites produits par le découpage. Ce critère ne semble pas bien choisi, puisqu'il conduit à des algorithmes globaux et coûteux sans pour autant simplifier l'interprétation ultérieure.

Malgré sa longueur, ce paragraphe est loin d'épuiser le sujet - on trouve encore de nombreux algorithmes dans la littérature, par exemple [Ramer72], [Tomek74], [Sklansky et Gonzales80], [Williams 81], [Dettori82].

D'autre part, les segments de droites ne sont pas le seul type d'indice visuel présent dans une ligne; que peut-on déduire de cette analyse pour l'extraction d'autres types de courbes, notamment pour les arcs de cercles et les arcs d'ellipses?

Dans tous les cas on doit résoudre les deux sous-problèmes qui sont la détermination des points de cassure et le calcul des paramètres de la courbe, le premier étant toujours plus difficile.

Pour les arcs de cercles, l'analyse de la courbure - cf. §3.2 et 3.3.7 - donne des résultats corrects. Elle a notamment été employée par [Perkins78]. L'approximation aux moindres carrés peut être réalisée de façon incrémentale, quoiqu'avec des calculs nettement plus lourds, étant donné qu'il faut résoudre un système de 3 équations non-linéaires. Par contre, la géométrie digitale du cercle est un sujet plus difficile, cf. [Nakamura et Aizawa84].

Cependant, les arcs de cercles ne sont pas un IV très intéressant, puisqu'ils se déforment en arcs d'ellipses dans des conditions de prise de vue générales. Or, le calcul d'arcs d'ellipses approchant un nuage de points est un problème difficile. Selon l'étude de [Agin 81] une bonne qualité d'approximation coûte chère en calculs (plusieurs dizaines de secondes sur un DEC 2060).

Parmi les algorithmes analysés dans ce paragraphe, seule la méthode de Hough se généralise naturellement à la recherche de tout type de courbe, et de nombreuses recherches en ce sens ont été effectuées, comme par exemple [Muller et Mirgoux 84] pour la recherche de quadriques; on trouvera d'autres références dans [Horain 84].

Nous pensons que la recherche de courbes peut être sensiblement simplifiée si l'on part d'une approximation polygonale d'une ligne. Dans ce cas, d'une part le nombre de données est fortement réduit, d'autre part les propriétés de l'approximation polygonale donnent une première caractérisation utile de la forme de la ligne. Dans ce contexte, le test de linéarité (pas 1 du découpage itératif) peut fournir des informations supplémentaires, comme la convexité d'un segment.

CHAPITRE 4

LE SYSTEME CAIMAN - REALISATION ET APPLICATIONS

Le chapitre 4 présente un système logiciel nommé CAIMAN - système Conversationnel d'Analyse d'IMAgés Numérisées - qui intègre l'ensemble des algorithmes que nous avons présentés dans les chapitres précédents dans un environnement complet pour la VO. CAIMAN est un système de vision de bas niveau avec une structure de contrôle flexible et pouvant dialoguer avec un programme d'interprétation. Les principes généraux et le contenu algorithmique du système étant décrits dans les chapitres précédents, la description que nous donnons dans les paragraphes 4.1 à 4.3 reste brève et concerne les aspects importants de la réalisation du système. Une description détaillée des différentes versions est consignée dans les manuels d'utilisation - [Lux 83a, ITMI 83, Gordillo 85] et un manuel de programmation [Lux 83b]. L'ambition du système CAIMAN étant l'utilité pratique, nous décrivons dans les paragraphes 4.4 à 4.6 trois projets d'application qu'il a permis de réaliser dans notre équipe. CAIMAN est également à la base du système PVV présenté au paragraphe III-2.3.

4.1 Présentation

Le système CAIMAN a l'ambition de mettre à la disposition du constructeur d'un système de vision un outil de base puissant et flexible, lui permettant de se concentrer sur des problèmes d'interprétation spécifiques à son application. Grâce à une structure interne extrêmement souple, CAIMAN est capable de remplir plusieurs rôles différents dans la recherche et dans le développement d'une application en VO:

- 1) CAIMAN est un système de développement d'algorithmes. Son utilisation interactive permet l'expérimentation, sur un type d'image donné, des nombreux algorithmes disponibles, afin de définir de nouveaux algorithmes de traitement.

- 2) CAIMAN est une bibliothèque de programmes permettant la mise en oeuvre rapide d'une application en VO. Son caractère modulaire permet de choisir les éléments nécessaires à une application afin de construire un programme de taille réduite.
- 3) CAIMAN définit une machine spécialisée dans l'analyse d'images de bas niveau. La possibilité de connexion avec un programme d'interprétation sur un autre ordinateur permet de réaliser des systèmes de VO utilisant des processeurs travaillant en parallèle.

4.2 COMPOSANTS D'UN SYSTEME INTERACTIF

La VO comportant un aspect expérimental essentiel, l'objectif premier du système CAIMAN vise à fournir un environnement matériel et logiciel facilitant l'expérimentation. Un système interactif est le mieux adapté à cet objectif, dans lequel l'utilisateur peut formuler une commande, observer l'exécution, et formuler la commande suivante en fonction des résultats.

4.2.1 Aspect matériel

Etant donné qu'une station de travail en VO est constituée de nombreux éléments matériels à haute technologie, la facilité d'accès aux possibilités du matériel est un aspect important d'un environnement expérimental pour la VO. Le système CAIMAN inclut des commandes adressant les fonctions élémentaires de ce matériel. Ces fonctions concernent essentiellement

- * la prise de vue et la numérisation d'images
- * les affichages graphiques.

Un souci majeur lors de la réalisation de CAIMAN a été la transportabilité du système. Il a été implanté dans des contextes matériels très variés, dérivés de deux configurations type:

- * Version LSI-11 avec Grinnell.
Dans la version initiale au LIFIA, le système résulte de la connexion d'un micro-ordinateur LSI-11/23 (sous RT-11) avec un système Grinnell GRT-27. Le système Grinnell assure alors les fonctions de conversion et de stockage d'une image en noir et blanc, et fournit les primitives d'affichage graphique. L'ensemble des traitements algorithmiques est réalisé sur le LSI-11/23.

* Version LSI-11 avec GTR.

Le GTR - Gradient Temps Réel - est un processeur spécialisé développé à l'ENSERG fournissant en temps réel une image de points de contraste. En comparaison avec la configuration précédente, le GTR remplace à la fois les fonctions prise de vue et numérisation d'images du GRINNELL, et le module logiciel calculant le gradient. L'affichage graphique est assuré par une console de type SEKAPA, et l'ensemble du logiciel est implanté sur un LSI-11/23.

Le première version a été adaptée à un GRINNELL GMR 274 couleur, la deuxième est aussi opérationnelle sur ordinateur HP1000. Le logiciel, entièrement écrit en FORTRAN-IV, a en partie été développé sur un SOLAR-65 et sur HB-68 avec MULTICS.

4.2.2 Traitement et interprétation d'images

Nous avons vu à plusieurs reprises que l'interprétation d'une image passe par l'élaboration de multiples niveaux de structures de données intermédiaires. Les niveaux présents dans le système CAIMAN sont décrits par la figure II-43. Des commandes permettent d'activer les opérateurs aux différents niveaux de cette structure. On peut, par exemple

- * commander une prise de vue
- * inspecter les pixels de l'image
- * calculer le gradient d'un pixel
- * suivre une ligne de contraste
- * chercher les segments de droites dans une ligne.

Ainsi on peut observer le comportement des opérateurs à chaque niveau, avec une quantité de traces modulable, et les enchaîner explicitement dans une série de commandes. Ce type d'expérimentation permet

* La formulation de nouveaux algorithmes.

Le système possédant une structure ouverte, il est également aisé d'y inclure un nouvel algorithme, complétant l'ensemble des commandes. En particulier, l'état actuel du système est le résultat d'une expérience longue de cinq années pendant lesquelles de très nombreux algorithmes ont été testés pour en retenir les plus intéressants.

* L'étude de faisabilité d'une application simple.

Les éléments difficiles à évaluer dans un grand nombre d'applications simples de la VO se situent dans le bas niveau. Pour établir la faisabilité, il s'agit alors d'expérimenter avec différentes conditions de prise de

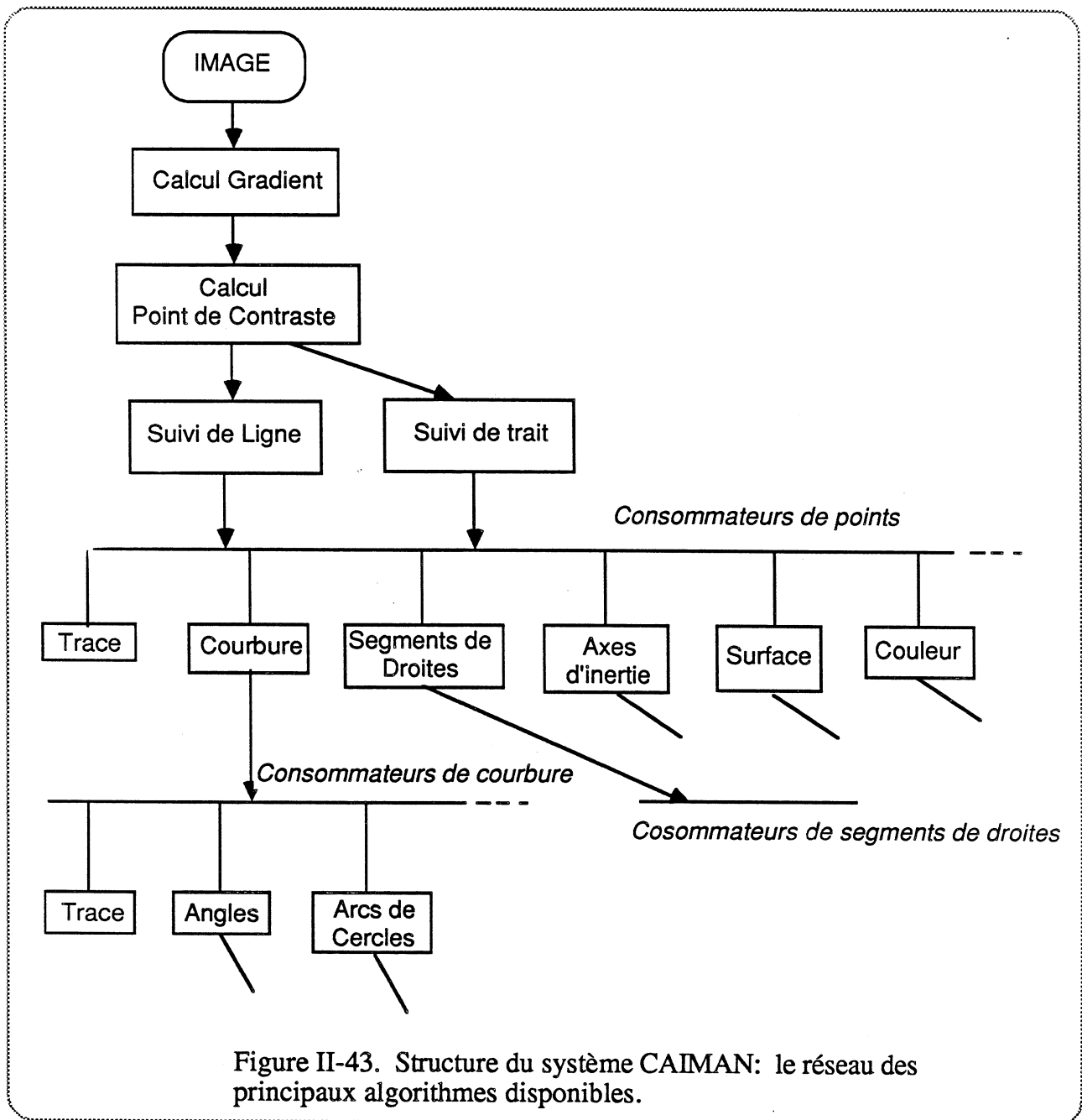


Figure II-43. Structure du système CAIMAN: le réseau des principaux algorithmes disponibles.

vue (éclairage, optique, arrière-plan), et avec différents algorithmes, afin de répondre à la question "Les indices visuels nécessaires à l'interprétation peuvent-ils être extraits de façon suffisamment fiable?".

Dans ces conditions la réalisation d'une application procède de façon progressive, en enrichissant le système avec des commandes composant les modules de l'application.

4.3 ASPECT "MACHINE SPECIALISEE"

Pour des projets de recherche plus ambitieux en VO le type d'utilisation du système décrit ci-dessus n'est plus satisfaisant.

- * Si le langage FORTRAN est acceptable pour la programmation du bas niveau - calcul de gradients, extraction d'indices visuels - il devient un obstacle pour l'implantation des concepts d'Intelligence Artificielle, tant pour les structures de données que pour les structures de contrôle.
- * La taille mémoire réduite de l'ordinateur sur lequel CAIMAN est implanté impose ses limites.

Afin de compléter l'utilité de CAIMAN avec le confort d'un grand système à temps partagé, un deuxième mode d'utilisation a été introduit, dans lequel les commandes sont lues sur une ligne de liaison externe qui peut être connectée à un autre ordinateur (HB-68, VAX, ou LSI dans le cas de notre laboratoire).

Dans ces conditions, on peut considérer l'ensemble CAIMAN - logiciel et matériel - comme un processeur spécialisé en vision pouvant fournir des indices visuels à un programme d'interprétation sur un autre processeur.

Cette possibilité procure non seulement l'avantage du confort de programmation par lequel nous l'avons motivé, mais aussi la possibilité d'un système de vision repartit. Le programme d'interprétation définit des buts pour CAIMAN, puis lance une analyse. Les algorithmes de CAIMAN étant incrémentaux, le système d'interprétation peut traiter les premiers résultats pendant que l'analyse de CAIMAN continue, et il peut interrompre cette analyse si bon lui semble.

Le programme d'interprétation peut également être constitué par un programme de commande de robot écrit en LM; la liaison externe est ainsi un moyen simple et général d'inclure CAIMAN dans une application en robotique.

Les applications décrites dans les paragraphes suivants font toutes appel à la liaison externe, avec des configurations variables.

4.4 NIVEAUX D'INTERPRETATION DANS LE MONDE DES BLOCS

Nous avons à plusieurs reprises souligné l'importance des structures de données intermédiaires pendant l'élaboration de l'interprétation d'une image. L'objectif du DEA de Yves Demazeau était l'analyse détaillée des niveaux dans le cas du monde des blocs, partant du constat que les systèmes que nous avons décrit au premier chapitre négligent cet aspect, ce qui leur confère un caractère ad-hoc. Dans une perspective à plus long terme, la définition des niveaux intermédiaires et des algorithmes passant de l'un à l'autre est une étape essentielle vers l'étude de stratégies de contrôle dans un système de vision.

Ce projet fut également le premier "utilisateur" de CAIMAN: sa réalisation a entièrement eu lieu en LISP sous MULTICS, évitant toute perte de temps avec les problèmes d'extraction d'indices visuels. C'était donc l'occasion, en ce qui concerne CAIMAN, d'adapter les mécanismes de définition de but et de communication avec un processus externe d'interprétation en fonction d'un cas concret.

La réalisation dans le cadre du DEA couvre l'ensemble des primitives construisant les données à chaque niveau à partir du niveau précédant.

4.4.1 Description des niveaux

Au dessus du niveau image, ce projet fait essentiellement apparaître trois niveaux:

1) Le niveau "indice image" regroupe les indices visuels extraits de l'image et qui la représentent sous une forme bidimensionnelle. Les indices utilisés dans le contexte du monde des blocs sont les noeuds-image et les droites-image.

2) Le niveau "indice-objet" interprète les indices-image par des éléments d'objets physiques. Il est important de le distinguer du niveau précédent parce qu'il n'y a pas de relation simple entre indices-image et indices-objet. Le cas banal, dans lequel, par exemple, une ligne-image correspond à une arête d'un objet, se trouve parmi plusieurs cas plus compliqués:

- * Un indice-image peut ne pas avoir de pendant dans la scène. C'est le cas notamment des noeuds-image de type T provenant d'une occultation entre deux objets.
- * Un indice-image peut être imparfait et se voir attribuer une correspondance avec un indice-objet de structure différente. Dans le monde des blocs, à un noeud-image de

- type L correspondra un noeud-objet de type triédrique.
- * La correspondance entre indice-image et indice-objet est localement ambiguë. Cet aspect a été le point central des travaux décrits dans la première partie.

3) Le niveau "objet" regroupe les indices-objet susceptibles d'appartenir à un même objet sous forme d'un ensemble de faces. Ce regroupement doit être validé par un modèle d'objet externe au processus de vision.

4.4.2 Construction d'indices-image - dialogue avec CAIMAN

La construction du niveau indice-image se fait par un dialogue avec le système CAIMAN. Il s'agit de construire un réseau composé de noeuds-image et de droites-image. Le réseau est initialisé avec un noeud-image défini par un premier point de contraste (commande IA). Le réseau est ensuite étendu itérativement par l'application de trois primitives.

1) Recherche de départs potentiels de droite-image.
A partir d'un noeud N choisi, on obtient un ensemble de points de contraste proches de N, débuts potentiels de droites-image partant du noeud N (commande AN).

2) Validation d'un départ potentiel de droite-image.
Pour un point de contraste C découvert à l'aide de la fonction précédente, on demande une poursuite de droite vers l'extérieur (par rapport à N). Ce suivi se termine avec succès si la ligne présente un nombre suffisant de points, et qu'elle est bien droite.

3) Poursuite d'une droite-image.
Cette primitive sert à aller "jusqu'au bout" d'un départ validé. Cela se traduit par un suivi avec but et contraintes, s'arrêtant sur une des conditions suivantes:

- Perte de contraste ou bord de l'image
- Fin de droite: le suivi est alors arrivé à un angle
- "Ligne latérale" - une autre ligne se trouve proche de la ligne suivie: on se trouve donc en proximité d'un noeud.

Dans les deux derniers cas un candidat pour un nouveau noeud-image a été détecté, qui doit être analysé suivant la primitive 1.

La construction de la structure indices-image doit activer ces primitives, et insérer les nouveaux indices trouvés dans le niveau indices-image.

4.4.3 Evaluation

Ce travail a permis de concrétiser dans le cadre d'une réalisation les propositions pour la structure d'un système de vision que nous avons présentées au paragraphe I-2.2. Il montre également comment un système de vision peut être conforme au schéma de l'IA, tout en contenant une collection d'algorithmes. Cependant, la programmation s'est limitée aux parties K et I, le problème du contrôle ayant seulement été posé en conclusion. Avoir créé un cadre précis pour un programme de contrôle est un résultat positif en soi, dans la mesure où les modèles de structures de contrôle actuellement existant se situent à un niveau d'abstraction trop élevé.

4.5 Localisation d'objets dans un montage complexe

Le projet "montage d'un amortisseur" avait pour objectif d'intégrer dans une application industrielle complexe différents résultats de recherches du laboratoire concernant les langages pour la programmation des robots (LM), l'utilisation de capteurs visuels - noir et blanc, vision 3-D - et des outils graphiques au développement de programmes de robot (simulateur). Il s'agissait donc de montrer l'état de l'art en production automatique à l'aide d'une manipulation concrète et crédible.

Dans ce montage, la vision a été utilisée avec deux fonctionnalités:

- * Repérage de trois pièces métalliques de forme cylindrique sur la table de travail.
- * Guidage de l'insertion de la tige dans le corps intérieur de l'amortisseur.

La première de ces fonction a été réalisée à partir du système CAIMAN, la deuxième fait appel à des outils de vision active 3-D.

Ce travail a constitué le projet de fin d'étude de quatre étudiants de l'ENSIMAG. Nous considérons que dans ce projet le système CAIMAN a été utilisé comme un outil par des ingénieurs non-spécialistes en VO, ce qui correspond à des conditions d'utilisation (presque) industrielles. La partie du projet que nous décrivons a été réalisée par Michel Crebassa et Kurt Akermark.

4.5.1 Approche du problème

Il s'agit de relever la position de trois pièces de forme cylindrique (bague en laiton, joint à lèvres, palier supérieur) posées sur la table de travail, à partir d'une image comme celle de la figure II-44. Comme les pièces sont métalliques et brillantes, l'image fait apparaître les contours des pièces avec des ombres et quelques reflêts, et il est difficile d'obtenir des images "propres", en vue, par exemple, d'une binarisation.

Plutôt que contourner cette difficulté par le choix d'un éclairage parfaitement contrôlé (et aussi difficile à imposer dans notre laboratoire), nous avons cherché une solution plus générale et plus souple: au lieu de considérer les ombres comme un bruit nuisible qu'il faut éliminer à tout prix, elles sont donc utilisées comme éléments d'information positive.

Le programme a la faculté d'analyser l'ensemble formé par l'image d'un cylindre et de son ombre. Il fait les hypothèses de travail suivantes:

- 1) Il y a une seule source de lumière, située loin de la scène. La direction de toutes les ombres de la scène est alors la même.
- 2) Le cylindre est en position debout, et vu approximativement de dessus.

4.5.2 Organisation générale

Le système comporte une phase de mise en oeuvre et une phase d'exploitation proprement dite. La mise en oeuvre comporte deux étalonnages indépendants.

- * Etalonnage vision - robot.
Il s'agit essentiellement de déterminer la transformation 2-D permettant d'associer à un point de l'image des coordonnées sur la table de travail, exprimées dans un repère connu du robot.
- * Détermination du repère ombre.
Un cylindre est placé sous la caméra. Dans l'image, l'ombre du cylindre est caractérisée par deux segments de droites parallèles. En supposant que la source de lumière est placée en bas de l'image, on définit le "vecteur ombre", axe des X du "repère ombre", auquel on ajoute un vecteur Y perpendiculaire, formant ainsi un repère orthonormé (figure II-45).

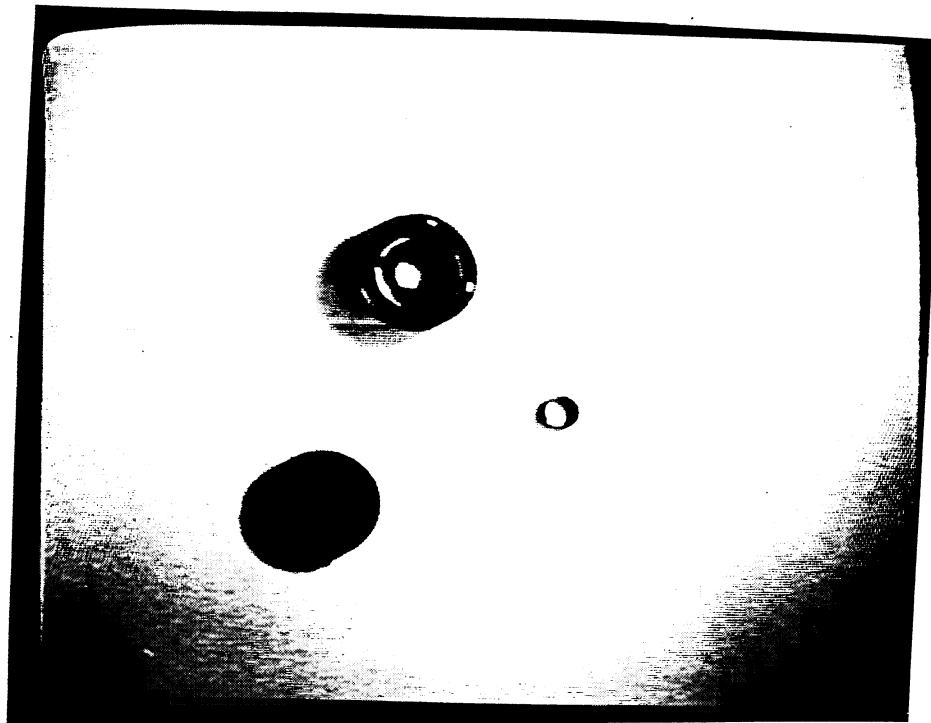


Figure II-44. Image digitale de trois objets cylindriques en métal.

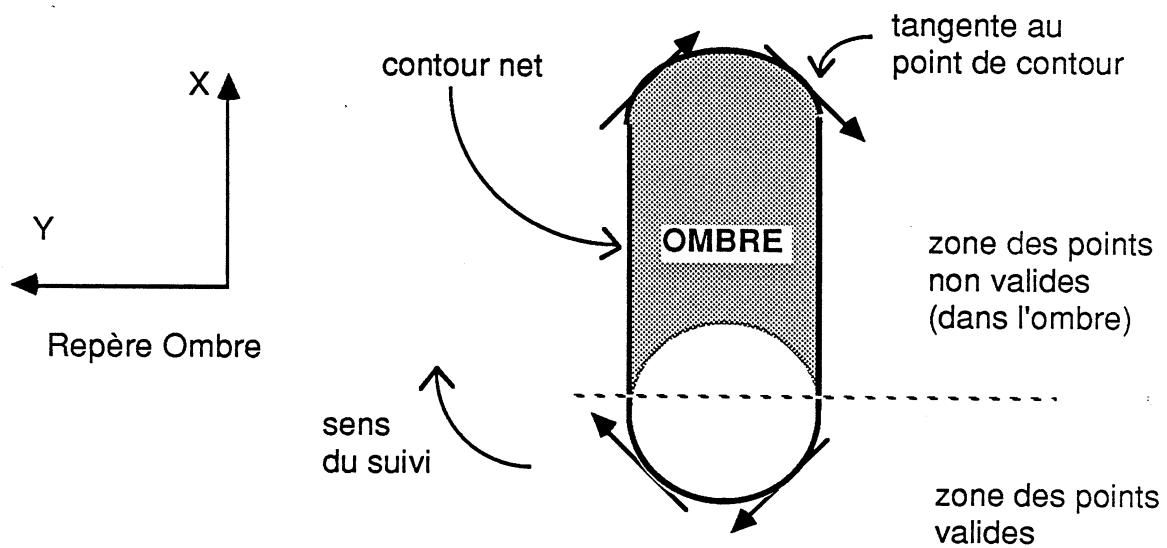


Figure II-45. Configuration type pour la localisation d'un objet cylindrique avec ombre (vue de dessus).

La phase d'exploitation comporte une analyse globale de l'image, qui consiste en un balayage de l'image avec recherche d'un point de contraste, puis l'analyse de la ligne de contraste. Une ligne peut être interprétée comme un bruit et être rejetée, si elle est trop courte, sinon elle est considérée comme l'image d'un cylindre.

4.5.3 Analyse d'une ligne

Dans la phase d'exploitation, l'image est balayée et les lignes de contraste sont analysées afin d'être identifiées avec les objets. L'objectif de l'analyse d'une ligne est la séparation des "points d'ombre" des "points valides" de l'objet. Cette séparation est effectuée en fonction du produit scalaire entre la tangente du point de l'axe Y du repère ombre. Ce produit est positif pour les points appartenant à l'objet, négatif pour les points d'ombre. A partir des points valides un arc de cercle est calculé par une approximation aux moindres carrés. La position du centre du cercle est suffisante pour calculer les paramètres de prise de l'objet.

Nous avons vu à propos de la courbure discrète que le calcul de tangente était fortement affecté par le bruit de discrétisation spatiale. Un simple seuillage du produit scalaire n'est donc pas très fiable. On utilise en plus une contrainte de continuité. Les points valides forment une partie connexe du contour, il suffit donc de localiser les deux points de passage entre la partie éclairée de l'objet et l'ombre.

Dans la mesure où l'objectif du suivi est la localisation de la partie objet, une optimisation intéressante est possible en fonction du point initial, si celui-ci se trouve dans l'ombre: Si la tangente en ce point a un produit scalaire positif avec l'axe X du repère ombre, le chemin le plus court pour atteindre un point valide de l'objet consiste à suivre la ligne en sens inverse (opposé au sens des aiguilles de la montre).

4.5.4 Evaluation

Ce programme résoud de façon simple et efficace un problème pris dans le contexte de la robotique, qu'aucun des systèmes de type industriel décrit au paragraphe I-2.3.8 ne pourrait résoudre - ce problème dépasse le cadre strict de la vision 2-D. L'originalité et l'intérêt de la solution proposée viennent de l'exploitation des ombres, et de l'absence de

contraintes d'éclairage qu'elle entraîne. Nous regrettons seulement quelques restrictions imposées par le programme, même si elles ne sont pas gênantes pour l'ensemble du projet, notamment pour la position des cylindres. Il n'aurait pas été plus difficile de traiter la reconnaissance d'un cylindre bien éclairé en position quelconque (debout ou couché). Le cas général de ce problème (pour les cylindres généralisés) a été étudié par [Shafer 83].

4.6 REGLAGE D'UN TERMINAL A ECRAN CATHODIQUE

4.6.1 Objectifs du projet

Lorsqu'il sort de la chaîne de production, un terminal à écran a besoin d'une série de réglages afin d'assurer la qualité de l'image sur l'écran. D'un point de vue naïf, il s'agit de tourner un certain nombre de vis, de potentiomètres, etc., jusqu'à ce que l'image soit "bonne". Cette manipulation ne se fait pas au hasard - l'image peut présenter plusieurs défauts qu'il faut identifier visuellement afin de déterminer l'action de réglage appropriée. Les images utilisées sont des mires spécialement conçues à cet usage.

L'étude de faisabilité de l'automatisation de cette activité a été le sujet de DEA de Thierry Mondot. L'étude comporte deux parties relativement indépendantes: l'analyse des images par un système de VO afin de déterminer les défauts et de définir les actions de correction, et l'exécution des actions par un robot. Les deux parties ont été réalisées avec le matériel de l'équipe. La figure II-46 montre la configuration matérielle utilisée. Notre description se limitera à l'aspect VO qui était l'aspect prépondérant de l'étude.

Du point de vue de la VO, cette application est originale, parce qu'on connaît parfaitement l'image que l'on souhaite voir apparaître sur le moniteur, mais on ne dispose pas d'informations précises à priori sur l'image apparaissant réellement. On travaille donc initialement dans une incertitude totale qu'il s'agit de lever progressivement. Cependant, il n'est pas réaliste de prévoir à chaque instant tous les défauts ou problèmes possibles. Un aspect essentiel de ce travail consiste donc à rendre explicites les hypothèses faites à chaque étape du réglage.

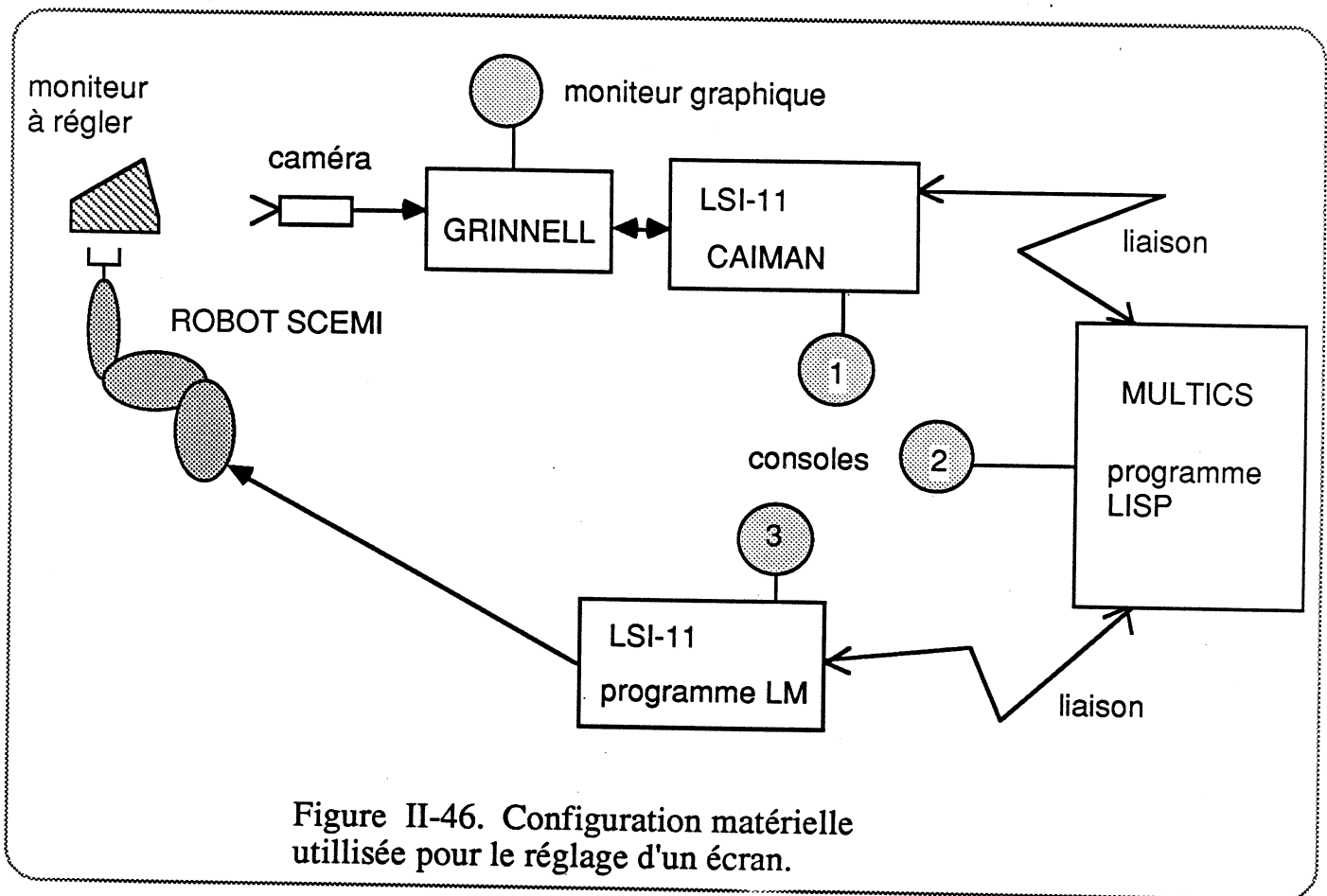


Figure II-46. Configuration matérielle utilisée pour le réglage d'un écran.

4.6.2 Analyse du problème

Un réglage "professionnel" d'un moniteur exige la possibilité d'une mise au point séparé de chaque aspect de l'image. Un cahier des charges fourni par le constructeur décrit les types de défaut à prendre en considération, l'ordre des réglages, la ou les mires sur lesquelles chaque défaut peut apparaître, et les objets de réglage à actionner afin de corriger chaque défaut. Ces défauts concernent

- * la présence d'une image (fonctionnement du tube)
- * la stabilité de l'image
- * les niveaux de luminosité et de contraste
- * les dimensions de l'image
 - hauteur et largeur
 - centrages horizontal et vertical
 - géométrie de l'image: pas de coins coupés,

- arrondis ou étirés
- horizontalité de l'image
- * linéarités verticale et horizontale
- * la netteté de l'image
- * la qualité des caractères: dimensions identiques sur la totalité de l'écran.

Cet ordre des réglages correspond à une certaine logique, allant des contrôles plus généraux - stabilité, dimensions - vers les plus détaillés - qualité des caractères. Cela permet de supprimer, au fur et à mesure des contrôles, les difficultés d'analyse visuelle de l'image. Le réglage de la synchronisation permet de travailler ensuite sur des images stables; les réglages de contraste et de luminosité sont un préalable indispensable à la détection de certaines caractéristiques des images à l'occasion d'autres tests, comme la netteté.

Chaque réglage suit l'algorithme général suivant, dans lequel on compare une propriété de l'image, `val_à_vérifier`, à une valeur nominale, puis opère une opération de correction

```

reglé := faux;
tantque non réglé faire
  debut
    val := val_à_vérifier;
    si abs(val - val_nominale) < ε alors réglé := vrai
    sinon debut
      calculer action de réglage;
      exécuter action de réglage
    fin
  fin

```

`val_à_vérifier` est fournie à chaque itération par l'analyse d'image; `val_nominale` et ϵ sont spécifiques du réglage considéré.

Avec cet algorithme on fait l'hypothèse que le bon réglage peut toujours être atteint (sans quoi il ne se terminerait pas). Cela signifie que

- * Le moniteur n'a pas de vice de construction empêchant le réglage.
- * Les conditions de mise en oeuvre de la méthode de calcul de `val-à-vérifier` sont correctes: les réglages précédant celui en cours ont pu être réalisés, et les réglages suivants ne perturbent pas la recherche des indices visuels nécessaires au calcul de `val-à-vérifier`. Ceci implique l'indépendance entre les réglages.

L'hypothèse d'indépendance entre réglages n'est cependant pas complètement vérifiée. Il se peut qu'un réglage perturbe un contrôle déjà effectué. Par exemple, le réglage de la

linéarité horizontale (ou verticale) peut, si le défaut est important, interagir sur le réglage de la largeur (ou hauteur). On peut alors obtenir une mire qui soit hors tolérances, bien que ses dimensions aient été considérées comme acceptables un instant plus tôt. De même, le réglage de la netteté peut, si l'image était très floue, mettre en défaut divers calculs réalisés précédemment, parce que la distance entre deux points est plus ou moins exacte suivant que ces points sont nets ou non.

Le problème d'interaction est résolu par un contrôle général final avec reprise de certains réglages (s'il y a lieu). Cette façon de procéder réduit considérablement les effets des interactions. A la deuxième itération, les réglages réalisés étant bien moins importants qu'à la première, les interactions potentielles n'en sont que plus faibles et prises pour négligeables.

4.6.3 Le réglage du tilt

Pour chaque réglage il s'agit d'analyser l'image d'une mire connue, pouvant être déformée en raison des réglages non encore réalisés. Le problème de l'analyse de l'image est la localisation de certaines caractéristiques de la mire quelque soit cette déformation, et cela dans un temps raisonnable. Ce problème est résolu par un traitement spécifique à chaque réglage, mettant en oeuvre une stratégie adaptée.

Nous illustrons la démarche par l'exemple de l'analyse de l'image de la mire servant au réglage du tilt. Ce réglage est effectué après le réglage de stabilité, de luminosité, mais avant les réglages de hauteur, largeur, et de centrage. La mire utilisée est donnée par la figure II-47.

Le réglage doit faire en sorte que la distance séparant le bord gauche du capot de la croix centrale de la mire soit sensiblement la même en haut et en bas de la mire. Plus précisément, lorsque la valeur $|T1 - T2|$ (cf. figure II-48) est supérieure à 2 mm il faut effectuer une action de réglage. Pour calculer T1 et T2 avec la plus grande précision possible, on recherche les deux extrémités de la droite verticale de la croix DVC.

L'analyse de l'image procède en deux étapes:

- a) Détermination du centre CC
- b) Recherche de T1 et T2

La deuxième étape est réalisée simplement, utilisant quelques commandes de CAIMAN.

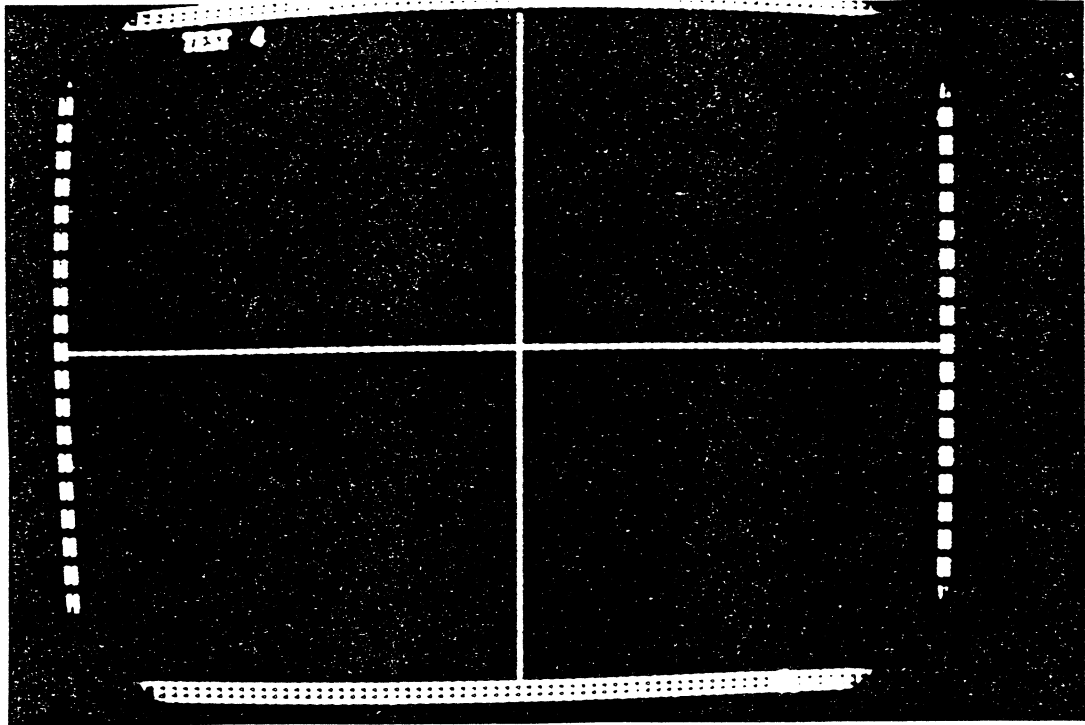


Figure II-47. Mire utilisée pour le réglage du tilt par [Mondot 84].

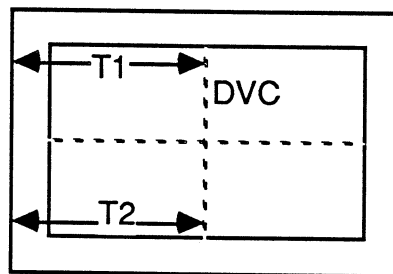


Figure II-48. Paramètres pour le réglage du tilt.

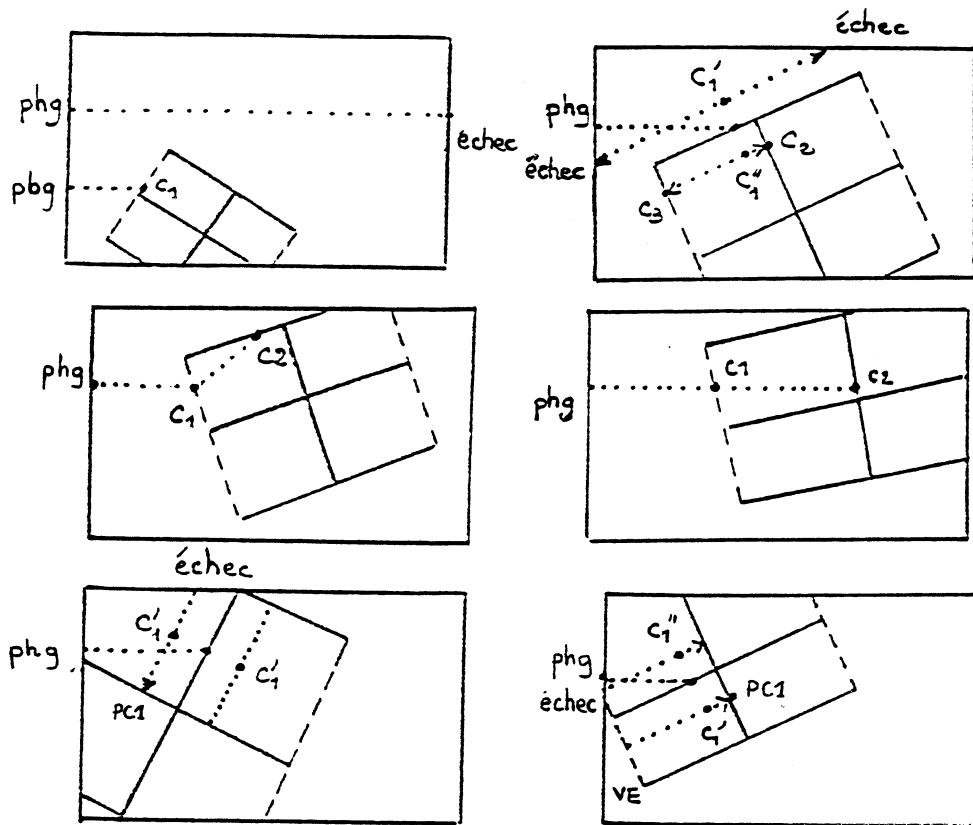


Figure II-49. Configurations possibles pour la recherche du point CC.

- * On recherche autour du point CC quatre points de contraste (commande AN).
- * Parmi les quatre points trouvés, on extrait les deux points C1 et C2 de la droite verticale DVC.
- * On recherche les droites DVC1 et DVC2 passant respectivement par les points C1 et C2. Ces droites peuvent être égales.
- * On extrait des droites trouvées les points PT1, point supérieur de la droite DVC1, et PT2, point inférieur de la droite DVC2.
- * On prend alors
 - T1 = distance de Pt1 au bord gauche du capot
 - T2 = distance de PT2 au bord gauche.

La réalisation de la première étape, la détection du centre CC, s'avère plus problématique. Comme l'illustre la figure II-49 il faut considérer un grand nombre de cas suivant la déformation de la mire. Cependant, ces difficultés ont un caractère gratuit. En choisissant une mire ne contenant que les éléments utiles au réglage du tilt, à savoir la croix, la recherche du centre devient également simple.

4.6.4 Evaluation

En ce qui concerne l'application, cette étude a amené à une réflexion sur la méthode de réglage. Dans un premier temps, cette réflexion aboutit à un changement dans l'ordre des réglages, et à la définition de mires mieux adaptées à la VO, les mires initiales étant adaptées à la vision humaine.

En ce qui concerne le système CAIMAN, la possibilité de définir des buts en termes d'indices visuels a montré son utilité et son efficacité.

PARTIE III

LA RECONNAISSANCE D'OBJETS

La partie III concerne le passage entre les niveaux "indice visuel" et "objet" - la reconnaissance d'objets. Comme la notion intuitive de reconnaissance d'objet recouvre une grande variété de phénomènes mentaux nous présentons d'abord une formalisation de la reconnaissance en termes de correspondances entre ensembles, ce qui nous permet de situer dans un cadre formel des notions comme reconnaissance, objet, visibilité partielle, indice visuel, modèle, ... Cette formalisation fournit la spécification du problème informatique de reconnaissance. Nous analysons les problèmes posés par cette spécification sur le plan algorithmique (chapitre 2) et sur le plan de l'IA (chapitre 3).

PRINCIPAUX SYMBOLES UTILISES DANS LA PARTIE III

IV	indice visuel
ID _j	indice données
IM _i	indice modèle
m	nombre d'indices modèle
d	nombre d'indices données
H	hypothèse
C	correspondance
T	ensemble de contraintes
θ	transformation
t	nombre de coefficients d'une θ
p	nombre d'équations par association
PL	fonction de plausibilité
r	rang de la matrice d'une H
n	[t/p]
s	nombre de types d'IVs différents

CHAPITRE 1

UNE FORMALISATION DE LA NOTION DE RECONNAISSANCE

Dans ce chapitre nous présentons le problème de reconnaissance d'objets partiellement visibles du point de vue de la complexité algorithmique; nous nous intéressons principalement à la combinatoire des structures de données, ainsi qu'au coût en temps et en place des algorithmes.

Afin de ne pas alourdir l'exposé, nous traitons d'abord le problème sous une forme épurée: reconnaissance d'objets d'un seul type d'objet, employant un seul type d'indice visuel, le segment de droite, et des correspondances géométriques exactes. Toutefois, nous n'émettons pas d'hypothèse concernant la dimension de l'univers (2D ou 3D). Il s'avère que, d'un point de vue combinatoire, cette forme simplifiée est le cas le plus défavorable. Elle permet donc de mettre en évidence les difficultés fondamentales soulevées.

Quelques généralisations simples sont discutées au paragraphe 1.6: prise en compte de différents types d'indices et de différents types d'objets.

1.1 ASPECTS INTUITIFS DU PROBLEME

1.1.1 Présentation du problème

Considérons le problème de "reconnaitre" un triangle dans une image. La figure III-1 esquisse les structures intervenant dans ce problème. La figure III-1-a montre le modèle de l'objet, la figure III-1-b représente l'image. La figure III-1-c montre plusieurs possibilités d'identification du triangle. En effet, nous supposons que le triangle est partiellement visible, c'est à dire que le modèle peut être plus complet que l'information directement fournie par l'image. La figure III-1-d montre les éléments d'une "hypothèse" d'objet: une "correspondance" entre certains

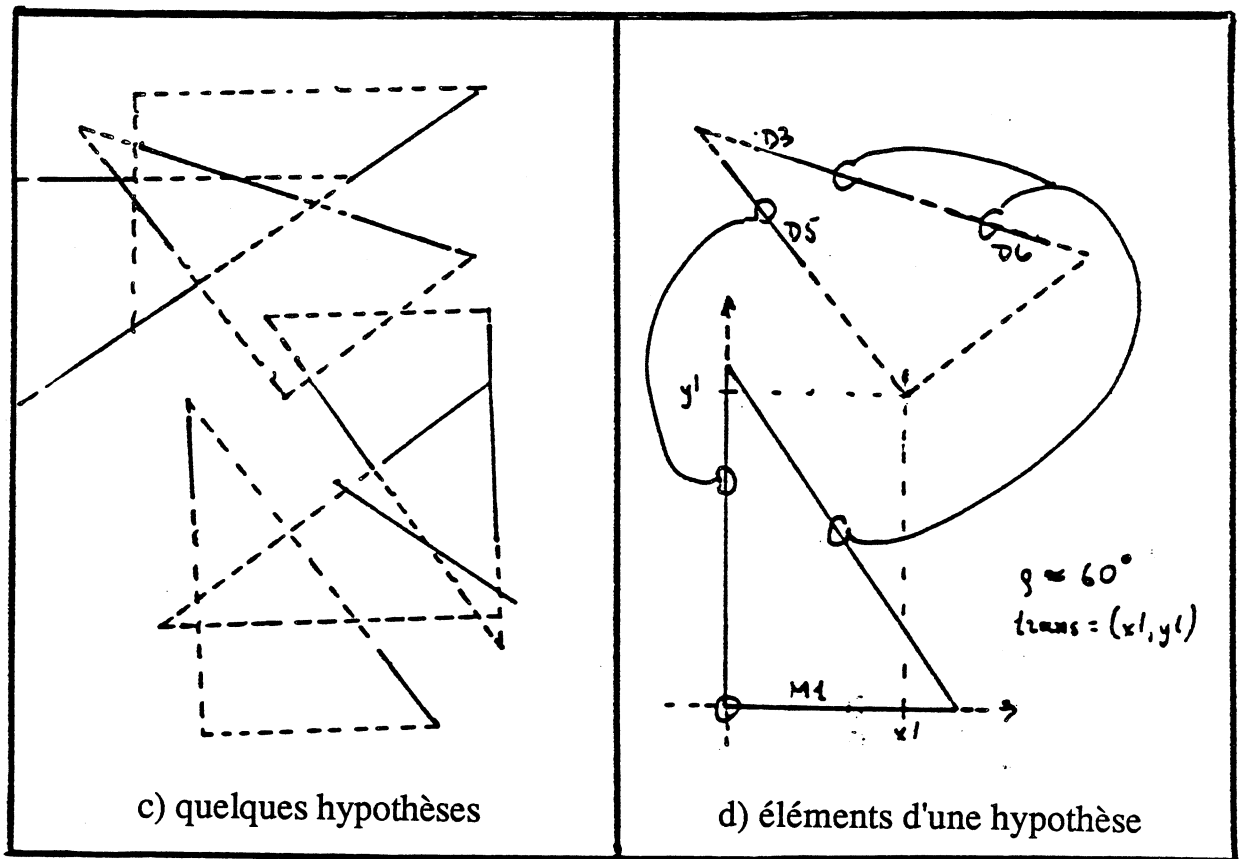
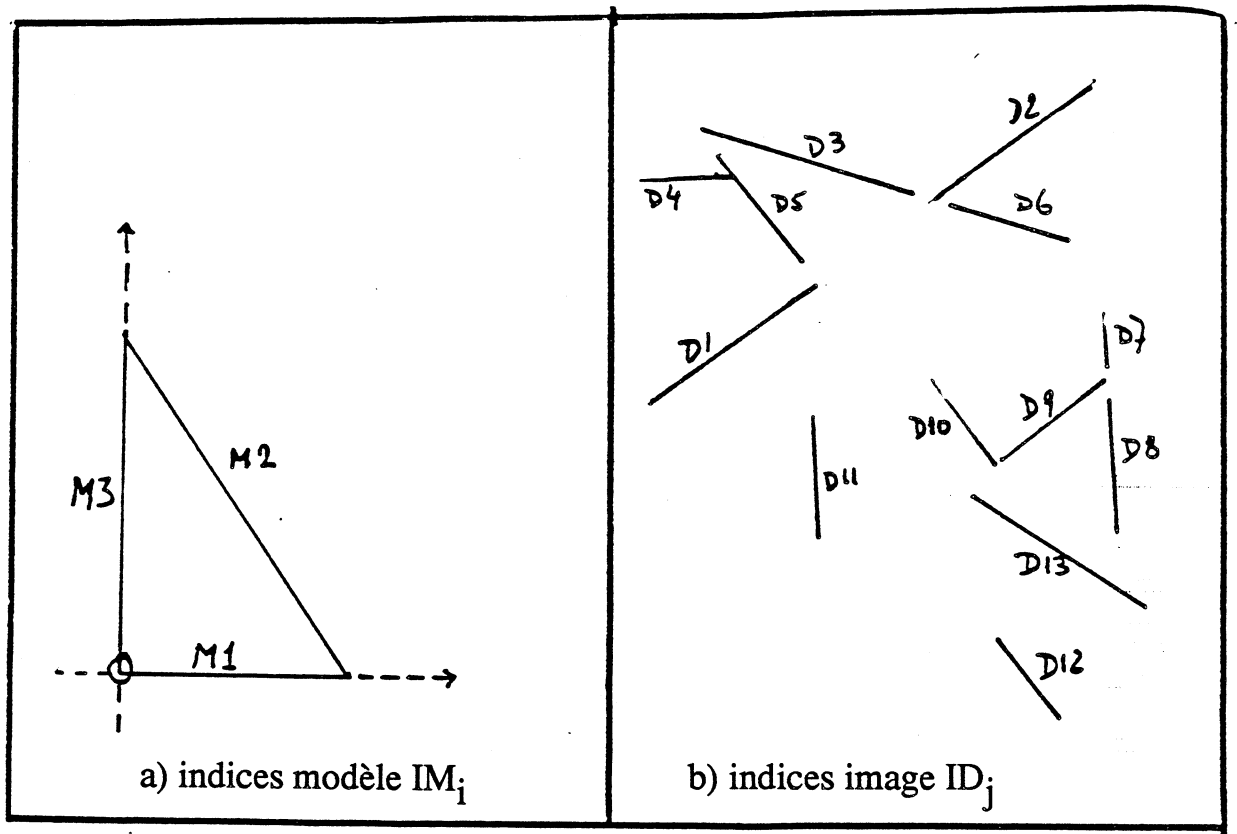


Figure III-1. Notions de base pour la reconnaissance d'objets.

indices du modèle et des indices de l'image, et une "transformation" superposant le modèle avec les indices de l'image.

Cet exemple amène à un premier énoncé du problème de reconnaissance d'objet: étant donné un **modèle** d'objet et une **description** de l'image, l'objet modélisé est-il **présent** en un ou plusieurs exemplaires, et si oui, dans quelles **positions**?

1.1.2 L'aspect "visibilité partielle"

Avec la paraphrase "objet partiellement visible" nous décrivons la situation suivante: dans une correspondance entre modèle et image, une partie seulement du modèle est effectivement repérable dans l'image. D'un point de vue physique, cela peut avoir plusieurs raisons. En particulier:

- * Faces auto-occultées. Un modèle peut décrire la structure 3-D d'un objet, alors qu'une image ne présente qu'une vue particulière, avec une partie avant visible, et une partie arrière non-visible.
- * Faces cachées. Dans une scène comportant plusieurs objets, certaines parties de l'objet considéré peuvent être cachées par des objets placés devant.
- * Une partie de l'objet peut se trouver à l'extérieur de l'image.
- * Le module de description de l'image, qui produit les indices-image, donne des résultats imparfaits. Nous nous contentons donc d'une postcondition relativement faible pour ce module, à savoir qu'il trouve une partie des indices correspondant aux objets de la scène.

Ces arguments montrent la nécessité de l'hypothèse de visibilité partielle dans un système de vision quelque peu général. Seulement dans des cas très restrictifs, il peut y avoir correspondance entre la totalité d'un modèle et des indices image. Ceci est d'autant plus vrai que la notion de modèle peut recouvrir des informations autres qu'une description purement "picturale" ou géométrique, à savoir toutes les "connaissances" sur un objet.

1.1.3 Problèmes liés à la visibilité partielle

La contrainte d'une reconnaissance d'objet à partir d'indices partiels est à l'origine de deux difficultés fondamentales:

- * Elle amène une combinatoire extrêmement forte (exponentielle avec le nombre d'indices), qui résulte de l'ambiguïté des interprétations locales. Comment maîtriser cette combinatoire?
- * Il ne peut plus y avoir de certitude absolue qu'un objet est bien présent dans la scène. Quels sont les critères selon lesquels on décide de la présence d'un objet?

Nous approfondirons essentiellement le premier de ces problèmes. Nous reviendrons brièvement sur le deuxième problème à au paragraphe 2.1 et au chapitre 4.

1.2 DEFINITIONS PRELIMINAIRES

Ce paragraphe introduit les notions d'indice modèle, d'indice image, et de correspondance.

1.2.1 Indices visuels

D'un point de vue strictement formel les IVs forment le vocabulaire terminal de la description des objets. D'un point de vue pratique, un IV doit être une entité géométrique simple. Les indices visuels jouent un rôle clé dans un système de vision, parce qu'ils forment la charnière entre la description-segmentation et l'interprétation. Un bon choix d'IVs doit réaliser un compromis entre les exigences de ces deux modules:

- * La description doit extraire les types d'IVs choisis de façon fiable d'une image. Il doit donc admettre une définition en termes de primitives-image, c'est à dire en pixels.
- * Les IVs forment le vocabulaire de construction de modèles d'objet. Un IV doit donc être une entité suffisamment abstraite afin de pouvoir modéliser l'apparence d'un objet dans une grande variété de prises de vue (cf. paragraphe 1.6.1).

Dans un premier temps, nous considérons un seul type d'IV, le segment de droite, représenté par ses extrémités qui sont des

points.

1.2.2 Modèles

Un modèle est une abstraction picturale d'une classe d'objets: il décrit l'apparence des objets de cette classe dans une image. Un modèle M est un ensemble d'indices visuels

$$M = \{IM_1, \dots, IM_n\}$$

Nous parlerons des "indices visuels modèle" ou plus simplement des "indices modèle".

Les indices modèle décrivent un objet dans une position de référence dans un repère cartésien propre à l'objet.

1.2.3 Indices image

Les "indices image" ou "indices données" sont des IVs extraits d'une image par un système de description. Nous avons étudié les problèmes liés à l'extraction d'indices visuels dans la partie II.

Les définitions des paragraphes suivants portent toutes sur un ensemble d'IV modèle $\{IM_i\}$ et un ensemble d'IV image $\{ID_j\}$.

1.2.4 Correspondances

Soit un ensemble d'IVs $\{IM_i\}$ - les indices modèle - et un ensemble d'IVs $\{ID_j\}$ - les indices image.

Une **correspondance** est un sous-ensemble du produit cartésien

$$\{IM_i\} \times \{ID_j\}$$

représentée par une liste d'associations

$$((IM_{m_1} ID_{j_1}) \dots (IM_{m_L} ID_{j_L}))$$

Le nombre d'associations L est la **longueur** de la correspondance.

L'inclusion ensembliste définit une relation d'ordre sur les correspondances que nous appelons "plus informée que".

Définition: Si C_1 et C_2 sont deux correspondances telles que $C_1 \subset C_2$, alors C_2 est plus informée que C_1 .

L'utilisation des correspondances pour la reconnaissance d'objets constitue une approche très générale. Elle accepte un certain nombre de cas particuliers pouvant paraître

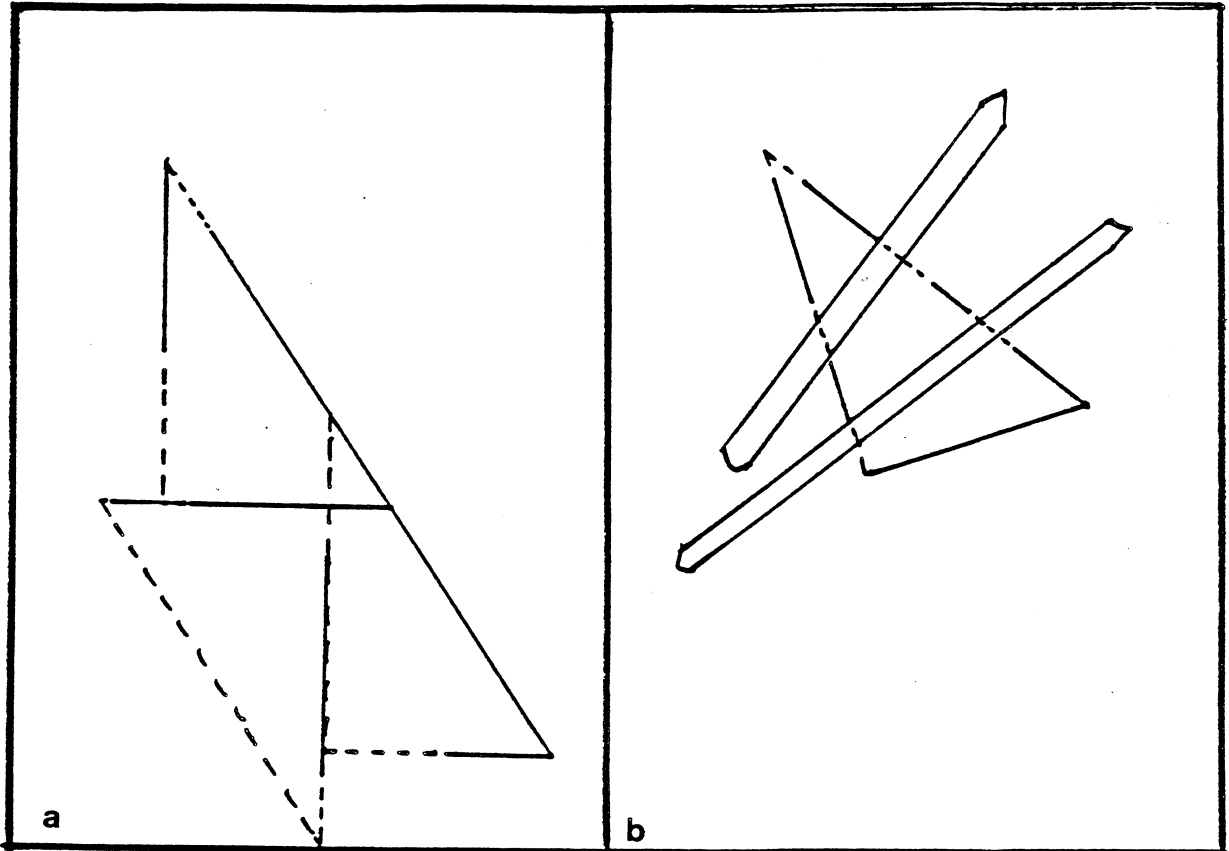


Figure III-2. Cas particuliers: a) alignement b) occultation.

pathologiques, mais qui sont physiquement possibles:

- * A plusieurs indices modèle peut correspondre un seul indice image (figure III-2-a). Dans certaines conditions, ce cas est appelé "alignement accidentel".
- * A plusieurs indices image peut correspondre un seul indice modèle (figure III-2-b). Cela peut notamment arriver en cas d'occultation par un autre objet. Si on ne veut pas imposer trop de contraintes à la description, les indices image en question peuvent même se recouvrir partiellement.

1.3 Hypothèses

1.3.1 Définition

L'ensemble des correspondances forme un espace de base simple à partir duquel on peut construire des hypothèses d'objet. Une hypothèse s'obtient par l'interprétation d'une correspondance en ce sens que les différents IDs représentent effectivement l'image de l'objet. Une telle interprétation impose implicitement de nombreuses contraintes de cohérence aux éléments de la correspondance, qui permettent de sélectionner les correspondances intéressantes. Dans une hypothèse, certaines de ces contraintes sont représentées explicitement.

DEFINITION: Une **hypothèse** H est une correspondance C garnie d'un ensemble de contraintes T

$$H = (C, T).$$

Une hypothèse est **cohérente** si et seulement si les éléments de la correspondance satisfont les contraintes dans T.

La relation "plus informée que" s'étend naturellement aux hypothèses:

Définition. Une hypothèse $H_2=(C_2, T_2)$ est **plus informée** qu'une hypothèse $H_1=(C_1, T_1)$ si et seulement si C_2 est plus informée que C_1 .

Il est clair qu'une hypothèse plus informée est également plus intéressante pour résoudre le problème de reconnaissance. Par contre, les hypothèses ne sont pas toutes comparables par cette relation. A cette fin nous définissons les fonctions de plausibilité:

Définition: Fonction de plausibilité.

Une fonction de plausibilité PL est une fonction de l'ensemble des hypothèses dans un ensemble totalement ordonné, et compatible avec la relation "plus informée que", c'est à dire

$$\text{Si } H_2 \text{ plus informée que } H_1 \text{ alors } PL(H_2) \geq PL(H_1).$$

La définition de la plausibilité pour les hypothèses (et non pour les correspondances) permet de faire intervenir les contraintes dans cette fonction.

Exemples:

- 1) $PL(H)$ = nombre d'indices modèle intervenant dans une correspondance.
- 2) $PL(H)$ = pourcentage du total des longueurs des indices

modèle réalisés par la correspondance.

1.3.2 Reconnaissance d'un objet

La fonction de plausibilité nous fournit une solution simple au deuxième problème évoqué au paragraphe 1.1.3.1, à savoir du critère de décision "objet présent". On estimera qu'un objet est présent s'il figure dans une hypothèse ayant une plausibilité dépassant un certain seuil.

Après ces définitions, nous sommes en mesure d'énoncer l'objectif d'un programme de reconnaissance d'objets:

Etant donné un modèle $\{IM_i\}$ et un ensemble d'IV image $\{ID_j\}$ trouver une ou plusieurs hypothèses cohérentes dont la plausibilité dépasse un seuil donné.

1.3.3 Transformations géométriques

La définition d'une hypothèse au paragraphe précédent est très générale (au point de paraître vague), laissant toute latitude quant à la nature des contraintes. Cette généralité est voulue, parce que de la richesse des contraintes effectivement prises en compte dépend l'intelligence d'un système de reconnaissance d'objets.

Dans la suite de ce chapitre, nous traitons un cas particulier important: les contraintes T déterminent une transformation géométrique θ_T qui "superpose" les IMs aux IDs. Dans ce cas, une hypothèse est cohérente si pour chaque association (IM_{m_i}, ID_{j_i}) la transformée $\theta_T(IM_{m_i})$ est compatible avec ID_{j_i} . "Compatible" est à prendre ici dans un sens géométrique. Par exemple, deux segments de droites sont compatibles s'ils se recouvrent partiellement. (Nous reviendrons sur le problème de compatibilité de segments au chapitre 2).

Dans la suite nous nous limitons au cas dans lequel la transformation θ est définie par un système d'équations linéaires. On suppose que θ est définie par t coefficients, et que chaque association produit p équations dans les coefficients de θ . Une correspondance de longueur l définit alors un système de $p \cdot l$ équations linéaires à t inconnues. Suivant le rang r et la condition de ce système nous distinguons trois cas:

1) $r < t$.

Le système est sous-déterminé. La correspondance ne fournit pas assez d'informations pour définir une

position unique de l'objet. Ce sera en général le cas lorsque $l \cdot p < t$ (il n'y a alors pas assez d'associations) ou lorsque certaines équations sont linéairement dépendantes (segments de droites parallèles). Une telle hypothèse est appelée "hypothèse sous-déterminée".

2) $r = t$.

Il y a une seule solution. L'hypothèse est cohérente et représente l'objet dans une position bien définie. Le plus souvent ce sera le cas pour $l \cdot p = t$. Une telle hypothèse sera appelée "admissible". Une hypothèse admissible est cohérente si elle vérifie en plus les conditions de compatibilité.

3) système incohérent.

Les équations sont contradictoires. Aucune transformation ne peut superposer le modèle aux indices image de la correspondance. La correspondance est donc sans intérêt pour notre problème. Il est évident que l'ajout de nouvelles équations ne pourra réduire le rang r . Donc: si H_1 est incohérent et H_2 plus informée que H_1 , alors H_2 est incohérent.

1.4 ESPACES DE RECHERCHE

1.4.1 Le graphe des hypothèses

La génération des correspondances est sujette à une très forte combinatoire. Si m est le nombre d'indices modèle, et d le nombre d'indices image, il y a

$$\binom{m \cdot d}{1}$$

correspondances de longueur 1, soit $2^{m \cdot d}$ au total. Il est donc totalement exclu qu'un algorithme construise explicitement l'ensemble des correspondances pour ensuite sélectionner les plus plausibles. Dans ces conditions, il nous semble mieux approprié de formuler le problème de reconnaissance comme un problème de recherche dans un graphe, c'est à dire dans la terminologie de l'IA.

Le "graphe des correspondances" (cf. figure III-3-a) se définit naturellement comme un graphe orienté sans circuit. Le noeud initial est la correspondance vide. Les successeurs d'un noeud sont obtenus en ajoutant une association supplémentaire.

Remarque: il serait inefficace de traiter ce graphe comme un arborescence, puisque chaque correspondance de

longueur L possède L prédécesseurs, et qu'elle peut être atteinte suivant $L!$ chemins différents, l'ordre des associations étant indifférent.

Le graphe des hypothèses s'obtient à partir du graphe des correspondances en complétant chaque correspondance par les contraintes associées. Ce graphe est facile à construire, et possède plusieurs propriétés simples en rapport avec notre problème:

- * Il permet de minimiser le nombre d'hypothèses incohérentes à considérer. Dans la mesure où tous les successeurs d'une hypothèse incohérente sont également incohérents, les hypothèses incohérentes correspondent aux feuilles (d'échec) du graphe à engendrer.
- * Le long d'un chemin du graphe, la plausibilité croît de façon monotone.

1.4.2 Hypothèses équivalentes

Considérons le sous-graphe d'une solution $H_0 = (C_0, T_0)$, c'est à dire le graphe de toutes les hypothèses $H_i = (C_i, T_i)$ moins informées que H_0 .

Ce sous-graphe est composé de deux sous-graphes:

- G_s sous-graphe des hypothèses sous-déterminées. Ce sous-graphe est connexe.
- G_a sous-graphe des hypothèses admissibles.

Les transformations θ_{T_i} des hypothèses de G_a sont toutes les mêmes, puisque les hypothèses de G_a correspondent au même objet dans la même position. Elles sont donc équivalentes pour notre problème.

Définition. Deux hypothèses $H_1=(C_1, T_1)$ et $H_2=(C_2, T_2)$ sont **équivalentes** ssi elles définissent les mêmes transformations.

1.4.3 Le graphe réduit

L'équivalence entre hypothèses possède une propriété simple: les noeuds d'une classe d'équivalence forment un sous-graphe connexe du graphe des hypothèses (figure III-3-b). D'autre part, le nombre de noeuds dans une classe d'équivalence peut être important. Dans le cas extrême - lorsque tous les indices image peuvent être réunis dans une seule hypothèse cohérente - toutes les hypothèses admissibles forment une seule classe.

Pour la solution de notre problème il sera donc intéressant de développer le quotient du graphe des hypothèses par rapport à la relation d'équivalence "même transformation" et que nous appelons "graphe réduit".

Le graphe réduit possède la structure suivante. Les noeuds non-terminaux contiennent les hypothèses sous-déterminées, et il y a deux types de feuilles:

- hypothèse incohérente,
- classe d'équivalence.

Le second type de noeud sera appelé "hypothèse complète".

Le nombre de noeuds du graphe réduit est facile à estimer. Soit

$$n = \lceil t/p \rceil$$

le nombre d'association qu'il faut normalement pour déterminer une transformation. En général, les hypothèses de longueur $< n$ seront sous-déterminées, celles de longueur n soit admissibles, soit incohérentes.

Le graphe réduit possède donc

$$\sum_{j=1}^{n-1} \binom{m*d}{j}$$

noeuds non-terminaux, et la construction des feuilles concerne

$$\binom{m*d}{n}$$

hypothèses - toutes les hypothèses de longueur n doivent être examinées pour la construction des classes d'équivalence.

Le nombre de feuilles de type classe d'équivalence ne peut être indiqué en absence de données statistiques concernant la distribution des indices image. Ce nombre dépend du nombre d'objets dans la scène, des ressemblances fortuites avec des parties de l'objet recherché (ambiguïté locale), et des alignements accidentels.

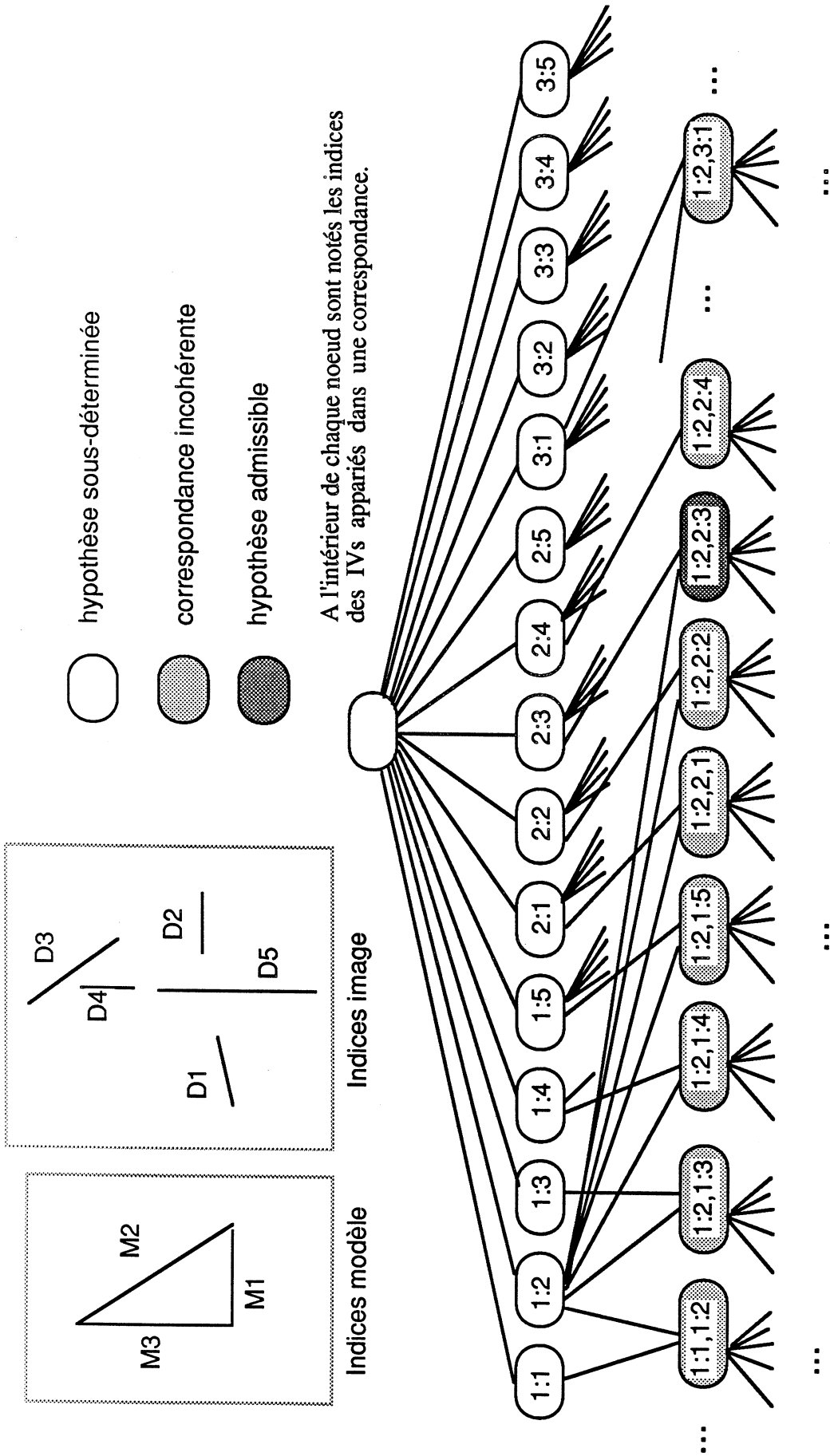


Figure III-3-a. Illustration du graphe des hypothèses.

Il s'agit du cas de figure le plus simple de reconnaissance, avec rotation-translation du modèle; les paramètres du problème sont $t=3$, $p=2$, $m=3$, $d=5$.

Malgré la simplicité de l'exemple, la taille du graphe des hypothèses est considérable, puisqu'il comporte $2^{**}15$ noeuds. Cette figure montre les 15 noeuds du premier niveau, et une partie des 105 noeuds du deuxième niveau (le troisième niveau possède 455 noeuds).

Premier niveau du graphe réduit identique au premier niveau du graphe des hypothèses

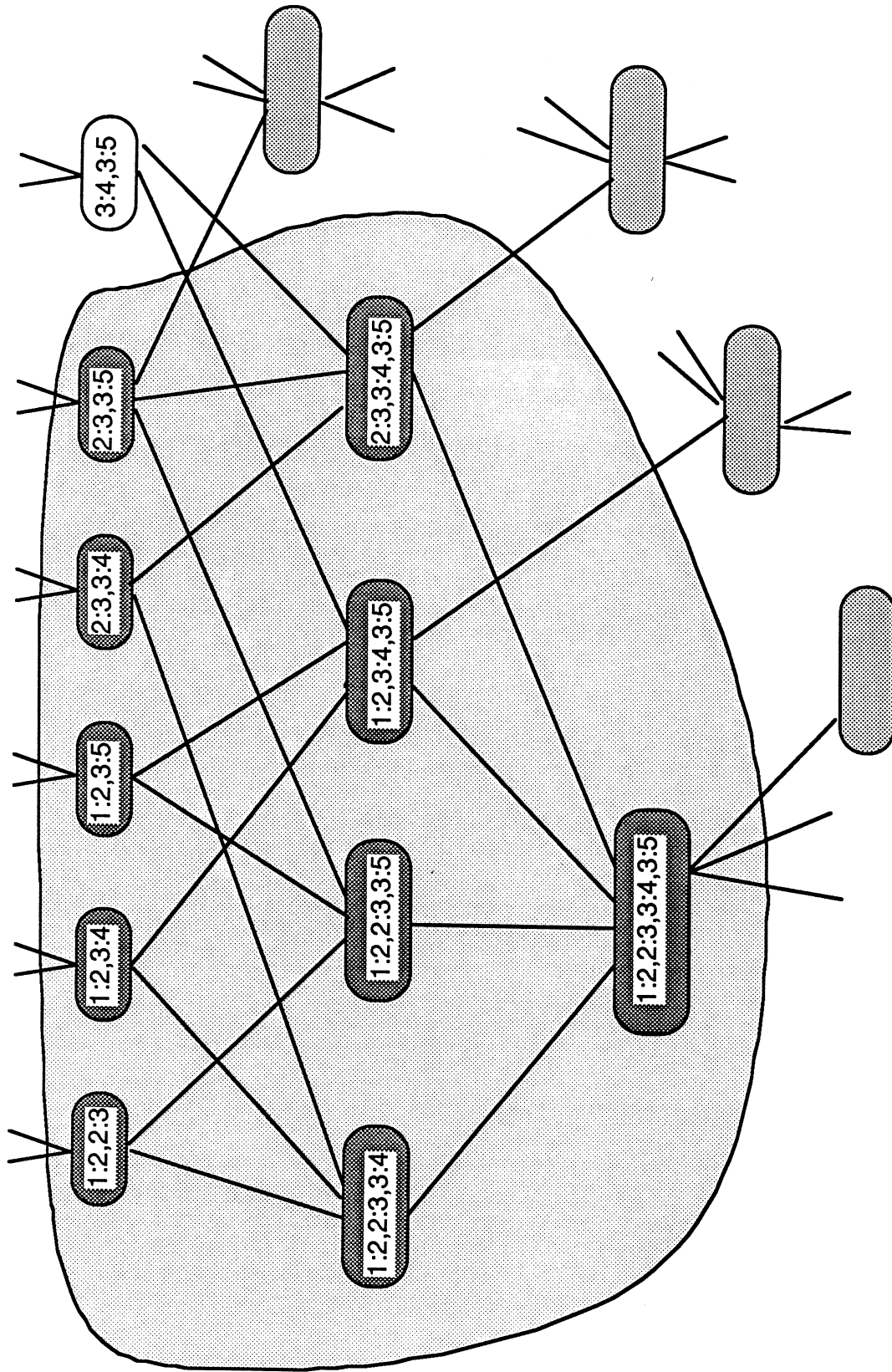


Figure III-3-b. Graphe des hypothèses et graphe réduit.

Pour l'exemple de la figure 3-a, cette figure montre une hypothèse complète du graphe réduit (la seule de l'exemple). L'hypothèse complète comporte tous les noeuds du graphe des hypothèses fournissant la même position. Les arcs sortant de la classe d'équivalence mènent tous vers des hypothèses incohérentes, donc sans intérêt.

1.5 ALGORITHME DE RECONNAISSANCE

L'algorithme de reconnaissance utilisant le graphe réduit doit résoudre deux sous-problèmes:

- 1) Trouver les classes d'équivalence par parcours du graphe réduit.
- 2) Etant donné une classe d'équivalence, trouver une hypothèse suffisamment plausible.

Ces deux sous-problèmes possèdent une forte combinatoire, mais sont de natures très différentes. Le premier est de l'ordre $(m*d)**n$. Une solution "intelligente" de ce problème nécessite l'utilisation de connaissances plus riches que des modèles d'objets, à savoir des connaissances sur l'agencement des objets dans la scène. Cela amène naturellement à l'utilisation de techniques spécifiques de l'IA, en particulier la génération de plans d'actions, et sera approfondi au chapitre III-3. Le deuxième problème est bien adapté à la prédiction-vérification, que nous introduisons ci-après.

Pour la solution du deuxième problème, nous nous plaçons dans le cadre légèrement plus général de la recherche d'hypothèses à plausibilité maximale. On peut alors envisager deux algorithmes très différents: la recherche dans le graphe d'hypothèses et la recherche par prédiction.

1.5.1 Recherche dans le graphe d'hypothèses

A partir d'un noeud quelconque de la classe, chercher un successeur plus plausible, jusqu'à arriver sur une feuille.

Si on suppose que $m \gg p$ (c'est à dire que les indices modèle sont nombreux), le coût de cette recherche est dominé par le terme

$$m * d$$

1.5.2 Recherche par prédiction

La transformation θ de la classe d'équivalence représente une contrainte très forte concernant les associations nouvelles, qui permet de construire directement l'hypothèse la plus plausible de la classe. En effet, pour un indice modèle donné, ses associations sont définies par

$$\{ID_j \mid \theta(IM_i) \text{ compatible } ID_j\}$$

La construction de ces associations demande m recherches dans les ID_j . Si les IDs sont rangés dans un tableau, une recherche séquentielle résultera dans un coût total de l'ordre

$$m * d$$

une recherche dichotomique donnera

$$m * \log(d)$$

et avec une organisation plus sophistiquée (par exemple iconomorphe) le coût devient proportionnel essentiellement à la somme des longueurs des segments du modèle, donc de l'ordre de m . Remarquons qu'il faut alors que le coût de la construction de l'organisation soit aussi linéaire en m .

Moyennant une organisation appropriée des données, la recherche par prédiction permet donc de ramener le coût de la solution du deuxième sous-problème à l'ordre m , ce qui est nettement meilleur que le comportement d'ordre $m*d$ de la recherche dans le graphe des hypothèses.

1.6 GENERALISATIONS

Nous analysons encore brièvement l'incidence de deux généralisations sur l'aspect combinatoire du problème: l'utilisation de différents types d'IVs et le cas de plusieurs modèles.

1.6.1 Différents types d'IVs

L'utilisation de différents types d'IVs amène à définir le modèle comme somme de plusieurs ensembles

$$M = M_1 + M_2 + \dots$$

Nous engendrons alors seulement des correspondances à base d'associations entre IVs de même type.

Cela revient à poser une condition très forte: le type d'un IV doit être indépendant des paramètres de prise de vue. Cela interdit, par exemple, de considérer des arcs de cercle et des arcs d'ellipse comme des types d'IV différents (dans des

conditions de prise de vue générales). Des exemples de "bons" indices par rapport à ce critère sont

- segment de droite
- virage à droite
- virage à gauche
- point d'inflexion
- angle
- couleur d'une région

Le nombre total de correspondances s'élève dans ces conditions à

$$\prod 2^{(m_k * d_k)} = 2^{(\sum m_k * d_k)}$$

(m_k est le nombre d'indices modèle dans M_k).
Si le nombre d'occurrences dans le modèle de chaque type d'indice est le même, et qu'il y a s types différents d'IVs, cela revient à

$$2^{(M*D/s)} \text{ où } M = \sum m_k, D = \sum d_k$$

Si nous avons M indices modèle et D indices image d'un seul type, nous aurions ($s=1$)

$$2^{(M*D)} = 2^{(\sum m_k * \sum d_k)}$$

correspondances. A nombre total d'IVs constant, l'introduction de différents types d'IVs entraîne donc une réduction significative de la taille du graphe des hypothèses.

Remarque: Pour chaque nouveau type d'IV il faut revoir la notion de compatibilité (1.3.3), et la nature des contraintes qu'il impose à la cohérence d'une hypothèse.

1.6.2 Cas de plusieurs modèles

La recherche d'objets de différents types n'introduit pas d'éléments nouveaux pour l'évaluation combinatoire. Dans une première approximation on peut estimer que les possibilités se multiplient et que le nombre de correspondances est

$$2^{(\sum \mu_i * d)}$$

(μ_i est le nombre d'indices du i° modèle).
Bien que la présence simultanée de plusieurs objets implique de nombreuses contraintes symboliques (cf. paragraphe 2.1), l'incidence sur la taille du graphe des correspondances est difficile à évaluer dans le cas général.

L'aspect le plus important apporté par cette généralisation réside dans les nombreuses contraintes symboliques qu'elles introduit. Cet aspect sera pris en compte dans les chapitres suivants. Elle introduit aussi le problème de cohérence de plusieurs hypothèses dans le contexte d'une reconnaissance de plusieurs objets dans une même image, problème qui se complique s'il y a ambiguïté entre les modèles.

CHAPITRE 2

MISE EN OEUVRE - APPLICATIONS

Le chapitre 2 analyse l'utilisation du formalisme de reconnaissance dans un système de VO. Il devient alors important de considérer deux aspects nouveaux - la correspondance approximative (§2.1), nécessitée par la nature bruitée et imparfaite des images, et l'interaction interprétation - description, que nous avons évoquée au §1-2.2. Cette analyse débouche sur la définition de la stratégie de prédiction - vérification au §2.2.3. Ces développements sont illustrés à l'aide de quelques systèmes de vision existants, notamment le système PVV issu de notre équipe, puis comparés à d'autres stratégies de reconnaissance.

2.1 Correspondance approximative

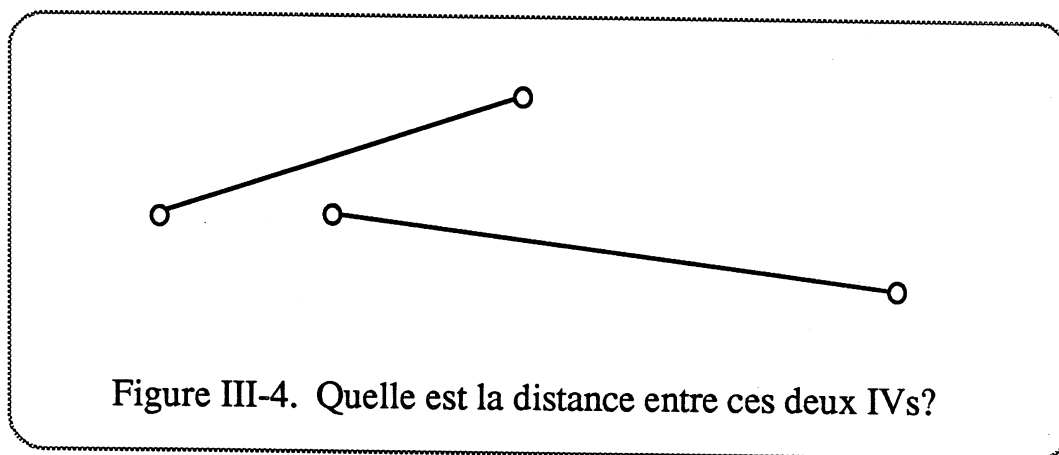
Les indices image, et dans une moindre mesure les indices modèle sont affectés d'une erreur. Il en résulte que les correspondances entre indices ne sont jamais exactes, mais doivent être prises à un epsilon près. La nécessité de considérer des correspondances approximatives entre indices modèle et indices image nous amène à affiner l'analyse du problème à résoudre, et à décomposer la reconnaissance en deux sous-problèmes différents que nous appelons **identification** et **localisation**.

L'identification est un problème de raisonnement symbolique et correspond aux décisions suivantes:

- * L'objet est présent,
- * Les segments $ID_{j,i}$ représentent les (ou des) parties visibles de l'objet.

Le problème de localisation est de fournir la transformation T. La localisation suppose résolue l'identification et est un problème à caractère numérique.

La démarche suivie dans le chapitre précédent résoud les deux problèmes d'un coup, par la solution du système d'équations associé à une correspondance. La solution du problème d'identification est alors liée à l'existence d'une solution numérique du problème de localisation. Or, dans le cas d'une correspondance approximative le système d'équations est (pratiquement) toujours incohérent.



Cela ne met pas forcément en cause les principes de localisation. Au lieu de déterminer la solution d'un système d'équations, nous pouvons calculer l'approximation (nous choisissons l'approximation au sens des moindres carrés). Notons cependant une difficulté importante concernant la définition de la "distance" entre deux IVs que nous cherchons à minimiser (cf. figure III-4). Par contre, le problème d'identification devient plus délicat à résoudre, car un calcul d'approximation acceptera (presque) tout ensemble de données ID_j - la qualité de l'approximation sera plus ou moins bonne.

La réponse la plus simple à cette difficulté consiste à définir la cohérence d'une hypothèse à partir d'un seuil sur l'erreur. Cependant, de façon très générale, la prise d'une décision symbolique par seuillage possède l'inconvénient grave de donner un comportement instable - et non-intelligent - à proximité du seuil.

Pour résoudre le problème de l'identification d'objets, on s'efforcera alors de trouver des critères "symboliques" de décision. Cela nous oblige à considérer des contraintes plus générales que les conditions de compatibilité du §1.3.3, liées au calcul de θ .

$\theta(IM_{m_i})$ compatible avec ID_{m_i}

et de soumettre la transformation d'une hypothèse non seulement à un ensemble d'équations, mais à un ensemble de contraintes au sens général.

Même dans le cas particulier considéré dans ce chapitre, il est facile de trouver d'autres contraintes utiles. Par exemple, tout modèle impose des inégalités liées à la taille

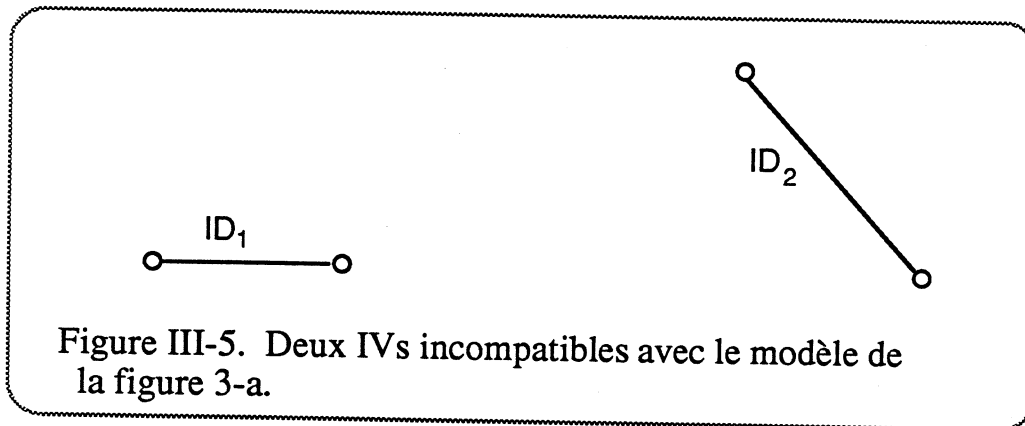


Figure III-5. Deux IVs incompatibles avec le modèle de la figure 3-a.

de l'objet. Dans le cas de la figure III-5 la correspondance

$((IM_1, ID_1) (IM_2, ID_2))$

est incohérente, ID_1 et ID_2 étant trop éloignés l'un de l'autre.

Lorsqu'on utilise différents types d'IVs, les contraintes (numériques et symboliques) deviennent de plus en plus nombreuses. Le bénéfice de l'utilisation de différents types d'IVs (cf. paragraphe 1.6.1) provient de la contrainte symbolique qui impose que deux IVs associés dans une correspondance soient du même type.

Dans le système ACRONYM (cf. §I-2.3.4) nous trouvons un traitement très sophistiqué d'une grande variété de contraintes symboliques et numériques.

2.2 LA COMMUNICATION INTERPRETATION - DESCRIPTION

2.2.1 Motivation

Plaçons maintenant le problème de reconnaissance d'objet dans le contexte d'un système de vision. Dans ce qui précède nous avons traité la reconnaissance en fonction de deux ensembles $\{IM_i\}$ et $\{ID_j\}$ supposés connus au départ. Cela correspond à un système de vision à deux traitements séquentiels avec le paradigme segmentation - interprétation. Nous avons vu au paragraphe I-2.2.1 que cette approche se heurte à plusieurs difficultés:

- * La "bonne" segmentation de l'image est difficile à obtenir. Plus précisément, la segmentation est fonction de nombreux paramètres qu'il est souhaitable d'adapter en fonction de l'interprétation.
- * La segmentation a tendance à produire une quantité importante d'IVs. Elle coûte donc chère en place mémoire, et, si elle est réalisée par logiciel, en temps de calcul.

Même si ces problèmes pouvaient être maîtrisés par le moyen d'un matériel spécialisé pour la segmentation de l'image, la quantité d'IVs noierait l'interprétation dans la combinatoire élevée du problème.

2.2.2 La prédiction - vérification

Dans ce contexte apparaît alors pleinement l'intérêt d'une stratégie d'interprétation par prédiction - vérification. Le principe de la recherche par prédiction donné au paragraphe 1.5.1 se combine avantageusement avec une communication entre description et interprétation.

Dans ces conditions, une méthode de prédiction - vérification procède de la façon suivante:

- * A partir d'un modèle et d'un ensemble $\{ID_j\}$, un module de génération engendre des hypothèses admissibles.
- * Un module de vérification construit les hypothèses les plus plausibles.
- * Un module de décision distribue le contrôle entre description, génération et vérification, et s'arrête lorsqu'une hypothèse d'une plausibilité satisfaisante est trouvée.

On ne suppose pas que l'ensemble $\{ID_j\}$ est complet: chaque prédiction est adressée au module de segmentation, demandant de vérifier la présence des indices prédits. On peut même ne pas supposer que l'ensemble $\{ID_j\}$ soit fini. Ses éléments peuvent être fournis par la description au fur et à mesure que les modules d'interprétation en ont besoin.

L'existence d'une prédiction change complètement le problème de la segmentation. Alors que le problème "trouver tous les IVs dans une image" est en général difficile et coûteux à traiter - on peut même contester qu'il s'agit d'un problème bien défini - le problème "vérifier l'existence d'un IV précis" peut souvent être résolu par des algorithmes relativement simples.

Ce point est particulièrement bien illustré par le système de Yachida et Tsuji décrit au paragraphe I-2.3.8. Ce système ne contient aucune primitive permettant une construction systématique de lignes de contraste, ou de segmentation de lignes de contraste en segments de droite ou arcs de cercle. Par contre, il contient des algorithmes plus faibles, capables de répondre aux questions

"Y-a-t-il un segment de droite entre les points P1 et P2?"
et "Y-a-t-il un arc de cercle de rayon R entre P1 et P2?".

Dans le contexte de la vérification, la segmentation a donc un objectif précis, ce qui simplifie les algorithmes à mettre en oeuvre.

D'autre part, la notion de segmentation exhaustive disparaît.

En ce qui concerne la complexité de l'ensemble du système, il est important de noter que

- * Le temps de vérification est $\sim m$, et inclut des appels à la segmentation.
- * Il existe des algorithmes de segmentation s'exécutant dans un temps proportionnel à la longueur des IVs recherchés. (Par exemple le découpage itératif pour la recherche de segments de droites).

Si la segmentation est effectuée par logiciel, la prédiction - vérification permet de réduire sensiblement les temps d'exécution.

2.3 PREMIER EXEMPLE: LE SYSTEME PVV

Nous avons présenté une formalisation très générale de la notion de reconnaissance et les principes de base justifiant l'intérêt que nous portons à la stratégie de prédiction - vérification. Une implantation de ce formalisme suppose encore de nombreux choix, et peut aboutir à des systèmes sensiblement différents. Dans ce paragraphe, nous donnons un premier exemple avec le système PVV qui utilise les éléments de notre formalisation, complétés par un ensemble cohérent de décisions annexes.

PVV - Prédiction - Vérification en Vision - a été développé dans notre laboratoire par Viviane Souvignier. Après un premier prototype, écrit en POP-2 avec le concours de R. Popplestone dans le cadre d'un DEA [Souvignier 81], la version complète décrite ici a fait l'objet d'une thèse de troisième cycle [Souvignier 83].

D'un point de vue fonctionnel, le système PVV est conçu autour de trois processus liés aux principales structures de données développées dans les paragraphes précédents (figure III-6): la description construisant l'ensemble des IDs, la génération d'hypothèses construisant le graphe réduit, et la vérification d'hypothèses construisant l'hypothèse à plausibilité maximale à partir d'une hypothèse complète du graphe réduit. L'ensemble du système est muni d'une structure de contrôle par but qui décompose le problème "trouver un objet" selon la figure III-7.

2.3.1 Description

PVV effectue une reconnaissance d'objets à base de modèles bidimensionnels. Une phase préalable d'étalonnage, qui établit la correspondance entre le repère image et le repère externe (table de travail), permet d'adapter le système à un point de vue arbitraire. La transformation déterminée par PVV en localisant un objet est alors définie par trois paramètres (rotation - translation dans le plan).

Les IVs utilisés sont des segments de droites. Etant donné que les extrémités des segments ne sont pas significatives avec l'hypothèse de visibilité partielle, une association IM - ID fournit deux équations à partir des équations des droites. Par rapport aux formules du premier chapitre, nous avons donc

$$\begin{aligned}t &= 3 \\p &= 2\end{aligned}$$

et il faut au moins deux correspondances pour déterminer une hypothèse complète:

$$n = 2$$

Le graphe réduit produit par le module de génération contient $m*d$ noeuds, et la génération des hypothèses complètes demande l'analyse d'environ $(m*d)^2/2$ associations. Si le nombre d'IVs est important, cela correspond déjà à un nombre élevé.

La structure de contrôle de PVV a pour objectif de trouver au plus vite une solution, limitant au maximum le nombre d'associations effectivement engendrées. Elle applique une stratégie de décomposition de problème résumée par la figure III-7, et qui consiste à activer chacun des trois modules principaux du système selon les principes suivants.

- Développement limité du graphe réduit.

La construction du graphe réduit demande l'acquisition préalable de quelques IDs par le module de description; le nombre d'ID's initiaux est défini par la stratégie de contrôle. On engendre ensuite toute la partie du graphe réduit découlant des ID's initiaux.

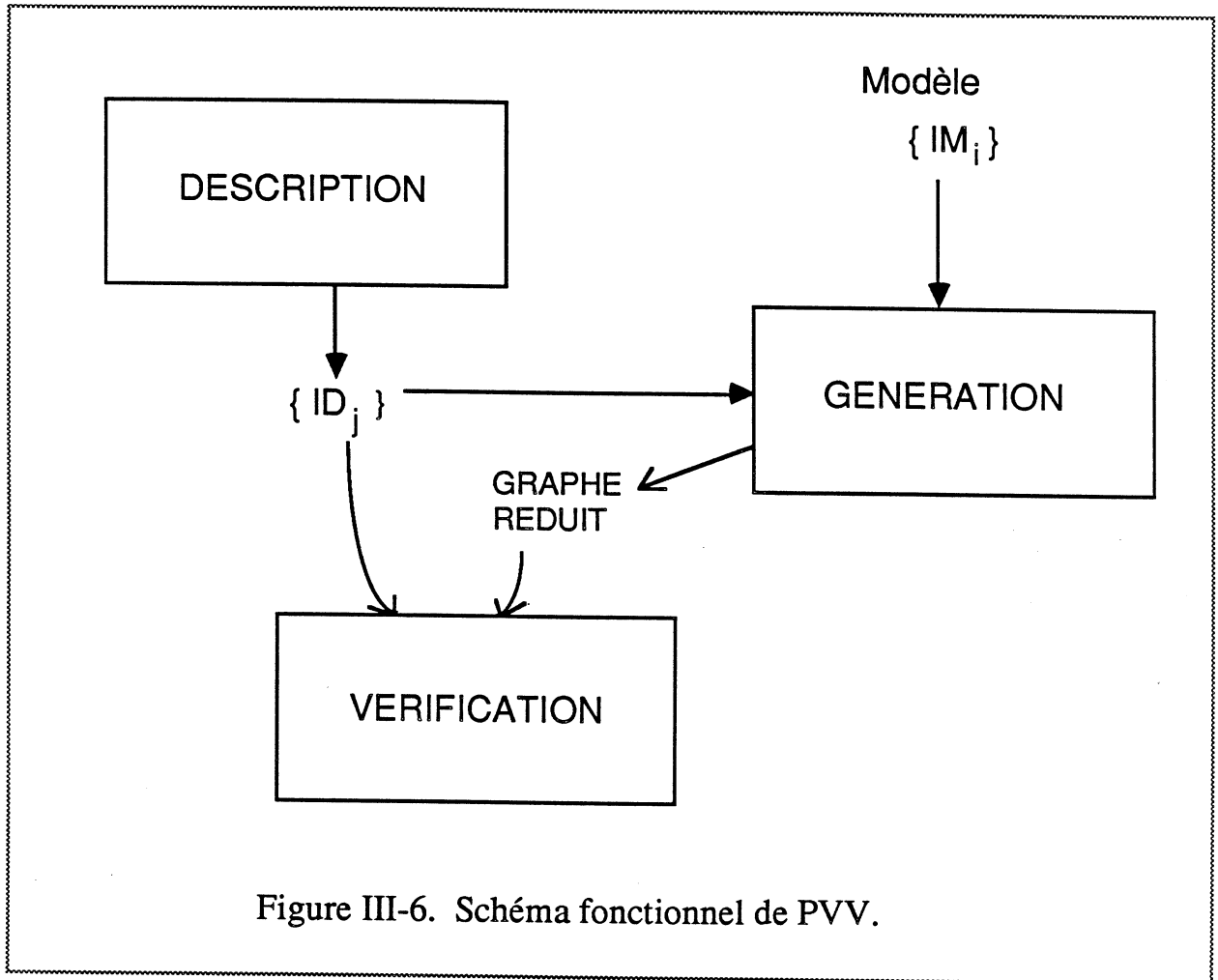
- Vérification.

Les hypothèses complètes engendrées avec le graphe sont ordonnées suivant leur plausibilité, servant ici de fonction heuristique, puis soumis à la vérification. La vérification traite chaque hypothèse, en demandant au module de description de vérifier la présence des indices absents de l'hypothèse, jusqu'à une décision définitive. Si la plausibilité devient suffisante, l'hypothèse est retenue comme solution par le système; si la description ne trouve pas assez de IDs confirmant l'hypothèse, celle-ci est rejetée en fonction d'une valeur de validité. La validité, en l'occurrence le compte du nombre d'échecs de confirmation d'indices, sert donc d'heuristique d'abandon d'une hypothèse.

- Traitement d'échecs.

Un échec à un niveau quelconque de la hiérarchie des buts amène à considérer l'alternative au niveau supérieur.

- a) Un échec de vérification d'un IV est pris en compte par le module de vérification.
- b) L'échec de vérification d'une hypothèse complète entraîne la vérification de l'hypothèse suivante.
- c) L'échec de vérification de toutes les hypothèses entraîne la génération d'une nouvelle partie du graphe réduit, ce qui peut nécessiter l'acquisition de nouveaux IDs par le module de description.
- d) Il y a échec final lorsque le graphe réduit défini par l'ensemble des IDs fournis par la description a été exploré.



La figure III-8 montre les principales étapes du traitement d'un exemple.

S'il veut améliorer la performance du système, l'utilisateur dispose d'un certain nombre de paramètres ad-hoc permettant de limiter le nombre d'associations considérées en restreignant la généralité de la reconnaissance. Cela concerne, par exemple, l'interdiction de cas extrêmes d'alignement, ou l'utilisation d'IDs trop courts.

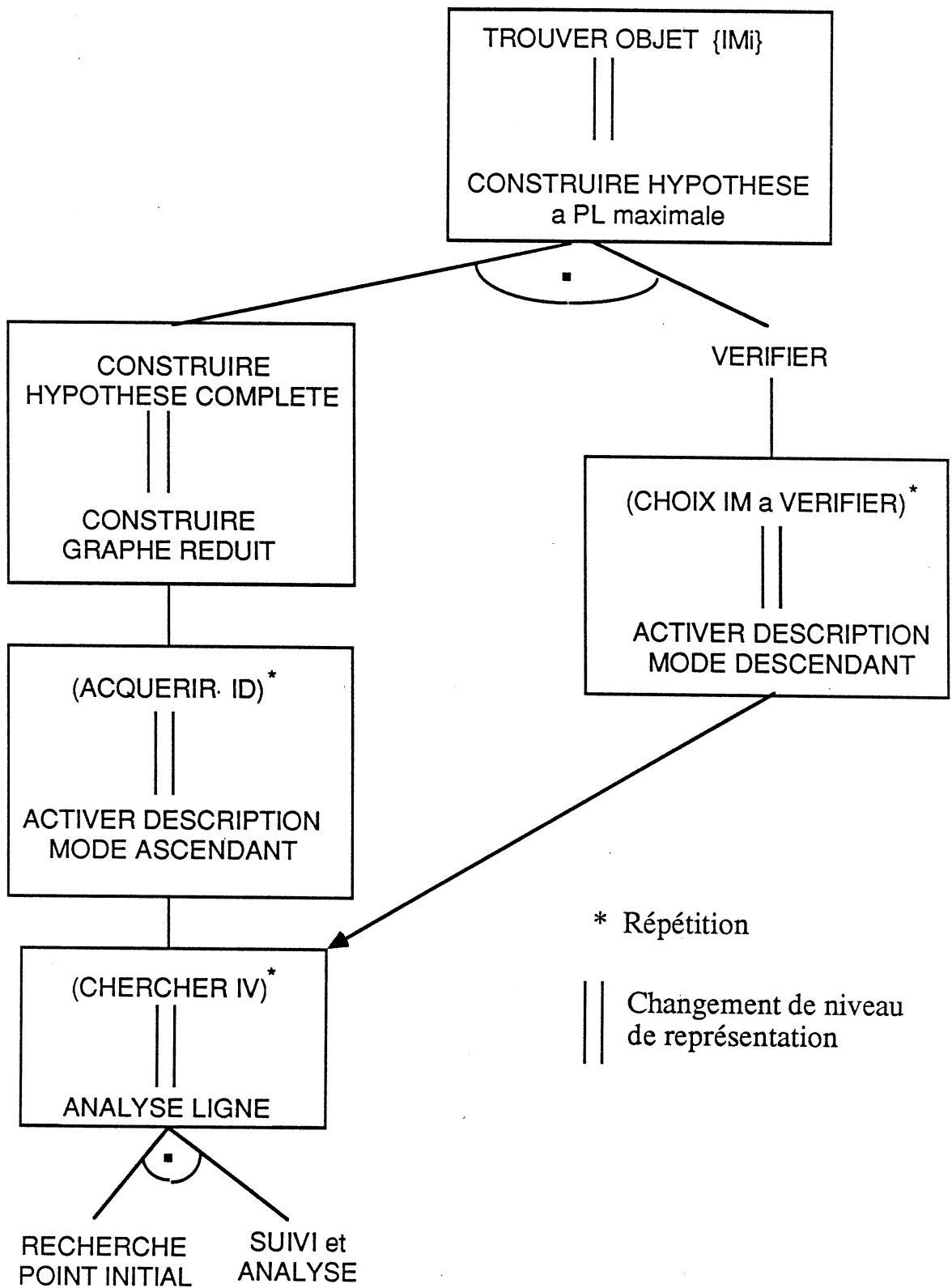
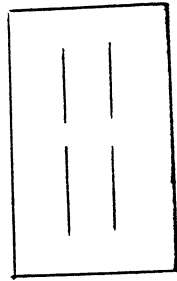
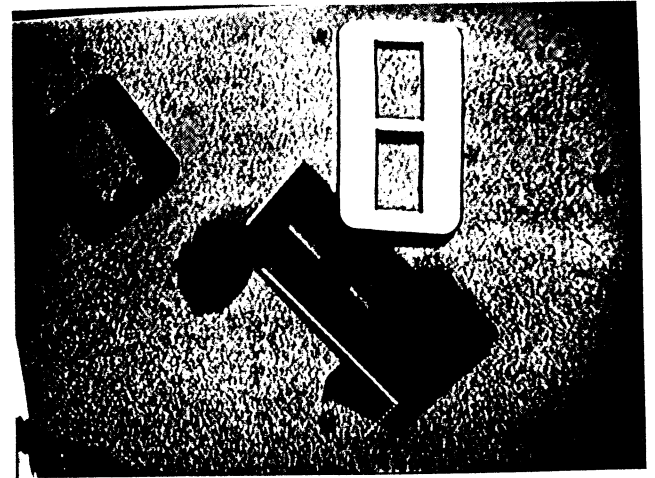


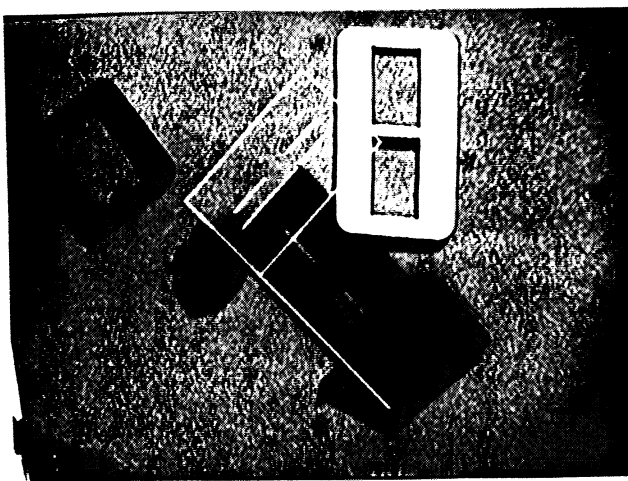
Figure III-7. Décomposition d'un problème dans PVV.



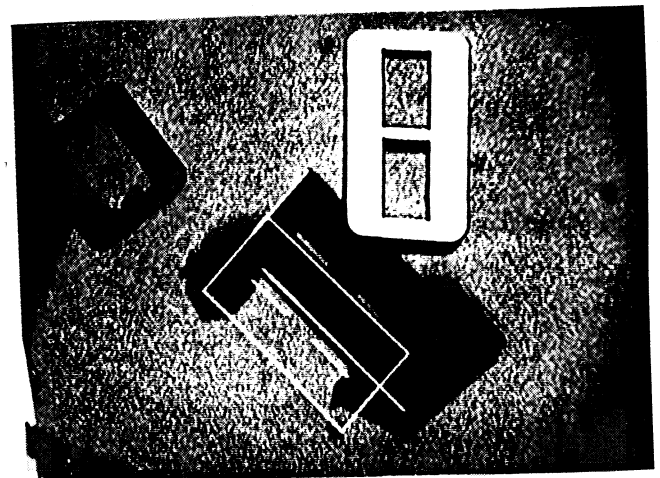
a) Modèle composé de 8 IVs.



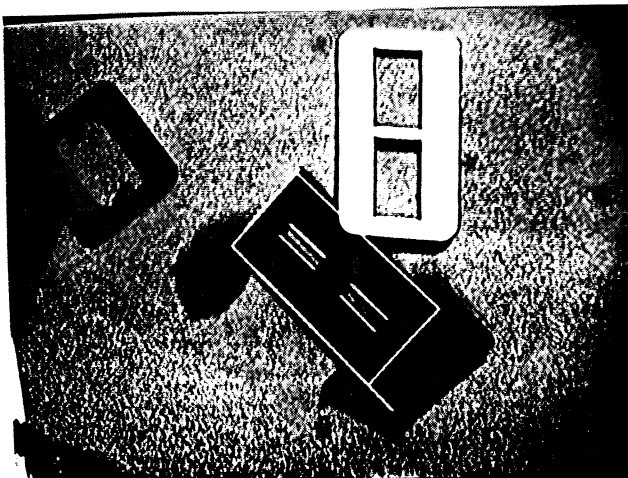
b) IVs initiaux: trois segments de droites.



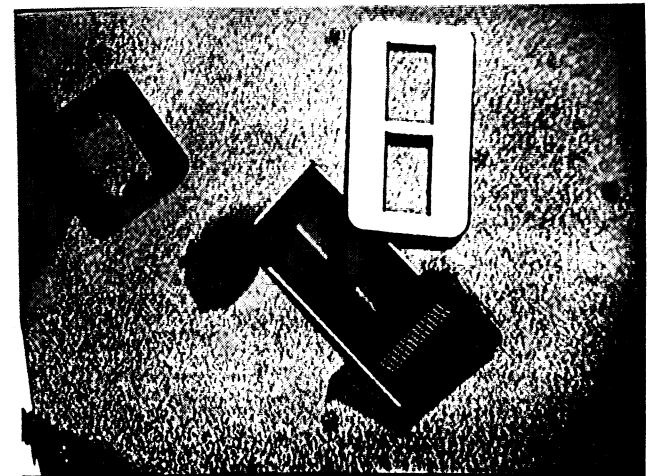
c) Hypothèse complète trouvée à partir des indices initiaux.



d) Une autre hypothèse complète.



e) L'hypothèse complète la plus plausible (étant donnés les IVs initiaux).



f) Zone de recherche d'un IV prédit à partir de l'hypothèse e).

Figure III-8. Exemple de fonctionnement de PVV [Souvignier83].

2.3.2 Le module de description et le dialogue PVV - CAIMAN

Le module de description de PVV est constitué par CAIMAN, qui fonctionne ici comme un ensemble de sous-programmes. Dans le contexte de PVV, le module de description est appelé avec un objectif précis et bien délimité.

- * Appelé depuis le module de génération d'hypothèses, la description doit fournir un nombre limité d'indices image. Ces indices ne sont soumis à aucune contrainte, et sont recherchés à partir de différents endroits de l'image.
- * Appelé depuis le module de vérification, la description doit fournir le ou les indices image nécessaires à la vérification: elle doit alors scruter une région précise de l'image pour y chercher des indices fortement contraints.

Dans les deux cas, les algorithmes incrémentaux de CAIMAN permettent d'atteindre l'objectif sans analyse superflue - seulement des parties pertinentes de l'image sont effectivement analysées.

D'autre part, les données de la tâche de reconnaissance d'objet permettent de fixer les nombreux paramètres existants dans CAIMAN. Par exemple, lors de la recherche des IDs initiaux, on imposera des valeurs assez fortes au gradient des lignes, et au paramètre MINLO pour la recherche de segments de droites. L'objectif étant de trouver des segments initiaux significatifs, d'assez grande taille et peu influencés par du bruit. Par contre, pendant la phase de vérification, on peut souhaiter trouver des indices même bruités et donner aux paramètres des valeurs peu restrictives.

2.3.3 Evaluation

L'ambition de PVV est de combiner l'utilisation d'un formalisme général et complet avec les critères de performance des systèmes industriels, comme ceux du §I-2.3.8.

- D'une part, PVV est un outil de recherche qui a été réalisé comme un système interactif en extension de CAIMAN. Des commandes spécifiques, divers paramètres et des possibilités de traces modulées permettent d'étudier au niveau de détail voulu chaque aspect du système: construction d'hypothèses, calcul de transformations, vérification d'hypothèses, échecs.

- D'autre part, PVV peut être directement connecté à un programme de commande de robots écrit en LM, et fonctionne alors comme une machine de vision au niveau objet. Dans la configuration matérielle dont nous disposons au LIFIA, les performances sont limitées. Le temps d'exécution est dominé par le temps de lecture sur disque des branches du programme, et par le temps de calcul de gradient. Avec une structure matérielle adaptée, le temps d'exécution pour un exemple comme celui de la figure III-8 est de l'ordre de 1 seconde. Il faut pour cela une unité centrale disposant d'au moins 64K mots de mémoire centrale (comme un HP1000), et un matériel spécialisé pour l'extraction de points de contraste, comme le GTR.

Une originalité spécifique de PVV vient du fait qu'il implémente effectivement la liaison interprétation - description décrite au paragraphe 2.2. Une prédiction au plus haut niveau dans la hiérarchie des buts définit une structure de contrôle à tous les niveaux inférieurs, jusqu'à la gestion de fenêtres de l'image au niveau pixel. C'est cette liaison qui justifie l'intérêt d'un contrôle incrémental dans CAIMAN (cf. I-3.4).

2.4 Comparaison avec d'autres systèmes

A première vue PVV répond au mêmes spécifications qu'un grand nombre de systèmes de vision de type industriel, dont nous avons présenté certains au §I-2.3.8. Toutefois, parmi ces systèmes PVV est le seul à être fondé sur la définition de reconnaissance du chapitre 1, et à explorer de façon systématique le graphe d'hypothèses. Cela confère à PVV une propriété de complétude - l'objet sera reconnu même en présence des plus faibles indices - au prix d'une plus forte combinatoire. PVV cherche à limiter cette combinatoire par le choix d'un domaine bidimensionnel, et par une stratégie de contrôle sophistiquée. Cependant, le cas le plus défavorable fait apparaître un besoin en taille mémoire et en temps d'exécution considérable.

D'autres systèmes de vision réduisent la combinatoire de l'espace des hypothèses par un aménagement de la notion de reconnaissance; ils n'ont alors pas la propriété de complétude selon nos définitions. Cela signifie que dans certains cas de visibilité partielle l'objet ne sera pas reconnu, ce qui peut être parfaitement acceptable dans le cas d'une application industrielle. Par exemple, [Ayache 83] suppose que certains IVs principaux sont entièrement visibles dans l'image, ce qui élimine les hypothèses incomplètes ($n=1$).

Un autre système intéressant à cet égard est celui basé sur la méthode LFF (§2.3.8.3): il réduit la combinatoire de l'espace des hypothèses en effectuant une reconnaissance à deux niveaux.

Au niveau supérieur, chaque groupement local d'indices joue le rôle d'IV, et est représenté par l'indice central. Les associations à considérer concernent donc seulement les indices centraux, ce qui revient à poser, dans les formules d'évaluation, une valeur $m' \ll m$.

Une seule association au niveau supérieur implique une contrainte forte: que les autres indices du groupement (au moins certains) puissent être associés avec des IDs se trouvant dans une fenêtre de l'image définie par l'indice central, et que toutes ces associations soient géométriquement cohérentes. Ceci définit le niveau inférieur de reconnaissance, dans lequel le groupement d'indices joue le rôle de modèle, et auquel intervient une partie seulement et des IMS et des IDs. La recherche d'une hypothèse à plausibilité maximale au niveau inférieur est réalisée par la recherche de cliques maximales.

La reconnaissance au niveau inférieur définit des hypothèses complètes, ce qui élimine complètement les hypothèses incomplètes de l'espace de recherche au niveau supérieur (c'est à dire $n=1$).

En contrepartie, dans certains cas de visibilité partielle la reconnaissance échoue: il faut que pour au moins un groupement, l'indice central et un nombre suffisant d'indices secondaires soient visibles.

2.5 DEUXIEME EXEMPLE: ANALYSE DE DONNEES 3-D

Lorsqu'on se place dans un univers 3-D, la taille de l'espace des hypothèses augmente considérablement; le nombre de paramètres d'une transformation étant de 6, on est obligé de considérer des hypothèses incomplètes, toujours très nombreuses. Dans un travail récent, [Faugeras et Hébert 85] utilisent le graphe des hypothèses construit à partir d'indices 3-D (que l'on peut obtenir notamment par l'analyse d'images de distances) avec une recherche liant l'identification à la localisation par la solution approchée d'un système d'équations. Les indices considérés sont des points, des segments de droites, et des faces planes. Afin de réduire le nombre d'hypothèses incomplètes ils analysent la cohérence de celles-ci. L'analyse géométrique du problème, assez simple dans le cas 2-D, devient complexe dans le cas

3-D, et conduit à l'utilisation d'algorithmes sophistiqués.

Lozano-Perez et Grimson ont proposé l'utilisation d'un "arbre d'interprétations" (IT = interpretation tree) proche du graphe d'hypothèses pour la reconnaissance 3-D à partir d'une faible quantité de données capteurs [Lozano-Perez et Grimson 84, Grimson et Lozano-Perez 84]. Ils situent leur travail dans le contexte suivant: des capteurs fournissent la position et éventuellement l'orientation (normale à la surface) d'un petit nombre de points sur la surface d'un ou de plusieurs objets. Ces données peuvent provenir, par exemple, d'un capteur tactile, d'un système de vision 3-D, ou d'un capteur à ultrasons. Disposant d'un modèle géométrique de l'objet, l'objectif est l'identification et la localisation de l'objet dans le nuage de points.

Lozano-Pérez et Grimson considèrent des modèles polyédriques et proposent de construire un arbre d'hypothèses à d niveaux de la façon suivante: au niveau 0 la racine contient la correspondance vide; les descendants d'un noeud de niveau k ($k=0 \dots d-1$) sont les m associations (IM_i, ID_{k+1}) entre un ID de type point et un IM de type face plane. Un chemin dans cet arbre définit une séquence de correspondances de plus en plus informées, auxquelles on associe des contraintes pour former des hypothèses. La transformation T doit faire coïncider un point quelconque de la face-modèle avec le point-données. Dans ces conditions, le nombre d'hypothèses incomplètes est important. L'utilisation de contraintes géométriques simples permet cependant de déterminer l'incohérence d'hypothèses incomplètes dans un grand nombre de cas, et de les éliminer.

- * La distance entre deux points-données est contrainte par les faces du modèle.
- * L'angle défini par les orientations de deux points-données doit être compatible avec l'angle entre les faces.
- * L'orientation des points sur un chemin entre deux points-données (que l'on obtient naturellement avec un capteur tactile) doit être compatible avec le modèle.

L'implantation réalisée concerne pour l'instant le cas 2-D et montre une certaine parenté avec le système PVV. Notons cependant qu'il ne s'agit pas d'un système de vision proprement dit, puisque la donnée initiale n'est pas une image.

L'avantage de l'arbre d'interprétations par rapport au graphe des hypothèses vient du fait qu'il ne développe pas toutes les correspondances, et qu'un arbre est plus facile à gérer qu'un graphe. En contrepartie, cette formulation s'appuie sur un ordre fixe entre les IDs (il y a $d!$ ITs différents), ce qui donne un certain biais à la construction d'hypothèses, et elle exclut l'association d'un indice données avec plusieurs indices modèle (alignement accidentel, cf. §1.2.4), ce qui

limite la méthode aux IVs de type point.

2.6 COMPARAISON AVEC D'AUTRES ALGORITHMES DE RECONNAISSANCE

Le problème de reconnaissance d'objet, tel que nous l'avons énoncé au paragraphe 1.3.2, peut être résolu par des algorithmes sensiblement différents de celui que nous avons présenté. Ce paragraphe a pour objectif de situer trois autres solutions par rapport à la prédiction - vérification: la recherche de cliques maximales, la méthode de Hough, et la propagation de contraintes.

2.6.1 Recherche de cliques maximales

L'utilisation de cliques maximales pour la reconnaissance d'objets a été proposée par [Ambler 75] et est pratiquée en particulier par [Bolles et Cain 82] avec la méthode LFF que nous venons déjà d'évoquer au paragraphe précédent.

Principe

Etant donné les ensembles $\{IM_i\}$ et $\{ID_j\}$, on construit un graphe de la façon suivante:

- * les noeuds sont les correspondances de longueur 1, c'est à dire les associations (IM_{m_i}, ID_{j_i}) .
- * un arc entre deux noeuds indique que les deux associations sont compatibles. Dans le cas de la méthode LFF, deux associations sont considérées compatibles si elles définissent deux hypothèses équivalentes.

En théorie de graphes, une clique est un sous-graphe dont tous les noeuds sont connectés entre eux. Une clique de s noeuds, comporte donc $(s*s-1)/2$ arcs. Une clique maximale du graphe ainsi construit met en évidence un ensemble maximal d'associations compatibles deux à deux, ce qui équivaut aux hypothèses à plausibilité maximale. Cette méthode est justifiée par Bolles et Cain par l'utilisation d'un algorithme simple et très rapide pour la recherche de toutes les cliques maximales.

Analyse et évaluation

Cette méthode comporte deux étapes bien distinctes: construction du graphe et recherche de cliques maximales.

La construction du graphe crée d'abord $m*d$ noeuds, et doit effectuer $m*d*(m*d-1)/2$ calculs de compatibilité deux à deux pour la création des arcs. Dans ce contexte, la notion de compatibilité est nettement plus restrictive que celle que nous avons utilisée puisqu'il faut que la compatibilité d'associations prises deux à deux soit équivalente à la compatibilité globale. Nous considérons le cas le plus simple dans lequel les correspondances de longueur 1 définissent une transformation complète ($n=1$). Dans ce cas, la recherche de cliques maximales, comme la prédiction - vérification, résoud le problème de recherche de l'hypothèse à plausibilité maximale à partir d'une hypothèse complète, c'est à dire le problème 2 du §1.5.

La construction du graphe demande alors $m*d$ solutions de systèmes de t équations, et $m*d*(m*d-1)/2$ comparaisons de transformations.

Le coût de la recherche des cliques maximales dépend de plusieurs facteurs: de la taille du graphe, du nombre de cliques, c'est à dire du nombre d'hypothèses d'objet distinctes, et de la taille des cliques, qui est bornée par le nombre d'indices modèle. Dans le cas considéré, une seule clique maximale sera vraiment importante. Le coût de la recherche des cliques sera marqué par le fait qu'il faut analyser au moins une fois chaque arête du graphe, et il sera du même ordre que celui de la construction du graphe. La complexité algorithmique de cette méthode est donc essentiellement $(m*d)**2$ en temps d'exécution aussi bien qu'en place mémoire (pour la représentation des arcs).

Il y a donc des différences significatives par rapport à l'algorithme employé par PVV:

- * La construction des $m*d$ noeuds du graphe est équivalente aux $m*d$ calculs de transformations nécessaires à la détermination des hypothèses complètes. Par contre, PVV n'inclut aucune opération équivalente aux $(m*d)*(m*d-1)/2$ comparaisons pour la création des arcs.
- * Le calcul d'une clique maximale nécessite au plus la vérification de $m*(m-1)/2$ arcs, alors que le coût de vérification d'une hypothèse par PVV est proportionnel à m .

Dans le contexte de la méthode LFF, ces différences importent peu dans la mesure où la recherche des cliques intervient seulement au deuxième niveau de reconnaissance, avec un nombre d'IVs réduit par une prédiction venant du premier niveau.

D'un point de vue plus fondamental, nous constatons que la recherche des cliques maximales possède plusieurs inconvénients par rapport à la prédiction - vérification:

- * elle est applicable seulement s'il n'y a pas d'hypothèses incomplètes.
- * elle possède une plus forte complexité, et demande un nombre plus élevé de calculs de transformations, qui ont tendance à être relativement coûteux en temps.
- * elle utilise les ensembles $\{IM_i\}$ et $\{ID_j\}$ complets et ne s'intègre pas naturellement dans un dialogue entre description et interprétation.

2.6.2 La méthode de Hough généralisée

Dans notre modèle de reconnaissance, une hypothèse cumule les contraintes globales provenant des évidences locales à différents endroits d'une image, ce qui permet de construire des hypothèses pour des vues fortement dégradées. Nous trouvons ici la caractéristique la plus importante de la méthode de Hough, dont le principe, décrit au §II-2.3.3 pour la recherche de segments de droites, se généralise à la recherche de toute forme géométrique, y compris de modèles d'objets (cf. [Ballard 81]).

Il est en effet possible de construire une parallèle entre la méthode de Hough et la reconnaissance par le graphe des hypothèses.

- * Les hypothèses incomplètes du graphe sont une représentation d'un ensemble de positions E_i (dans la terminologie du paragraphe II-3.3.8).
- * Les arcs du graphe explicitent l'ordre dans lequel on considère les intersections entre ensembles E_i .
- * Les hypothèses complètes du graphe correspondent aux cases portant une valeur élevée du tableau d'accumulation. (Plus exactement à un "nuage" de cases, puisqu'une hypothèse peut être discrétisée sur plusieurs cases voisines). La valeur contenue dans une case joue précisément le rôle d'une fonction plausibilité.

Nous considérons alors les deux méthodes comme deux formulations algorithmiques différentes, fondées sur le même modèle de la reconnaissance d'objets à base d'indices visuels. Chaque méthode représente explicitement certains aspects de la solution, ce qui lui confère une combinaison d'avantages et d'inconvénients.

- Le graphe d'hypothèses représente explicitement toutes les hypothèses possibles et les relations entre elles. Il permet la mise en oeuvre d'une stratégie de reconnaissance, c'est à dire un raisonnement global par rapport à l'ensemble de solutions, comme par exemple la prédiction - vérification. Par contre, l'ensemble des hypothèses est fortement combinatoire; l'élimination des hypothèses incohérentes, qui forment en général la plus grande partie du graphe, demande l'exploitation des contraintes inhérentes à une hypothèse, pour laquelle il n'existe pas de méthode générale.

- La méthode de Hough traduit explicitement, en termes de positions possibles, toutes les contraintes géométriques dues à une association. Elle met donc directement en évidence les hypothèses cohérentes. Par contre, le tableau d'accumulation les représente sans relation les unes avec les autres, il ne donne aucune structure globale au problème. Nous avons aussi vu les difficultés fondamentales liées à la mise en oeuvre de la méthode de Hough à l'occasion du paragraphe II-3.3.8. Notons également que cette méthode ne concerne pas les contraintes symboliques et non-géométriques que nous discuterons au chapitre suivant.

2.6.3 Propagation de contraintes

La propagation de contraintes, telle que nous l'avons décrite à propos du travail de Waltz (I-2.1.3.2) et de MSYS (I-2.3.2) utilise un graphe différent du graphe des hypothèses. Elle traite un cas de reconnaissance dans lequel des contraintes symboliques peuvent être exploitées localement. On travaille dans les conditions suivantes:

- * Le modèle est défini par un graphe dont les noeuds sont les indices IM_i , les arcs correspondant à une relation de voisinage.
- * Une association (IM_p, ID_q) impose des contraintes sur les associations possibles des indices voisins de IM_p .
- * Toutes les contraintes propres à une correspondance s'expriment ainsi.

Dans ces conditions, un algorithme de propagation de contraintes, comme celui introduit par Waltz, permet de trouver tous les étiquetages cohérents.

Cette approche est fondée sur une formalisation de la notion de reconnaissance différente de celle que nous avons présentée: elle identifie un objet en fonction de son contexte, et non en fonction des relations entre ses parties.

CHAPITRE 3

UTILISATION DE CONNAISSANCES: LA GENERATION DE PLANS

L'approche algorithmique développée dans les chapitres 1 et 2 fait apparaître une forte combinatoire du problème de reconnaissance, qui semble être en contradiction avec notre intuition. Nous expliquons cette contradiction par le fait que la vision humaine intègre dans la reconnaissance d'innombrables connaissances concernant la structure du monde, qui se traduisent par des contraintes pour le processus de reconnaissance. Nous considérons donc l'aspect IA de la vision. Sans proposer un modèle général pour un tel mécanisme, ce chapitre décrit des idées existantes, et les relie aux développements précédents.

3.1 Motivation

Nous avons vu que la technique de prédiction - vérification permet de réduire le coût du traitement d'une hypothèse complète; ce coût est proportionnel à la "taille" du modèle. Par contre, nous n'avons présenté aucune technique réduisant le nombre d'hypothèses incomplètes correspondant au problème 1 du paragraphe 1.5 et dont le nombre est essentiellement $(m*d)^n$. Or, si $n > 1$, ce nombre devient important. Une voie intéressante à cet égard apparaît au paragraphe 2.2 avec l'introduction de contraintes symboliques. En ce qui concerne la construction d'hypothèses, une contrainte symbolique peut servir de filtre éliminant des associations, sans connaissance complète de la transformation géométrique à appliquer.

Il s'avère que plus une scène est complexe, et plus l'ensemble d'indices visuels utilisé est riche (c'est à dire plus la description de l'image est variée), plus on peut introduire de telles contraintes. Par contre, ces contraintes sont très diverses, et leur traitement paraît ad-hoc.

Cette difficulté est prise en compte dans une description de la situation dans la terminologie de l'IA. On dispose de

connaissances variées concernant les objets dans une scène, leur agencement, les relations entre objets. Ces connaissances impliquent des contraintes de toutes sortes concernant l'image. Le problème est alors le suivant: à partir d'un ensemble de connaissances, déduire un ensemble de contraintes utiles pour un problème de vision particulier, et les intégrer dans le système de vision.

Si cette tâche est résolue par un système à deux étapes, au sens du paragraphe I-1.3.3, cela revient à produire automatiquement un programme de reconnaissance d'un objet à partir d'un ensemble de connaissances. Nous décrivons dans ce chapitre un système de vision fondé sur ce principe - le programme de [Garvey 76].

3.2 EXEMPLE: LE SYSTEME DE GARVEY

3.2.1 Vue d'ensemble

Le système de Garvey rentre bien dans le schéma général "modélisation - analyse statique - description - interprétation" du paragraphe I-2.2.3. L'aspect prédominant est cependant l'analyse statique, c'est à dire la synthèse du programme de reconnaissance. Garvey utilise les concepts et la terminologie de la génération de plans d'actions, qui est un problème de même nature que la synthèse de programmes. Avec cette terminologie, l'interprétation de l'image s'effectue par l'exécution du plan.

- * Modélisation.
Un objet est modélisé par des propriétés de pixels, et par des relations dans l'image avec d'autres objets.
- * Description.
Les modules de description permettent d'extraire de l'image des pixels ayant certaines propriétés, et de vérifier des propriétés.
Dans ce système la description de l'image est entièrement fonction de l'objectif. D'autre part, elle se situe seulement au niveau pixel. Il n'y a pas de construction d'indices visuels plus complexes.
- * Analyse statique.
A partir des modèles d'objets et de l'objectif - qui est toujours la localisation d'un objet par encadrement dans l'image - le système construit un plan. Ce plan décompose le but global hiérarchiquement en un ensemble de sous-buts, puis en appels de modules de description.

* Interprétation.

L'exécution du plan fait appel aux primitives de description qui sont interprétées par rapport à l'objectif.

Un aspect original de l'interprétation concerne le choix dans le plan de la meilleure primitive. Etant donné que la modélisation d'objets fait appel à des primitives nombreuses, simples, mais peu discriminantes, il y a plusieurs possibilités de localiser un objet, donnant des résultats plus ou moins surs avec des temps d'exécution sensiblement différents. Le plan représente l'ensemble de ces possibilités, et l'interprétation doit effectuer des choix en fonction de la qualité et du coût des résultats.

3.2.2 Modélisation de l'univers

Les objets sont définis suivant la grammaire suivante:

```
objet ::= objetsimple | objetcomposé | objet + relations
objetsimple ::= faces + relations
objetcomposé ::= objetsimples + relations
```

Les faces sont décrites par des "attributs de pixels". Les relations sont définies dans une base de données. La figure III-9 donne la description de quelques objets, utilisés pour les exemples cités dans ce paragraphe.

3.2.2.1 Attributs de pixels

Les attributs de base dont on dispose pour chaque pixel sont

- couleur
- distance
- réflectivité
- brillance
- saturation
- orientation de surface
- hauteur

Les faces d'un objet sont décrites par des prédicats à partir de ces attributs de base. Un système interactif - ISIS - permet de créer ces prédicats à partir d'une image de l'objet.

Exemple: lors d'une "séance d'apprentissage" l'utilisateur a indiqué des pixels appartenant à la table; le système a alors

Object	Description	Relations
TABLETOP	BGHT: 18 to 26 HUE: 26.0 to 58.0 SAT: .23 to .32 HEIGHT: 26.0 27.5 ORIENT: -7.0 to 5.5	SUPPORTS TELEPHONE .6 SUPPORTS BOOK .4 IN-FRONT-OF WALL 1.0
TELEPHONE	BGHT: 4 to 8 HUE: 72.0 125.0 SAT: .15 to .22 HEIGHT: 5.0 to 6.0 ORIENT: -20. to 95.0	SUPPORTED-BY TABLETOP 1.0 IN-FRONT-OF WALL 1.0
PICTURE	BGHT: 7 22 HUE: 15.9 85.6 SAT: .15 .398 HEIGHT: 35.0 47.0 ORIENT: 82.0 92.0	IN-FRONT-OF WALL 1.0 ABOVE TABLETOP 1.0
SEAT	BGHT: 7 to 12 HUE: 110.0 to 146.0 SAT: .42 to .47 HEIGHT: 14.0 15.0 ORIENT: -15.0 to 10.0	BELOW BACK1 .4 BELOW BACK2 .6 IN-FRONT-OF WALL 1.0
BACK1	BGHT: 6 to 9 HUE: 115.0 to 130.0 SAT: .4 to .45 HEIGHT: 18.0 to 28.0 ORIENT: 75.0 to 90.0	ABOVE SEAT 1.0 IN-FRONT-OF WALL 1.0
BACK2	BGHT: 8 to 14 HUE: 75.0 to 240.0 SAT: .24 to .31 HEIGHT: 18.0 to 28.0 ORIENT: 60.0 90.0	ABOVE SEAT 1.0 IN-FRONT-OF WALL 1.0

Figure III-9. Modélisation d'objets dans [Garvey 76].

enregistré les valeurs extrêmes des attributs des ces pixels, ce qui définit les prédicats

(HUEEXT TABLETOP)
(HEIGHTEXT TABLETOP)

(HUEEXT TABLETOP) fournit l'intervalle des teintes rencontrées dans cette exemple, (HEIGHTEXT TABLETOP) donne l'intervalle des hauteurs. Le "prédicat caractéristique" de la table devient

```
(AND (COLORP (HUEEXT TABLETOP)
              (HUE X))
      (LIMITP (HEIGHTEXT TABLETOP)
              (HEIGHT X)))
```

La fonction COLORP vérifie l'appartenance d'une teinte à un intervalle, LIMITP vérifie un interval de hauteur.

3.2.2.2 Relations dans une scène

Les relations considérées sont des relations spatiales comme "adjacent à", "posé sur", "à gauche de". Dans cette description, il y a une certaine ambiguïté entre relations dans l'image et relations dans la scène.

3.2.3 Description de l'image

Les modules de description de l'image sont étroitement liés à la stratégie globale de localisation d'un objet par les trois étapes acquisition, validation, encadrement.

1) Acquisition.

L'étape d'acquisition fournit des premières hypothèses concernant l'emplacement de l'objet en utilisant deux primitives de filtrage:

1-a) Echantillonnage - filtrage d'une fenêtre. Une fenêtre de l'image est parcourue avec un pas d'échantillonnage (fonction de la taille de l'objet cherché), et les pixels sont soumis à un test. Le résultat est constitué de la liste de pixels ainsi filtrés.

1-b) Filtrage le long d'une droite. Etant donné un point et une direction, on cherche le premier pixel possédant certains attributs. Ce type de primitive est utilisé, par exemple, pour chercher le bord d'un objet à partir un pixel intérieur.

2) Validation.

Dans la phase de validation on cherche à discriminer les pixels de l'objet recherché d'autres objets ayant certaines caractéristiques en commun.

Cela se fait à l'aide de prédicats plus complexes, donc plus coûteux en temps d'exécution, et de primitives d'analyse de contours.

3) Encadrement.

Cette étape produit le contour précis de l'objet. Elle fait appel à trois types de primitives.

3-a) Des procédures spécialisées existent pour certaines formes particulières, comme un rectangle horizontal, un rectangle vertical.

3-b) S'il existe un modèle géométrique de l'objet, sa projection sur la scène produit un masque pouvant être vérifié.

3-c) Dans le cas général on peut appliquer une procédure de croissance de région à partir des zones validées, et déterminer le contour par l'enveloppe convexe.

3.2.4 Génération de plans et exécution

3.2.4.1 Vue d'ensemble

Le système résout un problème de vision en trois étapes qui seront développées en détail dans les paragraphes suivants.

1) Génération de plan.

A partir du problème posé et du modèle de l'univers, incluant en particulier des modèles d'objets et des relations entre objets, un plan est construit. Le plan est obtenu par décomposition hiérarchique du problème. Il représente un programme résolvant le problème, ainsi que le raisonnement par lequel le programme a été obtenu.

2) Evaluation statique.

Un plan contient effectivement plusieurs solutions du problème, et l'ordre sur les actions n'est pas entièrement fixé. L'évaluation statique attache à chaque noeud du plan une mesure numérique de "qualité" permettant de classer les alternatives et de linéariser le plan.

3) Exécution.

Les actions de la meilleure alternative sont exécutées une par une. A chaque pas d'exécution l'évaluation du plan restant est mise à jour, mettant en cause

l'alternative courante en fonction des événements.

La figure III-10 montre l'exemple d'un plan. Un cas d'exécution de ce plan est décrit au paragraphe 4.2.4.5.

3.2.4.2 Génération de plans

Construction d'un plan

La génération de plan part d'un but initial à partir duquel est développé un ensemble de sous-buts, composant la solution du problème. Les sous-buts sont à leur tour considérés récursivement. La récursion s'arrête sur des buts primitifs. Un but primitif peut être soit un but exécutable, soit un but insatisfaisable. Un but insatisfaisable correspond à un problème insoluble et ne doit pas faire partie d'un plan final. Un but exécutable contient l'appel d'une procédure de description.

Le but initial est toujours de la forme (FIND objet). La décomposition type produit trois sous-buts: acquisition, validation, encadrement (cf.§3.2.3). La phase la plus complexe est l'acquisition. Elle peut se faire directement, à partir de propriétés des pixels attachées à l'objet, ou indirectement, en fonction des relations de l'objet dans la base de données.

Par exemple, la relation "objet posé sur table" amène à la décomposition

- 1) Acquisition table
- 2) Acquisition objet dans une fenêtre de l'image définie par la position de la table.

Un sous-but peut introduire un nouvel objet. Par exemple, la recherche du téléphone peut passer par la recherche de la table. Le générateur de plan gère une liste globale des objets figurant dans le plan. Les informations concernant chaque objet sont initialisées à partir du modèle fourni par la description de l'univers.

Si l'objet introduit par un sous-but figure déjà dans le plan, il n'y a pas génération d'un nouveau noeud, mais création d'un pointeur vers le noeud de cet objet. Cela peut introduire des boucles dans un plan. Avant d'engendrer des nouveaux sous-buts, le générateur vérifie l'existence de modules spécialisés dans le type du problème courant. Cela permet à l'utilisateur de fournir des "conseils" pour certains problèmes. En absence de programmes spécialisés, on fait appel à des méthodes standard qui sont utiles pour tous les objets.

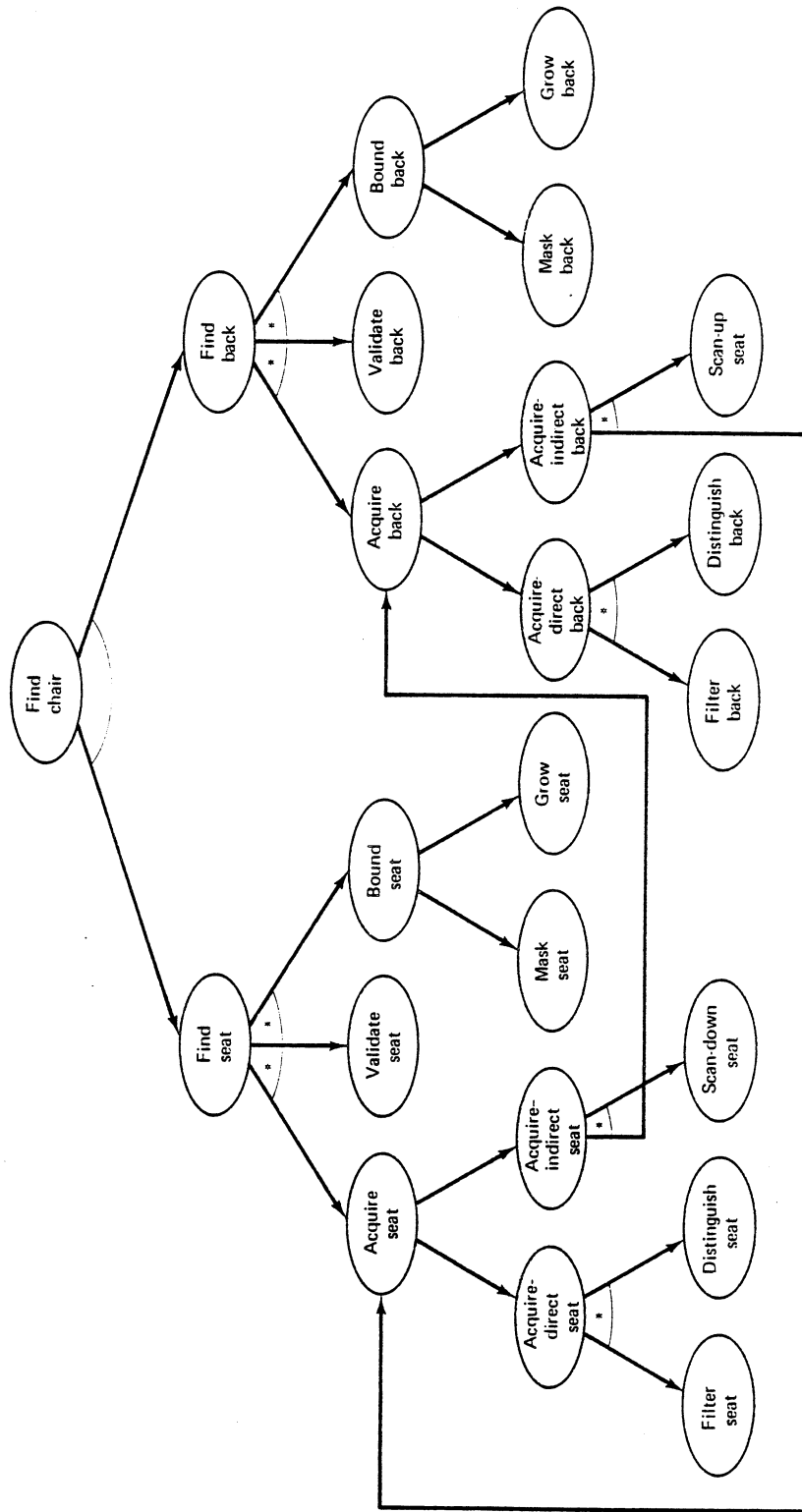


Figure III-10. Un plan pour trouver une chaise ([Garvey 76]).

Chaque module de description doit posséder un descripteur permettant au générateur de plan de savoir dans quelles conditions appeler ce module, et comment choisir entre plusieurs modules applicables.

Exemple. Dans le cas du problème

(ACQUIRE TABLE)

le générateur peut utiliser le programme de recherche d'un plan horizontal, si le plateau d'une table est défini comme étant une face plane horizontale. Alternativement, si cette information est absente du modèle de la table, le générateur utilisera les programmes standard associés au problème ACQUIRE.

Un autre exemple, faisant appel à une méthode générale, est fourni par le graphe de la figure III-10. Un programme standard associé au problème FIND indique que la recherche d'un objet peut passer par la recherche de ses composants. Donc, la recherche d'une chaise est décomposée en recherche de siège et recherche de dossier.

En général, plusieurs programmes peuvent être associés à un type de problème.

Structure d'un plan

Un plan peut être considéré comme la somme d'un programme et du raisonnement qui l'a produit. Il possède la structure d'un graphe dont les feuilles (noeuds sans successeur) forment le programme. Les autres noeuds définissent les étapes de la synthèse du programme; ils peuvent être de plusieurs types:

- * Un noeud de type ET. La solution du problème associé à un noeud de type ET demande la solution de tous les problèmes associés aux descendants de ce noeud, dans un ordre quelconque.
- * Un noeud de type *ET. Les successeurs d'un noeud de type *ET doivent être élaborés dans l'ordre.
- * Un noeud de type OU. Un noeud de type OU demande la solution d'au moins un de ses successeurs.
- * Un noeud de type |. Les noeuds de type ET, *ET, OU possèdent plusieurs successeurs. Un noeud de type | n'a qu'un successeur. Il représente donc le remplacement d'un problème par un autre.

Par le biais des noeuds OU un plan contient plusieurs solutions différentes du problème. Une solution spécifique est obtenue par un sous-graphe du plan dans lequel on retient un seul descendant pour chaque noeud OU.

3.2.4.3 Valuation d'un plan

Principe

Lors de l'exécution d'un plan, un problème nouveau se pose: quel sous-but chercher à atteindre, étant donné qu'en général plusieurs possibilités se présentent au noeuds de type OU. Ce choix dépend d'abord du coût de la solution d'un sous-but, mais aussi de critères plus globaux. Un autre but peut voir son coût décroître une fois qu'un sous-but est atteint, ou sa vraisemblance d'obtenir un résultat correct peut changer.

Ces considérations amènent à la définition d'une valeur numérique pour chaque noeud d'un plan - le score. Le score est défini comme le quotient de deux attributs d'un noeud: le coût K en temps de calcul divisé par une estimation de la qualité C du résultat.

L'exécution d'un plan procède alors comme suit. On dispose d'un plan valué, dans lequel chaque noeud est affecté d'un score. A partir de la racine, on descend dans le plan, en passant aux noeuds descendant à score minimal. L'action correspondant au noeud terminal est alors exécutée, résultant dans un succès ou dans un échec. L'ensemble des scores des noeuds du plan est alors mise à jour en fonction de ce résultat, et, si le problème initial n'est pas résolu, l'exécution répète ces opération.

La valuation sert donc à un double objectif:

- 1) Le plan produit par le générateur de plan regroupe (par le biais des noeud OU) plusieurs séquences d'actions permettant d'atteindre l'objectif. A l'aide du score, ces alternatives sont ordonnées de façon à essayer l'alternative la plus prometteuse avant les autres.
- 2) Les actions d'un plan sont linéarisées à l'aide du score, de façon à exploiter au mieux l'information fournie par la solution d'un sous-problème pour d'autres sous-problèmes.

Valuation d'une feuille

Une feuille d'un plan définit une procédure de description T. L'exécution d'un module T se termine soit avec succès, soit en échec. Cependant, le succès "subjectif" de la procédure ne signifie pas forcément que son objectif vis à vis du plan a été atteint. Cette situation est modélisée par deux paramètres associés à T.

$P(T)$ est la vraisemblance d'un succès.

$P(a|T)$ est la vraisemblance que le module donne l'objectif visé, sachant qu'il s'est terminé avec succès.

Ces notions sont illustrées par l'exemple du module d'acquisition par échantillonnage de la table. Ce module reportera un succès s'il trouve un nombre suffisant de pixels ayant à la fois la bonne hauteur et la bonne couleur. Par contre, ces pixels peuvent éventuellement appartenir à d'autres objets: dossier de chaise, téléphone, livres, etc.

Les paramètres $P(T)$ et $P(a|T)$ sont déterminés par le système grâce à des formules générales associées aux primitives de description, et par analyse globale des objets dans la scène (taille notamment).

Le coût de l'exécution d'une primitive de description est déduit du nombre de pixels auxquels elle est appliquée.

Valuation d'un noeud non-terminal

Les noeuds non-terminaux d'un plan reçoivent un score en fonction du coût K et de la qualité C des noeuds successeurs. Le principe est le suivant:

* Noeud OU. Le score d'un noeud OU est le score de son successeur au meilleur score. Ceci correspond à l'hypothèse optimiste que le successeur le moins cher donnera un succès - il n'y a pas de pondération des scores des autres successeurs.

* Noeud ET, *ET, |
Les successeurs d'un noeud ET sont ordonnés suivant la valeur de

$$K_i / (1 - C_i)$$

Les successeurs d'un noeud *ET sont déjà ordonnés.
Les valeurs d'un noeud ET sont alors

$$C_{ET} = \Pi C_i$$

$$P(T_{ET}) = \prod P(T_i)$$

$$K_{ET} = \sum K_i * \prod_{j=1}^i P(T_j)$$

La dernière formule vient du fait que le i° successeur d'un noeud ET est exécuté seulement si les successeurs T_1, \dots, T_{i-1} se terminent tous avec succès. Le coût lié au noeud T_i est le coût du sous-problème multiplié par la probabilité qu'il sera à résoudre.

Algorithme de propagation de scores

Les formules de calcul du score d'un noeud en fonction de ses successeurs sont appliquées à un plan par l'algorithme suivant. Rappelons que les noeuds terminaux possèdent initialement un score.

- 1) Initialisation à INFINI de tous les scores des noeuds non-terminaux.
- 2) Chainage de tous les sous-buts dans la liste VERIF.
- 3) TANTQUE VERIF # VIDE FAIRE
 - 3.1 COURANT := PREMIER(VERIF)
VERIF := RESTE(VERIF)
 - 3.2 SI NON FEUILLE(COURANT)
ET COURANT peut être évalué (cf. remarque)
ALORS
calculer le score de COURANT
si la valeur est différente de la valeur précédente,
ajouter les prédecesseurs de COURANT à la liste VERIF.

Remarque: un noeud ET peut être évalué si tous ses successeurs possèdent un score fini, un noeud OU si au moins un successeur possède un score fini.

Si le plan a une structure d'arbre, cet algorithme calcule les scores en remontant dans l'arbre, passant exactement une fois dans chaque noeud. La situation est plus complexe en présence de boucles dans le plan. Pour assurer la terminaison de l'algorithme il faut imposer deux conditions supplémentaires.

- a) Lorsqu'un noeud OU est évalué plusieurs fois à l'étape 3.2, son score ne doit pas augmenter.

b) Chaque boucle du plan contient au moins un noeud OU "exécutable". Un noeud exécutable peut être

- * une feuille
- * un noeud ET (*ET, |) dont tous les successeurs sont exécutables.
- * un noeud OU ayant au moins un successeur exécutable.
- * une boucle contenant un noeud OU exécutable.

3.2.4.4 Exécution d'un plan

A l'exécution on dispose d'un plan entièrement valué. On répète alors les opérations suivantes, jusqu'à ce que le problème soit résolu (ou jusqu'à un échec global).

- 1) Choix d'un opérateur de description.
Etant donné que les successeurs de chaque noeud du plan ont été ordonnés lors de la valuation, on peut choisir la première feuille à gauche. Cela correspond en particulier, pour les noeuds OU, au choix du successeur au meilleur score.
- 2) Exécution de l'opérateur. L'exécution se termine en succès ou en échec.
- 3) Mise à jour de la valuation. Le résultat de l'exécution fournit des informations nouvelles modifiant la valuation du plan restant, qui doit donc être mise à jour.

3.2.4.5 Exemple: Recherche d'une chaise

Une chaise étant un objet composé, le plan (cf. figure III-10) prévoit d'abord la recherche de l'un des deux composants, puis une recherche "indirecte" de l'autre.

Le système décide de chercher d'abord le siège, puis le dossier: le siège ayant une face horizontale, plus facile à détecter et plus discriminante que les éléments de description du dossier, dont l'apparence peut beaucoup changer suivant l'orientation. L'acquisition du siège commence avec un échantillonnage de la scène (figure III-11-a) pour trouver des points de hauteur comprise entre 14 et 15 pouces. Les points

sélectionnés (figure III-11-b) sont validés en fonction de la couleur (figure III-11-c), puis le siège est encadré par croissance de région (figure III-11-d).

La recherche du dossier admet trois alternatives: recherche directe (score 45300), définition d'une fenêtre à partir de la position du siège (score 36600), ou recherche "au-dessus" du siège (score 34000). La dernière alternative est choisie, la recherche s'effectue en haut des lignes de la figure III-11-e. Finalement, le dossier est encadré par suivi de ligne.

3.3 DISCUSSION

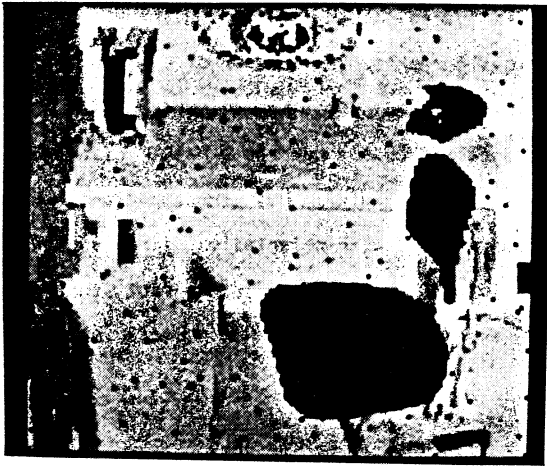
3.3.1 Comparaison avec la prédiction - vérification

Nous distinguons trois composants essentiels dans le système de Garvey:

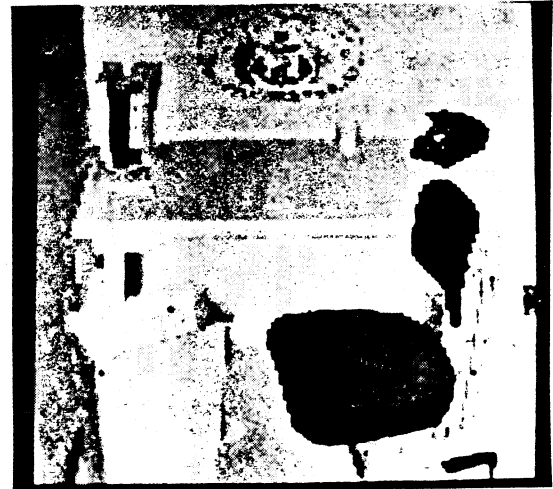
- * Des connaissances, ou un modèle de l'univers, décrivant les objets à reconnaître et des propriétés des scènes à analyser.
- * Un générateur de plans capable d'exploiter ces connaissances en fonction d'un problème de vision donné. "Exploiter" signifie ici: transformer les éléments appropriés du modèle de l'univers en éléments d'un plan de reconnaissance. D'un point de vue fondamental, cela est un problème du type "synthèse de programmes à partir de connaissances".
- * Un critère de valuation. Dans la mesure où le modèle de l'univers permet d'envisager différentes séquences d'actions, il faut un moyen de décider du meilleur choix. Garvey utilise une approche probabiliste à la valuation.

Cette structure possède un aspect commun avec la prédiction - vérification. Dans les deux cas il s'agit d'exploiter des connaissances sur la scène afin d'orienter l'analyse de l'image. Nous relevons également des différences importantes.

- Le programme de Garvey considère plusieurs types de connaissances variées qui sont amalgamées dans un plan: relations entre objets dans la scène, composition d'un objet à partir de plusieurs faces, description des faces en termes de pixels. Cela permet une optimisation globale du plan. A la place d'une communication entre description et interprétation, nous trouvons ici la production d'un programme de description spécialisé.



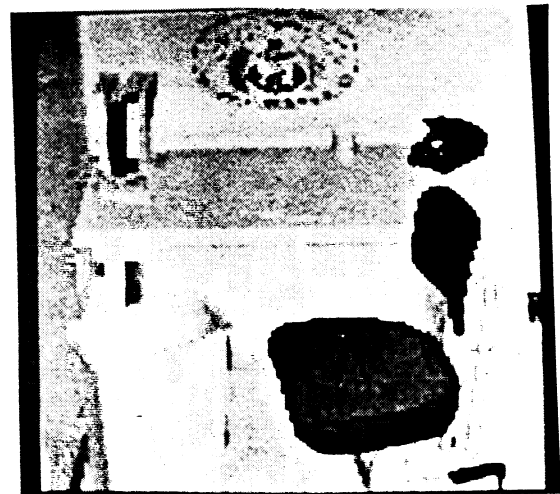
a) Points échantillonnés pour la recherche du siège.



b) Points à la bonne hauteur.



c) Points validés pour le siège.



d) Siège encadré.



e) Recherche du dossier.



f) Résultat final: siège et dossier.

Figure III-11. Les étapes de l'exécution du plan pour trouver une chaise ([Garvey 76]).

- La génération de plans considère plusieurs alternatives pour reconnaître l'objet, alors que la prédiction - vérification en déduit une seule directement du modèle.

Par rapport à la prédiction - vérification, la génération de plans comporte donc un niveau d'abstraction supplémentaire, considérant un ensemble d'opérations comme une entité, qui peut être analysée et comparée. Cette différence explique la raison pour laquelle la prédiction - vérification s'applique naturellement dans le contexte dans lequel nous l'avons placée dans les chapitres précédents: lorsqu'une hypothèse est complète, il n'y a plus qu'une possibilité à considérer.

3.3.2 Comparaison avec la génération de plans d'actions

La comparaison du système de Garvey avec la classification de [Bessière 83] montre un lien de parenté très net avec les générateurs de plans hiérarchiques, et plus spécialement avec NOAH, le système de génération de plans par "affinement d'ossature" [Sacerdoti 75]. Nous constatons également plusieurs différences importantes entre les deux systèmes.

- La réserve que l'on peut formuler à propos de NOAH - la décomposition d'un problème se faisant par des procédures spécialisées, le programmeur doit explicitement prévoir toutes les alternatives, ce qui appauvrit le système en généralité - ne s'applique pas au système de Garvey. En effet, la décomposition est élaborée en fonction de données déclaratives, à savoir le modèle de l'objet et la base de données (cf. figure III-9). Ce système fait donc un pas de plus vers l'ambition de l'IA qui est l'exploitation d'une représentation déclarative des connaissances pour la résolution d'un problème.

- Par rapport à la génération de plans en robotique, la VO présente une particularité importante: chaque opération élémentaire du système de VO augmente les connaissances du système. Un opérateur possède une liste de préconditions, et une liste d'ajout, mais il n'y a pas de liste de retrait. Cela signifie que dans le contexte de la VO on n'est pas confronté au problème des buts interdépendants.

3.3.3 Généralisation

La description des objets implémentée par Garvey utilise peu de propriétés, et les relations entre objets sont peu nombreuses. Il nous semble important d'admettre une plus grande variété pour la description de l'univers. Ceci nous amène aux problèmes de modélisation géométrique et de représentation de connaissances.

Toutefois, un tel enrichissement pose un problème fondamental nouveau. Un modèle de l'univers plus complet implique que la taille d'un plan sera plus importante. En effet, chaque "chemin de relations" partant de l'objet recherché produit une alternative dans le plan engendré. Bien entendu, les bonnes alternatives sont peu nombreuses, la plus grande partie des possibilités correspondant à des "idées tordues", comme par exemple de chercher la corbeille à papier après avoir trouvé la table à l'aide du téléphone.

Le problème de l'approche de Garvey vient du fait que la valuation ne peut se faire que sur un plan complet, puisque l'algorithme de valuation part des feuilles et doit parcourir les boucles.

Dans un système plus général il faut donc introduire de nouveaux critères de valuation permettant l'élagage du plan pendant la construction. Certains systèmes offrent ce type de possibilités, par exemple par le biais des critiques de NOAH.

Notons également, si on compare avec le comportement humain, que la séparation entre la génération et l'exécution d'un plan, si commode pour limiter le coût informatique de la reconnaissance, doit en partie être effacée: si toutes les stratégies de reconnaissance raisonnables a priori échouent à l'exécution, on reprend la phase de génération afin de déterminer d'autres stratégies (en tenant en plus compte du travail d'interprétation déjà effectué).

CHAPITRE 4

CONCLUSION SUR LA RECONNAISSANCE D'OBJETS

4.1 ADEQUATION AU PROBLEME PHYSIQUE

Nous avons, au premier chapitre, donné une formalisation du problème de reconnaissance d'objets, que nous avons affinée, afin de tenir compte du bruit dans le processus de la vision, en deux problèmes qui sont l'identification et la localisation; et nous avons développé une approche par prédiction - vérification, que nous avons située par rapport à plusieurs autres algorithmes.

Ce modèle de reconnaissance, est-il adéquat à la réalité physique?

La confrontation avec le monde réel révèle un problème lié à la relation "plus informé que". La relation "plus informé que" est d'une importance capitale pour notre approche à la reconnaissance - c'est elle qui structure le graphe des hypothèses, c'est elle, par le biais de la fonction plausibilité, qui définit la qualité d'une hypothèse et d'une solution. Cependant, son utilisation repose sur deux hypothèses concernant l'univers: l'hypothèse de bonne structuration de l'univers, et l'hypothèse de l'univers clos.

L'hypothèses de bonne structuration de l'univers

Si un ensemble d'indices données $\{ID_j\}$ est cohérent par rapport à un modèle $\{IM_i\}$ à l'intérieur d'une hypothèse à plausibilité maximale,
alors
les $\{ID_j\}$ appartiennent effectivement à un même objet.

Force est de constater que cette hypothèse ne correspond pas à une loi de la nature - elle n'est pas toujours vérifiée. Elle correspond néanmoins à une situation normale. Nous estimons que cette hypothèse est nécessaire pour un algorithme de reconnaissance performant, mais qu'elle ajoute une difficulté majeure au problème d'identification: découvrir l'exception -

s'apercevoir des rares cas où cette hypothèse doit être mise en cause.

Nous avons ainsi introduit un problème de raisonnement non-monotone dans l'identification.

L'exemple d'une telle exception est illustré par la figure III-12, utilisant les triangles. Doit-on choisir une hypothèse plus plausible avec une erreur assez importante (figure III-12-c), ou deux hypothèses moins plausibles avec une erreur d'approximation faible (figure III-12-d)? La difficulté réside moins dans le choix entre ces solutions que dans le fait même de poser ces alternatives.

Dans le cas de l'exemple, une analyse de recouvrement, basée sur la connaissance de la façon dont un objet peut se placer devant un autre, devrait résoudre le problème.

La difficulté soulevée par cette hypothèse n'est pas du tout intuitive: en termes anthropomorphes, elle dit que "l'on voit ce que l'on croit voir". La vision humaine utilise des informations fortement redondantes; le recouvrement d'informations de natures différentes nous permet de confirmer l'identité de plusieurs façons indépendantes, donc d'en être très sûr. Mais le système visuel humain est aussi toujours capable de retracter une hypothèse d'identification et d'en proposer une autre.

Nous pensons que la stratégie de prédiction - vérification contribue à vérifier subtilement l'hypothèse de bonne structuration: contrairement à une segmentation sans but de l'image, dans laquelle on s'attend volontier à des alignements accidentels, chaque réussite d'une vérification vote en faveur de l'hypothèse d'objet; par définition, les alignements accidentels sont exceptionnels et ne se cumulent pas. Cet argument justifie aussi l'introduction d'une fonction validité, comme c'est le cas dans PVV, au-delà d'un simple argument d'efficacité: si un trop grand nombre de prédictions n'est pas vérifié, cela ne saurait être dû au hasard, mais à l'absence réelle de l'objet.

Par contre, on peut aussi trouver des arguments opposés. Par définition, un système prédictif ne cherchera que les informations qui arrangent ses hypothèses - c'est la raison même de son efficacité - et n'aura pas la curiosité de chercher des indices qui risquent de les détraquer. D'une façon générale, un système de vision "intelligent" doit donc combiner une structure de contrôle prédictive, qui assure son efficacité, avec des éléments d'une structure ascendante afin de vérifier la bonne structuration de l'univers.

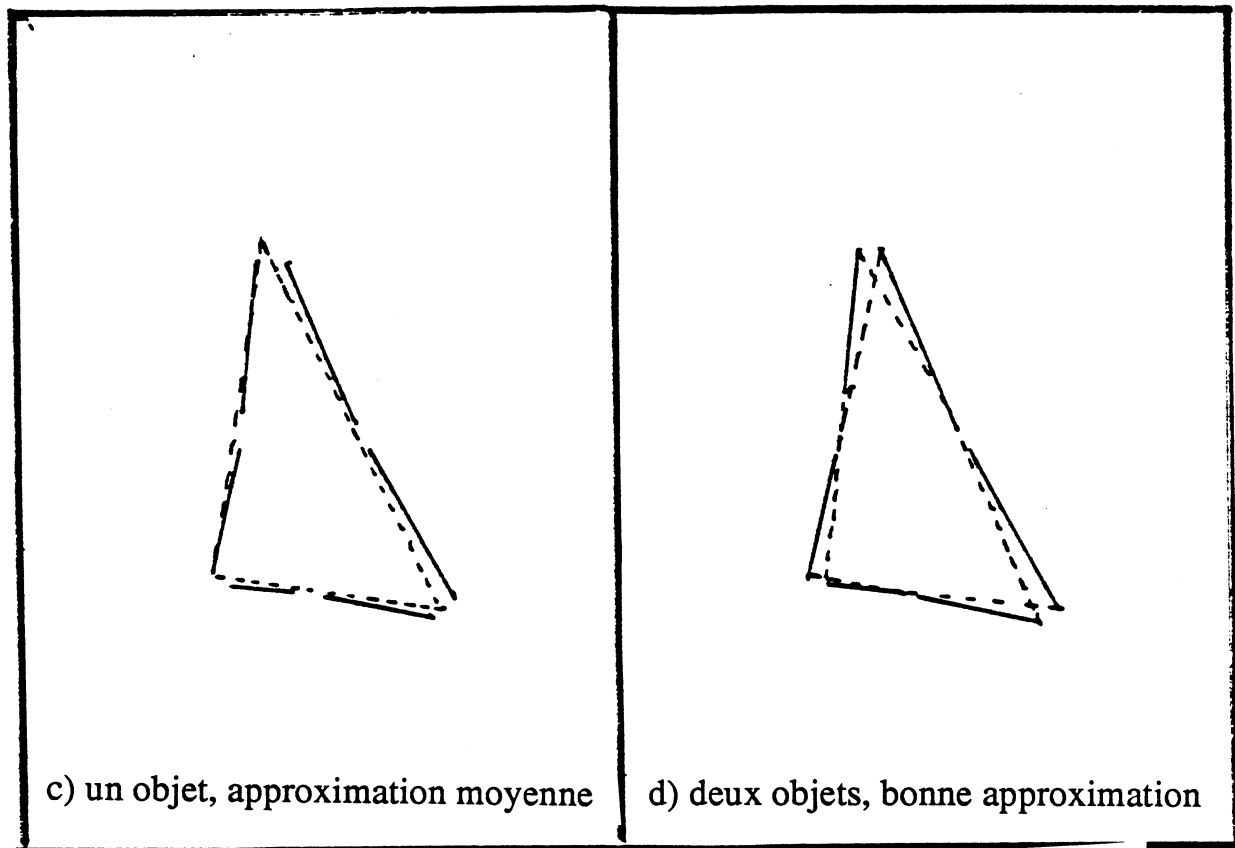
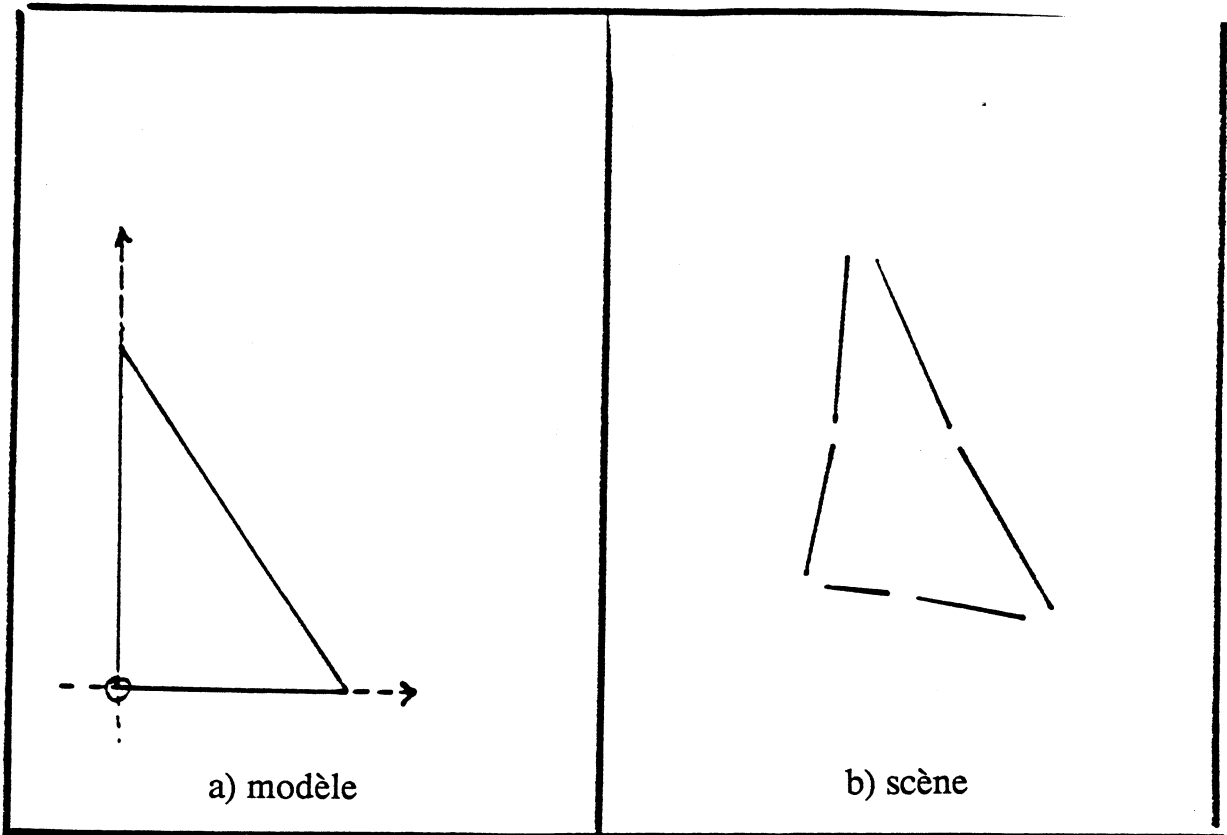


Figure III-12. Une scène ambiguë, avec deux interprétations très différentes (c et d).

L'hypothèse de l'univers clos

A l'heure actuelle, tous les systèmes qui raisonnent dans l'univers d'un robot - génération de plans, moniteur d'exécution - font fatalement l'hypothèse de connaître tous les objets dans cet univers. Pour ce qui est de la vision, cette hypothèse n'est guère plus forte que celle de bonne structuration:

Dans les conditions énoncées pour l'hypothèse de bonne structuration, l'objet défini par l'ensemble $\{ID_j\}$ appartient effectivement à la classe d'objets définie par le modèle $\{IM_i\}$.

Cette hypothèse est également non-intuitive: la redondance permet à l'homme d'identifier comme tels des "objets inconnus" et d'apprendre ainsi des nouveaux objets. Par contre, elle est nécessaire pour notre formalisme de reconnaissance d'objets, même si l'on exclut le problème d'une éventuelle ambiguïté des modèles, qui complique la discussion sans en changer le fond.

4.2 L'IDENTIFICATION D'OBJETS EST UN PROBLEME D'IA

Nous avons justifié la séparation du problème de reconnaissance en identification et localisation par le bruit présent dans une image. Nous avouons que cela n'était qu'un argument de circonstance. L'identification d'un objet doit être le résultat d'un raisonnement symbolique complexe dans lequel interviennent de nombreux aspects, dont nous avons discutés quelques-uns seulement. En fait, on met facilement en évidence dans le problème de reconnaissance d'objets les questions importantes dans la recherche en IA:

*** l'incertain.**

Toute hypothèse d'objet doit faire face à l'incertain sous différentes formes. La donnée initiale de l'image est affectée d'un bruit qui se transmet aux indices visuels; les modèles d'objets sont applicables seulement avec l'hypothèse de bonne structuration de l'univers; la possibilité d'objets inconnus, en conjonction avec le fait qu'un objet n'est en général que partiellement visible, signifie qu'une hypothèse d'objet, aussi plausible soit-elle, ne peut jamais être certaine. Dans ces conditions, la reconnaissance doit cumuler autant d'évidence que possible, afin d'augmenter la certitude grâce à la redondance. En ce sens, toutes les connaissances du système peuvent contribuer à la reconnaissance, même si elles n'ont aucun rapport direct

avec la vision.

* le **temps** et l'**espace**.

Les objets à reconnaître, ainsi que le capteur oeil lui-même, évoluent dans le temps et dans l'espace. Comment peut-on raisonner symboliquement dans ces catégories?

* L'**apprentissage**.

Si l'on veut aller au-delà de l'hypothèse de l'univers clos, le système de vision doit être capable de classifier un objet comme inconnu, puis en construire un modèle. L'apprentissage est un problème transversal, dans lequel se posent tous les problèmes précédents, puisqu'un modèle est une description qui doit faire abstraction des variations avec le temps et dans l'espace.

Comme on le voit dans cette énumération, la VO et l'IA se confondent complètement dans le problème de reconnaissance d'objets. Ceci est d'autant plus le cas qu'un objet (au sens intuitif) est le plus souvent défini par sa fonction et non par sa forme.

Cette réflexion nous incite à proposer une ligne de séparation entre la VO et l'IA. La VO doit construire une description, dans les catégories de l'espace et du temps, des éléments d'une scène. Cette description se fait par des volumes et des surfaces, avec leurs propriétés physiques visuelles, et par des relations dans l'espace et dans le temps. La reconnaissance d'objets, dans la mesure où ceux-ci ne sont pas définis par des propriétés visuelles, est un problème pour l'IA.

Avec cette convention, la discussion des chapitres 1 et 2 traite bien de l'aspect vision de la reconnaissance d'objets puisqu'elle concerne la mise en correspondance entre ensembles d'indices **visuels**. Par contre, dans le chapitre 3 elle n'a pas été respectée. Une "chaise" est un objet générique défini par une fonction, qui a un rapport lointain seulement avec sa forme. Reconnaître une chaise n'est plus un problème de vision. Dans la description du programme de Garvey nous n'avons pas utilisé la notion de chaise selon cette acception courante, mais pour désigner une forme géométrique particulière définie dans la figure III-9.

CONCLUSION GENERALE

La Vision par Ordinateur est un sujet vaste et passionnant dont l'étude dépasse rapidement le cadre strict de l'informatique. Dans ce rapport nous avons présenté

- * une réflexion sur la nature de la VO et la structure qu'elle induit pour un système informatique.
- * une étude algorithmique de quelques problèmes précis soulevés par la VO.
- * l'implantation d'un noyau de système polyvalent, servant d'outil pour la recherche et pour la réalisation d'applications simples.
- * une étude du problème de reconnaissance d'objets, dans laquelle apparait clairement le lien entre vision et intelligence.

Etant donné l'ampleur du sujet, notre contribution reste fragmentaire - en particulier, nous n'avons pas abordé le problème de construction d'indices 3-D à partir d'une image. Après des conclusions à caractère technique qui se trouvent à la fin de plusieurs paragraphes de cette rédaction nous ajoutons ici quelques remarques plus spéculatives. Ces conclusions étant très personnelles, j'abandonne le *pluralis majestatis* au profit de la première personne plus directe.

Je pense que les recherches en Vision par Ordinateur ont souffert de la concentration des efforts sur la reconnaissance d'objets, et sur l'analyse d'une seule image, en noir et blanc de surcroît. C'est certainement lié aux problèmes de matériel, mais c'est aussi la voie de la facilité.

La reconnaissance n'est qu'une tâche parmi d'autres pour un système de vision, peut être la plus difficile, certainement celle qui efface le plus la limite entre VO et IA, permettant par là d'esquiver les vrais problèmes de la vision.

L'analyse d'une séquence d'images - la vision mobile - ne doit pas tomber dans le piège de vouloir appliquer à plusieurs images les techniques reconnues trop lourdes déjà pour une seule image. Rappelons nous de la leçon de Waltz, que les contraintes d'interprétation peuvent croître plus vite que la quantité d'informations (ce qui est vrai aussi pour la couleur). Cependant, l'impératif d'un traitement en temps réel, dû à l'échantillonnage temporel, pose un problème fondamental de cohérence d'un processus d'analyse lent - l'interprétation ne peut en aucun cas se faire dans le temps d'une seule image - dont les données évoluent à un rythme rapide; problème qui n'a pas été abordé à ce jour. La quatrième dimension doit aussi intervenir explicitement dans la description d'une scène, comme l'illustre si bien le travail de [Neumann84]. En récompense à ces difficultés, le mouvement permet d'accéder à la définition biologique de la

notion d'objet. Un objet est ce qui se meut ensemble [Uexküll], définition compatible avec celle du potentiel de fonction utile en IA.

Les chercheurs en Intelligence Artificielle peuvent dans les systèmes de vision trouver des beaux exemples de concepts encore mystérieux; je pense notamment aux notions d'abstraction et d'apprentissage, fortement interconnectées. L'abstraction suppose un changement de représentation, l'apprentissage a besoin de mettre en correspondance différentes représentations du problème. Or, les niveaux de données successifs d'un système de vision sont des véritables abstractions, dont la plus noble est celle d'objet. Les mécanismes d'abstraction derrière une telle structure sont des plus généraux. J'interprète l'intuition de K. Lorenz, selon laquelle la vision, en tant que perception de l'univers dans les dimensions de l'espace et du temps, est une précondition à la genèse de l'homme [Lorenz 54], en ce sens que les mécanismes de l'intelligence se sont développés par la vision.

En ce qui concerne l'algorithmique, l'effort de programmation de systèmes de vision a créé, par dessus une quantité de beaux résultats, un sentiment diffus d'insatisfaction. Résoudre des problèmes intuitivement très simples, comme le suivi d'une ligne de contraste dans une image GTR, ou la recherche de segments de droites, demande des mois d'efforts, et malgré tout les programmes manquent de souplesse - il faut encore les adapter aux applications nouvelles. L'argument souvent avancé - que le raisonnement humain fait intervenir des connaissances de haut niveau - n'est pas justifié pour ces problèmes de vision de bas niveau; ni peut on trouver une réponse dans l'IA. Je crois que cette difficulté vient de l'inadéquation de nos concepts. Aujourd'hui, pour programmer il faut spécifier, formaliser, et l'approche par l'IA n'y échappe pas.

Il doit être possible de programmer autrement.

REFERENCES

Abbreviations:

CGIP Computer Graphics and Image Processing
CVS Computer Vision Systems, édité par A.R.Hanson et
E.M.Riseman, Academic Press 1978
IJCAI International Joint Conference on Artificial
Intelligence
IJRR International Journal of Robotics Research

[Agin81]

Agin, G.J.(1981)
Fitting ellipses and general second-order curves
Robotics Inst., CMU, CMU-RI-TR-81-5

[Akermark84]

Akermark, K., Crebassa, M., Salmon, F., Schmitt, P.(1984)
Montage d'un amortisseur
Rapport de projet, ENSIMAG

[Ambler75]

Ambler, A.P., et al.(1975)
A versatile computer-controlled assembly system
Artificial Intelligence, vol.6,2, p.129-156

[Ayache83]

Ayache, N.(1983)
Un système de vision bidimensionnelle en robotique
industrielle
Thèse de docteur-ingénieur, Université de Paris-Sud

[Baird84]

Baird, H.S.(1984)
Model-based image matching using location
PhD Thesis, Princeton University

[Ballard81]

Ballard, D.H.(1981)
Generalizing the Hough transform to detect arbitrary
shapes
Pattern Recognition, vol.13, p.111-122

[Ballard et Brown82]

Ballard, D.H., and Brown, C.M.(1982)
Computer vision
Prentice Hall, Englewood Cliffs, N.J.

[Ballard77]

Ballard, D.H., et al.(1977)
An approach to knowledge-directed image analysis
IJCAI-77, p.664-670

- [Ballard78]
Ballard, D.H., et al.(1978)
An approach to knowledge-directed image analysis
in CVS, p.271-281
- [Barr et Feigenbaum81]
Barr, A., et Feigenbaum, E.A.(1981)
The handbook of artificial intelligence
vol. 1 et 2, William Kaufman Inc.
- [Barrow78]
Barrow, H.G.(1978)
Recovering intrinsic scene characteristics from images
in CVS, p.3-26
- [Barrow76]
Barrow, H.G., and Tenenbaum, J.M.(1976)
MSYS: A system for reasoning about scenes
Technical Note 121, Artificial Intelligence Center,
Stanford Research Institute
- [Berlioux et Bizard83]
Berlioux, P., et Bizard, Ph.(1983)
Algorithmique
Dunod
- [Binford82]
Binford, T.O.(1982)
Survey of model-based image analysis systems
IJRR 1,1,p. 18-64
- [Bledsoe75]
Bledsoe, W.W.(1975)
A new method for proving certain Presburger formulas
IJCAI 4, p.25-31
- [Bolles82]
Bolles, R.C., and Cain, R.A.(1982)
Recognizing and locating partially visible objects: The
Local-Feature-Focus method
IJRR 1,3, p.57-82
- [Brice70]
Brice, C.R., and Fennema, C.L.(1970)
Scene analysis using regions
Artificial Intelligence 1, p.205-226
- [Brooks81a]
Brooks, R.A.(1981)
Symbolic reasoning among 3-D objects and 2-D models
Artificial Intelligence 17, p.285-348

- [Brooks81b]
Brooks, R.A.(1981)
Model based three-dimensional interpretation of
two-dimensional images
IJCAI 7, p.619-624
- [Canny83]
Canny, J.F.(1983)
Finding edges and lines in images
Technical Report no.270, MIT, Boston
- [Clowes71]
Clowes, M.B.(1971)
On seeing things
Artificial Intelligence 2,p.79-116
- [Cohen et Feigenbaum82]
Cohen, P.R., et Feigenbaum, E.A.(1982)
The Handbook of Artificial Intelligence, vol.3
William Kaufman Inc.
- [Crowley84]
Crowley, J.L.(1984)
A computational paradigm for three dimensional scene
analysis
CMI-RI-TR-84-11, The Robotics Inst., CMU, Pittsburgh
- [Demazeau82]
Demazeau, Y.(1982)
Niveaux de représentation pour la compréhension d'images
Rapport de DEA, INP Grenoble
- [Demazeau84]
Demazeau, Y.(1984)
Niveaux de représentation pour la compréhension d'images
4ème Congrès AFCET "Reconnaissance de Formes et
Intelligence Artificielle", Paris
- [Dessimoz80]
Dessimoz, J.D.(1980)
Traitement des contours en reconnaissance de formes
visuelles: application en robotique
Ecole Polytechnique Fédérale de Lausanne, thèse n.387
- [Dettori82]
Dettori, G.(1982)
An on-line algorithm for polygonal approximation of
digitized plane curves
6th Intl. Conference on Pattern Recognition, p.840-842

- [Dorst et Smeulders84]
Dorst, L., et Smeulders, A.W.M.(1984)
Discrete representation of straight lines
IEEE PAMI-6,4, p.450-463
- [Draper81]
Draper, S.W.(1981)
The use of gradient and dual space in line-drawing
interpretation
Artificial Intelligence 16, p.461-508
- [Duda et Hart73]
Duda, R.O., and Hart, P.E.(1973)
Pattern classification and scene analysis
John Wiley and Sons
- [Erman75]
Erman, L.D., Lesser, V.R.(1975)
A multi-level organization for problem solving using
many, diverse, cooperating sources of knowledge
IJCAI IV, p.483-490
- [Falk72]
Falk, G.(1972)
Interpretation of imperfect line data as a
three-dimensional scene
Artificial Intelligence 3, p.101-144
- [Faugeras et Hébert85]
Faugeras, O.D., et Hebert, M.
The representation, recognition, and positioning of 3-D
shapes from range data
IEEE Transactions PAMI, à paraitre
- [Frisby81]
Frisby, J.(1981)
De l'oeil à la vision
Fernand Nathan
Version originale: Seeing (Oxford University Press,
1979)
- [Garvey76]
Garvey, T.D.(1976)
Perceptual strategies for purposive vision
SRI Artificial Intelligence Center, Techn.Nte 117, Menlo
Park, Ca.
- [Gordillo85]
Gordillo, J.L.(1985)
CAICOU: un système de développement interactif pour
l'analyse d'images en couleur
Note interne, LIFIA

- [Grimson et Lozano-Perez84]
Grimson, W.E., et Lozano-Perez, T.(1984)
Model-based recognition and localization from sparse
range or tactile data
IJRR vol.3,3, p.3-37
- [Grimson81]
Grimson, W.E.L.(1981)
From images to surfaces: A computational study of the
human early visual system
MIT Press, Cambridge, Mass.
- [Guzmán68]
Guzmán, A.(1968)
Computer recognition of three-dimensional objects in
visual scenes
Techn. Report MAC-TR-59, AI Laboratory, MIT, Cambridge,
Mass.
- [Hall79]
Hall, E.L.(1979)
Computer image processing and recognition
Academic Press, New York
- [Hanson78a]
Hanson, A.R., and Riseman, E.M.(1978)
Segmentation of natural scenes
in CVS, p.129-163
- [Hanson78b]
Hanson, A.R., and Riseman, E.M.(1978)
VISIONS: a computer system for interpreting scenes
in CVS, p.303-333
- [Hildreth80]
Hildreth, E.C.(1980)
Implementation of a theory of edge detection
AI-TR-579, AI-Lab, MIT, Boston
- [Hildreth83]
Hildreth, E.C.(1983)
Measurement of visual motion
MIT Press
- [Horain84]
Horain, P.(1984)
Vision par Ordinateur: extraction de primitives dans des
images tridimensionnelles
Thèse de Docteur Ingénieur, INP Grenoble

- [Horaus81]
Horaus, P.(1981)
Extraction et segmentation de contours dans une image,
application à l'inspection automatique
Thèse de Docteur Ingénieur, ENSIEG, Grenoble
- [Horn72]
Horn, B.K.P.(1972)
VISMEM: a bag of "robotics" formulae
publié dans Winston(ed.): Progress in vision and
robotics, report AI-TR 281, MIT, 1973
- [Horn77]
Horn, B.K.P.(1977)
Understanding image intensities
Artificial Intelligence Journal vol.8, p.201-231
- [Huffman71]
Huffman, D.A.(1971)
Impossible objects as nonsense sentences
In R.Meltzer and D.Michie(eds), Machine Intelligence 6.
New York, Elsevier, p.295-323
- [ITMI83]
Manuel de référence CAIMAN
ITMI SA, Meylan
- [Ikeuchi81]
Ikeuchi, K., et Horn, B.K.P.(1981)
Numerical shape from shading and occluding boundaries
Artificial Intelligence Journal vol.17, p.141-184
- [Kanade79]
Kanade, T.(1979)
Recovery of three-dimensional shape of an object from a
single view
Rep. No. CMU-CS-79-153, Carnegie-Mellon Univ. (Also in
Artificial Intelligence 17, p.409-461 (1981))
- [Kanade80]
Kanade, T.(1980)
A theory of Origami world
Artificial Intelligence 13, p. 279-311
- [Lieberman79]
Lieberman, L.(1979)
Model-driven vision for industrial automation
in "Advances in Digital Image Processing", Plenum Press
- [Logan77]
Logan, B.F.(1977)
Information in the zero-crossings of band pass signals
Bell System Technical Journal 56, p.487-310

- [Lorenz54]
 Lorenz, K.(1954)
 Psychologie und Stammesgeschichte
 Traduction: "Psychologie et phylogénèse" dans "Trois
 essais sur le comportement animal et humain", Editions du
 Seuil 1970
- [Lozano-Perez et Grimson84]
 Lozano-Perez, T., et Grimson, W.E.(1984)
 Recognition and localization of overlapping parts from
 sparse data
 2ième ISRR, Kyoto, p.28-33
- [Lux80]
 Lux, A.(1980)
 Visual localization of automobile parts without
 significant outline
 10th Intl. Symposium on Industrial Robots, Milan
- [Lux81a]
 Lux, A.(1981a)
 Un système de développement interactif pour le traitement
 d'images
 8ième Colloque sur le Traitement du Signal et ses
 Applications, Nice
- [Lux81b]
 Lux, A.(1981b)
 Un système pour le développement interactif d'algorithmes
 d'analyse de scènes
 Congrès AFCET Automatique, Nantes
- [Lux83]
 Lux, A.(1983)
 CAIMAN au LIFIA - Manuel d'utilisation
 Note interne, Laboratoire LIFIA/IMAG
- [Lux84]
 Lux, A.(1984)
 Du problème d'extraction de segments de droites d'une
 séquence de points
 Premier Colloque Image, CESTA, Biarritz
- [Lux83 et Souvignier]
 Lux, A., et Souvignier, V.(1983)
 PVV - a goal-oriented system for industrial vision
 IJCAI 8, Karlsruhe (RFA)
- [Lux et Souvignier84]
 Lux, A., et Souvignier, V.(1984)
 PVV - un système de vision appliquant une stratégie de
 prédiction - vérification
 4ème Congrès AFCET "Reconnaissance de Formes et
 Intelligence Artificielle", Paris

- [Mackworth73]
Mackworth, A.K.(1973)
Interpreting pictures of polyhedral scenes
Artificial Intelligence 4, p.121-137
- [Mackworth74]
Mackworth, A.K.(1974)
On the interpretation of drawings as three-dimensional scenes
Doctoral dissertation, Lab. Experimental Psychology,
Univ. Sussex
- [Marr77]
Marr, D.(1977)
Artificial Intelligence - a personal view
Artificial Intelligence Journal vol.9,1, p.37-48
- [Marr82]
Marr, D.(1982)
Vision
Freeman, San Francisco
- [Merø75]
Merø, L, et Vassy, Z.(1975)
A simplified and fast version of the Hueckel operator for
finding optimal edges in pictures
IJCAI 4, Tbilisi, p.650-655
- [Minsky75]
Minsky, M.(1975)
A framework for representing knowledge
dans: P. Winston(ed.) "The psychology of computer vision"
- [Mondot84]
Mondot, T.(1984)
Application de méthodes de vision par ordinateur à
l'automatisation du réglage d'un moniteur de télévision
Rapport de DEA, INP Grenoble
- [Muller et Mirgaux84]
Muller, Y., et Mirgaux, A.(1984)
Détection de quadriques par transformée de Hough
Congrès AFCET "Reconnaissance de Formes et Intelligence
Artificielle", Paris
- [Nagao80]
Nagao, M., and Matsuyama, T.(1980)
A structural analysis of complex aerial photographs
Plenum Press

- [Nakamura84]
Nakamura, A., and Aizawa, K.(1984)
Digital Circles
Computer Vision, Graphics, and Image Processing, vol.26,
p.242-255
- [Neumann84]
Neumann, B.(1984)
Natural language description of time-varying scenes
Universit de Hambourg, FB Informatik (en allemand)
- [Nevatia82]
Nevatia, R.(1982)
Machine perception
Prentice Hall
- [Nevatia80]
Nevatia, R., et Babu, K.R.(1980)
Linear feature extraction and description
Computer Graphics and Image Processing vol.13, p.257-269
- [Newell82]
Newell, A.(1982)
The knowledge level
Artificial Intelligence Journal vol.18,1, p.87-127
- [Nilsson80]
Nilsson, N.J.(1980)
Principles of Artificial Intelligence
Tioga Publishing Comp.
- [Ohta80]
Ohta, Y. et al.(1980)
Color information for region segmentation
CGIP 13, p.222-241
- [Ohta80]
Ohta, Y.(1980)
A region-oriented image-analysis system by computer
Doctoral dissertation, Information Sc. Dept., Kyoto
University
- [Pavlidis82]
Pavlidis, T.(1982)
Algorithms for graphics and image processing
Springer Verlag
- [Pavlidis74]
Pavlidis, T., and Horowitz, S.(1974)
Segmentation of planar curves
IEEE Trans. Comput., C-23, p.860-870

- [Perkins77]
Perkins, W.A.(1977)
Model based vision system for scenes containing multiple parts
IJCAI-77, p.678-684
- [Perkins78]
Perkins, W.A.(1978)
A model-based vision system for industrial parts
IEEE Transactions on Computers, vol. C-27,2, p.126-143
- [Proffitt82]
Proffitt, D.(1982)
The measurement of circularity and ellipticity on a digital grid
Pattern Recognition, vol.15,5, p.383-387
- [Ramer72]
Ramer, U.E.(1972)
An iterative procedure for the polygonal approximation of plane curves
Computer Graphics and Image Processing, vol.1, p.244-256
- [Reichardt76]
Reichardt, W., et Poggio, T.(1976)
Visual control of orientation behaviour in the fly
Quarterly Review of Biophysics, vol.9,3, p.311-375 et 377-438
- [Roberts65]
Roberts, L.(1965)
Machine perception of three-dimensional solids
In J. Tippett(Ed.), Optical and electro-optical information processing. MIT Press, Cambridge, p.159-197
- [Rosenfeld74]
Rosenfeld, A.(1974)
Digital straight line segments
IEEE Trans. Comp., vol. C-23,12, p.1264-1269
- [Rosenfeld76]
Rosenfeld, A., and Kak, A.C.(1976)
Digital picture processing
Academic Press, New York
- [Rubin78]
Rubin, S.(1978)
The ARGOS image understanding system
Doctoral Dissertation, Carnegie Mellon University

- [Shafer83]
Shafer, S.A.(1983)
Shadow geometry and occluding contours of generalized
cylinders
Phd. thesis, CMU
- [Shirai73]
Shirai, Y.(1973)
A context sensitive line finder for recognition of
polyhedra
Artificial Intelligence 4, p.95-119
- [Shirai78]
Shirai, Y.(1978)
Recognition of real-world objects using edge cues.
in CVS, p.353-362
- [Shostak75]
Shostak, R.S.(1975)
On the completeness of the sup-inf method
SRI International
- [Sklansky80]
Sklansky, J., and Gonzales, V.(1980)
Fast polygonal approximation of digitized curves
Pattern Recognition vol.12, p.327-331
- [Souvignier81]
Souvignier, V.(1981)
Prédiction et vérification en vision
Rapport de DEA, Laboratoire IMAG
- [Souvignier83]
Souvignier, V.(1983)
PVV - un système d'interprétation d'images par prédiction
- vérification
Thèse de troisième cycle, INP Grenoble
- [Tenenbaum77]
Tenenbaum, J.M., and Barrow, H.G.(1977)
Experimentation in interpretation-guided segmentation
Artificial Intelligence 8, p.241-274
- [Tomek74]
Tomek, I.(1974)
Two algorithms for piecewise linear continuous
approximation of functions of one variable
IEEE Trans. Comput., C-23, p.445-448
- [Tsuji81]
Tsuji, S., and Nakano, H.(1981)
Knowledge-based identification of artery branches in
cine-angiograms
IJCAI 7, p.710-715

- [Ullman78]
Ullman, S.(1978)
The interpretation of visual motion
MIT Press, Cambridge, Mass.
- [Wall84]
Wall, K., and Danielson, P.(1984)
A fast sequential method for polygonal approximation of
digitized curves
Computer Vision, Graphics, and Image Processing, vol.28,
p.220-227
- [Waltz72]
Waltz, D.(1972)
Generating semantic descriptions from drawings of scenes
with shadows
Techn. Report AI-TR-271, Project MAC, MIT, Cambridge,
Mass.
- [Williams81]
Williams, C.(1981)
Bounded straight-line approximation of digitized planar
curves and lines
Computer Graphics and Image Processing, vol.16, p.380-381
- [Winston75]
Winston, P.H.(éditeur)
The psychology of computer vision
McGraw Hill
- [Winston84]
Winston, P.H.(1984)
Artificial Intelligence
2nd edition, Addison Wesley Publishing Comp.
- [Yachida77]
Yachida, M., and Tsuji, S.(1977)
A versatile machine vision system for complex industrial
parts
IEEE Trans.Comput. C-26, p.882-894

AUTORISATION de SOUTENANCE

VU les dispositions de l'article 5 de l'arrêté du 16 avril 1974

VU les rapports de présentation de Messieurs

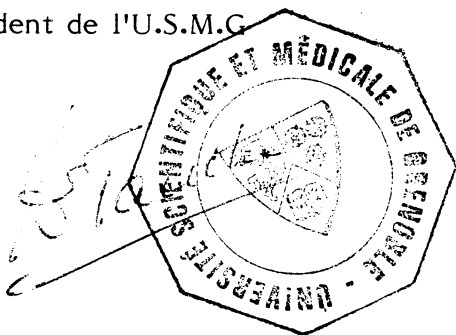
- . O.D FAUGERAS, Directeur Scientifique INRIA
- . J.C LATOMBE, Professeur
- . R. MOHR, Professeur

Monsieur LUX Augustin

est autorisé à présenter une thèse en soutenance en vue de l'obtention du grade de
DOCTEUR D'ETAT ES SCIENCES.

Fait à Grenoble, le 6 septembre 1985

Le Président de l'U.S.M.G



Le Président
BLOCH

Le Président de l'I.N.P.-G

D. BLOCH
Président
de l'Institut National Polytechnique
de Grenoble

RESUME: Cette thèse traite de la Vision par Ordinateur - VO - et de sa relation avec l'Intelligence Artificielle - IA. Elle est composée de trois parties. La partie I "La Vision par Ordinateur - présentation et réflexions" donne, après une définition des deux domaines, un aperçu des approches théoriques et des réalisations existantes en VO utilisant des concepts de l'IA. Nous développons ensuite notre approche qui intègre le concept de structures de contrôle au sens de l'IA dans un système algorithmique incrémental (à base de coroutines). Cette approche, et quelques applications du système CAIMAN qui en résulte, sont détaillées dans les parties II et III. La partie II "De la construction d'indices visuels" analyse des problèmes algorithmiques importants pour la vision "de bas niveau", notamment celui de l'extraction de segments de droites. La partie III "Sur la reconnaissance d'objets" étudie le problème de l'interprétation d'indices visuels en termes de modèles symboliques, et analyse plusieurs méthodes d'IA permettant d'en maîtriser la combinatoire, en particulier la stratégie de prédiction - vérification.

MOTS-CLES: Vision par Ordinateur, Intelligence Artificielle, Structures de Contrôle, Indices Visuels, Extraction de segments de droites, Prédiction-Vérification, algorithmes incrémentaux.