



HAL
open science

La détection des contours dans des images à niveaux de gris : mise en oeuvre et sélection de détecteurs

Djemel Ziou

► **To cite this version:**

Djemel Ziou. La détection des contours dans des images à niveaux de gris : mise en oeuvre et sélection de détecteurs. Traitement des images [eess.IV]. Institut National Polytechnique de Lorraine - INPL, 1991. Français. NNT: . tel-00812973

HAL Id: tel-00812973

<https://theses.hal.science/tel-00812973>

Submitted on 14 Apr 2013

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.

**La détection de contours
dans des images à niveaux de gris :
mise en œuvre et sélection de détecteurs**

THÈSE

présentée et soutenue publiquement le **11 octobre 1991**

pour l'obtention du

**Doctorat de l'Institut National Polytechnique de Lorraine
(Spécialité Informatique)**

par

Djemel ZIOU¹

devant le jury composé de :

Président : Jean-Paul HATON

Rapporteurs : Serge CASTAN
René HUSSON

Examineurs : Gérard GIRAUDON
Gérald MASINI
Roger MOHR

¹Adresse de l'auteur à partir du 1/2/92 : EERIE-LERI, Parc Scientifique George Besse, 30000 Nimes

Je voudrais exprimer tout d'abord mes remerciements à ceux qui m'ont fait l'honneur de participer au jury de cette thèse,

Monsieur Jean-Paul Haton, Professeur à l'Université de Nancy 1, qui m'a accueilli dans l'équipe Reconnaissance de Formes et Intelligence Artificielle du CRIN, et qui me fait l'honneur de présider ce jury. Je le remercie pour ses encouragements et le grand intérêt qu'il a toujours porté à mes travaux.

Monsieur Serge Castan, Professeur de l'Université Paul-Sabatier à Toulouse, et Monsieur René Husson, Professeur à l'Institut National Polytechnique de Lorraine, qui ont accepté d'être rapporteurs de ce travail et de siéger à ce jury. Qu'ils trouvent ici l'expression de ma gratitude pour l'intérêt qu'ils ont porté à ce travail.

Monsieur Gérard Giraudon, Directeur de Recherche à l'INRIA de Sophia-Antipolis, qui m'a proposé un sujet de recherche passionnant et original et qui a accepté d'examiner ce travail et de le juger. Je tiens particulièrement à le remercier pour les nombreuses discussions qui ont guidé mes travaux. J'ai particulièrement apprécié les qualités humaines et scientifiques que notre collaboration a permis de révéler.

Monsieur Gérard Masini, Chargé de Recherche CNRS au CRIN, qui m'a fait l'honneur de participer à ce jury. L'intérêt qu'il a toujours porté à mes travaux et les conseils qu'il a bien voulu me prodiguer ont été pour moi les plus précieux des encouragements. Il m'a apporté la garantie d'un soutien et d'une amitié. Je le remercie pour sa disponibilité, pour les très nombreux moments qu'il a consacré à la lecture de mes articles et de ce mémoire, ainsi que pour tous les entretiens que nous avons pu avoir ensemble.

Monsieur Roger Mohr, Professeur à l'Ecole Nationale Supérieure d'Informatique et de Mathématiques Appliqués de Grenoble, qui a assumé la direction de cette thèse. Il m'a encouragé de ses conseils tout au long de mon apprentissage du métier de chercheur. Je tiens particulièrement à le remercier pour la liberté qu'il m'a accordé pendant mes recherches, liberté sans laquelle je n'aurais pas pu mener à bien l'étude de ce difficile problème.

Je ne peux pas oublier de remercier Yolande Anglade et Abderafiia Koukam pour leur amitié et pour les moments qu'ils ont passés à m'écouter ou à me relire, Dominique Antoine pour ses remarques issues d'une lecture passionnée.

Cette thèse a été réalisée dans le groupe Vision du CRIN et je tiens à remercier tous ceux qui, de près ou de loin, en ont facilité l'aboutissement, notamment Salvatore Tabbone, Eric

Thirion, Karl Tombre et Brigitte Wrobel-Dautcourt. Je remercie Tahar Khammaci qui m'a initié à l'informatique, Bechir El-Ayeb, avec qui j'ai partagé de longues soirées au CRIN, Sarra et Jalel Mzali, Monjia et Toufik Gabsi, Philippe Anglade, mes deux compagnons du bureau Hassan Bentefrit et Pierre Marquis, ainsi que tous ceux qui m'ont apporté leur soutien amical.

Enfin, que tous mes proches, parents et amis, trouvent ici le témoignage de l'expression de ma plus sincère gratitude, en particulier Kamila, Myriam et Abdelghani, que je remercie pour leur patience, soutien et leurs encouragements.

- Merci à tous -

Résumé

La détection de contours a pour objectif l'acquisition d'une représentation compacte de toutes les propriétés significatives de la scène perçue. Elle doit être efficace et fiable car la validité et l'efficacité ainsi que les possibilités de réalisation des traitements ultérieurs y sont directement liées. Cependant, la présence du bruit dans l'image rend difficile la conception d'un algorithme général de détection de contours satisfaisant d'une manière optimale ces exigences. C'est la raison pour laquelle de nombreux détecteurs ont été proposés, souvent différents par leur objectif ainsi que par leurs propriétés mathématiques et algorithmiques. Le problème auquel se trouvent souvent confrontés les utilisateurs de la détection de contours est le choix du détecteur le plus approprié pour l'application envisagée.

Le thème central de cette thèse est la détection de contours dans des images à niveaux de gris, en visant la mise en œuvre d'une sélection automatique de détecteurs et de leurs paramètres les plus appropriés pour mettre en évidence un contour donné.

Pour disposer d'un ensemble d'outils nécessaires à la détection de contours, nous proposons deux détecteurs de contours originaux. Le premier est destiné aux lignes de crête. Il est optimal au sens de Canny et implanté efficacement par des équations aux différences. Le second est un détecteur de contours fermés de type marches fondé sur des techniques d'apprentissage. Il est destiné à des images comportant quelques objets posés sur un fond. Cet algorithme peut être utilisé pour la détection de contours et pour l'extraction de régions.

La définition d'un critère de sélection de détecteurs et le calcul automatique de leurs paramètres nécessite la spécification de la relation de cause à effet entre les caractéristiques du contour et les propriétés des détecteurs. Dans ce contexte, nous proposons une étude bibliographique approfondie et une étude théorique mettant en évidence l'influence des attributs des contours sur les performances des détecteurs. Cette étude validée par des expérimentations a permis d'obtenir des résultats originaux.

En ce qui concerne la sélection des détecteurs de contours, nous proposons une approche incrémentale qui consiste à choisir un seul détecteur à chaque itération. La combinatoire de l'approche proposée est faible grâce à l'utilisation d'heuristiques. Le système implanté est composé d'algorithmes originaux. Il a l'avantage de fournir une solution efficace assurant le compromis entre deux critères de performance qui sont de nature antagoniste : la détection et la localisation.

Mots Clés :

détection de contours, sélection de détecteurs, analyse de contours, formation de l'image, fusion des contours, fermeture de contours, multi-échelle.

Table des matières

Introduction générale	7
I Introduction	9
1 La vision par ordinateur	13
1.1 Définition	13
1.2 Organisation d'un système de vision	13
1.2.1 Extraction de primitives	15
1.2.2 Reconstruction 3D	16
1.2.3 Interprétation	17
1.3 Conclusion	18
2 La formation de l'image	19
2.1 Introduction	19
2.2 La fonction image	19
2.3 Les procédés de formation	20
2.3.1 Image de rayon X	20
2.3.2 Image optique	20
2.4 Que peut-on extraire d'une image ?	22
3 La détection de contours	25
3.1 Notion de contours	25
3.2 La détection de contours est un problème mal posé	27
3.3 Organisation d'un détecteur de contours	28
3.3.1 Lissage de l'image	29
3.3.2 Différentiation de l'image	30
3.3.3 Commutativité des opérations de lissage et de différentiation	32
3.4 Conclusion	33
4 Les travaux antérieurs	35
4.1 Introduction	35

4.2	Détecteurs autonomes de contours	35
4.2.1	Détecteurs informels de marches	35
4.2.2	Détecteurs optimaux de marches	36
4.2.3	Détecteurs d'autre types de contours	39
4.2.4	Détecteur de contours rectilignes	40
4.3	Détecteurs contextuels de contours	41
4.3.1	Détecteurs guidés par le modèle	41
4.3.2	Générateur de détecteurs	42
4.4	L'approche multi-échelle	43
4.4.1	Définition	43
4.4.2	La fusion des contours résultats	43
4.4.3	La problématique de la détection multi-échelle de contours	45
4.5	Le seuillage	46
4.6	Conclusion	48
5	Problématique et Conclusion	51
5.1	Caractérisation des résultats de la détection de contours	51
5.2	Problèmes liés à la définition du contour	52
5.3	Problèmes liés aux détecteurs	53
5.3.1	Dérivation de détecteurs	53
5.3.2	Implantation d'un détecteur	53
5.3.3	Evaluation des résultats d'un détecteur	54
5.4	Méthodologie de détection de contours	55
5.5	Conclusion	55
II	Les détecteurs de contours	57
1	Quelques détecteurs optimaux	61
1.1	Détecteur de J.F. Canny	61
1.1.1	Critères de Canny	61
1.1.2	Le détecteur de marche	62
1.2	Détecteur de R. Deriche	64
1.3	Détecteur de J. Shen et S. Castan	65
1.4	Conclusion	65
2	Un détecteur optimal de lignes de crête	67
2.1	Introduction	67
2.2	La dérivation du détecteur 1D	68
2.3	Le détecteur 2D	70
2.3.1	Extension du filtre en deux dimensions	70

2.3.2	Identification du contour	71
2.4	Implantation du détecteur 2D	74
2.4.1	Représentation de la fonction de détection	74
2.4.2	Représentation de la fonction de projection	76
2.5	Résumé de l'algorithme de détection des lignes de crête	76
2.6	Evaluation des performances du détecteur	77
2.6.1	Performances du détecteur proposé	77
2.6.2	Choix des paramètres du détecteur	79
2.6.3	Complexité de l'algorithme	80
2.6.4	Résultats expérimentaux	81
2.6.5	Discussion	82
2.7	Conclusion	83
2.8	Annexe A	84
2.9	Annexe B	85
3	Détecteur de contours à partir de l'histogramme	87
3.1	Introduction	87
3.2	Approximation de l'histogramme	88
3.2.1	Estimation des paramètres	89
3.2.2	Valeurs initiales des paramètres à estimer	90
3.2.3	Décomposition d'histogramme	90
3.3	Détermination des seuils	91
3.4	Principe de l'algorithme	93
3.5	Evaluation de l'algorithme	94
3.5.1	Résultats expérimentaux	94
3.5.2	Comparaison de l'algorithme	98
3.6	Conclusion	101
3.7	Annexe A	102
III	Sélection automatique de détecteurs de contours	107
1	Présentation globale de l'approche	111
1.1	Position du problème	111
1.2	Principe de l'approche de sélection	112
1.2.1	Quelles informations peut-on utiliser?	112
1.2.2	Exemple d'un détecteur de contours	113
1.2.3	Critère de sélection	114
1.3	Description globale du système de sélection	115
1.3.1	Schéma général de l'algorithme	115
1.3.2	Bibliothèque de détecteurs	116

1.3.3	Architecture du système	116
1.4	Conclusion	118
2	Sélection d'un détecteur	119
2.1	Principe de l'algorithme de sélection d'un détecteur	119
2.2	Analyse du contour exemple	119
2.2.1	Estimation du bruit	120
2.2.2	Identification du modèle de contour	123
2.2.3	Estimation des attributs d'un contour de type marche simple	124
2.2.4	Estimation des attributs d'une double marche	126
2.2.5	Segmentation du contour exemple	128
2.3	Choix du détecteur	129
2.3.1	Choix des détecteurs applicables	129
2.3.2	Sélection du meilleur détecteur	130
2.4	Description de détecteurs	132
2.4.1	Attributs intrinsèques d'un détecteur	132
2.4.2	Attributs de performance	133
2.4.3	Influence de l'orientation d'un contour sur les performances d'un détecteur	135
2.4.4	Influence des attributs image d'un contour sur les performances d'un détecteur	137
2.4.5	Influence des propriétés d'un détecteur sur ses performances	141
2.4.6	Une synthèse	143
2.5	Etude d'un exemple	144
2.5.1	Résultats de l'étape d'analyse du contour	144
2.5.2	Résultats de la sélection	147
2.6	Conclusion	149
2.7	Annexe A	151
2.7.1	Cas d'une marche simple	152
2.7.2	Cas d'une double marche	154
2.8	Annexe B	157
2.9	Annexe C	159
3	Analyse des résultats	163
3.1	Analyse des résultats	163
3.2	Fusion d'images de contours	164
3.2.1	Notre approche de fusion	165
3.2.2	La règle de fusion des contours	166
3.2.3	Résultats de la fusion	168
3.3	Evaluation	168
3.3.1	Génération du contour exemple	169

3.3.2	Mesure de la différence	170
3.4	Fermeture des contours	171
3.5	La mise en correspondance de segments	174
3.5.1	Génération d'hypothèses	175
3.5.2	Estimation de probabilités	176
3.5.3	Initialisation des probabilités	177
3.5.4	Sélection de correspondants	178
3.5.5	Résumé de l'algorithme	179
3.6	Conclusion	180
3.7	Annexe A	181
3.8	Annexe B	182
3.9	Annexe C	185
3.10	Annexe D	187
4	Résultats expérimentaux et conclusion	189
4.1	Les règles de sélection	189
4.1.1	Cas des marches simples	190
4.1.2	Cas des doubles marches	192
4.2	Les résultats expérimentaux	193
4.3	La sélection de détecteurs	198
4.3.1	Le bilan	198
4.3.2	Comparaison avec le système IMPRESS	200
4.4	Les améliorations	200
4.4.1	Analyse du contour exemple	201
4.4.2	Acquisition de règles de sélection	202
4.4.3	Seuillage automatique	203
4.4.4	La mise en correspondance	203
4.5	Les intérêts du système proposé	204
	Conclusion générale	207
	Bibliographie	211

Introduction générale

L'image est la donnée d'un système de vision par ordinateur dont l'objectif est la construction de descriptions explicites et sensées de la scène représentée. L'image formée par une projection de cette scène sur un plan 2D est une structure contenant une quantité considérable d'informations, ambiguës et difficiles à exploiter de façon automatique. Par conséquent, il est souvent intéressant d'acquérir une représentation de l'image explicitant les informations utiles face à un but à atteindre. Parmi ces informations, on distingue les discontinuités des propriétés photométriques, géométriques et physiques des objets de la scène. De telles informations, appelées contours physiques, engendrent des variations locales dans la fonction image, appelées contour. Les variations les plus utilisées sont le point d'inflexion (ou point de discontinuité selon certains auteurs), appelé contour de type marche et les optima locaux, appelés contour de type crête.

L'objectif de la détection de contour est alors de mettre en évidence ces variations et d'identifier les phénomènes physiques qui leur sont associés. Cette phase initiale requiert une attention particulière car elle va déterminer les possibilités de réalisation, l'efficacité et la validité des traitements ultérieurs. Pour répondre à de telles exigences, la détection de contour doit fournir une représentation complète et correcte de l'image dans le sens où elle doit capter toutes les propriétés significatives de la scène perçue.

Pour ce faire, les dérivées de l'image sont calculées. Or, la différentiation d'une fonction discrète et bruitée (l'image) est un problème mal posé; c'est-à-dire que le résultat de la différentiation dépend du bruit. D'où la nécessité d'une étape de lissage pour le réduire. Cependant, le lissage à son tour possède des effets indésirables, voir une perte d'informations. La présence du bruit dans l'image rend difficile la conception d'un algorithme général de détection de contour qui répond aux exigences de la vision par ordinateur. C'est la raison pour laquelle de nombreux détecteurs ont été proposés, souvent différents par leurs propriétés mathématiques et algorithmiques (les propriétés du filtre de lissage, l'ordre et le type de l'opérateur de différentiation) et leur objectif.

Le problème auquel se trouvent souvent confrontés les concepteurs de système de vision est le choix d'un détecteur à utiliser dans l'application envisagée. Au cours de notre travail, nous nous sommes intéressés à la définition d'un critère de sélection de détecteurs face à un but à atteindre. Plus précisément, l'objectif de cette thèse est l'étude d'une approche concernant la sélection automatique de détecteurs et de leurs paramètres nécessaires pour mettre en évidence un contour donné par un utilisateur. Notre contribution centrale est la conception et l'implantation de deux détecteurs de contours de type marche et ligne de crêtes, la mise en œuvre d'un système de sélection automatique de détecteurs et la spécification de la relation de cause à effet entre les caractéristiques d'un contour, les

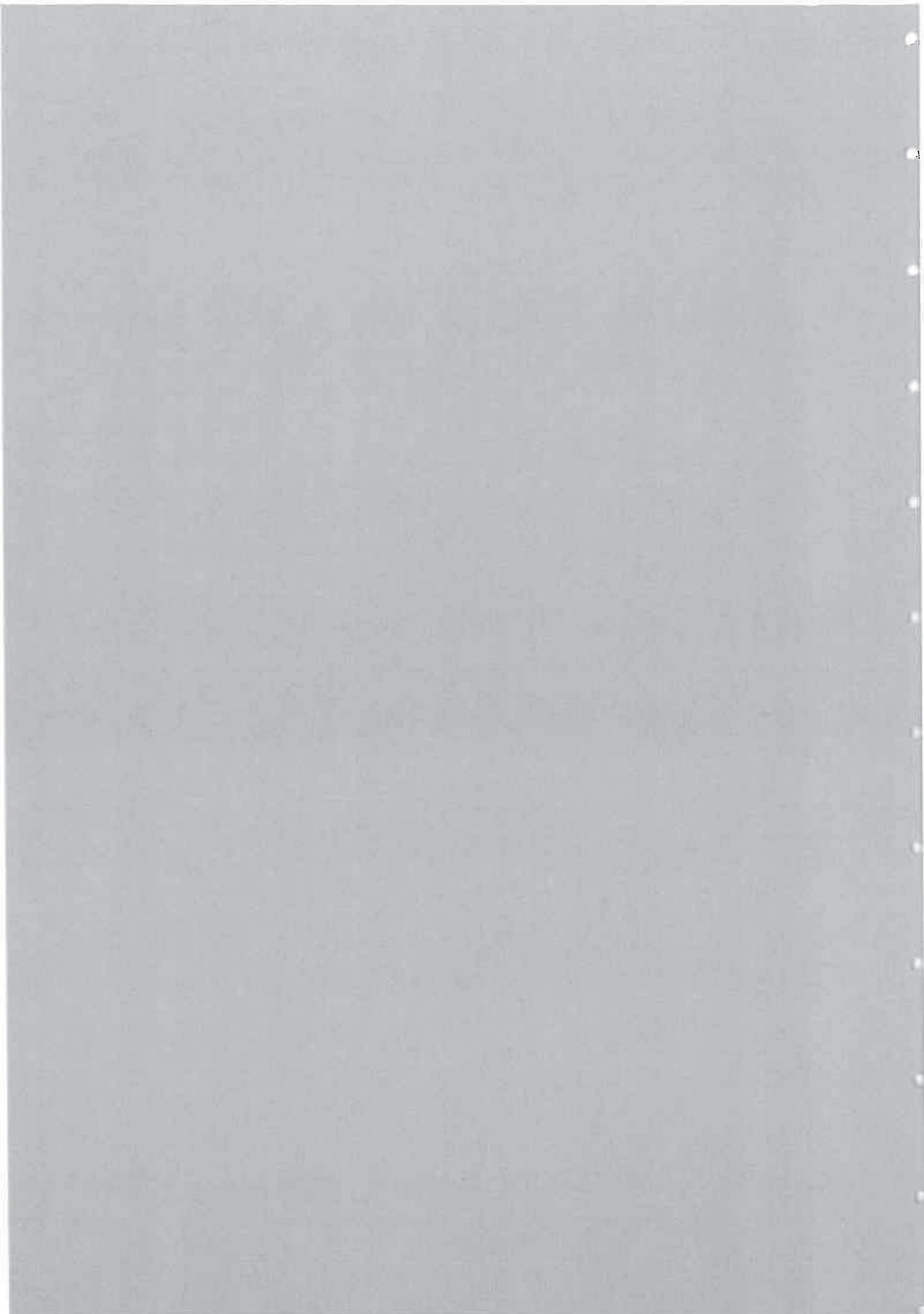
propriétés d'un détecteur et ses performances.

Le plan du document s'articule autour des trois parties. Dans la première, nous présentons une synthèse sur la détection de contours. Nous avons mené cette étude pour acquérir le savoir nécessaire à la définition du critère de sélection de détecteurs et le calcul automatique de leurs paramètres. Nous présentons dans le chapitre 1 une vue globale de la vision par ordinateur où nous montrons le rôle de la détection de contours. Dans le chapitre 2, nous examinons le processus de formation de l'image et montrons l'intérêt de l'information contour pour la perception visuelle. Le chapitre 3 est consacré à l'analyse de l'influence des caractéristiques du contour et des propriétés des détecteurs sur ses performances. Nous donnons dans le chapitre 4 les travaux concernant la détection de contour qui ont influencé le notre et explicitons nos contributions dans la détection multi-échelle de contours et dans l'élimination des faux points de contours. Enfin, nous menons dans le chapitre 5 des réflexions concernant la problématique de la détection de contours.

La deuxième partie décrit deux détecteurs de contours que nous avons conçu et implanté. Nous présentons au chapitre 1 les critères de performance de Canny qui sont à la base de l'un de nos deux détecteurs, le détecteur de R. Deriche et la famille des détecteurs de J. Shen et S. Castan. Tous ces outils font partie du système de sélection de détecteurs que nous avons développé. Nous décrivons dans le chapitre 2 un détecteur optimal de lignes de crête issu des critères de Canny. Enfin, le chapitre 3 est consacré à un détecteur de contours de type marche à partir de la distribution fréquentielle de l'image.

La troisième partie présente l'approche de sélection de détecteurs de contours et de leurs paramètres pour mettre en évidence un contour donné par un utilisateur. Le chapitre 1 de cette partie décrit une vue globale de cette approche. Nous donnons au chapitre 2 le critère de sélection, les algorithmes pour l'analyse du contour donné et une étude de la relation contour-détecteur pour compléter celle donnée en première partie. Au chapitre 3 nous présentons l'algorithme pour l'analyse des résultats de l'étape de sélection. Enfin, nous montrons au chapitre 4, les résultats obtenus et les améliorations qui restent à apporter.

PARTIE I
Introduction



Résumé

L'objectif de cette partie est de présenter à la fois une synthèse et la problématique de la détection de contours. Cette étude permet de situer les problèmes abordés et d'acquérir une expertise nécessaire à la résolution du problème abordé dans la troisième partie. Nous exposons dans le chapitre 1 l'organisation d'un système général de vision dans lequel nous situons la détection de contours. Le chapitre 2 expose le procédé de fabrication de l'image et met en évidence l'intérêt du contour pour la perception visuelle. Les notions de contours, de détection de contours et de détecteurs ainsi que les différents travaux sous-jacents sont présentées dans le chapitre 3 et 4. Enfin, le chapitre 5 expose les différents problèmes de la détection de contours et notre apport dans le domaine.

1

La vision par ordinateur

Nous présentons dans ce chapitre l'organisation d'un système de vision par ordinateur qui est simple et dans lequel nous situons la détection de contours et nous montrons son intérêt.

1.1 Définition

La vision par ordinateur est l'une des outils qui vise à libérer l'être humain des tâches répétitives et fastidieuses, souvent sujettes à de nombreuses erreurs. Elle tend à donner aux machines la faculté de voir pour se déplacer, comprendre et être autonomes quelque soit leur environnement. Cependant, cette définition ne reflète qu'une ambition de chercheurs et ne précise ni les problèmes sous-jacents, ni les outils nécessaires à la réalisation de ces tâches. Nous présentons ci-dessous deux définitions plus explicites de la vision par ordinateurs :

D.H. Ballard et M.C. Brown [Bal 82] : *la construction de descriptions explicites et sensées d'objets physiques à partir d'images. Les descriptions sont un pré-requis pour reconnaître, manipuler et penser au sujet des objets.*

A. Lux [Lux 85] : *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 (l'objectif) du système.*

1.2 Organisation d'un système de vision

Un système de vision par ordinateur peut être défini par ses entrées, sa sortie et l'ensemble des opérations nécessaires à la description de la scène [Bar 81a]. Il a en entrée une ou plusieurs images (stéreo, séquence) représentant la scène, enregistrées par des capteurs différents (Caméra CCD, Scanner, laser, infra-rouge, etc.). La figure 1.1 présente un exemple d'une image à niveaux de gris (image de luminance).

La sortie du système est une interprétation sémantique ou une description concise de la scène représentée dans l'image (fig. 1.2). Le type et le nombre d'images ainsi que le contenu de la description dépendent de l'objectif et du contexte du système [Nev 78]. L'objectif du système est défini par l'ensemble des tâches qu'il est capable d'accomplir. Le contexte est défini par la structure de l'univers du système (type de scène, type et nombre

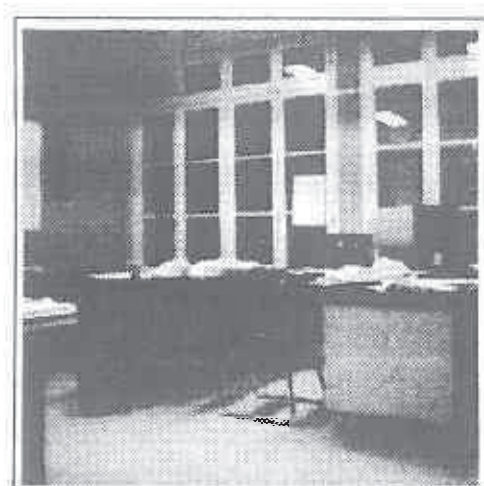


Figure 1.1. Un exemple d'image à niveaux de gris.

d'objets, leur taille, leur position, leur orientation, etc.) et par les conditions de prise de vue.

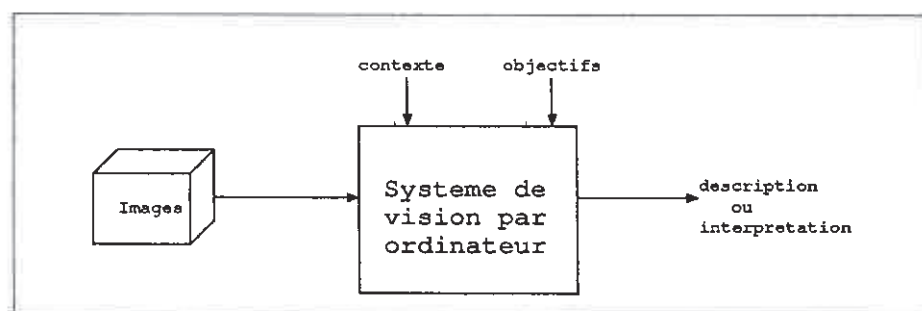


Figure 1.2. Vue générale d'un système de vision par ordinateur [Cle 90]

Pour atteindre l'objectif fixé, un système de vision par ordinateur doit avoir des connaissances sur son univers, en particulier l'ensemble des objets qu'il est susceptible de reconnaître. Cet ensemble de connaissances est appelé modèle. Le contenu d'un modèle varie d'un système à l'autre et dépend du domaine d'application. Ainsi, le modèle d'un système de vision général (un système dont les capacités sont comparables à celles de l'être humain dans un environnement composé d'objets complexes) peut inclure [Bar 81a]:

- la description des objets de la scène : leur forme, la nature de leur surface (texture, couleur, etc.) et un ensemble de relation entre les objets (partie-de, est-un, posé-sur, adjacent, au-dessus, etc.).
- la description des sources lumineuses : leur localisation, leur extension spatiale, leur intensité, leur couleur, etc.,
- la description photométrique de la scène et la description géométrique du système de formation d'images. Ces descriptions permettent de prédire comment une scène par-

ticulière vue par un observateur à partir d'un point de vue particulier sous certaines conditions d'illumination est transformée en une image.

La nature des informations de l'image et du modèle fait que l'interprétation ou la description ne peut être accomplie qu'en plusieurs étapes. Le but de ces étapes est de transformer progressivement l'information contenue dans l'image initiale jusqu'à l'obtention d'informations ayant un sens dans notre univers. Le nombre et la nature de ces étapes différent d'un système à l'autre ; elles dépendent par ailleurs du contexte et de l'objectif. Ainsi, un système de vision par ordinateur peut être défini comme étant une succession de niveaux hiérarchiques assurant la transformation progressive de données physiques en une représentation sémantique. Chaque niveau fait appel à un ensemble spécifique de connaissances [Hat 87].

Pour mieux comprendre la structure d'un système de vision par ordinateur et montrer le rôle de la détection de contours (à laquelle nous nous intéressons), nous présentons un schéma classique et dominant (illustré par la figure 1.3) d'un système de vision 3D dont l'objectif est l'interprétation des scènes d'intérieur. Supposons que les données du système soient une ou plusieurs images à niveaux de gris enregistrées successivement par une seule caméra, ou par l'emploi d'autant de caméras situées à des emplacements différents que d'images désirées. Nous nous intéressons aux principaux modules qui sont : l'extraction de primitives, la reconstruction 3D et l'interprétation.

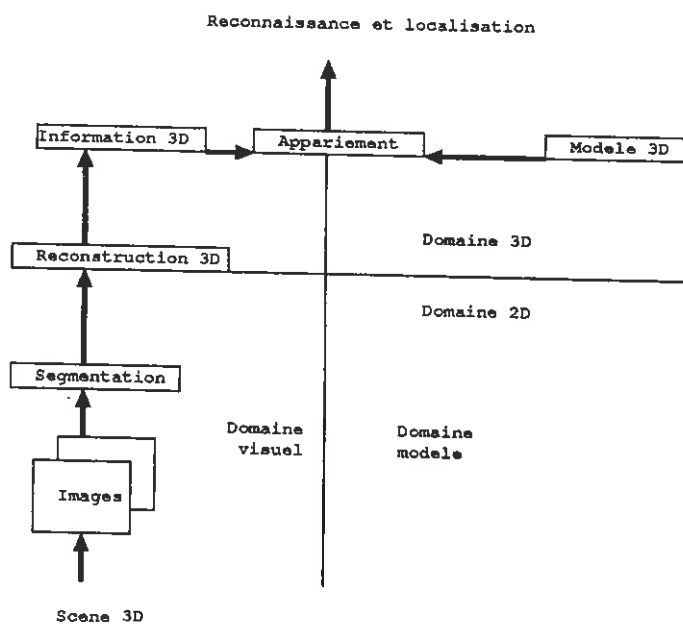


Figure 1.3. Un exemple de système de vision par ordinateur fondé sur la reconstruction tridimensionnelle [Qua 89].

1.2.1 Extraction de primitives

L'extraction de primitives est souvent la première étape dans un système de vision qui vise à réduire la quantité d'informations contenues dans l'image. Elle consiste à sélectionner les informations les plus significatives, en particulier celles relatives aux frontières des

objets. Les frontières des objets (ou contours physiques selon certains auteurs) constituent une information très importante pour la vision humaine. En effet, il a été montré qu'un objet peut être identifié uniquement à partir de ses contours physiques [Bal 82].

Aux frontières des objets de la scène correspondent souvent des discontinuités dans la fonction image. Ces discontinuités sont appelées points de contours ou simplement contours. La recherche des frontières peut être alors effectuée par la détection des contours, puis le regroupement de ces derniers pour former les frontières. Dans le cas où le fossé entre les contours détectés et les frontières est assez grand (ce qui est souvent le cas, comme le montre l'image de contours décrite par la figure 1.4), une représentation intermédiaire devient nécessaire. Le problème de recherche des frontières qui définissent une entité sémantique est transformé en un problème de détection de contours qui constituent une entité indépendante de toute interprétation. La détection de contours fait l'objet de cette thèse, et le chapitre 3 traitera tout particulièrement la définition de la détection de contours ainsi que les notions sous-jacentes.

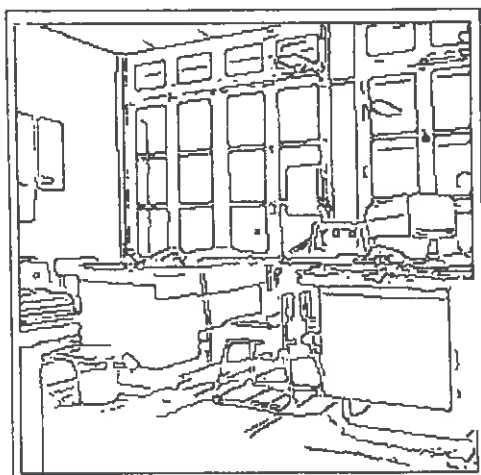


Figure 1.4. Un exemple d'image de contours (extraits de l'image présentée dans la figure 1.1).

Il existe une approche "duale" à la détection de contours et dont l'objectif est aussi la recherche des frontières des objets. La philosophie de cette approche est de chercher les zones de l'image qui correspondent à un objet ou à une partie de l'objet. Une zone, ou région, est un ensemble de points adjacents possédant une distribution du niveau de gris "presque" uniforme. Les frontières des objets correspondent aux bords des régions. Du point de vue algorithmique, cette approche consiste à partitionner l'image en régions homogènes en se fondant sur un critère de proximité et d'homogénéité. Généralement, l'image est partitionnée en groupement de pixels contigus et de valeurs de niveau de gris similaires. Les partitions ainsi obtenues sont raffinées par une suite de fusions ou de fusions-divisions. Le lecteur intéressé peut se référer aux travaux de [Zuc 76, Naz 84, Har 85b, WD 88].

1.2.2 Reconstruction 3D

Les informations obtenues dans l'étape précédente sont bidimensionnelles, tandis que l'interprétation est effectuée dans un espace tridimensionnel. La reconstruction 3D a

pour objectif alors d'inférer à partir des images, ou des primitives résultats de l'étape précédente, des informations 3D qui peuvent être l'orientation et la profondeur des surfaces visibles ainsi que les coordonnées 3D de la scène. Les approches existantes permettant de reconstruire ces caractéristiques 3D diffèrent essentiellement par le nombre d'images à considérer. A ce sujet, on distingue :

- la vision monoculaire qui consiste à déduire à partir d'une seule image, l'orientation et la profondeur des surfaces visibles. La plupart des approches proposées sont fondées sur les propriétés photométriques, géométriques et physiques des objets contenus dans la scène et les variations en conséquence du niveau de gris. On trouve des approches fondées sur l'information contour "*shape from contour*" [Bar 81b, Bra 84, Ulu 88], d'autre sur l'information de texture "*shape from texture*" [Bal 82], d'autre sur le niveau de gris "*shape from shading*" [She 86, Hor 90]. L. Quan [Qua 89] propose une approche fondée sur le regroupement perceptuel des segments de contours et l'utilisation des points de fuite (un point de fuite est le point d'intersection de droites parallèles transformées selon une certaine projection perspective). Cette approche fondée sur l'effet de perspectives peut être qualifiée de "*shape from perspective*".
- la stéréovision est une technique importante pour reconstruire des informations tridimensionnelles à partir de deux (ou plusieurs) images bidimensionnelles représentant des prises de vue d'une même scène sous un angle légèrement différent. Elle cherche la correspondance entre des primitives homologues dans les images. La primitive peut être le point (par exemple, le point de contours [She 86, Zah 89], de Moravec [LL 86]), le segment [LL 86, Aya 88], la région [WD 88, Gag 89], la facette [Cha 91]. Une fois que les primitives homologues sont détectées dans les images stéréoscopiques, les coordonnées des points correspondants dans l'espace peuvent être déterminées par triangulation.
- le mouvement, contrairement à la stéréovision (qui utilise une des images prises en différentes positions spatiales), utilise une séquence d'images d'une scène qui, soit évolue dans le temps, soit est prise par un capteur mobile. On peut citer au moins deux approches : la première consiste en la mise en correspondance entre les points caractéristiques dans les images successives et la détermination de l'information 3D [Fau 88, Bla 90, Arb 90]. La seconde appelée *flot optique* consiste à calculer le champ de l'image en déplacement, puis la détermination du champ 3D correspondant [Hor 81].

1.2.3 Interprétation

La nature de l'information fournie par les étapes précédentes et celle contenue dans le modèle rendent nécessaire l'introduction d'une étape de regroupement des primitives 2D et 3D. Cette étape vise à former suivant la loi de Gestalt [Low 87, Moh 89b, Qua 89], d'autres primitives de haut niveau (e.g., des facettes, des arêtes, des objets, etc.) à partir de celles données dans les étapes décrites précédemment.

L'interprétation est le processus de mise en correspondance entre les primitives de l'image et le modèle. Son objectif est de déterminer l'identité des objets perçus, leurs positions dans l'image ou par rapport au capteur, et éventuellement d'autres informations nécessaires pour une application précise. Il existe plusieurs outils pour réaliser la

mise en correspondance. Citons entre autre, la relaxation discrète [Moh 88], la relaxation continue [Fau 81], la recherche d'une clique maximale dans un graphe [Hor 89], la prédiction/vérification d'hypothèses [Lux 85, Aya 86], etc.

1.3 Conclusion

Nous avons présenté dans ce chapitre la structure d'un système de vision par ordinateur dans lequel nous avons montré le rôle de la détection de contours. Cette dernière constitue le sujet de notre thèse.

Les domaines d'application de la vision par ordinateur sont variés et leurs nombres ne cessent d'augmenter. La robotique, l'astrologie, la télédétection, la bureautique et la médecine constituent des exemples de tels domaines. Une grande variété de solutions a été proposée pour la conception et l'implantation d'un système de vision par ordinateur ; citons les systèmes ACRONYM [Bro 81b], LFF [Bol 82], VISIONS [Han 86], HYPER [Aya 86], TRIDENT [Thi 89]. La conception et l'implantation d'un système de vision par ordinateur restent délicates du fait qu'il faut mettre en œuvre une quantité énorme de connaissances souvent imprécises et de natures diverses : le procédé de formation de l'image, le domaine d'application, les objets de la scène et leurs propriétés, le contexte, etc. [Hat 87].

Cette particularité de la vision par ordinateur engendre nécessairement les problèmes de représentation de toutes ces connaissances et de structuration des systèmes. Il n'existe pas de principes généraux pour spécifier la façon d'intégrer les différents modules d'un système de vision destiné à accomplir une tâche donnée [Ros 84]. Cependant, le développement des autres thèmes de l'intelligence artificielle a permis à la vision par ordinateur d'avancer dans cette voie.

Une autre particularité est due au fait que les données d'un système de vision par ordinateur sont erronées. En effet, l'image est une fonction discrète et bruitée. En général, elle contient une quantité considérable d'informations qui sont ambiguës et difficiles à recenser. Par conséquent, la modélisation de l'image ainsi que son analyse doivent être précédées par sa compréhension. C'est pourquoi nous lui consacrons le chapitre suivant.

2

La formation de l'image

Ce chapitre introduit la fonction image. Après l'avoir défini, nous exposons deux procédés différents de formation d'images. En examinant le contenu des images résultats de ces deux procédés, nous montrons que le contour est une représentation de phénomènes physiques présents dans la scène et qu'il possède des significations différentes selon le type de procédé de formation utilisé.

2.1 Introduction

L'image est la donnée d'un système de vision par ordinateur. Elle constitue la représentation d'une scène 3D et inclut des informations relatives aux phénomènes physiques présents dans cette scène. Il existe plusieurs techniques différentes pour fabriquer une image. Le choix d'une de ces techniques est lié aux informations de la scène que l'on veut inclure dans l'image. Par conséquent, c'est l'objectif du système de vision qui détermine la technique de fabrication à utiliser.

La description de la scène représentée dans l'image passe nécessairement par la compréhension de l'image elle-même : que contient-elle comme informations ? Que représentent ces informations dans la scène ? Quelles sont les informations à considérer pour atteindre l'objectif fixé ? Nous présentons dans ce chapitre quelques exemples de procédés de fabrication d'images. Ces exemples nous conduisent à mettre en évidence la signification de la notion de contours et l'utilité de cette information pour un système de vision.

2.2 La fonction image

L'image est une représentation d'une scène, formée par contact entre les rayons de lumière provenant des objets de la scène et un capteur. Elle peut être représentée mathématiquement par une fonction à plusieurs variables spatiales ou spatiales-temporelles $I(X)$ dite *fonction image*. La valeur de $I(X)$ exprime les informations de la scène qu'on veut prendre en considération. Les caractéristiques les plus utilisées sont la luminance et la couleur. Dans le cas de la luminance, $I(X)$ est une fonction de $\mathcal{R}^2 \rightarrow \mathcal{R}$. Dans le cas de la couleur $I(X)$ est une fonction de $\mathcal{R}^2 \rightarrow \mathcal{R}^3$, plus précisément, $I(X)$ est un vecteur de fonctions que l'on note $(f_{rouge}, f_{bleu}, f_{vert})(X)$.

La fonction image la plus utilisée est discrète et bornée, c'est-à-dire que les arguments et les valeurs de la fonction sont des entiers compris entre deux valeurs connues à l'avance. L'image de luminance à deux arguments, dite à deux dimensions est au centre de notre

intérêt. Elle prend ses valeurs dans l'ensemble $\{0, \dots, 255\}$, où le 0 et 255 correspondent respectivement au blanc et au noir et les valeurs intermédiaires correspondent aux différentes transitions entre les deux (voir fig. 1.1). Cette image est appelée souvent *image à niveaux de gris*. Chaque point (x, y) du plan image est appelé *pixel* ou *pel*, diminutif de *picture elements*. Le plan image est divisé en trames de forme rectangulaire ou hexagonale permettant ainsi de définir la notion de voisinage et de distance entre pixels.

2.3 Les procédés de formation

La formation de l'image fait intervenir un dispositif physique capable de détecter les informations de la scène et de les coder dans l'image. Les dispositifs physiques sont multiples. Ils diffèrent essentiellement par le type de capteur utilisé (caméra, scanner, rayon X, laser, etc.). Les images formées par des procédés différents représentent des aspects différents de la scène. Pour mieux comprendre, nous présentons ci-dessous deux procédés différents de formation d'images.

2.3.1 Image de rayon X

Considérons un système de formation d'images radiographiques conventionnelles à géométrie parallèle et possédant une source ponctuelle de rayons X (fig. 2.1). La création d'une telle image est due à l'atténuation différentielle d'un faisceau de rayons X par les structures de la matière composant l'objet qui absorbe les photons X de manière variable. La relation entre l'intensité I observée au point (x, y) de l'image et les différents facteurs d'atténuation est donnée par [Hei 88] :

$$I(x, y) = \int I_0(\lambda) e^{-\int (\mu_p(x, y, z, \lambda) + \mu_r(x, y, z, \lambda) + \mu_c(x, y, z, \lambda)) dz} \lambda$$

avec λ la longueur d'onde, $I_0(\lambda)$ la répartition spectrale de la source et μ_p , μ_r , μ_c les facteurs d'atténuation. Ces facteurs peuvent être estimés à partir des propriétés atomiques des objets (l'énergie du photon, le numéro atomique, la densité, le nombre d'électrons, etc.). La figure 2.2 présente une image rayon X à niveaux de gris.

2.3.2 Image optique

Considérons un autre exemple de système de fabrication d'images utilisant une caméra de type CCD (fig. 2.3). Supposons que la caméra soit correctement focalisée : un rayon de lumière réfléchi par une surface infinitésimale S_o de l'objet est projeté sur une surface S_i du plan image et aucun rayon d'une autre surface de l'objet n'est projeté sur S_i .

La description photométrique de ce système permet d'introduire les trois paramètres suivants :

- la radiance réfléchie L_i définie comme étant la somme de flux émis par unité de surface vue en perspective par unité d'angle solide,
- l'irradiance E_i d'une image qui est le flux incident par unité de surface dans le plan image,

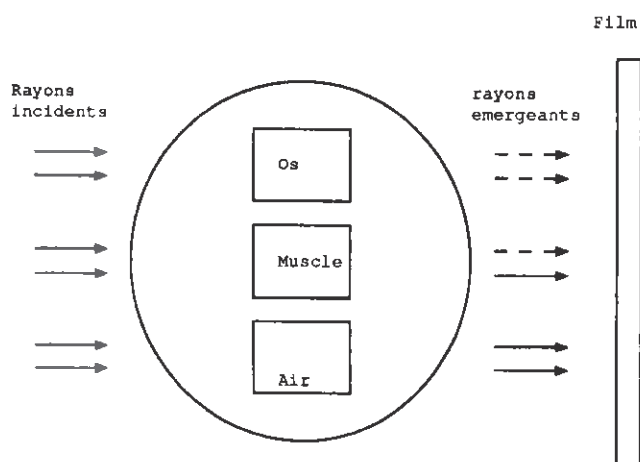


Figure 2.1. La géométrie de la formation d'image rayon X

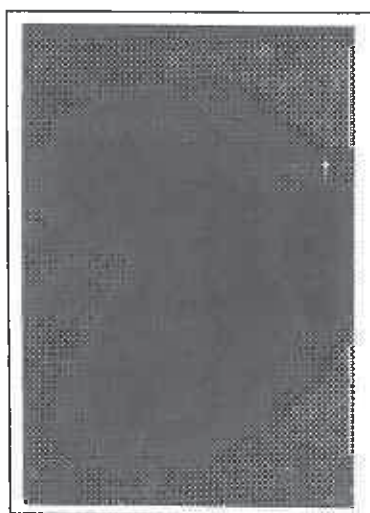


Figure 2.2. Une image rayon X à niveaux de gris

- la réflectance r d'une surface qui représente la fraction de lumière incidente réfléchiée par la surface dans une direction particulière.

La relation entre ces différents paramètres est donnée par les deux équations suivantes [Hor 77, She 86] (voir figure 2.3) :

$$E_i = \frac{\pi}{4} \left(\frac{d}{f_p} \right)^2 L_i \cos^4(\alpha) \quad (2.1)$$

$$L_i = E r \quad (2.2)$$

avec E l'intensité de la lumière incidente ou illumination. Le niveau de gris est une mesure de l'irradiance de l'image. L'équation (2.1) donne la relation entre le niveau de gris et la radiance réfléchiée. Elle explicite également la relation entre le niveau de gris et

l'angle α . Ainsi, une image doit être calibrée, dans le sens où il faut la corriger pour la rendre indépendante de l'angle α .

L'équation (2.2) indique que la radiance (donc, le niveau de gris) est une fonction des propriétés réfléchives de la surface. La détermination de ces propriétés nécessite la connaissance des caractéristiques optiques de la surface et de la géométrie du système. Elle peut être effectuée analytiquement pour certaines surfaces (surface miroir, lambertienne). En générale, elle est déterminée empiriquement.

La formation de ce type d'images nécessite un dispositif physique formateur d'images et s'effectue en trois étapes (Fig. 2.4). La première est la propagation du rayonnement et la formation de l'image résultat d'un processus de répartition d'énergie. La seconde est la détection physique de l'image qui produit une distribution analogique de la tension électrique. La dernière étape est la conversion analogique/numérique. Cette dernière étape n'est nécessaire que dans le cas de traitement d'images numériques. Le lecteur intéressé trouvera un complément d'informations dans [Bal 82, Duv 88].

L'image résultat d'un système d'acquisition optique est une projection de la scène 3D sur un plan 2D. A un point de l'image peut correspondre plusieurs points de la scène. L'image est alors une représentation ambiguë de la scène. De plus, elle est souvent caractérisée négativement par des dégradations dues à des phénomènes statistiques, échantillonnage, propriétés des surfaces des objets présents dans la scène, etc. L'image à niveaux de gris présentée dans la figure 1.1 est un exemple de résultat d'un système d'acquisition optique.

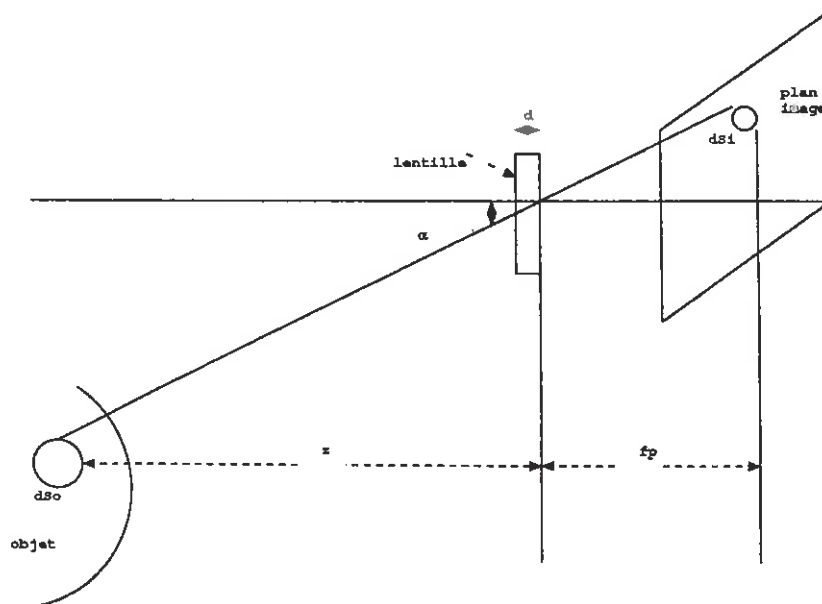


Figure 2.3. La géométrie de la formation d'image optique [She 86].

2.4 Que peut-on extraire d'une image ?

L'image est une structure contenant une quantité énorme d'informations caractérisant la scène observée. Leur nombre et leur nature diffèrent d'une image à une autre suivant le procédé de formation. A ce sujet, reprenons l'exemple de l'image optique et de l'image

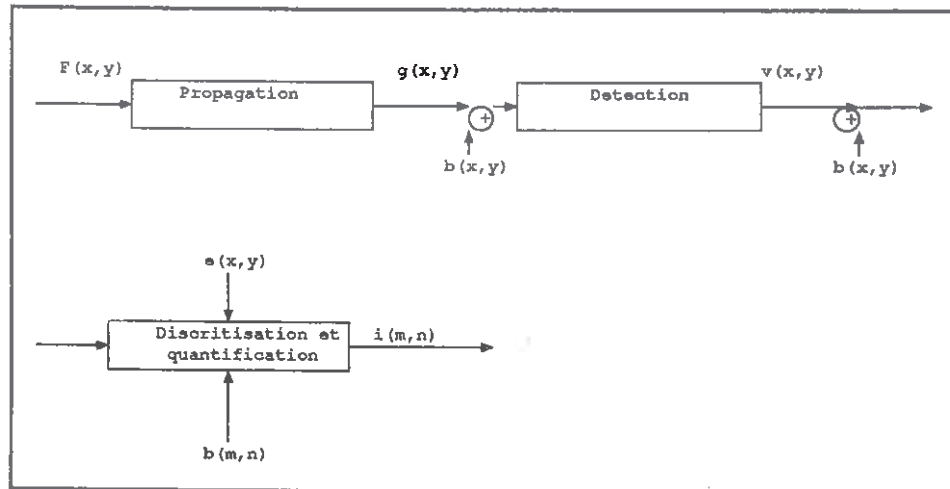


Figure 2.4. Un exemple d'un simple système de formation de l'image. $F(x,y)$: la distribution objet, $g(x,y)$: la distribution image, $b(x,y)$: le bruit, $v(x,y)$: l'image détectée, $e(x,y)$: la fenêtre d'échantillonnage spatial et $i(n,m)$: image numérisée et discrétisée spatialement.

rayon X.

- **Image optique** : nous avons montré dans le paragraphe précédent que le niveau de gris est une fonction de la réflectance de la surface, de l'illumination et de certains facteurs géométriques du système de formation. L'image contient alors des informations relatives aux surfaces des objets et leur géométrie (leur orientation, leur discontinuité, leur continuité, leur micro-structure, etc) et des informations relatives à l'illumination (par exemple l'ombre). A partir de telles informations il est possible d'en déduire d'autres, par exemple la géométrie de l'objet : concavité, convexité, etc. [Hor 77].
- **Image rayon X** : le procédé de formation des images rayon X nous conduit à déduire que l'image obtenue contient seulement des informations relatives à la structure de la matière de l'objet : sa composition, sa densité et son homogénéité. Une image rayon X conventionnelle ne contient pas d'informations relatives aux propriétés géométriques des surfaces des objets qui composent la scène. Ceci explique le fait qu'un tel type d'image est utilisé dans les applications d'analyse de la matière qui compose un objet (applications médicales, contrôle de qualité, analyse des objets d'art, etc.).

Les informations qu'on peut extraire d'une image sont multiples et difficiles à énumérer. De ce fait, on cherche à extraire les informations dont on a besoin. Ainsi, l'objectif et l'approche utilisés limitent le nombre et la nature des informations nécessaires, donc le choix du système de formation.

Dans le cas des deux procédés de formation d'images exposés ci-dessus, les variations dans la fonction image est une information significative. En effet, considérons par exemple le système de formation d'image optique. D'après l'équation (2.1), le niveau de gris peut

être considéré comme fonction des trois paramètres suivants : I indiquant l'illumination, G signifiant le facteur géométrique et R la réflectance. Un point de l'image est représenté par le triplet (I, G, R) tel que I, G, R prennent la valeur 1 pour une discontinuité et la valeur 0 pour une continuité. Ainsi, $(0,0,0)$ veut dire qu'il n'y a pas de variation, $(0,0,1)$ correspond à une variation de la réflectance (e.g., changement dans la micro-structure de la surface), $(0,1,0)$ correspond à une variation géométrique (e.g., discontinuité dans l'orientation de la surface, ou limite dans la surface visible) et $(1,0,0)$ correspond à une variation dans l'illumination (par exemple dans l'ombre). On peut avoir une combinaison de plusieurs variations différentes. Par exemple le triplet $(1,0,1)$ représente à la fois une variation d'illumination et de réflectance.

D'une manière générale, une variation dans le niveau de gris est une conséquence de un ou plusieurs phénomènes physiques et possède ainsi une signification dans la scène représentée par l'image.

La variation dans l'intensité image est l'information souvent utilisée dans les systèmes de vision. C'est pourquoi la recherche d'une telle information est l'un des plus anciens axes de recherche dans le domaine de la vision par ordinateur et du traitement d'image. Les approches proposées par les chercheurs reposent essentiellement sur la forme de cette variation : c'est-à-dire la façon dont la fonction niveau de gris varie localement. Ces variations sont appelées *contours*. La recherche de ces variations est appelée *détection de contours*.

En conclusion, le contour est une information capitale pour la description d'une scène. Elle est disponible souvent dans les images. Cependant, elle peut avoir des significations différentes dans la scène selon le procédé de formation utilisé. Cette notion de contours constitue le sujet principal de cette thèse. C'est pourquoi nous l'étudions de manière plus détaillée au chapitre suivant.

3

La détection de contours

Ce chapitre présente un aperçu synthétique de la détection de contours. Nous proposons une définition précise de la notion de contours. Nous exposons la difficulté principale de la détection de contours à savoir le problème de différentiation d'une image numérique. Nous décrivons ensuite les propriétés mathématiques d'un détecteur, la relation entre ses propriétés et ses performances et l'ordre dans lequel doivent s'exécuter ses deux opérations : lissage et différentiation.

3.1 Notion de contours

Le contour est la représentation dans l'image des frontières des objets et des variations dans les propriétés géométriques et physiques de leurs surfaces (e.g., orientation, matière, texture, couleur, etc.). Traditionnellement, la notion de contour englobe souvent les points de discontinuité de la fonction image. Cependant, la discontinuité n'est pas la seule variation du niveau de gris importante pour un système de vision. Horn [Hor 77] a procédé à une classification des variations dans le cadre des objets polyédriques, suivant le phénomène physique qui les engendre. Nous donnons ci-dessous les variations de contours les plus importants pour un système de vision.

- les marches : elles existent généralement entre deux régions de niveaux de gris constants et différents. Ce type de contours peut être dû au recouvrement ou l'obscurcissement d'un objet par un autre. Un point de contours de type marche correspond au point de discontinuité dans la fonction niveau de gris. Son profil est présenté dans la figure 3.1.a. Dans la pratique, le contour est dégradé par la présence du bruit et correspond à des variations brusques représentées par les points d'inflexions de la fonction image (fig. 3.1.b). Ainsi, il peut être mis en évidence soit par la recherche du maximum de la première dérivée (Fig. 3.1.c) soit par la recherche du point de passage par zéro de la seconde dérivée (Fig. 3.1.d). En 2D, la dérivée seconde peut être approchée par le laplacien.

La définition du contour de type marche ci-dessus est locale dans la mesure où elle n'inclut pas sa distribution spatiale. C'est la raison pour laquelle, il est considéré parfois comme étant une combinaison linéaire de n points d'inflexions. Le cas le plus fréquent étant $n = 2$, on dit alors qu'il s'agit d'une double marche. Il existe deux types de double marche, les crêtes et les escaliers (figs. 3.1.e, 3.1.f).

- les toits : ils existent à la rencontre de deux surfaces d'orientation différente résultant de l'illumination mutuelle entre ces deux surfaces où d'un objet mince posé sur un

fond. Le profil d'un toit idéal est présenté dans la figure 3.2.a. Une variation de ce type correspond au maximum local (ou bien au minimum local) de la fonction niveau de gris. Elle peut être mise en évidence alors par la recherche du passage par zéro de la dérivée première. Cette façon de procéder pose cependant certains problèmes que nous traitons dans la deuxième partie de cette thèse.

D'autres variations moins fréquentes dans une image sont définies dans l'article de J. Ponce et M. Brady [Pon 85]. Notons que pour des raisons pratiques, les points de jonctions sont considérés à part (fig. 3.2.b). On parle alors de contours de type jonction ou tout simplement de coins. D'une manière générale, nous appelons contour toute variation (ou combinaison linéaire de variations) dans le niveau de gris, conséquence d'un phénomène physique possédant une signification dans la scène.

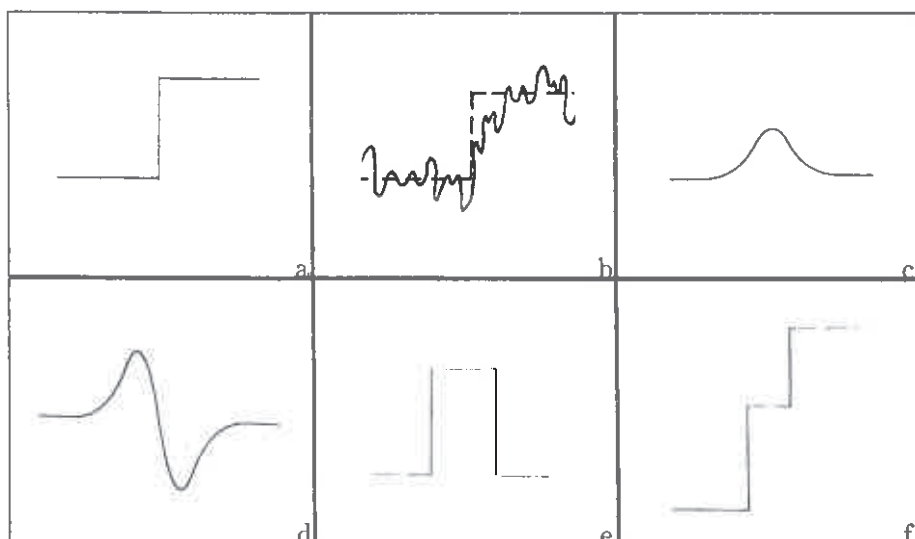


Figure 3.1. Profil de contour de type marche. a) marche idéale, b) marche bruitée, c) et d) dérivée première et dérivée seconde de la marche idéale lissée, e) double marche idéale de type crête. f) double marche idéale de type escalier.

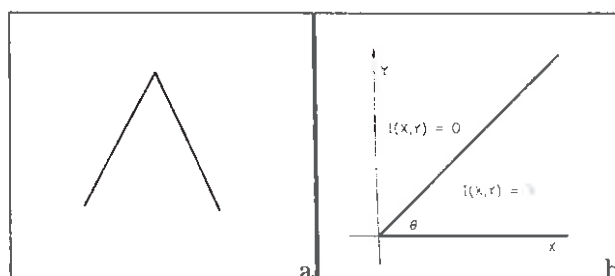


Figure 3.2. Profil des autres types de contours. a) contour de type toit, b) exemple de contours de type coins.

3.2 La détection de contours est un problème mal posé

La détection de contours implique la recherche des variations locales et significatives dans la fonction image comme nous l'avons signalé auparavant. Ces variations sont généralement mises en évidence à partir des dérivées de cette fonction. Le type et l'ordre de ces dérivées sont déterminés en général par le type du contour et par son utilisation ultérieure.

La différentiation numérique de l'image est une tâche délicate dans le sens où elle se détériore rapidement en présence de bruit. Il en résulte que la différentiation d'une image discrète et bruitée est un problème mal posé¹ [Pog 84, Tor 86].

Pour rendre la différentiation robuste, la technique dite de régularisation est utilisée. Il s'agit d'imposer de nouveaux critères au problème de différentiation pour restreindre l'espace des solutions possibles (c'est-à-dire, forcer la solution à être dans le sous-espace des solutions bien définies) et choisir celle qui minimise une fonctionnelle adéquate. Tikhonov et Arsenin [Tik 77] montrent que la régularisation numérique du problème de différentiation peut être effectuée en lissant l'image par un filtre $f(x)$ ayant les propriétés suivantes : soit $F(w, a)$ la transformée de Fourier de $f(x)$, où $a \geq 0$ est un paramètre,

1. $F(w, a)$ est une fonction définie pour toute valeur de w ,
2. $F(w, a)$ est une fonction paire par rapport à w et possédant une énergie finie dans l'intervalle $] -\infty, +\infty[$,
3. $jw F(w, a)$ possède aussi une énergie finie dans l'intervalle $] -\infty, +\infty[$, où $j = \sqrt{-1}$
4. $\lim_{|w| \rightarrow \infty} F(w, a) \rightarrow 0, \quad \forall a > 0,$
5. $\lim_{a \rightarrow 0} F(w, a) \rightarrow 1$ et $F(w, 0) = 1$

Poggio, Vorhees et Yuille [Pog 84, Pog 85] montrent analytiquement que le filtre $f(x)$ est une spline cubique ayant une forme très proche d'une gaussienne. Sa largeur est contrôlée par un paramètre de régularisation assurant le compromis entre le degré de régularisation de la solution et sa conformité aux données initiales. La différentiation est alors accomplie par la convolution des données avec la dérivée appropriée de ce filtre. Il est important de mentionner que le filtre de régularisation proposé dans [Pog 85] possède la réponse impulsionnelle suivante :

$$R(x, \mu) = \frac{\mu}{2\sqrt{2}} e^{-\mu|x|} \left(\cos\left(\frac{\mu x}{\sqrt{2}}\right) + \sin\left(\frac{\mu x}{\sqrt{2}}\right) \right) \quad (3.1)$$

où μ est le paramètre de régularisation. La gaussienne est aussi une bonne approximation de la spline cubique. Les détecteurs proposés par Canny [Can 83], Marr et Hildreth [Mar 80] ainsi que celui de Deriche [Der 87] permettent alors une régularisation de la différentiation. Ces détecteurs sont détaillés dans le chapitre 4 de cette partie.

1. Un problème mathématique est dit bien posé si la solution existe, est unique, et dépend de manière continue des données initiales. Intuitivement, un problème est bien posé si sa solution est robuste.

3.3 Organisation d'un détecteur de contours

Un détecteur de contours a pour donnée une image discrète et fournit en résultat une description de cette image en terme de contours (voir fig. 1.4). Le contour inclut explicitement des informations concernant sa position, son orientation, l'échelle à laquelle il a été détecté et une valeur pour caractériser sa force (le gradient, la courbure, la pente, etc.). L'exemple présenté dans la figure 1.4 inclut uniquement des informations de position.

Les détecteurs existants diffèrent essentiellement par la connaissance utilisée pour mettre en évidence le contour. On distingue deux grandes classes de détecteurs :

- **les détecteurs autonomes** : un détecteur de cette classe est considéré comme une boîte noire dans le sens où il n'est influencé ni par les résultats d'autres modules ni par les informations contextuelles. L'avantage de ce type de détecteur est la généralité : il s'exécute avec n'importe quelle image. Cependant, ces détecteurs possèdent une vision locale ; la décision de l'étiquetage d'un point comme contour est fondée sur des critères locaux.
- **les détecteurs contextuels** : un détecteur contextuel est considéré comme un module guidé par les résultats des autres modules et/ou par un modèle. Cette seconde classe de détecteurs a l'avantage d'utiliser des connaissances *a priori* sur le contour contenu dans l'image et/ou la structure de l'objet de la scène. Cependant, le détecteur ne s'exécute que dans un contexte précis.

En se fondant sur les caractéristiques de chacune de ces deux classes, on peut affirmer que l'utilisation des détecteurs autonomes se justifie dans le cas d'un système général. Or, l'utilisation des détecteurs contextuels est plus adaptée dans le cas d'une application dont l'image traitée contient les mêmes objets.

Du point de vue architecture, un détecteur autonome comme un détecteur contextuel inclut au moins trois opérations : lissage, différentiation et correction (fig., 3.3). La correction est la dernière opération à effectuer. Cependant, l'ordre dans lequel se déroulent le lissage et la différentiation reste à définir (cf. 3.3.3). Le rôle de ces opérations est le suivant :

- **le lissage** : il a pour objectif la réduction de la quantité de bruit contenu dans l'image. Cette opération est souvent effectuée par une simple convolution de l'image et d'un filtre de type passe-bas. Le filtre peut être représenté par une fonction $f(x, a)$ appelée la réponse impulsionnelle du filtre (a étant l'échelle du filtre ou paramètre de régularisation). Par exemple, l'échelle d'un filtre dont la réponse impulsionnelle est une gaussienne est l'écart-type. Cette opération de lissage a pour effet indésirable, de modifier la structure de l'image : élimination et déplacement de certains contours et création de nouveaux contours. Bien que les filtres assurant un compromis entre l'élimination du bruit et la préservation de contours sont ceux qui ont été proposés pour régulariser la différentiation (cf. 3.2), il se pose le problème du choix de leurs échelles.
- **la différentiation** : elle est l'opération de calcul des dérivées nécessaires pour mettre en évidence les contours. Les opérateurs de différentiation utilisés diffèrent par leur ordre, la variance ou l'invariance par rotation, etc. (cf. 3.3.2). C'est le type du contour et celui du filtre qui déterminent le type et l'ordre de l'opérateur à utiliser.

- la correction : elle a pour objectif l'augmentation du rapport signal/bruit du détecteur par une suppression des faux points de contours et par la fermeture des trous dans les contours résultats des opérations ci-dessus.

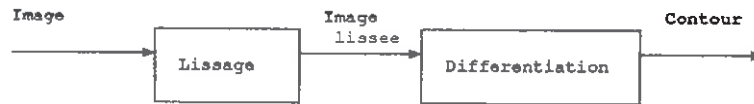


Figure 3.3. Architecture d'un détecteur autonome de contours.

Il est évident que le rôle de l'opération de correction dépend des performances des deux opérations précédentes. En effet, si le lissage élimine le bruit en préservant les contours et la différentiation fournit ces contours alors le problème de correction est résolu. C'est la raison pour laquelle nous nous intéressons particulièrement aux deux premières étapes. A ce sujet signalons que la détection de contours en 1D est effectuée par une simple opération de filtrage (ce n'est pas toujours vraie en 2D, voir cf. 3.3.3). C'est pourquoi nous utiliserons parfois le terme filtre à la place du terme détecteur.

Les propriétés d'un détecteur de contours sont alors celles qui sont relatives à la fois au filtre de lissage et à l'opérateur de différentiation. Il nous paraît important de présenter de telles propriétés et leur influence sur le contour.

3.3.1 Lissage de l'image

La plupart des filtres de lissage utilisés en détection de contours sont des passe-bas linéaires. Les propriétés de ces filtres sont complètement définies par la durée de ses deux réponses impulsionnelle et fréquentielle. Examinons d'une façon informelle les performances des trois filtres passe-bas les plus utilisés en détection de contours.

- un filtre à bande limitée vérifie les conditions de Tikhonov. Il permet alors la régularisation de la différentiation. Ce filtre produit des contours ayant une faible erreur de localisation (c'est-à-dire que l'erreur dans la position des contours localisés est faible) et une mauvaise détection (c'est-à-dire qu'il ne produit pas tous les contours et les contours trouvés sont bruités). Un exemple de ce filtre est celui utilisé par Shanmugam [Sha 79].
- un filtre à support limité favorise la détection par rapport à la localisation. Ce type de filtre ne permet pas la régularisation de la différentiation. Son utilisation dans le cas des images bruitées est alors non souhaitable. Le détecteur de Herskovitz et Binford [Her 70] connu sous la dénomination "*difference of boxes*" en est un exemple.
- un filtre à incertitude minimale² permet la régularisation de la différentiation. Il est l'unique filtre qui assure un bon compromis entre la détection et la localisation et qui est stable dans un espace échelle. La gaussienne en est un exemple.

Une autre propriété du filtre de lissage qui influence directement ses performances est son variance ou invariance par rotation. Son invariance par rotation assure que les

2. traduction de *minimum uncertainty*

différents effets indésirables du lissage (cf. 3.3) ne sont pas accentués dans une direction particulière.

Une fonction $f(x, y)$ est dite invariante par rotation si et seulement si $f(x, y) = f(X, Y)$, avec $X = x\cos(\theta) + y\sin(\theta)$ et $Y = -x\sin(\theta) + y\cos(\theta)$. On peut vérifier facilement que la gaussienne est invariante par rotation et que le filtre défini par l'équation 3.1 ne l'est pas.

3.3.2 Différentiation de l'image

Définition

Un opérateur de différentiation est souvent caractérisé par :

- l'ordre : il est défini par l'ordre des dérivées partielles qui sont utilisées et la manière dont elles sont combinées,
- l'invariance ou la variance par rotation : la définition est similaire à celle donnée dans la section précédente,
- directionnel ou non directionnel : un opérateur de différentiation est dit directionnel s'il inclut dans sa définition l'orientation du contour, sinon il est non directionnel,
- linéaire ou non linéaire : un opérateur \mathcal{O} est dit linéaire si et seulement si pour tout scalaire positif α et β et pour toute fonction $f(x)$ et $g(x)$, on a $\mathcal{O}(\alpha f(x) + \beta g(x)) = \alpha \mathcal{O}(f(x)) + \beta \mathcal{O}(g(x))$.

Examinons maintenant les opérateurs les plus utilisés en détection de contours :

- opérateur du premier ordre : il est défini à partir des dérivées partielles $\partial/\partial x$ et $\partial/\partial y$. Les opérateurs les plus utilisés en détection de contours sont :

$$\sqrt{\left(\frac{\partial}{\partial x}\right)^2 + \left(\frac{\partial}{\partial y}\right)^2} \quad \text{et} \quad \left|\frac{\partial}{\partial x}\right| + \left|\frac{\partial}{\partial y}\right|$$

Ces deux opérateurs sont généralement appelés gradient. Ils sont tous les deux non directionnels et non linéaires. Cependant seul le premier est invariant par rotation.

Signalons que l'identification du contour nécessite le calcul des dérivées de l'image dans les deux directions x et y , puis le calcul du module et de l'orientation. Son module est calculé par l'un des deux opérateurs ci-dessus. Son orientation est normale à la direction du gradient. Cette dernière est calculée à partir des dérivées partielles du premier ordre par :

$$\theta = \arctg\left(\frac{\partial}{\partial y} / \frac{\partial}{\partial x}\right)$$

Dans le cas d'une image bruitée, l'utilisation d'un opérateur défini à partir des dérivées calculées dans plusieurs directions permet de fournir des informations contours plus fiables.

- opérateur du second ordre : il est défini à partir des dérivées partielles $\partial^2/\partial x^2$, $\partial^2/\partial x\partial y$ et $\partial^2/\partial y^2$. Les opérateurs les plus utilisés en détection de contours sont le laplacien et la seconde dérivée directionnelle prise dans la direction du gradient θ . Ces deux opérateurs sont définis respectivement par :

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad \text{et} \quad \frac{\partial^2}{\partial \theta^2} = \frac{\partial^2}{\partial x^2} \cos^2(\theta) + \frac{\partial^2}{\partial x\partial y} \sin(\theta)\cos(\theta) + \frac{\partial^2}{\partial y^2} \sin^2(\theta)$$

Ces deux opérateurs sont invariants par rotation. Cependant, seul le laplacien est linéaire.

Influence de l'opérateur de différentiation sur les contours

S'il est évident qu'un opérateur du premier et du second ordre fournit théoriquement en 1D le même contour, il n'en est pas de même en 2D. Un certain nombre de questions se posent : sous quelles conditions existe-il une équivalence entre ces opérateurs ? Quelles sont les propriétés des contours détectés par chacun de ces opérateurs ?

Dans le cas de contours de type marche, les réponses à ces deux questions sont données par [Tor 86] :

- la différence entre le laplacien et la deuxième dérivée prise dans la direction du gradient est proportionnelle à la courbure moyenne de la fonction image. Les deux opérateurs sont équivalents, c'est-à-dire que leurs zéros coïncident, si et seulement si cette courbure est nulle et le gradient est différent de zéro, ou si la fonction image $I(x, y)$ ne dépend que d'une seule variable (e.g., $I(x, y_0)$, où y_0 est une constante).
- l'utilisation du laplacien assure l'obtention de contours fermés. Cependant les contours résultants de l'utilisation de la seconde dérivée prise dans la direction du gradient ne possèdent pas des propriétés géométriques spécifiques ; par exemple, ils ne sont pas fermés. En général, ce type d'opérateur permet une bonne précision dans la localisation des contours détectés.

Une étude des performances des opérateurs de différentiation dans la détection des jonctions a été effectuée récemment par E. De Micheli et al [Mic 89]. Nous présentons ci-dessous le résumé de leurs résultats :

- les jonctions de type T ou Y (trois lignes de contours incidentes) sont mieux localisées par un opérateur gradient. Un opérateur de second ordre introduit des faux contours dans leurs voisinages (fi. 3.4). D'une façon générale, l'opérateur gradient est plus robuste qu'un opérateur de second ordre.
- les jonctions de type L ou V (angle entre deux lignes de contours) ne sont localisées correctement ni par un opérateur gradient, ni par un opérateur laplacien. Cependant, les points de jonctions sont mieux localisés par un opérateur laplacien comme le montre la figure 3.4.

Nous avons examiné jusqu'à maintenant les propriétés de chacune des deux opérations formant le détecteur de contours et leur influence sur le contour. Nous discutons dans le paragraphe suivant l'influence de ces propriétés sur l'ordre dans lequel ces opérations doivent s'exécuter.

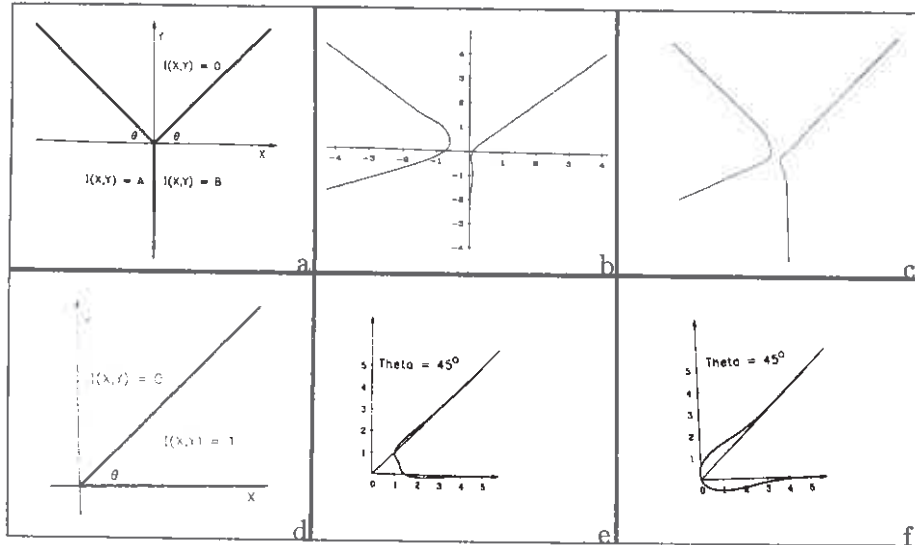


Figure 3.4. a) une jonction de type Y (extrait de [Mic 89]), b) contours obtenus aux maxima du gradient, c) contours obtenus aux passages par zéro du laplacien, d) un coin de type V (extrait de [Mic 89]), e) contours obtenus aux maxima du gradient, f) contours obtenus aux passages par zéro du laplacien.

3.3.3 Commutativité des opérations de lissage et de différentiation

Rappelons qu'un détecteur de contours est formé de deux opérations : lissage et différentiation (Cf. 3.3). La question fondamentale que se pose est la suivante : faut-il lisser l'image avant ou après la différentiation ?

L'ordre de déroulement de ces deux opérations dépend essentiellement de la linéarité de l'opérateur de différentiation [Yui 86]. Cet ordre est défini comme suit :

- opérateur linéaire : il est commutatif et associatif avec l'opération de convolution. Formellement, soient $I(x, y)$ et $f(x, y)$ deux fonctions et \mathcal{O} un opérateur non linéaire :

$$\mathcal{O}(I(x, y) * f(x, y)) = (\mathcal{O}I(x, y)) * f(x, y) = I(x, y) * (\mathcal{O}f(x, y))$$

Par conséquent, l'ordre dans lequel s'exécute les étapes du lissage et de différentiation n'est pas important.

- opérateur non linéaire : en général, ce type d'opérateur n'est, ni commutatif ni associatif avec la convolution. La considération de la régularisation de la différentiation (Cf. 3.2) nous conduit à ordonner les deux étapes de la manière suivante : lissage d'abord, puis différentiation.

Dans le cas de la détection multi-échelles de contours (c'est-à-dire, un détecteur exécuté plusieurs fois avec des paramètres différents, pour plus de détails voir le paragraphe Cf. 4.4) et si le détecteur est formé d'un lissage gaussien et de la dérivée seconde prise dans la direction du gradient comme opérateur de différentiation,

alors l'ordre d'exécution des deux opérations est différent de celui présenté ci-dessus. Ce changement d'ordre a pour objectif de préserver la propriété de stabilité de la gaussienne dans un espace échelle.

Plus précisément, dans le cas général, l'utilisation de la dérivée seconde directionnelle ne fournit pas un détecteur stable dans un espace échelle. Ceci n'est pas nécessairement vrai dans le cas où le filtrage est effectué après la différentiation. L'ordre peut être défini alors par une différentiation suivie d'une succession de lissages à différentes échelles. Cependant, la considération de la théorie de régularisation implique une étape de lissage avant toute différentiation (cf. 3.2). L'ordre final ne peut être que le lissage pour régulariser, la différentiation suivie d'une succession de lissages à différentes échelles.

3.4 Conclusion

Nous considérons comme contour tout point (ou une suite de points adjacents) de l'image où une variation de niveau de gris apparaît. Les points de jonction dans une image ne sont pas généralement localisés par les détecteurs de contours classiques. C'est la raison pour laquelle ces points sont considérés comme étant un contour à part pour lequel il existe des détecteurs spécialisés.

La difficulté principale de la détection de contours provient du fait que la différentiation d'une image numérique est un problème mal posé. Par conséquent, un détecteur de contour est composé au moins d'une opération de lissage et d'une opération de différentiation. Nous avons présenté dans ce chapitre les propriétés mathématiques de chacune de ses opérations, leur influence sur le contour résultat et l'ordre dans lequel elles s'exécutent. Remarquons que, si l'influence des propriétés d'un détecteur sur ses performances n'est pas en général définie d'une manière précise, l'influence des caractéristiques d'un contour l'est moins. Par conséquent, le choix d'un détecteur, étant donnée une image ou un ensemble d'images, est délicat. Ce problème auquel nous nous sommes confronté souvent reste ouvert.

Après cette étude sur la détection de contours, nous présentons dans le chapitre suivant quelques exemples de détecteurs existants.

4

Les travaux antérieurs

Ce chapitre inclut une description des principaux détecteurs de contours existants. Nous exposons également la détection multi-échelle de contours et la problématique d'élimination des faux points de contours, dans lesquels nous présentons brièvement notre contribution.

4.1 Introduction

Depuis l'apparition de la vision par ordinateur, le nombre d'algorithmes de détection de contours ne cesse d'augmenter et il est difficile de nos jours de les recenser. La plupart des détecteurs de contours sont autonomes et se composent principalement d'une étape de lissage et d'une étape de différentiation. Ils diffèrent essentiellement par :

- les propriétés de l'opérateur de lissage (filtre à bande limitée, filtre à support limité, filtre avec un minimum d'incertitude),
- les propriétés mathématiques de l'opérateur de différentiation (linéaire, non linéaire, directionnel, invariant par rotation),
- le type du contour recherché (marche, toit, coin),
- le fondement mathématique utilisé pour dériver le détecteur (informel, optimal).

Les détecteurs contextuels sont rares et sont tous destinés au contour de types marches. Ils diffèrent essentiellement par les connaissances utilisées pour mettre en évidence le contour objectif.

Peu d'études de synthèse ont été effectuées dans le domaine du contour, le lecteur peut cependant se référer à [Dav 75, Tor 86, Sta 87, Ser 89]. Notre objectif n'est pas la présentation d'une étude complète de l'existant. Nous nous limiterons donc aux travaux qui nous ont influencés.

4.2 Détecteurs autonomes de contours

4.2.1 Détecteurs informels de marches

Dès les années 60, on vit apparaître des détecteurs localisant le contour par estimation du gradient ou du laplacien de la fonction image. Ces détecteurs correspondent à des opé-

rateurs bidimensionnels de différentiation discrète. Par exemple, l'estimation des dérivées partielles du premier ordre de la fonction $I(x, y)$ est effectuée de la façon suivante :

$$\frac{\partial I(x, y)}{\partial x} = I(x, y) - I(x - 1, y)$$

$$\frac{\partial I(x, y)}{\partial y} = I(x, y) - I(x, y - 1)$$

L'opérateur de différentiation correspondant est représenté par les deux masques suivants : et $\Delta_x = [1 \quad -1]$ et $\Delta_y = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

L'estimation du vecteur gradient au point (x, y) est effectuée par la convolution discrète de ces deux masques et la fonction $I(x, y)$. Remarquons que les opérateurs Δ_x et Δ_y ne sont pas symétriques par rapport au point (x, y) . Cependant, ils peuvent l'être par l'utilisation de masques de taille impaire, par exemple 3, 5, 7, etc.

L'extension en 2D de tels opérateurs est immédiate. Un exemple d'opérateur pour l'estimation du vecteur gradient est celui de Prewitt [Pre 70] défini par :

$$\Delta_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{et} \quad \Delta_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

L'estimation du laplacien peut être effectuée par l'opérateur suivant [Bal 82] :

$$\Delta^2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

En plus du problème crucial commun à tous les détecteurs de contours, à savoir le choix de la taille du masque approprié, ces détecteurs se limitent à un opérateur de différentiation. Leurs résultats se détériorent rapidement en présence de bruit. Rosenfeld et Thurston [Ros 71] ont été parmi les premiers à avoir introduit une étape de lissage dans le processus de différentiation de l'image traitée. Ce processus a pour objectif la réduction du bruit de l'image par moyennage sur une fenêtre carrée. Malgré l'étape de lissage introduite, ces opérateurs demeurent sensibles au bruit.

Par ailleurs, tous ces détecteurs sont heuristiques dans la mesure où ils n'utilisent aucun modèle pour représenter le contour. Leurs performances sont analysées sur des bases informelles. L'introduction de l'étape de lissage pose le problème primordial de choix du filtre à utiliser. D'où la nécessité de recherche d'une méthode de lissage assurant le compromis entre l'élimination du bruit et la conservation des contours.

4.2.2 Détecteurs optimaux de marches

Rappelons que le processus de lissage permettant de rendre la détection de contours moins sensible au bruit possède des effets indésirables qui peuvent être résumés par une perte d'informations (cf. 3.3). Plusieurs travaux ont été consacrés au problème de la

recherche d'un filtre minimisant cette perte et différents critères ont été proposés. On distingue deux orientations différentes. Dans la première on cherche à ajuster au mieux l'image par une base de fonctions données. La seconde orientation consiste à déterminer le détecteur optimisant un critère donné.

Approches fondées sur l'ajustement

Ces approches consistent souvent à ajuster au mieux l'image par une base de fonctions données, puis mettre en évidence le contour. L'intérêt principal de l'ajustement est de résoudre analytiquement le problème de différentiation. Nous présentons dans ce qui suit deux exemples de détecteurs utilisant cette technique.

Approche d'ajustement de contours – Algorithme d'Hueckel [Hue 71, Hue 73]

A chaque point de l'image est associé un voisinage de forme circulaire appelé région. Chacune de ces régions est comparée à un modèle de contour idéal proposé par Hueckel. La décision de l'acceptation d'un contour est fondée sur la valeur de l'erreur quadratique utilisée pour mesurer la différence entre le contour idéal et le contour actuel. Si cette valeur est inférieure à un seuil donné, la région en question est considérée comme étant un contour. Pour résoudre analytiquement le problème d'ajustement, Hueckel propose d'approcher à la fois le modèle contour et chaque région par un ensemble de huit fonctions orthogonales à support circulaire. Il argumente l'utilisation seulement de huit premières fonctions parmi l'ensemble infini de fonctions de base qu'il a défini, par le fait que les composantes à haute fréquence contiennent plus de bruit que de signal. Il en résulte que les performances du détecteur ne sont pas affectées. Aucune démonstration à ce sujet n'a été présentée, plus tard Abdou [Abd 78] a mis en évidence les effets indésirables de cette limitation sur les performances du détecteur d'Hueckel.

Un autre problème de cette approche est le fait que l'image est discrète, alors que les fonctions de base sont continues. Ainsi, des dégradations sont introduites lors du passage du discret au continu. Abdou [Abd 78] propose une approche similaire à celle de Hueckel fondée sur l'utilisation de fonctions de base discrètes.

Approche d'ajustement de l'image – Algorithme d'Haralick [Har 84]

Les travaux d'Haralick sont un autre exemple de l'approche de détection de contours fondée sur l'ajustement. Il propose dans son article [Har 84] un algorithme de détection de contours dans lequel un point est marqué comme un contour s'il existe un passage par zéro de la seconde dérivée directionnelle prise dans la direction du gradient. Pour calculer les dérivées analytiquement, il approche l'image discrète par une combinaison linéaire de polynômes discrets de Tchebychev d'ordre inférieur ou égal à trois. L'étape de filtrage n'a pas été mentionnée dans son article. Dans sa thèse, Canny [Can 83] a montré que cette approximation est équivalente à l'utilisation d'une étape de lissage suivie d'une étape de différentiation. De plus, la dérivée première de la combinaison linéaire de polynômes de Tchebychev d'ordre inférieur ou égal à trois est une bonne approximation de son filtre optimal décrit plus loin.

Il existe d'autres algorithmes de détection de contours fondés sur l'ajustement de l'image et du contour [Nev 77, Hum 79]. Les performances de ce type d'algorithme sont déterminées entièrement par le choix des fonctions de base. Aucune analyse de l'optimalité des fonctions utilisées pour approcher l'image, ni la justification de leur choix ne sont données.

Approches fondées sur la détection optimale du contour

Il s'agit dans cette approche de déterminer le détecteur de contours permettant de répondre aux exigences des applications actuelles. L'idée principale est de définir des critères de performances d'un détecteur puis de mettre au point le détecteur optimisant ces critères. Ce type d'approche semble le plus adéquat et connaît un succès croissant.

Les travaux de K.S Shanmugam, F.M. Dickey, et J.A Green [Sha 79]

Selon Shanmugam et al. un détecteur de contours doit fournir une énergie maximale à l'intérieur d'un intervalle fini. En utilisant ce critère, ils dérivent un filtre linéaire à bande limitée qui est une approximation du laplacien d'une gaussienne. Le détecteur résultant est caractérisé négativement par une mauvaise localisation. De plus, il répond à un seul point de contours par deux maxima (c'est-à-dire, deux points de contours) comme le montre la figure 4.1. Ceci est dû au fait qu'ils utilisent un filtre symétrique pour localiser des contours aux maxima locaux de l'image résultat. Or, un détecteur de contours aux maxima locaux doit être anti-symétrique.

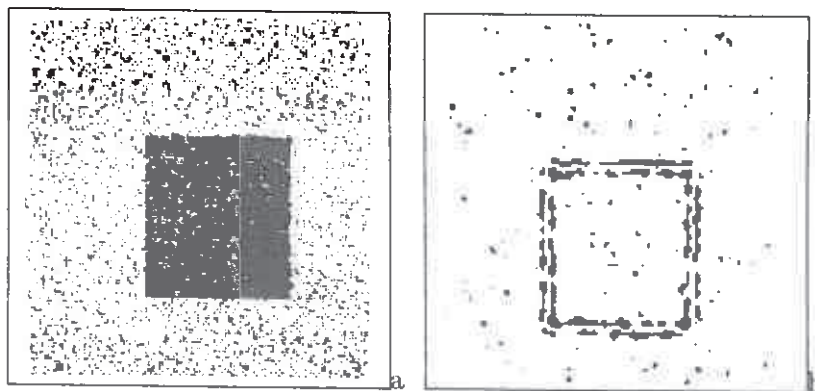


Figure 4.1. a) Image originale. b) Les contours obtenus par le détecteur de Shanmugam et al.

Les travaux de D. Marr et E.C. Hildreth [Mar 80]

Marr et Hildreth [Mar 80] puis Hildreth [Hil 83] proposent le laplacien d'une gaussienne comme détecteur de contours. Le choix d'un lissage gaussien est fondé sur les propriétés mathématiques de la fonction de Gauss : elle est l'unique filtre permettant d'effectuer un compromis optimal entre le rapport signal/bruit et l'erreur de localisation. Le choix de l'opérateur laplacien est motivé par l'efficacité. En effet un opérateur directionnel nécessite le calcul de la dérivée dans plusieurs directions. Ce détecteur est l'un des plus connus, cependant l'erreur de la localisation des contours résultats est relativement élevée.

Les travaux de Canny [Can 83, Can 86]

L'approche adoptée par Canny est similaire à celle de Shanmugam et al. [Sha 79]. En effet, Canny a formulé les propriétés souhaitables d'un détecteur de contours, fondées sur l'exhaustivité de la détection et l'exactitude dans la localisation. En utilisant le calcul variationnel, il a montré que le filtre 1D, répondant aux exigences formulées est une combinaison linéaire de quatre exponentielles. Pour faciliter son implantation, Canny a utilisé la première dérivée d'une gaussienne, qui est une bonne approximation (elle possède un graphe similaire) de son filtre. Le filtre est anti-symétrique, ce qui permet de détecter les

contours au maximum de la sortie. L'extension en deux dimensions est effectuée par un masque bidimensionnel résultat de la convolution du filtre 1D obtenu dans la direction du contour et d'une fonction de projection (elle est égale à la primitive de son filtre) dans la direction normale au contour.

R. Deriche [Der 87] a repris le filtre initial obtenu par Canny, qui est une combinaison linéaire de quatre exponentielles. En introduisant des contraintes nécessaires à la détection de contours au maximum de la sortie du filtre (filtre anti-symétrique, à support infini, etc.), il obtient un détecteur dont les performances sont supérieures à celles du détecteur implanté par Canny.

Les travaux de Canny et Deriche sont détaillés dans la deuxième partie de cette thèse.

Les travaux de J. Shen et S. Castan [She 85, She 86, Cas 90]

Dans l'optique de recherche des détecteurs optimaux. Shen et Castan [She 85, She 86] ont proposé trois critères pour déterminer un détecteur de contours au maxima de la dérivée première. Un filtre de lissage selon J. Shen et S. Castan doit minimiser l'énergie du bruit et maximiser l'énergie du signal. Puisque ce filtre est destiné à la détection de contour, sa dérivée première doit aussi minimiser l'énergie du bruit. En combinant ces trois critères et en utilisant le calcul variationnel, ils obtiennent un filtre optimal, y compris dans le cas de contours multi-marches (c'est-à-dire, une marche suivie d'une marche inverse qui se répète aléatoirement). Sa réponse impulsionnelle est une exponentielle ce qui lui donne l'avantage d'être séparable et donc extensible facilement en 2D. Il est implanté par des équations aux différences d'ordre un. Ce qui lui permet d'être efficace.

Une comparaison théorique des détecteurs de Canny, de Deriche et de Shen dans le cas d'un signal 1D a été effectuée récemment par S. Castan et al. [Cas 90]. Le résultat de cette étude montre que les performances du détecteur de Shen sont supérieures aux performances du détecteur de Deriche et que ce dernier est plus performant que celui de Canny.

Ces travaux sont également détaillés dans la deuxième partie de cette thèse.

4.2.3 Détecteurs d'autre types de contours

S'il est difficile, pour ne pas dire impossible, de recenser tous les détecteurs de contours de type marche, les détecteurs de contours d'autres types sont rares. Nous n'avons rencontré pendant notre investigation que les détecteurs décrits ci-dessous.

Détecteur de lignes

- Des algorithmes classiques de squelettisation [Arc 85, Chi 87, Smi 87] qui peuvent être adaptés à la recherche des contours de type toit ou de type crête dans des images à niveaux de gris [Dye 79, Jar 87].
- Un détecteur proposé par R.M. Haralick [Har 83] est destiné aux contours de type toit. Il est fondé sur une approximation de l'image par une combinaison linéaire de polynômes de Tchebychev d'ordre inférieur ou égale à trois (de la même façon que dans l'algorithme présenté dans la section 4.2.2). Un point de contours est marqué au passage par zéro de la dérivée première prise dans la direction maximisant la dérivée seconde directionnelle.

- G. Giraudon [Gir 85, Gir 86] propose un algorithme pour la détection des lignes aux extrema locaux du laplacien de l'image. Pour ce faire, il propose l'utilisation de l'opérateur de Keskes qui est équivalent à la convolution de l'image par la différence de deux gaussiennes. La direction de la ligne en un point donné est définie comme étant l'orientation du pixel appartenant à la fenêtre centrée sur ce point et possédant le plus fort gradient.

Détecteur de coins

Dans les travaux de K. Rangarajan, M. Shah et D.V. Brackle [Ran 89] un coin est défini comme étant un point de jonction de deux contours rectilignes. A partir de cette définition et en utilisant les critères de Canny, ils dérivent un détecteur optimal. Le détecteur résultant ne peut localiser qu'une seule configuration de coin (un coin formé par deux droites symétriques par rapport à l'axe des x positifs). Pour mettre en évidence tous les coins contenus dans une image, ils considèrent douze configurations de coins différentes par leur orientation et par l'angle entre les deux lignes de contours. Il en résulte douze masques différents. L'algorithme comporte les étapes suivantes : convolution de l'image avec les douze masques (à un coin dans l'image initiale correspond un maximum local dans l'image résultat de la convolution), ensuite, seuillage de l'image de sortie en vue d'une suppression des non maxima. La figure 4.2 décrit un exemple de contours de type coin obtenus par cet algorithme. Le lecteur intéressé peut se référer à l'article [EB 90].

Notons que la conception des détecteurs de coins est en expansion. Le lecteur intéressé peut consulter les articles [Meh 90, Der 90b].

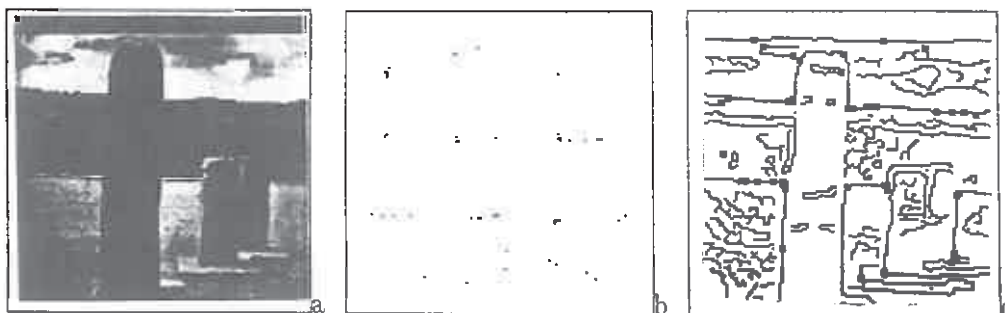


Figure 4.2. a) Image originale, b) image de contours de type coins [Ran 89], c) image de coins superposée sur l'image de contours de type marche.

4.2.4 Détecteur de contours rectilignes

La plupart des détecteurs autonomes que nous avons exposés jusqu'à maintenant sont généraux dans le sens où ils sont destinés à mettre en évidence tous les contours d'un type particulier contenus dans l'image. Ces détecteurs peuvent être adaptés pour mettre en évidence un contour de forme particulière. Nous décrivons ci-dessous deux exemples de détecteurs.

Un détecteur de contours rectilignes de longueur prédéfinie est présenté par A. Mansouri, A.S. Malowany et M.D. Levine [Man 87]. L'algorithme comporte deux étapes :

- étape de prédiction : on cherche à former un ensemble de points susceptible de constituer une droite en répétant les trois étapes suivantes jusqu'à ce que tous les points de l'image soient traités : choix d'un point de l'image, construction de l'équation de la droite dans la direction normale au gradient et passant par ce point, et enfin, formation du segment possible par la considération de tous les points se situant sur cette droite et dont la valeur du module du gradient est supérieure à un seuil donné. Le gradient de l'image est calculé en utilisant l'opérateur de Sobel.
- étape de vérification : à chaque point de la droite formée dans l'étape précédente est associée son orientation. Cette droite est considérée seulement si sa longueur est comparable à celle du contour attendu et la variance de l'orientation est faible.

J.B. Burns, A.R. Hanson et E.M. Riseman [Bur 86] proposent aussi un algorithme d'extraction de contours rectilignes procédant en trois étapes :

- construction de régions support d'une ligne de contours par le calcul du gradient de l'image et le regroupement de pixels adjacents et ayant la même orientation du gradient,
- ajustement des régions de l'image initiale et de l'image gradient en utilisant les modèles facettes [Har 81]. Pour augmenter la contribution des points à gradient fort et diminuer la contribution des points à faible gradient, ils utilisent la technique des moindres carrés pondérée par le gradient.
- détermination de la ligne contour : une ligne contour correspond à l'intersection du plan portant la région de l'image initiale et du plan portant la région de l'image gradient.

4.3 Détecteurs contextuels de contours

Les détecteurs présentés jusqu'ici possèdent les caractéristiques suivantes :

- autonomie : un détecteur est considéré comme une boîte noire dans un système de vision. Le processus d'étiquetage d'un point comme contour s'appuie sur une décision locale et ne tient pas compte des résultats des autres modules du système.
- simplicité : les détecteurs optimaux sont dérivés à partir d'une modélisation mathématique, c'est-à-dire un modèle simplifié et idéal. Le profil du contour dans une image naturelle est souvent déformé par rapport à celui d'un contour idéal. De plus, le bruit n'est pas toujours blanc, ni additif.

Nous présentons ci-dessous deux exemples de détecteurs ayant pour objectif de mettre en évidence un contour donné. Ces deux détecteurs tirent partie de la connaissance *a priori* sur le contour.

4.3.1 Détecteurs guidés par le modèle

L'idée de base est de produire un détecteur de contours guidé par un modèle (des connaissances sémantiques et/ou photométriques) et ayant la capacité de s'adapter à un

contour donné. Le détecteur introduit par Kelly [Kel 81] est un bon exemple. Il a pour objectif de mettre en évidence les contours limitant le visage humain de la façon suivante : détection des contours dans l'image réduite, utilisation des contours résultats comme un plan pour guider la recherche des contours dans l'image initiale. La réduction de l'image initiale a pour objectif la réduction de l'espace de recherche. Si le détecteur obtenu a l'avantage d'être global, son inconvénient est d'être non générique, dans le sens où il ne peut s'exécuter que dans un contexte bien précis.

4.3.2 Générateur de détecteurs

Dans l'approche précédente, le contexte est utilisé directement pour trouver un contour. Une autre idée consiste à utiliser le contexte pour trouver le détecteur permettant de trouver au mieux ce contour. Il devient alors possible d'alléger le problème de la généralité.

J. Hasegawa et al. [Has 86] proposent un système permettant de produire le meilleur détecteur pour un contour donné par un utilisateur. L'utilisateur spécifie le contour voulu par l'ensemble de points qui le constituent (*sample figure*). Le système cherche le meilleur détecteur permettant de trouver ce contour. Pour ce faire, le système est doté d'une bibliothèque contenant neuf détecteurs différents. Les neuf détecteurs sont exécutés en parallèle. Un autre processus est lancé pour construire l'image initiale à partir du contour donné (fig. 4.3). Il s'agit d'un processus de synthèse de l'image à partir du contour donné. Les résultats intermédiaires de chacun des détecteurs sont comparés aux résultats du processus de synthèse et le meilleur détecteur est retenu. Notons que l'idée de spécifier l'objectif par un exemple et de le représenter par "*sample figure*" est naturelle et facilite la tâche de l'utilisateur. Elle semble très intéressante, nous y reviendrons plus tard.

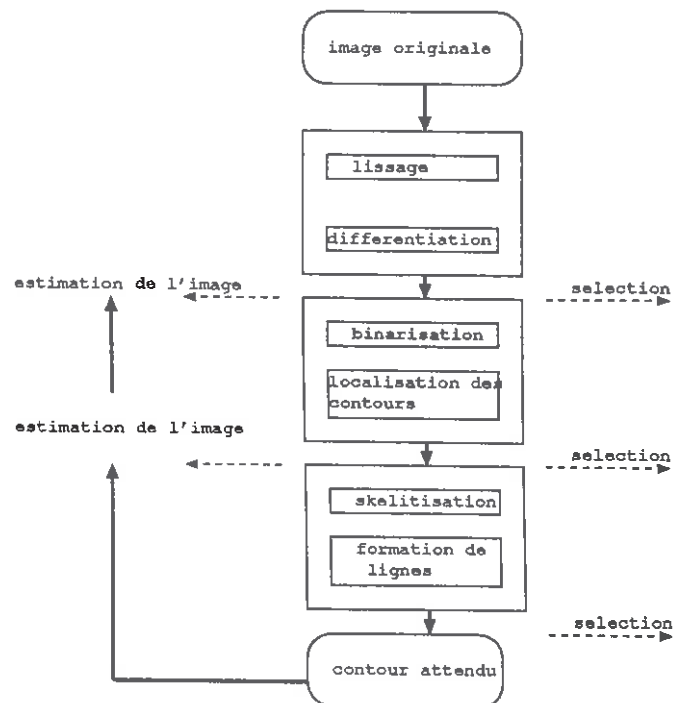


Figure 4.3. Schéma Fonctionnel du système IMPRESS.

En conclusion, ces deux approches complètement différentes, peuvent être caractérisées

communément par leur caractère contextuel et font appel aux techniques de l'intelligence artificielle. L'introduction de telles techniques dans la détection de contours ajoute des problèmes supplémentaires, en particulier la généralité et la combinatoire. En revanche, elles permettent l'utilisation des connaissances d'une façon non figée portant sur le contexte, comme sur l'expérience que les chercheurs ont acquis dans ce domaine pour dépasser les limites des mathématiques : *"les choses dans le monde ne sont pas assez simples pour être traitées par des outils mathématiques [Nag 84]"*.

4.4 L'approche multi-échelle

4.4.1 Définition

Un détecteur de contours comporte souvent un ou plusieurs paramètres. Une seule valeur utilisée lors de son exécution ne produit pas un détecteur optimal pour tous les contours contenus dans l'image. En d'autres termes, la qualité de leurs résultats ne dépend pas uniquement de la qualité de l'image, mais aussi de l'échelle utilisée ; c'est-à-dire la valeur de ses paramètres. Considérons un détecteur utilisant un lissage gaussien, avec comme paramètre l'écart-type σ :

- à grande échelle (c'est-à-dire, en utilisant une grande valeur de σ), le détecteur localise mieux certains types de contours (e.g. les contours dûs à l'illumination) comme le montre la figure 4.4.
- à basse échelle (c'est-à-dire, une petite valeur de σ), les doubles marches sont mieux localisées. Cependant, l'image de contours contient trop de détails (voir fig 4.4).

La détection de contours multi-échelle consiste alors à exécuter un détecteur à différentes échelles et à combiner (on dit aussi fusionner) les contours résultats pour aboutir à une seule image de contours.

4.4.2 La fusion des contours résultats

La fusion est le processus permettant de fabriquer une seule image de contours à partir de celles obtenues à différentes échelles. Il s'agit ici d'un problème délicat du fait qu'il faut être capable d'étiqueter identiquement un contour qui apparaît à plusieurs échelles. Cet étiquetage peut être effectué en se fondant sur la proximité des contours, leur orientation, etc. Cependant, une telle approche est souvent complexe et difficile à mettre en œuvre.

Le problème de fusion peut être simplifié, comme on le verra plus loin, si on se limite à l'utilisation d'un détecteur stable. Des études théoriques [Yui 86, Bab 86, Lu 89] ont montré l'existence de la propriété suivante de la gaussienne : en 1D, la gaussienne est l'unique filtre ne créant pas de nouveaux contours quand l'échelle augmente. En d'autres termes, un contour qui apparaît à une échelle donnée apparaîtra à toutes les échelles inférieures. Signalons que cette propriété de stabilité de la gaussienne n'est valable en 2D qu'avec l'utilisation du laplacien comme opérateur de différentiation [Yui 86, Bab 86].

Plusieurs travaux [Ros 71, Mar 80] suggèrent l'utilisation de plusieurs échelles pour aboutir à une description complète de l'image. Cependant aucune solution n'a été donnée pour résoudre le problème de fusion. Witkin [Wit 83] est l'un des premiers à proposer

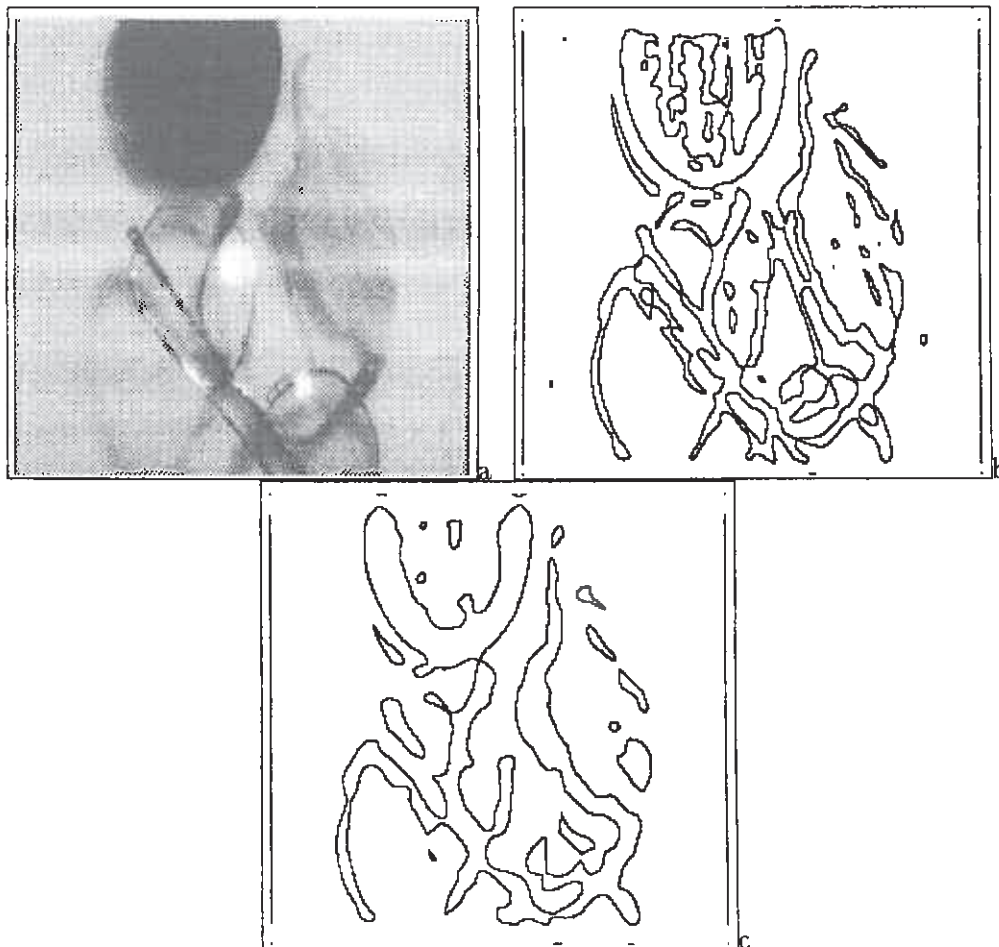


Figure 4.4. a) image originale, b) et c) contours résultats d'un laplacien d'une gaussienne à basse (masque de 25×25 pixels) et haute (masque de 41×41 pixels) échelle.

une solution dans le cas d'un signal 1D. En utilisant le détecteur de Marr et Hildreth, il introduit une représentation dans un plan $x - \sigma$ des passages par zéro obtenus à plusieurs échelles. Cette représentation, appelée *espace-échelle*, est construite pour tous les choix possibles de l'échelle σ .

L'espace de Witkin est alors une description à plusieurs échelles du signal étudié. La fusion est réalisée en représentant chaque point de contour par un couple tel que :

- la première composante indique la position du passage par zéro, qui est la position au point $\sigma = 0$,
- La seconde composante indique la plus grande échelle où la discontinuité est apparue.

D'autres travaux sur la fusion d'images de contours peuvent être consultés dans Canny [Can 83], Bergholm et Sjoberg [Sjo 87, Ber 87], Lacroix [Lac 90] et dans Ziou et Tabbone [Zio 91b, Tab 91a]. Tous ces travaux sont présentés dans la troisième partie de

cette thèse.

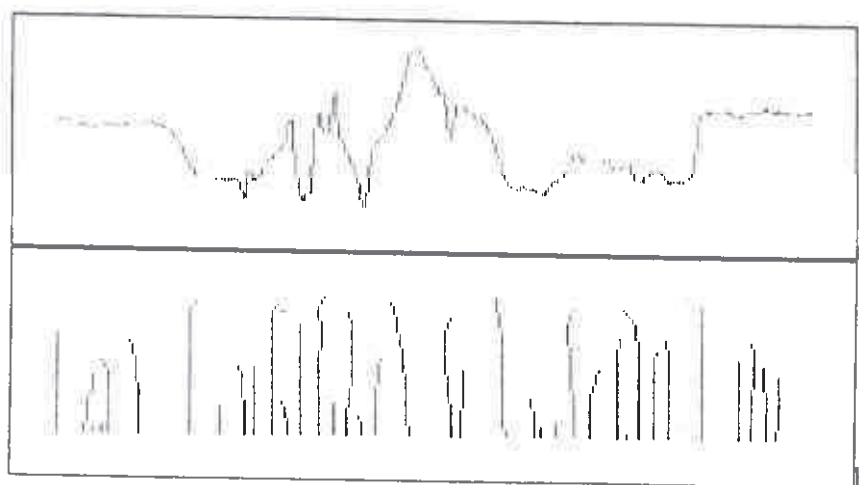


Figure 4.5. *L'espace-échelle de Witkin, a) un signal 1D, b) les passages par zéro correspondant, obtenus à des échelles variées. Un arche dans cet espace représente l'évolution du passage par zéro en fonction de l'échelle du détecteur utilisé.*

4.4.3 La problématique de la détection multi-échelle de contours

Rappelons que la méthode de détection de contours à plusieurs échelles telle qu'elle est décrite ci-dessus consiste à exécuter un détecteur à différentes échelles : $\sigma_1, \sigma_1 + \Delta\sigma, \sigma_1 + 2\Delta\sigma, \dots, \sigma_2$, avec $\Delta\sigma$ un nombre réel. Si $\Delta\sigma$ est négatif, il s'agit de détecter tout d'abord les contours les plus significatifs, puis les raffiner d'où l'appellation *coarse-to-fine*, s'il est positif c'est donc la stratégie inverse *fine-to-coarse*. La valeur optimale de $\Delta\sigma$ a été déterminée par Bergholm [Ber 87] tel que le déplacement d'un contour entre deux échelles consécutives permet une fusion des résultats obtenus. Il suggère dans le cas d'un détecteur gaussien l'utilisation d'un $\Delta\sigma = 0.5$. Il montre qu'un $\Delta\sigma = 1$ provoque une erreur de localisation de l'ordre de 2 pixels.

Cette approche de détection de contours pose un énorme problème d'efficacité que l'on peut résumer en deux points :

- Dans le cas d'un détecteur stable, un contour qui apparaît à haute échelle apparaîtra à toutes les échelles inférieures. Est-il nécessaire de traiter ce contour à chaque apparition ? A notre avis, la théorie des ondelettes apporte certainement une réponse à cette question. Cette technique consiste à décomposer un signal en un ensemble de fonctions de base orthogonales. Cette décomposition définit une représentation multi-échelle en ondelettes du signal original sur les deux domaines spatial et fréquentiel. Chacune des fonctions de base est une approximation du signal original à une échelle 2^n ($n = 1, 2, \dots$). Cette approximation est égale à la projection orthogonale du signal original sur l'espace V_{2^n} (l'ensemble de toutes les approximations possibles des fonctions à énergie finie à une échelle 2^n). L'approximation d'un signal à une échelle 2^{n+1} fournit toutes les informations nécessaires au calcul de l'approximation du signal à l'échelle 2^n . La différence d'informations (par exemple les nouveaux

contours) entre deux échelles successives 2^n et 2^{n+1} peut être déterminée par projection du signal original sur l'espace complémentaire de V_{2^n} . Pour plus de détails, on peut se référer aux travaux [Dor 88, Mal 89]. Les applications de cette technique dans les images commencent à voir le jour et en particulier dans les domaines de la segmentation d'images astrologiques [Bij 89], de codage d'image et d'analyse de texture [Mal 89].

- Est-il nécessaire d'utiliser un pas régulier ? La figure 4.5 montre un exemple où aucun nouveau contour n'apparaît à l'échelle $\sigma - \Delta\sigma$. Le problème d'efficacité peut être résolu si on est capable de prédire, à partir de l'échelle courante, l'échelle suivante qui permet de fournir des nouveaux points de contours et d'estimer le déplacement du contour entre les deux échelles. Malheureusement, il n'y a pas de bases solides pour choisir une échelle adéquate à partir des mesures *a priori* effectuées sur l'image. L'estimation automatique de l'échelle d'un détecteur est l'un des objectifs de nos travaux.

Certains auteurs [Sjo 87, Zio 91b, Tab 91a] proposent une remise en cause de l'approche multi-échelle telle qu'elle est décrite ci-dessus. En particulier, D. Ziou et S. Tabbone [Zio 91b, Tab 91a] proposent une étude de comportement de certains types de contours (marches, crêtes, escaliers) dans un espace échelle. Celle-ci permet de montrer que dans le cas d'une image ne contenant que ce type de contours, l'exécution d'un détecteur à deux échelles seulement (la plus haute et la plus basse) est suffisante pour la description complète de l'image.

En conclusion, l'approche multi-échelle n'est pas spécifique à la détection de contours. Elle a servi pour résoudre d'autres problèmes de la vision par ordinateur : extraction de régions [Blo 89], description d'objets [Cro 84]. En détection de contours, elle semble être une approche prometteuse, qui devrait être prise en compte dans de futures travaux. A notre avis, l'amélioration de cette approche, en particulier la recherche de solutions aux problèmes d'efficacité et de fusion doit commencer par des études sur le comportement des contours dans un espace échelle. C'est l'orientation explorée actuellement par S. Tabbone et D. Ziou. D'autres travaux sur le comportement du contour ont été proposés ces dernières années. Le lecteur intéressé peut se référer à l'article [Lu 89].

Enfin, notons que selon [Yui 86, Bab 86], seule le laplacien d'une gaussienne est le détecteur stable en 2D. Or, selon [Tor 86] la seconde dérivée directionnelle du laplacien l'est aussi si on effectue la différentiation avant le lissage. Selon [Wu 90], le filtre de lissage défini par $\exp(-a|x|)$ est stable aussi. Ceci explique le fait que la classe des détecteurs stables reste à définir. La généralisation de l'approche multi-échelle afin d'inclure les détecteurs dits non stables est aussi souhaitable.

4.5 Le seuillage

L'étape de seuillage a pour objectif d'augmenter le rapport signal/bruit d'un détecteur par élimination des faux points de contours. Malgré l'importance de cette étape, peu de travaux lui ont été consacrés ces dernières années. Ceci s'explique peut être par le fait que les détecteurs sont de plus en plus robustes. Or, les faux points de contours ne sont pas engendrés seulement par le bruit, ils peuvent l'être aussi par le contour lui même comme nous le montrons plus loin. Les algorithmes les plus utilisés pour éliminer les faux points de contours sont :

- **seuillage simple** : c'est l'algorithme le plus connu, il consiste à supprimer tous les points de contours dont la force (mesure de la plausibilité du contour, définie généralement par la valeur du module de son gradient ou de sa pente) est inférieure à un seuil donné. Cet algorithme a l'avantage d'être facile à implanter. Cependant, la décision d'éliminer un point n'est fondée que sur sa force indépendamment de son voisinage. Il en résulte des coupures de contours.
- **seuillage par hystérésis** [Can 83] : cet algorithme utilise deux seuils (un seuil haut et un seuil bas) et combine à la fois la continuité du contour et sa force de la façon suivante : une chaîne de contours possédant un certain nombre de points dont la force est supérieure à un seuil haut est un "vrai" contour. Il en est de même pour une chaîne dont tous les points possèdent une force supérieure à un seuil bas.

Quelle mesure de plausibilité du contour faut-il utiliser ?

Si la réponse est évidente dans le cas d'un détecteur utilisant un opérateur de différentiation du premier ordre, elle l'est moins dans le cas d'un opérateur de différentiation du second ordre. Certains auteurs [She 85, Cla 89] proposent d'utiliser le gradient comme mesure de plausibilité des contours résultats d'un détecteur utilisant un opérateur du second ordre. Or, à un "vrai" passage par zéro peut correspondre un gradient nul (e.g. fonctions harmoniques). L'utilisation du gradient pour caractériser la force des contours détectés aux passages par zéro n'est alors pas souhaitable. Pour pallier à ces problèmes, S. Tabbone et D. Ziou [Tab 91b] suggèrent l'utilisation de l'amplitude du passage par zéro comme mesure de plausibilité.

Origines des faux contours

Un détecteur de contours utilisant un opérateur de différentiation du second ordre (laplacien, dérivée seconde prise dans la direction du gradient) fournit au moins deux types de faux points [Tab 91b] :

- le premier type de faux points de contours est le plus connu. Il est souvent engendré par le bruit ("électronique"¹ et "sémantique"²) (cf. 2.2.1).
- le second type est moins connu. Il est engendré par les doubles marches de type escalier : à basse échelle, ce type de contour engendre trois passages par zéro dont un est faux, comme le montre la figure 4.6. Ces faux points correspondent aux minima locaux positifs ou au maxima locaux négatifs dans la dérivée première de l'image. Ils vérifient ainsi la condition suivante : le produit du gradient et de la troisième dérivée en ces points est strictement positif.

Ce dernier type de faux contours forme généralement une courbe continue prolongeant les vrais contours (fig. 4.7). Dans le cas de l'utilisation du gradient comme mesure de plausibilité du contour, il est évident que la prise en considération de la continuité lors du seuillage ne permet pas d'éliminer ce dernier type de faux points de contours. Ainsi, on peut affirmer que contrairement à ce qu'on croit, la prise en considération de la continuité du contour lors du seuillage n'est pas toujours souhaitable (voir fig. 4.7).

Dans le cas de l'utilisation de la pente, les faux contours engendrés par le bruit possèdent une valeur de plausibilité faible. Cependant, les faux engendrés par les doubles

1. bruit introduit par le dispositif physique lors de la formation de l'image

2. bruit dû à la nature de la scène observée, par exemple la texture.

marches possèdent une force comparable aux vrais contours. Le seuillage doit être alors effectué en deux étapes. La première a pour objectif l'élimination des faux points engendrés par les doubles marches en utilisant la condition ci-dessus. La seconde consiste en l'élimination des faux points engendrés par le bruit en utilisant un algorithme de seuillage classique (voir fig 4.7). Le lecteur intéressé peut trouver un complément d'informations dans les articles de S. Tabbone et D. Ziou [Tab 91b, Tab 91a].

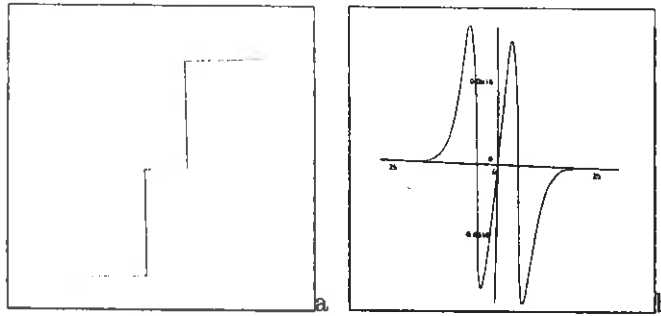


Figure 4.6. *Un exemple de l'origine des faux points de contours, a) une double marche de type escalier, b) sa seconde dérivée.*

Seuillage automatique

Le problème primordial du seuillage est le fait de fixer les seuils par expérience. S. Delpiane et al. [Del 89] ont déterminé empiriquement pour le détecteur de Marr et Hildreth une règle pour le calcul automatique du seuil en fonction des caractéristiques de l'image et de l'échelle du détecteur. Cette règle n'a été expérimentée pour le moment que sur des images de synthèse.

S. Tabbone et D. Ziou [Tab 91b] proposent un algorithme de seuillage automatique destiné à l'élimination des faux contours obtenus à différentes échelles. Cet algorithme fondé sur le comportement du contour dans un espace échelle (Cf. 4.4) procède par propagation de seuil. Il s'agit de déterminer expérimentalement le seuil dans l'image de contours obtenus à haute échelle, puis de calculer le seuil de n'importe quelle autre image en se fondant sur le seuil déjà calculé et la variation de l'échelle. Cet algorithme prend en considération les origines des faux contours.

4.6 Conclusion

La plupart des détecteurs rencontrés dans la littérature sont destinés aux contours de type marche. Ceci est dû au fait que les chercheurs dans ce domaine considèrent que la notion de contours est intuitivement liée à la notion de frontière des objets d'une scène et que les frontières engendrent des discontinuités dans la fonction image (Cf. 1.2.1). Il en résulte que l'on suppose acquise la notion de contours et on cherche les algorithmes adéquats. A notre avis, l'acquisition de toutes les informations contours passe par une compréhension de l'image : quelles sont les origines du contour et ses significations dans la scène ? Comment est-il représenté dans l'image ?

L'origine du contour reste mal connu et peu d'études lui ont été consacrées. Nous avons donné quelques éléments de réponse à ces questions dans les chapitres 2 et 3. Dans

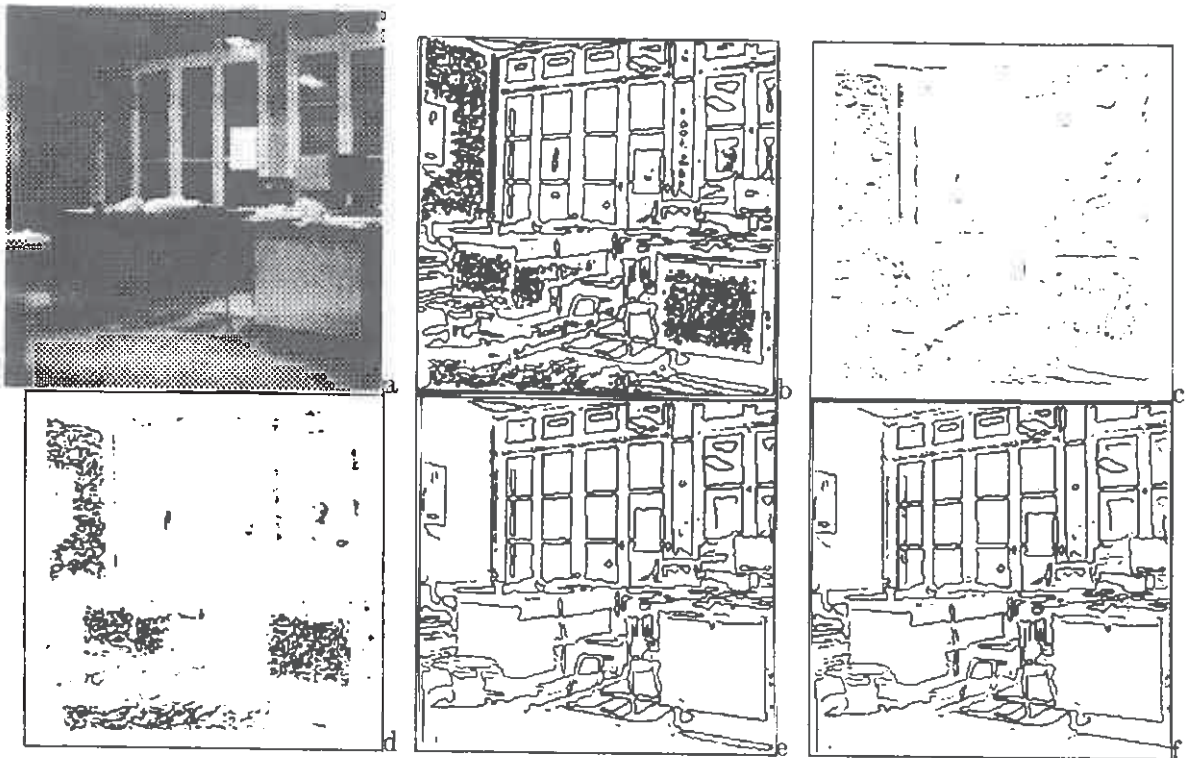


Figure 4.7. a) image originale, b) image de contours, c) image de faux contours engendrés par les doubles marches, d) image de faux contours engendrés par le bruit, e) image de contours seuillée par hystérésis f) image de contours en utilisant l'hystérésis et en éliminant les faux contours dus aux doubles marches.

ce contexte, nous avons montré que la prise en compte des autres types de contours semble être nécessaire. Nous reviendrons sur ce sujet dans la partie 2.

Enfin, nous distinguons les détecteurs de contours qui sont des outils et les méthodes de détection de contours. Une méthode inclut la façon d'utiliser un détecteur pour mettre en évidence le contour. Nous avons exposé la détection multi-échelle de contours qui est la seule méthode existante à notre connaissance. Plusieurs problèmes sous-jacents à cette méthode sont encore ouverts tels que la fusion, le nombre d'échelles et la stabilité. Apporter une solution à ces problèmes semble être une nécessité.

5

Problématique et Conclusion

Dans ce chapitre nous discutons les problèmes relatifs aux trois aspects suivants de la détection de contours : définition de contours, conception et implantation de détecteurs, méthodologie de détection de contours. Enfin, nous présentons les problèmes abordés dans les deux parties suivantes de cette thèse.

5.1 Caractérisation des résultats de la détection de contours

La détection de contours n'est pas un objectif en soi, elle est la première étape dans un système de vision. En effet, la plupart des détecteurs fournissent en résultat un ensemble de points. Ceux-ci sont regroupés pour former des chaînes de points de contours qui peuvent être considérées comme une représentation intermédiaire. La qualité des résultats et le temps de traitement de ce système dépendent de la qualité de cette représentation intermédiaire. Un bon détecteur a tendance à former des chaînes de points à partir desquelles une forme ou un objet peut être reconstruit avec un minimum de recherche (le minimum de divisions ou d'assemblages nécessaires pour trouver la forme reconnaissable). En revanche, un détecteur qui introduit des erreurs dans l'estimation de la position du contour ou qui forme des chaînes fragmentées conduit à une explosion combinatoire.

Malgré les efforts considérables déployés dans ce domaine, les détecteurs de contours existants ne répondent pas encore aux exigences des applications de la vision. Les résultats de ces détecteurs sont généralement incomplets, ce qui conduit souvent un système de vision à un échec. Plus précisément, la plupart des détecteurs sont caractérisés négativement par : la non détection des "vrais" points de contours, l'étiquetage éroné de "faux" points, et enfin une mauvaise estimation dans la position des contours détectés [Abd 78, Pra 78].

Ces erreurs proviennent essentiellement de la présence du bruit dans l'image. Or, le bruit est un élément de l'image difficile à caractériser et donc difficile à éliminer en préservant les contours. Une détection correcte des contours nécessite alors la prise en considération des problèmes que l'on peut regrouper en trois classes : problèmes liés à la définition du contour, ceux liés aux détecteurs, et enfin ceux liés à la détection de contours. C'est pourquoi nous consacrons les sections suivantes à l'analyse de telles classes de problèmes.

5.2 Problèmes liés à la définition du contour

Le contour est défini dans la plupart des travaux comme étant une discontinuité locale dans la fonction image. Cette définition pose deux problèmes :

- un examen visuel d'une image montre qu'on peut désigner subjectivement un contour qui ne correspond pas forcément à une discontinuité dans la fonction image (fig. 5.1). Un tel contour appelé contour subjectif a une origine mal connue [Web 84]. Sa détection ne peut être effectuée qu'à partir des informations image plus élaborées, voir même extérieures à l'image (le contexte par exemple).
- la notion de discontinuité doit être élargie plutôt à la notion de variation. Certaines variations autre que la discontinuité peuvent être à la base des approches de type "shape from X" et des applications de type analyse de documents [Tom 87, Ant 91] ou reconnaissance de caractères [Bel 91], etc. Nous avons montré au chapitre 4 que les détecteurs de variations autres que le point d'inflexion sont rares. La prise en considération d'autres variations dans un système de vision permettra certainement une description structurale complète de l'image et conduit donc à des résultats meilleurs. Cependant, de nouveaux problèmes apparaissent, en particulier le problème d'intégration de différents types de contours. Le point de départ dans cette direction peut être le recensement des différentes variations à considérer en fonction du processus de formation d'images pour une amélioration des résultats de la détection de contours.

Par ailleurs, la nature locale de la définition du contour fait qu'un point de l'image est considéré d'une façon isolée des autres points. Il en résulte que la décision de marquer un point comme contour ne tient compte que d'un nombre très limité de points de contours voisins. Une amélioration de la définition du contour semble être la considération des modèles de contours de type multi-marche (c'est-à-dire, une marche suivie d'une marche inverse qui se répète aléatoirement) pour inclure d'autres phénomènes de contours (e.g., une marche de type escalier engendre un faux passage par zéro) et pour dériver des détecteurs adaptés [She 86, Che 89, Tab 91b].

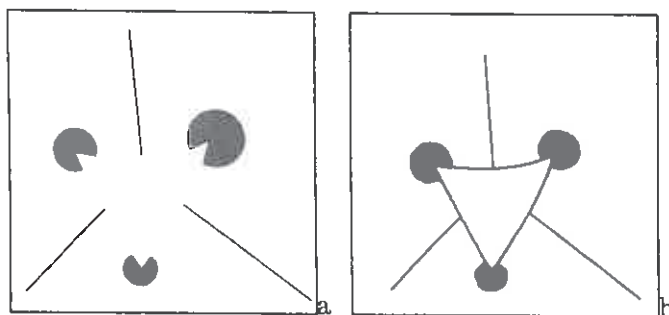


Figure 5.1. Exemple de contour subjectif, a) le triangle de Kanizsa, b) le contour subjectif détecté par l'algorithme de Webb [Web 84].

5.3 Problèmes liés aux détecteurs

5.3.1 Dérivation de détecteurs

La dérivation de détecteurs à partir d'une modélisation de contours est l'une des approches les plus répandues. Elle permet de fournir des détecteurs plus adaptés aux exigences des applications actuelles. Cependant, le modèle contour utilisé est souvent simple et ne simule pas avec précision la réalité. On peut citer comme exemple l'utilisation d'un modèle 1D pour dériver un détecteur destiné aux images 2D. L'utilisation de modèles plus réalistes (modèle 2D, multi-marche, multi-toit, etc) permettra certainement une amélioration des résultats de la détection de contours.

5.3.2 Implantation d'un détecteur

Un détecteur de contours résultat d'une longue et fastidieuse étude théorique doit être traduit en un programme pour pouvoir être utilisé. Dans le cas où le détecteur est un simple filtre (cf. 3.3), l'étape d'implantation consiste à développer un algorithme qui met en œuvre l'opération de filtrage. L'activité d'implantation d'un détecteur est parfois une tâche fastidieuse qui nécessite des développements mathématiques complexes.

Le traitement de signal fournit plusieurs méthodes d'implantation [She 86, For 88] : les masques classiques de convolution, la transformée de Fourier rapide (FFT), la corrélation hiérarchique, les filtres numériques, etc. Etant donné un détecteur de contours, quelle méthode d'implantation faut-il utiliser ? D'une façon générale, qu'attend-t-on d'une méthode d'implantation ?

Une bonne méthode d'implantation doit être caractérisée par :

- La préservation des propriétés du détecteur initial : le détecteur initial est souvent un filtre dont la réponse impulsionnelle est continue et à support infini. La plupart des méthodes d'implantation nécessitent un filtre discret et à support limité. Ainsi, lors de l'implantation d'un détecteur, des approximations du filtre initial sont souvent effectuées (l'échantillonnage, la coupure). Ces approximations donnent lieu à des effets indésirables (effet de Gibbs, recouvrement, etc) qui s'accroissent davantage en fonction de l'erreur et fournissent comme résultats des images de contours bruitées.

La plupart des travaux sur l'implantation proposés dans la littérature concernent le laplacien d'une gaussienne [Che 87, Sot 89, For 88]. Le masque de convolution est l'outil habituel pour implanter ce détecteur. L'inconvénient de l'utilisation de ce type de méthodes est le fait que les pixels qui se trouvent aux bords de l'image ne sont pas traités. La proportion de ces pixels peut atteindre jusqu'à 30% de l'image.

D'une façon générale, les erreurs d'approximation ont pour conséquence la non préservation des propriétés du détecteur initial. Le détecteur implanté ne possède alors pas les mêmes propriétés que le détecteur initial. Nous avons étudié un tel problème dans [Zio 91a]. Le problème de choix d'une méthode d'implantation préservant les propriétés initiales d'un détecteur donné est généralement complexe et reste ouvert.

- L'efficacité : un détecteur de contours doit être implanté sans oublier qu'il est souvent la première étape dans un système de vision. Il doit être implanté efficacement pour pouvoir s'exécuter dans un temps "raisonnable". Le choix de la méthode d'implantation est souvent fondé sur le seul critère d'efficacité.

A notre avis, le choix d'une méthode d'implantation doit être fondé sur l'erreur d'approximation. L'efficacité n'est pas un critère dominant, sauf si des contraintes de type temps réel ou autres sont imposées.

5.3.3 Evaluation des résultats d'un détecteur

D'une façon générale, l'évaluation des résultats d'un détecteur de contours permet de mettre en évidence ses failles, de qualifier ses performances et éventuellement, de produire un détecteur flexible (un détecteur ayant la capacité d'ajuster ses paramètres). Pour évaluer, il faut disposer d'un critère d'évaluation et/ou d'une référence exprimée dans un langage précis. On distingue deux méthodes d'évaluation : l'évaluation subjective et l'évaluation objective.

- Evaluation subjective : comme son nom l'indique, elle consiste à présenter les résultats d'un détecteur de contours à un être humain. Celui-ci porte un jugement sur le détecteur à travers ses résultats. Cette méthode est la plus répandue ; elle a l'avantage d'être simple. Cependant, elle pose certains problèmes :
 - cette méthode s'appuie essentiellement sur la visualisation de l'image de contours. Généralement, seule la position, et éventuellement la force des points de contours, sont visibles. L'évaluation ne tient alors pas compte de toutes les informations contours et elle fait intervenir d'autres informations extérieures à l'image (des connaissances *a priori* sur l'image, des contours subjectifs).
 - une évaluation nécessite la présence de critères et d'une référence. Or, les critères et la référence utilisés par un être humain sont mal connus et dépendent de son expérience dans le domaine et de son "attache" à l'approche adoptée. D'une façon générale, on ne regarde pas si le détecteur est conforme à ses spécifications initiales, mais plutôt on regarde s'il fournit des contours qui répondent à nos besoins.

L'évaluation subjective n'est donc que vague. Par conséquent, elle ne peut être utilisée pour mesurer les performances d'un détecteur, mais seulement pour détecter ses failles.

- Evaluation objective : Il s'agit de mesurer les performances d'un détecteur relativement à un objectif exprimé par un utilisateur dans un langage précis. Deux approches sont proposées dans la littérature.
 - La première appelée *évaluation quantitative* ne concerne que l'étape de lissage du détecteur de contours. L'objectif est de mesurer dans une image lissée le compromis entre la préservation des contours et la quantité de bruit éliminé. Une étude de synthèse des techniques utilisées peut être consultée dans l'article de R.T. Chin et C.L. Yeh [Chi 83].
 - La seconde approche a pour objectif l'évaluation des performances d'un détecteur à partir de ses résultats. Il s'agit de proposer des critères de performance exploitables pour décrire les qualités requises d'un détecteur de contours. Pratt [Pra 78] et Abdou [Abd 78] ont proposé un critère fondé sur l'exhaustivité de détection et l'exactitude dans la localisation. La mise en œuvre de leur critère

nécessite la connaissance de la structure du contour, en particulier sa position. Il en résulte une limitation quant à son utilisation dans le cas des images de synthèse.

En conclusion, le problème d'évaluation des performances d'un détecteur de contours est souvent passé sous silence, malgré l'importance capitale qu'il représente. Les méthodes d'évaluation doivent être développées avec le même intérêt que les méthodes de lissage ou de différentiation. A notre avis, une évaluation correcte ne peut être faite que sur des images naturelles et doit tenir compte du contexte dans lequel s'exécute le détecteur et de son utilisation ultérieure. Une telle méthode d'évaluation reste à concevoir.

5.4 Méthodologie de détection de contours

La plupart des détecteurs n'incluent ni leur contexte d'utilisation (c'est-à-dire le type d'images et les caractéristiques du contour où ils peuvent être utilisés avec succès) ni les règles pour choisir l'échelle adéquate. La sélection et l'utilisation d'un détecteur, étant donnée une image, est un problème complexe. En pratique, la plupart du temps, on intègre dans un système de vision un détecteur de contours connu. Celui-ci est exécuté sur toutes les images utilisées par le système pour déterminer tous les contours. Cette méthode ne peut pas conduire à des résultats satisfaisants pour les raisons déjà évoquées au chapitre 3 et 4, que nous résumons ci-dessous :

- les algorithmes proposés dans la littérature sont multiples et souvent destinés à un contour de type particulier, de forme particulière dans une image particulière. De plus l'utilisation d'une seule échelle ne peut produire un détecteur optimal pour tous les contours de l'image.
- les détecteurs d'un même type de contours possèdent des propriétés différentes et peuvent fournir donc des contours différents.

Il n'existe pas d'algorithme qui s'exécute mieux que les autres sur toutes les images ; ils ont chacun leur domaine d'application *ad hoc* pour lequel ils fournissent de bons résultats [Zio 91c]. L'approche de détection de contours multi-échelle telle qu'elle a été présentée dans le chapitre 4 apporte une solution particulière aux problèmes de choix de l'échelle d'un détecteur. Cependant, il reste à apporter une solution au problème clé de la fusion. A notre avis cette approche est insuffisante pour une description complète de l'image, puisqu'elle se limite à l'utilisation d'un seul détecteur.

5.5 Conclusion

L'objectif de cette première partie est de rassembler un maximum d'expertises indispensables pour aborder le problème présenté dans la troisième partie. C'est pourquoi nous avons présenté un aperçu sur la détection de contours en abordant ses différents aspects. Ainsi, nous avons mis en évidence la notion de contours et l'influence des propriétés d'un détecteur sur le contour résultat.

A travers l'étude des détecteurs existants, nous avons soulevé plusieurs problèmes. En particulier, ceux relatifs aux trois aspects suivants de la détection de contours : la

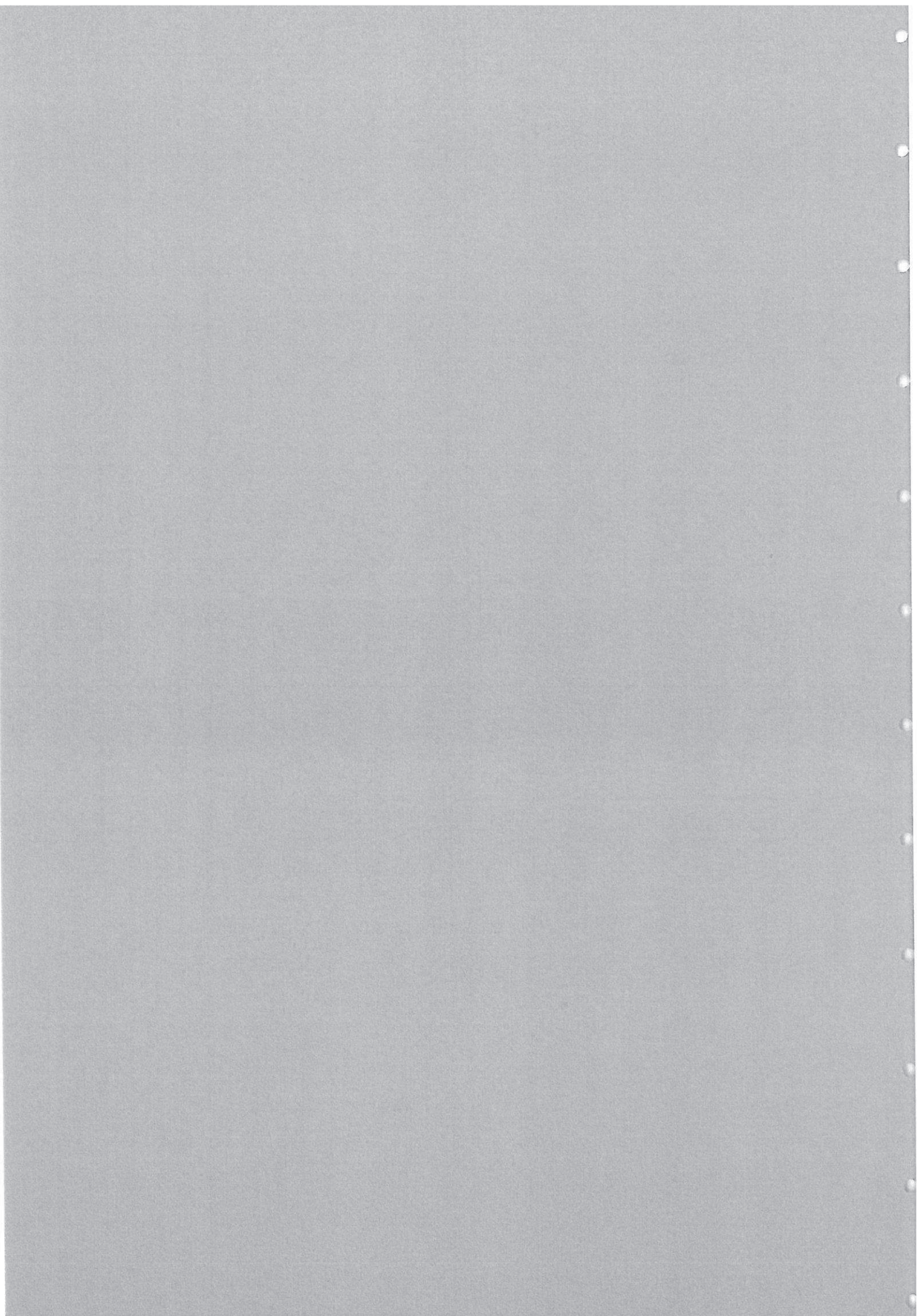
définition du contour, la conception et l'implantation de détecteurs, la méthodologie de détection de contours.

Pour notre part, nous nous sommes intéressés particulièrement aux problèmes liés à la méthodologie de la détection de contours. Dans ce contexte, nous proposons :

- dans la partie 2 de cette thèse, un détecteur optimal de lignes de crêtes issu des critères de Canny et un détecteur de contours de type marche destiné à des images simples. Nous abordons également les problèmes d'évaluation et d'implantation d'un détecteur.
- un système de sélection de détecteurs de contours et leurs paramètres nécessaires à la mise en évidence d'un contour donné, dans la partie 3.

PARTIE II

Les détecteurs de contours



Résumé

Cette partie est consacrée essentiellement à la présentation de détecteurs de contours développés pour être intégrés dans le système de sélection de détecteurs qui sera décrit dans la troisième partie.

Nous présentons dans le chapitre 1 les critères de performances de Canny et les détecteurs de contours de Shen et Castan ainsi que celui de Deriche. Ceux-ci seront utilisés dans la troisième partie.

Le chapitre 2 décrit un détecteur de contours de type ligne de crête que nous avons conçu et implanté. Ce détecteur est optimal au sens des critères de Canny.

Enfin, le chapitre 3 inclut un détecteur de contours de type marche à partir de la distribution fréquentielle d'une image. Ce détecteur que nous avons conçu et implanté est destiné à des images comportant quelques objets posés sur un fond.

1

Quelques détecteurs optimaux

Ce chapitre a pour objectif de décrire les critères de performances d'un détecteur de contours proposés par Canny, qui sont la base de nos travaux. Nous présentons aussi les détecteurs de Shen et Castan ainsi que celui de Deriche que nous utilisons dans la 3^{ème} partie de cette thèse.

1.1 Détecteur de J.F. Canny

1.1.1 Critères de Canny

Canny a étudié les propriétés souhaitées d'un détecteur de contours performant. Elles sont fondées sur l'exhaustivité de la détection et l'exactitude de la localisation des points marqués *contours*. Il propose dans sa thèse [Can 83] de quantifier la qualité de détection par la probabilité de ne pas omettre un vrai point de contours et de ne pas marquer *contour* un point image qui ne le soit pas. L'erreur de localisation est quantifiée par la distance entre le vrai point de contours et le point image marqué *contour*. Un "bon" détecteur de contours doit alors maximiser la probabilité d'une détection exhaustive et minimiser l'erreur de localisation. Ceci revient, selon Canny, à maximiser le rapport signal/bruit de la réponse du détecteur et à minimiser la variance de l'erreur de localisation.

Plus précisément, Canny considère que l'image perçue, notée $I(x)$ est définie par :

$$I(x) = F(x) + N(x) \quad (1.1)$$

avec $F(x)$ le signal non bruité et $N(x)$ le bruit blanc.

L'objectif est de déterminer la réponse impulsionnelle du filtre $f(x)$ qui répond à un point de contours par un maximum local. Pour ce faire, Canny considère le rapport signal/bruit, noté $\Sigma(f)$ comme le rapport de la réponse de $f(x)$ à l'image $F(x)$ sur la réponse moyenne du filtre au bruit $N(x)$.

Le rapport signal/bruit peut être alors défini par :

$$\Sigma(f) = \frac{\int_{-\infty}^{+\infty} F(-x)f(x)dx}{n_0 \sqrt{\int_{-\infty}^{+\infty} f^2(x)dx}} \quad (1.2)$$

où n_0 est l'énergie du bruit.

Au lieu de considérer l'erreur de localisation, telle qu'elle est définie ci-dessus, Canny considère la localisation comme étant l'inverse de la variance de la distance entre le vrai point de contours et le point de contours marqué. La localisation est donnée par :

$$\Lambda(f) = \frac{\int_{-\infty}^{+\infty} F(-x) f''(x) dx}{n_0 \sqrt{\int_{-\infty}^{+\infty} f'^2(x) dx}} \quad (1.3)$$

Le critère de performances d'un détecteur est défini par le produit du rapport signal/bruit et de la localisation. Il transforme ainsi la recherche d'un "bon" détecteur en un problème d'optimisation : on cherche le détecteur f qui maximise le critère $C(f)$ donné par :

$$C(f) = \Sigma(f) \Lambda(f) \quad (1.4)$$

en respectant les contraintes suivantes :

- le détecteur f ne doit fournir en réponse qu'un seul point marqué lorsqu'il n'existe qu'un seul point de contours effectif,
- d'autres contraintes liées à la fois à l'ordre de l'opérateur de différentiation et au type de contours (c'est-à-dire le profil $F(x)$) que l'on veut considérer. Ces contraintes sont détaillées dans le paragraphe suivant pour les contours de type marche et dans le paragraphe 2.2 pour les contours de type crête.

1.1.2 Le détecteur de marche

Il s'agit d'adapter le critère de performances ci-dessus pour dériver un détecteur de contours de type marche aux maxima de la dérivée première. Considérons que le profil 1D idéal d'un contour de type marche est défini par :

$$F(x) = \begin{cases} A & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases} \quad (1.5)$$

En remplaçant $F(x)$ par sa valeur dans le rapport signal/bruit défini par l'équation 1.2, on obtient :

$$\Sigma(f) = \frac{\int_{-\infty}^0 f(x) dx}{\sqrt{\int_{-\infty}^{+\infty} f'^2(x) dx}} \quad (1.6)$$

Il est de même pour l'équation 1.3 qui devient :

$$\Lambda(f) = \frac{|f'(0)|}{\sqrt{\int_{-\infty}^{+\infty} f'^2(x) dx}} \quad (1.7)$$

Enfin, le critère de performances (eq. 1.4) s'écrit sous la forme suivante :

$$C(f) = \frac{\int_{-\infty}^0 f(x) dx}{\sqrt{\int_{-\infty}^{+\infty} f'^2(x) dx}} \times \frac{|f'(0)|}{\sqrt{\int_{-\infty}^{+\infty} f'^2(x) dx}} \quad (1.8)$$

La recherche du filtre $f(x)$ revient à maximiser $C(f)$ en présence des trois contraintes suivantes :

- $f(x)$ fournit des contours suffisamment distants les uns des autres d'environ kW , où k est de l'ordre de 0.59. Cette contrainte permet d'éviter des détecteurs à réponses multiples,
- $f(x)$ est anti-symétrique, c'est-à-dire que $f(x) = -f(-x)$. Cette contrainte est due simplement au choix de la dérivée première comme opérateur de différentiation,
- $f(x)$ est à support limité, c'est-à-dire que $f(x) = 0$, si $x \notin [-W, W]$. Cette contrainte n'est introduite que pour simplifier la résolution du problème.

Ce problème est résolu par le biais de méthodes variationnelles. Le détecteur résultat est donné par :

$$f(x) = \begin{cases} \sin(wx)(a_1e^{\alpha x} + a_3e^{-\alpha x}) + \cos(wx)(a_2e^{\alpha x} + a_4e^{-\alpha x}) + c & \text{si } |x| \leq W \\ 0 & \text{sinon} \end{cases} \quad (1.9)$$

où α et w sont des constantes réelles. Les coefficients a_1, a_2, a_3, a_4, c peuvent être déterminés grâce aux conditions limites suivantes :

$$f(0) = 0, \quad f(W) = 0, \quad f'(0) = S, \quad f'(W) = 0 \quad (1.10)$$

où S est une constante. Pour faciliter son implantation, Canny propose d'utiliser comme détecteur la première dérivée d'une gaussienne, qui est une bonne approximation du détecteur initial (1.9) :

$$f(x) = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \quad (1.11)$$

Il s'agit maintenant d'étendre ce filtre en 2D pour être utilisé en détection de contours dans des images à niveaux de gris.

Pour ce faire, il propose l'utilisation du filtre 2D séparable suivant :

$$f(x, y) = -\frac{x}{\sigma^2} e^{-(x^2+y^2)/2\sigma^2} - \frac{y}{\sigma^2} e^{-(x^2+y^2)/2\sigma^2} = f_x(x, y) + f_y(x, y) \quad (1.12)$$

L'algorithme pour détecter les contours de type marche est alors immédiat :

- convolution de l'image originale $I(x, y)$ avec un filtre $f_x(x, y)$, fournissant $I_x(x, y)$ comme image résultat. $I(x, y)$ est de nouveau convoluée avec le filtre $f_y(x, y)$. Soit $I_y(x, y)$ l'image résultat. Intuitivement, $I_x(x, y)$ (resp. $I_y(x, y)$) correspond à la dérivée première par rapport à x (resp. y) de l'image lissée $I(x, y) * g(x, y)$, où $*$ dénote la convolution et $g(x, y)$ étant la gaussienne.
- identification des points de contours dans l'image gradient, c'est-à-dire l'image où à chaque point est affecté la valeur du module du gradient. Pour ce faire, Canny propose un algorithme de recherche des maxima locaux dans la direction du gradient appelé algorithme de *suppression des non-maxima*. Les règles de calcul du module du gradient et de son orientation sont données en paragraphe 3.3.2.

1.2 Détecteur de R. Deriche

Le filtre initial de Canny (eq. 1.9) est à support limité. Deriche [Der 87] propose l'extension de l'étendue de ce filtre à l'infini ($W \rightarrow \infty$). En utilisant les conditions limites données dans (eq. 1.10) avec ($W \rightarrow \infty$), c'est-à-dire :

$$f(0) = 0, \quad f(\infty) = 0, \quad f'(0) = S, \quad f'(\infty) = 0,$$

Deriche obtient le détecteur suivant :

$$f(x) = -ce^{-\alpha|x|} \sin(wx) \quad (1.13)$$

avec c constante réelle, α et w constantes positives.

Ce détecteur est utilisé sur des images bidimensionnelles d'une façon similaire à celle de Canny. Plus précisément

$$f(x, y) = -\frac{ck}{\alpha^2 + w^2} e^{-\alpha(|x|+|y|)} [\sin(wx)(\alpha \sin(w|y|) + w \cos(w|y|)) + \sin(wy)(\alpha \sin(w|x|) + w \cos(w|x|))] \quad (1.14)$$

L'algorithme de détection de contours est semblable à celui de Canny présenté dans le paragraphe précédent. Cependant, Deriche implante son détecteur efficacement par des équations aux différences d'ordre deux. Nous avons réalisé et expérimenté ce détecteur. Les résultats obtenus dans le cas des images de scènes d'intérieur sont en général satisfaisants (fig. 1.1.a et 1.1.b).

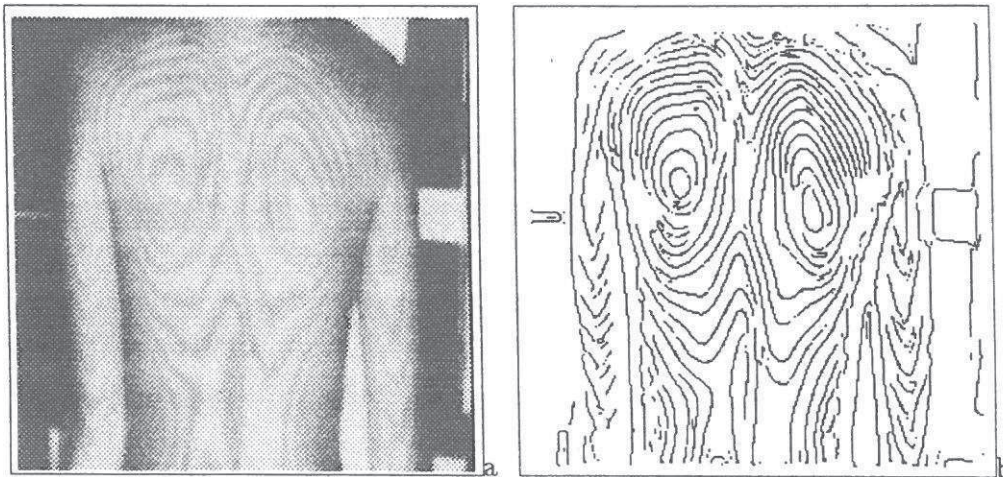


Figure 1.1. Un exemple de résultat du détecteur de Deriche avec $\alpha = 1$. (a) image originale, (b) image de contours.

1.3 Détecteur de J. Shen et S. Castan

Shen [She 86] propose trois critères que doit vérifier un détecteur de contours au maxima de la dérivée première :

- le premier critère concerne le filtre de lissage, noté $f(x)$. Il indique que l'énergie E_n de la sortie de $f(x)$ en réponse au bruit doit être minimale,
- les deux autres critères concernent les propriétés différentielles de $f(x)$. Pour que ce filtre détecte les contours aux maxima de la dérivée première, l'énergie E_a de la réponse de $f'(x)$ au bruit doit être minimale et l'énergie E_s de la réponse de ce même filtre $f'(x)$ au contour doit être maximale.

La recherche d'un détecteur selon Shen et Castan revient à minimiser la fonctionnelle suivante :

$$C(f) = \frac{E_n(f) E_a(f)}{E_s^2(f)} \quad (1.15)$$

Ce problème est résolu par des méthodes variationnelles dans le cas d'une marche idéale (eq. 1.5) et le filtre de lissage $f(x)$ résultat est donné par :

$$f(x) = ab^{|x|} \quad (1.16)$$

avec $a = -\frac{1}{2} \ln(b)$ et $0 < b < 1$.

La fonctionnelle (eq. 1.15) exprime uniquement un rapport bruit/signal. La localisation ne fait pas partie des critères de performances de Shen et Castan. Cependant, ils montrent dans [She 87, Cas 90] que tout détecteur utilisant ce filtre possède les caractéristiques suivantes :

- l'erreur de localisation est théoriquement égale à zéro,
- il est optimal aussi dans le cas des contours de type multi-marche.

Une variété de détecteurs de contours, appelé ISEF, a été proposée à partir du filtre exponentielle [Cas 90]; en particulier, les détecteurs DRF, GEF et SDEF. Le premier localise les contours aux passages par zéro du laplacien, le second aux maxima locaux du gradient et le dernier aux passages par zéro de la seconde dérivée directionnelle. Ces deux détecteurs sont implantés par le biais des équations aux différences d'ordre 1. Ils possèdent donc une complexité relativement faible. Nous avons également réalisé et expérimenté le détecteur DRF. Un exemple de résultat est présenté dans la figure 1.2.

1.4 Conclusion

Nous avons décrit dans ce chapitre les critères de performances d'un détecteur de contours proposés par Canny. Nous avons présenté ensuite trois détecteurs de contours de type marche. Le détecteur de Canny et celui de Deriche sont issus de critères de performances de Canny. Cependant, le détecteur de Shen et Castan est dérivé à partir

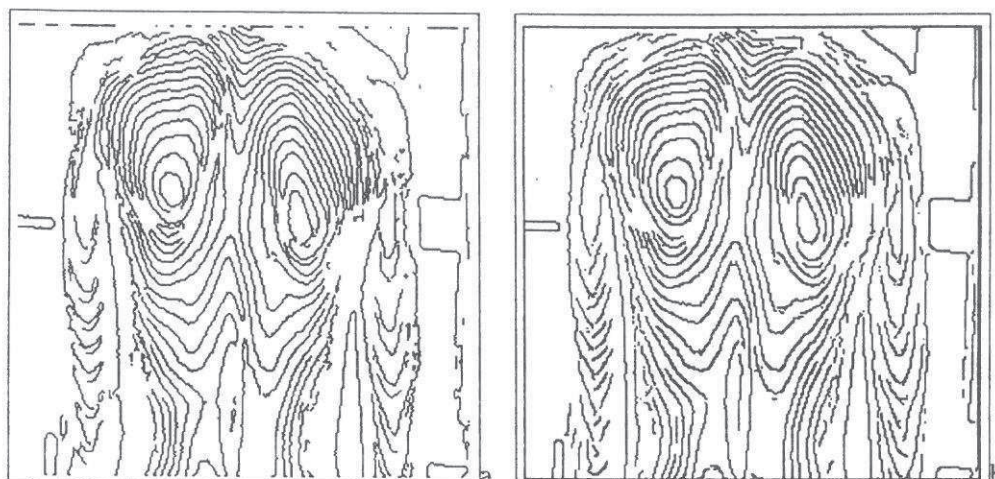


Figure 1.2. Exemple de contours obtenus par les détecteurs de Shen et Castan sur l'image donnée en figure 1.1. (a) résultat de DRF avec $a = 0.35$. (b) résultat de GEF avec $a = 0.35$.

d'autres critères qui sont également présentés. Ce dernier détecteur est optimal aussi selon les critères de Canny et possède de meilleures performances que les deux premiers. Nous avons implanté et expérimenté le détecteur de Deriche et le détecteur DRF de Shen et Castan. Leurs résultats semblent être satisfaisants au moins dans le cas des images de scènes d'intérieur.

Les critères de Canny ne se limitent pas aux détecteurs de contours de type marche. Ils ont servi pour dériver des détecteurs de contours de différents types. Nous présentons dans le chapitre suivant un détecteur de contours de type lignes de crête issus de ces critères de performances.

2

Un détecteur optimal de lignes de crête

Ce chapitre décrit un détecteur optimal de contours de type ligne de crête issu des critères de Canny. Dans les premiers paragraphes, nous exposons le détecteur 1D et son extension en 2D. Nous proposons l'utilisation de la variance directionnelle pour l'identification des contours. Le détecteur 2D étant séparable et est implanté par des équations aux différences. Enfin, nous présentons une évaluation de ses performances et nous montrons les problèmes qui peuvent se produire pour intégrer de telles informations dans un système de vision par ordinateur.

2.1 Introduction

La plupart des détecteurs existants sont destinés aux contours de type *marche* [Can 83], [Har 84], [She 86], [Tor 86], [Der 87], [Ber 87] ; ces contours sont effectivement les plus nombreux dans une image. Cependant, la prise en compte des autres types de contours s'avère nécessaire, et ceci essentiellement pour deux raisons :

- Certaines approches d'acquisition de l'information 3D, notamment les "*shape from contour*" [Mar 80] [Pon 85] nécessitent de déterminer tous les types de contours.
- Il existe des applications s'appuyant sur des contours pas forcément le type *marche*. Citons, par exemple, le traitement d'images de traits comme les empreintes digitales ou les caractères. En effet, ces images contiennent des lignes minces difficiles à localiser par un détecteur de contours de type *marche*.

Nous nous intéressons plus particulièrement aux contours de types lignes de crête. On appelle ligne de crête les variations du niveau de gris dans l'image correspondant à un *toit* ou une *crête* de largeur de l'ordre de quelques pixels (voir figure 2.1). Ces deux types de contours se ressemblent beaucoup et peuvent être détectés par le même outil.

Rappelons que Canny [Can 83] a proposé un ensemble de critères permettant de trouver un détecteur optimal d'un type de contours donné (Cf. 1.1). Notre travail s'inscrit dans cette direction : concevoir et implanter un détecteur optimal de contours *lignes de crête* à partir des critères de Canny [Zio 89c, Zio 91a]. Nous présentons :

- le détecteur optimal de lignes de crête obtenu à partir des critères de Canny,

- l'extension de ce filtre en deux dimensions et la définition de l'orientation d'un contour de ce type. Le filtre 2D étant séparable et est implanté alors efficacement à l'aide des équations aux différences,
- l'algorithme de détection de lignes de crête,
- l'évaluation des performances de ce détecteur, en particulier le rôle de chacun des paramètres du détecteur et leur influence sur ses performances, ainsi que les résultats expérimentaux. L'expérimentation a montré l'existence de certains phénomènes physiques qu'il faut considérer lors de la conception d'un détecteur de lignes de crête et qui seront détaillés dans le paragraphe 2.6.5.

2.2 La dérivation du détecteur 1D

L'objectif est de déterminer analytiquement la réponse impulsionnelle d'un filtre $f(x)$ permettant de détecter les *lignes de crête* (c'est-à-dire à la fois les contours *toits* et les contours *arêtes* représentés figure 2.1) aux maxima de la réponse de celui-ci au signal d'entrée $I(x)$ (1.1). Pour une raison de simplicité, nous modélisons le profil d'un contour de type ligne de crête par la fonction F :

$$F(x) = Ae^{-|x|} \quad (2.1)$$

avec A constante positive.

Avant de proposer la fonction $f(x)$ représentant ce profil, examinons ses propriétés :

1. Elle doit être paire : en effet si la réponse impulsionnelle n'est pas paire, le maximum de la sortie pourra ne pas correspondre au pic dans le signal d'entrée. De plus le rapport signal/bruit du détecteur se dégrade (voir la démonstration en Annexe A).
2. Elle doit fournir *zéro* en réponse au signal constant. Ainsi, l'amplitude de la réponse diminue d'une façon lisse et atteint zéro aux extrémités.
3. Elle doit être à support infini : la raison essentielle à cela est l'utilisation des équations aux différences pour implanter le détecteur.

Partant de ces propriétés, il s'agit de chercher la réponse impulsionnelle du détecteur maximisant la fonctionnelle donnée par l'équation 1.4. La résolution analytique de ce problème dans le cas de contours de type ligne de crête s'avère difficile. Canny procède par des méthodes d'optimisation numérique pour le résoudre. Ceci lui permet de déterminer les formes des détecteurs optimaux des contours de type toit et arête. La figure 2.1 décrit les deux graphes des filtres optimaux obtenus par une optimisation numérique de la fonctionnelle donnée par l'équation 1.4.

Un examen de la figure 2.1 nous conduit à modéliser ce filtre par la dérivée seconde d'une gaussienne. Cependant, nous lui préférons une autre fonction de forme similaire que l'on peut facilement étendre en 2D et qui peut être implantée par des équations aux différences. La réponse impulsionnelle de ce filtre est donnée par :

$$f(x) = (a \sin(w | x |) + b \cos(w | x |))e^{-\alpha|x|}$$

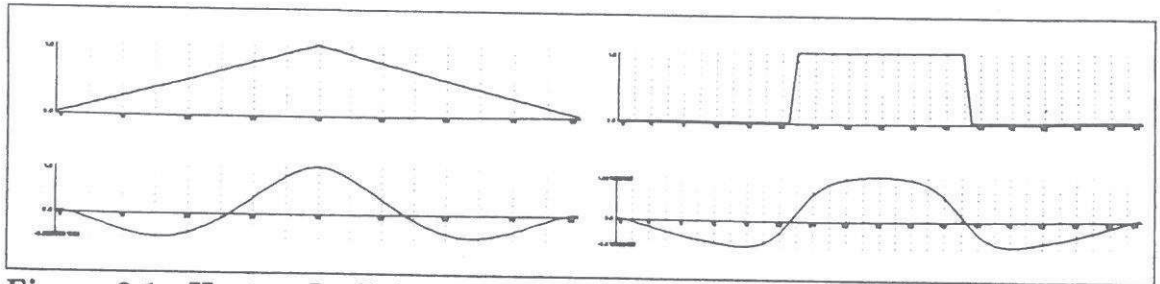


Figure 2.1. Haut : Profil du contour de type "toit" et d'un contour de type "arête". Bas : Profil du filtre optimal de détection des contours de type "toit" et "arête" [Can 83].

où a, b, α et w sont des constantes.

La fonction est symétrique (elle satisfait donc la première propriété). Elle est aussi à support infini (propriété 3). Cependant, elle ne fournit pas zéro en réponse au signal constant (propriété 2). Pour ce faire, il faut que :

$$\int_{-\infty}^{+\infty} f(x) dx = 0$$

Puisque $f(x)$ est symétrique, cette équation devient :

$$2 \int_0^{+\infty} f(x) dx = 0 \quad \text{donc} \quad \frac{2(aw + b\alpha)}{\alpha^2 + w^2} = 0$$

Il en résulte que : $aw = -b\alpha$

Une solution à cette équation est donnée par :

$$a = -c\alpha \quad b = cw$$

où c, w et α des constantes positives. Cette solution permet de diminuer le nombre de paramètres du filtre. La fonction finale s'écrit donc :

$$f(x) = (-c\alpha \sin(w |x|) + cw \cos(w |x|)) e^{-\alpha|x|} \quad (2.2)$$

La forme de cette fonction, présentée figure 2.2, respecte bien celle de la figure 2.1, à une différence d'échelle de représentation près.

Récemment, Petrou et Kittler [Pet 89] ont résolu analytiquement le problème présenté par l'équation 1.4. La réponse impulsionnelle du filtre optimal est similaire à celle que nous avons utilisée (eq. 2.2). La démonstration de similarité entre les deux réponses impulsionnelles est présentée en Annexe B.

Nous avons ainsi mis en évidence la forme analytique du détecteur 1D de contours de type ligne de crête. Nous proposons dans le paragraphe suivant l'extension en 2D de ce détecteur.

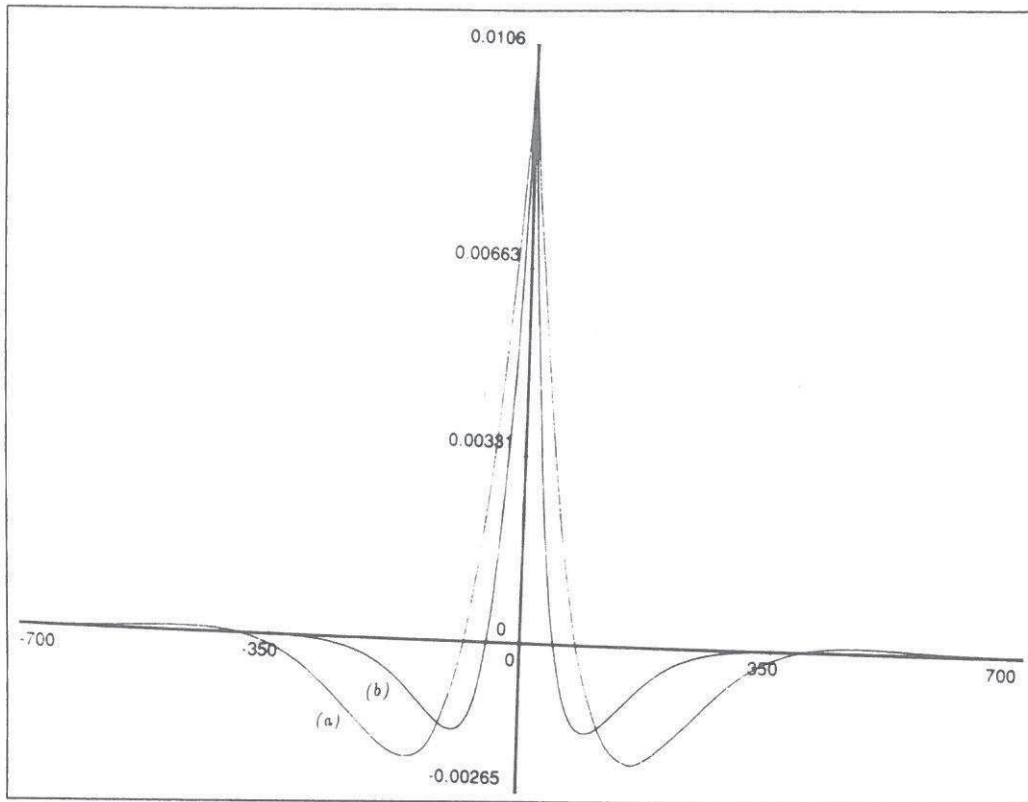


Figure 2.2. Profil de la fonction proposée pour la détection des "lignes de crête", avec différentes valeurs de α et w . (a) $\alpha = w = 0.01$. (b) $\alpha = 0.02, w = 0.01$

2.3 Le détecteur 2D

2.3.1 Extension du filtre en deux dimensions

Pour être capable d'utiliser ce filtre sur des données bidimensionnelles (images), il nous faut étendre le filtre $f(x)$ en deux dimensions. Puisque la fonction $f(x)$ est une bonne approximation de la dérivée seconde d'une Gaussienne (comme on le montre plus loin), son extension en 2D peut être effectuée par l'utilisation de deux filtres séparables et directionnels. Le filtre dans la direction des x (resp. des y) est le résultat du produit de la fonction de détection f (eq. 2.2) dans la direction des x (resp. des y), par une fonction de projection g (eq. 2.3) dans la direction des y (resp. des x). Ces deux filtres sont donnés par :

$$X(x, y) = f(x) g(y) \quad \text{et} \quad Y(x, y) = f(y) g(x)$$

La fonction $f(x)$ est donnée par l'équation 2.2. Il s'agit de déterminer la fonction $g(x)$. Le rôle de cette fonction est de lisser l'image. Il a été montré au paragraphe 3.2 que le filtre donné par l'équation 3.1 permet de lisser une image en régularisant la différentiation. Cependant, ce filtre possède un seul paramètre alors que f en possède deux. Pour des raisons d'homogénéité, nous proposons un filtre similaire à un paramètre près. Ce filtre est défini par :

$$g(x) = \frac{k}{\alpha^2 + w^2} (\alpha \sin(w | x |) + w \cos(w | x |)) e^{-\alpha|x|} \quad (2.3)$$

Cette fonction est une bonne approximation de la gaussienne pour deux raisons : elle vérifie l'équation de la chaleur et possède une forme similaire. Il est facile de vérifier que la fonction f correspond à la dérivée seconde de la fonction g . Par conséquent, f est une bonne approximation de la dérivée seconde de la gaussienne. Il en résulte que le filtre 2D formé de $X(x, y) + Y(x, y)$ est une approximation du laplacien d'une gaussienne.

Le filtre dans la direction des x s'écrit alors :

$$X(x, y) = (-c\alpha \sin(w | x |) + cw \cos(w | x |)) e^{-\alpha|x|} \frac{k}{\alpha^2 + w^2} (\alpha \sin(w | y |) + w \cos(w | y |)) e^{-\alpha|y|} \quad (2.4)$$

Symétriquement, le filtre dans la direction des y s'écrit :

$$Y(x, y) = (-c\alpha \sin(w | y |) + cw \cos(w | y |)) e^{-\alpha|y|} \frac{k}{\alpha^2 + w^2} (\alpha \sin(w | x |) + w \cos(w | x |)) e^{-\alpha|x|} \quad (2.5)$$

La première étape pour mettre en évidence les lignes de crête consiste à convoluer l'image originale par les deux filtres $X(x, y)$ et $Y(x, y)$ et à combiner les deux images résultats pour en former une seule. Il s'agit après cela d'identifier les contours dans cette image. Cette étape est décrite dans le paragraphe suivant.

2.3.2 Identification du contour

Etant données les deux images $I_x(x, y)$ et $I_y(x, y)$, résultats de la convolution de l'image originale avec les filtres $X(x, y)$ et $Y(x, y)$ respectivement, il s'agit de localiser les contours dans ces deux images. La démarche la plus répandue consiste à former une image à partir de $I_x(x, y)$ et $I_y(x, y)$ (où à chaque point sont associées la plausibilité du contour et son orientation) et à identifier ensuite les contours correspondant aux maxima locaux dans cette image. Pour ce faire, nous proposons un modèle mathématique pour calculer l'orientation et nous déduisons ensuite la force du contour.

Définition de l'orientation du contour : l'orientation du contour peut être définie comme étant l'orientation de la courbure maximale. Cependant, il a été montré par ailleurs [Bou 85, SM 88] que, d'une part, la courbure se détériore rapidement en présence du bruit et que, d'autre part, son estimation nécessite un temps de calcul relativement élevé. Nous définissons alors la direction d'un contour comme étant la normale à la direction qui maximise la variance des niveaux de gris de l'image filtrée. Cette direction peut être estimée comme nous le détaillons ci-dessous, à partir du résultat de la convolution de l'image originale avec les filtres 2D définis par les équations 2.4 et 2.5. Plus précisément, soient $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ l'orientation du contour et $I_x(x, y)$, $I_y(x, y)$ les images résultats de la convolution de l'image originale avec les deux filtres $X(x, y)$ et $Y(x, y)$ respectivement. Selon le signe de θ , on distingue deux cas :

1. La réponse du filtre à un contour d'orientation $\theta \in [0, \frac{\pi}{2}]$ est donnée par la formule suivante :

$$I(x, y) * F(x, y) = Ix(x, y) \cos(\theta) + Iy(x, y) \sin(\theta)$$

La moyenne de la réponse du filtre s'écrit :

$$\begin{aligned} E(\theta) &= W(x, y) * I(x, y) * F(x, y) \\ &= W(x, y) * (Ix(x, y) \cos(\theta) + Iy(x, y) \sin(\theta)) \end{aligned}$$

où $W(x, y)$ est une gaussienne telle que :

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} W(x, y) dx dy = 1$$

Pour simplifier la notation, nous remplaçons $Ix(x, y)$ par Ix , $Iy(x, y)$ par Iy et $W(x, y)$ par W . La variance de la réponse du filtre s'écrit sous la forme :

$$V(\theta) = W * (Ix \cos(\theta) + Iy \sin(\theta))^2 - (W * (Ix \cos(\theta) + Iy \sin(\theta)))^2 \quad (2.6)$$

Ou encore,

$$V(\theta) = a \cos^2(\theta) + b \sin(\theta) \cos(\theta) + c \sin^2(\theta) \quad (2.7)$$

$$\begin{aligned} \text{avec } a &= W * I^2x - (W * Ix)^2 \\ b &= 2W * (IxIy) - 2(W * Ix)(W * Iy) \\ c &= W * I^2y - (W * Iy)^2 \end{aligned}$$

Nous pouvons alors déterminer l'orientation du contour θ qui maximise la variance en calculant la première dérivée de la variance et en cherchant l'orientation qui annule cette dérivée :

$$V'(\theta) = (2c - 2a) \sin(\theta) \cos(\theta) + b \cos^2(\theta) - b \sin^2(\theta) = 0 \quad (2.8)$$

On obtient une équation du second degré de la forme :

$$-b \tan^2(\theta) + 2(c - a) \tan(\theta) + b = 0$$

qui a pour solution :

$$\theta = \arctan \left(\frac{a - c \pm \sqrt{(c - a)^2 + b^2}}{-b} \right) \quad (2.9)$$

Puisque $\{(c - a)^2 + b^2\}$ est un terme positif, l'équation du second degré admet toujours une ou deux solutions réelles. Dans le cas de deux solutions, nous retenons l'orientation pour laquelle la seconde dérivée de la variance est négative. Soit θ_1 cette orientation. La seconde dérivée de la variance s'écrit :

$$V''(\theta_1) = 2(c - a) \cos(2\theta_1) + 2b \sin(2\theta_1) < 0$$

D'où

$$\tan(2\theta_1) < \frac{a - c}{b}$$

2. La réponse du filtre à un contour d'orientation $\theta \in [-\frac{\pi}{2}, 0]$ est donnée par la formule :

$$I(x, y) * F(x, y) = I_x(x, y) \cos(\theta) - I_y(x, y) \sin(\theta)$$

En suivant la même démarche que dans le cas précédent on peut montrer aisément que l'orientation du contour est égale à θ_2 telle que $\theta_2 = -\theta_1$.

En résumé, nous avons deux orientations candidates θ_1 et $-\theta_1$ pour chaque point de contours. Un point de l'image est marqué contour lorsqu'il correspond au maximum local dans la direction θ_1 ou dans la direction $-\theta_1$. L'expérimentation a montré que l'utilisation de cette règle pour marquer les contours a pour conséquence des réponses multiples dans le voisinage d'un vrai point de contours. Pour pallier à ce problème, nous proposons de sélectionner l'une des deux orientations (c'est-à-dire θ_1 ou $-\theta_1$) telle que la dérivée seconde directionnelle de l'image de variance (l'image où chaque pixel a pour valeur la variance du contour) soit négative. En d'autres termes, nous retenons l'orientation qui maximise l'image de variance.

Définition de la plausibilité du contour : à partir de l'équation 2.8, on peut écrire :

$$2\theta = \arctan\left(\frac{b}{a - c}\right)$$

la variance du contour peut être mesurée par le module du vecteur $(b, a - c)$, soit :

$$M = \sqrt{b^2 + (a - c)^2}$$

L'expérimentation a montré que les valeurs de M obtenues à l'aide de cette formule sont très petites et qu'il est très difficile de distinguer des valeurs proches lorsqu'on travaille dans un domaine discret. La formule utilisée pour calculer la plausibilité est :

$$M = \sqrt{4(W * (IxIy))^2 + (W * I^2x - W * I^2y)^2} \quad (2.10)$$

Cette formule peut être obtenue en remplaçant la variance (eq. 2.7) par la moyenne de la réponse du détecteur au carré et en suivant la même démarche.

En résumé, l'identification du contour est effectuée en utilisant comme mesure de plausibilité la variance donnée par l'équation 2.10. En ce qui concerne l'orientation du contour, nous avons quatre directions candidates : les deux orientations θ_1 et θ_2 données par l'équation 2.9 et les deux autres qui sont obtenues par symétrie $-\theta_1$ et $-\theta_2$. Nous retenons celle pour laquelle les secondes dérivées de la variance (eq 2.7) et de l'image de variance (eq. 2.10) soient toutes les deux négatives.

2.4 Implantation du détecteur 2D

La plupart des filtres résultats de l'étape de conception sont continus et possèdent un support infini. Lors de l'implantation, ces filtres sont souvent coupés pour être approchés par des filtres numériques à support fini. Cette coupure a pour conséquence l'effet de Gibbs qui est équivalent à l'introduction du bruit dans leurs réponses. L'utilisation des filtres à réponse impulsionnelle infinie (RII) permet d'éviter ce phénomène et minimise la perte d'informations provoquée par cette coupure. Elle possède aussi d'autres avantages : efficacité, préservation des propriétés du filtre initial, etc. Nous avons présenté dans [Zio 91a] une étude plus complète concernant les avantages de cette méthode d'implantation.

Rappelons qu'un filtre linéaire RII est régi par une équation aux différences d'ordre M de la forme suivante :

$$\sum_{m=0}^M b_m y(k-m) = \sum_{n=0}^N a_n x(k-n) \quad (2.11)$$

où $y(n)$ est le signal en sortie et $x(n)$ est le signal en entrée. Notons que pour qu'un filtre RII puisse être implanté par des équations aux différences, il lui faut être stable et causal. Rappelons qu'un filtre $f(n)$ est dit stable si $\sum_{n=-\infty}^{+\infty} |f(n)| < \infty$. Il est dit causal si $f(n) = 0$ pour tout $n < 0$.

Les deux filtres donnés par les équations 2.4 et 2.5 sont séparables. Ils peuvent être implantés alors par des équations aux différences 1D. Il en résulte que le problème d'implantation des deux filtres 2D est transformé en un problème d'implantation des deux filtres 1D : la fonction de détection f et la fonction de projection g .

2.4.1 Représentation de la fonction de détection

L'implantation d'un filtre RII par des équations aux différences revient à estimer les coefficients a_n et b_n (eq. 2.11). Pour ce faire, nous proposons de discrétiser le filtre f par une simple substitution et de transformer en Z le filtre discret. Plus précisément, considérons la fonction discrète $f(n)$ obtenue par échantillonnage de $f(x)$. Sa transformée en Z s'écrit :

$$t(z) = \sum_{n=-\infty}^{+\infty} f(n)z^{-n}$$

Le filtre $f(n)$ est non causal. Pour pouvoir le réaliser nous adoptons la démarche qui consiste à décomposer le filtre $f(n)$ en deux autres filtres $f_-(n)$ et $f_+(n)$, réalisés chacun séparément et enfin à combiner leurs résultats. Plus précisément :

$$t(z) = \sum_{n=-\infty}^{-1} f_-(n)z^{-n} + \sum_{n=0}^{+\infty} f_+(n)z^{-n}$$

avec :

$$f_-(n) = \begin{cases} 0 & \text{si } n > 0 \\ c(\alpha \sin(wn) + w \cos(wn))e^{\alpha n} & \text{si } n < 0 \end{cases}$$

$$f_+(n) = \begin{cases} c(-\alpha \sin(wn) + w \cos(wn))e^{-\alpha n} & \text{si } n > 0 \\ 0 & \text{si } n < 0 \end{cases}$$

Le filtre $f_+(n)$ est causal et peut donc être réalisé par des équations aux différences. Sa fonction de transfert s'écrit alors sous la forme :

$$t_+(z) = \frac{a_0 + a_1 z^{-1}}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

En revanche, le filtre $f_-(n)$ n'est pas causal. Afin de forcer cette propriété, nous considérons qu'il correspond au parcours de l'image de la droite vers la gauche. Sa fonction de transfert devient alors :

$$t_-(z) = \frac{a_2 z + a_3 z^2}{1 + b_1 z + b_2 z^2}$$

avec $a_0 = cw$

$$a_1 = -c(w \cos(w) + \alpha \sin(w))e^{-\alpha}$$

$$a_2 = a_1 - a_0 b_1$$

$$a_3 = -a_0 b_2$$

$$b_1 = -2e^{-\alpha} \cos(w)$$

$$b_2 = e^{-2\alpha}$$

Examinons à présent la stabilité du filtre $f(n)$:

$$\begin{aligned} \sum_{n=-\infty}^{+\infty} |f(n)| &= \sum_{n=-\infty}^{+\infty} \left| (-c\alpha \sin(w|n|) + cw \cos(w|n|))e^{-\alpha|n|} \right| \\ &\leq (c\alpha + cw) \sum_{n=-\infty}^{+\infty} e^{-\alpha|n|} = \frac{c(1 + e^{-\alpha})(\alpha + w)}{1 - e^{-\alpha}} \end{aligned}$$

Nous venons de montrer que la somme des valeurs absolues de la fonction est finie pour α strictement positif, ce qui assure la stabilité du filtre. Les équations aux différences deviennent :

$$y_+(n) = a_0 x(n) + a_1 x(n-1) - b_1 y_+(n-1) - b_2 y_+(n-2)$$

$$y_-(n) = a_2 x(n+1) + a_3 x(n+2) - b_1 y_-(n+1) - b_2 y_-(n+2)$$

$$y(n) = y_-(n) + y_+(n)$$

La valeur de c est obtenue par la normalisation de la fonction de détection c'est-à-dire :

$$\sum_{n=-\infty}^{+\infty} f(n) = 1$$

$$c = \frac{1 + b_1 + b_2}{w - we^{-2\alpha} - 2\alpha \sin(w)e^{-\alpha}}$$

2.4.2 Représentation de la fonction de projection

En suivant le même raisonnement pour la fonction de projection (eq. 2.3), on obtient les deux fonctions de transfert suivantes :

$$t_-(z) = \frac{a_6z + a_7z^2}{1 + b_1z + b_2z^2} \quad \text{et} \quad t_+(z) = \frac{a_4 + a_5z^{-1}}{1 + b_1z^{-1} + b_2z^{-2}}$$

Les équations aux différences s'écrivent alors :

$$\begin{aligned} y_+(n) &= a_4x(n) + a_5x(n-1) - b_1y_+(n-1) - b_2y_+(n-2) \\ y_-(n) &= a_6x(n+1) + a_7x(n+2) - b_1y_-(n+1) - b_2y_-(n+2) \\ y(n) &= y_-(n) + y_+(n) \end{aligned}$$

avec $a_4 = c_2$

$$a_5 = (-c_2 \cos(w) + c_1 \sin(w))e^{-\alpha}$$

$$a_6 = a_1 - c_2b_1$$

$$a_7 = -c_2b_2$$

$$b_1 = -2e^{-\alpha} \cos(w)$$

$$b_2 = e^{-2\alpha}$$

$$c_1 = \frac{k\alpha}{\alpha^2 + w^2}$$

$$c_2 = \frac{k w}{\alpha^2 + w^2}$$

A partir de la normalisation de la fonction de projection on obtient :

$$k = \frac{(1 - 2e^{-\alpha} \cos(w) + 2e^{-2\alpha})(\alpha^2 + w^2)}{2\alpha e^{-\alpha} \sin(w) + w - w e^{-2\alpha}}$$

La stabilité de ce filtre est aussi assurée pour α strictement positif.

2.5 Résumé de l'algorithme de détection des lignes de crête

L'algorithme implanté procède en quatre étapes :

- Filtrage d'une image I de l lignes et c colonnes, dans la direction horizontale en utilisant l'opérateur séparable donné en (eq. 2.4). L'image I est filtrée dans la direction horizontale à l'aide des équations récursives représentant la fonction de détection et dans la direction verticale par les équations récursives représentant la fonction de projection. Soient D et P deux images intermédiaires de calcul, et I_x l'image résultat, l'opération de filtrage se passe comme suit :

$$\begin{aligned} D_+(i, j) &= a_0I(i, j) + a_1I(i, j-1) - b_1D_+(i, j-1) - b_2D_+(i, j-2) \\ i &= 1, 2, \dots, l \quad j = 3, 4, \dots, c \end{aligned}$$

$$\begin{aligned} D_-(i, j) &= a_2I(i, j+1) + a_3I(i, j+2) - b_1D_-(i, j+1) - b_2D_-(i, j+2) \\ i &= 1, 2, \dots, l \quad j = c-2, \dots, 1 \end{aligned}$$

$$\begin{aligned} D(i, j) &= D_-(i, j) + D_+(i, j) \\ i &= 1, 2, \dots, l \quad j = 1, 2, \dots, c \end{aligned}$$

$$P_+(i, j) = a_4 D(i, j) + a_5 D(i-1, j) - b_1 P_+(i-1, j) - b_2 P_+(i-2, j)$$

$$i = 3, 4, \dots, l \quad j = 1, 2, \dots, c$$

$$P_-(i, j) = a_6 D(i+1, j) + a_7 D(i+2, j) - b_1 P_-(i+1, j) - b_2 P_-(i+2, j)$$

$$i = l-2, \dots, 1 \quad j = 1, 2, \dots, c$$

$$Ix(i, j) = P_-(i, j) + P_+(i, j)$$

$$i = 1, 2, \dots, l \quad j = 1, 2, \dots, c$$

- Filtrage symétrique de l'image dans la direction verticale à l'aide de (eq. 2.5). Soit I_y l'image résultat.
- Suppression des non-maxima dans la direction qui maximise la variance (eq. 2.9). La règle de choix de la meilleure orientation est donnée dans la section 2.3.2. L'algorithme de suppression des non-maxima est donné dans [Can 86].
- Normalisation de l'image formée dans l'étape précédente, où à chaque point est associée la valeur du module M (eq.2.10). En effet, les maxima de cette image correspondent aux points de contours recherchés. Certains de ces points sont significatifs alors qu'ils ont une variance inférieure à d'autres points non significatifs (fig. 2.3). Il est donc nécessaire d'extraire les pics significatifs et d'éliminer les zones régulières de l'image. Pour ce faire, on transforme cette image en une autre image, par une technique de morphologie mathématique [Ser 82] : la transformation dite du "chapeau haut-de-forme".

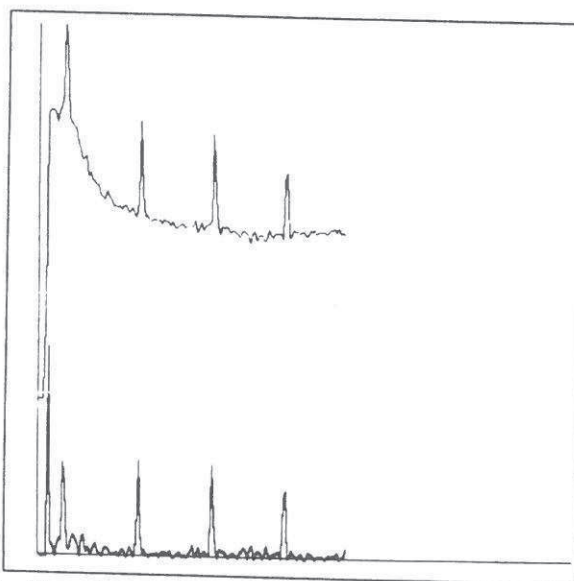


Figure 2.3. Transformation chapeau haut-de-forme d'un signal 1D.

2.6 Evaluation des performances du détecteur

2.6.1 Performances du détecteur proposé

L'évaluation des performances du filtre proposé en utilisant les critères définis par Canny conduit à calculer le rapport signal/bruit, la localisation et le nombre de points

marqués contours en réponse à un seul point de contours réel. Nous présentons ci-après une discussion sur les performances du détecteur (eq. 2.2) en fonction de ses paramètres.

- **Le rapport signal/bruit** : le calcul du numérateur et du dénominateur de l'équation 1.2 donne :

$$\int_{-\infty}^{+\infty} f(x)F(-x)dx = 2 \int_0^{+\infty} f(x)F(-x)dx = \frac{2Acw}{(\alpha + 1)^2 + w^2}$$

$$\int_{-\infty}^{+\infty} f^2(x)dx = 2 \int_0^{+\infty} f^2(x)dx = \frac{c^2w^2}{2\alpha}$$

d'où :

$$\Sigma(f) = \frac{2\sqrt{2\alpha}}{(\alpha + 1)^2 + w^2} \quad (2.12)$$

- **La localisation** : d'après la définition de la localisation donnée par l'équation 1.3, nous devons tout d'abord évaluer les deux premières dérivées de la fonction f .

$$f'(x) = \begin{cases} c((\alpha^2 - w^2) \sin(wx) - 2\alpha w \cos(wx))e^{-\alpha x} & \text{si } x > 0 \\ c((\alpha^2 - w^2) \sin(wx) + 2\alpha w \cos(wx))e^{\alpha x} & \text{si } x < 0 \end{cases}$$

La fonction $f(x)$ n'est pas dérivable au point 0, car :

$$f'(x) = \begin{cases} -2c\alpha w & \text{si } x = 0^+ \\ 2c\alpha w & \text{si } x = 0^- \end{cases}$$

$$f''(x) = \begin{cases} c((3\alpha w^2 - \alpha^3) \sin(wx) + (3\alpha^2 w - w^3) \cos(wx))e^{-\alpha x} & \text{si } x > 0 \\ c((\alpha^3 - 3\alpha w^2) \sin(wx) + (3\alpha^2 w - w^3) \cos(wx))e^{\alpha x} & \text{si } x < 0 \end{cases}$$

Afin de conserver $\int_{-\infty}^x f''(x)dx = f'(x)$, $f''(x)$ peut s'écrire sous la forme suivante :

$$f''(x) = c((3\alpha w^2 - \alpha^3) \sin(w|x|) + (3\alpha^2 w - w^3) \cos(w|x|))e^{-\alpha|x|} + 4c\alpha w \delta(x)$$

où $\delta(x)$ est la distribution de Dirac (cf. Généralité) L'évaluation de l'équation 1.3 donne :

$$\int_{-\infty}^{+\infty} F(-x)f''(x)dx = 2Ac \left(\frac{w^3(2\alpha - 1) + \alpha^2 w(2\alpha + 3)}{(\alpha + 1)^2 + w^2} + \alpha w \right)$$

$$\int_{-\infty}^{+\infty} f'^2(x)dx = \frac{c^2w^2(5\alpha^2 + w^2)}{2\alpha^2}$$

$$\Lambda(f) = 2\sqrt{2\alpha} \left(\frac{w^2(2\alpha - 1) + \alpha^2(2\alpha + 3)}{(\alpha + 1)^2 + w^2} + \alpha \right) \frac{1}{\sqrt{5\alpha^2 + w^2}} \quad (2.13)$$

- **Réponses multiples** : on entend par réponse multiple d'un détecteur le fait que celui-ci fournit plusieurs points de contours à proximité d'un seul point de contours réel. Ces points sont généralement proches les uns des autres. Il est donc difficile de sélectionner les "vrais" des "faux" points. D'où la nécessité de caractériser un détecteur par le nombre de "faux" points en réponse à un point de contours. Pour cela, on introduit une mesure permettant d'évaluer la dispersion des points de contours. Cette mesure est exprimée par la distance moyenne k entre les pics contenus dans la réponse du filtre à un bruit blanc. Elle est donnée par [Can 83] :

$$k = \sqrt{\frac{\int_{-\infty}^{+\infty} f^2(x) dx}{\int_{-\infty}^{+\infty} f'^2(x) dx}} \quad (2.14)$$

Or,

$$\int_{-\infty}^{+\infty} f^2(x) dx = \frac{c^2 w^2}{2\alpha}$$

k s'écrit alors :

$$k = \sqrt{\frac{\alpha}{5\alpha^2 + w^2}} \quad (2.15)$$

2.6.2 Choix des paramètres du détecteur

Les mesures de performances (2.12, 2.13, 2.15) dépendent des deux paramètres α et w . Nous discutons ci-après l'influence de ces paramètres sur les performances du détecteur pour en déduire l'ensemble de leurs valeurs possibles.

- **Influence des paramètres sur le critère de performances $C(f)$ (eq. 1.4)** : on distingue trois cas possibles :

$$\begin{aligned} 1. \alpha \gg w : \quad \Sigma(f) &\simeq 2\sqrt{2} \frac{\sqrt{\alpha}}{(\alpha+1)^2}, & \Lambda(f) &\simeq \frac{2\sqrt{2}}{\sqrt{5}} \frac{3\alpha^3 + 5\alpha^2 + \alpha}{(\alpha+1)^2}, \\ C(f) = \Sigma(f)\Lambda(f) &\simeq \frac{8}{\sqrt{5}} \frac{\alpha}{(\alpha+1)^4} (3\alpha^3 + 5\alpha^2 + \alpha), & k &\simeq \frac{1}{\sqrt{5\alpha}} \end{aligned}$$

$$\begin{aligned} 2. \alpha \ll w : \quad \Sigma(f) &\simeq 2\sqrt{2} \frac{\sqrt{\alpha}}{w^2 + 2\alpha + 1}, & \Lambda(f) &\simeq 2\sqrt{2} \frac{\alpha}{w} \frac{w^2(3\alpha-1) + \alpha}{w^2 + 2\alpha + 1}, \\ C(f) = \Sigma(f)\Lambda(f) &\simeq 8 \frac{\alpha}{w} \sqrt{\alpha} \frac{w^2(3\alpha-1) + \alpha}{(w^2 + 2\alpha + 1)^2}, & k &\simeq \sqrt{\frac{\alpha}{w^2}} \end{aligned}$$

$$\begin{aligned} 3. \alpha \simeq w : \quad \Sigma(f) &\simeq 2\sqrt{2} \frac{\sqrt{\alpha}}{(\alpha+1)^2 + \alpha^2}, & \Lambda(f) &\simeq \frac{2\sqrt{2}}{\sqrt{6}} \frac{6\alpha^3 + 4\alpha^2 + \alpha}{(\alpha+1)^2 + \alpha^2}, \\ C(f) = \Sigma(f)\Lambda(f) &\simeq \frac{8}{\sqrt{6}} \sqrt{\alpha} \frac{6\alpha^3 + 4\alpha^2 + \alpha}{((\alpha+1)^2 + \alpha^2)^2}, & k &\simeq \frac{1}{\sqrt{6\alpha}} \end{aligned}$$

Le rapport signal/bruit estimé dans le premier cas ($\alpha \gg w$) est supérieur à celui des deux autres estimations. Il en est de même pour la localisation et la dispersion, ainsi que pour le critère de performances.

- **Influence des paramètres sur la densité des contours détectés** : les points de passage par zéro de la dérivée première du signal qui correspondent aux points de passage par zéro de $f'(x)$, marquent chacun un maxima de $f(x)$, c'est-à-dire un point de contours. Le nombre de faux points de contours est alors proportionnel au nombre de points maxima de la fonction $f(x)$.

Examinons à présent $f(x)$ pour $x > 0$ (l'autre cas est symétrique). Les abscisses des points critiques de la fonction $f(x)$ sont :

$$x_0 = 0 \quad \text{et} \quad x_k = \frac{1}{w} \arctan \left(\frac{2\alpha w}{\alpha^2 + w^2} \right) + \frac{(k-1)\pi}{w} \quad k = 1, \dots, n+1$$

La distance entre deux maxima consécutifs, sauf entre le premier et le deuxième, ne dépend que de w . Elle est égale à :

$$|x_{k+1} - x_k| = \frac{2k\pi}{w}, \quad \forall k \neq 0$$

Cette distance est une fonction décroissante de w . Par exemple, si w est petit, la distance entre deux maxima devient grande. Il en résulte que le nombre de maxima de la fonction $f(x)$, soit le nombre de points de contours détectés, diminue.

En résumé, le choix de $\alpha \geq w$ permet d'obtenir un bon compromis entre le critère de performances et la dispersion. Etant donné un α , il suffit d'ajuster le w pour obtenir le nombre désiré de points de contours. Notons que la valeur de α doit être strictement positive pour avoir une stabilité du filtre.

2.6.3 Complexité de l'algorithme

L'opération de filtrage que nous proposons requiert 28 additions et 32 multiplications. Soit 60 opérations élémentaires. Le temps de calcul pour une image 256×256 sur un Convex est de l'ordre de 40 secondes. Une comparaison de cette méthode avec d'autres méthodes telle que la transformée de Fourier rapide (FFT) et les masques de convolution est présentée dans [Zio 91a] et peut être résumée par :

- le filtrage récursif d'ordre 2 est plus rapide que les masques de convolution et que la FFT,
- le filtrage récursif est plus économique en nombre de références en mémoire que la FFT et les masques de convolution,
- Le filtrage récursif est la seule méthode parmi les trois dont la complexité temporelle et spatiale n'est pas affectée par le changement d'échelle.

Le bilan de l'algorithme de suppression des non-maxima est coûteux si on le compare à celui utilisé dans le cas des contours de type marche. Cependant, le problème ici est plus complexe, car il n'y a pas de formule mathématique connue comme dans le cas de contours de type marche pour calculer l'orientation et la plausibilité.

2.6.4 Résultats expérimentaux

La détection de crête

Le modèle de contours utilisé pour dériver le détecteur (eq. 2.2) correspond au profil d'un contour de type toit (2.1). Nous avons défini une ligne de crête comme étant à la fois un contour de type toit et un contour de type crête. Si nous sommes certains que ce détecteur met en évidence les contours de type toit, il n'en est pas de même pour les contours de type crête. Pour le vérifier, considérons un contour 1D de type crête dont le plateau est de longueur égale à 8 pixels (fig. 2.4.a). La réponse de notre détecteur est donnée par la figure 2.4.b : elle correspond à un pic. Les contours de type crête sont ainsi pris en considération par notre détecteur. Il en résulte que le détecteur est donc conforme à la spécification initiale.

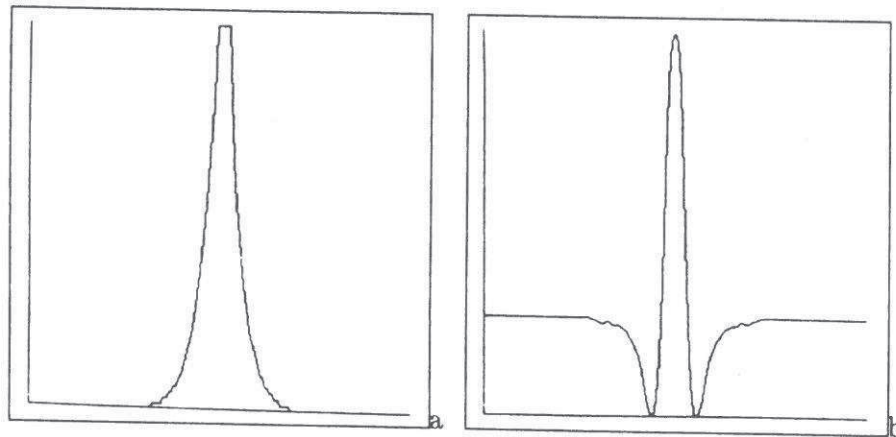


Figure 2.4. (a) Profil d'un contour de type crête. (b) La réponse du détecteur.

Images naturelles

La mesure des performances de notre détecteur est uniquement fondée sur les trois propriétés décrites dans la section 2.6.1. De plus, les hypothèses utilisées pour dériver ces détecteurs sont très simplistes par rapport à la complexité des images réelles. Une évaluation plus sensée consiste à estimer les résultats expérimentaux obtenus sur des images naturelles, en tenant compte du contexte dans lequel les détecteurs sont utilisés et de leur intégration dans un système de vision. Cette évaluation ne pouvant être faite de cette manière, nous nous contenterons d'une évaluation subjective des résultats obtenus.

Nous avons expérimenté notre détecteur sur un échantillon varié d'images à niveaux de gris. Nous en présentons ici deux exemples. La figure 2.5.a représente une image originale de 256×256 points et de 256 niveaux de gris. La scène contient des objets polyédriques posés sur une mire. Cette image contient des lignes de crête correspondant au tracé de la mire et aux différences d'orientation des surfaces. Nous présentons dans la figure 2.5.b) la réponse du détecteur optimal de contours de type marche de Deriche (Cf. 1.2) et dans (fig. 2.5.c) la réponse de notre détecteur optimal de lignes de crête. La figure 2.5.d illustre la réponse de notre détecteur sur le négatif de l'image originale. Nous avons également extrait les contours lignes de crête (fig. 2.6.b et 2.6.c) pour une image de 256×256 points

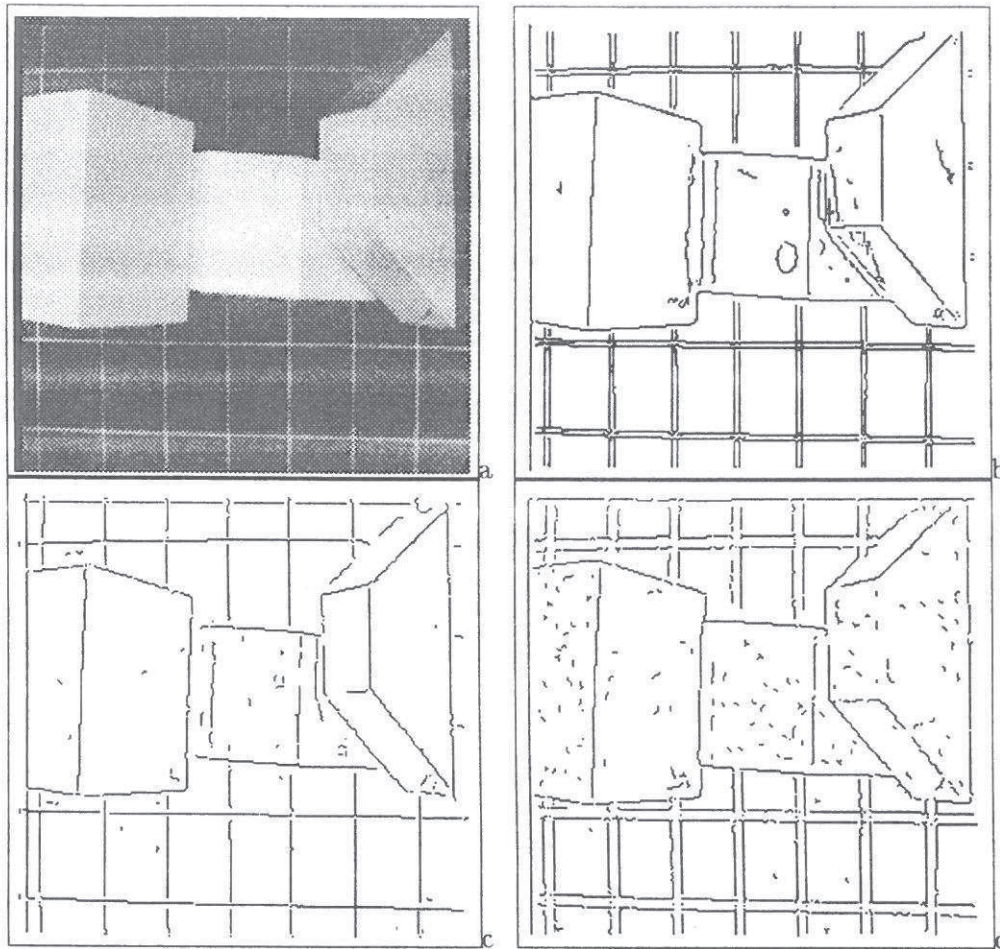


Figure 2.5. (a) Image originale (256×256). (b) Contours de type marche [Der 87]. (c) Contours de type lignes ($\alpha = 1, w = 0.75$). (d) Contours de type lignes obtenus à partir du négatif de l'image originale ($\alpha = 1., w = 0.75$).

et de 256 niveaux de gris (fig. 2.6.a). Cette image représente un test difficile pour un détecteur. En effet, les contours sont proches les uns des autres. En conséquence, la détection de l'un des contours est influencée par les contours voisins.

2.6.5 Discussion

Un examen des images présentées dans la section précédente montre l'existence de phénomènes qui doivent être pris en considération lors de la conception et de l'utilisation d'un détecteur de lignes de crête. Plus précisément, dans le voisinage d'un pic (resp. une vallée), il existe souvent deux vallées (resp. deux pics) comme le montre la figure 2.3. Si les pics et les vallées sont détectés en même temps, le détecteur répondra à une simple ligne de crête par trois lignes. Pour éviter cette redondance, un détecteur doit localiser, soit les pics, soit les vallées. Les figures 2.5.c et 2.5.d illustrent d'une part ce phénomène, et

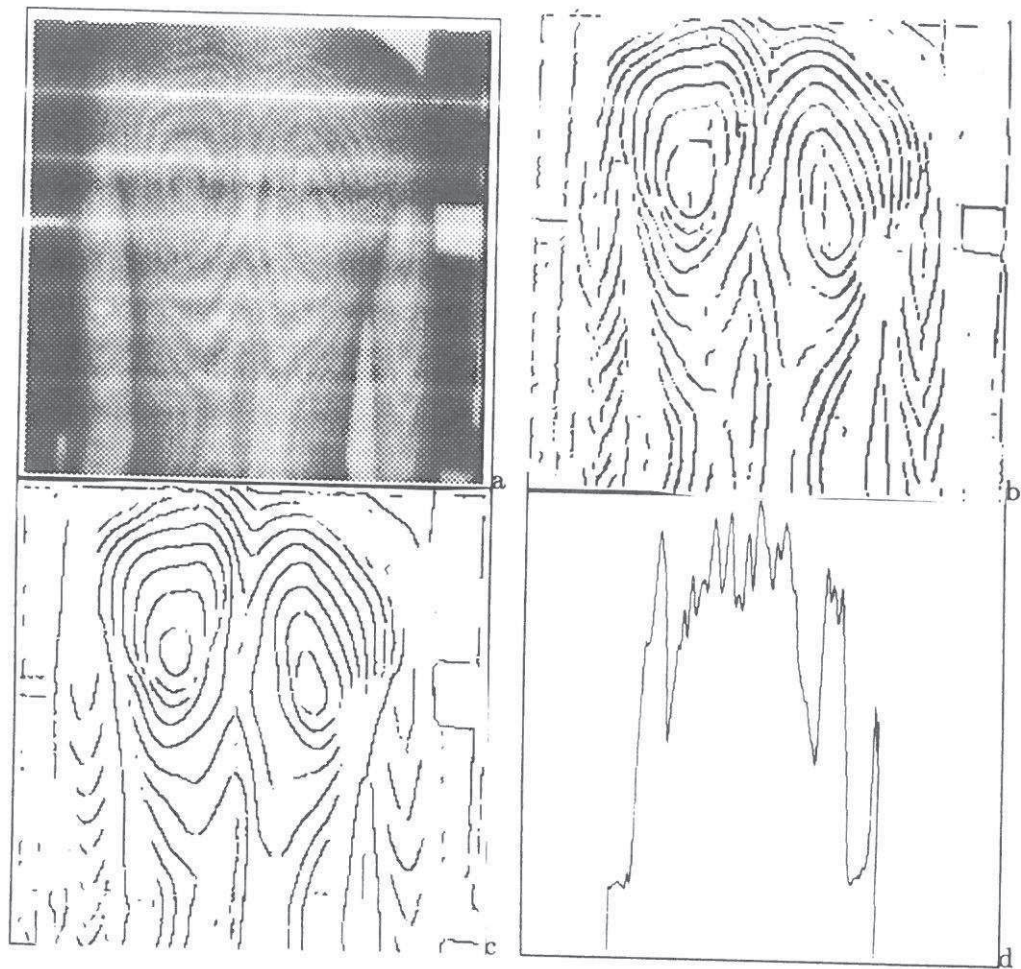


Figure 2.6. (a) Image originale (256×256). (b) Contours de type lignes ($\alpha = 1, w = 0.75$). (c) Contours de type lignes obtenus à partir du négatif de l'image originale ($\alpha = 1, w = 0.75$). (d) Coupe de l'image donnée en (a)

d'autre part la détection des vallées seules par notre détecteur. De plus, dans le voisinage d'une marche, il existe souvent un pic et une vallée comme le montre la figure 2.6.d. Pour cette raison, certains contours sont localisés par un détecteur de marche et par le détecteur de lignes de crête comme le montre les figures 2.5.c et 2.5.b. En réalité, ces deux contours ne correspondent pas au même phénomène physique et ils ne coïncident pas dans l'espace image. Ils peuvent être alors identifiés facilement.

2.7 Conclusion

Les informations fournies par un détecteur de lignes de crête sont complètement différentes de celles fournies par un détecteur de contours de type marche. Elles représentent d'autres aspects d'une scène [Hor 77, Bin 81]. La réponse d'un détecteur de marches à une ligne de crête se présente sous la forme de deux lignes parallèles et voisines comme

le montre l'image (fig. 2.5.b), alors que le détecteur de lignes de crête ne fournit qu'un seul contour. L'utilisation d'un détecteur de lignes de crête permet alors la réduction du nombre de primitives manipulées par un système de vision.

En utilisant les critères de Canny, nous avons dérivé un détecteur optimal de lignes de crête, qui localise à la fois les toits et les crêtes. Le détecteur 2D résultant est séparable. Il est alors implanté efficacement par des équations aux différences. Nous avons présenté un modèle mathématique pour l'identification des lignes de crête. Nous avons aussi analysé les performances du détecteur, ce qui nous a conduit à proposer des règles de sélection des meilleurs valeurs pour ses paramètres. Une évaluation plus complète de cette algorithme peut être trouvée dans [Zio 91a]. Les résultats expérimentaux de ce détecteur sont satisfaisants.

2.8 Annexe A

L'objectif de cette annexe est de montrer la nécessité d'utiliser un filtre symétrique pour la détection des contours de type lignes de crête.

Soit $S(x)$ la réponse du filtre au signal d'entrée non bruité, toute fonction $f(x)$ se décompose en : $f(x) = f_1(x) + f_2(x)$ où $f_1(x)$ est impaire et $f_2(x)$ est paire. La réponse du filtre s'écrit alors :

$$S(x) = F(x) * f(x)$$

où $*$ est le symbole de la convolution.

La dérivée de S s'écrit :

$$\begin{aligned} \frac{dS(x)}{dx} &= \frac{d}{dx}(F(x) * f(x)) \\ &= F(x) * (f_1'(x) + f_2'(x)) \\ &= F(x) * f_1'(x) + F(x) * f_2'(x) \end{aligned}$$

Cette dérivée doit être nulle à l'origine :

$$\left. \frac{dS(x)}{dx} \right|_{x=0} = \int_{-\infty}^{+\infty} F(x) f_1'(x) dx + \int_{-\infty}^{+\infty} F(x) f_2'(x) dx$$

Comme les fonctions $F(x)$ et $f_2(x)$ sont paires, nous avons :

$$\int_{-\infty}^{+\infty} F(x) f_2'(x) dx = 0$$

Pour que $\left. \frac{dS(x)}{dx} \right|_{x=0} = 0$, il faut que :

$$\int_{-\infty}^{+\infty} F(x) f_1'(x) dx = 0$$

Comme la fonction $F(x)$ est paire, alors $f_1'(x) = 0$, et donc $f_1(x) = C$, où C est une constante.

Examinons le rapport signal/bruit :

$$\begin{aligned}\int_{-\infty}^{+\infty} F(x)f(x)dx &= \int_{-\infty}^{+\infty} F(x)f_1(x)dx + \int_{-\infty}^{+\infty} F(x)f_2(x)dx \\ &= \int_{-\infty}^{+\infty} F(x)f_2(x)dx\end{aligned}$$

Soit :

$$\int_{-\infty}^{+\infty} f^2(x)dx = \int_{-\infty}^{+\infty} f_1^2(x)dx + 2 \int_{-\infty}^{+\infty} f_1(x)f_2(x)dx + \int_{-\infty}^{+\infty} f_2^2(x)dx$$

On a :

$$\int_{-\infty}^{+\infty} f_1^2(x)dx \geq 0 \quad (2.16)$$

$$\int_{-\infty}^{+\infty} f_1(x)f_2(x)dx = 0$$

Le terme (eq. 2.16) n'a aucun intérêt, au contraire il fait diminuer le rapport signal/bruit. Nous posons alors $f_1(x) = 0$.

2.9 Annexe B

L'objectif de cette annexe est de montrer la similarité entre notre filtre 1D et celui proposé récemment par Petrou et Kittler.

Petrou et Kittler [Pet 89] considèrent que le contour a pour profil :

$$F(x) = H e^{-s|x|} \quad (2.17)$$

Ils démontrent analytiquement que le filtre permettant de maximiser le produit des trois critères de Canny (équations 1.2, 1.3 et 2.14) a pour réponse impulsionnelle :

$$f(x) = (-k_1 \sin(ax) + k_2 \cos(ax))e^{-ax} + (-k_3 \sin(ax) + k_4 \cos(ax))e^{ax} + b e^{-sx} + k_5 \quad (2.18)$$

Pour $w \geq x \geq 0$ et avec les contraintes données par :

$$f(0) = k \quad (2.19)$$

$$f'(0) = 0 \quad (2.20)$$

$$f(\pm w) = 0 \quad (2.21)$$

$$f'(\pm w) = 0 \quad (2.22)$$

$$\int_{-w}^{+w} f(x)dx = 0 \quad (2.23)$$

Où k est l'amplitude du filtre et w la moitié de la largeur de la bande du filtre.

Considérons le cas où $w \rightarrow \infty$. Puisque $a > 0$ et $s > 0$, on obtient en utilisant la contrainte donnée par l'équation 2.21 :

$$f(x) = (-k_1 \sin(ax) + k_2 \cos(ax))e^{-ax} + be^{-sx} \quad (2.24)$$

Pour rendre le filtre indépendant du signal, on pose $b = 0$, on obtient alors :

$$f(x) = (-k_1 \sin(ax) + k_2 \cos(ax))e^{-ax} \quad (2.25)$$

En utilisant la contrainte (eq. 2.23), puis la contrainte (eq. 2.19), on trouve :

$$k_1 = k_2 = k \quad (2.26)$$

Finalement, la réponse impulsionnelle du filtre optimal devient :

$$f(x) = (-k \sin(ax) + k \cos(ax))e^{-ax} \quad (2.27)$$

Puisque $f(x)$ doit être paire (la démonstration est donnée en Annexe A), cela nous permet d'écrire :

$$f(x) = (-k \sin(a | x |) + k \cos(a | x |))e^{-a|x|} \quad (2.28)$$

Cette dernière fonction vérifie toutes les contraintes ci-dessus, sauf la contrainte donnée par l'équation 2.20. En effet, la dérivée première de cette fonction est discontinue à l'origine, avec un passage par zéro à l'origine. Cette discontinuité ne forme pas un handicap, en revanche elle permet d'augmenter la localisation du détecteur. Remarquons que cette fonction (eq. 2.18) est exactement celle que nous avons utilisée (pour $\alpha = w$) et qui est donnée par l'équation 2.2.

3

Détecteur de contours à partir de l'histogramme

L'objectif de ce chapitre est de décrire un algorithme de seuillage d'une image à niveaux de gris fondé sur l'apprentissage numérique. Cet algorithme est utilisé pour la détection de contours fermés dans des images comportant quelques objets posés sur un fond.

Nous exposons dans les premiers paragraphes les différentes étapes de l'algorithme, à savoir, l'approximation de l'histogramme par une combinaison linéaire de gaussiennes et l'estimation des seuils minimisant l'erreur bayésienne. Un résumé de l'algorithme est présenté ensuite. Enfin, nous décrivons une mesure de performances qui permet de comparer notre algorithme à d'autres.

3.1 Introduction

Dans certaines applications de la vision, nous avons à traiter des images contenant un ou plusieurs objets placés sur un fond. Si les distributions des niveaux de gris des points de surface des objets et du fond sont de faible variance et de moyenne différente, alors l'histogramme de l'image est multimodale, c'est-à-dire qu'il comporte plusieurs modes distincts. Etant donnée cette caractéristique, la dualité entre la détection de contours et l'extraction de régions semble être vérifiée dans le sens où il existe une coïncidence spatiale entre les frontières des régions et les contours. Les méthodes pour segmenter ces images peuvent être simples et rapides. La méthode la plus connue est le multi-seuillage. Son principe est simple : il s'agit de partitionner l'image $I(x, y)$ en un nombre de régions égal au nombre de modes de l'histogramme suivant la règle :

$$R(x, y) = \begin{cases} N - 1 & \text{si } I(x, y) > T_{N-1} \\ N - 2 & \text{si } T_{N-2} < I(x, y) \leq T_{N-1} \\ \cdot & \\ \cdot & \\ 0 & \text{si } I(x, y) < T_1 \end{cases}$$

Deux seuils consécutifs T_i et T_{i+1} limitent un mode. Le problème de segmentation, dans ce cas bien précis, se limite alors au problème d'estimation de ces seuils. La littérature est abondante sur le sujet [Sah 88]. Les algorithmes diffèrent essentiellement par la fonction utilisée pour représenter l'image (distribution de niveaux de gris [Pre 66, Ost 78, Tsa 85, Kit 86], fonction de répartition [Bou 85], matrice de co-occurrence [Ahu 78], etc.),

le domaine d'application (histogramme bimodal [Pun 81, Kit 86] ou multimodal [Bou 85]) et la technique utilisée pour estimer les seuils. A titre d'exemple, nous présentons deux techniques de détermination des seuils :

- par la recherche des minima locaux de l'histogramme [Pre 66]. Cette approche comporte deux inconvénients majeurs. Le premier est que l'histogramme d'une image réelle comporte des irrégularités dues à la présence de bruit dans l'image, qui se manifestent par des minima locaux. Le second inconvénient est que l'histogramme comporte parfois un plateau entre deux modes consécutifs (par exemple une image contenant deux régions de niveaux de gris constants et différents). Ainsi, aucun minimum local n'existe entre ces deux modes.
- optimisant un critère donné. Les algorithmes reposant sur cette idée diffèrent essentiellement par le critère utilisé. On trouve des critères fondés sur l'homogénéité des régions résultats [Ost 78], sur l'entropie *a posteriori* [Pun 81], sur la préservation des moments de l'image [Tsa 85], sur l'erreur de classification [Kit 86]. Les algorithmes obtenus sont optimaux aux sens du critère initial. Cependant, la plupart d'entre eux se limite aux cas des histogrammes bimodaux.

Nous proposons un algorithme fondé sur cette dernière idée. Plus précisément, nous proposons la détermination de seuils minimisant l'erreur de classification [Zio 89b]. Si cette idée n'est pas récente, il reste à trouver une méthode globale pour partitionner un histogramme multimodal, facile à mettre en œuvre et assurant une partition correcte de l'image en régions (c'est-à-dire que les régions sont uniformes et homogènes, les régions adjacentes possèdent des caractéristiques différentes, les frontières des régions sont lisses [WD 88]). L'algorithme comporte deux passes :

- approximation de l'histogramme par une combinaison linéaire de gaussiennes. Pour ce faire, nous proposons l'utilisation de l'apprentissage numérique qui consiste à partitionner l'histogramme en modes selon un critère de décision prédéfini.
- détermination des seuils minimisant une erreur de classification d'une façon rapide contrairement à l'algorithme de Nakagawa [Nak 79] et sans les problèmes d'initialisation et de convergence de celui de Kittler [Kit 86].

3.2 Approximation de l'histogramme

La distribution des niveaux de gris d'une image est représentée sous la forme d'un histogramme $h(x)$. Cet histogramme est composé d'un ensemble de modes représentant les objets et le fond de l'image. Il peut être vu alors comme une estimation de la densité de probabilité $p(X/\theta)$ de ces modes.

Il s'agit de déterminer les statistiques θ permettant de définir complètement $p(X/\theta)$. Ce problème est complexe dans le cas où la forme des densités de probabilité est inconnue. Nous supposons que la forme de la densité de probabilité de chaque mode est une gaussienne. Le problème se limite à l'estimation des statistiques d'ordre deux. Cette hypothèse semble être vérifiée si les images utilisées vérifient les conditions décrites dans le paragraphe 3.1. Puisque nous utilisons une règle de décision bayésienne pour calculer les seuils, nous avons besoin uniquement de la moyenne, de la variance et de la probabilité *a priori*.

3.2.1 Estimation des paramètres

Il s'agit d'estimer pour chaque mode la moyenne, la variance et la probabilité *a priori*. Pour ce faire, nous proposons l'utilisation de l'algorithme de maximum de vraisemblance. Cet algorithme a l'avantage de fournir dans le cas d'une distribution normale, une estimation de la moyenne et de la variance non biaisée, efficace, consistante et suffisante [Pug 82]. Son principe se résume par la détermination des paramètres pour lesquels on obtient le maximum de concordance entre l'échantillon observé et les distributions théoriques. Plus précisément : soient $h(x)$ un histogramme à M modes et $X = x_1, \dots, x_N$ des variables aléatoires représentant les abscisses de l'histogramme. Un mode i est caractérisé par le vecteur de paramètres θ_i et par la probabilité *a priori* $P(i)$. Le problème de détermination des vecteurs $\Theta = (\theta_1, \dots, \theta_M)$ et $(P(1), \dots, P(M))$ se pose de la façon suivante :

$$\max_{\Theta} p(X/\Theta) \quad (3.1)$$

avec les contraintes :

$$\begin{aligned} P(i) &\geq 0 \quad \forall i \in [1, M] \\ \sum_{i=1}^M P(i) &= 1 \end{aligned}$$

La présence de ces contraintes permet la prise en considération des probabilités *a priori*. Le problème ci-dessus (eq. 3.1) peut être ramené à un problème d'optimisation sans contrainte en utilisant les multiplicateurs de Lagrange. En d'autres termes, il s'agit de maximiser la fonction suivante :

$$\phi(X, \Theta, \Lambda) = \ln(p(X/\Theta)) + \lambda_1 \left(1 - \sum_{i=1}^M P(i)\right) + \lambda_2 P(1) + \dots + \lambda_{M+1} P(M) \quad (3.2)$$

où Λ sont les multiplicateurs de Lagrange. Nous avons remplacé la fonction $p(X/\Theta)$ (eq. 3.1) par la fonction $(\ln(p(X/\Theta)))$ (eq. 3.2) pour des raisons de commodité.

La résolution analytique de ce problème (voir Annexe A) nous conduit à écrire :

$$\hat{\mu}(i) = \frac{\sum_{k=1}^N \hat{h}(x_k) \hat{p}(i/x_k, \theta_i) x_k}{\sum_{k=1}^N \hat{h}(x_k) \hat{p}(i/x_k, \theta_i)} \quad (3.3)$$

$$\hat{\sigma}^2(i) = \frac{\sum_{k=1}^N \hat{h}(x_k) \hat{p}(i/x_k, \theta_i) (x_k - \hat{\mu}(i))^2}{\sum_{k=1}^N \hat{h}(x_k) \hat{p}(i/x_k, \theta_i)} \quad (3.4)$$

$$\hat{P}(i) = \frac{\sum_{k=1}^N \hat{h}(x_k) \hat{p}(i/x_k, \theta_i)}{\sum_{k=1}^N \hat{h}(x_k)} \quad (3.5)$$

où,

$$\hat{h}(x_k) = \sum_{i=1}^M \frac{\hat{P}(i)}{\sqrt{2\pi\hat{\sigma}^2(i)}} e^{-\frac{(x_k - \hat{\mu}(i))^2}{2\hat{\sigma}^2(i)}}$$

Les équations 3.3, 3.4 et 3.5 permettent de calculer itérativement les paramètres de chaque mode, cette estimation ne se réalise que si le nombre de modes qui composent l'histogramme est connu et que nous disposons d'un processus d'initialisation.

3.2.2 Valeurs initiales des paramètres à estimer

Les équations 3.3), 3.4 et 3.5 convergent vers le maximum local le plus proche. La valeur initiale des paramètres doit être proche du maximum global afin de permettre une convergence rapide et fiable de l'algorithme, soit :

$$\mu_i^{(0)} = \frac{\sum_{k=n_1(i)}^{n_2(i)} x_k h(x_k)}{\sum_{k=n_1(i)}^{n_2(i)} h(x_k)} \quad (3.6)$$

$$\sigma_i^{2(0)} = \frac{\sum_{k=v_1(i)}^{v_2(i)} (x_k - \mu_i^{(0)})^2 h(x_k)}{\sum_{k=v_1(i)}^{v_2(i)} h(x_k)} \quad (3.7)$$

$$P^{(0)}(i) = \frac{\sum_{k=v_1(i)}^{v_2(i)} h(x_k)}{\sum_{j=0}^N h(x_k)} \quad (3.8)$$

où $n_1(i)$ et $n_2(i)$ sont les rangs des deux points d'inflexion du $i^{\text{ème}}$ mode et $v_1(i)$ et $v_2(i)$ les rangs des points situés au fond des vallées de part et d'autre du $i^{\text{ème}}$ mode, correspondant à la moyenne des modes négatifs.

3.2.3 Décomposition d'histogramme

L'objectif maintenant est de déterminer le nombre de modes qui composent l'histogramme. Nakagaw et Rosenfeld [Nak 79] proposent une décomposition de l'histogramme fondée sur la recherche des minima locaux dans l'histogramme. Cette procédure s'exécute mal dans le cas où l'histogramme comporte des vallées plates ou irrégulières [Sah 88]. Pour pallier à ces problèmes, nous ne considérons que la partie du mode qui se trouve entre deux points d'inflexion. Le point d'inflexion d'une fonction se recherche soit par le maximum de la dérivée première, soit par le passage par zéro de la dérivée seconde. Si l'équivalence de ces deux méthodes est vérifiée pour des fonctions continues, elle ne l'est pas dans le cas des fonctions discrètes. La table 3.1 décrit un exemple où un point d'inflexion mis en évidence par la première dérivée sous forme d'un minimum local, alors qu'aucun passage par zéro correspondant n'est fourni par la dérivée seconde.

<i>Signal</i>	1	1	1	1	100	20	18	5	2	1	1	10	1	1	1	1
<i>première dérivée</i>	*	0	0	99	19	-82	-15	-16	-4	-1	9	0	-9	0	0	*
<i>dérivée seconde</i>	*	*	99	19	-181	-34	66	11	15	13	1	-18	0	9	0	*

Tableau 3.1. Un signal 1D et ses deux premières dérivées.

D'une façon générale, la dérivée seconde est plus sensible au bruit que la dérivée première. Elle possède alors un rapport signal/bruit inférieur. La première dérivée permet de détecter même les petits modes de l'histogramme qui peuvent correspondre à une région dans l'image. L'algorithme de décomposition de l'histogramme se compose d'une étape de différentiation et d'une étape de détermination de modes.

Calcul de la dérivée première

L'utilisation de l'histogramme sans aucune transformation préalable produit beaucoup d'effets non désirables, conséquences des variations subtiles causées par le bruit. Le lissage par une spline cubique est la méthode la plus appropriée pour régulariser un problème mal posé (cf. 3.2). Cependant, un histogramme est souvent composé de modes étroits. La détection d'un point d'inflexion d'un mode est alors influencée par la présence de l'autre. Il est bien connu que le filtre exponentiel est plus adapté dans le cas de multi-marche (cf. 1.3). De plus, il a l'avantage d'être réalisé efficacement par des équations aux différences d'ordre 1. Nous proposons alors l'utilisation de la première dérivée de ce filtre pour l'estimation des points d'inflexion de l'histogramme.

Couplage des points d'inflexion

Une fois les optima trouvés, il reste à décomposer l'histogramme en modes positifs. La décomposition de l'histogramme consiste à trouver un appariement des maxima et des minima résultats de la dérivée première. Le nombre d'appariements possible est une fonction exponentielle du nombre de couples d'optima (minimum, maximum). En effet, si on a M couples d'optima, le nombre d'appariements est égal à $M!$. Parmi toutes les possibilités, une et une seule permet une décomposition de l'histogramme en modes positifs. Cette décomposition se réalise par le parcours de l'histogramme dans un sens donné (ici suivant les x croissants), chaque maximum rencontré étant apparié avec le minimum suivant.

Remarquons que le nombre de modes est inversement proportionnel au degré de lissage de l'histogramme. Le choix de ce degré dépend de l'utilisation ultérieure.

3.3 Détermination des seuils

Le problème maintenant est de déterminer les seuils séparant les modes pour que l'erreur de classification soit minimale. Il est bien connu que dans le cas de deux classes, la règle de décision bayésienne (qui consiste à favoriser la classe ayant la probabilité *a posteriori* la plus élevée) est la solution à ce problème. Examinons le problème dans le cas de M modes : soient T_0, \dots, T_M les seuils à calculer et c_{ij} le risque d'affecter le niveau de gris x au $j^{\text{ème}}$ mode défini par l'intervalle $[T_{j-1} + 1, T_j]$ sachant que $x \in [T_{i-1} + 1, T_i]$, $i \neq j$. L'erreur de probabilité est alors :

$$e(T_0, \dots, T_M) = \sum_{j=1}^M \sum_{i=1}^M \int_{T_{j-1}+1}^{T_j} c_{ij} p(i/x) dx \quad (3.9)$$

où $T_0 = 0$, $T_M =$ niveau de gris maximal et $p(i/x)$ la densité de probabilité *a posteriori*.

On veut déterminer les seuils T_0, \dots, T_M qui minimisent cette fonction. La solution de ce problème est donnée dans [Fuk 72], l'erreur est minimale si :

$$P(i)p(x/i) \geq P(j)p(x/j) \quad \forall i \neq j \quad \text{si } x \in [T_{i-1} + 1, T_i] \quad (3.10)$$

D'où la règle de décision :

$$\begin{cases} P(i)p(x/i) \geq P(j)p(x/j) & \text{si } x \in [T_{i-1} + 1, T_i] \\ P(i)p(x/i) < P(j)p(x/j) & \text{si } x \in [T_{j-1} + 1, T_j] \end{cases}$$

Par hypothèse :

$$p(x/i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(x - \mu_i)^2}{2\sigma_i^2}\right) \quad (3.11)$$

En remplaçant les densités de probabilité conditionnelle données en (eq. 3.11) par leurs valeurs et en prenant le logarithme de chaque membre, la règle de décision limite s'écrit :

$$\ln(P(i)) - \ln(\sigma(i)) - \frac{1}{2}\left(\frac{T_i - \mu(i)}{\sigma(i)}\right)^2 = \ln(P(j)) - \ln(\sigma(j)) - \frac{1}{2}\left(\frac{T_i - \mu(j)}{\sigma(j)}\right)^2 \quad (3.12)$$

ou encore

$$\frac{1}{2}\left(\frac{1}{\sigma(i)^2} - \frac{1}{\sigma(j)^2}\right)T_i^2 + \left(\frac{\mu(j)}{\sigma(j)^2} - \frac{\mu(i)}{\sigma(i)^2}\right)T_i + \frac{\mu(i)^2}{2\sigma(i)^2} - \frac{\mu(j)^2}{2\sigma(j)^2} - \ln\left(\frac{P(i)\sigma(j)}{P(j)\sigma(i)}\right) = 0 \quad (3.13)$$

La recherche du seuil optimal T_i^* séparant les deux modes i et j se limite à la résolution de cette équation du deuxième degré. Pour un histogramme à M modes, le nombre de seuils possible est 2^{M-1} . Nous ne considérons que les seuils entre deux modes consécutifs (le mode i et le mode $j = i + 1$). Ainsi l'équation 3.13 devient :

$$\frac{1}{2}\left(\frac{1}{\sigma(i)^2} - \frac{1}{\sigma(i+1)^2}\right)T_i^2 + \left(\frac{\mu(i+1)}{\sigma(i+1)^2} - \frac{\mu(i)}{\sigma(i)^2}\right)T_i + \frac{\mu(i)^2}{2\sigma(i)^2} - \frac{\mu(i+1)^2}{2\sigma(i+1)^2} - \ln\left(\frac{P(i)\sigma(i+1)}{P(i+1)\sigma(i)}\right) = 0 \quad (3.14)$$

Un seuil entre un mode i et un mode $i + 1$ sera noté par T_i^* . D'après l'équation 3.14, le nombre de seuils possible dans le pire des cas (le cas où l'équation 3.14 admet deux solutions pour chaque i et $i + 1$) n'est que $2(M - 1)$. La sélection des $M - 1$ seuils nécessaires parmi les $2(M - 1)$ peut être effectuée par la recherche des seuils minimisant l'erreur (eq. 3.9) et par l'utilisation de la règle suivante : un seuil T_i^* doit appartenir à l'intervalle $[\mu_i, \mu_{i+1}]$.

Une solution réelle pour deux modes i et $i + 1$ de l'équation 3.14 n'existe que si et seulement si :

$$(\mu(i) - \mu(i+1))^2 \geq 2(\sigma(i)^2 - \sigma(i+1)^2) \ln\left(\frac{P(i)\sigma(i+1)}{P(i+1)\sigma(i)}\right)$$

Dans le cas où aucune solution réelle n'existe, aucun seuil ne sera déterminé entre ces deux modes. La figure 3.1 représente un histogramme où la solution n'existe pas. D'une façon générale, cette situation ne se présente que dans le cas où les deux modes i et $i + 1$ ne sont pas distincts. Dans la pratique, pour que les seuils soient corrects, il faut que l'histogramme comporte des modes distincts. Nous introduisons alors une mesure exprimant la confiance que l'on peut accorder à un seuil T_i^* .

Trois facteurs interviennent :

- l'espacement des deux modes consécutifs :

$$E(T_i^*) = \frac{\mu(i+1) - \mu(i)}{ng_{max} - ng_{min}} \quad (3.15)$$

- le rapport entre leur amplitude :

$$R(T_i^*) = \frac{\min_{x \in \{\mu(i), \mu(i+1)\}} p(x/\theta)}{\max_{x \in \{\mu(i), \mu(i+1)\}} p(x/\theta)} \quad (3.16)$$

- le rapport de l'amplitude de la vallée sur celui du mode.

$$V(T_i^*) = \frac{\min_{x \in \{\mu(i), \mu(i)+1, \dots, \mu(i+1)\}} p(x/\theta)}{\min_{x \in \{\mu(i), \mu(i+1)\}} p(x/\theta)} \quad (3.17)$$

Le degré de bimodalité de deux distributions de niveaux de gris consécutives peut être défini par :

$$B(T_i^*) = E(T_i^*) * R(T_i^*) * (1 - V(T_i^*)) \quad (3.18)$$

Les valeurs de chacune de ces trois fonctions (eq. 3.15), (eq. 3.16) et (eq. 3.17) appartiennent à l'intervalle $[0, 1]$. $B(T_i^*)$ prend aussi ses valeurs dans cet intervalle. Elle est d'autant proche de 1 que les distributions sont bimodales.

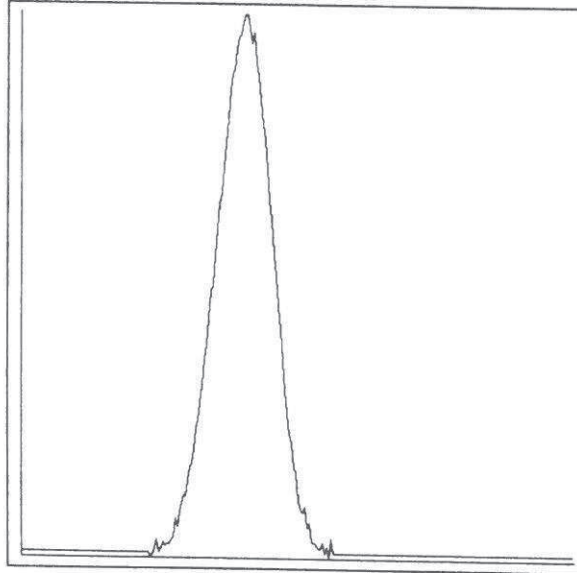


Figure 3.1. Un exemple d'histogramme limite de notre algorithme. Cet histogramme est composé de deux modes. Le premier a pour paramètres : moyenne 100, variance 150 et probabilité 0.62. Le deuxième : moyenne 110, variance 100 et probabilité 0.38.

3.4 Principe de l'algorithme

L'algorithme de recherche de contours se résume par :

- l'approximation de l'histogramme :
 - la décomposition de l'histogramme en modes par le calcul de sa dérivée première et le couplage des optima de celle-ci pour former les modes (cf. 3.2.3),
 - le calcul de paramètres initiaux à l'aide des équations 3.6, 3.7 et 3.8,
 - le raffinement itératif des valeurs de paramètres initiaux à l'aide des équations 3.3, 3.4 et 3.5. Le critère d'arrêt est donné par le fait que la différence entre les valeurs de la fonction (eq. 3.9) à deux étapes consécutives soit infiniment petite.
- le calcul de seuils :
 - pour chaque couple de modes consécutifs, calcul du seuil par l'équation 3.14,
 - le test de validité de seuils par la relation 3.18. Dans le cas où la valeur est faible, le seuil correspondant est ignoré.
- la mise en évidence des points de contours dans l'image initiale. On considère comme contours les frontières des régions. Les régions sont obtenues par une simple opération de seuillage.

3.5 Evaluation de l'algorithme

3.5.1 Résultats expérimentaux

Histogramme artificiel

Nous commençons par présenter les résultats de l'exécution de cet algorithme dans le cas des histogrammes multi-modaux idéaux, simulés par la somme des gaussiennes de paramètres connus. Il est important de rappeler qu'aucune image réelle ne possède un tel histogramme, mais ceci permet de montrer la précision de l'algorithme, chose qui ne peut pas être réalisée à partir des images réelles.

Considérons un histogramme comme la somme d'une composante "signal pur" et d'une composante "bruit blanc" $b(k)$:

$$h(x_k) = \sum_{i=1}^M \frac{P(i)}{\sqrt{2\pi}\sigma(i)} e^{-\frac{(x_k - \mu(i))^2}{2\sigma^2(i)}} + b(k)$$

La figure 3.2 et la table 3.2 montrent le cas d'un histogramme à quatre modes bien séparés. A cet histogramme est ajouté un bruit blanc de variance égale à 10. Le seuil de convergence est égal à 0.01. Les estimations obtenues sont satisfaisantes.

La figure 3.3 décrit un histogramme à quatre modes moins espacés que ceux du cas précédent, la variance du bruit ajouté est égale à 20. La table 3.3 représente les estimations des paramètres, le seuil de convergence étant de 0.05. Remarquons que, même si les erreurs sur les probabilités *a priori* sont importantes, les seuils obtenus sont corrects. La table 3.4 représente les estimations des paramètres à partir du même histogramme, en utilisant le même seuil de convergence. La différence réside dans l'utilisation d'une échelle plus élevée

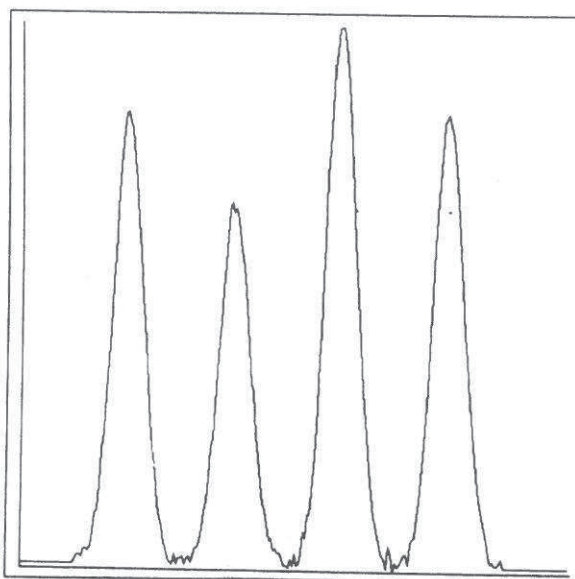


Figure 3.2. Premier exemple d'un histogramme.

	<i>vraies valeurs</i>				<i>valeurs estimées</i>			
	moyenne	variance	probabilité	seuil	moyenne	variance	probabilité	seuil
<i>mode 1</i>	50	55	0.25		49.04	54.28	0.254	
				75				75
<i>mode 2</i>	100	55	0.25		99.99	55.03	0.202	
				125				125
<i>mode 3</i>	150	55	0.25		150.00	55.68	0.293	
				175				175
<i>mode 4</i>	200	55	0.25		198.98	54.48	0.251	

Tableau 3.2. Résultats de l'estimation des paramètres de l'histogramme décrit par la figure 3.1

(car, le degré de lissage de l'histogramme est plus élevé). Nous remarquons que le second mode a disparu. Le premier mode n'est donc plus gaussien. Ce qui explique les erreurs commises sur les seuils obtenus.

Images naturelles

Nous présentons les résultats obtenus à partir des images réelles. La figure 3.4.a décrit une image 256x176 représentant une coupe de tibia. Son histogramme est donné dans la figure 3.4.b et les contours résultats sont donnés dans les figures 3.4.d et 3.4.e. Les résultats fournis par le détecteur de Deriche sont donnés dans la figure 3.4.c. Nous avons expérimenté cet algorithme également sur une image contenant des écrous posés sur un fond (fig. 3.5.a, 3.5.b). Les contours correspondant sont donnés dans la figure 3.5.c.

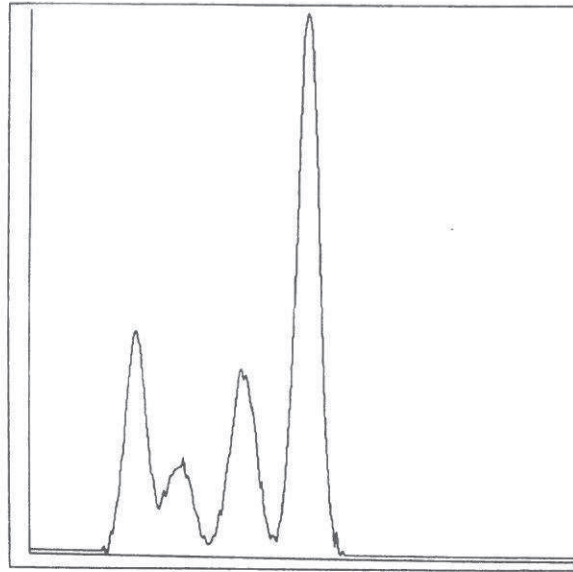


Figure 3.3. Second exemple d'un histogramme.

	<i>vraies valeurs</i>				<i>valeurs estimées</i>			
	moyenne	variance	probabilité	seuil	moyenne	variance	probabilité	seuil
<i>mode 1</i>	50	25	0.2		49.02	23.29	0.250	
				60				60
<i>mode 2</i>	70	30	0.1		70.00	30.97	0.156	
				85				84
<i>mode 3</i>	100	35	0.2		99.99	37.39	0.176	
				115				115
<i>mode 4</i>	130	25	0.5		129.99	24.65	0.417	

Tableau 3.3. Résultats de l'estimation des paramètres de l'histogramme de la figure 3.3

	<i>valeurs estimées</i>			
	moyenne	variance	probabilité	seuil
<i>mode 1</i>	52.29	44.45	0.0413	
				73
<i>mode 2</i>	99.99	100.60	0.023	
				113
<i>mode 3</i>	129.92	26.56	0.935	

Tableau 3.4. Résultats de l'estimation des paramètres de l'histogramme de la figure 3.3 obtenue après un lissage fort.

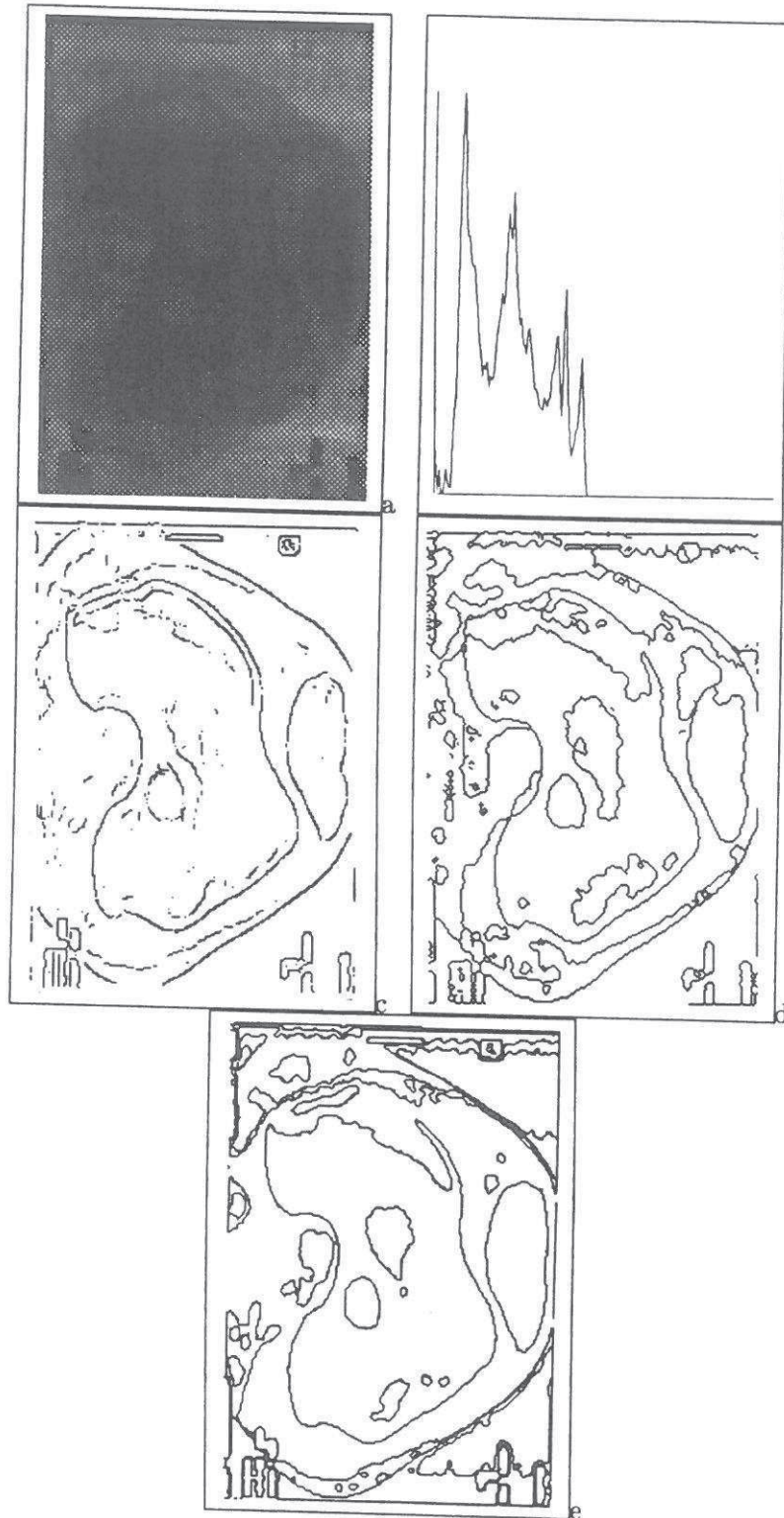


Figure 3.4. (a) Image rayon X (256×176), (b) l'histogramme correspondant, (c) contours extraits par Deriche ($\alpha = 1$), (d) et (e) contours obtenus par notre algorithme à deux échelles différentes.

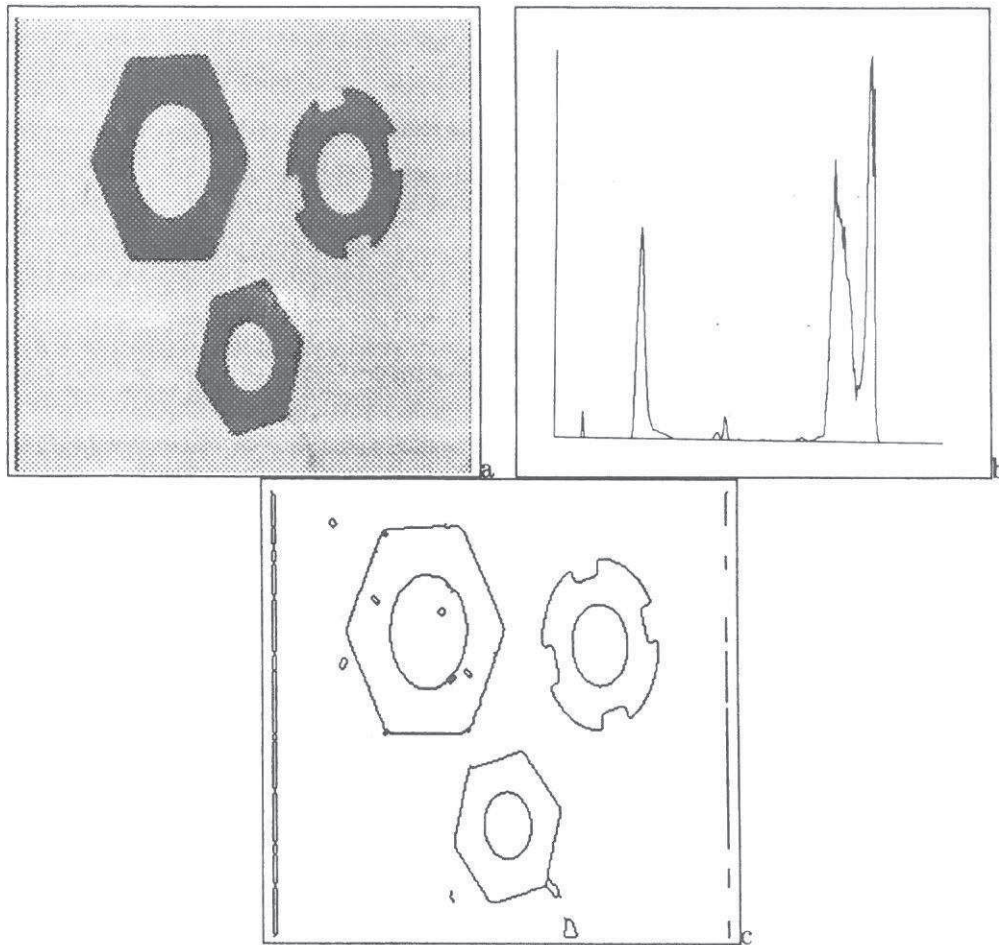


Figure 3.5. (a) Image originale (256 × 256), (b) l'histogramme correspondant, (c) contours obtenus par notre algorithme.

3.5.2 Comparaison de l'algorithme

Dans la classe d'images qui peuvent être traitées par cet algorithme, la recherche des discontinuités et des continuités dans les propriétés des objets semble être duale. En d'autres termes, un algorithme de seuillage peut être perçu comme étant un algorithme d'extraction de régions et également comme étant un détecteur de contours. L'évaluation de performances d'un tel algorithme doit alors tenir compte des propriétés des régions résultats et des propriétés de leurs frontières. On définit dans ce contexte une segmentation correcte comme étant une partition de l'image en régions uniformes possédant des frontières correctement localisées. Nous présentons la définition précise de ces deux mesures uniquement dans le cas d'un histogramme à deux modes pour pouvoir comparer notre détecteur avec deux autres qui sont destinés uniquement à ce type d'histogramme.

- La mesure d'uniformité [Lev 85] :

$$U(t) = 1 - \frac{\sigma^2(1) + \sigma^2(2)}{C} \quad (3.19)$$

avec $\sigma^2(i)$ la variance de la $i^{\text{ème}}$ région ($i = 1, 2$) et $C = \frac{\sum_{i=1}^{i=2} (ng_{max}(i) - ng_{min}(i))}{4}$ un facteur de normalisation

- La mesure de forme des régions [Sah 88] :

$$S(t) = \frac{\sum_{(x,y)} Sgn(I(x,y) - f_V(x,y)) \Delta(x,y) Sgn(I(x,y) - t)}{\sum_{(x,y)} \Delta(x,y)} \quad (3.20)$$

où $f_V(x,y)$ est la moyenne du niveau de gris du voisinage $V(x,y)$, t le seuil de l'image, $\Delta(x,y)$ représentant le module du gradient. La fonction $Sgn(x)$ est définie par :

$$Sgn(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ -1 & \text{si } x < 0 \end{cases}$$

Il s'agit maintenant de définir par combinaison des deux mesures (eq. 3.19) et (eq. 3.20) une mesure de performances de l'algorithme pour un seuil donné. Un "bon" algorithme maximise les mesures d'uniformité et de la forme. La mesure de performances est alors le produit des ces deux mesures. Elle est donnée par :

$$P(t) = U(t) S(t) \quad (3.21)$$

Les résultats de l'évaluation des performances de notre algorithme sur les images *cam*, *cam1* et *cam2* données respectivement dans les figures 3.6.a, 3.6.c, 3.6.e et sur l'image *geol* donnée dans la figure 3.6.g sont présentés dans la table 3.5. La table 3.6 présente les valeurs optimales des mesures ci-dessus. $U(t) = 0.928420(74)$ veut dire que la valeur du seuil pour lequel $U(t)$ est maximal est 74 et la valeur maximale de $U(t)$ est égale à 0.928420. Les valeurs présentées dans la première ligne sont les seuils pour lequel $P(t)$ est maximal. La dernière ligne, libellée bimodalité, comporte les mesures de validité du seuil défini par l'équation 3.18 pour chaque image.

Images	cam	cam1	cam2	geol
Seuils	142	80	92	118
$U(t)$	0.887395	0.964580	0.966003	0.929130
$S(t)$	0.784476	0.811750	0.637505	0.652588
$P(t)$	0.696140	0.782998	0.615832	0.606334

Tableau 3.5. Résultats de l'évaluation de notre algorithme.

Nous avons évalué les performances de deux autres algorithmes ne s'exécutant que sur des histogrammes à deux modes. Le premier est celui de Kittler [Kit 86] fondé également sur l'estimation de seuil minimisant l'erreur de classification. Les résultats de cet algorithme sont donnés dans la table 3.7. Le second est l'algorithme de Pun [Pun 81] fondé sur l'estimation de seuil minimisant l'entropie *a posteriori*. Les résultats sont donnés dans la table 3.8.

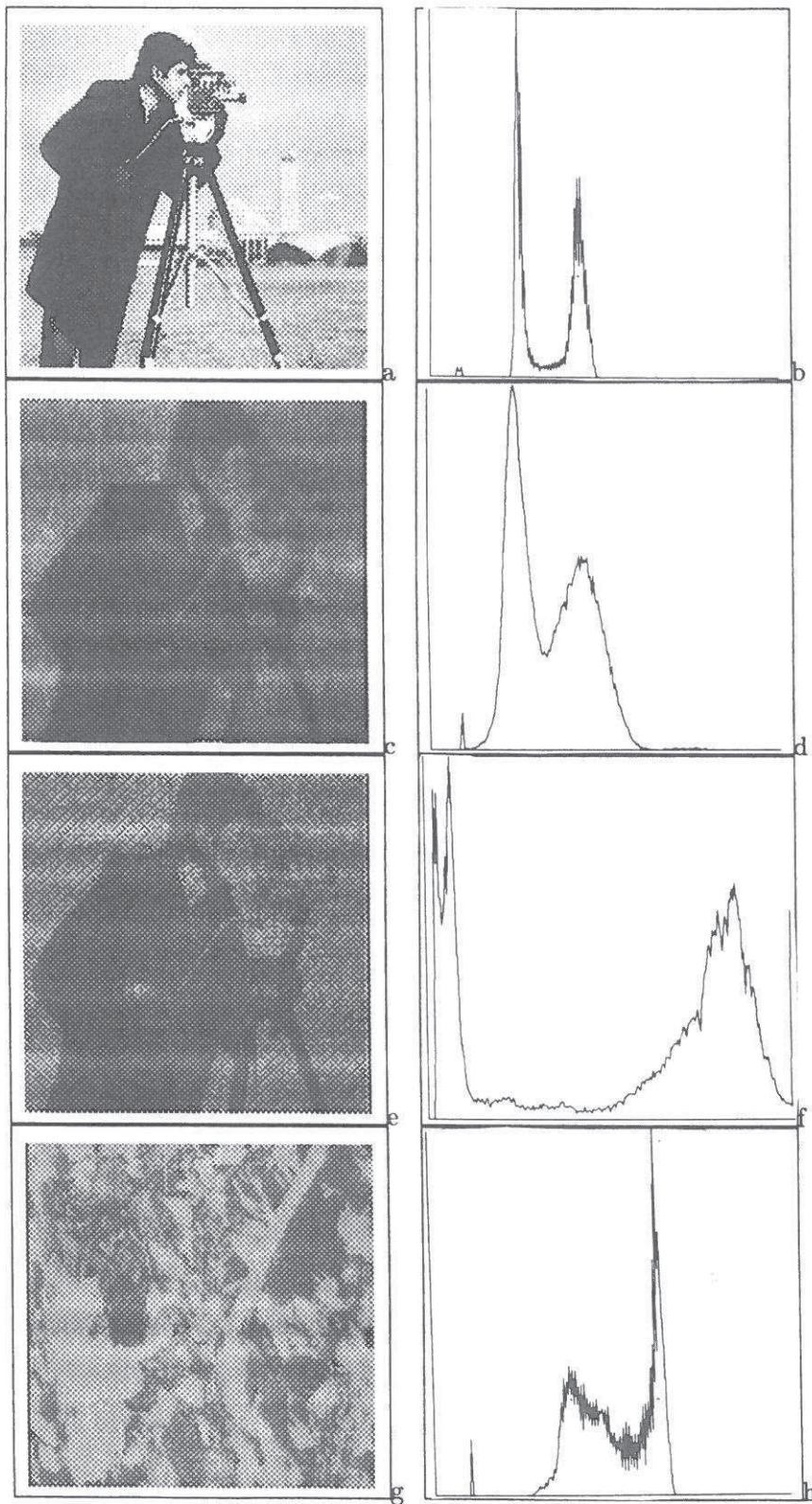


Figure 3.6. (a) Image originale du cameramen, cam, (256×256), (b) l'histogramme correspondant, (c) une deuxième version de l'image du cameramen, cam1, (d) l'histogramme correspondant, (e) une autre image du cameramen, cam2, (f) son histogramme, (g) image d'ardoise, geol, (h) son histogramme.

Images	cam	cam1	cam2	geol
Seuils	123	80	101	122
$U(t)$	0.928420 (74)	0.967232 (85)	0.968446 (83)	0.948113 (137)
$S(t)$	0.802793 (130)	0.811750 (80)	0.657929 (101)	0.655642 (119)
$P(t)$	0.730608 (123)	0.782998 (80)	0.630727 (101)	0.611898 (122)
bimodalite	0.001245	0.291760	0.014995	0.039607

Tableau 3.6. Valeurs optimales des mesures de performances. Un nombre entre parenthèses représente le seuil optimisant la mesure correspondante.

Enfin, la table 3.9 présente le temps écoulé lors de l'exécution de chacun des algorithmes. Elle contient aussi les degrés de bimodalité (eq. 3.18) pour chaque image (calculée au point T égale au seuil optimal présenté dans la table 3.6).

Cette évaluation montre que les performances des trois algorithmes sont liées au degré de bimodalité de l'histogramme. L'algorithme de Kittler se détériore rapidement en présence d'histogramme possédant un faible degré de bimodalité. Notre algorithme possède les meilleures performances. Il fournit des régions plus homogènes et des frontières mieux localisées.

En ce qui concerne l'efficacité, l'algorithme de Pun est le plus rapide. Le temps de calcul de l'algorithme de Kittler est le plus affecté par le degré de bimodalité de l'histogramme.

Images	cam	cam1	cam2	geol
Seuils	224	84	114	151
$U(t)$	0.701152	0.967213	0.941173	0.940810
$S(t)$	0.300376	0.789204	0.610265	0.134288
$P(t)$	0.210609	0.763329	0.574366	0.126340

Tableau 3.7. Résultats de l'évaluation de l'algorithme de Kittler.

Images	cam	cam1	cam2	geol
Seuils	189	92	84	130
$U(t)$	0.728106	0.964193	0.968381	0.945507
$S(t)$	0.545939	0.666798	0.620275	0.576546
$P(t)$	0.397501	0.642922	0.600663	0.545128

Tableau 3.8. Résultats de l'évaluation de l'algorithme de Pun.

3.6 Conclusion

Nous avons présenté un algorithme de détection de contours de type marche à partir d'histogramme. Il est conçu pour des images de niveaux de gris dont le degré de multimodalité est relativement élevé. L'histogramme est approché par une combinaison linéaire de gaussiennes en maximisant la vraisemblance. Ensuite, les seuils sont déterminés de telle

Images	cam	cam1	cam2	geol
Pun	0:01	0:01	0:01	0:01
Kittler	0:30	0:01	0:07	0:01
Ziou	0:19	0:08	0:13	0:09
bimodalite	0.001245	0.291760	0.014995	0.039607

Tableau 3.9. Le temps d'exécution des trois algorithmes.

façon que l'erreur de classification soit minimale. Les performances de notre algorithme sont évaluées et comparées à d'autres algorithmes connus. Il est plus performant que celui de Kittler [Kit 86] et celui de Pun [Pun 81]. Les contours obtenus sont satisfaisants. Cependant, plusieurs améliorations peuvent être apportées ; en particulier :

- L'algorithme d'approximation de l'histogramme est itératif. Il converge alors vers le minimum le plus proche des valeurs données par l'étape d'initialisation. De plus, l'estimation du nombre de modes de l'histogramme dépend du degré de lissage. Il est possible d'utiliser les mesures de performances (eq. 3.19), (eq. 3.20) et (eq. 3.21) et les mesures de validités de seuils (eq. 3.15), (eq. 3.16) et (eq. 3.17) pour produire un algorithme flexible qui est capable d'ajuster automatiquement ses paramètres, puisque l'estimation de seuil est effectuée à partir d'un histogramme 1D. Il semble facile d'utiliser une approche de segmentation multi-échelle pour produire les régions qui apparaissent à une échelle donnée.
- Le traitement d'images plus complexes par notre algorithme implique certainement la prise en compte de la distribution spatiale des niveaux de gris. Celle-ci peut être effectuée, par exemple, par des raffinements successifs de la façon suivante : la détermination des régions en considérant toute l'image, puis le raffinement des régions résultats, etc. On peut utiliser les mesures de performances définies dans le paragraphe 3.5 pour contrôler le processus de raffinement.
- L'algorithme que nous avons utilisé pour marquer les points de contours dans l'image initiale est trivial. Les contours obtenus ne sont pas nécessairement d'épaisseur 1. Pour toute utilisation ultérieure, une étape d'amincissement est donc nécessaire. Un algorithme de marquage assurant des contours d'épaisseur 1 n'est pas difficile à écrire ([Suk 88] présente un tel algorithme).

3.7 Annexe A

Cette annexe a pour objectif de résoudre le problème d'optimisation donné par l'équation 3.2. Pour ce faire, il suffit de déterminer la solution de l'équation suivante :

$$\frac{\partial}{\partial \Theta} \phi(X, \Theta, \Lambda) \Big|_{\Theta=\hat{\Theta}} = 0 \quad (3.22)$$

Si nous supposons que les variables $x_i, i = 1, \dots, N$ sont indépendantes, alors on peut écrire :

$$p(X/\Theta) = \prod_{k=1}^N p(x_k/\Theta)$$

$$p(x_k/\Theta) = \sum_{j=1}^M p(x_k/j, \theta_j) P(j)$$

En substituant ces équations dans l'équation 3.2, on obtient :

$$\phi(X, \Theta, \Lambda) = \sum_{k=1}^N \ln \left(\sum_{i=1}^M p(x_k/i, \theta_i) \right) + \lambda_1 \left(1 - \sum_{i=1}^M P(i) \right) + \lambda_2 P(1) + \dots + \lambda_{M+1} P(M) \quad (3.23)$$

Lors de l'expérimentation, nous avons constaté que la prise en considération des fréquences permet d'obtenir des résultats meilleurs. Au lieu de chercher les paramètres maximisant la fonction 3.23, nous considérons le problème suivant :

$$\phi(X, \Theta, \Lambda) = \sum_{k=1}^N h(x_k) \ln \left(\sum_{i=1}^M p(x_k/i, \theta_i) \right) + \lambda_1 \left(1 - \sum_{i=1}^M P(i) \right) + \lambda_2 P(1) + \dots + \lambda_{M+1} P(M) \quad (3.24)$$

En calculant la dérivée par rapport à θ_i on obtient :

$$\begin{aligned} \frac{\partial \phi(X, \Theta, \Lambda)}{\partial \theta_i} &= \frac{\partial}{\partial \theta_i} \sum_{k=1}^N h(x_k) \ln \left(\sum_{j=1}^M p(x_k/j, \theta_j) P(j) \right) \\ &= \sum_{k=1}^N \frac{h(x_k)}{\sum_{j=1}^M p(x_k/j, \theta_j) P(j)} \frac{\partial}{\partial \theta_i} \sum_{j=1}^M p(x_k/j, \theta_j) P(j) \\ &= \sum_{k=1}^N \frac{h(x_k)}{p(x_k/\Theta)} \frac{\partial}{\partial \theta_i} \sum_{j=1}^M p(x_k/j, \theta_j) P(j) \end{aligned}$$

D'après Bayes

$$p(i/x_k, \Theta) = \frac{p(x_k/i, \theta_i) P(i)}{p(x_k/\Theta)} \quad (3.25)$$

Alors

$$\frac{\partial \phi(X, \Theta, \Lambda)}{\partial \theta_i} = \sum_{k=1}^N h(x_k) p(i/x_k, \theta_i) \frac{\frac{\partial}{\partial \theta_i} \sum_{j=1}^M p(x_k/j, \theta_j) P(j)}{p(x_k/i, \theta_i) P(i)}$$

Finalement, on obtient :

$$\frac{\partial \phi(X, \Theta, \Lambda)}{\partial \theta_i} = \sum_{k=1}^N h(x_k) p(i/x_k, \theta_i) \frac{\partial}{\partial \theta_i} \ln(p(x_k/i, \theta_i)) \quad (3.26)$$

La dérivée de la fonction objective $\phi(X, \Theta, \Lambda)$ étant calculée, il s'agit maintenant d'estimer les statistiques (moyenne, variance et probabilité *a priori* de chacun des modes de l'histogramme.

1. Estimation de la moyenne et de la variance :

Puisque nous avons supposé que les modes suivent une loi normale, alors

$$p(x/i, \theta_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(x - \mu_i)^2}{2\sigma_i^2}\right) \quad (3.27)$$

En substituant dans l'équation 3.26 et en utilisant la relation 3.22 on obtient :

$$\frac{\partial \phi(X, \Theta, \Lambda)}{\partial \mu_i} = \sum_{k=1}^N h(x_k) p(i/x_k, \theta_i) \left(\frac{x_k - \mu_i}{\sigma_i^2}\right) \Big|_{\mu_i = \hat{\mu}_i} = 0 \quad (3.28)$$

$$\frac{\partial \phi(X, \Theta, \Lambda)}{\partial \sigma_i} = \sum_{k=1}^N h(x_k) p(i/x_k, \theta_i) \left(\frac{(x_k - \mu_i)^2 - \sigma_i^2}{\sigma_i^3}\right) \Big|_{\sigma_i = \hat{\sigma}_i} = 0 \quad (3.29)$$

A partir des équations 3.28 et 3.29, nous obtenons :

$$\hat{\mu}(i) = \frac{\sum_{k=1}^N \hat{h}(x_k) \hat{p}(i/x_k, \theta_i) x_k}{\sum_{k=1}^N \hat{h}(x_k) \hat{p}(i/x_k, \theta_i)} \quad (3.30)$$

$$\hat{\sigma}^2(i) = \frac{\sum_{k=1}^N \hat{h}(x_k) \hat{p}(i/x_k, \theta_i) (x_k - \hat{\mu}_i)^2}{\sum_{k=1}^N \hat{h}(x_k) \hat{p}(i/x_k, \theta_i)} \quad (3.31)$$

2. Estimation de la probabilité *a priori* :

Considérons que $p(i/x_k, \theta_i)$ ne dépend pas de $P(i)$. En appliquant le même raisonnement pour dériver l'équation 3.24 on trouve :

$$\frac{\partial \phi(X, \Theta, \Lambda)}{\partial P(i)} = \sum_{k=1}^N h(x_k) p(i/x_k, \theta_i) \frac{\partial}{\partial P(i)} \ln(p(x_k/i, \theta_i) - \lambda_1 + \lambda_i) \quad (3.32)$$

Evaluons dans un premier temps la dérivée $\frac{\partial}{\partial P(i)} \ln(p(x_k/i, \theta_i))$. L'utilisation de la relation 3.25, nous permet d'écrire :

$$\frac{\partial}{\partial P(i)} \ln(p(x_k/i, \theta_i)) = \frac{\partial}{\partial P(i)} \ln\left(\frac{p(i/x_k, \theta_i) p(x_k/\Theta)}{P(i)}\right)$$

Considérons les propriétés de la fonction logarithme, alors :

$$\begin{aligned} \frac{\partial}{\partial P(i)} \ln(p(x_k/i, \theta_i)) &= \frac{\partial}{\partial P(i)} \ln(p(i/x_k, \theta_i)) + \\ &\frac{\partial}{\partial P(i)} \ln \left(\sum_{j=1}^M p(x_k/j, \theta_j) P(j) \right) - \frac{\partial}{\partial P(i)} \ln(P(i)) \end{aligned}$$

Comme $p(i/x_k, \Theta)$ ne dépend pas de $P(i)$, alors :

$$\frac{\partial}{\partial P(i)} \ln(p(x_k/i, \theta_i)) = \frac{1}{\sum_{j=1}^M p(x_k/j, \theta_j) P(j)} \left(P(i) \frac{\partial}{\partial \theta_i} p(x_k/i, \theta_i) + p(x_k/i, \theta_i) \right) - \frac{1}{P(i)}$$

Donc,

$$\frac{\partial}{\partial P(i)} \ln(p(x_k/i, \theta_i)) = \frac{\frac{\partial}{\partial \theta_i} p(x_k/i, \theta_i)}{p(x_k/i, \theta_i)} \frac{P(i) p(x_k/i, \theta_i)}{\sum_{j=1}^M p(x_k/j, \theta_j) P(j)} + \frac{p(x_k/i, \theta_i)}{\sum_{j=1}^M p(x_k/j, \theta_j) P(j)} - \frac{1}{P(i)}$$

Ou encore,

$$\begin{aligned} \frac{\partial}{\partial P(i)} \ln(p(x_k/i, \theta_i)) &= \frac{\partial}{\partial \theta_i} \ln(p(x_k/i, \theta_i)) \frac{P(i) p(x_k/i, \theta_i)}{\sum_{j=1}^M p(x_k/j, \theta_j) P(j)} + \\ &\frac{p(x_k/i, \theta_i)}{\sum_{j=1}^M p(x_k/j, \theta_j) P(j)} - \frac{1}{P(i)} \end{aligned}$$

Par regroupement des termes $\frac{\partial}{\partial P(i)} \ln(\dots)$ on obtient :

$$\frac{\partial}{\partial P(i)} \ln(p(x_k/i, \theta_i)) \left(1 - P(i) \frac{p(x_k/i, \theta_i)}{\sum_{j=1}^M p(x_k/j, \theta_j) P(j)} \right) = \frac{p(x_k/i, \theta_i)}{\sum_{j=1}^M p(x_k/j, \theta_j) P(j)} - \frac{1}{P(i)}$$

En simplifiant, on trouve :

$$\frac{\partial}{\partial P(i)} \ln(p(x_k/i, \theta_i)) = \frac{-1}{P(i)}$$

Ainsi l'équation 3.32 devient :

$$\frac{\partial \phi(X, \Theta, \Lambda)}{\partial P(i)} = -\frac{1}{P(i)} \sum_{k=1}^N h(x_k) p(i/x_k, \theta_i) - \lambda_1 + \lambda_i |_{P(i)=\hat{P}(i)} = 0 \quad (3.33)$$

Les dérivées de (eq. 3.24) par rapport à λ_1 et λ_i sont données par :

$$\frac{\partial \phi(X, \Theta, \Lambda)}{\partial \lambda_1} = 1 - \sum_{i=1}^M P(i) = 0 \quad (3.34)$$

$$\frac{\partial \phi(X, \Theta, \Lambda)}{\partial \lambda_i} = P(i) - 1 = 0 \quad (3.35)$$

Supposons que $P(i) > 0 \quad \forall i \in [1, M]$, c'est-à-dire que les premières contraintes du problème initial (eq. 3.1) soient satisfaites et non saturées, alors :

$$\forall j \in [2, M + 1] \quad \lambda_j = 0$$

A partir de l'équation 3.33 on obtient :

$$P(i) = -\frac{1}{\lambda_1} \sum_{k=1}^N \hat{h}(x_k) p(i/x_k, \theta_i)$$

La considération de la dérivée par rapport à λ_i (eq. 3.34), nous permet d'écrire :

$$\sum_{i=1}^M P(i) = -\frac{1}{\lambda_1} \sum_{i=1}^M \sum_{k=1}^N \hat{h}(x_k) p(i/x_k, \theta_i) = 1$$

Puisque

$$\sum_{i=1}^M p(i/x_k, \theta_i) = 1$$

Alors

$$\sum_{i=1}^M P(i) = -\frac{1}{\lambda_1} \sum_{k=1}^N \sum_{i=1}^M \hat{h}(x_k) p(i/x_k, \Theta) = -\frac{1}{\lambda_1} \sum_{k=1}^N \hat{h}(x_k) = 1$$

Ainsi

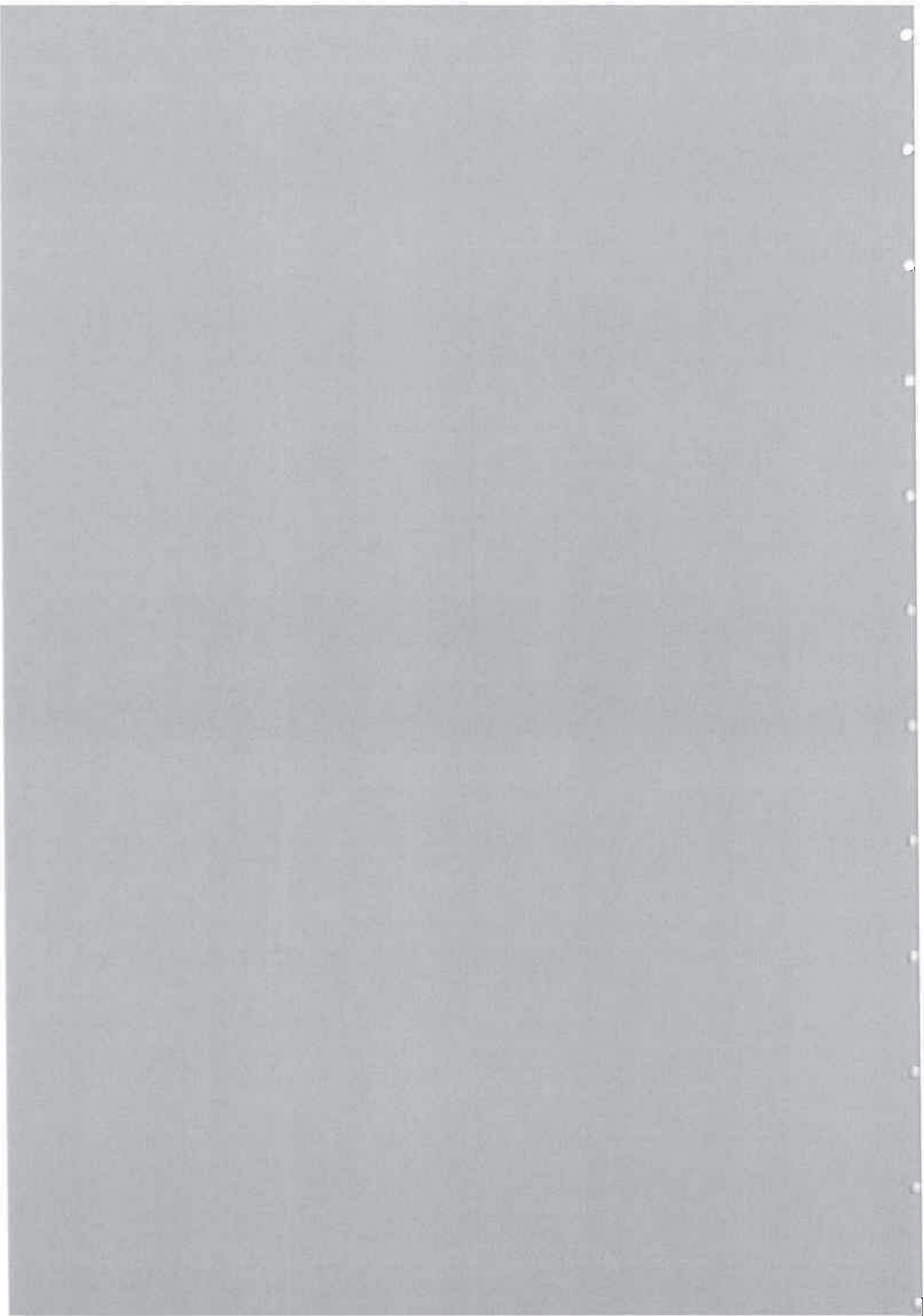
$$\lambda_1 = -\sum_{k=1}^N \hat{h}(x_k)$$

Finalement, on obtient :

$$\hat{P}(i) = \frac{\sum_{k=1}^N \hat{h}(x_k) \hat{p}(i/x_k, \theta_i)}{\hat{h}(x_k)} \quad (3.36)$$

PARTIE III

Sélection automatique de détecteurs de contours



Résumé

Cette partie décrit une approche de sélection de détecteurs et leurs paramètres les plus adéquats pour mettre en évidence un contour donné. La prise en compte de différentes sources d'informations a influencé le choix du critère de sélection et a donné lieu à une approche non combinatoire.

Le plan que nous suivons se présente ainsi :

- dans le chapitre 1, nous donnons une vue globale de l'approche qui se compose de deux phases : sélection d'un détecteur et analyse des résultats,
- le chapitre 2 est consacré à la définition du critère de sélection. Ce critère repose essentiellement sur l'analyse du contour donné et la définition de la relation de cause à effet entre le détecteur et le contour. Il est validé par l'étude d'un exemple,
- ensuite, nous décrivons dans le chapitre 3 l'étape d'analyse de résultats. Cette étape est nécessaire pour le choix de la suite du traitement à effectuer et à la correction des éventuelles erreurs de sélection. L'analyse des résultats est rendue efficace grâce à une combinaison des informations contours résultats des détecteurs sélectionnés et exécutés,
- enfin, les résultats expérimentaux et les règles de sélection utilisées sont décrits au chapitre 4. Ces règles sont déduites de l'étude menée dans la première partie de cette thèse et dans le chapitre 2.

1

Présentation globale de l'approche

Ce chapitre présente un aperçu général de l'approche choisie pour la sélection de détecteurs de contours et leurs paramètres afin de mettre en évidence un contour donné.

Après avoir posé le problème, nous présentons les différentes sources d'informations utilisées et le critère de sélection formé à partir de celles-ci. Ensuite, nous décrivons le système développé.

1.1 Position du problème

Lors du développement d'une application de vision, on est souvent confronté au problème de sélection d'un détecteur de contours parmi l'ensemble des détecteurs rencontrés dans la littérature. Rappelons qu'un détecteur de contours peut être perçu comme étant un couple formé d'un algorithme et d'un ou de plusieurs paramètres. L'approche la plus répandue consiste à choisir un détecteur d'une façon presque arbitraire et à utiliser celui-ci pour extraire les contours dans toutes les images traitées dans l'application. Le problème du choix ou de l'ajustement des paramètres est souvent négligé. En effet, les valeurs des paramètres sont, dans la plupart des cas, fixées par expérience et réutilisées pour toutes les images traitées.

Il est évident que cette approche ne peut pas conduire à des résultats satisfaisants : un seul détecteur exécuté sur une image en utilisant une seule valeur pour ces paramètres ne permet pas de fournir tous les contours (cf. chapitre 5 de la partie 1). Par conséquent, il est nécessaire d'utiliser pour une même image plusieurs détecteurs qui diffèrent par leurs échelles, leurs types, leurs propriétés mathématiques et par les propriétés géométriques du contour recherché (par exemple contour fermé, contour rectiligne, etc). Ceci nous conduit à proposer une approche que l'on peut décomposer en deux étapes essentielles :

- la sélection pour une image donnée de détecteurs adéquats et le calcul de leurs paramètres (leurs échelles),
- la fusion des résultats des différents détecteurs en une seule image de contours.

Le problème de sélection de détecteurs en vue d'une description complète et correcte de l'image est complexe dans sa généralité (cf. chapitre 5 de la partie 1). Aussi, nous n'apportons ici qu'une contribution partielle. Plus précisément,

- nous proposons un système permettant de choisir et d'utiliser les meilleurs détecteurs pour mettre en évidence un contour de positions connues à l'avance et le calcul de leurs paramètres. Nous montrons au chapitre 4 comment on peut généraliser cette approche pour extraire d'autres contours dans une image,
- nous ne considérons que les détecteurs qui sont :
 - autonomes et formés des deux étapes non dissociables : le lissage et la différenciation. Les autres modules (tels que le chaînage, fermeture de contours, approximation polygonale, squelettisation, etc.) qu'on peut associer à un détecteur sont complètement ignorés dans ce travail,
 - destinés aux contours de type marche. Les détecteurs d'autres types de contours posent des problèmes énormes au niveau de l'étape de fusion (voir paragraphe 2.6.5 de la partie 2). Ils sont donc ignorés. En revanche certains algorithmes que nous avons développés et qui font partie du système que nous proposons, à savoir l'estimation du bruit est aussi applicable à d'autres contours.

Enfin, signalons que le problème de sélection de détecteurs a été abordé par Hasegawa et al [Has 86] dans le système IMPRESS. Il est l'unique système concernant la sélection automatique de détecteurs de contours de type marche dont nous ayons connaissance. Ayant déjà présenté ce système dans la première partie (cf. 4.3.1), nous nous contentons ici de le comparer avec le notre, ce qui sera fait dans le chapitre 4.

1.2 Principe de l'approche de sélection

Avant de détailler une telle approche, il est important de décrire les informations sur lesquelles repose la sélection des détecteurs et de donner ensuite le critère de sélection.

1.2.1 Quelles informations peut-on utiliser?

Etant donné un ensemble de détecteurs et un contour, nommé par la suite *contour exemple*, il s'agit de sélectionner dans cet ensemble les détecteurs (et leurs paramètres) les plus adaptés pour mettre en évidence ce contour. La construction du couple (détecteurs, paramètres) peut être effectuée de façons différentes selon la source des informations utilisées. Ces informations peuvent être :

1. déduites des résultats effectifs des détecteurs, c'est-à-dire la conclusion de l'évaluation des résultats de l'exécution des détecteurs sur l'image contenant le contour exemple,
2. la connaissance *a priori* sur les détecteurs, en particulier leurs performances, leurs propriétés mathématiques, leurs propriétés algorithmiques et leurs objectifs,
3. les caractéristiques du contour exemple, en particulier son rapport signal/bruit, son type (double marche, simple marche), son contraste, sa largeur, sa netteté, etc.

La considération des informations portant seulement sur les résultats effectifs mène souvent à une solution coûteuse. En effet, il s'agit d'exécuter tous les détecteurs disponibles en utilisant plusieurs valeurs des paramètres. La sélection de la meilleure solution est fondée sur la comparaison des résultats obtenus et le contour donné.

L'utilisation des connaissances *a priori* sur les détecteurs et les caractéristiques des contours implique la définition de relations de cause à effets entre les détecteurs, leurs paramètres et le contour. En d'autres termes, étant données les caractéristiques d'un contour et les différentes propriétés des détecteurs, il s'agit d'établir un ensemble de règles pour désigner les détecteurs et les paramètres adéquats.

Pour ce faire, nous considérons un détecteur comme étant un couple non dissociable (algorithme, paramètres). Les paramètres prennent souvent leurs valeurs dans un intervalle infini. Nous proposons de discrétiser cet intervalle, c'est-à-dire de ne considérer pour chaque paramètre qu'un ensemble fini de valeurs possibles (voir l'exemple décrit dans le paragraphe suivant). Cet échantillonnage est motivé par le fait qu'une petite variation de l'échelle (par exemple, inférieure à 0.5 pour une gaussienne [Ber 87]) ne produit pas une différence significative dans les contours résultats.

Nous associons à chaque détecteur une précondition limitant le contexte où ce détecteur peut être utilisé avec succès. Ce contexte est défini par l'ensemble des caractéristiques du contour. A chaque contexte sont associées les performances de ce détecteur pour qualifier son exécution dans ce contexte. Les performances incluent la détection, l'erreur de localisation, le temps de calcul et un score indiquant le succès du détecteur selon son expérience dans ce contexte (cf. 2.3).

Pour mieux comprendre la structure d'un détecteur, nous considérons l'exemple décrit dans le paragraphe suivant.

1.2.2 Exemple d'un détecteur de contours

Considérons le détecteur DRF de Shen et Castan (voir paragraphe 1.3 de la partie 2). Ce détecteur possède un seul paramètre noté a et prenant ses valeurs dans l'intervalle $]0, 1[$. En pratique, dans le cas des images de type scènes d'intérieures, seules les valeurs appartenant à l'intervalle $[0.2, 0.35]$ fournissent des contours non bruités et correctement localisés. L'ensemble des valeurs retenues de a est $\{0.2, 0.25, 0.3, 0.35\}$. Les détecteurs associés sont :

$(DRF, a = 0.2)$, $(DRF, a = 0.25)$, $(DRF, a = 0.3)$, $(DRF, a = 0.35)$.

Considérons maintenant un contour de type marche simple, non bruité (le rapport contraste sur l'amplitude du bruit est supérieur à 6), nette (la pente est supérieur à 67 degrés) et lisse. Le détecteur adapté à sa mise en évidence doit vérifier alors les propriétés suivantes :

- l'opérateur de différentiation est un laplacien, car le contour est non bruité et cet opérateur fournit en général des contours lisses (cf. 3.3.2 de la partie 1),
- le lissage doit être faible pour que le contour obtenu soit bien localisé.

Par conséquent, nous proposons l'utilisation du détecteur DRF avec $a = 0.35$. Ce détecteur est non invariant par rotation. A l'exception du temps de calcul, ses performances dépendent alors de l'orientation du contour et sont symétriques par rapport aux orientations -45, 0 et 45 degrés (cf. 2.4.3). Le détecteur DRF est représenté dans ce contexte par la règle suivante :

*si le contour est une courbe lisse de type marche simple
et de rapport signal/bruit supérieur à 6*

et de netteté supérieur à 67

alors $DRF(a=0.35)$, 6,

($loc=0.001 det=1.000 exp=1 ang=0$), ($loc=0.015 det=1.000 exp=1 ang=22.5$), ($loc=0.075 det=1.000 exp=1 ang=45$).

Le temps de calcul du détecteur DRF pour traiter une image 256×256 sur un SUN4 est égale à 6 secondes. Nous associons à ce détecteur uniquement les performances dans le cas des orientations 0, 22.5 et 45. Par exemple, dans le cas où l'orientation du contour est égale à 0, l'erreur de localisation de ce détecteur est 0.001, son rapport signal/bruit vaut 1 et le score décrivant son expérience est égale à 1 (c'est la valeur affectée initialement à chaque détecteur). Ces performances sont mesurées sur un contour possédant les mêmes caractéristiques que celui décrit par la précondition de la règle ci-dessus. Dans le cas des orientations non précisées dans la règle ci-dessus, les performances sont obtenues par une interpolation (cf. 4.1).

1.2.3 Critère de sélection

D'une manière générale, la précondition d'un détecteur est constituée à partir des prédicats portant sur la forme du contour, son degré de lissage et les attributs caractérisant son profil (le type, le rapport signal/bruit, la netteté, etc.). En conséquence, une étape d'analyse du contour donné est nécessaire dont l'objectif est l'extraction des différents attributs du contour et la segmentation de celui-ci selon un critère de forme, de lissage et d'homogénéité des caractéristiques de son profil. Le résultat de la segmentation est un ensemble de contours homogènes, nommé par la suite h-contours, qui correspondent aux parties du contour exemple possédant des caractéristiques homogènes.

Soit un objectif défini par un ensemble de h-contours (tous les h-contours qui forment le contour exemple, ou seulement les h-contours qui n'ont pas encore été détectés), l'heuristique suivante est utilisée pour sélectionner un détecteur :

chercher le détecteur validé par les caractéristiques du contour et qui possède les meilleures performances, le meilleur score et qui peut détecter le maximum de vrais points de contours.

Cette heuristique n'est utilisée que pour sélectionner un seul détecteur. La sélection de tous les détecteurs nécessaires à la mise en évidence du contour exemple peut être effectuée de la manière suivante : on sélectionne tout d'abord le meilleur détecteur en considérant comme objectif le contour exemple, puis un nouveau détecteur avec comme objectif tous les h-contours qui n'ont pas été trouvés, etc.

Cette approche ne considère que les connaissances *a priori* sur les détecteurs et le contour exemple. Sa mise en œuvre présente quelques difficultés provenant essentiellement de :

- la description du contour : les irrégularités des surfaces des images rendent difficile la définition des caractéristiques pertinentes d'un contour,
- la définition de la relation détecteur-contour : la relation entre les caractéristiques d'un contour et un détecteur n'est pas bien définie. Il est difficile de connaître avec certitude dans le cas général l'influence d'un attribut d'un contour ou d'une propriété d'un détecteur sur ses performances. Ce problème a été longuement discuté au paragraphe 3.3.2 de la première partie et sera abordé au paragraphe 2.4.

En conséquence, des erreurs de sélection peuvent être introduites. Par exemple, les contours fournis par un détecteur d_1 incluent (au sens strict) tous les contours fournis par un détecteur d_2 . La sélection du détecteur d_2 est alors une erreur. Il est possible de pallier cet inconvénient en considérant également les résultats effectifs des détecteurs. Plus précisément, lors de la fusion des contours obtenus par les différents détecteurs une comparaison de ces contours est nécessaire. A partir de cette comparaison, il est possible de déterminer les détecteurs ayant été sélectionnés par erreur (cf. chapitre 3).

1.3 Description globale du système de sélection

Partant de l'approche pour la sélection décrite précédemment, nous avons conçu et implanté un système de sélection de détecteurs et de leurs paramètres pour mettre en évidence un contour similaire à celui indiqué par l'utilisateur [Zio 89a]. L'objectif de ce paragraphe est de définir l'algorithme général associé et les différents composants de ce système.

1.3.1 Schéma général de l'algorithme

Le système a pour donnée un objectif fourni par un utilisateur et décrit par :

- un contour exemple et l'image qui le contient. Ce contour est représenté par la position de ses points et est décomposé en primitives suivantes : courbes lisses ouvertes, courbes lisses fermées, droites fermées et droites ouvertes. La figure 1.1 décrit un contour exemple formé d'une courbe lisse,
- la qualité du résultat requis qui est exprimée par trois paramètres prenant leurs valeurs dans l'intervalle $[0, 1]$. Ces paramètres indiquent l'importance accordée à chacun des attributs de performance d'un détecteur qui sont la détection, la localisation et le temps de calcul nécessaire à la mise en évidence d'un contour.

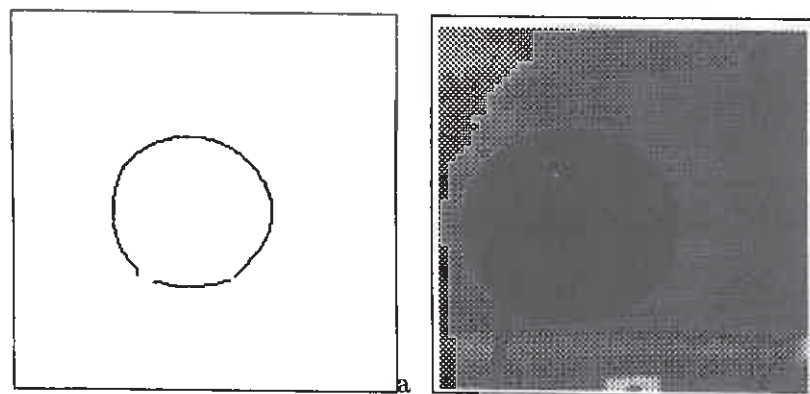


Figure 1.1. a) Un contour exemple formé d'une courbe lisse, b) l'image correspondante.

Les résultats de ce système sont les détecteurs qui ont mis en évidence le contour exemple et les caractéristiques de ce dernier.

Le schéma fonctionnel de ce système est donné par la figure 1.2).

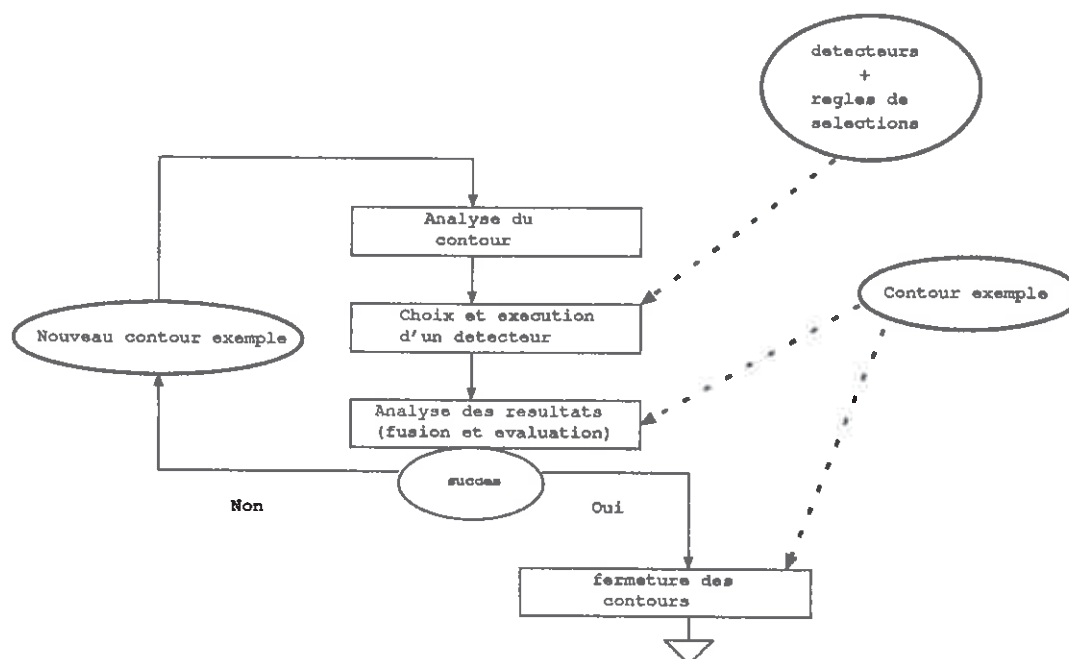


Figure 1.2. Structure de contrôle du système proposé.

1.3.2 Bibliothèque de détecteurs

La bibliothèque du système est constituée de l'ensemble des détecteurs de contours disponibles. Un détecteur de cette bibliothèque est considéré comme étant un triplet (*algorithme, paramètres, précondition*). Rappelons que la précondition est représentée par le contexte dans lequel le couple (*algorithme, ses paramètres*) s'exécutent correctement. Un algorithme est un sous-programme complètement paramétré écrit en langage C.

N'importe quel détecteur vérifiant les conditions indiquées dans la section 1.1 peut être introduit dans cette bibliothèque. L'ajout d'un détecteur doit être précédé par l'analyse de ses propriétés et l'évaluation quantitative de ses performances. Cette évaluation doit être effectuée en considérant plusieurs valeurs de ses paramètres et plusieurs contours de caractéristiques différentes.

1.3.3 Architecture du système

L'architecture du système que nous proposons découle directement du schéma fonctionnel décrit par la figure 1.2. Ce système contient les deux modules décrits dans les paragraphes suivants.

Module d'analyse de contours et de sélection d'un détecteur

Dans ce qui suit, nous appelons contour exemple, le contour fourni par l'utilisateur ou les "parties" de ce contour qui n'ont pas été encore mises en évidence. Etant donné un contour exemple, ce module a pour rôle de sélectionner dans la bibliothèque le meilleur détecteur qui peut mettre en évidence la plus grande partie de ces h-contours. La sélection

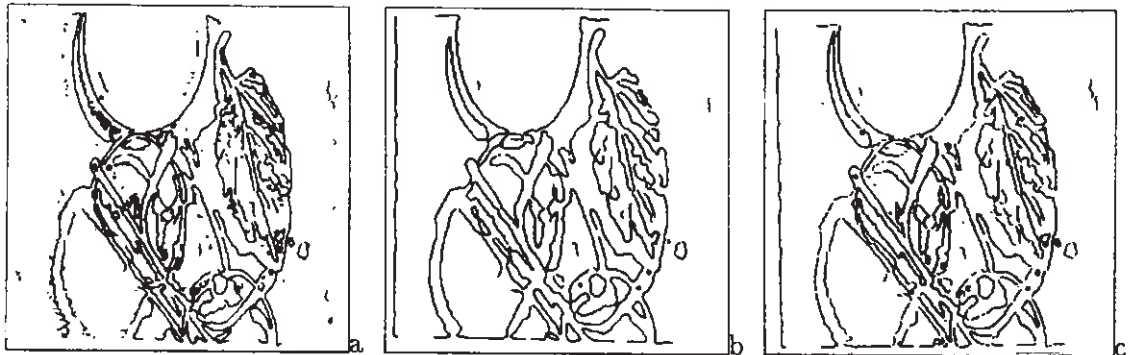


Figure 1.3. a) et b) deux images de contours obtenues par le détecteur DL (cf. 2.4) à deux échelles différentes ($\alpha = 0.75$ et $\alpha = 1.2$), c) résultat de la fusion de ces deux images.

est effectuée en trois étapes :

- description et segmentation du contour exemple. La description de ce contour a pour rôle la recherche des ses différentes caractéristiques. La segmentation consiste à le diviser en un ensemble de h-contours et à associer à chaque h-contour résultat un vecteur d'attributs formé, entre autres, de l'ensemble de ses caractéristiques,
- affectation à chaque h-contour des détecteurs applicables, c'est-à-dire ceux dont les préconditions sont validées par les caractéristiques de ce h-contour,
- choix dans l'ensemble des détecteurs applicables du meilleur détecteur en utilisant le critère décrit dans le paragraphe précédent.

Module d'analyse des résultats

Ce module a pour objectif de :

- mesurer la différence entre le contour donné par l'utilisateur et celui obtenu pour pouvoir décider de la prochaine action à mener qui peut être soit l'objectif fixé par l'utilisateur est atteint, il s'agit alors de déterminer la contribution de chaque détecteur dans la solution construite, soit au contraire cet objectif n'est pas atteint et il s'agit donc de déterminer les parties manquantes du contour donné par l'utilisateur qui seront considérées comme étant le nouveau objectif du système,
- corriger les éventuelles erreurs de sélection.

Pour une mise en œuvre efficace de cette analyse nous proposons de fusionner les contours du détecteur récemment sélectionné avec les contours résultats des détecteurs qui ont été sélectionnés avant lui. La fusion consiste à fabriquer une seule image à partir des contours obtenus par les détecteurs exécutés. Pour fixer les idées, les figures 1.3.a, 1.3.b et 1.3.c présentent respectivement deux images de contours issues d'un détecteur laplacien

utilisant le même fonction de lissage que celui de Deriche (cf. 2.4) avec les paramètres $\alpha = 0.75$ et $\alpha = 1.2$ et l'image résultat de la fusion.

Enfin, signalons que l'objectif fixé par l'utilisateur peut être considéré comme atteint, alors qu'il reste quelques "trous" de longueur faible. Nous proposons la fermeture des contours résultats de la fusion en se fondant sur la proximité, la continuité lissée de segments de contours et les caractéristiques du contour donné.

1.4 Conclusion

Nous avons introduit dans ce chapitre une approche de sélection automatique de détecteurs pour mettre en évidence un contour donné. Cette approche est fondée à la fois sur l'utilisation des connaissances *a priori* relatives aux détecteurs et sur leurs résultats ce qui lui permet d'être non combinatoire. Elle combine efficacement des détecteurs différents par leurs échelles, leurs propriétés mathématiques et algorithmiques. Son intérêt principal est le fait qu'elle établisse la relation de cause à effet entre les détecteurs et les contours. L'utilisateur a la possibilité de spécifier par l'intermédiaire des poids attribués aux différents critères de performance la qualité des résultats requis. Bien que cette possibilité soit limitée, le contour résultat peut être localisé correctement. Ainsi, la solution obtenue est le compromis entre deux caractéristiques antagonistes d'un détecteur qui sont la détection et la localisation.

Le problème majeur d'une telle approche est que le contour désigné par un utilisateur peut être subjectif (cf. 5.2 de la partie 1). Il ne correspond donc pas nécessairement à une variation du niveau de gris dans l'image. La considération d'un tel type de contours semble être difficile dans l'état actuel de nos connaissances et ne relève pas du traitement du signal comme nous le faisons ici. Nous n'apportons donc aucune réponse à ce problème.

2

Sélection d'un détecteur

L'objectif de ce chapitre est de décrire l'algorithme de sélection d'un détecteur de contours développé et intégré dans le système donné au chapitre 1 de cette partie.

Nous présentons dans le premier paragraphe le principe général de cet algorithme, et détaillons ensuite ses deux étapes qui sont l'analyse du contour exemple et le choix du meilleur détecteur.

Cet algorithme repose sur la définition de la relation de causes à effets liant les performances d'un détecteur, ses propriétés et les caractéristiques du contour. C'est pourquoi nous proposons des mesures de performances et présentons une étude de cette relation plus complète que celle donnée dans la première partie.

Enfin, les derniers paragraphes sont consacrés à la présentation des résultats expérimentaux et d'une conclusion qui met en évidence les avantages et les problèmes de cette approche de sélection.

2.1 Principe de l'algorithme de sélection d'un détecteur

Partant d'une bibliothèque de détecteurs de contours, il s'agit de choisir le meilleur détecteur pour mettre en évidence un contour donné. Le critère de sélection a été discuté dans le paragraphe 1.2.3. Rappelons que le terme contour exemple veut dire les "parties" du contour donné par l'utilisateur et dont nous n'avons pas encore déterminé les détecteurs et leurs paramètres qui peuvent les mettre en évidence. L'algorithme de sélection d'un détecteur comporte deux étapes :

- analyse du contour exemple : cette étape consiste à estimer les différents attributs du contour et à segmenter celui-ci en h-contours. Rappelons qu'un h-contour est une partie du contour exemple possédant des caractéristiques homogènes,
- choix du détecteur possédant les meilleures performances, le meilleur score et qui peut donner le maximum de points de contour.

Nous détaillons dans les deux paragraphes qui suivent chacune de ces deux étapes.

2.2 Analyse du contour exemple

Comme nous l'avons mentionné au paragraphe précédent, l'analyse du contour a pour objectif l'extraction des attributs pertinents du contour exemple et sa segmentation en

h-contours. La définition de ces d'attributs est difficile du fait que l'image est bruitée et formée de surfaces non linéaires. Nous ne considérons alors que les deux classes d'attributs suivantes :

- les attributs relatifs à la structure d'un contour, c'est-à-dire ceux qui peuvent être calculés à partir de l'intensité de l'image. On distingue, les attributs qui caractérisent le modèle de contour de type marche (c'est-à-dire le profil du contour) et ceux qui caractérisent le bruit. Les modèles de contours de type marche les plus connus sont les marches simples, les doubles marches de type crête et les doubles marches de type escalier (cf. 3.1, partie 1). La considération de ces trois modèles est une tâche importante et donne lieu à la conception d'algorithmes d'analyse propre à chaque modèle. Pour des raisons de simplicité et sachant que les doubles marches de type escalier sont relativement peu nombreuses dans une image, nous nous limitons aux marches simples et aux marches doubles de type crête,
- les attributs relatifs à la nature de la courbe support du contour (courbe lisse, courbe fermée, contour rectiligne, etc.). Ces derniers sont supposés être des données.

Partant de ces deux classes d'attributs, l'algorithme d'analyse du contour se décompose en quatre étapes :

- l'estimation de l'énergie du bruit,
- l'identification du modèle de contour qui peut être une marche simple ou une marche double de type crête,
- l'estimation des attributs relatifs au modèle du contour,
- la segmentation du contour exemple en h-contours.

Notons que l'énergie du bruit est estimée en premier parce qu'elle permet de calculer automatiquement les seuils utilisés dans l'étape d'identification du modèle de contour. Nous détaillons dans ce qui suit les quatre étapes de l'algorithme d'analyse du contour.

2.2.1 Estimation du bruit

Le bruit introduit lors de la fabrication d'une image est engendré par deux phénomènes d'origine différente. Le premier est intrinsèque à la scène. Il peut être considéré comme étant un bruit "sémantique" dans le sens où on considère généralement comme bruit toute variation engendrée par les petites irrégularités contenues dans les surfaces des objets de la scène. En général, il est difficile à caractériser et à traiter. Il est donc ignoré dans notre travail.

Le second est introduit par le dispositif de la fabrication d'images lors de son acquisition. Sous certaines hypothèses, ce type de bruit peut être mesuré et éliminé facilement, ce qui permet de restaurer l'image. Par exemple, un bruit blanc, gaussien de moyenne nulle, est complètement défini par son écart-type. Il existe plusieurs approches pour mesurer le bruit contenu dans un signal, entre autres, le filtre de Wiener [Bij 84] et l'analyse krigeante [Ma 87].

Nous proposons dans ce qui suit un algorithme pour l'estimation de la composante bruit dans un signal fondé sur le filtrage au sens de Wiener. Cet algorithme a l'avantage d'être simple et efficace. Sa justification mathématique est donnée en Annexe B. Soit $i(x)$ un signal 1D et $f(x)$ un filtre de lissage de type passe bas. L'algorithme comporte trois étapes :

1. Convolution de $i(x)$ avec la première dérivée du filtre $f(x)$. Soit $o(x)$ le signal résultat. Nous proposons l'utilisation du filtre de lissage $f(x)$ défini par :

$$f(x) = (\alpha \sin(w|x|) + w \cos(w|x|))e^{-\alpha|x|} \quad (2.1)$$

où α est le paramètre contrôlant le degré de lissage. Le choix de ce filtre est motivé par le fait qu'il lisse l'image en conservant les contours (cf. 3.2 de la partie 1). Signalons que tout autre filtre passe-bas peut être utilisé.

2. Estimation de l'énergie du bruit dans le signal de sortie par l'élimination des valeurs élevées dans la sortie $o^2(x)$, puisque les valeurs élevées correspondent généralement à l'amplitude du contour (l'amplitude du bruit est moins importante que l'amplitude d'un contour). La séparation du bruit et du signal peut être effectuée par une simple technique de seuillage consistant à construire l'histogramme de la réponse $o^2(x)$ et à prendre en compte les valeurs faibles (soit les 80% de l'histogramme).
3. Estimation de l'énergie du bruit dans le signal d'entrée. Etant données la réponse impulsionnelle du filtre $f'(x)$ et l'énergie du bruit dans la réponse de $f'(x)$, l'énergie du signal d'entrée $n^2(x)$ est donnée par :

$$n^2(x) = \frac{o^2(x)}{\int_{-\infty}^{+\infty} f'^2(x) dx} \quad (2.2)$$

L'extension de cet algorithme en 2D pour inclure l'estimation du bruit dans une image est immédiate. Il suffit de calculer la dérivée de l'image lissée dans les deux directions, celles des x et celles des y à l'aide du détecteur de Deriche (car celui-ci utilise le filtre de lissage $f(x)$ donné par l'équation 2.1, voir paragraphe 1.2 de la partie 2). L'énergie du bruit d'un point donné est alors calculée comme suit : si le point n'appartient pas au contour, l'énergie du bruit est égale à la moyenne des deux images résultats du détecteur. Dans le cas inverse, on considère comme énergie du bruit la réponse du détecteur dans la direction normale au gradient. L'étape 3 est utilisée pour chacun des deux masques séparément.

A cause de l'irrégularité de l'énergie du bruit, la mise en œuvre de la procédure qui vérifie si un point est un contour est délicate. Pour cela, nous proposons l'utilisation de la règle suivante : un point (x, y) est considéré comme contour si et seulement si la moyenne du module au carré du gradient normalisé par l'étape 3, calculée sur la fenêtre centrée sur (x, y) est supérieur à un seuil donné. La valeur du seuil est environ 80% de l'histogramme. La taille de la fenêtre est $M \times M$ où M est donné par la formule suivante :

$$3\sqrt{2}\pi / \text{echelle de } f = 3\sqrt{2}\pi \times w / \arctg(w/\alpha)$$

Un examen de cette formule montre que la dimension M de la fenêtre est inversement proportionnelle à l'échelle de f . Ce choix est motivé par le fait que si une haute échelle

(resp. petite échelle) est utilisée alors le rapport signal/bruit de la réponse de f est élevée (resp. faible) et il suffit donc de considérer un voisinage de petite taille (resp. grande taille).

Cet algorithme a été testé sur une variété d'images naturelles ou de synthèse. La figure 2.1 présente une image contenant un contour de type marche, circulaire de netteté égale à 85 degrés et de contraste égal à 50. Cette image est perturbée par un bruit gaussien de moyenne égale à zéro et présentant deux variances différentes : la variance du bruit ajouté aux pixels se trouvant au-dessus de la diagonale qui est égale à 36 (ce qui fait un rapport signal/bruit de 18.4 dB) et la variance du bruit ajouté aux autres pixels qui est égale à 25 (donc, 20 dB). Le rapport signal/bruit est calculé par la formule :

$$rsb = 20 \log_{10}(A/n_0) \quad (2.3)$$

où A est le contraste et n_0^2 l'énergie du bruit.

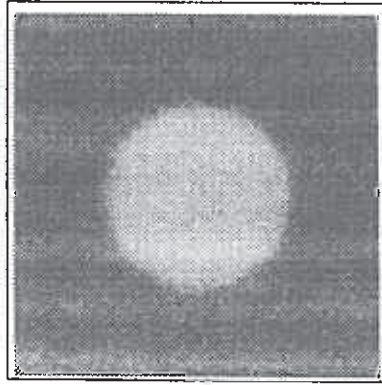


Figure 2.1. Image de synthèse de rapport signal/bruit égale à 18.4 dB pour les pixels qui se trouvent au-dessus de la diagonale et 20 dB pour ceux qui se trouvent au-dessous de la diagonale.

La table 2.1 présente les résultats de l'exécution de cet algorithme pour plusieurs valeurs de α sur l'image précédente. Signalons que le paramètre w du filtre $f(x)$ (eq. 2.1) est ignoré, c'est-à-dire qu'on se limite au cas où ce filtre possède les meilleures performances ($\alpha \gg w$). Ces résultats montrent que :

- les erreurs d'estimation du bruit sont faibles aussi bien dans la région contenant le contour que dans celle qui ne le contient pas,
- l'énergie du bruit estimé ne dépend pas de l'échelle du détecteur. En d'autres termes, le problème du choix de l'échelle du filtre de lissage ne se pose pas. Ce résultat provient de l'utilisation de l'équation 2.2 qui permet de rendre l'estimation du bruit indépendante de toutes échelles.

Canny [Can 86] propose un algorithme pour l'estimation du bruit dans un signal 1D fondé sur des idées similaires. Aucune proposition n'est formulée pour l'extension et l'utilisation de son algorithme en 2D. Une comparaison entre son algorithme et celui proposé ci-dessus montre deux différences principales. La première provient du fait que Canny utilise la dérivée seconde de son filtre pour lisser le signal d'entrée, ce qui revient à utiliser la troisième dérivée du filtre passe-bas $f(x)$. Il est bien connu que les dérivées d'ordre

α	<i>R.S.B = 18.4 dB (6)</i>			<i>R.S.B = 20 dB (5)</i>		
	<i>région non contour</i>	<i>région contour</i>	<i>image</i>	<i>région non contour</i>	<i>région contour</i>	<i>image</i>
0.8	6.30 (+0.30)	5.45 (-0.55)	6.00 (0.00)	5.37 (+0.37)	4.85 (-0.15)	5.18 (+0.18)
1.3	5.50 (-0.50)	5.80 (-0.20)	5.61 (-0.39)	4.43 (-0.57)	5.47 (+0.47)	4.84 (-0.16)
1.8	5.54 (-0.46)	6.08 (+0.08)	5.75 (-0.25)	4.46 (-0.54)	5.73 (+0.73)	4.97 (-0.03)
2.3	5.59 (-0.41)	6.31 (+0.31)	5.86 (-0.14)	4.50 (-0.50)	5.57 (+0.57)	4.93 (-0.07)

Tableau 2.1. Estimation de l'énergie du bruit de l'image contenant le cercle. Le bruit est estimé sur les pixels de l'image se trouvant au-dessus et au-dessous de la diagonale. La première ligne de ce tableau présente le rapport signal/bruit de chacune des deux parties de l'image. Le nombre entre parenthèses indique l'écart-type du bruit. Les mesures présentées dans ce tableau correspondent à l'écart-type du bruit. Les nombres entre parenthèses indiquent la valeur de l'erreur.

supérieur introduisent plus de bruit. La seconde différence provient du fait que Canny ne présente aucune méthode pour calculer l'énergie du bruit d'entrée à partir de l'énergie du bruit de sortie. Il considère comme énergie du bruit d'entrée l'énergie du bruit de sortie.

2.2.2 Identification du modèle de contour

L'identification du modèle est typiquement un problème de classification. En effet, il s'agit d'attribuer à chaque contour (ou point de contour), en se fondant sur la variation du niveau de gris, une classe qui peut être une marche simple presque idéale, une marche simple brouillée, une marche double de type crête, etc. A ce sujet Sleigh [Sle 86] propose un système à base de règles de production.

Nous proposons une approche de classification par raffinements successifs, fondée sur l'analyse de la distribution des niveaux de gris. Rappelons que nous ne considérons comme modèle de contours que les marches simples et les marches doubles de type crête. Les règles de classification d'un point appartenant au contour exemple sont données par :

- marche simple : soient moy_1 , var_1 et moy_2 , var_2 représentant la moyenne et la variance de la distribution des niveaux de gris qui se trouvent de part et d'autre du point de contour comme le montre la figure 2.2. Un point de contour est considéré comme marche simple si :

$$|moy_1 - moy_2| \geq s_m \quad \text{et} \quad (var_1 + var_2) \leq 2\sigma_b^2 \quad (2.4)$$

où s_m est le plus petit contraste accepté. Nous proposons un calcul automatique du seuil de la variance. En effet, σ_b est l'écart-type du bruit calculé par l'algorithme donné dans le paragraphe précédent. Intuitivement, cette valeur provient du fait que dans le cas idéal la variance doit être nulle. Pour un contour bruité, elle doit être de l'ordre de l'énergie du bruit.

- double marche : soient g_{max} le niveau de gris maximal, var_1 , g_{min1} et var_2 , g_{min2} la variance et le niveau de gris minimal de la distribution des niveaux de gris qui se trouvent de part et d'autre du point de contour (voir figure 2.2). Un point de contour est considéré comme double marche de type crête si la hauteur et la variance moyenne sont élevées :

$$hauteur > \sigma_b \quad \text{et} \quad (var_1 + var_2) > 2\sigma_b^2 \quad (2.5)$$

où

$$hauteur = \begin{cases} g_{max} - (g_{min1} + g_{min2})/2 & \text{s'il s'agit d'une crête} \\ g_{min} - (g_{max1} + g_{max2})/2 & \text{s'il s'agit d'une vallée} \end{cases}$$

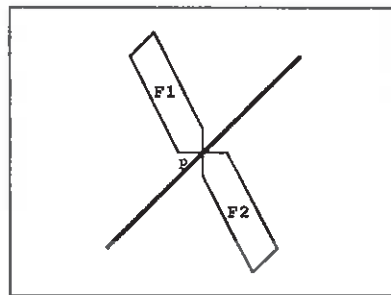


Figure 2.2. Analyse du profil d'un contour. $F1$ et $F2$ constituent le voisinage du point de contours P à partir duquel les différentes statistiques sont calculées.

La classification est effectuée au niveau du point en considérant un voisinage de quelques pixels (e.g. $(2 \times 6 + 1)$ pixels). Ce processus est répété plusieurs fois en augmentant à chaque itération la taille du voisinage jusqu'à ce qu'aucun nouveau point ne soit classé. La classification est effectuée ensuite au niveau chaîne de contours par un simple vote : la chaîne prend l'étiquette de la majorité de ses points.

L'expérimentation montre que ce processus produit des chaînes non étiquetées (cf. 2.5). Pour ce faire, nous considérons les deux règles ci-dessus et affectons chaque point à la classe dont il est le plus proche en utilisant une simple distance entre la valeur des attributs du point considéré et les valeurs limites de la classe (c'est-à-dire les seuils).

Une fois les modèles du contour exemple identifiés, il s'agit d'estimer les différents attributs de ces modèles. Nous présentons dans un premier temps l'estimation des attributs de la marche simple, puis nous exposons l'estimation des attributs de la double marche.

2.2.3 Estimation des attributs d'un contour de type marche simple

Les attributs définissant une marche simple que nous considérons sont (voir figure 2.3) :

- le contraste : cet attribut est défini par le cumul du changement d'intensité qui a lieu dans la direction normale au contour. Il caractérise la force du contour : un contour significatif doit avoir un contraste élevé et inversement.
- la largeur : la taille de l'intervalle pris dans la direction normale au contour et dans lequel le changement de l'intensité a lieu.

- la netteté : elle peut être définie par la pente de la surface appartenant à l'intervalle pris dans la direction normale au contour et dans lequel le changement de l'intensité a lieu.

Dans le cas où les surfaces images sont linéaires, il suffit de mesurer deux attributs, le troisième peut être obtenu directement (voir figure 2.3). En effet, ces trois attributs sont liés par la relation :

$$\text{contraste} = \text{nettete}' \times \text{largeur} \quad (2.6)$$

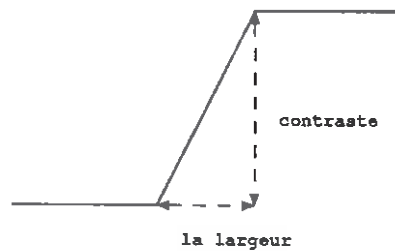


Figure 2.3. La relation entre paramètres d'une marche : largeur, le contraste et la netteté.

Cependant, les surfaces d'une image réelle ne sont pas souvent linéaires. Par conséquent, il est difficile de mesurer avec précision les différents attributs. Plus précisément, le contraste est un attribut peu sensible au changement de la forme du profil, parce qu'il ne dépend que du changement de l'intensité cumulée. En revanche, la netteté et la largeur sont plus sensibles aux irrégularités des surfaces.

Nous proposons de lisser l'image pour réduire ces irrégularités et pour pouvoir considérer les surfaces de l'image comme étant linéaires. Ce qui permet l'utilisation de la relation 2.6 en cas de nécessité. Plus précisément, nous proposons de déterminer les paramètres définissant une marche en nous fondant sur les techniques d'interpolation [The 86, Pre 88]. Il s'agit d'ajuster chaque fenêtre $N \times N$ de l'image par l'ensemble des polynômes discrets de Tchebycheff d'ordre inférieur ou égal à trois et d'en déduire le contraste et la netteté. Un tel choix de polynômes de Tchebycheff avait déjà été fait par Haralick [Har 84, Har 85a]. Les détails mathématiques de l'algorithme ci-dessus sont présentés en Annexe A. Soit $I(r, c)$ le résultat d'ajustement de la fenêtre de taille $N \times N$, nous avons :

$$I(r, c) = b_0 + b_1 r + b_2 c + b_3 r^2 + b_4 r c + b_5 c^2 + b_6 r^3 + b_7 c r^2 + b_8 r c^2 + b_9 c^3$$

La règle de calcul des coefficients b_i est donnée en Annexe A. Les attributs d'une marche simple que sont le contraste h , la netteté p sont donnés par :

$$h = 0.1202 \frac{(c_2^2 - 3c_1 c_3)^{3/2}}{3\sqrt{3}c_3^2} \quad \text{et} \quad p = 0.1163 N h^{2/3} c_3^{1/3}$$

avec $c_0 = b_0$, $c_1 = b_1 \sin(\theta) + b_2 \cos(\theta)$, $c_2 = b_3 \sin^2(\theta) + b_4 \sin(\theta) \cos(\theta) + b_5 \cos^2(\theta)$
 $c_3 = b_6 \sin^3(\theta) + b_7 \sin^2(\theta) \cos(\theta) + b_8 \sin(\theta) \cos^2(\theta) + b_9 \cos^3(\theta)$, $\theta = \arctg(b_1/b_2)$

Nous introduisons une mesure s de la validité de l'ajustement :

$$s = 0.9682 N \frac{c_3^{1/3}}{h^{1/3}}$$

Cette mesure s prend ses valeurs dans l'intervalle $[0, 1]$. Elle est d'autant plus proche de 1 que la marche est idéale.

A titre d'exemple, considérons la marche représentée dans la table 2.2 dont le contraste est égale à 64 et l'orientation du contour est égale à 34.4 degrés. Les valeurs estimées des paramètres de cette marche que nous avons obtenues pour $r = c = 5$ sont : le contraste 64.02, la netteté 88.70 degrés, la validité de l'ajustement 0.6843, l'orientation 34.1 degrés. Remarquons que les erreurs d'estimation sont faibles.

135	132	132	127	118
121	133	110	101	88
111	110	72	75	61
108	78	61	69	71
71	63	53	61	66

Tableau 2.2. Une image dont le rapport signal/bruit est égale à 18 dB. Le contraste est égale à 64. L'orientation du contour est 34.4 degrés. La distance du contour à partir du centre est égale à 0.236 pixel.

L'estimation des attributs des points du contour exemple étiquetés marche simple est effectuée à l'aide des formules ci-dessus pour chaque point en considérant un petit voisinage (e.g, 5×5). Ce processus est répété en augmentant la taille du voisinage de 2 et en considérant comme meilleure estimation celle qui maximise la valeur de s où le critère d'arrêt étant la stabilité des estimations obtenues. Pour diminuer les irrégularités des valeurs estimées le long d'un contour, un lissage de ces valeurs par une gaussienne est effectué.

2.2.4 Estimation des attributs d'une double marche

La marche double est composée de deux marches simples proches l'une de l'autre. Ce type de contour est considéré à part, car des facteurs supplémentaires interviennent lors de sa détection. En effet, la détection de l'une des deux marches peut être influencée par la détection de l'autre marche. En conséquence, sa mise en évidence ne nécessite pas uniquement la considération des attributs de chacune des deux marches, mais aussi d'autres attributs qui caractérisent les deux marches ensemble.

Nous proposons de considérer la double marche comme étant une seule entité qui peut être caractérisée par les paramètres suivants :

- la largeur : cet attribut est défini par la distance entre les deux marches,
- la hauteur : elle est défini par la différence entre le niveau de gris maximal et minimal de la double marche. La hauteur est une caractéristique de la plausibilité du contour.

- la netteté : cet attribut est définie par la netteté moyenne de chacune des deux marches ; la netteté d'une marche simple étant définie dans le paragraphe précédent,
- la symétrie : elle est définie par le rapport des deux distances entre la position de chacun des points d'inflexion et la position de la crête (c'est-à-dire, le point de l'image ayant le niveau de gris maximal). Cet attribut caractérise la différence existante entre la pente des deux marches qui forment la crête.

La détermination des ces différents paramètres est effectuée en utilisant la technique d'interpolation comme dans le cas d'une marche simple. Haralick [Har 83] propose l'utilisation de la base de polynômes de Tchebycheff de degré inférieur ou égale à trois pour l'approximation des marches doubles. L'expérimentation a montré que cette base ne permet pas souvent une approximation correcte dans le cas des doubles marches. Ainsi, nous proposons d'ajuster chaque fenêtre $N \times N$ de l'image par l'ensemble des polynômes discrets de Tchebycheff d'ordre inférieur ou égale à quatre (voir Annexe A). Soit $I(r, c)$ le résultat d'ajustement du point de la fenêtre de taille $N \times N$, nous avons :

$$I(r, c) = a_0 + a_1r + a_2c + a_3r^2 + a_4rc + a_5c^2 + a_6r^3 + a_7cr^2 + a_8rc^2 + a_9c^3 + a_{10}r^4 + a_{11}r^3c + a_{12}r^2c^2 + a_{13}c^3r + a_{14}c^4$$

La règle de calcul des coefficients a_i est donnée en Annexe A.

Les paramètres de la double marche : le contraste h , la largeur l , la symétrie s et la netteté p sont donnés respectivement par :

$$h = c_0 - \max(g_\theta(\rho_1), g_\theta(\rho_2)), \quad l = N \left| \frac{\sqrt{9c_2^2 - 24c_1c_3}}{6c_3} \right|, \quad s = \left| \frac{-3c_2 + \sqrt{9c_2^2 - 32c_1c_3}}{-3c_2 - \sqrt{9c_2^2 - 32c_1c_3}} \right|$$

$$p = \frac{c_0 - \max(g_\theta(\rho_3), g_\theta(\rho_4))}{2} \frac{-12c_3c_2}{3c_2^2 - 4c_1c_3}$$

$$\begin{aligned} \text{avec } b_0 &= a_0, & b_1 &= a_1 \sin(\theta) + a_2 \cos(\theta), & b_2 &= a_3 \sin^2(\theta) + a_4 \sin(\theta) \cos(\theta) + a_5 \cos^2(\theta), \\ b_3 &= a_6 \sin^3(\theta) + a_7 \sin^2(\theta) \cos(\theta) + a_8 \sin(\theta) \cos^2(\theta) + a_9 \cos^3(\theta), \\ b_4 &= a_{10} \sin^4(\theta) + a_{11} \sin^3(\theta) \cos(\theta) + a_{12} \sin^2(\theta) \cos^2(\theta) + a_{13} \sin(\theta) \cos^3(\theta) + a_{14} \cos^4(\theta), \\ c_0 &= b_0 - b_1 a + b_2 a^2 - b_3 a^3 + b_4 a^4, & c_1 &= b_2 - 3b_3 a + 6b_4 a^2, & c_2 &= b_3 - 4b_4 a, & c_3 &= b_4, \\ \theta &= \mp 0.5 \arctg(a_5 / (a_6 - a_4)), & g_\theta(\rho) &= c_0 + c_1 \rho^2 + c_2 \rho^3 + c_3 \rho^4 \end{aligned}$$

où a est la translation de I tel que son maximum soit à l'origine. Le signe négatif de θ est choisi quand il s'agit d'une crête et le signe positif quand il s'agit d'une vallée.

A titre d'exemple, considérons la double marche représentée dans la table 2.3 dont les attributs sont : l'orientation 0 degré, le contraste 50, la largeur 4 pixels, la netteté 87.71 degrés et la symétrie 1. Les valeurs estimées des paramètres de cette marche sont dans le cas ou $r = c = 3$: l'orientation 0.006 degrés, le contraste 58.06, la largeur 4.39 pixels, la netteté 89.08 degrés et la symétrie 1.02.

L'algorithme d'estimation des attributs d'une marche double est similaire à celui donné dans le paragraphe précédent. La qualité de l'estimation est mesurée par l'erreur quadratique de l'ajustement. Notons que le calcul des attributs d'une marche double en utilisant les formules ci-dessus est plus compliqué que celui d'une marche simple. Tous les détails nécessaires à son implantation sont donnés en Annexe A.

47	48	54	42	47	56	50
45	50	54	58	48	49	46
76	79	82	78	81	80	80
98	103	107	100	100	105	96
76	82	80	82	82	81	80
50	50	40	52	49	53	52
51	50	55	47	47	49	52

Tableau 2.3. Une image dont le rapport signal/bruit est égal à 6 dB. Les paramètres de cette crête sont : le contraste égal à 50, la largeur à 4, la symétrie à 1 et la netteté à 87.71, l'orientation étant égale à 0 degré.

2.2.5 Segmentation du contour exemple

Il s'agit maintenant de segmenter le contour exemple en h-contours et d'associer à chacun des h-contours un vecteur d'attributs. La segmentation du contour est réalisée en utilisant un critère fondé sur un ou plusieurs attributs particuliers. Les contours résultats sont segmentés à leur tour en utilisant un autre critère, etc. Les trois critères suivants sont utilisés :

1. Le premier critère porte sur la forme du contour. Rappelons que le contour donné par l'utilisateur, noté C , est supposé composé d'un ensemble de primitives, notées, $\{c_1, \dots, c_n\}$, où chaque c_i peut être soit une courbe lisse fermée, soit des droites fermées, soit une courbe lisse non fermée, soit des droites non fermées. Le contour exemple, noté O , est segmenté selon ce critère de forme et de lissage par une simple mise en correspondance avec le contour C , soit $\{o_1, \dots, o_m\}$ le résultat. Chaque o_i est composé d'un ensemble de segments.
2. Le second critère repose sur le modèle de contour. Il s'agit de diviser chaque o_i en h-contours tel que chaque h-contour résultat appartienne à une et une seule classe (marche simple, marche double de type crête). Une fois la division finie, les attributs de chaque h-contour sont calculés et lissés. Notons que la segmentation du contour exemple selon ce critère doit être effectuée avant l'estimation des attributs définissant le modèle de contour.
3. Enfin, le troisième critère est fondé sur les attributs de profil (rapport signal/bruit, netteté, largeur). Les h-contours résultats de l'étape précédente sont segmentés en adoptant un algorithme classique de segmentation en régions. L'algorithme que nous avons utilisé est inspiré de celui proposé dans [WD 88] et qui procède par fusion de régions. Le principe de cet algorithme se résume par : chaque h-contour résultat de l'étape précédente est divisé selon un critère de dynamique des attributs contour et longueur (les h-contours de petite longueur ne sont pas divisés). Les attributs considérés dépendent du modèle du contour. Les h-contours résultats de la division sont fusionnés ensuite selon un critère de proximité, de longueur et de moyenne des attributs. Les h-contours résultats de cette dernière étape sont fusionnés selon un

critère de proximité et de variance des attributs. Lorsque plusieurs h-contours sont candidats pour la fusion, le meilleur couple au sens du critère utilisé est retenu. Ces critères sont détaillés au paragraphe 2.5.

A chacun des h-contours est associé un vecteur d'attributs dont les composantes sont la forme, le modèle, le rapport signal/bruit, la netteté et la largeur. Le modèle et la forme sont déduits directement à partir du σ_i qui les a engendrés. En revanche, les autres attributs sont obtenus à partir des attributs des points qui constituent ce h-contour par une simple moyenne.

Nous venons de présenter un algorithme pour l'analyse du contour exemple. Cet algorithme comprend l'estimation des caractéristiques relatives à la structure d'un contour, l'identification des modèles de ce contour et sa segmentation en h-contours. Nous avons donné jusqu'à maintenant que les résultats de certaines composantes de cet algorithme. Pour des raisons de lisibilité, nous préférons étudier au paragraphe 2.5 un exemple comprenant tous les aspects de la sélection d'un détecteur où nous montrons les résultats de cet algorithme d'analyse du contour.

2.3 Choix du détecteur

Après avoir analysé le contour exemple, il s'agit de choisir le détecteur validé par les caractéristiques du contour et possédant les meilleures performances. L'algorithme de choix d'un tel détecteur comporte deux étapes :

- la recherche des détecteurs applicables : cette étape a pour objectif la recherche de l'ensemble des détecteurs qui sont validés par les caractéristiques des h-contours obtenus dans l'étape précédente,
- le choix dans l'ensemble des détecteurs applicables du meilleur.

2.3.1 Choix des détecteurs applicables

En général, les détecteurs existants n'incluent ni le contexte précis dans lequel ils peuvent être utilisés ni la façon de calculer leurs paramètres. Un détecteur de contours est traditionnellement représenté par une fonctionnelle $f(I(x, y), p_1, p_2, \dots, p_n)$, où $I(x, y)$ est l'image à traiter et p_1, p_2, \dots, p_n sont les paramètres. Nous proposons une extension de la notion de détecteur pour inclure son contexte d'utilisation. En d'autres termes, il s'agit d'établir une relation de cause à effet entre les caractéristiques d'un contour et un détecteur. Nous considérons alors le détecteur comme étant un triplet : un algorithme de détection de contours, ses paramètres et une précondition limitant son domaine d'application. Ces composantes ont été présentées dans le paragraphe 1.2. Un détecteur est alors représenté par une règle de la forme :

$$\text{si } (situation_1, v_{11}, \dots, v_{1m}) \vee \dots \vee (situation_n, v_{n1}, \dots, v_{nm}) \\ \text{alors } (algorithme, param\grave{e}tres)$$

où *situation* est une conjonction de prédicats portant sur les caractéristiques du contour, v_{ij} est un vecteur de performances du couple (algorithme, paramètres) dans la

$i^{\text{ème}}$ situation, mesurées dans l'orientation j et m est le nombre d'orientations considérées. Un exemple de détecteur est présenté dans le paragraphe 1.2.

Etant donné un contour, la sélection d'un détecteur passe tout d'abord par la recherche de tous les détecteurs qui peuvent être utilisés. Soit $C = \{c_1, c_2, \dots, c_n\}$ le contour exemple où c_i est un h-contour :

- à chaque c_i sont affectés les détecteurs qui sont susceptibles de le mettre en évidence. Cette affectation est effectuée par la recherche des règles dont la précondition est validée par les caractéristiques de c_i ,
- à chaque segment de droite de c_i sont affectés les détecteurs de contours rectilignes validés par les caractéristiques de c_i ,
- si le regroupement d'un ensemble de c_i , est un contour fermé et s'il existe des détecteurs validés par les caractéristiques de c_i , alors ces détecteurs sont retenus.

La recherche des détecteurs dont les préconditions sont valides est effectuée par un algorithme classique de type chaînage avant inspiré du système PENELOP [Voy 87].

2.3.2 Sélection du meilleur détecteur

Soient D l'ensemble de détecteurs applicables, c_i un h-contour et d un détecteur validé par les caractéristiques de c_i . Trois facteurs interviennent dans le processus de sélection :

- la détection : elle indique la probabilité pour que le détecteur s'exécute avec succès dans une situation particulière. L'estimation de cette probabilité fait intervenir les facteurs suivants :
 - le score : il indique le taux de réussite du détecteur dans cette situation particulière en se fondant sur son historique. Ce score peut être mis à jour de la façon suivante : le pourcentage du contour promis et non détecté est calculé et le score est mis à jour par une moyenne pondérée favorisant le résultat le plus récent.
 - l'exhaustivité : cette mesure est le résultat d'une évaluation quantitative. Elle est inversement proportionnelle aux différentes erreurs de détection commises par un détecteur (voir paragraphe 2.4.2) :

$$exhaustivité = 1 - (e_o + e_a + e_s) / 3$$

Les deux attributs score et exhaustivité sont similaires (ils apportent presque la même information). Cependant, ils proviennent de deux origines différentes. Ils peuvent être combinés pour fournir une seule mesure. Pour un c_i , cette mesure est définie par :

$$t(d, i) = \alpha_1 \text{score}(d, i) + (1 - \alpha_1) \text{exhaustivite}(d, i)$$

où α_1 est un scalaire compris entre zéro et un pour indiquer l'importance accordée au score relativement à l'exhaustivité.

- la longueur du h-contour susceptible d'être mis en évidence par ce détecteur. Cette mesure exprime la "promesse" donnée par le détecteur. La promesse du détecteur d pour mettre en évidence un contour c_i peut être mesurée par :

$$promesse(d, i) = \frac{lg(c_i)}{l_e}$$

où $lg(c_i)$ est la longueur de c_i et l_e la longueur de tous les h-contours appartenant au contour exemple.

La probabilité du succès d'un détecteur d pour mettre en évidence c_i est une combinaison linéaire des deux mesures précédentes :

$$p(d, i) = \alpha_2 t(d, i) + (1 - \alpha_2) promesse(d, i)$$

Cette mesure est d'autant plus proche de un que le détecteur fournit une "grande" promesse et qu'il possède des bonnes performances et un score élevée. On peut généraliser cette mesure pour inclure tous les c_i qui ont validé le détecteur d (soit m h-contours). Au lieu de considérer la probabilité relative au succès d'un détecteur, nous utilisons celle relative à son échec. Cette dernière est donnée par :

$$P(d) = \frac{\sum_{i=1}^m (1 - p(d, i))}{\sum_{a \in D} P(a)}$$

- la localisation : soit $lc(d, i)$ l'erreur de localisation d'un détecteur d dans la situation i . La mesure de localisation peut être définie par :

$$loc(d, i) = \frac{lc(d, i)}{\sum_{a \in D} lc(a, i)}$$

Cette mesure peut être généralisée pour inclure tous les c_i de la même façon :

$$L(d) = \frac{\sum_{i=1}^m lc(d, i)}{m}$$

- le temps de calcul : soit $tc(d)$ le temps de calcul d'un détecteur d , la mesure du temps de calcul est supposée indépendante de c_i . Elle est définie par :

$$tc(d) = \frac{tc(d)}{\sum_{a \in D} tc(a)}$$

La même démarche est utilisée pour les détecteurs de contours rectilignes et fermés en considérant respectivement comme primitives les segments de droite composant chacun des c_i et l'ensemble des c_i formant un contour fermé.

La sélection du meilleur détecteur se limite alors à la recherche du détecteur d^* de D qui minimise la combinaison linéaire des trois facteurs ci-dessus et qui est définie par :

$$c(d) = \alpha P(d) + \beta L(d) + (1 - \alpha - \beta)tc(d) \quad (2.7)$$

Les scalaires α et β décrivent l'objectif défini par l'utilisateur (cf. 1.3). Ils contrôlent la qualité du contour résultat et influencent donc le choix des détecteurs.

En pratique, le nombre de détecteurs et le nombre de h-contours étant relativement faible, la solution à ce problème d'optimisation (eq. 2.7) est immédiate : pour chacun des détecteurs retenus dans l'étape de validation, cette mesure est évaluée et le détecteur possédant la valeur minimale est retenu.

2.4 Description de détecteurs

L'approche de sélection que nous avons présentée précédemment nécessite la définition de la relation entre les propriétés d'un détecteur, ses performances et les caractéristiques d'un contour. Ce paragraphe a pour objectif la définition des lois générales pour décrire cette relation. Il constitue la suite de l'étude menée dans le chapitre 3 de la première partie. Pour ce faire, nous considérons trois détecteurs de contours : le détecteur de Deriche (nous l'appelons par la suite DGD, détecteur gradient de Deriche), le détecteur laplacien DRF de Shen et Castan et un détecteur que nous utilisons dans le cadre des travaux sur la détection multi-échelle de contours [Zio 91b, Tab 91a]. Ce dernier utilise le même filtre de lissage que DGD et le même opérateur de différentiation que DRF. Il est appelé par la suite DL (détecteur laplacien). Ce détecteur est détaillé dans les travaux [Tab 90, Der 90a].

Ces trois détecteurs sont implantés par des équations aux différences. En conséquence, par comparaison des performances des trois détecteurs, il est possible d'en déduire l'influence sur le contour détecté de chacun des deux modules : le filtre de lissage et l'opérateur de différentiation.

Nous nous intéressons dans ce qui suit aux caractéristiques pertinentes d'un détecteur de contours. On distingue deux classes d'attributs : les attributs intrinsèques et les attributs de performance. Nous réalisons ensuite une étude sur l'influence des caractéristiques du contour et les propriétés mathématiques d'un détecteur sur ses performances.

2.4.1 Attributs intrinsèques d'un détecteur

On appelle attribut intrinsèque d'un détecteur, tout attribut qui porte sur les propriétés mathématiques, les propriétés algorithmiques et l'objectif d'un détecteur. Les propriétés mathématiques d'un détecteur sont celles relatives aux propriétés du filtre de lissage et de l'opérateur de différentiation. Ces propriétés ont été longuement discutées dans la première partie de cette thèse. Nous les rappelons brièvement :

- les propriétés de l'opérateur de différentiation : opérateur linéaire ou non linéaire, opérateur invariant ou non par rotation.
- les propriétés du filtre : la durée de sa réponse impulsionnelle (filtre à bande limitée, filtre à support limité, filtre à incertitude minimale) et son invariance ou non invariance par rotation.

En plus des attributs relatifs aux propriétés mathématiques, il existe d'autres caractéristiques qui ne sont pas déductibles à partir des propriétés mathématiques. En particulier :

- l'objectif du détecteur : cet attribut peut être défini par certaines caractéristiques du contour susceptible d'être mis en évidence par le détecteur. En particulier, le type du contour (marche, ligne, etc.) et sa forme (rectiligne, fermée, autre).
- la complexité du détecteur : elle peut être mesurée par le nombre d'opérations élémentaires par pixel. Pour un détecteur donné, cette mesure n'est pas toujours facile à réaliser. Il est alors commode de considérer le temps de calcul par pixel (ou par groupement de pixels) sur une machine particulière.

Selon l'objectif des détecteurs, nous dégageons les trois classes suivantes :

- les détecteurs de contours fermés : cette classe englobe les algorithmes classiques de seuillage d'image à niveaux de gris. Il est bien connu que ce type d'algorithme fournit l'information région. Cependant, si l'image est simple (c'est-à-dire que l'histogramme associé est composé de modes distincts), les frontières de régions correspondent aux contours (cf. 3.1 de la partie 2). Un détecteur laplacien invariant par rotation fournit aussi des contours fermés. En général, ceci n'est valable que s'il s'agit d'un "petit" contour fermé. Les détecteurs de contours fermés que nous considérons dans ce travail sont les algorithmes de seuillage,
- les détecteurs de contours rectilignes : il s'agit de l'ensemble des détecteurs des contours qui sont utilisés seulement dans une direction particulière. Canny [Can 83] montre que l'utilisation d'un détecteur dans la direction du gradient permet d'augmenter le rapport signal/bruit et de diminuer l'erreur de localisation. Un tel détecteur, appelé par la suite détecteur directionnel, se compose en général d'un filtre directionnel de lissage normal à la direction du gradient et d'un opérateur de différentiation parallèle à cette direction.
- les détecteurs de contours généraux : cette classe contient tous les détecteurs qui n'appartiennent pas aux deux classes précédentes.

2.4.2 Attributs de performance

Les attributs intrinsèques d'un détecteur précédemment définis influencent directement les qualités du contour détecté. Cependant, il n'est pas souvent facile de définir analytiquement une relation de cause à effet entre ces attributs, les caractéristiques du contour et les performances d'un détecteur. C'est pourquoi nous procédons généralement d'une manière empirique en définissant un ensemble de mesures pour caractériser ses performances. Cette technique est qualifiée d'évaluation quantitative (cf. 5.3.3 de la partie 1).

A ce sujet, rappelons que Pratt [Pra 78] propose une mesure de performance appelée *merit figure* qui est une combinaison de trois erreurs : l'oubli de marquer un vrai point de contour, l'étiquetage contour d'un faux point de contour et le déplacement entre le vrai point de contour et le point détecté. Cette mesure ne fournit qu'une seule valeur. En conséquence, il est difficile de connaître le type d'erreur commise par le détecteur. Par exemple, la mesure de Pratt fournit la même valeur pour les deux contours présentés dans les figures 2.4.b et 2.4.c. Elle présente ainsi un cas de confusion entre l'erreur de localisation (fig. 2.4.c) d'une part et les erreurs d'oubli et de réponses multiples (fig. 2.4.b) d'autre part.

Par ailleurs, il est difficile en général de mesurer les performances d'un détecteur si l'image de test inclut des contours qui ne soient pas rectilignes horizontaux et verticaux. Ceci est dû à la géométrie du plan image (la forme du pixel, la définition de la trame, etc.).

Pour pallier ces deux inconvénients, nous proposons l'utilisation de plusieurs mesures distinctes en considérant le plan image comme réel et en introduisant les trois classes de contours notées ambiguë, fausse et vraie ci-dessous.

Soient C le contour calculé, E_t l'ensemble des pixels traversés par C , E_c une partie de E_t formée des pixels qui sont divisés par C en deux parties importantes et E_a l'ensemble des pixels voisins (au sens des 8 connexités) à au moins un pixel de E_t (voir figure 2.5). Un point de contour p est dit :

- ambigu si $p \in (E_t \ominus E_c) \cup E_a$, où \ominus est la différence ensembliste,
- faux si $p \notin E_t \cup E_a$,
- vrai s'il n'est ni ambigu ni faux.

Les mesures de performance sont de deux types : les mesures concernant l'exhaustivité des points de contour et une mesure qui concerne l'erreur de localisation.

- l'exhaustivité : elle est caractérisée par les trois attributs suivants :
 - erreur d'omission : cette erreur pénalise un détecteur qui oublie de marquer les vrais points de contour. Elle est définie par le rapport entre le nombre de points oubliés et le nombre de vrais points de contours :

$$e_o = \frac{\text{nombre total de points oubliés}}{\text{nombre total de vrais points de contour}}$$

Le nombre de points de contour oubliés est la différence entre le nombre de points attendus et le nombre de vrais points détectés.

La définition de l'erreur d'omission n'inclut pas la répartition spatiale des points oubliés. Dans la pratique, il est plus facile de corriger des trous de quelques pixels qui se répètent le long d'un contour que de corriger un seul trou de plusieurs pixels. On introduit alors une mesure supplémentaire pour pénaliser un détecteur produisant des contours dont les trous sont importants :

$$e_t = \frac{\text{longueur moyenne des trous}}{\text{longueur moyenne des trous} + \text{longueur moyenne des contours}}$$

- erreur d'ambiguïté : elle pénalise un détecteur à réponse multiple. Elle peut être mesurée par :

$$e_a = \frac{\text{nombre total de points ambigus}}{\text{nombre total de vrais points de contour}}$$

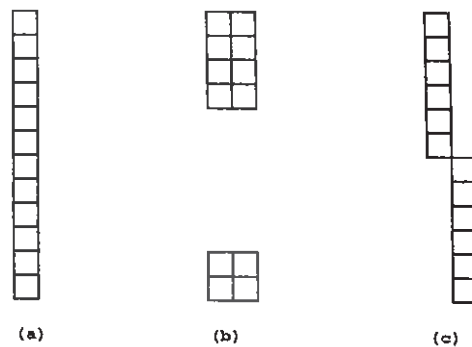


Figure 2.4. Un exemple de confusion par la mesure de Pratt. a) représente le contour idéal, b) et c) deux contours calculés. La mesure de Pratt fournit la même valeur pour ces deux contours.

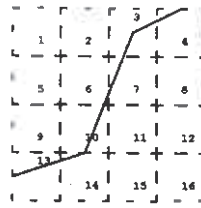


Figure 2.5. Un exemple de pixels traversés par un contour. Les pixels sont numérotés. $E_t = \{3, 4, 6, 7, 10, 13, 14\}$, $E_c = \{3, 7, 10, 13\}$ et $E_a = \{1, 2, 5, 8, 9, 11, 12, 15\}$.

- erreur de sensibilité : elle pénalise un détecteur sensible au bruit contenu dans l'image.

$$e_s = \frac{\text{nombre total de faux points}}{\text{nombre total de points de contour}}$$

- la localisation : elle est définie par le déplacement moyen par pixel ; la distance moyenne entre les vrais points de contour et le contour attendu.

$$l = \frac{\text{somme des déplacements}}{\text{nombre total de vrais points de contour}}$$

Ces mesures sont utilisées pour sélectionner un détecteur et forment la base de l'étude qui suit.

2.4.3 Influence de l'orientation d'un contour sur les performances d'un détecteur

Pour mieux cerner cette influence, nous introduisons la notion d'invariance par rotation d'un détecteur de contours. Dans le cas où l'estimation du module et de l'orientation du contour ne dépend pas du sens de celui-ci, alors le détecteur est dit invariant par rotation. A l'opposé, un détecteur non invariant par rotation a une préférence dans une orientation particulière [Dan 90].

Il s'agit maintenant d'examiner les conséquences de cette propriété sur les performances d'un détecteur. Pour ce faire, nous considérons trois détecteurs gradient : celui de Deriche, de Canny et le détecteur GEF de Shen et Castan. Nous nous limitons aux détecteurs gradient, puisque cette étude est fondée sur des critères de performance exprimés en 2D (obtenus par extension des critères de Canny) qui sont spécifiques à ce type de détecteur. Nous étudions dans un premier temps l'influence de l'orientation du contour sur les performances de ces trois détecteurs et généralisons ensuite le résultat obtenu. Notons que seuls les résultats sont présentés dans ce paragraphe. Les détails mathématiques et les démonstrations sous-jacentes sont donnés en Annexe C. Le modèle du contour considéré est donné par :

$$c(x, y, \theta) = \begin{cases} A & \text{si } x \cos(\theta) + y \sin(\theta) \geq 0 \\ 0 & \text{sinon} \end{cases}$$

où A est le contraste et θ l'orientation du contour supposée appartenir à l'intervalle $[0, \pi/2]$ (le cas des orientations $\theta \in]0, -\pi/2]$ est symétrique).

Le détecteur DGD

Ce détecteur est non invariant par rotation. L'influence de l'orientation du contour θ sur le rapport signal/bruit, noté N_{sb} et la localisation, notée N_{loc} , est donnée par :

$$N_{sb}(\theta) = \frac{1 + 3\sin(\theta)\cos(\theta)}{(\sin(\theta) + \cos(\theta))^3}, \quad N_{loc}(\theta) = \frac{\sqrt{\sin^6(\theta) + \cos^6(\theta)}}{(\sin(\theta) + \cos(\theta))^3}$$

Ces deux fonctions possèdent un minimum au point $\theta = \pi/4$. Elles sont symétriques par rapport à $\theta = \pi/4$, car $N_{sb}(\pi/2 - \theta) = N_{sb}(\theta)$ et $N_{loc}(\pi/2 - \theta) = N_{loc}(\theta)$. Il en résulte que les performances du détecteur de Deriche sont affectées par l'orientation du contour. La figure 2.6 décrit la variation du rapport signal/bruit et de la localisation en fonction de θ .

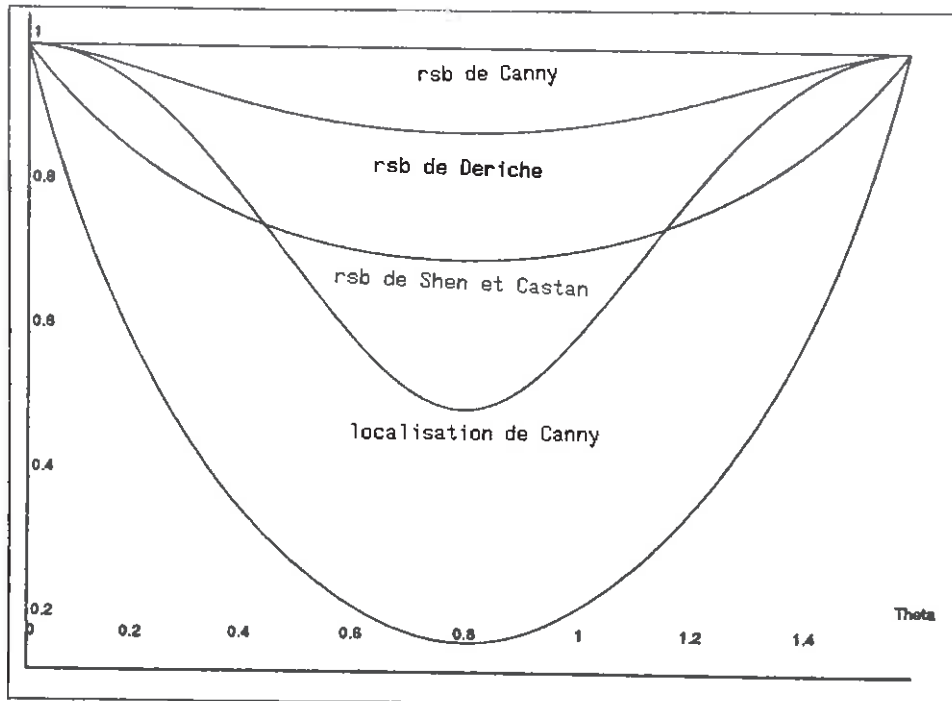


Figure 2.6. La localisation et le rapport signal/bruit des détecteurs de Deriche, de Canny, et de Shen et Castan en fonction de l'orientation du contour.

Détecteur de Canny

Le détecteur de Canny est invariant par rotation. L'influence de θ sur ses performances est donnée par :

$$N_{sb}(\theta) = 1, \quad N_{loc}(\theta) = \sqrt{\sin^6(\theta) + \cos^6(\theta)}$$

Ces deux expressions montrent que :

- le rapport signal/bruit du détecteur de Canny est indépendant de l'orientation du contour,

- bien que ce détecteur soit invariant par rotation, sa localisation dépend de l'angle θ . L'erreur de localisation est maximale au point $\theta = \pi/4$. La fonction $N_{loc}(\theta)$ est symétrique par rapport à $\theta = \pi/4$. La figure 2.6 décrit la variation de la localisation en fonction de θ .

Détecteur GEF de Shen et Castan

Ce détecteur est non invariant par rotation. L'influence de θ sur ses performances est donnée par :

$$N_{sb}(\theta) = \frac{1}{\cos(\theta) + \sin(\theta)}$$

Dans le cas d'un contour de marche idéale l'erreur de localisation du détecteur GEF est égale à zéro. Elle n'est donc pas affectée par l'orientation du contour.

Cependant, le rapport signal/bruit atteint son niveau le plus faible au point de symétrie $\theta = \pi/4$ (voir figure 2.6).

Résultat général

D'une manière générale, nous démontrons le résultat suivant (cf. Annexe c) : la localisation de tout détecteur utilisant $(\frac{\partial}{\partial x})^2 + (\frac{\partial}{\partial y})^2$ comme opérateur de différentiation est affectée par l'orientation du contour.

L'influence de l'orientation sur les performances d'un détecteur de contours se résume par :

- le rapport signal/bruit et la localisation d'un détecteur gradient variant par rotation sont affectés par l'orientation du contour. En général, les performances d'un tel détecteur se détériorent et atteignent leur minima aux points $\theta = -\pi/4, \pi/4$. Elles sont symétriques par rapport aux orientations $\theta = -\pi/4, 0, \pi/4$.
- pour un détecteur gradient invariant par rotation, seule la localisation est affectée par l'orientation du contour. Elle est minimale aux points $\theta = -\pi/4, \pi/4$ et est symétrique par rapport aux orientations $\theta = -\pi/4, 0, \pi/4$.

Contrairement à la définition donnée par M. Brady et B.K. Horn [Bra 83], l'invariance par rotation d'un détecteur signifie seulement que l'erreur d'estimation du module et de l'orientation du contour est indépendante de son sens. Les performances d'un détecteur utilisant comme opérateur de différentiation le gradient (que ce soit $(\frac{\partial}{\partial x})^2 + (\frac{\partial}{\partial y})^2$ ou $\max(|\frac{\partial}{\partial x}|, |\frac{\partial}{\partial y}|)$ ou $|\frac{\partial}{\partial x}| + |\frac{\partial}{\partial y}|$) sont affectées par l'orientation du contour.

Enfin, notons que les détecteurs DL et DRF sont non invariants par rotation puisqu'ils utilisent les mêmes filtres de lissage que les détecteurs DGD et GEF.

2.4.4 Influence des attributs image d'un contour sur les performances d'un détecteur

Nous présentons dans ce paragraphe l'influence des attributs relatifs à la structure de l'intensité image dans le voisinage d'une marche simple sur les performances d'un détecteur. Nous étudierons ensuite le cas de la marche double.

Cas d'une marche simple :

Soit un modèle 1D de contour de type marche simple formé par un signal $F(x)$ et un bruit blanc additif $N(x)$. Supposons que $F(x)$ soit définie par :

$$F(x) = \begin{cases} A(1 - \frac{1}{2}e^{-\mu x}) & \text{si } x \geq 0 \\ \frac{A}{2}e^{\mu x} & \text{sinon} \end{cases} \quad (2.8)$$

où A et μ sont des réels positifs. Ce contour est une marche brouillée (on peut vérifier aisément que si $\mu \rightarrow \infty$, alors le profil résultat correspond à une marche idéale).

Considérons comme mesure de performance le rapport signal/bruit et la localisation définis dans la partie 2 par les équations 1.2 et 1.3. Examinons à titre d'exemple les performances du détecteur DGD en 1D. Le rapport signal/bruit et la localisation sont respectivement donnés par :

$$\Sigma = Sb_i \frac{\sqrt{2}}{\sqrt{\alpha}} - Sb_i \alpha \sqrt{2\alpha} \frac{1}{(\alpha + \mu)^2} \quad (2.9)$$

$$\Lambda = Sb_i \frac{\sqrt{2\alpha\mu^2}}{(\alpha + \mu)^2} \quad (2.10)$$

où $Sb_i = \frac{A}{n_0}$ est le rapport signal/bruit de l'image originale. Selon la valeur des variables n_0 et μ on constate que :

- l'augmentation de l'énergie du bruit n_0 a pour conséquence une diminution du rapport signal/bruit du filtre et une augmentation de l'erreur de localisation,
- pour un contour de type marche idéale ($\mu \rightarrow \infty$), le second terme de l'équation 2.9 est nul. La présence de μ entraîne alors une diminution du rapport signal/bruit. Un examen de l'équation 2.10 indique qu'une diminution de μ entraîne une augmentation de l'erreur de localisation.

Soit l'image donnée dans la figure 2.7.a, fabriquée à partir du modèle de contour 2.8. A cette image est ajouté un bruit blanc de variance différente pour le triangle formé des pixels qui se trouvent au-dessus de la diagonale et le triangle inférieur (cf. 2.2.1). Les figures 2.7.b, 2.7.c, 2.7.d présentent les contours résultats des trois détecteurs DGD, DRF et DL exécutés sur l'image dont le paramètre $\mu = 1$. Le paramètre α de DGD et DL vaut 1, le paramètre a de DRF vaut 0.3.

sb = 18.4 dB	Image ($\mu = 0.5$)			Image ($\mu = 1$)		
	DGD	DRF	DL	DGD	DRF	DL
erreur						
omission	0.07	0.00	0.18	0.04	0.00	0.15
ambiguïté	0.1	1.01	0.99	0.06	0.32	0.36
sensibilité	0.87	0.91	0.90	0.88	0.92	0.93
localisation	0.1	0.31	0.38	0.06	0.27	0.28

Tableau 2.4.

Examinons les performances de ces trois détecteurs exécutés sur une image de synthèse similaire à celle donnée dans la figure 2.7.a. Les tables 2.4 et 2.5 montrent les performances des trois détecteurs en fonction de la netteté μ et le rapport signal/bruit Sb . Le paramètre

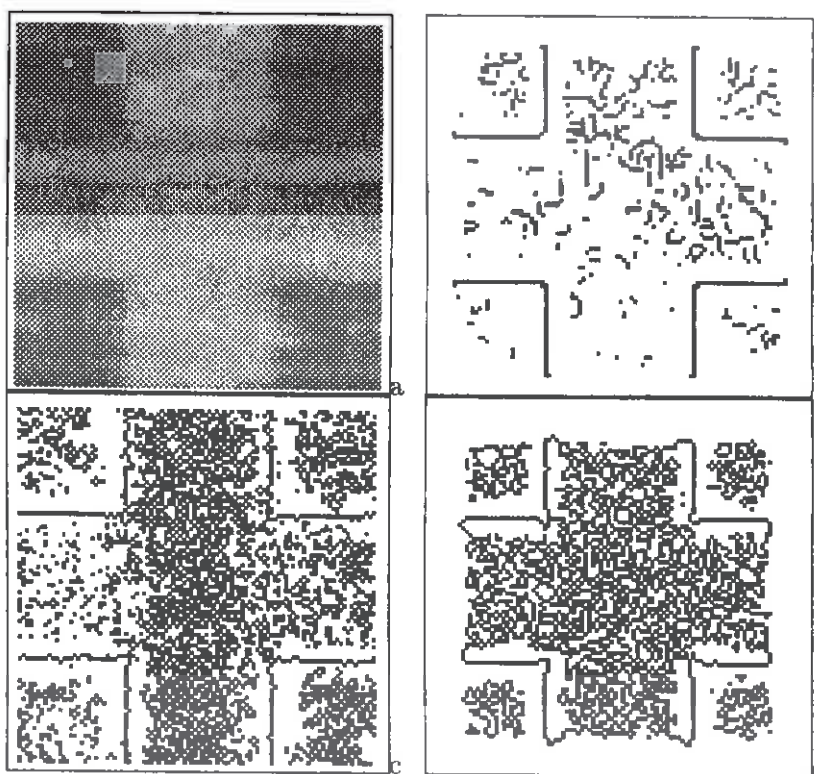


Figure 2.7. a) Image de synthèse de netteté différente pour les contours se trouvant au-dessus et au-dessous de la diagonale. b) Résultat de DGD ($\alpha = 1$), c) Résultat de DRF ($\alpha = 0.3$). d) Résultat de DL ($\alpha = 1$)

des détecteurs DGD et DL vaut 1 et celui de DRF vaut 0.2. Notons que les performances sont estimées à partir des images de contours sans aucune élimination préalable des faux points de contours. Un examen de la table 2.4 dans le cas de contour peu net et bruité montre que :

- le détecteur DGD possède des faibles erreurs de localisation, de sensibilité et d'ambiguïté. Ces dernières sont peu sensibles à la variation de la netteté,
- le détecteur DL possède les erreurs les plus élevées. Contrairement à la sensibilité du détecteur, l'erreur d'ambiguïté est fortement liée à la variation de la netteté.
- le détecteur DRF possède la plus faible erreur d'omission et d'importantes erreurs d'ambiguïté, de localisation et de sensibilité. L'erreur d'omission est peu affectée par le changement de la netteté.

On constate à partir de la table 2.5 que :

- le détecteur DGD possède une faible erreur de sensibilité et de localisation. Ce détecteur est peu sensible au changement du rapport signal/bruit,

- le détecteur DL possède les plus mauvaises performances. La localisation et l'ambiguïté sont sensibles à la variation du rapport signal/bruit,
- le détecteur DRF possède les plus faibles erreurs d'omission et d'ambiguïté. Son erreur de sensibilité est élevée. La localisation et l'ambiguïté sont sensibles à la variation du rapport signal/bruit.
- d'une manière générale, le rapport signal/bruit d'un contour a une influence sur l'ambiguïté, la sensibilité et la localisation d'un détecteur. Cependant, l'omission est peu affectée,

$\mu = 10$	Image (sb = 18.4dB)			Image (sb = 22dB)		
erreur	DGD	DRF	DL	DGD	DRF	DL
omission	0.04	0.00	0.11	0.04	0.00	0.11
ambiguïté	0.05	0.05	0.11	0.04	0.00	0.04
sensibilité	0.89	0.95	0.94	0.86	0.95	0.94
localisation	0.04	0.06	0.16	0.04	0.04	0.09

Tableau 2.5.

Cas d'une marche double :

Considérons le modèle $F(x)$ d'une double marche comme étant défini par :

$$F(x) = \begin{cases} 0 & \text{si } x \leq -l \\ A & \text{si } -l \leq x \leq l \\ 0 & \text{si } x \geq l \end{cases}$$

où A et l sont respectivement l'amplitude et la largeur du contour. La réponse dans l'intervalle $[-l, l]$ du détecteur DGD à ce signal est donnée par :

$$r(x) = (\alpha \sin(w(l-x)) + w \cos(w(l-x)))e^{-\alpha(l-x)} - (\alpha \sin(w(l+x)) + w \cos(w(l+x)))e^{-\alpha(l+x)}$$

L'objectif est d'examiner le comportement de la position des maxima de $r(x)$ en fonction de la largeur l . Soit la dérivée première de $r(x)$ donnée par :

$$r'(x) = (\alpha^2 + w^2)((\sin(w(l-x))e^{-\alpha(l-x)} + \sin(w(l+x))e^{-\alpha(l+x)})$$

Dans le cas où $\alpha \gg w$ et w est proche de zéro, les valeurs de $r'(x)$ aux points l et $-l$ sont :

$$r'(x) \simeq \begin{cases} 2wle^{-2\alpha l} & \text{si } x = l \\ -2wle^{-2\alpha l} & \text{si } x = -l \end{cases}$$

Selon les valeurs de α et de l on distingue deux cas :

- si $\alpha \ll l$ alors la fonction $r'(x)$ est donnée par :

$$r'(x) \simeq \begin{cases} 2wl & \text{si } x = l \\ -2wl & \text{si } x = -l \end{cases}$$

Ce résultat indique que la position du passage par zéro ne coïncide pas avec les points l et $-l$. L'erreur de localisation est due à l'influence mutuelle exercée par chacune des deux marches.

- si $\alpha \gg l$ alors $r'(x) \simeq 0$ aux points $x = l$ et $x = -l$. Ce résultat indique que dans le cas où l'échelle du détecteur est négligeable devant la largeur de la double marche, aucune influence n'est exercée par une marche sur l'autre. En conséquence, l'erreur de localisation tend vers zéro.

La table 2.6 montre les performances des trois détecteurs exécutés sur l'image de synthèse 2.8, en fonction de la largeur l . Pour les détecteurs DGD et DL le paramètre α est égale à 1, pour DRF le α vaut 0.3. Les résultats obtenus montrent que :

- le détecteur DGD possède la plus petite erreur de sensibilité et les erreurs de localisation et d'ambiguïté les plus élevées. Son erreur de localisation est peu affectée par le changement de la largeur de la marche,
- le détecteur DL possède l'erreur d'omission la plus élevée. Il est très sensible au changement de la largeur,
- le détecteur DRF possède la plus petite erreur d'omission et d'ambiguïté. Ce détecteur est sensible aussi à la variation de la largeur.

sb = 20 dB	Image ($l = 3$)			Image ($l = 5$)		
erreur	DGD	DRF	DL	DGD	DRF	DL
<i>omission</i>	0.07	0.00	0.13	0.02	0.00	0.06
<i>ambiguïté</i>	0.27	0.07	0.14	0.14	0.02	0.02
<i>sensibilité</i>	0.53	0.80	0.76	0.50	0.78	0.76
<i>localisation</i>	1.00	1.00	1.00	0.99	0.81	0.61

Tableau 2.6.

2.4.5 Influence des propriétés d'un détecteur sur ses performances

Les résultats donnés dans les deux paragraphes précédents permettent de constater que les performances d'un détecteur sont affectées par son échelle aussi bien dans le cas d'une marche simple que dans le cas d'une marche double. En effet, l'augmentation de l'échelle du détecteur implique une augmentation dans le rapport signal/bruit et une erreur de localisation plus importante.

L'échelle d'un détecteur impose aussi certaines contraintes lors de l'implantation et l'utilisation d'un détecteur (cf. 5.3.2 de la partie 1). La table 2.7 montre les performances des trois détecteurs exécutés sur l'image de synthèse 2.7 en fonction de l'échelle du détecteur. Ces résultats montrent que :

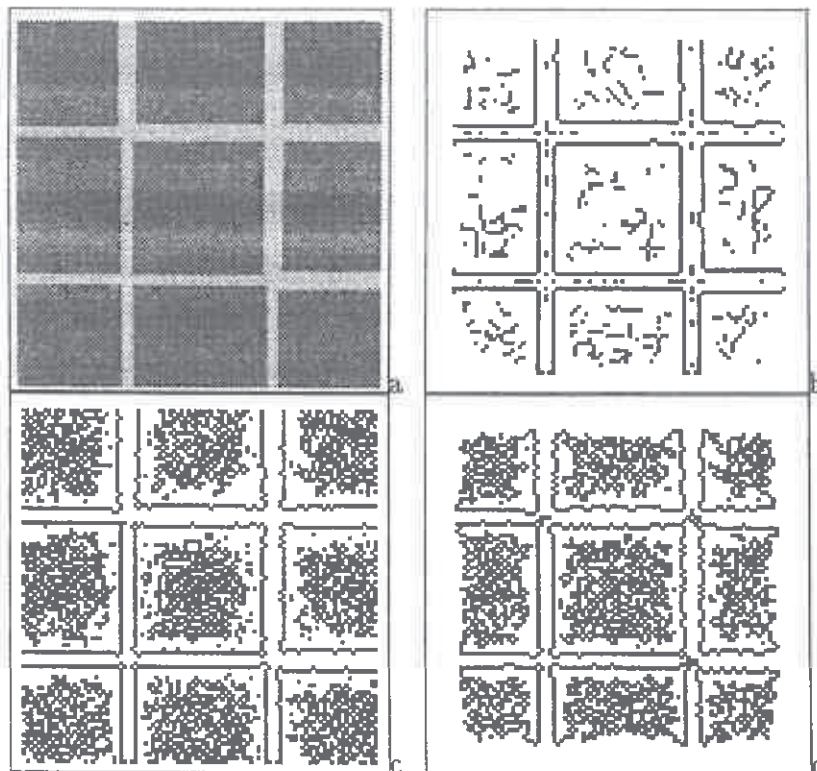


Figure 2.8. a) Image de synthèse. b) Résultat de DGD ($\alpha = 1$), c) Résultat de DRF ($a = 0.3$). d) Résultat de DL ($\alpha = 1$)

- contrairement à l'erreur d'ambiguïté, les erreurs d'omission et de localisation augmentent lorsque l'échelle augmente (l'échelle est inversement proportionnelle au paramètre a du détecteur DRF et au paramètre α des détecteurs DGD et DL).
- le détecteur DGD possède les plus faibles erreurs de sensibilité, de localisation et d'ambiguïté. Cette dernière est peu affectée par la variation de l'échelle. Les autres performances se dégradent lorsque l'échelle augmente.
- le détecteur DL possède les plus mauvaises performances. Son erreur de sensibilité est peu affectée par la variation de l'échelle. Les autres performances sont sensibles.
- le détecteur DRF possède la plus faible erreur d'omission. Les autres erreurs sont élevées. Ses erreurs de sensibilité et d'omission sont peu affectées par le changement d'échelle.

Nous nous intéressons maintenant à l'influence des propriétés mathématiques d'un détecteur sur ses performances. Un examen des résultats décrits dans les tables 2.4, 2.5, 2.6 et 2.7 montre que :

- l'erreur d'omission est affectée par l'opérateur de différentiation et par le filtre de lissage. Le filtre de lissage de DRF engendre la plus faible erreur d'omission. Cependant, l'opérateur laplacien fournit une erreur d'omission plus élevée qu'un opérateur

sb= 18.4 dB, $\mu = 0.5$	DGD		DRF		DL	
	$\alpha = 1$	$\alpha = 0.8$	$a = 0.3$	$a = 0.20$	$\alpha = 1.$	$\alpha = 0.8$
erreur						
<i>omission</i>	0.07	0.11	0.00	0.00	0.18	0.48
<i>ambiguïté</i>	0.1	0.09	1.01	0.64	0.99	0.42
<i>sensibilité</i>	0.87	0.91	0.91	0.91	0.90	0.91
<i>localisation</i>	0.1	0.14	0.31	0.33	0.38	0.55

Tableau 2.7.

gradient. Le détecteur DRF possède la plus faible erreur qui est peu affectée par la variation des caractéristiques du contour et de l'échelle,

- l'erreur d'ambiguïté est plus affectée par l'opérateur de différentiation que par le filtre de lissage. Dans le cas d'un contour de type marche simple, un opérateur gradient fournit peu de réponses multiples. En revanche, un laplacien semble être mieux adapté du point de vue ambiguïté à la détection des contours de type double marche,
- l'erreur de sensibilité n'est affectée que par l'opérateur de différentiation. Un opérateur laplacien fournit plus de faux points de contours,
- l'erreur de localisation est affectée par l'opérateur de différentiation et par le filtre de lissage. Plus précisément,
 - dans le cas d'un contour de type marche simple, un opérateur gradient fournit des contours de meilleure localisation. Le filtre du détecteur DRF possède une meilleure précision,
 - en revanche, un laplacien semble être mieux adapté du point de vue localisation à la détection des contours de type double marche. Le filtre des détecteurs DGD et DL possède une meilleure précision.

2.4.6 Une synthèse

En se fondant sur les études effectuées dans ce paragraphe, dans la première partie et dans les travaux de [Zio 91b, Tab 91a], nous déduisons les résultats suivants :

- un détecteur laplacien semble être mieux adapté à la mise en évidence des jonctions, des contours lisses, des doubles marches et à la détection multi-échelle des contours (car ses performances sont plus sensibles à la variation de l'échelle),
- un détecteur utilisant un opérateur de différentiation du premier ordre semble être mieux adapté à la mise en évidence de contours peu nets et de contours bruités,
- un détecteur invariant ou non par rotation et utilisant un opérateur de différentiation du premier ordre possède des performances dépendant de l'orientation du contour. Ces performances sont symétriques par rapport aux orientations $-\pi/4$, 0 et $\pi/4$ qui constituent leurs points critiques,

- le filtre utilisé par le détecteur DRF fournit des contours mieux localisés dans le cas des contours de type marche simple. Il engendre aussi une très faible erreur d'omission,
- le filtre utilisé par les détecteurs DGD et DL fournit des contours mieux localisés dans le cas des contours de type double marche. Le détecteur DL semble être le mieux adapté pour la détection multi-échelle de contours.
- un détecteur directionnel semble être mieux adapté à la détection de contours rectilignes,

La définition des heuristiques relatives au choix de l'échelle la plus adaptée pour un contour semble être plus difficile. Nous déduisons les deux règles suivantes :

- un contour de type marche simple qui est bruité ou possédant une faible netteté nécessite l'utilisation d'une échelle élevée. Les marches brouillées font partie de cette classe de contours. Cependant, un contour net et non bruité nécessite l'utilisation d'une échelle basse,
- dans le cas d'une double marche, l'échelle du détecteur doit être très petite par rapport à la largeur de cette double marche pour diminuer l'influence de la détection d'une marche sur l'autre.

En conclusion, les règles de sélection utilisées dans notre système sont dérivées à partir de ces heuristiques et sont validées en général par l'expérimentation. Il est difficile de distinguer avec certitude l'influence des propriétés du filtre de lissage de l'influence de l'opérateur de différentiation. Nous avons obtenu parfois des résultats contradictoires. Une étude plus complète nécessite un temps considérable pour réaliser des développements théoriques parfois complexes et des expérimentations nombreuses sur des images à caractéristiques variées.

2.5 Etude d'un exemple

L'objectif de ce paragraphe est de valider l'approche de sélection ainsi que les différents algorithmes proposés sur un exemple réel. Pour ce faire, considérons l'image originale décrite par la figure 2.9 et la fenêtre sur cette image qui contient le contour exemple donné par la figure 2.10. Cette fenêtre constitue l'image en donnée du système, c'est-à-dire que tous les traitements sous-jacents à la sélection de détecteurs sont faites en ne considérant que cette fenêtre. Le contour exemple est formé d'un ensemble de segments de droites. Certaines parties de ce contour, en particulier les segments numérotés 1 et 2, sont peu nets. Ce qui rend leur détection difficile, comme le confirme le résultat du détecteur de GEF de Shen et Castan avec $a = 0.5$ donné par la figure 2.11.

2.5.1 Résultats de l'étape d'analyse du contour

Calcul du type du contour : rappelons que le type du contour est estimé par une analyse de son profil. Initialement la classification est effectuée au niveau du point de contour. En utilisant les règles d'étiquetage données par les équations 2.4 et 2.5, on

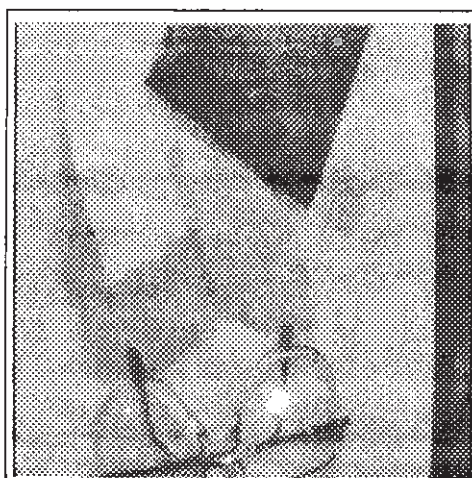


Figure 2.9. *Image originale*

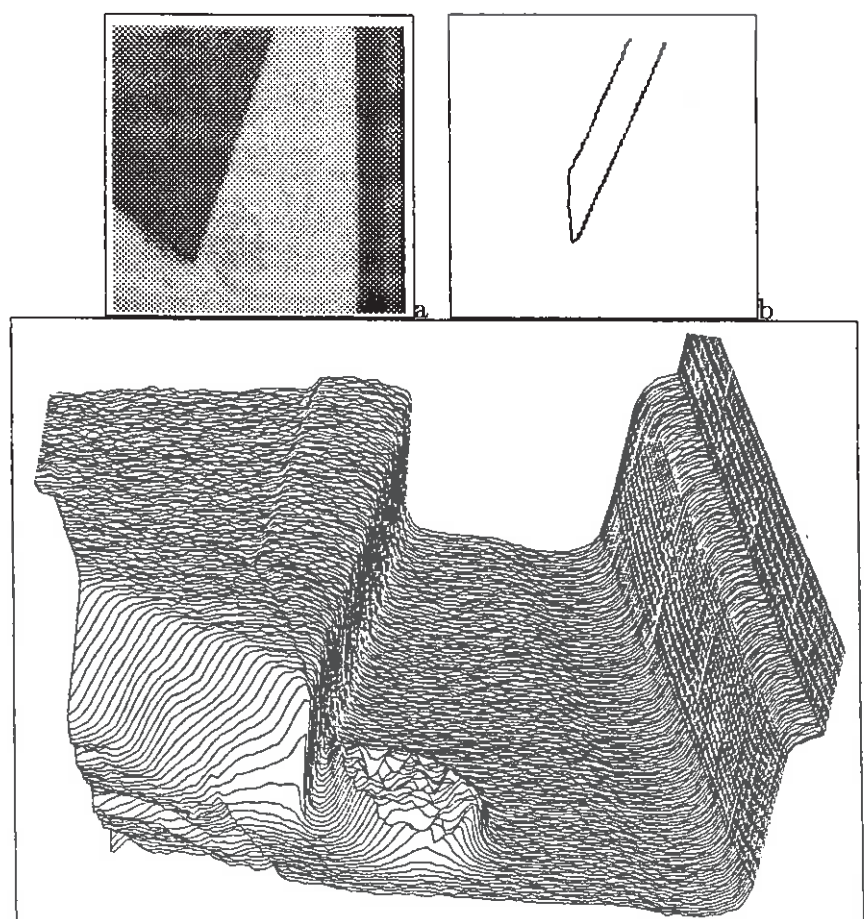


Figure 2.10. *a) Contour exemple, b) et c) Fenêtre contenant le contour exemple et les surfaces correspondantes.*

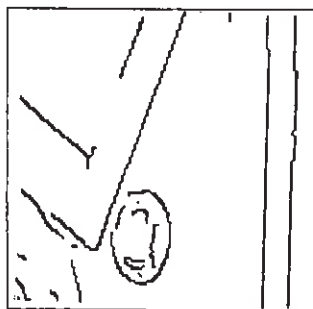


Figure 2.11. Contour résultat du détecteur GEF avec $a = 0.5$.

obtient 59% des points classés "marche simple", 14% de points classés "marche double" et 27% n'appartiennent à aucune des deux classes précédentes.

Rappelons que le seuil de la variance et celui de la hauteur sont calculés directement à partir de l'énergie du bruit. Le seuil relatif au niveau de gris moyen s_m vaut 5. La taille de la fenêtre 1D définissant le voisinage vaut 13. Notons que la taille du voisinage et s_m varient respectivement par pas de 2 et 1 à chaque itération.

En utilisant la règle d'étiquetage complémentaire définie dans le paragraphe 2.2.2, les 27% des points ambigus sont classés "marche simple".

Enfin, le résultat précédent est raffiné en considérant les chaînes de contours, c'est-à-dire que la chaîne prend l'étiquette de la majorité de ses points. Le résultat final est alors 100% des chaînes sont étiquetées "marche simple".

Estimation du bruit, netteté et contraste : elle est effectuée en utilisant les algorithmes décrits dans les paragraphes 2.2.1, 2.2.3 et 2.2.4. La taille des masques de Tchebycheff est variable. Une première estimation est effectuée en utilisant un masque de taille 5×5 . Elle est augmentée ensuite par pas de 2. Les différents paramètres estimés sont lissés par une gaussienne d'écart-type égale à un.

Les distributions fréquentielles des ces différents paramètres sont représentées par les histogrammes donnés par la figure 2.12. Les figures 2.12.a, 2.12.b, 2.12.c, 2.12.d sont respectivement les histogrammes du contraste, de la netteté, du bruit et du rapport signal/bruit. Deux combinaisons différentes de ces caractéristiques sont illustrées par les figures 2.12.e et 2.12.f, respectivement le module des deux vecteurs (contraste, netteté, bruit) et (netteté, rapport signal/bruit). Un examen visuel des différents histogrammes et en particulier ceux relatifs au rapport signal/bruit, à la netteté et au module du vecteur formé à partir de ces deux attributs montre que le contour est non homogène.

Résultat de la segmentation du contour : la segmentation du contour exemple est effectuée en associant à chaque point du contour le vecteur de composantes (contraste, netteté, bruit). Le critère de la partition initiale est donné par :

$$(lg \geq 5) \text{ et } ((c \geq 5) \text{ ou } (p \geq 10) \text{ ou } (b \geq 2)),$$

où lg est la longueur de la courbe support du contour, c le contraste, p la netteté et b le bruit. La division d'une courbe est effectuée au point milieu. Le nombre de courbes résultats de cette étape est de 50. La croissance des courbes utilise séquentiellement trois critères formés à partir de trois mesures définies pour chaque deux courbes adjacentes i et j de la manière suivante :

- la dynamique : $v_{ij}^{(1)} = |v_{max_{ij}} - v_{min_{ij}}|;$

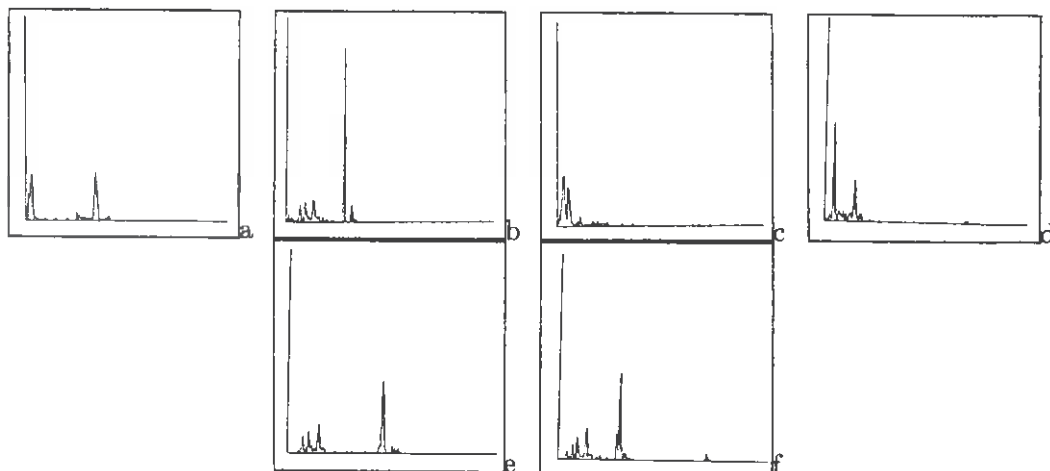


Figure 2.12. Les histogrammes de a) contraste, b) nettené, c) bruit, d) rapport signal/bruit, e) module du vecteur dont les composantes sont : contraste, nettené et bruit, f) module du vecteur dont les composantes sont : nettené et rapport signal/bruit.

- la moyenne : $v_{ij}^{(2)} = |v_{moy_i} - v_{moy_j}|$;
- la disparité : $v_{ij}^{(3)} = \frac{\sum_{p_i} \sum_{p_j} |v_{p_i} - v_{p_j}|}{\sum_{p_i} \sum_{p_j} 1}$;

Soit $cr^{(n)} = \alpha c^{(n)} + \beta p^{(n)} + (1 - \alpha - \beta)b^{(n)}$, où $c^{(n)}$, $p^{(n)}$ et $b^{(n)}$ sont respectivement le contraste, la nettené et le bruit du vecteur $v_{ij}^{(n)}$ et α , β sont des scalaires.

Les deux courbes i et j sont fusionnées si et seulement si elles sont adjacentes et la condition suivante est vérifiée pour tout couple de courbes adjacentes (l, k) :

$$cr_{ij}^{(n)} = \min_{(k,l)} cr_{l,k}^{(n)} \text{ et } cr_{ij}^{(n)} < \text{seuil}^{(n)} \text{ ou } \min(\lg(i), \lg(j)) < \lg_{min}.$$

Le résultat final de l'étape de segmentation est composé de deux courbes (voir fig. 2.13) dont les caractéristiques sont résumées dans la table 2.8. Les valeurs des différents seuils utilisés sont : $\alpha = \beta = 0.333$, $\text{seuil}^{(1)} = \text{seuil}^{(2)} = \text{seuil}^{(3)} = 5$ et $\lg_{min} = 5$.

Considérons comme mesure de qualité de la segmentation la combinaison linéaire des composantes du vecteur formé à partir de la différence absolue des caractéristiques de chaque couple de courbes adjacentes. La colonne nommée *qualité* dans la table 2.8 décrit les valeurs de cette mesure.

2.5.2 Résultats de la sélection

Nous considérons dans ce paragraphe qu'un sous-ensemble de règles de sélection parmi celles décrites au chapitre 4. Ces règles sont données ici sans aucune justification. Pour plus de détails on pourra consulter le chapitre 4.

R1 : si le contour est de type marche simple et de rapport signal/bruit supérieur à 6 et

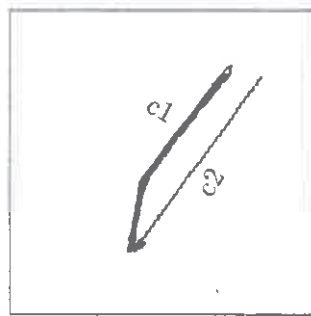


Figure 2.13. Les courbes résultats de la segmentation.

courbe	type	forme	longueur	bruit écart-type	contraste	netteté en degrés	rsb en dB	qualité
c1	marche simple	droites non fermées	101.43	15.05	10.72	41.48	- 2.94	
								42.41
c2	idem	idem	100.57	16.82	102.69	76.26	15.72	

Tableau 2.8. Résultat final de la segmentation.

de netteté supérieur à 67.5 alors

utiliser le détecteur DGD avec $\alpha = 1.5, 18$, ($loc=0.000 det=1.000 ang=0$), ($loc=0.016 det=1.000 ang=22.5$), ($loc=0.142 det=1.000 ang=45$).

La règle R1 signifie que si le contour est une marche simple, net et non bruité alors le détecteur de Deriche avec $\alpha = 1.5$ peut être utilisé.

R2 : si le contour est de type marche simple et de rapport signal/bruit dans l'intervalle [2,4] ou de netteté dans l'intervalle [22.5, 45] alors

utiliser le détecteur DGD avec $\alpha = 1., 18$, ($loc=0.106 det=0.994 ang=0$), ($loc=0.315 det=0.991 ang=22.5$), ($loc=0.404 det=0.988 ang=45$)

Les règles R2 peuvent être interprétées de la même façon que la règle R1. Dans ce cas, le contour est à la fois bruité et brouillé. Ce qui explique le choix des paramètres pour lisser fort.

R3 : si le contour est de type marche simple et de forme rectiligne et

de rapport signal/bruit supérieur à 6 et

de netteté supérieur à 67.5 alors

utiliser le détecteur DGD directionnel avec $\alpha = 1.5, 15$, ($loc=0.000 det=1.000$)

R4 : si le contour est de type marche simple et

de forme rectiligne et

de rapport signal/bruit dans l'intervalle [2,4] ou

de netteté dans l'intervalle [22.5, 45] alors
utiliser le détecteur DGD directionnel avec $\alpha = 1., 15$, ($loc=0.106$ $det=0.994$).

Les règles R3 et R4 sont similaires respectivement aux règles R1 et R2. La différence réside dans la forme rectiligne du contour. Ces deux derniers détecteurs correspondent au détecteur de Deriche directionnel (paramétré par l'orientation). Leurs performances sont indépendantes de l'orientation du contour.

Les détecteurs validés par les courbes résultats de la segmentation sont résumés dans la table 2.9. Supposons que le résultat attendu par l'utilisateur possède les caractéristiques suivantes :

- probabilité de succès : la contribution du rapport signal/bruit du détecteur α_1 est égale à 1 (l'expérience du détecteur n'intervient pas). La contribution de la détection et celle de la promesse faite par le détecteur est égale à 0.5 ($\alpha_2 = 0.5$),
- mesure de sélection : les contributions de la probabilité de succès, de la localisation ainsi que du temps de calcul sont égaux ($\alpha = \beta = 0.33$).

La table 2.10 présente les différents détecteurs validés par le contour exemple triés dans un ordre décroissant selon le critère de sélection. Le meilleur détecteur selon ce critère est celui décrit par la règle (R3, 18). Ce choix semble être adéquat du fait que les contours $c1$ et $c2$ possèdent des longueurs comparables. De plus, les détecteurs R1 et (R3, 18) possèdent les meilleures performances et celles de ce dernier sont indépendantes de l'orientation du contour.

Supposons maintenant que l'utilisateur ne souhaite pas la mise en œuvre des détecteurs directionnels. En utilisant les mêmes taux de contribution que dans le cas précédent nous avons : R1 validé par $c2$ et R2 par $c1$. La valeur de la mesure de sélection est (R1, $c2$, 0.322) et (R2, $c1$, 0.678). Le détecteur retenu est donc R1.

Courbe	$c1$	$c2$
Détecteurs	R2, (R4, 20), (R4, -8)	R1, (R3, 18)

Tableau 2.9. Les détecteurs validés par le contour exemple. (R_x, a) signifie le détecteur directionnel décrit par la règle R_x , a est l'orientation normale au contour exprimé en degrés dans lequel le détecteur doit être exécuté s'il est choisi.

Détecteurs	(R3, 18)	R1	(R4, 20)	R2	(R4, -8)
Performances	0.116	0.121	0.246	0.255	0.262

Tableau 2.10. Résultat de l'évaluation de la mesure de sélection d'un détecteur.

2.6 Conclusion

Dans ce chapitre, nous avons décrit un algorithme de sélection d'un détecteur pour mettre en évidence un contour donné. Cet algorithme se compose d'une étape d'analyse du

contour et d'une étape de choix du détecteur. La première étape a pour objectif la définition des attributs pertinents du contour et la segmentation de celui-ci en h-contours. Nous avons présenté des algorithmes originaux pour l'estimation du bruit et l'estimation des attributs qui définissent le modèle de contour. L'algorithme de segmentation du contour est une adaptation de celui proposé par B. Wrobel-Dautcourt [WD 88].

La seconde étape consiste à déterminer les détecteurs validés par les h-contours résultats de la segmentation et le choix parmi ceux-ci de celui qui possède les meilleures performances et qui fournit un maximum de points de contour.

Notre approche de sélection nécessite l'extension de la notion de détecteur pour inclure son contexte d'utilisation. Après avoir défini les attributs et un ensemble de mesures de performance d'un détecteur, nous avons présenté une étude à la fois théorique et expérimentale de l'influence de ces attributs et ceux du contour sur ses performances. Nous avons montré le résultat suivant : les performances d'un détecteur de contours utilisant un opérateur de différentiation du premier ordre, qu'il soit invariant ou non par rotation, sont affectées par l'orientation du contour. Cette étude a permis de déduire des heuristiques qui nous ont guidé pour la définition du contexte d'utilisation d'un détecteur.

Le point fort de notre approche réside dans l'utilisation de l'objectif décrit par l'utilisateur, des caractéristiques du contour et les propriétés des détecteurs. Ceci a permis d'éviter une approche coûteuse comme celle proposée dans le système IMPRESS. Un autre avantage est que les algorithmes d'estimation du bruit et des attributs relatifs au profil sont peu sensibles à la position du contour. Il en résulte qu'une erreur de quelques pixels dans la position du contour exemple introduite lors de son acquisition n'induit pas une erreur d'estimation de ces attributs.

Le point faible de cette approche provient de l'identification du modèle de contours. L'algorithme proposé est sensible à la position du contour. De plus, l'étiquetage des points de contour ambigus est simpliste.

Les améliorations à apporter à cette approche concernent essentiellement l'étape d'analyse du contour. En effet, la description présentée des contours peut être considérée comme étant une première esquisse d'une analyse complexe des structures sous-jacentes aux surfaces d'une image. La liste des attributs qui vient d'être décrite est loin d'être exhaustive. Une description plus complète du contour nécessite certainement :

- la considération de plusieurs modèles de contours de type marche et l'introduction d'autres attributs relatifs aux processus de fabrication des images [Lun 86a, Lun 86b],
- l'utilisation de bases de polynômes différentes pour la description d'un seul modèle de contour et la sélection de la meilleure en se fondant par exemple sur l'erreur d'approximation,
- l'utilisation d'une analyse plus poussée pour l'identification du contour qui doit être indépendant de sa position,
- enfin, notons que la définition du contexte d'utilisation d'un détecteur nécessite un travail fastidieux à la fois théorique et expérimental. Les résultats sont souvent peu précis et contradictoires. Pour combler ce vide, une étape d'acquisition automatique de règles de sélection semble être nécessaire.

Ces différentes améliorations sont détaillées dans le chapitre 4.

2.7 Annexe A

Cette annexe présente un algorithme pour l'estimation des attributs d'un contour de type marche. Cet algorithme est fondé sur les techniques d'ajustement. Il s'agit d'approcher chaque fenêtre de taille $N \times N$ de l'image par le polynôme discret de Tchebycheff et d'en déduire les attributs du contour. Le degré du polynôme dépend du modèle de contour. Rappelons que nous considérons deux modèles de contour : marche simple et marche double de type crête. Pour une marche simple un polynôme de degré 3 semble être satisfaisant. Cependant, pour une double marche, il faut, au moins un polynôme de degré 4. Nous présentons dans ce qui suit l'algorithme d'ajustement de la fenêtre $N \times N$ par un polynôme de Tchebycheff de degré 4. Nous montrons ensuite comment utiliser le résultat de cet ajustement pour déduire les attributs d'une marche simple et les attributs d'une double marche de type crête.

Considérons la base des polynômes de Tchebycheff suivante :

$$\{1, T_1(x), T_1(y), T_2(x), T_1(x)T_1(y), T_2(y), T_3(x), T_2(x)T_1(y), T_1(x)T_2(y), T_3(y), T_4(x), T_3(x)T_1(y), T_2(x)T_2(y), T_1(x)T_3(y), T_4(y)\} \quad (2.11)$$

où $T_n : [-1, 1] \rightarrow [-1, 1]$ est le polynôme de Tchebycheff de degrés n défini par la relation suivante :

$$T_n(z) = \cos(n \arccos(z))$$

Le polynôme T_n possède n zéros dans l'intervalle $[-1, 1]$ dont la position est donnée par :

$$z_k = \cos\left(\frac{\pi(2k-1)}{2n}\right), \quad k = 1, \dots, n$$

Les polynômes discrets $T_i(x)T_j(y)$ sont orthogonaux par rapport au produit scalaire dans l'intervalle $[-1, 1]$. Soient x_k ($k = 1, \dots, m$) et y_l ($l = 1, \dots, n$) les m et n zéros de $T_m(x)T_n(y)$, $f(r, c)$ une fonction arbitraire définie par : $[-1, 1] \rightarrow \mathfrak{R}$ et α_i , ($i = 0, \dots, 14$) des coefficients définis par :

$$\begin{aligned} \alpha_i &= \frac{2}{N \times N} \sum_{k=1}^N \sum_{l=1}^N f(x_k, y_l) B_i(x_k, y_l) \\ &= \frac{2}{N \times N} \sum_{k=1}^N \sum_{l=1}^N f\left(\cos\left(\frac{\pi(2k-1)}{2N}\right), \cos\left(\frac{\pi(2l-1)}{2N}\right)\right) \\ &\quad \cos\left(\frac{\pi i_1(2k-1)}{2N}\right) \cos\left(\frac{\pi j_1(2l-1)}{2N}\right) \end{aligned} \quad (2.12)$$

où $B_i(x_k, y_l) = T_{i_1}(x_k)T_{j_1}(y_l)$ est le $i^{\text{ème}}$ élément de la base 2.11, alors la fonction f peut s'écrire sous la forme suivante [Pre 88] :

$$f(x, y) \simeq -0.5\alpha_0 + \sum_{i=0}^{14} \alpha_i B_i(x, y) \quad (2.13)$$

Dans le cas où les variables x et y de la fonction f prennent leurs valeurs dans $[a, b] \times [c, d]$ au lieu de $[-1, 1] \times [-1, 1]$, il est nécessaire d'effectuer le changement de variable suivant :

$$X = \frac{2x - (b + a)}{b - a} \quad \text{et} \quad Y = \frac{2y - (d + c)}{d - c}$$

Toute évaluation de la fonction f doit être faite au point (X, Y) . Posons $r = \cos(\arccos(x))$ et $c = \cos(\arccos(y))$, la fonction 2.13 formée à partir de la base 2.11 peut s'écrire sous la forme suivante :

$$f(r, c) = 0.5a_0 + a_1r + a_2c + a_3r^2 + a_4rc + a_5c^2 + a_6r^3 + a_7cr^2 + a_8rc^2 + a_9c^3 + a_{10}r^4 + a_{11}r^3c + a_{12}r^2c^2 + a_{13}c^3r + a_{14}c^4 \quad (2.14)$$

L'ajustement de l'image $N \times N$ par le polynôme 2.14 revient à calculer la suite des $\alpha_i, (i = 0, 1, \dots, 14)$ en utilisant 2.12. Les coefficients a_i sont obtenus à partir des α_i par un simple changement de base. Considérons le polynôme $f_\theta(\rho)$ ajustant l'image dans la direction θ . En posant $r = \rho \sin(\theta)$ et $c = \rho \cos(\theta)$, le polynôme 2.14 devient :

$$f_\theta(\rho) = b_0 + b_1\rho + b_2\rho^2 + b_3\rho^3 + b_4\rho^4 \quad (2.15)$$

avec

$$\begin{aligned} b_0 &= 0.5a_0, & b_1 &= a_1 \sin(\theta) + a_2 \cos(\theta), & b_2 &= a_3 \sin^2(\theta) + a_4 \sin(\theta) \cos(\theta) + a_5 \cos^2(\theta) \\ b_3 &= a_6 \sin^3(\theta) + a_7 \sin^2(\theta) \cos(\theta) + a_8 \sin(\theta) \cos^2(\theta) + a_9 \cos^3(\theta) \\ b_4 &= a_{10} \sin^4(\theta) + a_{11} \sin^3(\theta) \cos(\theta) + a_{12} \sin^2(\theta) \cos^2(\theta) + a_{13} \sin(\theta) \cos^3(\theta) + a_{14} \cos^4(\theta) \end{aligned}$$

θ étant l'orientation du contour, sa définition dépend du type de contour comme nous le montrons plus loin.

Nous présentons dans ce qui suit les algorithmes pour l'estimation des attributs d'une marche simple et d'une marche double de type crête.

2.7.1 Cas d'une marche simple

Rappelons qu'un polynôme de Tchebycheff de degré 3 semble être satisfaisant pour l'ajustement d'une marche simple. Ainsi, les coefficients a_i tels que $i > 9$ de la fonction donnée par l'équation 2.14 sont nuls. Il en résulte que le coefficient b_4 de la fonction (eq. 2.15) est égal aussi à zéro.

Rappelons que les attributs d'une marche simple sont le contraste et la netteté. Considérons le polynôme $g_\theta(\rho)$ ajustant l'image dans la direction du contour θ . g_θ est obtenu à partir du polynôme 2.14 ($b_4 = 0$) par translation telle que le point d'inflexion soit à l'origine. La fonction g_θ est donnée par :

$$g_\theta(\rho) = c_0\rho - c_1\rho^3 \quad (2.16)$$

avec

$$c_0 = \frac{(b_2^2 - 3b_1b_3)}{3b_3}, \quad c_1 = b_3, \quad \theta = \arctg\left(\frac{a_1}{a_2}\right) \quad (2.17)$$

Etant donné le polynôme 2.16, il s'agit d'estimer le contraste et la netteté du contour. Dans le cas d'un contour de type marche simple, g_θ possède deux extrema tels que :

- leurs amplitudes ne dépendent que du contraste. Elles sont proportionnelles au contraste,
- leurs positions ne dépendent que de la façon dont le niveau de gris varie dans la direction normale au contour. Nous appelons ce dernier paramètre "mesure de validité de l'ajustement". Les positions des extrema sont inversement proportionnelles à cette mesure,
- la pente du point d'inflexion (comme d'autres paramètres de la marche) est proportionnelle au contraste et à la mesure de validité de l'ajustement.

La pente p du point d'inflexion et la position r_0 des extrema ainsi que leur amplitude h sont données respectivement par :

$$p = g'_\theta(0) = c_0 = k_1 h s^n \quad (2.18)$$

$$r_0 = \mp \sqrt{\frac{c_0}{3c_1}} = \mp k_2 / s \quad (2.19)$$

$$h = \mp k_3 \frac{2}{3} c_0 \sqrt{\frac{c_0}{3c_1}} \quad (2.20)$$

où s est la mesure de validité de l'ajustement de la marche et les k_i sont des constantes positives. La résolution du système d'équations 2.18, 2.19 et 2.20 donne :

$$c_0 = k_1 h s \quad (2.21)$$

$$c_1 = \frac{k_1}{3k_2^2} h s^3 \quad (2.22)$$

En remplaçant c_0 et c_1 par leurs valeurs données par 2.17 et 2.18 dans 2.21, 2.22, on en déduit que :

$$h = k_h \frac{(b_2^2 - 3b_1 b_3)^{3/2}}{b_3^2} \quad (2.23)$$

$$s = k_s \frac{b_3^{1/3}}{h^{1/3}} \quad (2.24)$$

$$p = k_p h^{2/3} b_3^{1/3} \quad (2.25)$$

Contrairement au contraste, les estimations de la netteté et de la mesure de validité de l'ajustement sont fortement liées à la dimension N de la fenêtre. Elles sont inversement proportionnelles à N . Nous proposons de déterminer les paramètres k_h , k_p et k_s tels que l'estimation soit moins sensible à la variation de N . Pour ce faire, considérons une marche 1D discrète définie par :

$$u(r) = \begin{cases} -1 & \text{si } r < 0 \\ 1 & \text{sinon} \end{cases}$$

Les résultats de l'estimation des paramètres de la marche en utilisant 2.23, 2.24, 2.25 sont donnés dans le tableau 2.11. Etant donné que les paramètres de cette marche sont : le contraste égal à 2, la pente égale à 2 et l'échelle égale à 1, on peut en déduire que :

$$k_h = 1/\text{moy}(h) = 0.1202, \quad k_s = 1/\text{moy}(s) = 0.9682, \quad k_p = 1/\text{moy}(p) = 0.1163,$$

N	contraste $h = h/k_h$	validité $s = s/k_s$	pente $p = p/k_p$	erreur quadratique d'ajustement
5	16.8880	1.0725	18.1122	0.5143
7	16.7021	1.0449	17.4524	0.8571
9	16.6436	1.0342	17.2124	1.1688
11	16.6174	1.0289	17.0971	1.4685
13	16.6034	1.0258	17.0325	1.7622
17	16.5894	1.0227	16.9662	2.3406
21	16.5829	1.0212	16.9345	2.9127

Tableau 2.11. Les paramètres d'une marche 1D idéale. Le contraste, la mesure de validité d'ajustement et la pente de cette marche sont respectivement 2, 1 et 1. L'erreur quadratique d'ajustement correspond à la différence au carré entre la fenêtre originale $N \times N$ et la fonction $f(r, c)$ donnée par l'équation 1.11.

La mesure s décrit la différence entre une marche idéale et le contour analysé. Si cette quantité est supérieur à 1 (resp. inférieur à 1) alors les deux extrema du polynôme sont plus loins (res. plus proches) l'un de l'autre que ceux d'une marche idéale. L'approximation de la fenêtre $N \times N$ est correcte (c'est-à-dire cette fenêtre contient un contour de type marche) si et seulement si cette quantité est proche de 1.

2.7.2 Cas d'une double marche

Rappelons qu'un polynôme de Tchebycheff de degré 4 est nécessaire pour l'ajustement d'une double marche de type crête. Les attributs d'une double marche de type crête sont le contraste, la netteté, la largeur et la symétrie.

L'angle θ correspond à l'orientation qui maximise la dérivée seconde de la fonction f_θ (c'est-à-dire, l'orientation normale à la ligne de crête). Il est donné par :

$$\theta = \mp \frac{1}{2} \text{arctg} \left(\frac{a_5}{a_6 - a_4} \right) \quad (2.26)$$

Le signe $-$ est choisi quand il s'agit d'une crête et le signe $+$ quand il s'agit d'une vallée.

Dans le cas de l'approximation d'une crête, la fonction f_θ possède trois extrema (deux minima et un maximum). On veut fabriquer une fonction g_θ à partir du polynôme 2.15 par translation, tel que le maximum soit à l'origine.

$$g_\theta(\rho) = f_\theta(\rho - a) = b_0 + b_1(\rho - a) + b_2(\rho - a)^2 + b_3(\rho - a)^3 + b_4(\rho - a)^4 \quad (2.27)$$

Pour que le maximum soit à l'origine, il faut que $g'_\theta(0) = 0$ et $g''_\theta(0) < 0$. Ainsi, il s'agit de trouver la translation a^* solution du système suivant :

$$b_1 - 2ab_2 + 3b_3a^2 - 4b_4a^3 = 0 \quad \text{et} \quad g''_\theta(0) < 0 \quad (2.28)$$

La fonction 2.27 s'écrit alors sous la forme :

$$g_\theta(\rho) = c_0 + c_1\rho^2 + c_2\rho^3 + c_3\rho^4 \quad (2.29)$$

où

$$c_0 = b_0 - b_1a + b_2a^2 - b_3a^3 + b_4a^4, \quad c_1 = b_2 - 3b_3a + 6b_4a^2, \quad c_2 = b_3 - 4b_4a, \quad c_3 = b_4$$

Etant donné le polynôme 2.29, il s'agit d'estimer le contraste, la largeur, la symétrie et la netteté de la crête. Dans le cas d'une double marche de type crête, g_θ possède deux minima et deux points d'inflexion. Soient les dérivées de la fonction 2.29 :

$$g'_\theta(\rho) = 2c_1\rho + 3c_2\rho^2 + 4c_3\rho^3 \quad \text{et} \quad g''_\theta(\rho) = 2c_1 + 6c_2\rho + 12c_3\rho^2$$

Les minima de g_θ sont donnés par :

$$\rho_{1,2} = \frac{-3c_2 \mp \sqrt{9c_2^2 - 32c_1c_3}}{8c_3} \quad (2.30)$$

Les points d'inflexion de g_θ sont donnés par :

$$\rho_{3,4} = \frac{-3c_2 \mp \sqrt{9c_2^2 - 24c_1c_3}}{12c_3} \quad (2.31)$$

Les différents paramètres de la crête sont donnés par :

- le contraste h : il est mesuré par la profondeur de la crête, c'est-à-dire la valeur absolue de la différence de la fonction g_θ à l'origine et au point ρ_1 ou ρ_2 :

$$h = g_\theta(0) - \max(g_\theta(\rho_1), g_\theta(\rho_2)) \quad (2.32)$$

- la largeur l : elle peut être définie par la distance entre les deux points d'inflexions :

$$l = R|\rho_3 - \rho_4| = \left| \frac{\sqrt{9c_2^2 - 24c_1c_3}}{6c_3} \right| \quad (2.33)$$

- la symétrie s : elle est définie par le rapport de ρ_1 et de ρ_2 :

$$s = \left| \frac{\rho_1}{\rho_2} \right| = \left| \frac{-3c_2 + \sqrt{9c_2^2 - 32c_1c_3}}{-3c_2 - \sqrt{9c_2^2 - 32c_1c_3}} \right| \quad (2.34)$$

- la netteté p : elle est définie par la netteté moyenne des deux marches composantes de la crête :

$$p = \frac{c_0 - \max(g_\theta(\rho_3), g_\theta(\rho_4))}{2} \frac{-12c_3c_2}{3c_2^2 - 4c_1c_3} \quad (2.35)$$

Examinons maintenant les résultats de cette estimation en considérant un toit 1D discret (une crête dont la largeur du plateau est égale à zéro) défini par :

$$u(r) = \begin{cases} 2 & \text{si } r = 0 \\ 1 & \text{sinon} \end{cases}$$

Les résultats de l'estimation des paramètres de la crête donnés par les équations 2.32, 2.33, 2.34 et 2.35 sont présentés dans le tableau 2.12. Les paramètres de cette crête sont le contraste égal à 1, la largeur égale à 2, la netteté est égale à 1.11 et la symétrie est égale à 1. On remarque que les valeurs calculées des paramètres de cette crête dépendent de la taille de la fenêtre d'approximation utilisée. L'estimation de ces différents attributs se détériore rapidement en fonction de la taille du voisinage considéré. Ceci s'explique par le fait que l'influence exercée par une marche sur l'autre est proportionnelle à la taille du voisinage et dépend aussi de la largeur de la crête. En général, il est préférable d'utiliser la plus petite fenêtre possible (c'est-à-dire, celle qui permet un ajustement et qui fournit la plus petite erreur d'approximation).

N	contraste	largeur	netteté	symétrie	erreur quadratique d'ajustement
5	1.0000	1.8257	0.9892	1.0000	0.1600
7	0.7143	2.6515	0.8268	1.0000	0.5054
9	0.5556	3.6515	0.7016	1.0000	0.6405
11	0.4545	4.5644	0.6049	1.0000	0.7158
13	0.3846	5.4772	0.5293	1.0000	0.7648
25	0.2000	10.9545	0.2952	1.0000	0.8848
35	0.1429	15.5188	0.2138	1.0000	0.9195

Tableau 2.12. Les paramètres d'une double marche 1D idéale de type crête. Le contraste, la largeur, la netteté et la symétrie de cette crête sont respectivement 1, 2, 1.11 et 1.

En résumé, l'estimation des paramètres de la crête se fait comme suit :

1. approximation de l'image $N \times N$ par la fonction 2.14. Les coefficients α_i ($i = 0, \dots, 14$) sont calculés à l'aide de 2.12. Les coefficients a_i sont obtenus par un simple changement de base,

2. calcul de la fonction 2.29, l'orientation θ est donnée par 2.26, la translation a est obtenue en résolvant le système 2.28,
3. estimation des paramètres de la crête à l'aide des relations 2.32, 2.33, 2.35 et 2.34.

2.8 Annexe B

L'objectif de cette annexe est de déterminer un filtre capable d'estimer la composante bruit d'un signal donné. Soit $i(x)$ un signal lisse formé d'un bruit additif $b(x)$ et un signal pur $s(x)$, $i(x)$ s'écrit sous la forme :

$$i(x) = (s_1(x) + b_1(x)) * f(x) = s(x) + b(x) \quad (2.36)$$

où $f(x)$ est la réponse impulsionnelle d'un filtre de lissage de type passe-bas et $*$ l'opération de convolution.

La restauration du signal $i(x)$ peut être effectuée par l'estimation de la composante $b(x)$ telle que la différence entre $s(x)$ et $i(x)$ soit minimale :

$$e^2 = \int (i(x) - s(x))^2 dx$$

Minimiser cette erreur quadratique revient à déterminer le filtre $w(x)$ défini par [Bij 84, Max 77] :

$$b(x) = w(x) * i(x) \quad (2.37)$$

La résolution de ce problème n'est possible que si les fonctions de corrélation de $i(x)$, de $s(x)$ et de $b(x)$ sont connues. Soient $c_{ii}(\sigma)$ la fonction d'autocorrélation du signal $i(x)$, c_{ss} celle du signal $s(x)$, $c_{bb}(\sigma)$ la fonction d'autocorrélation du bruit $b(x)$ et $c_{sb}(\sigma)$ la fonction de corrélation de $s(x)$ et $b(x)$. La corrélation de deux fonctions $s(x)$ et $b(x)$ peut être définie par :

$$c_{sb}(\sigma) = \int_{-\infty}^{+\infty} s(x)b(x - \sigma)dx$$

En utilisant l'équation 2.37 on peut écrire :

$$c_{bb}(\sigma) = c_{ww}(\sigma) * c_{ii}(\sigma) \quad (2.38)$$

De l'équation 2.36, on obtient :

$$c_{ii}(\sigma) = c_{ss}(\sigma) + c_{bb}(\sigma) + c_{bs}(\sigma) + c_{sb}(\sigma) \quad (2.39)$$

$$c_{bb}(\sigma) = c_{b1b1}(\sigma) * c_{ff}(\sigma) \quad (2.40)$$

$$c_{ss}(\sigma) = c_{s1s1}(\sigma) * c_{ff}(\sigma) \quad (2.41)$$

Supposons que les conditions suivantes soient vérifiées :

1. le signal $s(x)$ et le bruit $b(x)$ ne sont pas corrélés, c'est-à-dire que :

$$c_{sb} = c_{bs} = 0 \quad (2.42)$$

2. la fonction d'autocorrélation du bruit $c_{bb}(\sigma)$ tend vers zéro lorsque σ augmente. On peut déduire alors que :

$$c_{ss}(\sigma) \simeq c_{ss}(\sigma) + c_{bb}(\sigma) \quad (2.43)$$

3. le bruit $b_1(x)$ est blanc, c'est-à-dire que :

$$c_{b_1b_1}(\sigma) = \delta(\sigma) \quad (2.44)$$

Avec $\delta()$ la distribution de Dirac.

En remplaçant c_{sb} , c_{bs} et c_{ss} par leur valeur (eq. 2.42, 2.43), l'équation 2.39 devient :

$$c_{ii}(\sigma) \simeq c_{ss}(\sigma) \quad (2.45)$$

En remplaçant c_{ii} par sa valeur (eq. 2.45) dans l'équation 2.38, on obtient :

$$c_{bb}(\sigma) \simeq c_{ww}(\sigma) * c_{ss}(\sigma) \quad (2.46)$$

Par remplacement de c_{bb} par sa valeur (eq. 2.44) dans l'équation 2.40, on obtient :

$$c_{bb}(\sigma) = c_{ff}(\sigma) \quad (2.47)$$

A partir des équations 2.46 et 2.47, on peut déduire que :

$$c_{ff}(\sigma) \simeq c_{ww}(\sigma) * c_{ss}(\sigma) \quad (2.48)$$

En substituant c_{ss} par sa valeur dans l'équation 2.41, on peut écrire :

$$c_{ff}(\sigma) \simeq c_{ww}(\sigma) * c_{s_1s_1}(\sigma) * c_{ff}(\sigma) \quad (2.49)$$

La détermination de $w(x)$ défini dans l'équation 2.37 revient alors à résoudre cette équation 2.49. Une solution possible est :

$$c_{ww}(\sigma) * c_{s_1s_1}(\sigma) = \delta(\sigma) \quad (2.50)$$

Dans le cas où $s_1(x)$ est une marche idéale, la solution à l'équation 2.50 est :

$$c_{ww}(\sigma) = \frac{\delta''(\sigma)}{A^2} \quad (2.51)$$

où A est l'amplitude de la marche. En effet,

$$c_{ww}(\sigma) * c_{s_1s_1}(\sigma) = \frac{1}{A^2} \delta''(\sigma) * c_{s_1s_1}(\sigma) = \frac{1}{A^2} \delta''(\sigma) * s_1(\sigma) * s_1(\sigma) = \delta(\sigma) \quad (2.52)$$

car, la dérivée d'une marche idéale est égal à $A \delta(\sigma)$.

On peut déduire à partir de l'équation 2.52 que :

$$w(x) = \frac{\delta'(x)}{A} \quad (2.53)$$

Le filtre $w(x)$ est égale à la première dérivée de la distribution de Dirac. L'équation 2.37 montre qu'il suffit de convoluer le signal original avec le filtre $f(x)$, puis de calculer la dérivée du signal lissé. Ce qui est équivalent à un lissage de signal original par $f'(x)$.

2.9 Annexe C

L'objectif de cette annexe est de montrer l'influence de la propriété de non invariance par rotation d'un détecteur sur ses performances. Rappelons qu'une fonction $f(x, y)$ est dite non invariante par rotation si et seulement si $f(x, y) = f(X, Y)$, avec :

$$X = x \cos(\theta) + y \sin(\theta) \quad \text{et} \quad Y = -x \sin(\theta) + y \cos(\theta) \quad (2.54)$$

Un opérateur est dit non invariant par rotation si et seulement si $\mathcal{O}_{x,y} = R \mathcal{O}_{X,Y}$, R étant la matrice de rotation.

Pour ce faire, nous étudions trois détecteurs gradient : Deriche, Canny et le détecteur GEF de Shen et Castan. Supposons que ces trois détecteurs utilisent comme opérateur de différenciation celui défini par :

$$\left(\frac{\partial}{\partial x}\right)^2 + \left(\frac{\partial}{\partial y}\right)^2$$

Soit un contour de contraste A et d'orientation $\theta \in [0, \pi/2]$ défini par :

$$c(x, y, \theta) = \begin{cases} A & \text{si } x \cos(\theta) + y \sin(\theta) \geq 0 \\ 0 & \text{sinon} \end{cases}$$

Nous proposons d'étendre en 2D les critères de Canny donnés au paragraphe 1.1 de la partie 2. Le rapport signal/bruit d'un détecteur gradient est donné par :

$$\Sigma = \frac{(f_x(x, y) * c(x, y, \theta))|_{(x,y)=(0,0)}^2}{n_0^2 \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f_x^2(x, y) dy dx} + \frac{(f_y(x, y) * c(x, y, \theta))|_{(x,y)=(0,0)}^2}{n_0^2 \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f_y^2(x, y) dy dx} \quad (2.55)$$

Supposons que le contour soit détecté à la position (x_0, y_0) au lieu de la position $(0, 0)$ et que x_0 et y_0 soient des variables aléatoires indépendantes. L'erreur de localisation peut être définie par $E(x_0^2) + E(y_0^2)$, nous traduisons la localisation par $1/E(x_0^2) + 1/E(y_0^2)$. Il en résulte la formule suivante :

$$\Lambda = \frac{(f_{xx}(x, y) * c(x, y, \theta))|_{(x,y)=(0,0)}^2}{n_0^2 \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f_{xx}^2(x, y) dy dx} + \frac{(f_{yy}(x, y) * c(x, y, \theta))|_{(x,y)=(0,0)}^2}{n_0^2 \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f_{yy}^2(x, y) dy dx} \quad (2.56)$$

où f_{xx} et f_{xxx} (resp. f_{yy} et f_{yyy}) sont respectivement la première et la seconde dérivée de f_x (resp. f_y) par rapport à x (resp. y).

Détecteur de Deriche

Considérons le détecteur 2D de Deriche (cas où $\alpha \gg w$). Ce détecteur utilise comme fonction de lissage :

$$f(x, y) = k(\alpha|x| + 1)(\alpha|y| + 1)e^{-\alpha(|x|+|y|)}$$

Montrons que ce détecteur est non invariant par rotation. En remplaçant dans cette fonction toute occurrence de x par X et de y par Y , on obtient $f(x, y) \neq f(X, Y)$. La

fonction $f(x, y)$ n'est pas alors invariante par rotation. Bien que l'opérateur $(\frac{\partial}{\partial x})^2 + (\frac{\partial}{\partial y})^2$ soit invariant par rotation, nous avons $(\frac{\partial f(x,y)}{\partial x})^2 + (\frac{\partial f(x,y)}{\partial y})^2 \neq (\frac{\partial f(X,Y)}{\partial x})^2 + (\frac{\partial f(X,Y)}{\partial y})^2$. Ainsi, l'utilisation de l'opérateur gradient ci-dessus (qui est invariant) et de la fonction de lissage de Deriche $f(x, y)$ (qui ne l'est pas) est un détecteur non invariant par rotation.

Etudions maintenant, l'influence de l'orientation du contour sur les performances du détecteur de Deriche 2D.

Le détecteur 2D est donné par :

$$f_x(x, y) = \frac{-k}{\alpha^2} x e^{-\alpha|x|} (\alpha|y| + 1) e^{-\alpha|y|} = f(x) g(y) \quad \text{et} \quad f_y(x, y) = f(y) g(x)$$

L'influence de l'orientation se résume par :

- rapport signal/bruit : les dénominateurs du premier et du second termes de 2.55 sont égaux et indépendants de θ . Ils sont ignorés puisque nous nous intéressons seulement à l'étude de l'influence de θ sur le rapport signal/bruit. Evaluons les numérateurs :

$$\begin{aligned} f_x(x, y) * c(x, y, \theta)|_{(x,y)=(0,0)} &= A \int_{-\infty}^{+\infty} f(x) \int_{-x \cot \theta}^{+\infty} g(y) dy dx & (2.57) \\ &= A \int_{-\infty}^0 f_-(x) \int_{-x \cot \theta}^{+\infty} g_+(y) dy dx + \\ &\quad \int_0^{+\infty} f_+(x) \left[\int_{-x \cot \theta}^0 g_-(y) dy + \int_0^{+\infty} g_+(y) dy \right] dx \\ &= -4Ak \frac{\cos(\theta)(1 + 3\sin(\theta)\cos(\theta))}{\alpha^5(\sin(\theta) + \cos(\theta))^3} \end{aligned}$$

$$\text{avec } f_-(x) = \begin{cases} 0 & \text{si } x \geq 0 \\ -kxe^{\alpha x} & \text{si } x < 0 \end{cases} \quad \text{et} \quad f_+(x) = \begin{cases} 0 & \text{si } x < 0 \\ -kxe^{-\alpha x} & \text{si } x \geq 0 \end{cases}$$

Il en est de même pour $g_-(y), g_+(y)$.

$$\begin{aligned} f_y(x, y) * c(x, y, \theta)|_{(x,y)=(0,0)} &= A \int_{-\infty}^{+\infty} f(y) \int_{-ytg(\theta)}^{+\infty} g(x) dx dy & (2.58) \\ &= A \int_{-\infty}^0 f_-(y) \int_{-ytg(\theta)}^{+\infty} g_+(x) dx dy + \\ &\quad \int_0^{+\infty} f_+(y) \left[\int_{-ytg(\theta)}^0 g_-(x) dx + \int_0^{+\infty} g_+(x) dx \right] dy \\ &= -4kA \frac{\sin(\theta)(1 + 3\sin(\theta)\cos(\theta))}{\alpha^5(\sin(\theta) + \cos(\theta))^3} \end{aligned}$$

Les deux expressions 2.58 et 2.59 montrent bien que l'orientation du contour fournie par ce détecteur est égale à θ . La racine du numérateur de 2.55 est donnée par :

$$N_{sb}(\theta) = 4kA \frac{1 + 3\sin(\theta)\cos(\theta)}{\alpha^5(\sin(\theta) + \cos(\theta))^3}$$

Cette fonction possède un minimum au point $\theta = \pi/4$. Elle est symétrique par rapport à $\theta = \pi/4$, car $N_{sb}(\pi/2 - \theta) = N_{sb}(\theta)$.

- la localisation : en suivant la même démarche, on obtient :

$$N_{loc}(\theta) = \frac{4kA \sqrt{\sin^6(\theta) + \cos^6(\theta)}}{\alpha^3 (\sin(\theta) + \cos(\theta))^3}$$

Détecteur de Canny

Considérons maintenant le détecteur de Canny donné au paragraphe 1.1, partie 2. Il est facile de vérifier que $f_x^2(x, y) + f_y^2(x, y) = f_x^2(X, Y) + f_y^2(X, Y)$. Ce détecteur est alors invariant par rotation. Examinons ses performances :

$$N_{sb}(\theta) = \sqrt{2\pi} A \sigma^3$$

$$N_{loc}(\theta) = \sqrt{2\pi} A \sigma \sqrt{\sin^6(\theta) + \cos^6(\theta)}$$

Détecteur de Shen et Castan

Il est facile de vérifier que ce détecteur (voir le paragraphe 1.3, partie 2) est non invariant par rotation.

$$N_{sb}(\theta) = \frac{1}{\cos(\theta) + \sin(\theta)}$$

Pour le contour défini ci-dessus, l'erreur de localisation de GEF est nulle. Il en résulte que $N_{loc}(\theta) = 0$.

Résultat général

Pour généraliser ce résultat, montrons que la localisation de tout détecteur invariant ou variant par rotation et utilisant $(\frac{\partial}{\partial x})^2 + (\frac{\partial}{\partial y})^2$ comme opérateur de différentiation est affectée par l'orientation du contour. Pour ce faire, il suffit de montrer que le numérateur de Λ est non invariant par rotation pour toute fonction $f(x, y)$. Ceci revient à montrer que l'opérateur suivant est non invariant par rotation :

$$\left(\frac{\partial^3}{\partial x^3}\right)^2 + \left(\frac{\partial^3}{\partial y^3}\right)^2 \quad (2.59)$$

Considérons l'opérateur :

$$\mathcal{O}_{x,y}^T M \mathcal{O}_{x,y}$$

avec

$$\mathcal{O}_{x,y} = \left[\frac{\partial^3}{\partial x^3}, \frac{\partial^3}{\partial x^2 \partial y}, \frac{\partial^3}{\partial x \partial y^2}, \frac{\partial^3}{\partial y^3} \right]^T \quad \text{et} \quad M = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

a_{ij} sont des scalaires positifs compris entre zéro et un. Rappelons que l'opérateur est invariant par rotation si et seulement si $\mathcal{O}_{x,y} = R \mathcal{O}_{X,Y}$, R étant la matrice de rotation. En conséquence, pour que :

$$\mathcal{O}_{x,y}^T M \mathcal{O}_{x,y} = \mathcal{O}_{X,Y}^T M \mathcal{O}_{X,Y}$$

il faut que :

$$R^T M R = M \quad (2.60)$$

Calculons maintenant R. En utilisant la règle de dérivation d'une fonction composée on obtient :

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial X} \frac{\partial X}{\partial x} + \frac{\partial}{\partial Y} \frac{\partial Y}{\partial x} \quad \text{et} \quad \frac{\partial}{\partial y} = \frac{\partial}{\partial X} \frac{\partial X}{\partial y} + \frac{\partial}{\partial Y} \frac{\partial Y}{\partial y}$$

Alors

$$\frac{\partial^3}{\partial x^3} = \frac{\partial^3}{\partial X^3} \cos^3(\theta) - \frac{\partial^3}{\partial X^2 \partial Y} \sin(\theta) \cos^2(\theta) + \frac{\partial^3}{\partial Y^2 \partial X} \sin^2(\theta) \cos(\theta) - \frac{\partial^3}{\partial Y^3} \sin^3(\theta)$$

$$\frac{\partial^3}{\partial x^2 \partial y} = \frac{\partial^3}{\partial X^3} \sin(\theta) \cos^2(\theta) + \frac{\partial^3}{\partial X^2 \partial Y} \cos^3(\theta) + \frac{\partial^3}{\partial Y^2 \partial X} \sin^3(\theta) - \frac{\partial^3}{\partial Y^3} \sin^2(\theta) \cos(\theta)$$

Les dérivées $\frac{\partial^3}{\partial y^3}$ et $\frac{\partial^3}{\partial x^2 \partial y}$ sont obtenues directement en remplaçant $\cos(\theta)$ par $-\sin(\theta)$ et $\sin(\theta)$ par $\cos(\theta)$. La matrice R est alors donnée par :

$$R = \begin{bmatrix} \cos(\theta)^3 & -\cos(\theta)^2 \sin(\theta) & \cos(\theta) \sin(\theta)^2 & -\sin(\theta)^3 \\ \cos(\theta)^2 \sin(\theta) & \cos(\theta)^3 & \sin(\theta)^3 & \cos(\theta) \sin(\theta)^2 \\ \cos(\theta) \sin(\theta)^2 & -\sin(\theta)^3 & \cos(\theta)^3 & -\cos(\theta)^2 \sin(\theta) \\ -\sin(\theta)^3 & -\cos(\theta) \sin(\theta)^2 & -\sin(\theta) \cos(\theta)^2 & -\cos(\theta)^3 \end{bmatrix}$$

Dans le cas de l'opérateur donné par 2.59, les a_{ij} sont égaux à zéro sauf a_{11} et a_{44} . Dans ce cas particulier, il est facile de vérifier que la condition donnée par 2.60 n'est vraie que si $a_{11} = a_{44} = 0$. Il en résulte que l'opérateur 2.59 n'est pas invariant par rotation. En conclusion, la localisation est affectée par l'orientation du contour.

3

Analyse des résultats

Nous avons présenté dans le chapitre précédent le module responsable de la sélection d'un détecteur et de son exécution. Dans ce chapitre, il s'agit d'évaluer ces résultats pour pouvoir décider de la prochaine action à entreprendre, déterminer le nouveau contour exemple et corriger les éventuelles erreurs de sélection. La mise en œuvre de ces différentes fonctions nécessite de combiner les informations contours issues des détecteurs qui ont été choisis et exécutés.

Le premier paragraphe est consacré à la description des fonctionnalités de ce module. Nous présentons ensuite l'algorithme de fusion dans lequel les détecteurs qui ont été sélectionnés sont examinés et les erreurs de sélection corrigées. Ensuite, nous proposons l'évaluation des résultats et la définition du nouveau contour exemple. Finalement, nous décrivons un algorithme pour la fermeture des contours et un algorithme de mise en correspondance des segments de contours.

3.1 Analyse des résultats

Dans ce chapitre nous supposons que le système de sélection de détecteurs présenté dans cette partie a effectué plusieurs itérations, c'est-à-dire que plusieurs détecteurs ont été sélectionnés et exécutés. Le problème auquel nous nous intéressons concerne l'analyse des résultats obtenus jusqu'à maintenant. Ces résultats sont formés d'un ensemble de couples (l'image de contours, le détecteur qui l'a engendré).

L'objectif de cette analyse est double :

- évaluer : mesurer la différence entre le contour obtenu et le contour donné par l'utilisateur pour pouvoir décider de l'action à entreprendre. Dans le cas où le contour donné par l'utilisateur est détecté à quelques "trous" près, alors il suffit de fermer ce contour. En revanche, si la différence entre les deux est assez grande alors un nouveau contour exemple est déterminé et le système est relancé. Ce contour exemple correspond aux parties du contour donné par l'utilisateur qui n'ont pas encore été trouvées,
- corriger : nous avons signalé auparavant (cf. 1.2) que l'approche de sélection proposée est sujette aux erreurs dues au mauvais préconditionnement des détecteurs. La correction a pour objectif de mettre à jour la contribution de chacun des détecteurs sélectionnés, et d'éliminer ceux qui ont été sélectionnés par erreur.

La mise en œuvre de l'évaluation nécessite d'effectuer une comparaison du contour donné par l'utilisateur et chacune des images de contours obtenue pour identifier les contours similaires à ceux de l'exemple. La correction, à son tour, nécessite une comparaison des différents couples (image de contours, détecteur) pour identifier dans les images obtenues les contours similaires. Pour réduire le nombre de comparaisons effectuées à chaque itération, nous proposons la démarche suivante :

- combiner les informations contours obtenues par le dernier détecteur sélectionné et exécuté avec les résultats des précédents détecteurs. A ce niveau, la contribution des détecteurs est mise à jour et les détecteurs sélectionnés par erreur sont éliminés,
- utiliser l'image résultat de la fusion et l'exemple pour évaluer et générer le nouveau contour exemple.

En résumé, l'analyse des résultats passe par la conception d'un algorithme de fusion. Cet algorithme inclut la phase de correction de l'erreur de sélection. Nous présentons dans ce qui suit les algorithmes de fusion et d'évaluation.

3.2 Fusion d'images de contours

Comme nous l'avons défini dans le paragraphe précédent, la fusion consiste à étiqueter identiquement, avec un minimum d'erreurs, un contour appartenant à plusieurs images et provenant d'un même phénomène physique et à combiner les informations des contours similaires pour aboutir à une seule image. L'image résultat de la fusion doit être complète, en ce sens qu'elle doit contenir tous les contours présents dans les images à fusionner. Ces contours résultats de la fusion doivent être d'épaisseur un et avoir la meilleure localisation possible.

Le problème de fusion est l'un des problèmes complexes du traitement d'images [Hil 83, Lac 88, Lu 89]. Il n'a été abordé que peu de fois et uniquement dans un cas particulier : un seul détecteur gaussien exécuté à plusieurs échelles [Mar 80, Can 83, Wit 83, Ber 87, Lac 90, Tab 91a] (cf. 4.4 de la partie 1).

Marr et Hildreth [Mar 80] proposent une approche de fusion tirant partie du phénomène physique qui engendre un contour. Ils formulent deux hypothèses pour aplanir le problème de fusion. La première concerne la coïncidence spatiale des contours appelée "*spatial coincidence assumption*". Cette hypothèse se résume par : un contour qui apparaît à toutes les échelles dans la même position et ayant la même orientation indique la présence d'une variation d'intensité, conséquence d'un seul phénomène physique (changement de réflectance, illumination, profondeur ou orientation de surfaces). La seconde hypothèse indique qu'un contour n'est issu que d'un seul phénomène physique. Marr et Hildreth considèrent les trois phénomènes physiques qui sont à l'origine des classes de contours suivantes. A chacune de ces classes est associée une règle de fusion appropriée :

- les doubles marches de type crête : ce type de contour est généralement mal localisé à une échelle élevée. La fusion consiste alors à choisir les contours résultats de la plus petite échelle,
- les taches¹ : ce sont les contours de textures. La règle de fusion est identique à celle des doubles marches,

1. Traduction de *blobs*

- contours isolés : ils sont combinés par une mise en correspondance de segments et par le choix des contours résultats de la plus petite échelle. Au contour retenu est associée une mesure de plausibilité égale à la racine carré de la somme au carré des deux pentes obtenues à basse et à haute échelle.

Canny [Can 86] propose l'utilisation de son détecteur à différentes échelles et d'une stratégie *fine-to-coarse*. Son détecteur est exécuté à une échelle basse. Les résultats ainsi obtenus sont utilisés pour prédire la réponse du détecteur à l'échelle juste supérieure. La décision de remplacer un ancien point de contours par un nouveau est fondée sur la comparaison de la valeur résultat de la prédiction et de la valeur effective.

Bergholm [Ber 87] propose l'utilisation de la même stratégie d'analyse que Witkin (cf. 4.4 de la partie 1), c'est-à-dire *coarse-to-fine*. Le détecteur de Canny est utilisé à échelles décroissantes. La fusion se limite à remplacer le contour détecté à une échelle donnée par celui qui est détecté à une échelle inférieure. Lacroix [Lac 90] propose un algorithme similaire à celui de Bergholm et utilisant la stratégie inverse dite *fine-to-coarse*. Ces deux algorithmes fondés sur le suivi de contours posent des problèmes énormes. Le lecteur intéressé trouvera plus de détails dans [Tab 91a].

Ziou et Tabbone [Zio 91b, Tab 91a] proposent un algorithme de fusion fondé sur le comportement du contour dans un espace échelle. Plus précisément, nous considérons quatre modèles de contours : marche simple, marche brouillée, double marche de type crête et double marche de type escalier. Des règles de fusion propres à chaque classe de contour sont déduites à partir de l'étude du comportement de ces quatre modèles. Cet algorithme surmonte les problèmes posés dans les travaux de Bergholm et Lacroix.

Tous ces travaux se limitent à l'utilisation d'un seul détecteur stable dans un espace échelle (voir paragraphe 4.4 de la partie 1). Il est possible alors de suivre l'évolution du contour dans un espace-échelle et de tirer partie de celle-ci : le comportement du contour dans un espace échelle est souvent un indicateur du phénomène physique qui l'engendre. Les algorithmes ainsi proposés sont ad-hoc et n'apportent qu'une solution particulière au problème de la fusion. Nous proposons dans le paragraphe suivant un algorithme de fusion de résultats de détecteurs différents et non nécessairement stables.

3.2.1 Notre approche de fusion

Nous nous intéressons au problème de la fusion de contours de type marche issus de détecteurs non nécessairement stables et possédant des propriétés différentes. Pour ce faire, nous proposons un algorithme de fusion fondé sur la mise en correspondance de segments de contours. Il s'agit de mettre en correspondance les images à fusionner, puis de combiner les contours étiquetés identiques par une règle que nous définissons plus loin. Plus précisément, étant donné le contour exemple "celui donné par l'utilisateur" I_e , les contours résultats de la dernière fusion F_{N1} et les contours résultats de l'exécution du dernier détecteur I_d , l'algorithme de fusion de F_{N1} et I_d comporte trois étapes :

- l'élimination dans I_d de tous les contours qui n'appartiennent pas au contour donné par l'utilisateur. Cette élimination est effectuée par une mise en correspondance de I_d et de I_e en considérant que des contours de I_d ayant un correspondant dans I_e ,
- la mise en correspondance de I_d et F_{N1} suivant l'algorithme donné dans la section 3.5.5,

- la fusion des correspondants résultats de l'étape précédente.

Le choix du segment comme primitive de l'image est motivé par le fait qu'il permet de simplifier la tâche délicate de la mise en correspondance : le nombre de possibilités d'association de primitives est fortement diminué [WD 88]. Le segment est une primitive géométrique facile à modéliser et à extraire des images de contours par une approximation polygonale des chaînes construites par suivi de points de contours. A ce sujet, G. Giraudon [Gir 87] propose un algorithme efficace et performant de chaînage et d'approximation polygonale. Le logiciel correspondant est intégré dans notre système.

Nous présentons ci-dessous la troisième étape de l'algorithme de fusion des contours. L'algorithme de mise en correspondance est utilisé aussi dans l'étape d'évaluation. Nous le présentons en fin de ce chapitre. Notons que la règle d'élimination des contours décrite dans l'algorithme de fusion ci-dessus est triviale et ne nécessite pas un développement particulier. Elle n'est donc pas présentée dans ce qui suit.

3.2.2 La règle de fusion des contours

Le problème de fusion peut être posé de la manière suivante : soient I une image donnée, I_m et I_c deux images de contours obtenues à partir de I par des détecteurs différents. On cherche l'image F_N la plus complète possible, telle que les contours soient bien localisés, d'épaisseur 1 et rapidement calculables. On note :

$$F_N = \{c_1, \dots, c_N\}$$

avec $c_i \in I_m$ ou $c_i \in I_c$ ou $c_i = s_i \oplus (s_{i_1}, s_{i_2}, \dots)$ ou $c_i = (s_{i_1}, s_{i_2}, \dots) \oplus s_i$, $s_i \in I_m$ et $s_{i_j} \in I_c$.

s_i désigne un segment de l'image ; \oplus une opération non commutative ; $s_i \oplus s_j$ signifie "compléter le segment s_i par le segment s_j ".

La figure 3.1 décrit les quatre configurations utilisées lors de la fusion de deux segments de contours.

La construction de l'ensemble F_N est effectuée de manière incrémentale : pour chaque couple de correspondant, le meilleur c_i est retenu. Plus précisément, l'ensemble D de toutes les combinaisons possibles (soit 4^N possibilités) est défini par :

$$D = \{F_N^1, \dots, F_N^{4^N}\}$$

Pour réduire la combinatoire, nous considérons une fonction heuristique $C(n)$, où n indique le numéro de l'étape courante ($n \in [0, N]$). Cette fonction définie sur D représente le coût nécessaire pour l'ajout d'un c_i à la solution déjà construite.

La fusion se limite alors à la recherche de l'ensemble F_N^* qui minimise la fonction $C(n)$

La définition de la fonction coût $C(n)$ fait intervenir les caractéristiques du détecteur, les caractéristiques du contour calculé, le contour donné par l'utilisateur et la qualité du résultat requis. Plus précisément, soient le quadruplet $(c_n^{(1)} = m_n, c_n^{(2)} = S_n^{p_n}, c_n^{(3)} = m_n \oplus S_n^{p_n}, c_n^{(4)} = S_n^{p_n} \oplus m_n)$ et Ω_i l'ensemble des détecteurs ayant mis en évidence les contours de $c_n^{(i)}$. L'estimation du coût nécessaire à l'ajout de $c_n^{(i)}$ à la solution déjà construite F_{n-1} repose sur les facteurs suivants :

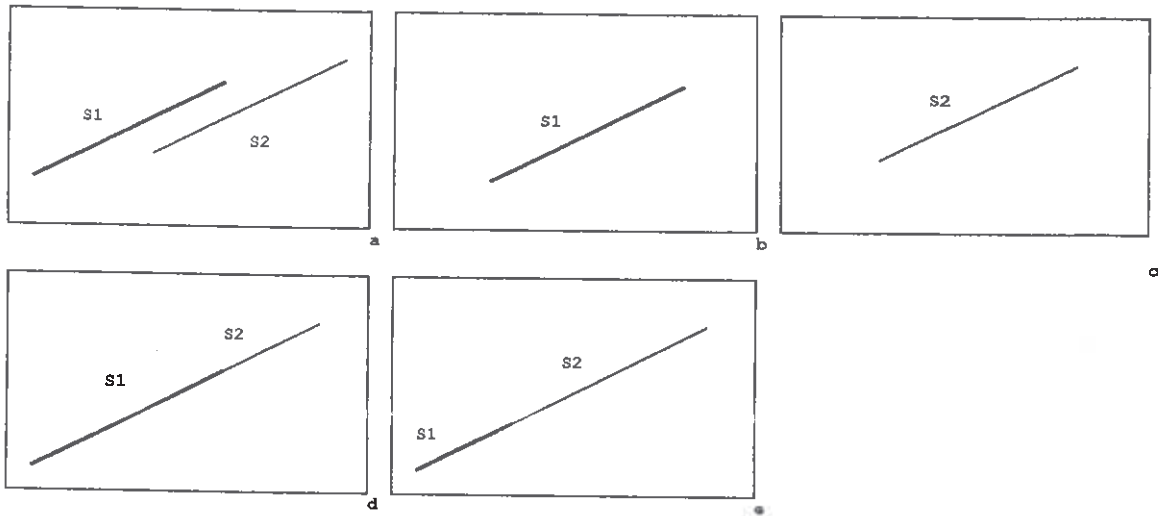


Figure 3.1. a) deux segments de contours à fusionner, b), c), d) et e) représentent les quatre solutions possibles : s_1 , s_2 , $s_1 \oplus s_2$ et $s_2 \oplus s_1$.

- la longueur de $c_n^{(i)}$ est définie par :

$$e_{lg}(n) = e(n-1) + 1 - \frac{\sum_{d \in \Omega_i} lg(d)}{\sum_{i=1}^4 \sum_{d \in \Omega_i} lg(d)} \quad (3.1)$$

où $lg(d)$ est la longueur des contours résultat d'un détecteur d et $e_{lg}(0) = 0$.

- l'erreur de localisation est donnée par :

$$e_{loc}(n) = e_{loc}(n-1) + \frac{\sum_{d \in \Omega_i} loc(d)}{\sum_{i=1}^4 \sum_{d \in \Omega_i} loc(d)} \quad (3.2)$$

où $loc(d)$ est l'erreur de localisation du détecteur d et $e_{loc}(0) = 0$

- le temps de calcul est défini par :

$$e_t(n) = e_t(n-1) + \frac{\sum_{d \in \Omega_i} tc(d)}{\sum_{i=1}^4 \sum_{d \in \Omega_i} tc(d)} \quad (3.3)$$

où $tc(d)$ est le temps de calcul du détecteur d mesuré sur une machine donnée et $e_t(0) = 0$

La fonction $C(n)$ est une combinaison linéaire des trois facteurs (eqs. 3.1, 3.2, 3.3). Elle est donnée par :

$$C(n) = \alpha e_{lg}(n) + \beta e_{loc}(n) + (1 - \alpha - \beta) e_t(n) \quad (3.4)$$

où α et β sont des scalaires compris entre 0 et 1 et ayant pour objectif la caractérisation de la solution requise (cf. 1.3). La mesure $C(n)$ inclut des informations relatives

aux performances du détecteur. Elle produit donc un algorithme de fusion général qui inclut des détecteurs non nécessairement stables. Cette mesure peut fournir une image de contours détectés correctement, possédant une faible erreur de localisation et se calculant rapidement.

Etant donnée la fonction $C(n)$, il s'agit de déterminer l'image de fusion F_n . Ce problème est un classique de l'algorithmique et de l'intelligence artificielle [Nil 71, Aho 74, Bar 87, Pai 88, Hat 90]. Il est connu sous la dénomination de problème de recherche heuristique. L'algorithme de recherche de la solution optimale est connu sous la dénomination A^* . Cet algorithme est donné en Annexe D.

Examinons à présent l'admissibilité de l'algorithme A^* (c'est-à-dire, la garantie de la solution optimale) en utilisant la fonction $C(n)$. Dans le cas général, de tel algorithme ne peut garantir une solution optimale que si pour tout $n > 1$ on a $C(n) - C(n-1) > 0$ et $C(n^*) \leq C(n)$ ($C(n^*)$ est la coût estimé et $C(n)$ le coût réel) [Nil 71]. Il est facile de voir que notre fonction $C(n)$ (eq. 3.4) vérifie la première condition. Quant à la seconde condition, le coût estimé à l'aide de $C(n)$ correspond au coût exacte du chemin optimal entre l'étape 0 et l'étape n . $C(n)$ produit donc un algorithme admissible.

3.2.3 Résultats de la fusion

Considérons les deux images de contours décrites par les figures 3.2.a et 3.2.b. Ces deux images sont obtenues à partir de l'image 2.9 par les détecteurs DRF avec $a = 0.27$ et DGD avec $\alpha = 0.9$. Les contours dans les deux images qui ne correspondent pas au contour donné par l'utilisateur ont été éliminés. Les deux images sont approchées par des segments de droites en utilisant le logiciel de chaînage et d'approximation polygonale de G. Giraudon [Gir 87]. Elles sont ensuite mises en correspondance.

L'image présentée dans la figure 3.2.c décrit le résultat de la fusion dans le cas où les scalaires intervenants dans la définition de la fonction heuristique pour spécifier la contribution de chacun de ces composantes ont pour valeur $\alpha = 1$ et $\beta = 0$. Les contours résultats de la fusion sont tous issus du détecteur DGD (la contribution du détecteur DRF est de zéro pixel). Un examen de ces résultats montre que le contour oblique est trouvé à la fois par les deux détecteurs. Lors de la fusion le contour issu du détecteur DGD a été sélectionné. Ce choix n'est qu'arbitraire, puisque le critère de fusion n'est fondé que sur la longueur des contours et ces deux contours possèdent la même longueur.

Cependant, la prise en considération du temps de calcul et de l'erreur de localisation ($\alpha = \beta = 0.5$) fournit le résultat décrit par l'image 3.2.d. Dans ce cas, c'est le contour fourni par DRF qui est considéré, car ce détecteur est plus efficace (3 fois plus rapide que DGD) et possède en général une faible erreur de localisation. Dans l'image résultat de la fusion, 55,7% des points de contours sont dûs au détecteur DRF et le reste au détecteur DGD.

3.3 Evaluation

Rappelons que l'évaluation a pour objectif de déterminer le nouveau contour exemple et de mesurer la différence entre le contour calculé et celui donné par l'utilisateur. Ces deux fonctions sont décrites ci-dessous.

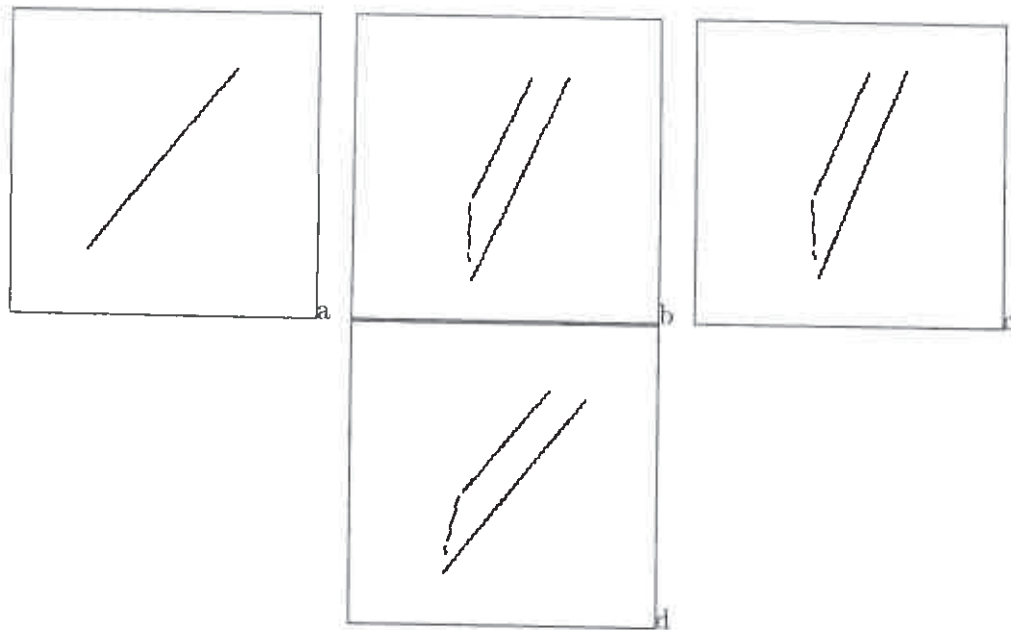


Figure 3.2. a) et b) images de contours résultats des détecteurs *DRF* avec $a = 0.27$ et *DGD* avec $\alpha = 0.9$, c) et d) images résultats de la fusion pour ($\alpha = 1$ et $\beta = 0$) et ($\alpha = \beta = 0.5$).

3.3.1 Génération du contour exemple

Etant donné le contour fourni par l'utilisateur, la relation d'adjacence entre ces contours et le contour calculé, il s'agit de déterminer les parties du contour donné par l'utilisateur qui n'ont pas été encore détectées. Ces contours forment le contour exemple qui sera considéré comme étant l'objectif à atteindre dans la prochaine itération. La relation d'adjacence de segments est construite à partir de la relation de voisinage donnée en Annexe C. Deux segments de contours sont dits adjacents s'ils sont voisins et la distance entre deux de leurs extrémités est faible.

La génération du contour exemple est effectuée par comparaison du contour calculé et du contour donné par l'utilisateur en considérant les trois configurations données par la figure 3.3. La détermination de ces trois types de "trous" nécessite une mise en correspondance des contours résultats de la fusion et du contour donné. Puisque tous les contours appartenant à l'image de fusion ont été mis en correspondance avec le contour exemple, l'appariement est obtenu directement (c'est-à-dire sans exécuter l'algorithme donné au paragraphe 3.5) en sauvegardant pour chaque segment de contour calculé son correspondant dans le contour donné.

La figure 3.4 montre le nouveau contour exemple obtenu à partir du résultat de la fusion décrit dans la figure 3.2.a.

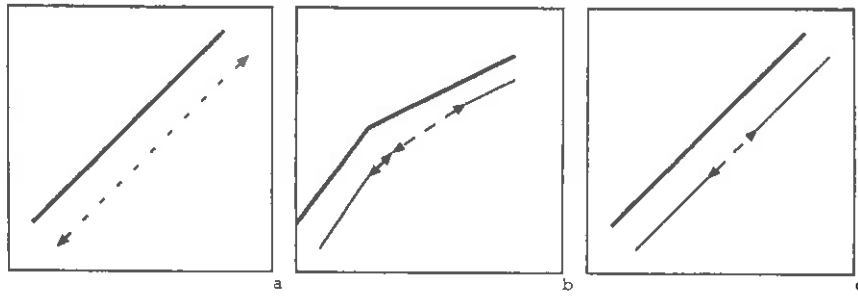


Figure 3.3. Détermination des trous dans le contour calculé : a) trou engendré par un segment du contour donné n'ayant pas de correspondant dans le contour calculé, b) trou entre segments adjacents du contour donné, c) trou entre segments du contour calculé ayant le même correspondant dans le contour donné.

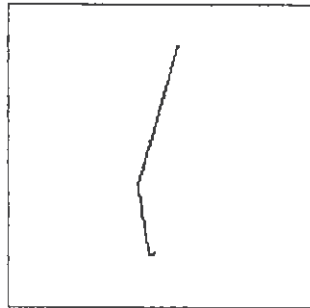


Figure 3.4. Le contour exemple obtenu à partir des résultats de la fusion donnés par la figure 3.2.a

3.3.2 Mesure de la différence

Le contour exemple ainsi déterminé exprime la différence entre le contour donné par l'utilisateur et celui qui est calculé. Il s'agit maintenant de définir une mesure pour quantifier cette différence. Cette mesure sert de condition d'arrêt, c'est-à-dire qu'elle permet de vérifier si le contour donné est trouvé. Pour ce faire, nous décrivons les trois facteurs intervenant dans la définition de cette mesure :

- le cumul de la longueur des h-contours non encore trouvés : il est mesuré par le rapport de ce cumul l_c sur la longueur l_e du contour donné :

$$c = \frac{l_c}{l_e}$$

- la longueur moyenne des "trous" : cette mesure indique si les trous sont de longueur importante. Elle est définie par :

$$l_T = \sum_{\forall t} \min(1, \frac{l_t}{l_{tm}}) / \text{nombre de trous}$$

où l_{tm} est la longueur maximale autorisée d'un trou, l_t la longueur du trou.
 l_r est d'autant plus proche de 0 que la longueur moyenne des trous est faible.

- la longueur relative moyenne des h-contours : cette mesure supplémentaire indique si les contours adjacents à un trou sont relativement plus importants que le trou. L'intérêt principal de cette mesure réside dans le fait que pour certaines méthodes de fermeture de contour, il est plus facile de fermer un contour si la longueur du trou est moins importante que la longueur des contours adjacents. Soit t un trou entre les segments s_i , la longueur relative de t est mesurée par :

$$l_r = \frac{1}{2} - \sum_{\forall t} \frac{l_s - l_t}{2 \max(l_s, l_t)} / \text{nombre de trous}$$

où l_s est la longueur moyenne des segments s_i , l_t la longueur du trou. La mesure l_r est d'autant plus proche de 0 que la longueur des segments est plus importante que la longueur des trous.

La mesure de différence peut alors être définie par :

$$diff = c * \max(l_T, l_r)$$

Cette mesure est d'autant plus proche de 0 que le contour calculé est proche du contour donné. La fonction $\max()$ permet d'obtenir un contour calculé dont la longueur et la longueur relative des trous sont toutes deux faibles.

Quand la mesure *diff* est faible (ou qu'aucun détecteur ne peut être sélectionné), il s'agit de calculer la contribution de chacun des détecteurs qui font partie de la solution. Pour ce faire, à chaque contour est associé le détecteur qui l'a mis en évidence. Dans l'implantation actuelle, la contribution d'un détecteur est mesurée seulement par le pourcentage des points de contours qu'il a mis en évidence et qui appartiennent au contour exemple.

3.4 Fermeture des contours

La fermeture de contours est effectuée une fois que l'objectif est atteint ou bien qu'aucun détecteur ne peut être sélectionné. Son but est de compléter le contour résultat. Elle peut être considérée comme étant un détecteur particulier et ajoutée à la solution finale. Le problème de la fermeture de contours soulève les deux questions suivantes : quand faut-il fermer ? comment fermer ? En l'absence d'information image (e.g., la plausibilité du contour), la fermeture de contours est considérée souvent comme étant un processus de regroupement perceptuel. En effet, il s'agit de former à partir des contours de l'image d'autres contours plus longs en utilisant les principes d'organisation perceptuelle connus sous la dénomination Gestalt [Par 85, Moh 89a, Tuc 87, Ahu 89]. Une fois la fermeture de contours décidée, il suffit de joindre chaque paire de contours par une courbe. La nature de la courbe à utiliser a fait l'objet de quelques travaux, en particulier dans le cas du contour subjectif. Ullman [Ull 76] suggère de fermer les trous par deux cercles. Le premier tangent au premier contour et le second tangent au deuxième de manière à ce que les deux arcs possèdent une courbure minimale. W.R. Rutkowski [Rut 79] propose de fermer par un polynôme de degrés trois. J.A. Webb and E. Pervin [Web 84] proposent de fermer avec une parabole.

L'algorithme que nous proposons est inspiré de celui proposé dans [Tuc 87]. Il utilise comme critère de fermeture la proximité et la continuité lissée. Nous avons adapté cet algorithme à notre cas en considérant le contour exemple et en l'enrichissant par la prise en compte de l'information photométrique. Pour des raisons de simplicité, la fermeture est effectuée par des segments de droites. Cet algorithme est fondé sur le paradigme prédiction/vérification d'hypothèses. Il s'agit dans une première étape de générer les différentes hypothèses, c'est-à-dire les différentes possibilités de fermeture des contours. Ces possibilités sont les trous obtenus par l'étape de génération du contour exemple. A chaque trou sont associés les étiquettes trou (T) et non trou (NT).

La vérification des hypothèses générées est effectuée en associant à chaque étiquette une mesure de plausibilité et en affinant cette mesure par relaxation continue. L'algorithme de relaxation est présenté dans la section 3.5.2. Trois facteurs interviennent dans l'estimation des probabilités initiales :

- La longueur du trou : soient i un trou et $V(i)$ l'ensemble des trous voisins de i (l'ensemble des trous partageant un contour avec i); la longueur relative de i est définie par :

$$d_i = (lm_i - l_i) / \max(lm_i, l_i)$$

où lm_i est la longueur moyenne des arcs $j \in V(i)$ et l_i la longueur de i . La longueur relative d_i prend ses valeurs dans l'intervalle $[-1, 1]$. Elle est positive si l_i est inférieure à la longueur moyenne des trous de $V(i)$ et est négative sinon. La valeur positive renforce l'étiquette T et la valeur négative renforce l'étiquette NT .

- Le contraste du trou : soit s_1, s_2, \dots, s_n des segments partageant au moins une extrémité avec le trou i , le contraste relatif de i est défini par :

$$c_i = (v_i - v_{s_i}) / \max(v_i, v_{s_i})$$

où v_{s_i} et v_i sont respectivement le contraste moyen des segments s_i et du trou i . Le contraste c_i prend ses valeurs dans l'intervalle $[-1, 1]$. Il est positif si v_i est supérieur à v_{s_i} et est négatif sinon. Comme dans le cas précédent, la valeur positive renforce l'étiquette T et la valeur négative renforce l'étiquette NT .

Puisque nous fixons comme objectif la fermeture des trous de petites longueurs relatives ou bien ceux possédant un contraste fort, les deux mesures d_i et c_i sont alors combinées de la manière suivante :

$$f_i = (c_i + d_i) / 2$$

La mesure f_i prend aussi ses valeurs dans l'intervalle $[-1, 1]$. Elle est proche de 1 (resp. -1) quand f_i et d_i sont tous les deux proches de 1 (resp. -1).

- L'orientation du trou : la décision de fermer un trou (c'est-à-dire de lui affecter l'étiquette T) est renforcée si la courbure totale du contour résultat est faible. La prise en considération de ce critère implique le calcul de l'orientation entre les segments qui partagent une extrémité avec le trou. Pour que le contour résultat soit

lisse, il faut que deux de ces segments s_1 et s_2 se trouvant de part et d'autre du trou i forment deux angles θ_1 et θ_2 qui soient tous les deux faibles. L'angle $\theta = (\theta_1 + \theta_2)/2\pi$ doit être alors petit. La contribution de l'orientation est donnée par :

$$o_i(T) = e^{-\pi(\theta/(1-\theta))^2} \quad \text{et} \quad o_i(NT) = \sqrt{1 - o_i^2(T)}$$

Une faible valeur de θ entraîne une valeur de $o_i(T)$ proche de 1 et une valeur de $o_i(NT)$ proche de 0.

La mesure d'orientation précédente ne tient compte que du trou i . Pour avoir une vue plus globale, l'étiquetage de i doit tenir compte de l'étiquetage de ses voisins $V(i)$. Une mesure d'orientation supplémentaire est alors introduite :

$$os_{ij}(\lambda) = \min(o_i(\lambda), o_j(\lambda))$$

os_{ij} est proche de 1 si la courbe définie par les deux trous i, j et le segment de contour qui se trouve entre les deux est lisse.

Soit s_i la contribution de la distance, du contraste et de l'orientation donnée par :

$$s_{i,\lambda} = \begin{cases} \min(f_i, f_i \times o_i(\lambda)) & \text{si } \lambda = T \\ \min(-f_i, f_i \times o_i(\lambda)) & \text{sinon} \end{cases}$$

Un examen de cette formule montre que l'orientation n'est utilisée que si $f_i > 0$. La courbe résultat est alors nécessairement lisse. Soient $j \in V(i)$, la compatibilité des deux trous i et j pour chaque couple d'étiquette (λ_1, λ_2) est définie par :

$$c_{ij}(\lambda_1, \lambda_2) = \begin{cases} (s_{i,\lambda_1} + s_{j,\lambda_2}) \times os_{ij} & \text{si } \lambda_1 = \lambda_2 = T \\ s_{i,\lambda_1} + s_{j,\lambda_2} & \text{sinon} \end{cases}$$

La mesure os_{ij} ne concerne que l'étiquette T . Elle assure ainsi une courbe résultat qui est lisse. Le coefficient c_{ij} est normalisé pour prendre ses valeurs dans l'intervalle $[-1, 1]$ de la façon suivante :

$$c_{ij}(\lambda_1, \lambda_2) = \frac{c_{ij}(\lambda_1, \lambda_2)}{\max_{(\lambda_1, \lambda_2)} c_{ij}(\lambda_1, \lambda_2) - \min_{(\lambda_1, \lambda_2)} c_{ij}(\lambda_1, \lambda_2)}$$

La compatibilité c_{ij} des deux trous i et j est proche de 1 si la longueur relative de i et j est positive et la courbe formée à partir des trois segments est lisse. En conséquence, la probabilité initiale est donnée par :

$$p_i^{(0)}(\lambda_1) = \sum_j w_{ij} \max_{\lambda_2} \{(c_{ij}(\lambda_1, \lambda_2) + 1)/2\}$$

Puisque nous avons $\sum_{\lambda} p_i^{(0)}(\lambda) = 1$, les $p_i^{(0)}$ sont alors normalisées par :

$$p_i^{(0)}(\lambda_1) = \frac{p_i^{(0)}(\lambda_1)}{\sum_{\lambda_2} p_i^{(0)}(\lambda_2)}$$

Les coefficients w_{ij} représentent la contribution de j dans $V(i)$:

$$w_{ij} = \text{card}(V(i)) \frac{\min_{k \in V(i)} l_k - 1}{l_j}$$

Les w_{ij} sont normalisés par :

$$w_{ij} = \frac{w_{ij}}{\sum_j w_{ij}}$$

La consistance initiale $q_i^{(0)}$ est donnée par la relation :

$$q_i^{(0)}(\lambda_1) = \sum_j w_{ij} \left(\sum_{\lambda_2} \frac{c_{ij}(\lambda_1, \lambda_2) + 1}{2} p^{(0)}(\lambda_2) \right)$$

Notons que dans la version implantée de l'algorithme de fermeture de contours, l'information photométrique n'est pas considérée. Cet algorithme a été testé sur une variété d'images de contours. La figure 3.5.b présente une image de contours fermés qui sont obtenus à partir de l'image résultat de la fusion donnée dans la figure 3.2.c. On remarque que ce contour n'a pas été fermé complètement à cause de la contrainte de lissage. La considération de l'information photométrique permettra certainement une fermeture complète. Les figures 3.5.c et 3.5.d présentent un autre exemple de résultats. Notons que dans ce cas, la fermeture est effectuée sans considérer le contour exemple. La génération des hypothèses de fermeture est accomplie à l'aide du graphe de Delaunay : chaque arc du graphe de Delaunay est une hypothèse (cf. Annexe C).

3.5 La mise en correspondance de segments

Etant données deux images de segment de contours I_m et I_c obtenues à partir de la même image de niveaux de gris, par le même détecteur exécuté à deux échelles différentes ou bien par deux détecteurs différents, il s'agit de trouver pour chaque segment de l'image I_m , les segments correspondants dans l'image I_c et inversement. Les particularités du problème de mise en correspondance sont les suivants :

- un segment de I_m peut avoir plusieurs segments de I_c comme correspondants,
- les contours, dans le cas des images de scènes extérieures, comportent souvent des "zigzags". Les segments possèdent alors une petite longueur. En conséquence, l'estimation de leurs attributs ne peut être qu'entachée d'erreurs,
- les contours des deux images à mettre en correspondance coïncident à une erreur de localisation près,
- aucune transformation géométrique n'est à considérer.

L'algorithme de mise en correspondance que nous proposons est fondé sur l'étiquetage stochastique. Il est inspiré des travaux [Fau 81, Bha 81, Bro 81a, LL 86, BA 87]. Cet algorithme comporte trois étapes :

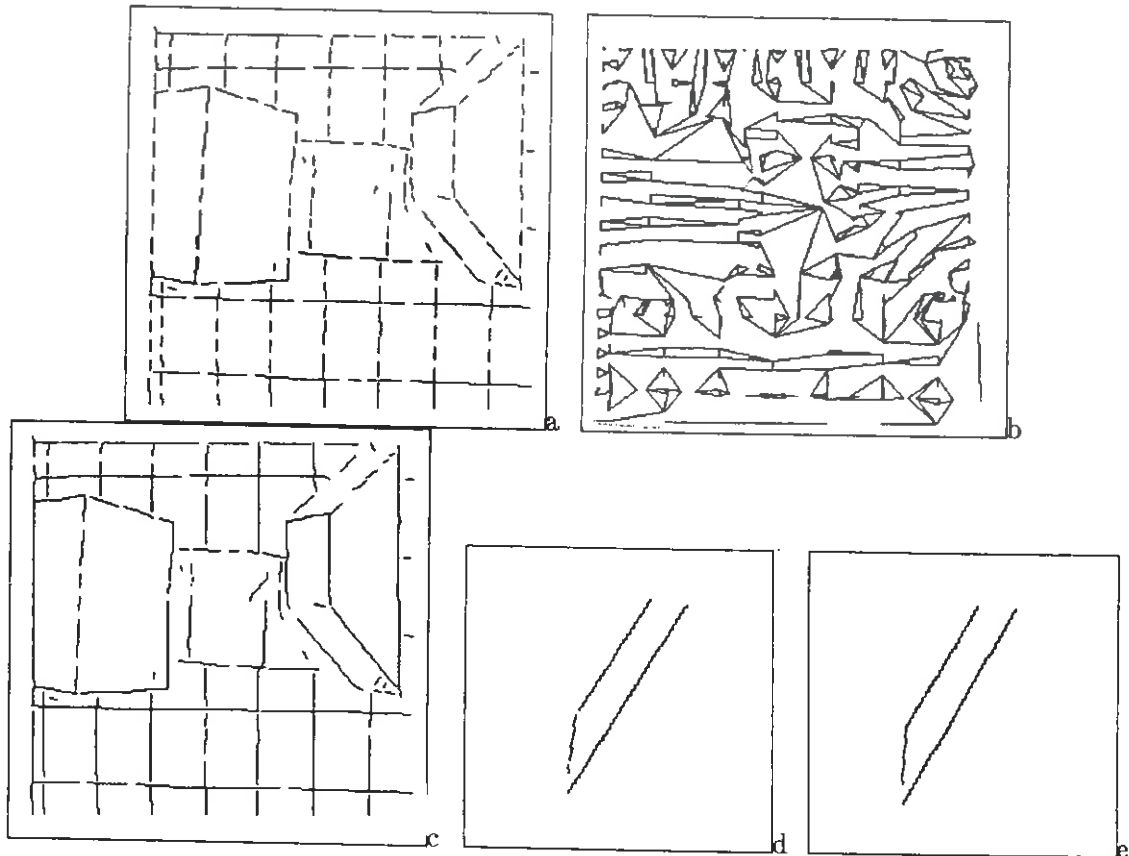


Figure 3.5. a) un premier exemple de contour à fermer, b) le graphe de Delaunay correspondant, c) les contours après fermeture, d) un deuxième exemple de contour à fermer, e) les contours après fermeture.

- la première étape a pour objectif de définir pour un segment de l'image I_m tous les segments de l'image I_c qui sont susceptibles d'être ses correspondants. Le couple $h_i = (m_i, (s_{i_1}, \dots, s_{i_{L_i}}))$ formé par le segment de I_m et les segments de I_c est appelé *hypothèse*.
- la deuxième étape consiste à estimer pour chaque couple (m_i, s_{i_k}) la probabilité pour que s_{i_k} soit correspondant de m_i .
- la dernière étape a pour objectif de sélectionner dans h_i les segments $(s_{i_{p_1}}, \dots, s_{i_{p_i}})$, où $p_i \leq L_i$, qui sont les meilleurs correspondants de m_i .

3.5.1 Génération d'hypothèses

Considérons I_m comme modèle et I_c comme image (ce choix est arbitraire). Supposons que $I_m = \{m_1, m_2, \dots, m_N\}$ et $I_c = \{s_1, s_2, \dots, s_L\}$, où m_i et s_j représentent respectivement un segment de I_m et un segment de I_c . Pour chaque segment m_i de I_m , on définit son voisinage dans l'autre image par la construction d'une fenêtre de longueur égale à la

longueur de m_i et de largeur donnée. Les segments $(s_{i_1}, s_{i_2}, \dots, s_{i_{L_i-1}})$ de I_c appartenant à cette fenêtre sont considérés comme les correspondants possibles du segment modèle. Nous avons ajouté aux correspondants de $m_i : (s_{i_1}, s_{i_2}, \dots, s_{i_{L_i-1}})$ un segment "fictif" $s_{i_{L_i}} = nil$. Son rôle est de donner au segment m_i la possibilité de ne pas avoir de correspondant. Le résultat de cette étape est un ensemble :

$$H = \{(m_1, (s_{1_1}, s_{1_2}, \dots, s_{1_{L_1}})), (m_2, (s_{2_1}, s_{2_2}, \dots, s_{2_{L_2}})), \dots, (m_N, (s_{N_1}, s_{N_2}, \dots, s_{N_{L_N}}))\}$$

3.5.2 Estimation de probabilités

Etant donné l'ensemble d'hypothèses $H = \{h_1, h_2, \dots, h_N\}$, où $h_i = (m_i, (s_{i_1}, s_{i_2}, \dots, s_{i_{L_i}}))$ est la $i^{\text{ème}}$ hypothèse, il s'agit d'estimer pour chaque couple (m_i, s_{i_k}) , pour tout $i \in [1, N]$ et pour tout $k \in [1, L_i]$, la probabilité $p(i, k)$ reflétant la confiance que l'on peut associer à cette correspondance.

Pour une hypothèse donnée, l'ensemble de ces probabilités forme un vecteur $\vec{P}_i = [p(i, 1), \dots, p(i, L_i)]^T$. Les $p(i, k)$ vérifient les deux contraintes probabilistes suivantes :

$$\sum_{k=1}^{L_i} p(i, k) = 1 \quad (3.5)$$

$$0 \leq p(i, k) \leq 1, \quad \forall k \in [1, L_i] \quad (3.6)$$

L'estimation d'un vecteur de probabilités \vec{P}_i doit être :

- consistante : la mise en correspondance de s_{i_k} et de m_i doit être compatible avec la mise en correspondance des tous les voisins de m_i . En d'autres termes, l'estimation du vecteur \vec{P}_i doit être contextuelle. Le contexte est représenté par l'ensemble des voisins de m_i ,
- non ambiguë : chaque segment s_{i_k} est mis en correspondance avec un et seul segment m_i .

La considération de l'information contextuelle nécessite l'introduction d'une mesure de compatibilité $q(i, j)$ pour chaque couple de segments (m_i, s_{i_k}) . L'ensemble de mesures pour une hypothèse forme un vecteur $\vec{Q}_i = [q(i, 1), \dots, q(i, L_i)]^T$. Intuitivement, ce vecteur représente ce que "pensent" les voisins de m_i de ses correspondants. Le problème d'estimation du vecteur de probabilité \vec{P}_i peut être transformé alors en un problème d'optimisation. Il s'agit de déterminer le vecteur \vec{P}_i qui maximise la consistance et minimise l'ambiguïté. Ceci revient alors à maximiser la quantité [Bha 81, Fau 81] :

$$C_i = \vec{P}_i \cdot \vec{Q}_i$$

où \cdot est le produit scalaire. La quantité C_i atteint sa valeur maximale 1 si $\vec{P}_i = \vec{Q}_i$. En revanche, elle est d'autant plus petite que la différence entre \vec{P}_i et \vec{Q}_i est plus grande. Il est possible de généraliser cette quantité locale pour prendre en compte toutes les hypothèses. Il s'agit de maximiser la quantité globale suivante en présence des contraintes (eqs. 3.5, 3.6) :

$$C = \sum_{i=1}^{i=N} \vec{P}_i \cdot \vec{Q}_i \quad (3.7)$$

où N est le nombre total d'hypothèses. Comme C est une fonction non linéaire, nous sommes alors en présence d'un problème d'optimisation d'une fonction non linéaire en présence de contraintes inégalités. Le gradient projeté est l'un des algorithmes classiques applicables dans ces cas. Les détails mathématiques relatifs à cet algorithme peuvent être trouvés dans [Bou 68, Fau 81, Bha 81]. La solution est donnée sous une forme récurrente par :

$$p^{(n+1)}(i, k) = p^{(n)}(i, k) + \rho_i^{(n)} Proj_i^{(n)} \left(\frac{\partial C}{\partial p(i, k)} \right) \quad (3.8)$$

où n le numéro de l'étape,

$Proj_i^{(n)}()$ est un opérateur de projection. $Proj_i^{(n)}(\partial C \partial p(i, k))$ correspond au pas de variation du gradient au point $\frac{\partial C}{\partial p(i, k)}$,

$\rho^{(n)}$ détermine le sens de variation du gradient.

Les règles de calcul du pas et du sens de variation sont données en Annexe A. La convergence de cet algorithme est assurée. Nous avons également implanté l'algorithme de Rosenfeld et al. [Ros 76]. L'expérimentation montre que ce dernier converge moins vite que celui décrit ci-dessus.

3.5.3 Initialisation des probabilités

Les attributs des segments de contours qui interviennent dans la mise en correspondance, plus précisément dans cette étape d'initialisation du processus de relaxation, sont uniquement de nature géométrique : la distance entre segments, le recouvrement, l'orientation des segments et la colinéarité. Ces attributs sont définis en Annexe B.

La probabilité initiale est donnée par :

$$p^{(0)}(i, k) = \begin{cases} 1 - \max_{k \in [1, p_i]} p(i, k) & \text{si } s_k = nil \\ e^{-\pi \left(\frac{c(i, k)}{1 - c(i, k)} \right)^2} & \text{sinon} \end{cases}$$

$c(i, k)$ est donné par l'équation 3.21. Puisque les $p(i, k)$ sont liées par la contrainte probabiliste (eq. 3.5), on obtient alors :

$$p^{(0)}(i, k) = \frac{p^{(0)}(i, k)}{\sum_{l=1}^{L_i} p^{(0)}(i, l)} \quad (3.9)$$

La compatibilité entre le couple (m_i, s_{i_k}) et le couple (m_j, s_{j_l}) , où m_j appartenant à l'ensemble des voisins de m_i , noté $voisin(m_i)$, est définie par :

$$r(i, k, j, l) = \begin{cases} p(i, k) & \text{si } l = nil \\ p(j, l) & \text{si } k = nil \\ 1 - c(i, k) \cdot c(i, l) & \text{sinon} \end{cases}$$

L'algorithme de construction de la relation voisinage est donné en Annexe C. Remarquons que $r()$ est symétrique, car nous avons $r(i, k, j, l) = r(j, l, i, k)$. Puisque le segment

m_j peut avoir L_j segments correspondants, la mesure de compatibilité peut alors être donnée par :

$$Q(i, k, j) = \sum_{l=1}^{L_j} r(i, k, j, l) p(j, l)$$

Cette quantité est d'autant plus grande que la correspondance de m_i et m_j est compatible. En généralisant à tous les voisins de m_i , on obtient :

$$Q(i, k) = \frac{\sum_{j \in \text{voisin}(m_i)} Q(i, k, j)}{\text{card}(\text{voisin}(m_i))}$$

En supposant que les $Q(i, k)$ sont également liés par la contrainte $\sum_{k=1}^{L_i} Q(i, k) = 1$. On obtient :

$$q^{(0)}(i, k) = \frac{Q(i, k)}{\sum_{l=1}^{L_i} Q(i, l)} \quad (3.10)$$

3.5.4 Sélection de correspondants

La relaxation permet d'estimer pour chaque couple (m_i, s_{i_k}) d'une hypothèse h_i la probabilité $p(i, k)$. Sachant qu'à un segment m_i peut correspondre plusieurs segments s_{i_k} , il s'agit alors de déterminer le sous-ensemble $S = \{c_1, \dots, c_m\} \subseteq \{s_{i_1}, \dots, s_{i_{L_i}}\}$ ayant la meilleure correspondance avec m_i . Deux facteurs interviennent :

- la probabilité : la probabilité pour que les segments de S soient correspondants à m_i doit être maximale. En d'autres termes, l'erreur de probabilité doit être minimale. Cette dernière est définie par :

$$e_p(i) = 1 - \min_{k \in [1, m]} p(i, k)$$

- la colinéarité : les segments appartenant au sous-ensemble S doivent être colinéaires. Soit E_S l'ensemble de toutes les combinaisons possibles des segments de S pris deux à deux.

$$E_S = \{(c_1, c_2), (c_1, c_3), \dots, (c_{m-1}, c_m)\}$$

Nous introduisons alors la mesure de colinéarité d'un sous-ensemble S de segments :

$$c(i) = \min_{s_{i_k}, s_{i_j} \in E_S} cc(k, j)$$

$cc(k, j)$ est une mesure de colinéarité donnée par l'équation 3.20. Nous proposons de normaliser la mesure $c(i)$ de la manière suivante :

$$e_g(i) = e^{-\pi \left(\frac{c(i)}{1-c(i)} \right)^2}$$

Le problème de recherche du sous-ensemble S revient à déterminer le sous-ensemble minimisant la combinaison linéaire de l'erreur de probabilité e_p et l'erreur de colinéarité e_g :

$$e(i) = \alpha e_p(i) + (1 - \alpha)e_g(i) \quad (3.11)$$

En général, le nombre de correspondants L_i est petit (par exemple, $L_i = 3$). Il est possible alors de former l'ensemble E de toutes les combinaisons possibles définies sur $\{s_{i_1}, \dots, s_{i_{L_i}}\}$ ($2^{L_i} - 1$ possibilités). Nous choisissons l'élément de E minimisant e (eq. 3.11).

Généralisant maintenant cette démarche pour inclure tous les segments m_i . Pour cela, nous supposons que les m_i sont ordonnés (par exemple, selon un critère de longueur) et définissons la fonction $C(n)$ suivante qui exprime l'erreur totale de probabilité et de colinéarité de la solution à l'étape n :

$$C(n) = \sum_{i=1}^n e(i) \quad (3.12)$$

La recherche des correspondants de l'image I_c dans l'image I_m peut être effectuée par un algorithme de type A^* comme dans le cas de la fusion (cf. 3.2).

Examinons l'admissibilité de l'algorithme A^* en utilisant la fonction $C(n)$ donnée par l'équation 3.12. Rappelons que de tel algorithme ne peut garantir une solution optimale que si $C(n) - C(n-1) > 0$ et $C(n^*) \leq C(n)$ ($C(n^*)$ est le coût estimé et $C(n)$ le coût réel). Un examen de la fonction $C(n)$ montre que :

$$C(n) - C(n-1) = \alpha e_p(n) + (1 - \alpha)e_g(n)$$

Cette quantité est non négative car pour tout n , nous avons $e_p(n) \geq 0$ et $e_g(n) \geq 0$. Ainsi, la première condition n'est pas vérifiée. Montrons que $C(n)$ produit un algorithme admissible. Pour cela, il suffit de montrer que l'algorithme A^* donné en Annexe D et utilisant $C(n)$ se termine en fournissant le chemin optimal :

- cet algorithme se termine, puisque l'espace de recherche que nous utilisons est un arbre fini,
- il atteint l'étape finale N , car il existe dans tous les cas un chemin menant de l'étape 0 à l'étape N ,
- il a été montré [Nil 71] que si l'algorithme A^* se termine en atteignant l'étape finale et si $C(n^*) \leq C(n)$ alors il fournit le chemin optimal. Cette dernière condition est également vérifiée par notre fonction $C(n)$, car le coût estimé à l'aide de cette fonction est égal au coût réel. Il en résulte que l'utilisation de $C(n)$ (eq 3.12) produit un algorithme admissible.

3.5.5 Résumé de l'algorithme

L'algorithme de mise en correspondance a pour donnée deux images de segments I_m et I_c . Il fournit en résultat l'ensemble de quadruplets suivants :

$$E = \{(m_1, S_1^{p_1}, \vec{P}_1, e_1), \dots, (m_N, S_N^{p_N}, \vec{P}_N, e_N)\} \quad (3.13)$$

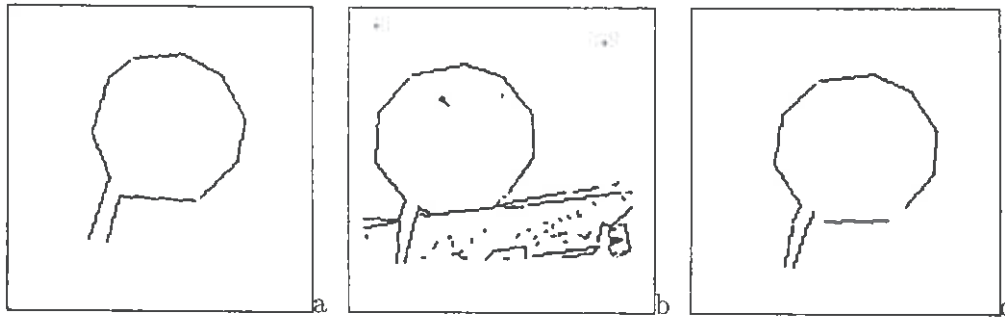


Figure 3.6. a) et b) deux images de contours, c) résultats de la mise en correspondance.

où m_i est un segment de I_m , $S_i^{p_i}$ un p_i -uplets de segments de I_c , \vec{P}_i le vecteur de probabilité de mise en correspondance et e_j l'erreur de la mise en correspondance.

Cet algorithme comporte quatre étapes :

- 1) construction du graphe de voisinage de l'image I_m en utilisant l'algorithme décrit en annexe C,
- 2) initialisation des probabilités et de la compatibilité en utilisant respectivement les équations 3.9 et 3.10,
- 3) estimation des probabilités en utilisant l'équation 3.8. Le critère d'arrêt étant : $|p^{(n+1)}(i, k) - p^{(n)}(i, k)| < \epsilon$. ϵ est la plus petite variation tolérée,
- 4) sélection pour chaque segment de I_m des meilleurs segments correspondants de I_c en utilisant la fonction $C(n)$ donnée par l'équation 3.12 et l'algorithme A^* donné en Annexe D.

Les résultats de cet algorithme sont décrits par la figure 3.6. Les deux images de contours 3.6.a et 3.6.b sont mis en correspondance. La figure 3.6.c présente le résultat de la mise en correspondance. Les paramètres de mise en correspondance sont : la contribution de chacun des attributs d'un segment de contours $\alpha_1 = \alpha_2 = 0.5$, la contribution de l'erreur de probabilité et de l'erreur géométrique ($\alpha = 0.5$), enfin la taille de la fenêtre nécessaire à la génération d'hypothèses ayant pour valeur 11.

3.6 Conclusion

Dans ce chapitre, nous avons présenté un algorithme pour analyser l'ensemble de couples (image de contours, détecteur) résultats de l'étape de sélection. Cette analyse a pour objectif d'évaluer la différence entre le contour exemple et le contour donné, de déterminer le contour exemple nécessaire à la sélection du prochain détecteur et enfin d'éliminer les détecteurs qui ont été sélectionnés par erreur. Notre contribution centrale dans ce chapitre est la conception et l'implantation de :

- un algorithme de type A^* pour combiner les informations contours résultats des détecteurs qui ont été choisis et exécutés, et pour corriger les erreurs de sélection. Cet algorithme a l'avantage d'être global et de fournir une solution efficace tout en assurant le compromis entre deux critères antagonistes que sont la détection et la localisation,
- un algorithme de mise en correspondance de segments de contours fondé sur la relaxation continue. Cet algorithme a la possibilité d'associer à un segment plusieurs correspondants "étiquetage mutiple", grâce à l'utilisation d'un algorithme de type A^* . Un autre avantage est le fait d'être globale, c'est-à-dire que cet algorithme fournit l'étiquetage minimisant l'erreur totale d'ambiguïté et maximisant la consistance globale. La relation de voisinage des segments construite et intégrée dans cet algorithme est fondée sur les diagrammes de Voronoï.

Par ailleurs, nous avons amélioré et implanté un algorithme global de fermeture de contours par relaxation continue. Cet algorithme utilise le paradigme prédiction/vérification d'hypothèses. La vérification d'hypothèses est fondée sur un critère utilisant des informations géométriques et photométriques pour fermer les contours.

Le problème principal du module d'analyse des résultats provient de l'algorithme de mise en correspondance de contours. Dans des images contenant des contours effilochés, les segments possèdent généralement une petite longueur. L'estimation de leurs attributs (e.g. orientation) est peu fiable. Leur mise en correspondance est alors difficile et souvent entachée d'erreurs. Pour ce faire, nous proposons d'améliorer cet algorithme par la considération des informations photométriques (e.g., la force du contour et le type de son modèle) lors de la mise en correspondance. Cette amélioration est détaillée dans le chapitre suivant.

3.7 Annexe A

Nous décrivons dans cette annexe les règles de calcul du pas de variation $Proj_i^{(n)}(\frac{\partial C}{\partial p(i,k)})$ et du sens de variation ρ_i^n (eq. 3.8) utilisés dans l'algorithme de relaxation continue présenté au paragraphe 3.5.2. La dérivée du critère C (eq. 3.7) est donnée par :

$$\frac{\partial C}{\partial p(i,k)} = q(i,k) + \sum_{j \in \text{voisin}(m_i)} \frac{1}{\sum_{l=1}^{L_j} Q(j,l)} \sum_{l=1}^{L_j} r(i,k,j,l)(p(j,l) - C_j)$$

L'opérateur de projection $Proj_i^{(n)}()$ peut être obtenu par :

$$a_i = Proj_i^{(n)}(\frac{\partial C}{\partial p(i,k)}) = \frac{\partial C}{\partial p(i,k)} - \frac{1}{\text{card}(K)} \sum_{l \in K} \frac{\partial C}{\partial p(i,l)}$$

où K est un ensemble d'indice et $\text{card}(K)$ est le cardinal de K . La définition de l'ensemble K est liée aux valeurs de probabilités $p(i,k)$ et la valeur a_i . Plus précisément :

- si aucun élément de \vec{P}_i n'est nul alors :

$$K = \{1, 2, \dots, L_i\}$$

- sinon, si $a_i \neq 0$ alors :

$$K = \{k, \text{ tel que } p^n(i, k) = 0\}$$

- sinon, on pose

$$K' = \{k, \text{ tel que } p^n(i, k) = 0\}$$

soit $l \in K$ tel que la projection du gradient sur le domaine défini par les équations suivantes possède la plus petite valeur négative :

$$\sum_{k=1}^{Li} p(i, k) = 1 \quad \text{et} \quad \forall k \in K' - \{l\} \quad p(i, k) = 0$$

S'il n'existe aucun l vérifiant une telle condition, alors $a_i = 0$, c'est-à-dire que la solution obtenue est maximale. Sinon on considère,

$$K = K' - \{l\}$$

Dans la prochaine itération, la contrainte $p(i, l) = 0$ doit être modifiée (il faut la rendre non saturée).

Le sens de la variation ρ_i dépend essentiellement de la projection du gradient et est donné par :

$$\rho_i^{(n)} = \begin{cases} \frac{1-p^{(n)}(i,k)}{Proj_i()} & \text{si } Proj_i() > 0 \\ \frac{p^{(n)}(i,k)}{Proj_i()} & \text{si } Proj_i() < 0 \end{cases}$$

3.8 Annexe B

Dans cette annexe, nous présentons les différents attributs de segments de contours intervenant dans l'algorithme de mise en correspondance décrit au paragraphe 3.5.

Nous considérons des attributs relatifs à la proximité et à la similarité de deux segments. Soient s_i et s_k deux segments appartenant ou non à la même image, nous définissons alors les attributs suivants :

- le facteur de distance : la distance entre deux segments peut être définie par la moyenne des quatre distances entre chaque extrémité d'un segment et la droite support de l'autre segment (voir figure 3.7).

Soit $dist(i, k)$ la distance euclidienne entre les deux segments (s_i, s_k). Le facteur de distance est calculé par :

$$d(i, k) = \min\left(1, \frac{dist(i, k)}{\text{largeur de la fenetre}}\right) \quad (3.14)$$

$d(i, k)$ prend ses valeurs dans l'intervalle $[0, 1]$. Il est proportionnel à $dist(i, k)$, c'est-à-dire qu'il est d'autant plus proche de 1 que la distance est plus grande.

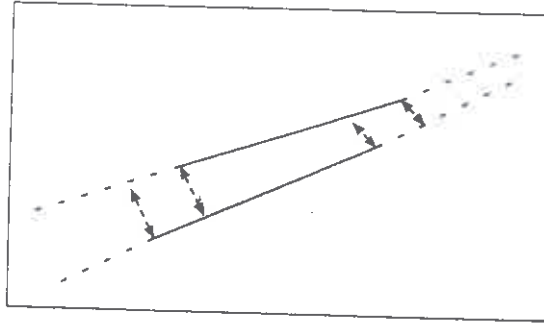


Figure 3.7. La distance entre deux segments.

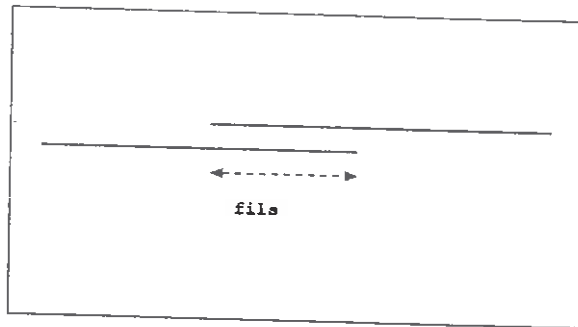


Figure 3.8. Recouvrement de deux segments.

- le recouvrement : on appelle *fils* de (s_i, s_k) , le morceau de segment commun à s_i et s_k après projection orthogonale de s_k sur s_i (voir figure 3.8).

Soient les deux coefficients :

$$l_f(i) = \frac{fils}{lg(s_i)} \quad \text{et} \quad l_f(k) = \frac{fils}{lg(s_k)}$$

où $lg(x)$ est la longueur de x .

Le facteur de recouvrement $r(i, k)$ est défini par :

$$r(i, k) = 1 - \max(l_f(i), l_f(k)) \quad (3.15)$$

$r(i, k)$ prend ses valeurs dans l'intervalle $[0, 1]$. Il est d'autant plus petit que le recouvrement entre les segments est grand.

- l'orientation : le facteur d'orientation est calculé comme suit :

$$o(i, k) = \frac{2\theta(i, k) \bmod \frac{\pi}{2}}{\pi} \quad (3.16)$$

où $\theta(i, k) \in [0, \pi]$ est l'angle intérieur entre les deux segments (voir figure 3.9). La valeur du facteur d'orientation est comprise entre 0 et 1. Elle est proportionnelle à l'angle $\theta(i, k)$.

- la colinéarité : la mesure de colinéarité entre deux segments s_i et s_k dépend de deux facteurs : la distance et l'orientation (voir figure 3.10).

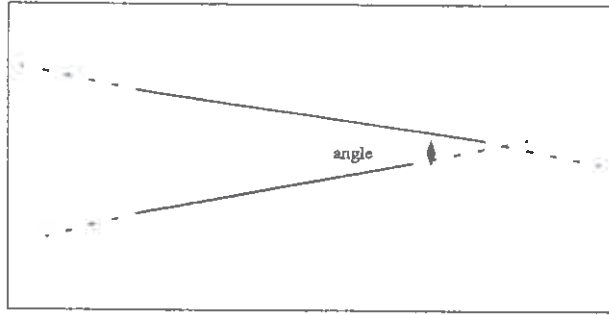


Figure 3.9. Angle intérieur entre deux segments.

— le facteur de distance peut être défini par :

$$md_1(i, k) = \frac{lg(s_i) + lg(s_k)}{lg(s_i) + lg(s_k) + dmin(i, k)} \quad (3.17)$$

avec $dmin(i, k)$ la distance entre les deux extrémités les plus proches. Ce coefficient prend ses valeurs dans l'intervalle $[0, 1]$. Il est proche de 1 si la distance minimale est petite (si $dmin(i, k) = 0$ alors $md_1(i, k) = 1$). Nous ajoutons un facteur de distance supplémentaire pour pénaliser tout recouvrement entre les deux segments.

$$md_2(i, k) = \frac{dmax(i, k)}{lg(s_i) + lg(s_k) + dmin(i, k)} \quad (3.18)$$

avec $dmax(i, k)$ la distance entre les extrémités les plus éloignées. Les valeurs possibles de $dmax(i, k)$ sont comprises entre 0 et 1. Si $dmin(i, k) = 0$ et $dmax(i, k) = lg(s_i) + lg(s_k)$ alors $md_2(i, k) = 1$.

— le facteur d'orientation est défini par :

$$mo(i, k) = \frac{\theta(i, k)}{\pi} \quad (3.19)$$

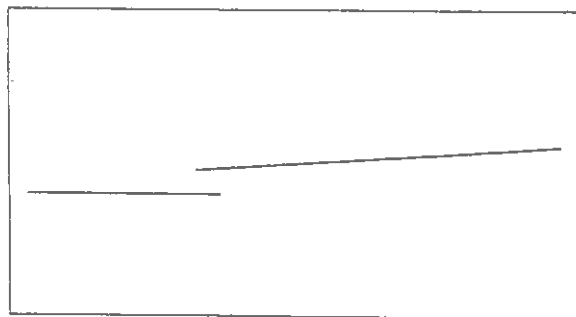


Figure 3.10. Colinéarité de deux segments.

$\theta(i, j)$ est l'angle intérieur entre les deux segments illustrée dans la figure 3.9. Ce coefficient prend aussi ses valeurs dans l'intervalle $[0, 1]$. Il est proche de 1 si les deux segments forment entre eux un angle π .

La mesure de colinéarité peut être obtenue à partir des équations 3.17, 3.18, 3.19.

$$cc(i, k) = md_1(i, k) \times md_2(i, k) \times mo(i, k) \quad (3.20)$$

Cet attribut prend ses valeurs entre 0 et 1. Il est proche de 1 (respectivement 0) si les deux segments sont colinéaires (resp. perpendiculaires).

La similarité et la proximité entre les deux segments m_i et $s_{i,k}$ sont combinées comme suit :

$$c(i, k) = \alpha_1 d(i, k) + \alpha_2 r(i, k) + (1 - \alpha_1 - \alpha_2) o(i, k), \quad \forall s_k \neq nil \quad (3.21)$$

Les attributs $d()$, $r()$, $o()$ sont donnés respectivement par les équations 3.14, 3.15 et 3.16. α_i exprime la contribution de la $i^{\text{ème}}$ caractéristique. $c(i, k)$ prend ses valeurs dans l'intervalle $[0, 1]$. Il est d'autant plus petit que les deux segments sont similaires et proches.

3.9 Annexe C

L'objectif de cette annexe est de présenter un algorithme de construction de la relation de voisinage entre des segments de droite. Cette relation est utilisée dans l'algorithme de mise en correspondance décrit au paragraphe 3.5.

La mise en correspondance de deux primitives image (par exemple, deux segments) ne dépend pas uniquement de la similarité et de la proximité de ces deux primitives, mais aussi de celles de leurs voisines. La définition des primitives voisines d'une primitive donnée est souvent une fonction d'un certain nombre de paramètres (distance, nombre de voisins, etc) [Fuk 72, O'C 75, Zuc 79]. Les performances du processus de recherche des voisins dépendent alors de la valeur des paramètres. Cette valeur est souvent fixée par expérimentation. Ahuja [Ahu 82] propose une définition du voisinage qui ne dépend que de la position des primitives et de leur répartition spatiale. Cette définition est donnée en terme de mosaïque² de Voronoï, son dual est le graphe de Delaunay.

Considérons dans un premier temps la définition de la relation de voisinage dans le cas où la primitive est un point. Nous montrerons ensuite comment étendre cette définition aux segments de droite.

La mosaïque de Voronoï d'un ensemble de points dans un plan est une partition de ce plan en régions. A chaque point m du plan est associée une région polygonale appelée le polygone de Voronoï. La région associée à un point m contient tous les points du plan qui sont plus proches du point m que de n'importe quel autre point. Deux points sont dits voisins si leurs polygones associés partagent un arc. La représentation duale de la mosaïque de Voronoï est le graphe de Delaunay qui est obtenu par toutes les paires de points qui sont voisins. La figure 3.11 montre un exemple de mosaïque de Voronoï.

Dans le cas des segments de droite, nous proposons de représenter un segment par ses deux extrémités. Deux segments sont dits voisins si et seulement s'ils possèdent au moins deux extrémités voisines au sens de Voronoï.

La construction du graphe de voisinage commence par la représentation de chaque segment par ses deux extrémités. Soient N le nombre de segments et n le nombre total

2. Traduction de tessellation

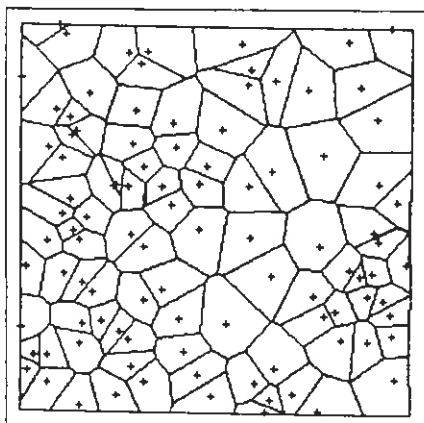


Figure 3.11. Un exemple de mosaïque de Voronoï où la primitive est le point.

d'extrémités. Nous avons $n \leq 2N$. En effet, si deux segments possèdent une extrémité commune alors ils ne sont représentés que par trois points.

Soit G un graphe où les nœuds sont les points et où les arcs indiquent la relation "voisin de". G est un graphe symétrique, car la relation "voisin de" l'est aussi. On suppose au départ que tous les points sont voisins. Formellement,

$$G(i, j) = \begin{cases} 1 & \forall i \neq j \\ 0 & \text{sinon} \end{cases}$$

1 indique que les deux points i et j sont voisins. La construction des mosaïques de Voronoï se fait de la façon suivante :

```

initialisation du polygone ← le plan image;
Pour  $i \leftarrow 1$  a  $n - 1$  faire
  Pour  $j \leftarrow i + 1$  a  $n$  faire
    debut
      si la bissectrice du segment formée par les deux points  $i$  et  $j \in$  polygone alors
        debut
          recalculer le polygone,
          marquer  $j$  comme voisin de  $i$ ,
          éliminer les points voisins de  $i$  dont la bissectrice ne fait,
            plus partie du nouveau polygone.
        fin
      fin
    fin
  fin

```

Cet algorithme a l'avantage d'être facile à mettre en œuvre et de s'exécuter dans tous les cas (c'est-à-dire, sans cas dégénéré). Sa complexité est égale à $\frac{n(n-1)}{2}$. Son coût est relativement élevé par rapport aux algorithmes existants [Ede 87, Boi 90]. A titre d'exemple, le coût de celui proposé dans [Ede 87] est de l'ordre de $n \log_2 n$. Notre objectif n'est pas la conception d'un algorithme efficace, mais plutôt d'un algorithme général. La figure 3.12.b décrit un exemple de graphe de Delaunay pour les segments de contours de l'image donnée par 3.12.a.

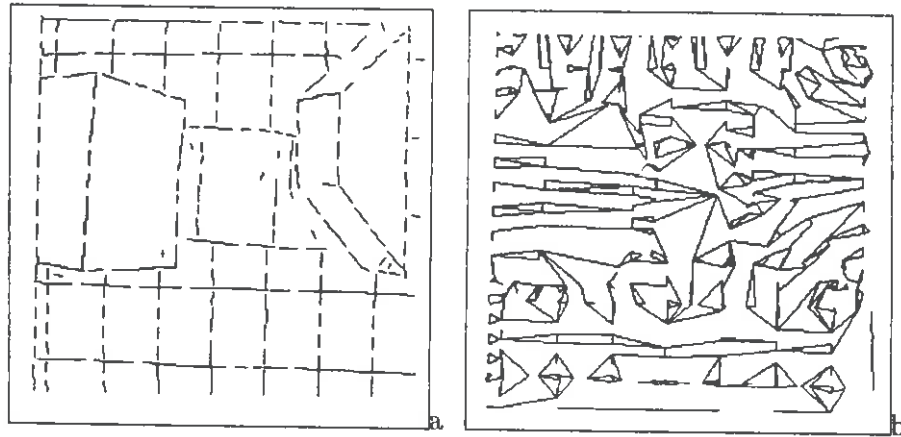


Figure 3.12. a) image de contours, b) le graphe de voisinage de segments de contours (graphe de Delaunay en considérant comme primitive le segment).

3.10 Annexe D

L'objectif de cette annexe est de présenter l'algorithme A^* utilisé pour la fusion et la mise en correspondance des contours. Nous donnons l'algorithme pour la fusion et montrons ensuite les modifications à lui apporter pour faire de la mise en correspondance. Soient *OUVERT* et *FERME* deux listes de segments. Les données de l'algorithme sont la fonction heuristique $C(n)$ (eq. 3.4) et l'espace de recherche ER ordonné et formé d'une liste de quadruplets définie comme suit :

$$ER = \{(m_1, S_1^{p1}, m_1 \oplus S_1^{p1}, S_1^{p1} \oplus m_1), \dots, (m_N, S_N^{pN}, m_N \oplus S_N^{pN}, S_N^{pN} \oplus m_N)\}$$

- 1) mettre un segment "fictif" dans la liste appelée *OUVERT*.
Poser la fonction cout $C(n) = 0$, avec $n = 0$,
- 2) sélectionner dans *OUVERT* un élément s , soit n le rang du quadruplet correspondant, tel que $C(n)$ soit minimale. Si plusieurs éléments sont candidats, en choisir un de façon arbitraire,
- 3) enlever de *OUVERT* l'élément s et le placer dans la liste *FERME*.
- 4) si $n = N$ alors fin de l'algorithme avec succès,
- 5) sinon, considérer les éléments du quadruplet de rang $n + 1$ dans l'espace *ER*.
Pour chacun des quatre éléments de ce nouveau quadruplet,
 - 5.1) évaluer la fonction cout $C(n + 1)$ donnée par l'équation 3.4.
 - 5.2) si l'élément actuel n'est ni dans la liste *OUVERT*, ni dans *FERME* alors l'ajouter à *OUVERT*. Lui lier un pointeur "arrière" vers son prédécesseur (dans le but de remonter une voie de solution une fois que le quadruplet final est traité),
aller à l'étape 2.

Cet algorithme peut être utilisé dans le cas de la mise en correspondance des contours en considérant la fonction coût donnée par l'équation 3.12. Dans ce cas, l'espace de recherche *ER* est composé de N éléments. Chaque élément est un $2^{L_i} - 1$ uplets correspondant à toutes les combinaisons possibles définies sur $\{s_{i_1}, \dots, s_{i_{L_i}}\}$.

4

Résultats expérimentaux et conclusion

Ce chapitre décrit une évaluation globale de l'approche de sélection de détecteurs et de leurs paramètres, permettant ainsi d'ouvrir les perspectives à venir.

4.1 Les règles de sélection

L'objectif de ce paragraphe est de présenter les règles de sélection utilisées. Ces règles sont déduites à partir des heuristiques présentées dans le chapitre 2. Rappelons que celles-ci sont en général peu précises, voire parfois contradictoires. A titre d'exemple, soient les deux heuristiques suivantes : la première indique qu'un détecteur laplacien est mieux adapté dans le cas de contours formés de courbes lisses et la seconde indique que, pour un contour bruité, il est préférable d'utiliser un opérateur du premier ordre. L'imprécision de ces deux heuristiques fait qu'en présence d'un contour bruité et formé de courbes lisses il est possible d'utiliser un laplacien ou un gradient. Or, si un laplacien est utilisé, il est nécessaire de réaliser un lissage fort. Il en résulte un contour lisse possédant une forte erreur de localisation. En revanche, l'utilisation d'un gradient conduit à un contour non nécessairement lisse, mieux localisé et moins bruité. D'une manière générale, le bruit est un élément nécessitant une attention particulière, voire des traitements appropriés. Ainsi, nous proposons de :

- considérer le bruit et la netteté d'un contour comme étant deux attributs dominants, en ce sens qu'il s'agit d'affecter un détecteur en favorisant ces deux attributs,
- associer à chaque contexte (ensemble de h-contours qui peut être déterminé par le même détecteur) plusieurs détecteurs. Rappelons que la sélection d'un détecteur fait introduire ses performances. Ainsi, si plusieurs détecteurs sont validés par les caractéristiques d'un contour, c'est le meilleur au sens des performances qui sera retenu.

Dans ce qui suit, nous proposons des règles de sélection. Nous distinguons les contours de type marche simple et marche double. Nous considérons les deux détecteurs DGD et DRF. La prise en compte d'autres détecteurs nécessite la maîtrise de leurs propriétés. Une fois celles-ci connues, les détecteurs peuvent être considérés en suivant la même démarche.

4.1.1 Cas des marches simples

Dans les règles définies ci-dessous, nous considérons les attributs de contours suivants : le rapport signal/bruit et la netteté. Nous proposons de discrétiser ces deux attributs de la façon suivante :

- le rapport signal/bruit : cet attribut est défini par le rapport du contraste sur l'écart-type du bruit et prend ses valeurs dans l'intervalle $[0, +\infty[$. Nous considérons les quatre cas suivants : inférieur à 2 (donc, 6 dB), appartenant à l'intervalle $[2, 4]$, appartenant à l'intervalle $[4, 6]$ et supérieur à 6 (donc, 16 dB),
- la netteté: elle prend ses valeurs dans l'intervalle $[0, 90]$. Nous considérons les quatre cas suivants : $[0, 22.5]$, $[22.5, 45]$, $[45, 67.5]$ et $[67, 90]$.

Avant de donner les règles de sélection dans le cas d'une marche simple, nous discutons brièvement les performances d'un détecteur.

Les performances

Rappelons que nous considérons comme attributs de performance d'un détecteur son temps de calcul, sa détection, son erreur de localisation et un score pour qualifier son exécution. A l'exception du temps de calcul, ces attributs de performance dépendent des caractéristiques du contour. Dans la réalisation actuelle, le score est initialisé à 1 pour tous les détecteurs. Il est alors omis dans les règles ci-dessous. La détection et l'erreur de localisation sont obtenues à partir des mesures données par les équations 2.55 et 2.56 et d'un modèle de contour de type marche brouillée définie par :

$$\tanh(\mu(x\cos(\theta) + y\sin(\theta)))$$

où μ représente la netteté du contour et θ son orientation. Le choix de ce modèle de contour est imposé par le fait que la dérivée première du filtre de lissage du détecteur DRF est discontinue à l'origine. Cette propriété rend sa manipulation mathématique parfois difficile, surtout dans le cas des modèles de contours discontinus.

Comme nous l'avons montré dans le paragraphe 2.4.3, les performances d'un détecteur DRF et DGD dépendent souvent de l'orientation du contour. Nous fournissons dans les règles ci-dessous les performances dans l'orientation 0, 22.5 et 45. Dans le cas des orientations non précisées dans les règles ci-dessous, les performances sont obtenues de la façon suivante :

- par interpolation, si l'orientation appartient à l'intervalle $[0, 45]$,
- par symétrie et interpolation si l'orientation n'appartient pas à l'intervalle $[0, 45]$.

Par ailleurs, le temps de calcul des détecteurs DRF et DGD pour traiter une image 256×256 sur un SUN4 est égale respectivement à 6 et 18 secondes.

Les règles

Règle 1 :

si le contour est de type marche simple et de rapport signal/bruit inférieur à 2 ou s'il possède une netteté inférieure à 22.5 alors utiliser le détecteur DGD avec le paramètre $\alpha = 0.75$

Cette règle décrit un contour bruité ou peu net. Dans ces deux cas, un détecteur utilisant un opérateur de différentiation du premier ordre semble être mieux adapté. C'est la raison pour laquelle nous proposons l'utilisation du détecteur DGD. En ce qui concerne le choix de son échelle, il est nécessaire de réaliser dans les deux cas un lissage très fort. Rappelons que ce détecteur n'est stable que lorsque α est strictement positif. En pratique, il devient peu stable lorsque α est inférieur à 0.7. Nous proposons alors $\alpha = 0.75$.

Dans les deux situations décrites par la précondition de cette règle, les performances du détecteur DGD sont données par :

- le rapport signal/bruit inférieur à 2 : (loc=0.375 det=0.749 ang=0), (loc=0.453 det=0.748 ang=22.5), (loc=0.476 det=0.746 ang=45),
- la netteté inférieure à 22.5 : (loc=0.368 det=0.962 ang=0), (loc=0.452 det=0.949 ang=22.5), (loc=0.476 det=0.941 ang=45).

Règle 2 :

si le contour est de type marche simple et de rapport signal/bruit dans l'intervalle [2,4] ou s'il possède une netteté dans l'intervalle [22.5, 45] alors utiliser le détecteur DGD avec $\alpha = 1$. ou le détecteur DRF avec $a = 0.25$

Cette règle décrit un contour plus net ou moins bruité que celui décrit par la règle 1. Dans ce cas, nous proposons l'utilisation des détecteurs DGD et DRF. Ce dernier utilise comme opérateur de différentiation un laplacien (donc, plus sensible au bruit). Il est préférable d'utiliser son échelle la plus élevée possible, soit $a = 0.25$.

Les performances du détecteur DGD avec $\alpha = 1$ sont les suivantes : (loc=0.106 det=0.994 ang=0), (loc=0.315 det=0.991 ang=22.5), (loc=0.404 det=0.988 ang=45).

Les performances du détecteur DRF avec $a = 0.25$ sont données par : (loc=0.075 det=0.999 ang=0), (loc=0.123 det=0.995 ang=22.5), (loc=0.139 det=0.992 ang=45).

Règle 3 :

si le contour est de type marche simple et de rapport signal/bruit supérieur à 4 et de netteté dans l'intervalle [45, 67.5] ou ce contour possède un rapport signal/bruit dans l'intervalle [4, 6] et de netteté supérieure à 45 alors utiliser le détecteur DGD avec $\alpha = 1.2$ ou le détecteur DRF avec $a = 0.3$

Dans ces deux cas, les performances du détecteur DGD sont données respectivement par : (loc=0.001 det=1.000 ang=0), (loc=0.129 det=1.000 ang=22.5), (loc=0.286 det=1.000 ang=45) et (loc=0.060 det=0.988 ang=0), (loc=0.265 det=0.984 ang=22.5), (loc=0.375 det=0.980 ang=45).

Les performances du détecteur DRF sont les suivantes :

(loc=0.010 det=1.000 ang=0), (loc=0.033 det=1.000 ang=22.5), (loc=0.055 det=1.000 ang=45) et (loc=0.040 det=0.999 ang=0), (loc=0.077 det=0.993 ang=22.5), (loc=0.090 det=0.989 ang=45).

Règle 4 :

si le contour est de type marche simple et de rapport signal/bruit supérieur à 6 et de netteté supérieure à 67.5 alors

utiliser le détecteur DGD avec $\alpha = 1.5$ ou le détecteur DRF avec $a = 0.35$

Cette règle décrit un contour non bruité et net. Dans ce cas, les deux détecteurs DRF et DGD peuvent être utilisés avec succès. En ce qui concerne le choix du paramètre α , un lissage très faible semble être meilleur. Nous proposons alors $\alpha = 1.5$ pour le DGD et $a = 0.35$ pour le DRF.

Les performances du détecteur DGD sont données par : (loc=0.000 det=1.000 ang=0), (loc=0.016 det=1.000 ang=22.5), (loc=0.142 det=1.000 ang=45).

Les performances du détecteur DRF sont données par : (loc=0.001 det=1.000 ang=0), (loc=0.015 det=1.000 ang=22.5), (loc=0.075 det=1.000 ang=45).

Remarque : dans le cas de contour rectiligne, on utilise seulement le détecteur DGD directionnel dans les mêmes conditions données par les règles ci-dessus. Les performances de ce détecteur ne dépendent pas de la direction du contour et sont les mêmes que ceux de DGD dans le cas de l'orientation égale à 0.

4.1.2 Cas des doubles marches

Les attributs d'une double marche sont le rapport signal/bruit et la largeur. La considération de ces attributs mène parfois à des contradictions. Par exemple, une double marche bruitée et de petite largeur nécessite une échelle élevée à cause du bruit et une échelle basse à cause de la petite largeur. Il est nécessaire alors de choisir l'échelle réalisant le compromis.

D'une manière générale, une double marche non bruitée nécessite souvent l'utilisation d'une petite échelle. Nous proposons alors l'heuristique suivante : *utiliser la plus petite échelle possible*. Le choix de cette échelle est fortement lié au rapport signal/bruit. C'est la raison pour laquelle la largeur du contour est omise des préconditions des règles ci-dessous.

Notons que le pas d'échantillonnage du rapport signal/bruit est identique à celui que nous avons utilisé dans le cas des marches simples et que les performances sont obtenues en suivant la même démarche et en considérant pour les mêmes raisons le modèle de contours suivant :

$$\operatorname{erf}((x \cos(\theta) + y \sin(\theta))/\mu) - \operatorname{erf}(((x - x_0) \cos(\theta) + (y - y_0) \sin(\theta))/\mu)$$

θ étant l'orientation du contour, $\sqrt{(x - x_0)^2 + (y - y_0)^2}$ la largeur de la double marche et μ un paramètre contrôlant la netteté de la double marche.

Règle 5

si le contour est de type marche double et de rapport signal/bruit inférieur à 2 alors

utiliser le détecteur DGD avec le paramètre $\alpha = 0.75$

Cette règle décrit un contour bruité de type double marche. La considération du

fait que ce contour est bruité implique l'utilisation du détecteur DGD. En revanche, la considération du fait que ce contour est de type double marche implique l'utilisation d'un détecteur DRF. Etant donné que le rapport signal/bruit est un élément dominant, nous retenons le détecteur DGD avec une échelle basse, soit $\alpha = 0.75$.

Les performances du détecteur DGD à cette échelle pour un tel contour sont données par : (loc=0.629 det=0.934 ang=0), (loc=0.823 det=0.918 ang=22.5), (loc=0.968 det=0.908 ang=45).

Règle 6

si le contour est une marche double et de rapport signal/bruit dans l'intervalle [2,4] alors utiliser le détecteur DGD avec $\alpha = 1$ ou le détecteur DRF avec $a = 0.25$

Les performances du détecteur DGD avec $\alpha = 1$ sont les suivantes : (loc=0.626 det=0.946 ang=0), (loc=0.810 det=0.934 ang=22.5), (loc=0.941 det=0.928 ang=45),

Les performances du détecteur DRF avec $a = 0.25$ sont les suivantes : (loc=0.518 det=0.882 ang=0), (loc=0.524 det=0.826 ang=22.5), (loc=0.526 det=0.809 ang=45).

Règle 7

si le contour est une marche double et de rapport signal/bruit dans l'intervalle [4,6] alors

utiliser le détecteur DRF avec $a = 0.3$ ou le détecteur DGD avec $\alpha = 1.2$

Les performances du détecteur DGD sont données par : (loc=0.585 det=0.994 ang=0), (loc=0.726 det=0.990 ang=22.5), (loc=0.880 det=0.987 ang=45),

Les performances du détecteur DRF sont données par : (loc=0.511 det=0.974 ang=0), (loc=0.515 det=0.937 ang=22.5), (loc=0.516 det=0.922 ang=45)

Règle 8

si le contour est une marche double et de rapport signal/bruit supérieur à 6 alors utiliser le détecteur DRF avec $a = 0.35$ ou le détecteur DGD avec $\alpha = 1.5$

Cette règle décrit un contour non bruité. Dans ce cas, les deux détecteurs DRF et DGD peuvent être utilisés avec succès. En ce qui concerne le choix du paramètre α , un lissage très faible semble être meilleur. Nous proposons alors $\alpha = 1.5$ pour le DGD et $a = 0.35$ pour le DRF.

Les performances des détecteurs DGD et DRF sont données respectivement par : (loc=0.584 det=0.998 ang=0), (loc=0.725 det=0.996 ang=22.5), (loc=0.885 det=0.995 ang=45) et (loc=0.510 det=0.993 ang=0), (loc=0.513 det=0.969 ang=22.5), (loc=0.514 det=0.958 ang=45).

4.2 Les résultats expérimentaux

Nous présentons ici une exécution de notre système sur l'exemple qui a été présenté dans le paragraphe 2.5 et qui est décrit par la figure 4.1. Rappelons que le contour exemple est peu net et possède un faible rapport signal/bruit, en particulier les deux segments numérotés 1 et 2.

Nous considérons comme caractéristique de la solution requise les suivantes : la contri-

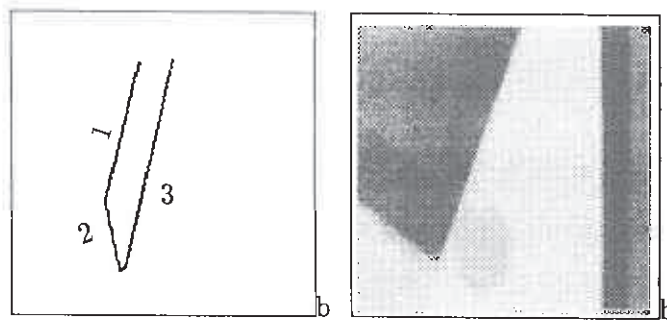


Figure 4.1. a) Contour exemple, b) image contenant ce contour.

bution de la détection est égale à 0.6 , les contributions de l'erreur de localisation et du temps de calcul sont identiques et sont égales à 0.2.

La figure 4.2 présente une exécution de notre système. La mise en évidence du contour donné est effectuée en trois itérations. Examinons dans ce qui suit chacune de ces itérations.

La première itération

Rappelons que la première étape effectuée par le système consiste à analyser le contour exemple : déterminer les h-contours qui le composent et leurs caractéristiques. Comme le montre l'exécution donnée par la figure 4.2, le résultat de cet étape étant deux h-contours de type marche simple :

- le premier h-contour est formé de trois segments, numérotés 1, 2 et 3 dans la figure 4.3.a. Ce h-contour est peu net (la netteté est égale à 41 degrés) et possède un faible rapport signal/bruit qui est égale à 0.589 (soit -4.6 dB),
- le second correspond au segment numéroté 4. Il est plus net et moins bruité que le précédent : sa netteté est égale à 76.7 et son rapport signal/bruit égale à 4.448 (soit 13 dB).

La sélection d'un détecteur est effectuée par l'affectation à chaque h-contour des détecteurs qui sont validés par leurs caractéristiques et le choix du meilleur au sens de performance. Le résultat de la sélection est donné par :

- le premier h-contour étant peu net et très bruité, les détecteurs (DRF, $\alpha = 0.25$), (DGD, $\alpha = 1$) et (DGD, $\alpha = 0.75$) (ce qui correspondent aux règles 2 et 1) sont validés par ses caractéristiques. En ce qui concerne l'affectation des détecteurs directionnels, seuls les détecteurs DGD prisent dans les directions -6 et 20 sont considérés (ce sont les orientations normales au contour). En effet, ce h-contour est composé de trois segments d'orientations 84, -70 et -19 degrés. Le segment d'orientation -19 possède une faible longueur (égale à $\sqrt{5}$). Il est considéré alors comme étant peu fiable est aucun détecteur ne lui est affecté,


```

Lecture des donnees
Generation de la base d'algorithmes
Generation du contour exemple

Execution numero 1

Analyse du contour exemple
+ Identification du contour et estimation de ses attributs
+ Segmentation des courbes
++ le nombre de courbes avant et apres la fusion est 49 et 2
Selection d'un detecteur
++ Caracteristiques de la courbe numero = 1
+++ forme = drto type = simple rzb = 0.589 nettete = 40.986
+++ liste des segments : [(54 4) (31 67)] [(31 67) (34 100)] [(35 100)
(37 99)]
+++ liste des detecteurs valides : (der4_d, -6, alpha=1.00) (der4_d, 20, alpha=1.00) (der4_d, 20, alpha=1.0)
(der1_ni, alpha=0.75)

++ Caracteristiques de la courbe numero = 2
+++ forme = drto type = simple rzb = 4.448 nettete = 76.713
+++ liste des segments : [(38 99) (69 5)]
+++ liste des detecteurs valides : (der6_d, 18, alpha=1.2) (shen6_ni, a=0.3) (der5_ni, alpha=1.2)

+++ Evaluation de la mesure de selection : (shen4_ni, 0.06163) (der4_ni, 0.02339) (der4_ni, 0.02339)
11599) : (der4_d, 20, 0.062228) (shen6_ni, 0.092967) (der4_d, -6, 0.129583) (der1_ni, alpha=0.75)
(der6_d, 18, 0.076925)

+++ L'algorithmme choisi est : shen6_ni
Execution du detecteur selectionne
Analyse des resultats
+ Fusion des contours
+ Generation du nouveau contour exemple
Evaluation du critere d'arret
++ Pourcentage du contour trouve = 0.554309

Execution numero 2

Analyse du contour exemple
+ Identification du contour et estimation de ses attributs
+ Segmentation des courbes
++ le nombre de courbes avant et apres la fusion est 18 et 1
Selection d'un detecteur
++ Caracteristiques de la courbe numero = 1
+++ forme = drto type = simple rzb = 0.589 nettete = 40.986
+++ liste des segments : [(54 4) (31 67)] [(31 67) (34 100)] [(35 100)
(37 99)]
+++ liste des detecteurs valides : (der4_d, -6, alpha=1.00) (der4_d, 20, alpha=1.00) (der4_d, 20, alpha=1.0)
(der1_ni, alpha=0.75)

+++ Evaluation de la mesure de selection : (shen4_ni, 0.023397) (der4_d, -6, 0.129583) (der4_ni, 0.06141)
d, -6, 0.272188) (der1_d, 20, 0.190485)

+++ L'algorithmme choisi est : shen4_ni
Execution du detecteur selectionne
Analyse des resultats
+ Fusion des contours
+ Generation du nouveau contour exemple
Evaluation du critere d'arret
++ Pourcentage du contour trouve = 0.934630

Liste des detecteurs resultats :
shen4_ni : a=0.3, Pourcentage : 0.582876 (109.110039 187.192688)
der4_ni : alpha=1.0, Pourcentage : 0.417125 (78.082657 187.192688)
Fermeture des contours

Execution numero 3

Analyse du contour exemple
+ Identification du contour et estimation de ses attributs
+ Segmentation des courbes
++ le nombre de courbes avant et apres la fusion est 18 et 1
Selection d'un detecteur
++ Caracteristiques de la courbe numero = 1
+++ forme = drto type = simple rzb = 0.589 nettete = 40.986
+++ liste des segments : [(54 4) (31 67)] [(31 67) (34 100)] [(35 100)
(37 99)]
+++ liste des detecteurs valides : (der4_d, -6, alpha=1.00) (der4_d, 20, alpha=1.00) (der4_d, 20, alpha=1.0)
(der1_ni, alpha=0.75)

+++ Evaluation de la mesure de selection : (shen4_ni, 0.023397) (der4_d, -6, 0.129583) (der4_ni, 0.06141)
d, -6, 0.272188) (der1_d, 20, 0.190485)

+++ L'algorithmme choisi est : shen4_ni
Execution du detecteur selectionne
Analyse des resultats
+ Fusion des contours
+ Generation du nouveau contour exemple
Evaluation du critere d'arret
++ Pourcentage du contour trouve = 0.554309

Execution numero 3

```

```

Analyse du contour exemple
+ Identification du contour et estimation de ses attributs
+ Segmentation des courbes
++ le nombre de courbes avant et apres la fusion est 18 et 1
Selection d'un detecteur
++ Caracteristiques de la courbe numero = 1
+++ forme = drto type = simple rzb = 0.589 nettete = 40.986
+++ liste des segments : [(54 4) (31 67)] [(31 67) (34 100)] [(35 100)
(37 99)]
+++ liste des detecteurs valides : (der4_d, -6, alpha=1.00) (der4_d, 20, alpha=1.00) (der4_d, 20, alpha=1.0)
(der1_ni, alpha=0.75)

+++ Evaluation de la mesure de selection : (shen4_ni, 0.023397) (der4_d, -6, 0.129583) (der4_ni, 0.06141)
d, -6, 0.272188) (der1_d, 20, 0.190485)

+++ L'algorithmme choisi est : der4_ni
Execution du detecteur selectionne
Analyse des resultats
+ Fusion des contours
+ Generation du nouveau contour exemple
Evaluation du critere d'arret
++ Pourcentage du contour trouve = 0.934630

Liste des detecteurs resultats :
shen4_ni : a=0.3, Pourcentage : 0.582876 (109.110039 187.192688)
der4_ni : alpha=1.0, Pourcentage : 0.417125 (78.082657 187.192688)
Fermeture des contours

Execution numero 3

Analyse du contour exemple
+ Identification du contour et estimation de ses attributs
+ Segmentation des courbes
++ le nombre de courbes avant et apres la fusion est 18 et 1
Selection d'un detecteur
++ Caracteristiques de la courbe numero = 1
+++ forme = drto type = simple rzb = 0.589 nettete = 40.986
+++ liste des segments : [(54 4) (31 67)] [(31 67) (34 100)] [(35 100)
(37 99)]
+++ liste des detecteurs valides : (der4_d, -6, alpha=1.00) (der4_d, 20, alpha=1.00) (der4_d, 20, alpha=1.0)
(der1_ni, alpha=0.75)

+++ Evaluation de la mesure de selection : (shen4_ni, 0.023397) (der4_d, -6, 0.129583) (der4_ni, 0.06141)
d, -6, 0.272188) (der1_d, 20, 0.190485)

+++ L'algorithmme choisi est : der4_ni
Execution du detecteur selectionne
Analyse des resultats
+ Fusion des contours
+ Generation du nouveau contour exemple
Evaluation du critere d'arret
++ Pourcentage du contour trouve = 0.934630

Liste des detecteurs resultats :
shen4_ni : a=0.3, Pourcentage : 0.582876 (109.110039 187.192688)
der4_ni : alpha=1.0, Pourcentage : 0.417125 (78.082657 187.192688)
Fermeture des contours

```

Figure 4.2. La réponse de notre système au contour donné par la figure 4.1.

- le second étant net et peu bruité, les deux détecteurs (DRF, $a = 0.3$) et (DGD, $\alpha = 1.2$) (donc, règle 3) sont validés par ces caractéristiques. Ce h-contour étant rectiligne et n'est formé que d'un seul segment d'orientation -72 degrés, alors le détecteur (DGD, $\alpha = 1.2$) pris dans la direction 18 est aussi validé.

Parmi tous ces détecteurs candidats, celui possédant les meilleurs performances au sens du critère défini dans le paragraphe 2.3 est (DRF, $a = 0.3$). L'exécution de ce détecteur fournit les contours donnés dans la figure 4.3.b.

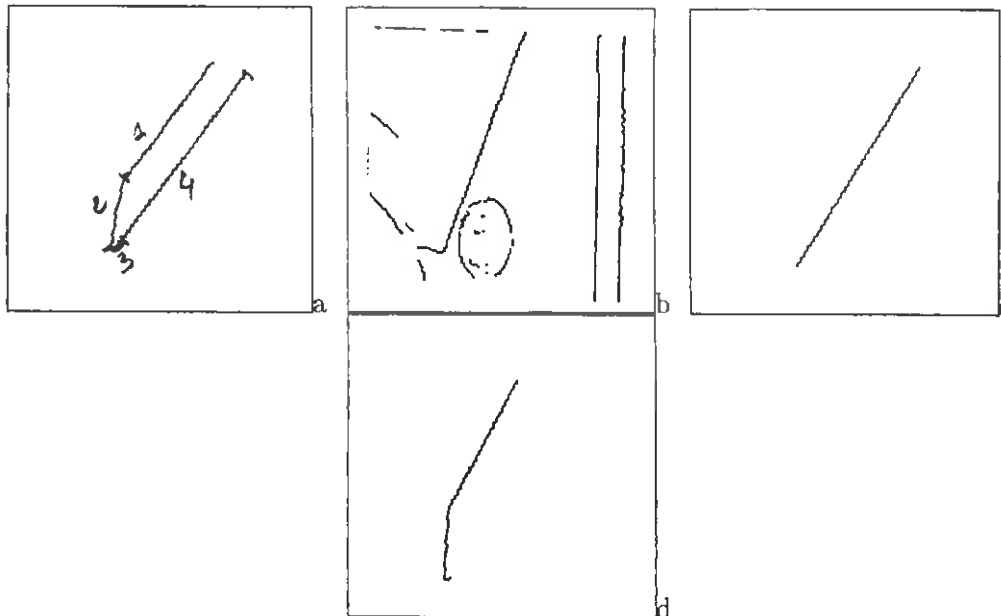


Figure 4.3. a) Les h-contours résultats de l'étape d'analyse du contour exemple, b) contour résultat du détecteur DRF avec $a = 0.3$, c) résultat de la mise en correspondance du contour résultat et du contour exemple, d) le nouveau contour exemple.

Il s'agit maintenant d'analyser les contours résultats du détecteur (DRF, $a = 0.3$). La figure 4.3.c décrit le résultat de l'étape d'élimination des contours ne correspondant pas à l'exemple. L'évaluation de la mesure de différence entre le contour calculé et celui donné (cf. 3.3) fournit 0.4457 (environ 44.6% du contour exemple n'est pas encore trouvé). La figure 4.3.d donne le contour qui sera considéré comme étant l'exemple dans l'itération suivante.

La deuxième itération

Dans cette itération le contour exemple considéré par le système est celui donné par la figure 4.3.d. Le résultat de l'étape d'analyse de ce contour montre qu'il correspond à un seul h-contour de type marche simple et est formé des segments numérotés 1, 2 et 3 dans la figure 4.3.a. Ce h-contour est le même que l'un des deux h-contours trouvé dans l'itération précédente. Le détecteur retenu est (DRF, $a = 0.25$). La figure 4.4.a présente les contours obtenus par ce détecteur. la figure 4.4.b décrit le résultat de l'étape d'élimination des contours ne correspondant pas à l'exemple.

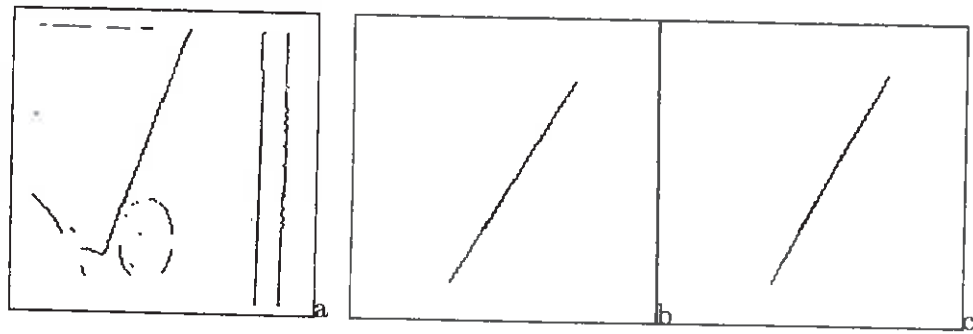


Figure 4.4. a) Contour résultat du détecteur DRF avec $a = 0.25$, b) résultat de la mise en correspondance du contour résultat et du contour exemple, c) résultat de la fusion de cette dernière image (fig. 4.4.b) et celle par la figure 4.3.c.

La figure 4.4.c décrit le résultat de la fusion des contours du détecteur (DRF, $a = 3$) donné par la figure 4.3.c et ceux du détecteur (DRF, $a = 0.25$) donnés par la figure 4.4.b. Notons que le contour résultat de la fusion est celui issu du détecteur (DRF, $a = 0.3$). En effet, les deux contours résultats de (DRF, $a = 0.3$) et de (DRF, $a = 0.25$) possèdent une longueur comparable et le détecteur (DRF, $a = 0.3$) possède de meilleures performances.

Cette itération n'a rien ajouté à la solution construite dans l'itération précédente : le détecteur (DRF, $a = 0.25$) a été sélectionné et exécuté, cependant le contour correspondant à l'exemple qui a fourni a été détecté par (DRF, $a = 0.3$).

La troisième itération

Dans cette itération le contour exemple considéré par le système est celui donné par la figure 4.3.d. Le résultat de l'étape d'analyse de ce contour est le même que celui de l'itération précédente. Le détecteur retenu est (DGD, $\alpha = 1$). Les résultats d'exécution de ce détecteur sont donnés dans la figure 4.5.a.

La figure 4.5.b décrit le résultat de l'étape d'élimination des contours ne correspondant pas à l'exemple. La figure 4.5.c présente les résultats de la fusion des contours résultat de la dernière fusion donnés par la figure 4.4.c et ceux du détecteur (DGD, $\alpha = 1$) donnés par la figure 4.5.b. Notons que le contour résultat est issu à la fois du détecteur (DRF, $a = 0.3$) et du détecteur (DGD, $\alpha = 1$). En effet, le contour numéroté 4 dans la figure 4.3.a est fourni par ces deux détecteurs et le détecteur (DRF, $a = 0.3$) possède de meilleures performances.

Le résultat de la mesure de la différence entre le contour calculé (fig. 4.5.c) et celui donné par l'utilisateur est égale à 0.065. A ce stade, l'objectif est considéré atteint. La contribution du détecteur dans la solution finale est de (DRF, $a = 0.3$) est d'environ 58.3% et celle du détecteur (DGD, $\alpha = 1$) est de 41.7%.

La figure 4.5.d présente le résultat de la fermeture du contour. Rappelons que l'algorithme de fermeture de contour implanté ne tient pas compte de l'information photométrique. Ceci explique le fait que l'un des trois "trous" n'a pas été fermé.

En conclusion, un examen des résultats obtenus montre qu'il est possible de mettre en évidence le contour donné par l'utilisateur en considérant seulement le détecteur (DGD,

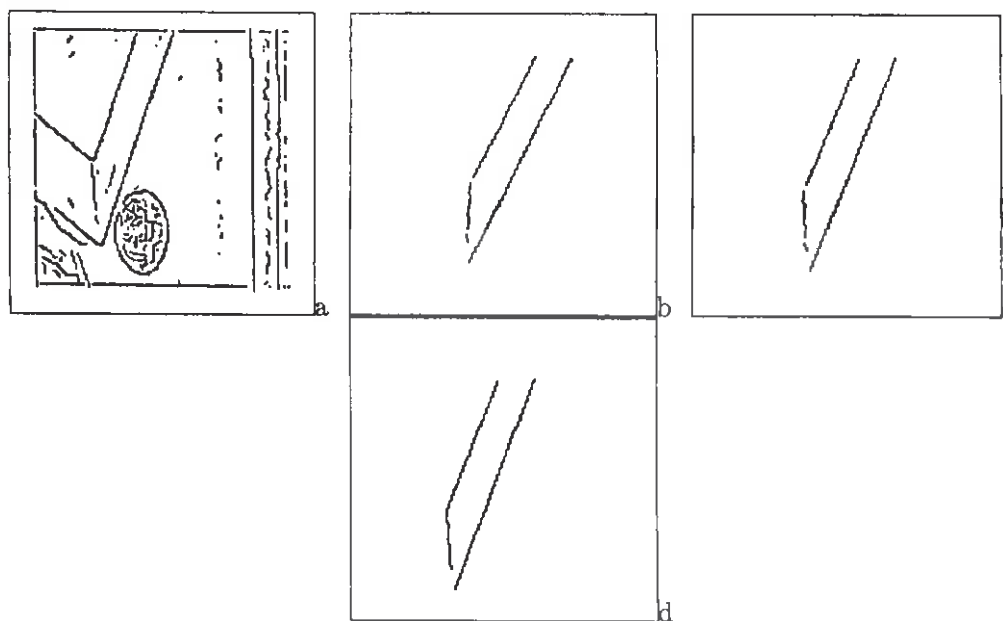


Figure 4.5. a) Contour résultat du détecteur DGD avec $\alpha = 1$, b) résultat de la mise en correspondance du contour résultat et du contour exemple, c) résultat de la fusion des contours donnés par les figures 4.4.c et 4.5.b, d) résultat de la fermeture des contours de l'image 4.5.c.

$\alpha = 1$). Cependant, l'utilisation à la fois des détecteurs (DRF, $a = 0.3$) et (DGD, $\alpha = 1$) conduit à un contour possédant une meilleure localisation et peut être calculé d'une manière efficace. Ces résultats montrent aussi la faisabilité de la sélection automatique de détecteurs bien que des erreurs de sélections sont possibles. Ces erreurs sont dues en générale à une mauvaise préconditionnement des détecteurs. Ce point sera détaillé plus loin.

4.3 La sélection de détecteurs

4.3.1 Le bilan

Nous avons proposé dans cette troisième partie une approche incrémentale pour la sélection des détecteurs de contours et leurs paramètres pour mettre en évidence un contour donné. Cette approche consiste à sélectionner un détecteur, qui est appliqué sur l'image, puis ses résultats sont ajoutés à ceux des détecteurs précédemment sélectionnés. A chaque étape, une évaluation des résultats obtenus est effectuée pour décider de la prochaine action à entreprendre.

Dans ce cadre, nous avons proposé et implanté un système comportant une bibliothèque contenant des outils de détection de contours (principalement, le détecteur DRF de Shen et Castan, celui de Deriche, celui de Deriche paramétré par l'orientation et le seuillage hystérésis) et deux modules ayant pour objectif :

1. la description du contour exemple : ce module extrait les caractéristiques du contour référence (son type, sa forme, son rapport signal/bruit, sa netteté et sa largeur) à l'aide d'algorithmes originaux que nous avons développés. Une fois les caractéristiques extraites, le contour référence est segmenté en h-contours.
2. la sélection d'un détecteur : le critère de sélection est fondé sur les caractéristiques précédemment définies, sur les propriétés des détecteurs disponibles (leur détection, leur localisation, leur complexité) ainsi que sur les qualités souhaitées du contour résultat (correctement localisé, détecté efficacement, etc).

Un détecteur de contours est considéré comme étant un triplet (algorithme, valeur de ses paramètres, précondition limitant son domaine d'application). La précondition est une disjonction de situations qui sont elles-mêmes des conjonctions de prédicats portant sur les caractéristiques du contour. A chaque situation sont associées les performances du détecteur pour qualifier son taux de réussite.

A chaque h-contour sont affectés les détecteurs qui sont susceptibles de le mettre en évidence. Ensuite, le meilleur détecteur au sens du critère précédemment évoqué est retenu.

La conception d'un tel critère de sélection nous a conduit à effectuer une étude bibliographique approfondie pour rassembler le maximum d'expertise, une étude en collaboration avec S. Tabbone sur le comportement des contours dans un espace échelle et une étude de l'influence des attributs d'un contour sur les performances d'un détecteur.

3. l'analyse des résultats : elle consiste à mesurer la différence entre le contour résultat et le contour donné par l'utilisateur, corriger les éventuelles erreurs de sélection en se fondant sur les résultats effectifs de ces détecteurs et générer le nouveau contour exemple.

Si l'objectif est atteint (à l'exception de quelques "trous") alors un module de fermeture de contours est exécuté. Dans le cas contraire, la sélection d'un autre détecteur est effectuée en considérant comme contour exemple celui qui vient d'être généré.

Pour diminuer le nombre de comparaisons entre les différentes images de contours obtenues et le contour donné par l'utilisateur, nous avons proposé de procéder à une fusion des contours résultats. La fusion est l'opération de fabrication d'une image de contours à partir d'autres images de même type. Le problème est posé de la façon suivante : étant données deux images de contours, il s'agit de construire une image par fusion de segments similaires de manière à minimiser l'erreur totale de fusion. Cette erreur est calculée à partir des propriétés des détecteurs déjà utilisés ainsi qu'à partir d'autres informations géométriques, en particulier la colinéarité. L'algorithme est de type A^* et fondé sur la mise en correspondance de segments par relaxation continue.

L'analyse des résultats peut être caractérisée par la définition d'un critère de fusion des contours qui est général : il ne se limite pas aux détecteurs stables dans un espace échelle et par l'algorithme de mise en correspondance qui est global : il est posé comme un problème de recherche de l'étiquetage minimisant l'erreur globale de mise en correspondance.

L'approche de sélection proposée possède deux points importants en sa faveur :

- grâce aux critères de sélection et de fusion définis, elle fournit une solution efficace en assurant le compromis entre la détection et la localisation qui sont en général antagonistes,
- grâce à l'utilisation de différentes sources d'information, notre approche est incrémentale et non combinatoire contrairement à celle adaptée dans le système IMPRESS. L'étape de sélection d'un détecteur utilise conjointement les propriétés des détecteurs et les caractéristiques du contour. L'étape d'analyse utilise les deux informations précédentes plus les contours résultats de l'exécution des détecteurs sélectionnés.

Enfin, notons que ce travail est loin d'être achevé. Le système proposé n'a été que peu expérimenté et les améliorations qui restent à apporter sont multiples. Elles sont données plus loin.

4.3.2 Comparaison avec le système IMPRESS

Pour se convaincre des avantages de notre approche, nous proposons de la comparer avec les travaux de Hasegawa et al [Has 86]. Rappelons à ce sujet que se sont les seuls travaux relatifs à la sélection de détecteurs de contours dont nous ayons connaissance. Dans leur système baptisé IMPRESS (cf. 4.3.1 de la partie 1), un détecteur de contours est considéré comme étant composé de trois modules (module de lissage et différentiation, module de binarisation et module de squelettisation et de chaînage). La sélection d'un détecteur est effectuée en exécutant tous les modules de lissage et différentiation et en retenant celui qui fournit le meilleur résultat, puis tous les modules de binarisation, etc. Contrairement à l'approche que nous proposons, IMPRESS a les caractéristiques suivantes :

- il est conçu pour sélectionner un seul détecteur de contours de type marche. Ce système n'effectue aucun retour arrière, c'est-à-dire qu'il se "contente" du résultat de la première exécution ;
- la sélection d'un détecteur est fondée uniquement sur le résultat de chacun des modules. Aucune autre source d'information n'est utilisée. Ce qui l'oblige à rechercher toutes les combinaisons possibles ;
- l'évaluation des résultats est effectuée pour chacun des modules d'un détecteur. Cette approche est difficile à mettre en œuvre puisqu'elle nécessite la fabrication à partir du contour donné du résultat de chacun de ces modules. Cependant, elle semble être intéressante car elle permet de fabriquer un nouveau détecteur composé de modules issus de détecteurs différents.

4.4 Les améliorations

Les limites de notre système sont dues essentiellement à :

- l'étape de sélection : signalons à ce niveau les trois problèmes suivants :

- l'algorithme responsable de l'identification des types du contour exemple suppose la connaissance de la position exacte de ce contour. En général, lors de la saisie de ce contour, qui est désigné par un utilisateur, des erreurs dans sa position sont introduites. Il en résulte alors des erreurs dans son identification,
- nous avons signalé plusieurs fois que la détermination des caractéristiques pertinentes d'un contour et la définition précise de la relation détecteur-contour n'est pas facile à réaliser dans l'état actuel de nos connaissances,
- l'estimation des performances d'un détecteur est effectuée en utilisant des critères 2D issus de ceux de Canny. Ces critères sont théoriques, ils reposent sur des hypothèses simplificatrices. L'estimation effectuée n'est alors qu'approximative.

En conséquence, la sélection de détecteurs ne peut être qu'entachée d'erreurs. Plus précisément, les deux situations suivantes peuvent se produire : un détecteur est sélectionné pour produire un contour et, une fois exécuté, il ne fournit pas ce contour ; ou ce détecteur fournit le contour attendu, mais, lors de la fusion, ce contour est complètement éliminé,

- l'élimination des faux points de contours : ce problème a été ignoré dans ce travail. Dans la réalisation actuelle, nous utilisons l'algorithme d'hystérésis avec le seuil haut égal à trois fois le seuil bas. Ce dernier est fixé pour un détecteur donné. Il ne dépend ni de l'échelle du détecteur ni des caractéristiques du contour. Il en résulte des images contenant parfois des contours troués et parfois trop bruités,
- la mise en correspondance : l'algorithme proposé utilise comme primitive le segment et repose essentiellement sur des informations de type géométrique (e.g., orientation du segment, sa longueur, la proximité et le voisinage). Son utilisation se limite en général à des images ne contenant pas des contours denses.

Nous discutons dans ce qui suit les solutions à ces différents problèmes.

4.4.1 Analyse du contour exemple

L'amélioration apportée au module de l'analyse du contour concerne essentiellement l'organisation des données. Il s'agit de concevoir un module permettant une meilleure identification des types du contour exemple et qui soit peu sensible à la position de ce contour. Ce module doit avoir en données le contour exemple, l'image qui le contient et l'ensemble des classes possibles et fournir une représentation hiérarchique de ce contour en termes de classes. Chaque classe est un ensemble de points appartenant au même type de contours et possédant des sous-classes correspondant à une description plus fine de ces points de contours. Pour ce faire, les techniques de la classification numérique semblent être adaptées. A ce sujet on pourra consulter les travaux [Vog 79, Dec 89].

Le module d'identification doit aussi tirer partie des résultats des détecteurs qui ont été sélectionnés et exécutés. En particulier, le comportement du contour dans un espace échelle. Une telle étude est en cours de développement actuellement en collaboration avec S. Tabbone dans le cas d'un seul détecteur exécuté à différentes échelles. Cependant, sa généralisation pour inclure d'autres détecteurs pose les deux problèmes suivants :

- L'étude du comportement d'un contour dans un espace échelle nécessite l'utilisation de détecteurs stables. Or, la définition de la liste des détecteurs vérifiant cette

propriété reste un problème ouvert. Yuille, Babaud et al. [Yui 86, Bab 86] ont montré que seul le laplacien d'une gaussienne est stable dans un espace échelle. Torre [Tor 86] signale (sans fournir une démonstration) qu'un détecteur effectuant d'abord une différentiation à l'aide de la deuxième dérivée directionnelle (donc, le gradient), puis un lissage gaussien est stable aussi. Enfin, Wu et al. [Wu 90] affirment (sans le montrer) que les filtres populaires de lissage $f(x) = e^{-\alpha|x|}$ et $f(x) = 1, |x| < 1$ et 0 sinon sont stables. Cependant, rien n'est dit quant à leur association avec un opérateur de différentiation.

- Dans le cas des détecteurs différents, il reste à établir une relation d'équivalence entre les échelles de ces détecteurs. En d'autres termes, étant donné un contour et l'échelle d'un détecteur, il s'agit de définir quelle est l'échelle d'un autre détecteur tel que les deux détecteurs fournissent des réponses possédant des propriétés similaires (un rapport signal/bruit comparable, une erreur de localisation comparable, etc.). Cette relation permettra de mener des raisonnements dans des espaces échelles appartenant à des détecteurs différents.

Enfin, rappelons que l'estimation des attributs du contour (la netteté, le contraste, la largeur, etc.) repose sur les techniques d'ajustement de l'image par des polynômes de Tchebycheff d'ordre trois ou quatre selon le type du contour. Nous proposons de renforcer ce module par l'utilisation de plusieurs bases de polynômes différentes et par le choix de la base qui ajuste au mieux le contour. On pourra consulter à ce sujet les travaux de [Nal 86].

4.4.2 Acquisition de règles de sélection

Pour remédier au problème de la définition de la relation de cause à effet entre les propriétés d'un détecteur, ses performances et les caractéristiques du contour, nous proposons de doter ce système d'un module d'apprentissage produisant un système évolutif qui tire profit de son expérience pour améliorer ses performances. L'objectif de ce module est double :

- la maintenance d'un historique sur les exécutions de chaque détecteur. Pour ce faire, l'approche connue sous la dénomination d'apprentissage par récompense et punition [Lan 83] (*"credit and blame"*) semble être adaptée,
- l'analyse et la correction des erreurs commises par le système à l'aide d'une évaluation des différents résultats. Les principales erreurs qui peuvent être commises sont celles dues à un mauvais préconditionnement des détecteurs : définition incomplète des caractéristiques d'un contour, mauvaise définition de la relation détecteur-contour, estimation incorrecte des attributs du contour.

La correction des différentes erreurs peut être effectuée en ajoutant un nouvel attribut à l'ensemble des attributs d'un contour, en supprimant un attribut, ou en ajustant le score d'un détecteur.

Pour aboutir à des mesures de performance plus précises, nous proposons d'intégrer un module d'évaluation quantitative dans ce système. Plus précisément, étant donné un ensemble de détecteurs validés par les caractéristiques d'un contour exemple, il s'agit de fabriquer un contour possédant les mêmes caractéristiques que ce contour exemple et

de procéder ensuite à une évaluation quantitative en utilisant les mesures de performance définies au paragraphe 2.4.2. A ce sujet, notons que la combinaison des approches d'analyse et de synthèse d'images est actuellement en cours d'expansion [Has 86, Gag 89].

4.4.3 Seuillage automatique

Le problème de l'élimination automatique des faux points de contours a été longuement discuté au paragraphe 4.5 de la partie 1. Nous proposons de doter notre système de l'algorithme que nous avons développé avec S. Tabbone et qui a été présenté brièvement au paragraphe 4.5 de la partie 1. Rappelons que cet algorithme procède par propagation du seuil dans un espace échelle. Il tient compte des origines des faux points de contours en associant à chacune des classes de faux points une règle d'élimination appropriée.

Notons que dans cet algorithme le seuil pour l'élimination des faux points de contours engendrés par le bruit est initialement fixé par expérience. Nous proposons la règle suivante de calcul du seuil :

$$T = n_0 \sqrt{\int_{-\infty}^{+\infty} f^2(x) dx}$$

où $f(x)$ est la réponse impulsionnelle du détecteur et n_0 l'énergie du bruit. Cette dernière peut être estimée en utilisant l'algorithme proposé dans le paragraphe 2.2.1.

La justification de cette règle est donnée comme suit : soit $S(x)$ un signal défini par :

$$S(x) = F(x) + N(x)$$

où $N(x)$ est un bruit additif blanc de moyenne nulle et de variance n_0 . La variance de la réponse de $f(x)$ au signal $S(x)$ est égale à :

$$v(x) = n_0^2 \int_{-\infty}^{+\infty} f^2(x) dx$$

L'amplitude maximale due à la composante bruit ne dépasse pas la valeur $\sqrt{v(x)}$. Tout point de contour ayant une plausibilité inférieure à $\sqrt{v(x)}$ peut être considéré comme étant dû au bruit. La mesure de plausibilité peut être le gradient ou l'amplitude du passage par zéro selon que le détecteur utilise un opérateur de différentiation du premier ou second ordre.

4.4.4 La mise en correspondance

Nous avons montré auparavant (cf. 3.5) que les performances de l'algorithme de mise en correspondance proposé se détériorent rapidement en présence d'images contenant des contours denses (e.g. image de scène d'extérieur) ou d'images contenant des contours tordus et discontinus. Pour pallier au premier inconvénient, nous proposons de renforcer cet algorithme en considérant à la fois des informations géométriques et structurelles (c'est-à-dire celles relatives à la structure du contour). Plus précisément, les contours sont mis en correspondance si et seulement si le critère défini dans le paragraphe 3.5 est vérifié et s'ils sont issus d'un même phénomène physique. L'identification du phénomène associé

à un contour peut être effectuée en se fondant sur les résultats du module d'analyse du contour : le comportement du contour, son type, ses attributs, etc.

Pour pallier le second problème, nous proposons de réaliser la mise en correspondance des contours qui n'ont pas été appariés par l'algorithme décrit ci-dessus en se fondant sur le point. Il existe plusieurs algorithmes pour ce faire [She 86, Zio 91b].

4.5 Les intérêts du système proposé

Les applications d'un tel système sont multiples. Citons :

- L'intégration dans un système de vision par ordinateur : le système de détection de contours que nous proposons peut être intégré dans un système de vision par ordinateur, qui pour compléter une description, ou pour "prendre" une décision, revient sur la phase de détection de contours avec un objectif bien précis de type chercher un contour ayant telles caractéristiques (ou seulement vérifier son existence) dans telle zone de l'image, calculer les caractéristiques de tel contour, etc.
- Le pilotage d'une bibliothèque de détection de contours : il nous arrive souvent d'être confronté aux questions de type : j'ai une image dont les contours possèdent telles propriétés, pouvez-vous nous dire quel détecteur utiliser ? Pour répondre à de telles demandes, on peut imaginer une bibliothèque de détecteurs de contours (par exemple, une bibliothèque similaire à celle proposée par Noesis [VIS 88]), dotée d'un module d'aide à la sélection des détecteurs de contours et de leurs paramètres. Le système décrit dans cette partie de la thèse peut répondre à ce besoin.
- La détection de contour par apprentissage : il arrive de traiter dans une application de la vision par ordinateur une image contenant plusieurs fois le même objet ou plusieurs images contenant les mêmes objets (à des changements de conditions de prise de vue près). En supposant que ces objets engendrent des contours similaires, il est possible alors d'envisager la démarche suivante pour la détection de leurs contours :
 - la désignation d'un contour exemple de cette image (éventuellement un contre-exemple) et la recherche des détecteurs et de leurs paramètres qui lui sont appropriés,
 - la recherche de tous les contours similaires au contour donné, dans la même image ou bien dans une autre image de même type, en utilisant les résultats de l'étape précédente.

Notre système peut être utilisé dans la première étape. Cependant, la mise en œuvre de cette approche pose certainement le problème suivant : une acquisition d'un objet dans des conditions de prise de vue différentes ou dans des positions différentes engendre des contours différents. En effet, les attributs photométriques du contour changent rapidement en fonction de leurs positions dans l'image et des conditions de prise de vue (voir chapitre 2 de la partie 1). A notre avis, ce problème limite l'utilisation de cette démarche à des applications où l'objet possède des propriétés morphologiques et photométriques proches de celui d'apprentissage.

Outre la sélection des détecteurs, notre système permettra d'approfondir la notion de contours, en particulier :

- la définition des caractéristiques d'un contour,
- la définition des règles de calcul des paramètres d'un détecteur en fonction des données,
- la définition des propriétés d'un détecteur.

La définition correcte de la notion de contour et de calcul automatique des paramètres d'un détecteur permettra certainement une description plus complète de l'image, mieux adaptée aux exigences des applications de la vision par ordinateur, ainsi qu'une meilleure modélisation de l'image.

Conclusion générale

Dans cette étude, nous avons abordé le problème de la sélection automatique de détecteurs et de leurs paramètres pour mettre en évidence un contour donné par un utilisateur. Ce problème vaste et peu étudié nous a conduit à mener des études à la fois théoriques et expérimentales pour mettre en place une approche qui soit non combinatoire. Délibérément, nous nous sommes efforcés d'apporter des réponses précises, fondées et parfois originales aux problèmes sous-jacents qui sont en grande partie ouverts, en particulier :

- la mise en œuvre de détecteurs de contours,
- la spécification de la relation de cause à effet entre les caractéristiques d'un contour, les propriétés d'un détecteur et ses performances.
- la mise en œuvre d'un système de sélection automatique de détecteurs.

Pour disposer d'un ensemble d'outils nécessaires à la détection de contours, nous avons implanté plusieurs algorithmes, surtout nous avons développé et implanté deux détecteurs de contours originaux :

- un détecteur optimal de lignes de crête : celui-ci est issu d'un formalisme récent et universel proposé par Canny. Le problème primordial que nous avons rencontré lors de la conception de ce détecteur est son extension en 2D. En effet, il n'existe pas à notre connaissance de modèle mathématique pour l'identification d'un contour de type crête. Pour ce faire, nous avons proposé l'utilisation de la variance directionnelle. Enfin, ce détecteur est implanté efficacement par des équations aux différences. A ce propos, nous avons montré dans [Zio 91a] que cette méthode d'implantation est l'une des plus efficaces et surtout permet de préserver les propriétés du détecteur initial.
- un détecteur de marches à partir de la distribution fréquentielle de l'image : il est fondé sur les techniques d'apprentissage numériques. Cet algorithme destiné à des images comportant quelques objets posés sur un fond peut être utilisé pour la détection de contours et pour l'extraction de régions. En utilisant un critère d'évaluation fondé sur la conservation de la forme des objets et sur l'homogénéité des régions obtenues, nous avons montré que notre algorithme possède de meilleures performances que d'autres algorithmes connus (en particulier, celui de Pun et celui de Kittler).

Ces deux algorithmes décrits dans la partie 2 ont été développés sans qu'il puisse être intégrés dans le système de sélection de détecteurs.

La définition d'un critère de sélection de détecteurs nécessite de mesurer l'influence des caractéristiques du contour et des propriétés mathématiques d'un détecteur sur ses performances. Dans ce contexte, nous avons effectué :

- une étude bibliographique approfondie pour rassembler le maximum d'expertise nécessaire à la définition de cette relation. Cette étude est présentée en partie 1,
- une étude sur la détection multi-échelle des contours. Cette étude réalisée en collaboration avec S. Tabbone permet d'acquérir des règles de comportement du contour dans le cas des détecteurs dits stables dans un espace-échelle. Elle n'est décrite que brièvement dans cette thèse,
- une étude théorique vérifiée par une évaluation quantitative de l'influence des attributs de contours sur les performances d'un détecteur pour laquelle nous avons obtenu des résultats originaux.

En ce qui concerne la sélection des détecteurs de contours, nous avons proposé une approche incrémentale et non combinatoire utilisant plusieurs sources de connaissance qui sont l'image, les propriétés des détecteurs et leurs résultats effectifs. Le système implanté est composé d'algorithmes parfois originaux et a l'avantage de fournir une solution efficace et assurant le compromis entre deux critères de performance d'un détecteur qui sont de natures antagonistes : la détection et la localisation.

Nous avons considéré un détecteur de contours comme étant un triplet (algorithme, valeur de ses paramètres, précondition limitant son domaine d'application). Un détecteur est représenté par une règle de production issue des heuristiques obtenues dans l'étude mentionnée précédemment et mettant en correspondance les caractéristiques du contour et un détecteur. La sélection est effectuée en considérant le meilleur détecteur au sens des performances qui est validé par les caractéristiques du contour exemple.

La mise en œuvre de ce critère nécessite l'extraction des caractéristiques du contour (son type, sa forme, son rapport signal/bruit, sa netteté et sa largeur) et sa segmentation en h-contour. Il faut ici encore souligner que le problème d'analyse de la structure d'un contour est peu étudié et reconnu difficile du fait que les images sont bruitées et composées de surfaces souvent non linéaires. Pour ce faire, nous avons développé des algorithmes originaux pour l'extraction du bruit dans une image et pour l'estimation des caractéristiques (contraste, netteté, largeur, etc.) d'un contour de type marche simple et de type marche double.

Pour analyser les résultats du système à un moment donné d'une manière efficace, nous avons proposé un algorithme de fusion de l'information contours. Ce problème de fusion posé comme étant un problème d'optimisation d'un critère global est résolu à l'aide d'un algorithme de type A^* . L'algorithme de fusion repose sur la mise en correspondance de segments à l'aide d'une approche globale fondée sur la relaxation continue.

Pour conclure, la sélection des détecteurs appropriés pour traiter une image est un problème important, peu étudié et vaste. Notre contribution à son étude reste modeste et ne peut être considéré que comme étant un début vu que nous sommes encore loin d'une sélection correcte de détecteurs. Il reste à apporter de nombreuses améliorations concernant essentiellement l'introduction des techniques de type intelligence artificielle comme

l'apprentissage numérique pour combler aux inadéquations des approches mathématiques reposant souvent sur des hypothèses simplificatrices. Un système de sélection comprenant de telles améliorations permettra certainement une description plus complète de l'image qui soit mieux adaptée aux exigences des applications de la vision par ordinateur.

Bibliographie

- [Abd 78] I.E. Abdou. Quantitative Methods of Edge Detection. Technical report no. 830, Image Processing Institute, University of Southern California, 1978.
- [Aho 74] A.H. Aho, J.E. Hopcroft, et J.D. Ullman. *The Design and Analysis of Computer Algorithms*. Publishing Company, Reading, Mass., 1974. *Structures de données et algorithmes*, Inter Editions, une traduction d'une édition récente de l'ouvrage précédent, 1987, 1974.
- [Ahu 78] N. Ahuja et A. Rosenfeld. A Note on the Use of Second-Order Gray-Level Statistics for Threshold Selection. *IEEE Transactions. Systems, man, and cybernetics*, SMC-8:895-899, 1978.
- [Ahu 82] N. Ahuja. Dot Processing Using Voronoi Neighborhoods. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 4(3):336-343, May 1982.
- [Ahu 89] N. Ahuja et M. Tuceryan. Extraction of Early Perceptual Structure in Dot Patterns : integrating Region, Boundary, and Component Gestalt. *Computer Vision, Graphics and Image Processing*, 48:304-356, 1989.
- [Ant 91] D. Antoine. *Interprétation des plans cadastraux français à partir d'un modèle*. Thèse de Doctorat, Institut National Polytechnique de Lorraine, 1991.
- [Arb 90] E. Arbogast et R. Mohr. 3D Structures Inference from Images Sequences. H.S. Baird, éditeur, *Pre-proceedings of IAPR Workshop on Syntactic and Structural Pattern Recognition, Murray Hill, NJ (USA)*, pages 21-37, 1990.
- [Arc 85] C. Arcelli et G. Saniti di Baja. A Width-Independent Fast Thinning Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 7(4):463-474, Jul 1985.
- [Aya 86] N. Ayache et O.D. Faugeras. Hyper : A New Approach for the Recognition and Positioning of Two-Dimensional Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(1):44-54, Jan 1986.
- [Aya 88] N. Ayache. *Construction et fusion de représentations visuelles 3D, applications à la robotique mobile*. Thèse d'Etat, Université de Paris-Sud, Centre d'Orsay, 1988.
- [BA 87] J. Ben-Arie et A. Zvi Meiri. 3D Objects Recognition by Optimal Matching Search of Multinary Relations Graphs. *Computer Vision, Graphics and Image Processing*, 37:345-361, 1987.

- [Bab 86] J. Babaud, A.P. Witkin, M. Baudin, et R.O. Duda. Uniqueness of the Gaussian Kernel for Scale-Space Filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(1):26–33, Jan 1986.
- [Bal 82] D.H. Ballard et C.M. Brown. *Computer Vision*. Prestige Hall, 1982.
- [Bar 81a] H.G. Barrow et J.M. Tennenbaum. Computational Vision. *Proceedings of IEEE*, 69:572–594, 1981.
- [Bar 81b] H.G. Barrow et J.M. Tennenbaum. Interpreting Line drawings as Three Dimensional Surfaces. *Artificial Intelligence*, 17(1-3):75–116, 1981.
- [Bar 87] Á. Barr et E.A. Feigenbaum. *Le manuel de l'intelligence artificielle*. Eyrolles, 1987.
- [Bel 91] A. Belaid et J.C. Anigbogu. Text Recognition Using Stochastic Models. *Fifth International Symposium on Applied Stochastic Models and Data Analysis*, Granada, Spain, 1991.
- [Ber 87] F. Bergholm. Edge Focusing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 9(6):726–741, Nov 1987.
- [Bha 81] B. Bhanu. Shape Matching and Image Segmentation Using Stochastic Labeling. Technical report no. 1030, Department of Electrical Engineering, University of Southern California, 1981.
- [Bij 84] A. Bijaoui. *Image et information-Introduction en traitement numérique des images*. Masson, Paris, 1984.
- [Bij 89] A. Bijaoui, E. Slezak, G. Mars, et M. Giudicelli. Détection des objets faibles dans des images célestes à l'aide de la transformée ondelette. *Actes du 12^{ème} Colloque sur le Traitement du Signal et des Images (GRETSI)*, pages 209–212, Nice, 1989.
- [Bin 81] T.O. Binford. Inferring Surfaces from Images. *Artificial Intelligence*, 17:205–244, 1981.
- [Bla 90] A. Blake et R. Cipolla. Robust Estimation of Surface Curvature from Deformation of Apparent Contours. *Proceedings of the First European Conference on Computer Vision (ECCV)*, pages 465–474, april 1990.
- [Blo 89] D. Blostein et N.A. Ahuja. A Multiscale Region Detector. *Computer Vision, Graphics and Image Processing*, 45:22–41, 1989.
- [Boi 90] J.D. Boissonnat, O. Devillers, R. Schott, M. Teillaud, et M. Yvinec. Application of Random Sampling to On-line Algorithms in Computational Geometry. Rapport CRIN 90-R-157, Centre de Recherche en Informatique de Nancy, 1990.
- [Bol 82] R.C. Bolles et R.A. Cain. *Recognising and Locating Partially Visible Objects : the Local-Feature-Focus Method*. In *Robot Vision*, Ed. Alan Pugh, IFS (publications) Ltd., UK, 1982.
- [Bou 68] R. Boudarel, J. Delmas, et P. Guichet. *Commande Optimale des Processus*. Dunod, Paris, 1968.

- [Bou 85] S. Boukharouba, J.M. Rebordao, et P.L. Wendel. An Amplitude Segmentation Method Based on the Distribution Function of an Image. *Computer Vision, Graphics and Image Processing*, 29:47-59, 1985.
- [Bra 83] M. Brady et B.K.P. Horn. Rotationally Symmetric Operators for Surface Interpolation. *Computer Vision, Graphics and Image Processing*, 22:70-94, 1983.
- [Bra 84] M. Brady et A. Yuille. An Extremum Principle for Shape from Contour. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 6(3):288-301, 1984.
- [Bro 81a] A. Broder et A. Rosenfeld. A Note on Collinearity Merit. *Pattern Recognition*, 13(3):237-239, 1981.
- [Bro 81b] R.A. Brooks. Symbolic Reasoning Among 3-D Models and 2-D Images. *Artificial Intelligence*, 17:285-348, 1981.
- [Bur 86] J.B. Burns, A.R. Hanson, et E.M. Riseman. Extracting Straight Lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(4):425-455, Jul 1986.
- [Can 83] J.F. Canny. Finding Edges and Lines in Images. Technical report no. 720, Massachusetts Institut of Technology, 1983.
- [Can 86] J.F. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(6):679-698, Nov 1986.
- [Cas 90] S. Castan, J. Zhao, et J. Shen. Optimal Filter for Edge Detection Methods and Results. *Proceedings of the First European Conference on Computer Vision (ECCV)*, pages 13-17, april 1990.
- [Cha 91] H. Chabbi et G. Masini. A Combined Use of Regions and Segments to Construct Facets. *Proceedings IAPR Workshop on Image Analysis and processing*, 1991.
- [Che 87] J.S. Chen, A. Huertas, et G. Medioni. Fast Convolution with Laplacian-of-Gaussian Masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 9:584-590, 1987.
- [Che 89] J.S. Chen et G. Medioni. Detection, Localisation, and Estimation of Edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 11(2):191-198, 1989.
- [Chi 83] R.T. Chin et C.L. Yeh. Quantitative Evaluation of Some Edge Preserving Noise Smoothing Techniques. *Computer Vision, Graphics and Image Processing*, 23:67-91, 1983.
- [Chi 87] R.T. Chin, H.K. Wan, D.L. Stover, et R.D. Iverson. A One-Pass Thinning Algorithm and its Parallel Implementation. *Computer Vision, Graphics and Image Processing*, 40:30-40, 1987.
- [Cla 89] J.J. Clark. Authenticating Edges Produced by Zero-Crossing Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 11(1):43-57, Jun 1989.

- [Cle 90] V. Clement. Raisonnements cognitifs appliqués au pilotage d'algorithmes de traitement d'images. Thèse de Doctorat, Université de Nice - Sofia Antipolis, 1990.
- [Cro 84] J. Crowley et A. Parker. A Representation for Shape Based on Peaks and Ridges in the Difference of Low Pass Transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 6(2):156-170, 1984.
- [Dan 90] P.E. Danielsson et O. Seger. Rotation Invariance in Gradient and Higher Order Derivative Detectors. *Computer Vision, Graphics and Image Processing*, 49:198-221, 1990.
- [Dav 75] L.S Davis. A survey of Edge Detection Techniques. *Computer Graphics and Image Processing*, 4:248-270, 1975.
- [Dec 89] C. Decaestecker. Formation incrémentale de concepts par un critère d'adéquation. *Quatrième Journée Française de l'Apprentissage*, pages 63-77, 1989.
- [Del 89] S. Dellepiane, D.D. Guisto, et G. Vernazza. Automatic Parameter Computation for Edge Detection by the Zero-Crossing Method. *Actes du 12^{ème} Colloque sur le Traitement du Signal et des Images (GRETSI)*, pages 617-620, Nice, 1989.
- [Der 87] R. Deriche. Using Canny's Criteria to Derive a Recursive Implemented Optimal Edge Detector. *The International Journal of Computer Vision*, 1(2):167-187, 1987.
- [Der 90a] R. Deriche. Fast Algorithm for Low-Level Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 12(1):78-87, 1990.
- [Der 90b] R. Deriche et G. Giraudon. Accurate Corner Detection : An Analytical Study. *Proceedings of the Third International Conference on Computer Vision (ICCV)*, pages 66-70, Osaka, Japan, 1990.
- [Dor 88] C. Dorize. Ondelettes discrètes orthogonales et signaux acoustiques. Mémoire d'ingénieur acoustique CNAM, Ecole Nationale Supérieure de Télécommunications, 1988.
- [Duv 88] J. Duvernoy. Modélisation physique de la formation de l'image. *Actes du 2^{ème} atelier scientifique, TIPI*, pages XII-1-XII-22, Savoie, 1988.
- [Dye 79] C.R. Dyer et A. Rosenfeld. Thinning Algorithms for Gray-Scale Pictures. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1(1):88-89, Jan 1979.
- [EB 90] M. Elarbi-Boudihir, M. Dufour, et R. Husson. Guidage de robot mobile pour détection de bord de route : étude de la phase initiale de navigation. *Revue d'automatique et de productique appliquées*, 3(2):67-81, 1990.
- [Ede 87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, Berlin Heidelberg, Germany, 1987.

- [Fau 81] O.D. Faugeras et M. Berthod. Improving Consistency and Reducing Ambiguity in Stochastic Labeling: an Optimization Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 4:412-423, 1981.
- [Fau 88] O.D. Faugeras. A Few Steps toward Artificial 3D Vision. Rapport de Recherche no. 790, INRIA, Rocquencourt, France, 1988, 1988.
- [For 88] M.R.B. Forshaw. Speeding Up the Marr-Hildreth Edge Operator. *Computer Vision, Graphics and Image Processing*, 41:172-185, 1988.
- [Fuk 72] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic-Press, 1972.
- [Gag 89] A. Gagalowicz. Coopération entre l'analyse et la synthèse d'images. *Actes du 7^{ème} Congrès AFCET Reconnaissance des Formes et Intelligence Artificielle*, pages 1727-1758, Paris, 1989.
- [Gir 85] G. Giraudon. Edge Detection from Local Negative Maximum of Second Derivative. *Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 643-645, San-Francisco Ca. (USA), 1985.
- [Gir 86] G. Giraudon. Utilisation des extrema locaux de la dérivée seconde comme primitives stéréo. *Deuxième Colloque Image*, CESTA, pages 742-748, Nice, France, 1986.
- [Gir 87] G. Giraudon. An Efficient Edge Following Algorithm. *Proceedings of the 5th Scandinavian Conference on Image Analysis*, volume 2, pages 547-554, Stockholm, 1987.
- [Han 86] A.R. Hanson et E.M. Riseman. The VISIONS Image Understanding System. COINS Technical Report no. 86-62, University of Massachusetts, 1986.
- [Har 81] R.M. Haralick et L. Waston. A Facet Model for Image Data. *Computer Graphics and Image Processing*, 15:113-129, 1981.
- [Har 83] R.M. Haralick. Ridge and Valley on Digital Images. *Computer Vision, Graphics and Image Processing*, 22:28-38, 1983.
- [Har 84] R.M. Haralick. Digital Step Edges from Zero Crossing of Second Directional Derivatives. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 6(1):58-68, Jan 1984.
- [Har 85a] R.M. Haralick. Second Directional Derivative Zero-Crossing Detector using the Cubic Facet Model. *Proceedings of the 4th Scandinavian Conference on Image Analysis*, pages 17-30, Trondheim, 1985.
- [Har 85b] R.M. Haralick et L. Shapiro. "Survey": Image Segmentation Techniques. *Computer Vision, Graphics and Image Processing*, 29:100-132, 1985.
- [Has 86] J. Hasegawa, H. Kubota, et J. Toriwaki. Automated Construction of Image Processing Procedures by Sample-Figure Representation. *Proceedings of the 8th International Conference on Pattern Recognition (ICPR)*, pages 586-588, Paris, 1986.

- [Hat 87] J.P. Haton. Computer vision systems. A. Kusiak, éditeur, *Artificial Intelligence: Implications for Computer Integrated Manufacturing*, pages 71–87. IFS Publications Ltd. and Springer Verlag, 1987.
- [Hat 90] J.P. Haton et M.C. Haton. *Intelligence Artificielle*. Que sais-je, 1990.
- [Hei 88] F. Heitz. *Restitution de la radiographie d'un tableau dissimulé par une composition postérieure. Contribution à la détection de changements en imagerie multi-source*. Thèse de Doctorat, Ecole Nationale Supérieure des Télécommunications, 1988.
- [Her 70] A. Herskivitz et T. Binford. On Boundary Detection. M.I.T, AI Memo 183, 1970.
- [Hil 83] E.C. Hildreth. The Detection of Intensity Changes by Computer and Biological Vision Systems. *Computer Vision, Graphics and Image Processing*, 22:1–27, 1983.
- [Hor 77] B.K.P. Horn. Understanding Image Intensities. *Artificial Intelligence*, 8:201–231, 1977.
- [Hor 81] B.K.P. Horn et B.G. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17:185–204, 1981.
- [Hor 89] R. Horaud et T. Skordas. Stereo Correspondance Through Feature Grouping and Maximal Cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 11:1168–1180, 11 1989.
- [Hor 90] B.K.P. Horn. Height and Gradient from Shading. *The International Journal of Computer Vision*, 5:37–75, 1990.
- [Hue 71] M.H. Hueckel. An Operator Which Locates Edges in Digitized Pictures. *J. Assoc. Comput. Mach.*, 19(1):113–125, 1971.
- [Hue 73] M.H. Hueckel. A Local Visual Operator Which recognizes Edges and Lines. *J. Assoc. Comput. Mach.*, 20(4):634–647, 1973.
- [Hum 79] R.A. Hummel. Feature Detection Using Basis Functions. *Computer Graphics and Image Processing*, 9:40–55, 1979.
- [Jar 87] H. Jarosz et J.P. Crettez. Détection des lignes de crête et de vallées dans les images multi-niveaux. *Actes du 6^{ème} Congrès AFCET Reconnaissance des Formes et Intelligence Artificielle*, pages 839–847, Antibes, 1987.
- [Kel 81] M.D. Kelly. Edge Detection in Pictures by Computer Using Planning. in *Machine Intelligence VI*, pages 397–409, University Press, Edinburgh, 1981.
- [Kit 86] J. Kittler et J. Illingworth. Minimum Error Thresholding. *Pattern Recognition*, 19(1):41–47, 1986.
- [Lac 88] V. Lacroix. A Three-Module Strategy for Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 10(6):803–810, Nov 1988.

- [Lac 90] V. Lacroix. The Primary Raster: A Multiresolution Image Description. *Proceedings of the 10th International Conference on Pattern Recognition (ICPR)*, pages 903–907, 1990.
- [Lan 83] P. Langley. Learning Search Strategies Through Discrimination. *The International Journal of Man-Machine Studies*, 18:513–541, 1983.
- [Lev 85] M.D Levine et A.M Nazif. Dynamic Measurement of Computer Generated Image Segmentations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 7(2):155–164, Mar 1985.
- [LL 86] P. Limozin-Long. *Vision stéréoscopique appliquée à la robotique*. Thèse de Doctorat, Université de Nice, 1986.
- [Low 87] D.G. Lowe. Three-Dimensional Object Recognition from Single Two-Dimensional Images. *Artificial Intelligence*, 31:355–395, Mar 1987.
- [Lu 89] Y. Lu et R.M. Jain. Behavior of Edges in Scale Space. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 11(4):337–356, Apr 1989.
- [Lun 86a] W.H.H.J. Lunscher et M.P Beddoes. Optimal Edge Detector Design I : Parameter Selection and Noise Effects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(2):164–177, Mar 1986.
- [Lun 86b] W.H.H.J. Lunscher et M.P Beddoes. Optimal Edge Detector Design II : Coefficient Quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(2):178–187, Mar 1986.
- [Lux 85] A. Lux. *Algorithmique et contrôle en vision par ordinateur*. Thèse d'état, Université Scientifique et Médicale de Grenoble, Institut National Polytechnique de Grenoble, 1985.
- [Ma 87] Y.Z Ma. *Filtrage géostatistique des images numériques*. Thèse de Doctorat, Institut National Polytechnique de Lorraine, 1987.
- [Mal 89] S.G. Mallat. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 11(7):674–693, Jul 1989.
- [Man 87] A. Mansouri, A.S Malowany, et M.D. Levine. Line Detection in Digital Pictures: A Hypothesis Prediction / Verification Paradigm. *Computer Vision, Graphics and Image Processing*, 40:95–114, 1987.
- [Mar 80] D. Marr et E.C. Hildreth. Theory of Edge Detection. *Proceedings of the Royal Society of London B207*, pages 187–217, 1980.
- [Max 77] J. Max. *Méthodes et techniques de traitement du signal et applications aux mesures physiques*. Masson, Paris, 1977.
- [Meh 90] R. Mehrotra, S. Nichani, et N. Ranganathan. Corner Detection. *Pattern Recognition*, 23:1223–1233, 1990.

- [Mic 89] E. De Micheli, B. Caprile, P. Ottonello, et V. Torre. Localization and Noise in Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 11(10):1107–1116, 1989.
- [Moh 88] R. Mohr et G. Masini. Good Old Discrete Relaxation. *Proceedings of the 8th European Conference on Artificial Intelligence*, pages 651–656, Munich, 1988.
- [Moh 89a] R. Mohan et R. Nevatia. Segmentation and Description Based on Perceptual Organization. Rapport, Institute for Robotics and Intelligent Systems School of Engineering, University of Southern California, 1989.
- [Moh 89b] R. Mohan et R. Nevatia. Using Perceptual Organization to Extract 3-D Structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 11(11):1121–1139, 1989.
- [Nag 84] M. Nagao. Control Strategies in Pattern Analysis. *Pattern Recognition*, 17:45–56, 1984.
- [Nak 79] Y. Nakagawa et A. Rosenfeld. Some Experiments On Variable Thresholding. *Pattern Recognition*, 11:191–204, 1979.
- [Nal 86] V.S. Nalwa et T.O. Binford. On Detecting Edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(6):699–713, Nov 1986.
- [Naz 84] A.M. Nazif et M.D. Levine. Low Level Image Segmentation: An Expert System. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 6(5):555–576, Sep 1984.
- [Nev 77] R. Nevatia. Evaluation of a Simplified Hueckel Edge-Line Detector. *Computer Graphics and Image Processing*, 6:582–588, 1977.
- [Nev 78] R. Nevatia. Characterization and Requirements of Computer Vision Systems. A.R. Hanson et E.M. Riseman, éditeurs, *Computer Vision Systems*, pages 81–85. Academic-Press, New York, 1978.
- [Nil 71] N. J. Nilsson. *Problem-Solving Methods in Artificial Intelligence*. McGraw-Hill, 1971.
- [O’C 75] J.F. O’Callaghan. An Alternative Definition for Neighborhood of a Point. *IEEE Transactions on Computer*, C-24:1121–1125, 1975.
- [Ost 78] N. Ostu. A Threshold Selection Method from Gray-Level Histogram. *IEEE Transactions. Systems, man, and cybernetics*, SMC-8:62–66, 1978.
- [Pai 88] C. Pair, R. Mohr, et R. Schott. *Construire les algorithmes*. Dunod, 1988.
- [Par 85] P. Parent et S.W. Zucker. Trace Inference, Curvature Consistency, and Curve Detection. Rapport, Technical report no. CIM-86-3, McGill Research Center for Intelligent Machines, McGill University, Canada, 1985.
- [Pet 89] M. Petrou et J. Kittler. Detecting Edges with Intrinsic Length-Scale. *Proceedings of the 6th Scandinavian Conference on Image Analysis*, volume 2, pages 816–819, Oulu, Finland, 1989.

- [Pog 84] T. Poggio et V. Torre. Ill-Posed Problems and Regularization Analysis in Early Vision. Artificial Intelligence lab. Memo, No. 773, Massachusetts Institut of Technology, 1984.
- [Pog 85] T. Poggio et V. Torre. A Regularized Solution to Edge Detection. Artificial Intelligence lab. Memo, No. 833, Massachusetts Institut of Technology, 1985.
- [Pon 85] J. Ponce et M. Brady. Toward a Surface Primal Sketch. *Three Dimensional Vision*, T. Kanade, Ed. New York: Academic Press, 1985.
- [Pra 78] W.K Pratt. *Digital Image Processing*. Wiley-Interscience Publication, 1978.
- [Pre 66] J.M.S Prewitt et M.L. Mendelsohn. The Analysis of Cell Images. In *Ann. New York Acad. Sci, Vol 128*, pages 1035-1053, New York, 1966.
- [Pre 70] J.M.S Prewitt. Object Enhancement and Extraction. In *Picture Processing and Psychopictorics*, B.S. Lipkin and al., Eds. Academic Press, New York, 1970.
- [Pre 88] W.H. Press, B.P. Flannery, S.A. Teukolsky, et W.T. Vetterling. *Numerical Recipes in C*, The Art of Scientific Computing. Cambridge University Press, 1988.
- [Pug 82] V. Pugachev. *Théorie des probabilités et statistique mathématique*. Editions de Moscou, Mir, 1982.
- [Pun 81] T. Pun. Entropic Thresholding : A New Approach. *Computer Vision, Graphics and Image Processing*, 16:210-239, 1981.
- [Qua 89] L. Quan. *Contribution de la vision monoculaire à la perception tridimensionnelle*. Thèse de Doctorat, Institut National Polytechnique de Lorraine, 1989.
- [Ran 89] K. Rangarajon, M. Shah, et D.V. Brackle. Optimal Corner Detector. *Computer Vision, Graphics and Image Processing*, 48:230-245, 1989.
- [Ros 71] A. Rosenfeld et M. Thurston. Edge and Curve Detection for Visual Scene Analysis. *IEEE Transactions on Computer*, 20(5):562-569, May 1971.
- [Ros 76] A. Rosenfeld, R. Hummel, et S. Zucker. Scene Labeling by Relaxation Operations. *IEEE Transactions on Computer*, SMC-6:420-433, 1976.
- [Ros 84] A. Rosenfeld. Image Analysis : Problems, Progress and Prospects. *Pattern Recognition*, 17(1):3-12, 1984.
- [Rut 79] W.S Rutkowski. Note Shape Completion. *Computer Graphics and Image Processing*, 9:89-101, 1979.
- [Sah 88] P.K. Sahoo, S. Soltani, A.K.C. Wong, et Y.C. Chen. A Survey of thresholding techniques. *Computer Vision, Graphics and Image Processing*, 41:233-260, 1988.
- [Ser 82] J. Serra. *Image Analysis and Mathematical Morphology*. Academic-Press, London, 1982.
- [Ser 89] M.A. Serendero. *Extraction d'informations symboliques en imagerie SPOT : réseaux de communication et agglomérations*. Thèse de Doctorat, Université de Nice, 1989.

- [Sha 79] K.S. Shanmugam, F.M. Dickey, et J.A. Green. An Optimal Frequency Domain Filter for Edge Detection in Digital Pictures. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1(1):37-49, Jan 1979.
- [She 85] J. Shen et S. Castan. Un nouvel algorithme de détection de contours. *Actes du 5^{ème} Congrès AFCET Reconnaissance des Formes et Intelligence Artificielle*, pages 201-213, Grenoble, 1985.
- [She 86] J. Shen. Filtrage rapide en traitement d'image et vision 3d par ordinateur. Thèse d'Etat, Université Paul Sabatier, Toulouse, 1986.
- [She 87] J. Shen et S. Castan. Edge Detection Based on Multi-Edge Models. *SPIE*, pages 46-53, Cannes, France, 1987.
- [Sjo 87] F. Sjoberg et F. Bergholm. Extraction of Diffuse Edge by Edge Focusing. *Proceedings of the 5th Scandinavian Conference on Image Analysis*, volume 1, pages 23-33, Stockholm, 1987.
- [Sle 86] A.C. Sleight. The Extraction of Boundaries Using Local Measures Driven by Rules. *Pattern Recognition Letters*, 4:247-258, Sep 1986.
- [SM 88] P. Saint-Marc. Outils fondamentaux pour l'analyse d'images en robotique. Thèse de Doctorat, Université de Blaise Pascal, 1988.
- [Smi 87] R.W. Smith. Computer Processing of Line Images: A Survey. *Pattern Recognition*, 20(1):7-15, 1987.
- [Sot 89] G.E. Sotak et K.L. Boyer. The Laplacien-of-Gaussien Kernel: A Formal Analysis and Design Procedure for Fast, Accurate Convolution and Full-Frame Output. *Computer Vision, Graphics and Image Processing*, 48:147-189, 1989.
- [Sta 87] V.J. Stanger. A Comparative Study of Practical Image Segmentation Techniques. *Proceedings of the 5th Scandinavian Conference on Image Analysis*, volume 2, Stockholm, 1987.
- [Suk 88] M. Suk et S. Hong. An Edge Extraction Technique for Noisy Images. *Computer Vision, Graphics and Image Processing*, 25:24-45, 1988.
- [Tab 90] A. Tabbone. Vers une approche multi-échelle pour la détection de contours. Rapport de DEA, CRIN, 1990.
- [Tab 91a] S. Tabbone et D. Ziou. Description multi-échelle d'une image fondée sur le comportement du contour. *Actes du 8^{ème} Congrès AFCET Reconnaissance des Formes et Intelligence Artificielle, Lyon, à paraître*, 1991.
- [Tab 91b] S. Tabbone et D. Ziou. Elimination des faux contours par séparation et propagation de seuil. *13^{ème} Colloque sur le Traitement du Signal et des Images (GRETSI), Juan-les-Pins, à paraître*, 1991.
- [The 86] R. Theodor. *Initiation à l'analyse numérique*. Masson, 1986.
- [Thi 89] E. Thirion. *Interprétation et apprentissage géométrique en vision par ordinateur*. Thèse de Doctorat, Institut National Polytechnique de Lorraine, 1989.

- [Tik 77] A.N. Tikhonov et V.Y Arsenin. *Solution to Ill-posed Problems*. Winston and Sons, Washington D.C, 1977.
- [Tom 87] K. Tombre. *La saisie automatisée de documents composites : reconnaissance, codage et interprétation des parties graphiques*. Thèse de Doctorat, Institut National Polytechnique de Lorraine, 1987.
- [Tor 86] V. Torre et T.A. Poggio. On Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(2):147-163, Mar 1986.
- [Tsa 85] W. Tsai. Moment-Preserving Thresholding: A New Approach. *Computer Vision, Graphics and Image Processing*, 29:377-393, 1985.
- [Tuc 87] M. Tuceryan, A.K. Jain, et Y. Lee. Extracting Perceptual Structures in Line Patterns. *Proceedings of the 5th Scandinavian Conference on Image Analysis*, volume 2, pages 531-538, Stockholm, 1987.
- [Ull 76] S. Ullman. Filling-in-the-gaps: The shape of subjective contours and a model for their completion. *Biological Cybernetics*, 1(6), 1976.
- [Ulu 88] F. Ulupinar et R. Nevatia. Using Symmetries for analysis of Shape from Contour. *Proceedings of the Second International Conference on Computer Vision (ICCV)*, pages 414-426, Tampa, Florida, 1988.
- [VIS 88] VISILOG. User's Guide. Noesis, Jouy-en-Josas, France, 1988.
- [Vog 79] M.A. Vogel et A.K.C. Wong. PFS Clustering Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1(3):237-245, Jul 1979.
- [Voy 87] R. Voyer. *Moteurs de systèmes experts*. Eyrolles, 1987.
- [WD 88] B. Wrobel-Dautcourt. *Perception de la distance par mise en correspondance de régions dans des images stéréoscopiques*. Thèse de Doctorat, Institut National Polytechnique de Lorraine, 1988.
- [Web 84] J.A. Webb et E. Pervin. The Shape of Subjective Contours. *AAAI*, pages 340-343, 1984.
- [Wit 83] A. P. Witkin. Scale-Space Filtering. *Proceedings of the 4th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1019-1022, 1983.
- [Wu 90] L. Wu et Z. Xie. Scaling Theorems for Zero-Crossings. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 12(1):46-54, Jan 1990.
- [Yui 86] A.P Yuille et T.A. Poggio. Scaling Theorems for Zero-Crossings. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(1):15-25, Jun 1986.
- [Zah 89] J. Zaho. *Extraction d'information tri-dimensionnelle par stéréovision*. Thèse de Doctorat, Université Paul Sabatier, Toulouse, 1989.
- [Zio 89a] D. Ziou. Vers un système de détection de contours par apprentissage. Rapport CRIN 89-R-282, Séminaire GRECO-PRC Vision, Aydat, 1989.

- [Zio 89b] D. Ziou et B. Wrobel-Dautcourt. Filtres récurrents pour la détection de contours "ligne de crêtes" et "marches". *Actes du 12^{ème} Colloque sur le Traitement du Signal et des Images (GRETSI)*, pages 637-640, Nice, 1989.
- [Zio 89c] D. Ziou et B. Wrobel-Dautcourt. Un détecteur optimal de lignes de crêtes. *Actes du 7^{ème} Congrès AFCET Reconnaissance des Formes et Intelligence Artificielle*, pages 53-62, Paris, 1989.
- [Zio 91a] D. Ziou. Line Detection Using an Optimal IIR Filter. *Pattern Recognition*, 24(6):465-478, 1991.
- [Zio 91b] D. Ziou et S. Tabbone. Fusion des contours dans un espace échelle. *13^{ème} Colloque sur le Traitement du Signal et des Images (GRETSI), Juan-les-Pins, à paraître*, 1991.
- [Zio 91c] D. Ziou et B. Wrobel-Dautcourt. La problématique de la détection de contours. *Courrier du CNRS (numéro spécial : traitement du signal et des images)*, Juin, 1991.
- [Zuc 76] S.W. Zucker. "Survey": Region Growing: Childhood and Adolescence. *Computer Vision and Image Processing*, 5:382-399, 1976.
- [Zuc 79] S.W. Zucker. Toward a Low-Level Description of dot Clusters: Labelling Edge, Interior and Noise Points. *Computer Vision and Image Processing*, 9:213-233, 1979.

**AUTORISATION DE SOUTENANCE DE THESE
DU DOCTORAT DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE**

VU LES RAPPORTS ETABLIS PAR :

**Monsieur CASTAN, Professeur, IRIT/CERFIA Toulouse,
Monsieur HUSSON, Professeur, ENSEM/INPL.**

Le Président de l'Institut National Polytechnique de Lorraine, autorise :

Monsieur ZIOU Djemel

à soutenir devant l'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE, une thèse
intitulée :

**"La détection de contours dans des images à niveaux de gris : mise en oeuvre et
sélection de détecteurs"**

en vue de l'obtention du titre de :

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

Spécialité : **"INFORMATIQUE"**

Fait à Vandoeuvre le, **11 Octobre 1991**

Le Président de l'I.N.P.L.,

M. GANTOIS



