

Résumé

Rousseau, K. (2008) Restauration adaptative des contours par une approche inspirée de la prédiction des performances. Thèse de doctorat, Université de Sherbrooke, Canada, 147 p.

En télédétection, les cartes de contours peuvent servir, entre autres choses, à la restitution géométrique, à la recherche d'éléments linéaires, ainsi qu'à la segmentation. La création de ces cartes est faite relativement tôt dans la chaîne de traitements d'une image. Pour assurer la qualité des opérations subséquentes, il faut veiller à obtenir une carte de contours précise. Notre problématique est de savoir s'il est possible de diminuer la perte de temps liée au choix d'algorithme et de paramètre en corrigeant automatiquement la carte de contours.

Nous concentrerons donc nos efforts sur le développement d'une méthode de détection/restauration de contours adaptative. Notre méthode s'inspire d'une technique de prédiction des performances d'algorithmes de bas niveau. Elle consiste à intégrer un traitement par réseau de neurones à une méthode « classique » de détection de contours. Plus précisément, nous proposons de combiner la carte de performances avec la carte de gradient pour permettre des décisions plus exactes. La présente étude a permis de développer un logiciel comprenant un réseau de neurones entraîné pour prédire la présence de contours. Ce réseau de neurones permet d'améliorer les décisions de détecteurs de contours, en réduisant le nombre de pixels de fausses alarmes et de contours manqués.

La première étape de ce travail consiste en une méthode d'évaluation de performance pour les cartes de contours. Une fois ce choix effectué, il devient possible de comparer les cartes entre elles. Il est donc plus aisé de déterminer, pour chaque image, la meilleure détection de contours. La revue de la littérature réalisée simultanément a permis de faire un choix d'un groupe d'indicateurs prometteurs pour la restauration de contours. Ces derniers ont servi à la calibration et à l'entraînement d'un réseau de neurones pour modéliser les contours. Par la suite, l'information fournie par ce réseau a été combinée par multiplication arithmétique avec les cartes d'amplitudes de détecteurs « classiques » afin de fournir de nouvelles cartes d'amplitude du gradient. Le seuillage de ces contours donne des cartes de contours « optimisées ».

Sur les images aéroportées du jeu de données South Florida, la médiane des mesures-F de la pour

l'algorithme de Sobel passe de 51,3 % avant la fusion à 56,4 % après. La médiane des mesures-F pour l'algorithme de Kirsch amélioré est de 56,3 % et celle de Frei-Chen amélioré est de 56,3 %. Pour l'algorithme de Sobel avec seuillage adaptatif, la mesure-F médiane est de 52,3 % avant fusion et de 57,2 % après fusion. En guise de comparaison, la mesure-F médiane pour le détecteur de Moon, mathématiquement optimal pour contours de type « rampe », est de 53,3 % et celle de l'algorithme de Canny, est de 61,1 %. L'applicabilité de notre algorithme se limite aux images qui, après filtrage, ont un rapport signal sur bruit supérieur ou égal à 20.

Sur les photos au sol du jeu de données de South Florida, les résultats sont comparables à ceux obtenus sur les images aéroportées. Par contre, sur le jeu de données de Berkeley, les résultats n'ont pas été concluants. Sur une imagerie IKONOS du campus de l'Université de Sherbrooke, pour l'algorithme de Sobel, la mesure-F est de 45,7 % \pm 0,9 % avant la fusion et de 50,8 % après. Sur une imagerie IKONOS de l'Agence Spatiale Canadienne, pour l'algorithme de Sobel avec seuillage adaptatif, la mesure-F est de 35,4 % \pm 0,9 % avant la fusion et de 42,2 % après. Sur cette même image, l'algorithme de Argyle (Canny sans post-traitement) a une mesure-F de 35,1 % \pm 0,9 % avant fusion et de 39,5 % après.

Nos travaux ont permis d'améliorer la banque d'indicateurs de Chalmond, rendant possible le prétraitement avant le seuillage de la carte de gradient. À chaque étape, nous proposons un choix de paramètres permettant d'utiliser efficacement la méthode proposée. Les contours corrigés sont plus fins, plus complets et mieux localisés que les contours originaux. Une étude de sensibilité a été effectuée et permet de mieux comprendre la contribution de chaque indicateur. L'efficacité de l'outil développé est comparable à celle d'autres méthodes de détection de contours et en fait un choix intéressant pour la détection de contours. Les différences de qualité observées entre notre méthode et celle de Canny semble être dues à l'utilisation, ou non, de post-traitements.

Grâce au logiciel développé, il est possible de réutiliser la méthodologie; cette dernière a permis d'opérationnaliser la méthode proposée. La possibilité de réutiliser le filtre, sans réentraînement est intéressante. La simplicité du paramétrage lors de l'utilisation est aussi un avantage. Ces deux facteurs répondent à un besoin de réduire le temps d'utilisation du logiciel.

Mots-clés : détection de signal adaptative, caractérisation des performances, réseaux de neurones, imagerie panchromatique, analyse de contours.

Département de géomatique appliquée
Faculté des lettres et sciences humaines
Université de Sherbrooke

Restauration adaptative des contours
par une approche inspirée de la prédiction des performances

Kami ROUSSEAU

Thèse présentée pour l'obtention du grade de Philosophiae Doctor (Ph.D.) en télédétection

Septembre 2008

© Kami ROUSSEAU, 2008

F - 2315



Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*

ISBN: 978-0-494-42634-0

Our file *Notre référence*

ISBN: 978-0-494-42634-0

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.



Canada

Directeur de recherche : Goze B. Béné

Codirecteur de recherche : Langis Gagnon (CRIM)

Membres du jury :

Dong-Chen He

Samuel Corgne

Benoît St-Onge (UQAM)

Table des matières

Liste des figures.....	ii
Liste des tableaux.....	iii
Liste des annexes.....	iv
Liste des abréviations, des sigles et des acronymes utilisés.....	v
Remerciements.....	vi
1. Introduction.....	1
1.1 Rétrospective du problème.....	5
1.2 Problématique.....	6
1.3 Objectifs de recherche.....	7
1.4 Hypothèses de recherche.....	8
1.5 Délimitation de l'étude.....	8
1.6 Plan de la thèse.....	9
2. Cadre théorique.....	10
2.1 Détection de contours.....	10
2.1.1 Détection basée sur la dérivée première.....	14
2.1.2 Détection basée sur la dérivée seconde.....	16
2.1.3 Appariement de formes.....	18
2.1.4 Techniques diverses.....	19
2.1.5 Base théorique.....	21
2.2 Théorie de l'évaluation des performances.....	29
2.2.1 Utilisation d'une vérité-terrain.....	32
2.2.2 Méthode de mise en correspondance.....	35
2.2.3 Méthodes quantitatives d'évaluation avec vérité-terrain.....	36
2.3 Théorie de la prédiction des performances.....	41
2.4 Sélection d'images.....	44
3. Méthodologie.....	48
3.1 Méthodologie générale.....	48
3.1.1 Développement d'outils d'évaluation.....	50
3.1.2 Création de cartes de contours.....	52
3.1.3 Choix d'indicateurs.....	54
3.1.4 Calibration et entraînement de réseaux de neurones.....	55
3.1.5 Restauration de cartes de contours.....	59
3.1.6 Caractérisation de la robustesse au bruit.....	60
3.1.7 Comparaison avec d'autres jeux de données.....	61
3.1.8 Analyse de sensibilité.....	62
3.2 Ensembles de données.....	62
3.2.1 Photos aériennes et au sol.....	63
3.2.2 Images artificielles.....	65
3.2.3 Images de Berkeley.....	66
3.2.4 Images IKONOS.....	68
3.3 Conclusion partielle.....	71
4. Présentation des résultats.....	72
4.1 Mise au point du réseau de neurones.....	72

4.2	Restauration de cartes de contours.....	76
4.3	Robustesse au bruit.....	81
4.4	Comparaison avec d'autres jeux de données.....	83
4.5	Analyse de sensibilité.....	89
4.6	Conclusion partielle.....	90
5.	Interprétation des résultats.....	91
5.1	Mise au point du réseau de neurones.....	91
5.2	Restauration de cartes de contours.....	94
5.2.1	Effets attendus.....	94
5.2.2	Sobel global.....	96
5.2.3	Sobel adaptatif.....	97
5.2.4	Kirsch.....	99
5.2.5	Frei-Chen.....	100
5.2.6	Moon.....	101
5.2.7	Canny.....	102
5.2.8	Synthèse.....	103
5.3	Robustesse au bruit.....	104
5.4	Comparaison avec d'autres jeux de données.....	106
5.5	Analyse de sensibilité.....	108
5.6	Conclusion partielle.....	108
6.	Discussion des résultats.....	109
6.1	Simplicité d'utilisation et économie de temps.....	109
6.2	Adaptation de contours.....	110
6.3	Comparaison des effets observés à d'autres détecteurs.....	111
6.4	Conséquences et limites dues aux choix méthodologiques.....	112
6.4.1	Choix des indicateurs.....	113
6.4.2	Entraînement.....	113
6.4.3	Vérité-terrain.....	114
6.4.4	Type d'images.....	115
6.5	Conclusion partielle.....	115
7.	Conclusion et recommandations.....	116
7.1	Recherche effectuée.....	116
7.2	Limitations et recommandations.....	117
7.3	Perspectives.....	118
8.	Références.....	119

Liste des figures

2.1	Filtres de Roberts.....	14
2.2	Filtres de Prewitt.....	15
2.3	Filtres de Sobel.....	15
2.4	Masque laplacien.....	17
2.5	Filtres de Kirsch à 0° et 45°.....	19
2.6	Filtres « contours » de Frei et Chen.....	19
2.7	Filtres « lignes » de Frei et Chen.....	19
2.8	Filtre « moyenne » de Frei et Chen.....	19

2.9 Logiciel PerfPredict.....	45
2.10 Réduction de FP sur l'image « baseball ».....	46
3.1 Organigramme méthodologique.....	49
3.2 Méthodologie du calcul de contours améliorés basé sur l'indice de présence de contours.....	50
3.3 Architecture du réseau de neurones.....	59
3.4 Exemples d'images du jeu de données de South Florida.....	63
3.5 Image artificielle, SNR = 5.....	66
3.6 Images du jeu de données de Berkeley.....	67
3.7 Image IKONOS de l'Université de Sherbrooke.....	69
3.8 Agence Spaticale Canadienne, image IKONOS et vérité-terrain associée.....	70
4.1 Saisie d'écran du logiciel PerfAmel.....	72
4.2 Indicateurs de Chalmond.....	73
4.3 Indice de présence de contours (faible contraste et bruit).....	74
4.4 Indice de présence de contours.....	76
4.5 Fusion de contours (niveaux de gris égalisés).....	77
4.6 Fusion de contours (noir et blanc).....	78
4.7 Stratégies de seuillage.....	80
4.8 Sensibilité au bruit.....	81
4.9 Effet du lissage.....	83
4.10 Effet du lissage, image « aménagement paysager ».....	86
4.11 Fusion sur l'image du campus (avant et après), détecteur de Sobel classique.....	87
4.13 Fusion sur l'image de l'Agence (avant et après), détecteur de Sobel adaptatif.....	88
4.13 Effet de l'atténuation du signal, image du campus.....	90
5.1 Zone homogène (pâle) et zone texturée (foncé).....	93
5.2 Restauration de l'algorithme de Sobel.....	99
5.3 Restauration de l'algorithme de Kirsch.....	100
5.4 Restauration de l'algorithme de Frei-Chen.....	101
5.5 Détérioration de la mesure-F après la fusion (algorithme de Canny).....	103

Liste des tableaux

1. Mesure-F sur les images aériennes SouthFlorida (1).....	79
2. Mesure-F sur les images aériennes SouthFlorida (2).....	81
3. Mesure-F sur l'image « baseball+bruit » SouthFlorida.....	82
4. Mesure-F après entraînement sur image bruitée.....	83
5. Mesure-F sur les photos au sol SouthFlorida (Seuillage simple).....	84
6. Mesure-F sur les photos au sol SouthFlorida (Seuillage adaptatif).....	85
7. Mesure-F sur les images IKONOS.....	86
8. Indicateurs employés seuls pour détecter les contours.....	89
9. Combinaisons d'indicateurs employées pour la fusion.....	89

Liste des annexes

1. Bibliographie thématique.....	128
2. Analyse théorique de la figure de mérite de Pratt.....	131
3. Algorithme d'appariement de contours.....	133
4. Rendement et mesure-F.....	134
5. Utilisation du logiciel PerfAmel.....	136

Liste des abréviations, des sigles et des acronymes utilisés

*	opérateur de convolution
∂	dérivée partielle
$\nabla(x, y)$	gradient en un point
∇^2	opérateur laplacien
α	niveau de signification statistique
β	constante pénalisant les décalages dans la figure de mérite de Pratt
η	taux d'apprentissage
θ	pente
$\theta(x,y)$	direction, phase du gradient en un point
μ	constante d'inertie (<i>momentum</i>)
ξ	seuil (<i>threshold</i>)
$\rho(x, y)$	amplitude du gradient en un point
σ	paramètre de dispersion
ϕ	direction d'un filtre de Gabor
b	niveau de bleu
B	bruit
C(s)	confiance sur un site
CPN	coefficient de prédictivité négative
CPP	coefficient de prédictivité positive
d	distance entre un contour présumé et le contour réel le plus près
E	filtre de contours de Frei-Chen
F	mesure-F
FN	faux négatif
FP	faux positif
g	niveau de vert
h	hauteur
I	image
K_{ij}	coefficient dans un filtre de convolution
M_{Pratt}	figure de mérite de Pratt
N	largeur d'une fenêtre (nombre de pixels)
O(·)	complexité asymptotique d'un algorithme (notation grand O de Landau)
P	précision
P(A)	proportion de pixels de contours détectés par un algorithme
P(G)	prévalence des pixels de contours dans la vérité-terrain (<i>ground-truth</i>)
r	niveau de rouge
R	rappel
ROC	<i>receiver operating characteristic</i> , fonction d'efficacité de l'observateur
S	silence
SIG	systèmes d'information géographique
SNR	<i>signal to noise ratio</i> , rapport signal sur bruit
VN	vrai négatif
VP	vrai positif
w	filtre de convolution
Y	luminance

Remerciements

Cette thèse s'inscrit dans le cadre du projet SITI (Système Intelligent en Traitement d'Image), qui vise à créer un système intelligent de traitement d'images de télédétection. Plusieurs personnes et organisations ont contribué à sa réalisation.

Je remercie le Dr. Goze Bertin Bénié, mon directeur, pour son support inconditionnel durant toutes les étapes ayant mené à cette thèse. Merci au Dr. Langis Gagnon, qui, par ses encouragements, m'a fait apprécier l'importance de la rigueur scientifique et m'a permis de d'atteindre les résultats espérés. Merci aussi à la Dre. Claire Müller-Pointevien; son soutien moral m'a aidé à traverser les périodes difficiles et à mener à terme ce travail. Merci et bon succès à mes collègues étudiants du département de géomatique appliquée; vous m'avez fournit un environnement de travail stimulant.

Je suis reconnaissant au fond FRNT-NATEC, pour le support financier ayant facilité la réalisation de cette thèse. Au CRIM, pour le financement et l'image IKONOS de Montréal. Merci au Dr. Dong-Chen He, pour l'autorisation d'utiliser l'image de Sherbrooke. Merci à l'Université of South Florida, pour le jeu de données de contours et à l'Université Berkeley, pour le jeu de données de segmentation.

Je désire exprimer ma gratitude à mon père et ma mère, pour leur amour, leur soutien inconditionnel et pour m'avoir inculqué les valeurs essentielles.

Kami ROUSSEAU

Sherbrooke, Canada

1. Introduction

La disponibilité croissante des images de télédétection et la diversification des types de données disponibles contribuent au raffinement des méthodes employées dans les applications telles que l'agriculture de précision et le suivi du développement urbain. Ces images ne sont pas toujours directement utilisables. Il faut d'abord déterminer quelles images utiliser et en extraire l'information utile, sous forme, par exemple, d'une carte thématique. Bien qu'un humain puisse préparer une telle carte, le temps de préparation est élevé et la qualité du résultat est variable. L'automatisation partielle du processus de création cartographique est donc intéressante.

La carte est préparée à l'aide d'un système d'information géographique (SIG), un logiciel permettant de gérer et traiter l'information géographique. Pour faciliter leur développement informatique, les SIG sont séparés en plusieurs modules spécialisés, correspondant aux étapes de la préparation d'une carte.

Dans une telle approche modulaire, les images peuvent être d'abord mosaïquées, corrigées radiométriquement et géométriquement. Leur contenu est ensuite classifié pour déterminer l'occupation du sol. Lorsque le contenu de l'image est simple, cette classification peut être effectuée directement, en employant, par exemple, un algorithme de classification par maximum de vraisemblance (Duda *et al.*, 2000). Le plus souvent, il est nécessaire d'utiliser des méthodes de classification plus complexes. Ces méthodes exigent la décomposition préalable de l'image en primitives, comme des segments ou des contours. Les segments sont des régions ou entités homogènes. Les contours sont des courbes indiquant la présence de discontinuités englobant ces régions homogènes. Les propriétés de chaque primitive sont ensuite mesurées pour permettre d'appliquer un algorithme de classification, par exemple un système expert de classification (Voirin, 2004).

Sur certaines images, la segmentation par région a de la difficulté à localiser précisément les frontières. L'emploi d'une technique de segmentation par contour conjointement à la segmentation par région est alors indiqué. Parmi les domaines d'application pouvant bénéficier d'une telle segmentation hybride, notons la segmentation d'images pétrographiques (Zhoua *et al.*, 2004), la classification contextuelle agricole et urbaine (Stuckens *et al.*, 2000), la segmentation de

brûlis (Zhang *et al.*, 2005) et, en médecine, la segmentation d'images acquise par résonance magnétique (Jiménez-Alaniz *et al.*, 2006).

Le traitement peut aussi être axé exclusivement sur la localisation des frontières entre les objets, ou sur la localisation d'objets filiformes. Les primitives de types contours sont alors employées seules. La démarche consiste alors à décider, pour chaque pixel, si ce dernier est un contour ou s'il n'en est pas un. En imagerie médicale, cette façon de faire est parfois utilisée pour l'extraction (Sofka et Stewart, 2003; Mendonça et Campilho, 2006) et le recalage (Laliberté *et al.*, 2003; Lalonde *et al.*, 2000) de vaisseaux sanguins de la rétine.

En télédétection, la détection de contours est utilisée pour plusieurs tâches reliées à la planification du développement urbain, telles que : le repérage de points caractéristiques avant le recalage d'images (Klepko, 1997; Thepaut *et al.*, 2000), la détection automatique des routes (Yuille et Coughlan, 2000) ou des changements au niveau du bâti (Phalke et Couloigner, 2004). La présente étude s'intéresse à l'extraction de telles primitives.

La localisation des discontinuités dans l'image présente plusieurs défis, particulièrement quand elles sont diffuses dans l'espace ou quand le contraste de leurs niveaux de gris est faible. Les nouvelles images de télédétection ont un contenu complexe, plus riche en détails que celles à moyenne résolution. Il est donc nécessaire d'avoir de nouveaux outils de traitement adaptés à la nouvelle génération d'images de télédétection. D'un point de vue théorique, les facteurs de succès de la détection de contours sur de telles images gagneraient à être mieux compris. Étant donné que chaque type d'image possède des propriétés particulières, il est difficile de concevoir un algorithme de détection de contours bien adapté à plusieurs contextes.

Les détecteurs « classiques », basés sur la variation des niveaux de gris, sont déjà spécifiques aux contours; ils confondent peu l'arrière-plan et les contours. Ils conviennent aux images à moyenne et haute résolution. Cependant, ils ne sont pas adaptés au contexte actuel en télédétection; les images à très haute résolution spatiale contiennent plus de texture à l'intérieur des régions d'intérêts et la prévalence des pixels délimitant les régions d'intérêt est faible. L'opérationnalisation des détecteurs sur de telles images passe par une optimisation pour les rendre plus précis, ce qui consiste à réduire le nombre de faux contours détectés, qui tend à être élevé par rapport au nombre de contours détectés correctement.

Tout au long de l'histoire du traitement d'images numériques, plusieurs détecteurs de contours ont été conçus. Ils se distinguent par différentes propriétés mathématiques et algorithmiques (Ziou et Tabbone, 1998). Chaque détecteur possède son propre modèle pouvant expliquer les structures de l'image qu'il peut détecter et la détérioration de sa réponse lorsque l'image est bruitée. Par exemple, le détecteur de Canny (1986) est conçu pour répondre de façon optimale à un contour en « marche d'escalier, » tout en étant peu affecté par le bruit blanc gaussien. Bien que plusieurs concepteurs d'algorithmes fournissent une description du modèle théorique et de la démarche utilisée pour concevoir leur détecteur, cette description est parfois omise.

Encore aujourd'hui, le nombre d'algorithmes de détection de contours disponibles continue d'augmenter. Il devient difficile de déterminer quel algorithme convient le mieux pour traiter les images destinées à une application donnée. Quand les propriétés d'un détecteur sont peu ou pas connues, ou quand il faut traiter une image dont on connaît mal les caractéristiques, une question émerge naturellement : « pour cette image, est-ce que ce détecteur est le plus fiable parmi ceux disponibles ? »

Devant la variété de détecteurs disponibles, il est souvent difficile de faire un choix. Dans les cas ambigus, le choix du détecteur est fonction de l'expérience passée sur des images semblables. Bien qu'il se base sur son expérience, l'utilisateur risque de procéder par tâtonnement, d'essayer plusieurs détecteurs avant de fixer son choix. Il peut arriver que l'utilisateur choisisse un détecteur mal adapté à l'image. Dans ce cas, bien qu'il prenne le temps d'ajuster au mieux les paramètres, le résultat obtenu ne peut jamais être le meilleur. L'utilisateur pourrait aussi décider d'utiliser les mêmes paramètres pour toutes les images. Cela peut empêcher d'obtenir des résultats optimaux.

Pour assurer la qualité des opérations subséquentes, il faut veiller à obtenir une carte de contours précise. L'estimation des performances attendues pourrait, dans certains cas, aider à justifier l'achat de nouvelles images, si la précision attendue ne correspond pas aux besoins de l'application visée. Il nous apparaît aussi utile d'ajouter de l'intelligence aux SIG en les rendant capables de guider l'utilisateur lors du choix et de l'utilisation des algorithmes pouvant traiter efficacement une image. Pour ce faire, il existe trois familles d'approches :

- caractériser l'algorithme a posteriori pour son l'optimisation (évaluer les performances);

- caractériser l'algorithme a priori pour faciliter le choix des traitements (prédire les performances);
- corriger le résultat d'un détecteur de contours classique en s'inspirant des deux premières familles d'approches.

L'évaluation expérimentale des performances de détecteurs peut servir de base au choix d'algorithmes. Ces évaluations sont des études de l'effet du choix de paramètres sur la qualité du résultat fourni par un détecteur donné. Une telle étude est habituellement faite sur plusieurs images. Les résultats sont souvent présentés sous forme d'une courbe résumant la relation entre la sensibilité du détecteur et sa spécificité (ou sa précision). Dans une approche basée sur l'intelligence artificielle, les connaissances sur l'appariement d'images et d'algorithmes pourraient être intégrées dans un système à base de cas ou de règles (Vasconcelos *et al.*, 2002).

La prédiction des performances peut être utilisée dans les cas où l'évaluation est difficile à réaliser. Les performances de détecteurs de contours sont alors estimées via une approche numérique. Une telle estimation se base sur le contenu de chaque image à traiter. Ainsi, elle prend en compte des variations locales pour guider le choix du détecteur à utiliser. Il s'agit alors d'utiliser l'apprentissage statistique pour entraîner un opérateur estimant la probabilité qu'il y ait vraiment un contour quand un détecteur donné détermine qu'il y en a un, ainsi que la probabilité qu'il n'y ait pas de contour lorsque le détecteur détermine qu'il n'y en a pas (Chalmond *et al.*, 2001).

La prédiction des performances peut être une façon de démontrer à l'utilisateur si l'algorithme choisi a ou non la capacité de résoudre son problème. Ainsi, les pertes de temps dues à un choix d'algorithme par essai et erreur pourront être évitées. De plus, la création de cartes de confiance peut convaincre les usagers de la fiabilité des algorithmes qu'ils utilisent déjà. Dans un contexte où le choix de la meilleure méthode dépend de la disponibilité et de la qualité des données, un tel outil d'aide à la décision est souhaitable. De plus, bien connaître les performances des algorithmes peut contribuer à identifier leurs faiblesses, ce qui permettrait aux chercheurs de cibler leurs recherches sur ces aspects en particulier.

La présente recherche se propose d'étudier la faisabilité de la troisième approche, soit d'utiliser un

banc de filtres inspiré de la prédiction des performances pour améliorer la qualité de cartes de contours. Notre démarche est de concevoir, développer et valider un tel algorithme, basé sur l'utilisation d'un réseau de neurones.

1.1 Rétrospective du problème

Au cours des 40 dernières années, beaucoup d'algorithmes de détection de contours ont fait l'objet d'articles dans les périodiques de vision par ordinateur. Beaucoup de progrès ont été réalisés depuis le travail de Roberts (1963).

La méthodologie de l'évaluation des performances des algorithmes de détection de contours a été étudiée dans la littérature. Pour une rétrospective, voir (Heath *et al.*, 1997). Cependant, ces techniques formelles de mesure ne sont pas encore systématiquement utilisées. Les chercheurs ont souvent limité leurs évaluations au calcul d'indicateurs empiriques, à des analyses théoriques ou à des essais sur des images synthétiques contenant divers niveaux de bruit. Ces méthodologies permettent de caractériser les propriétés géométriques de contours et la sensibilité des détecteurs aux variations du niveau de bruit (Kanungo *et al.*, 1995). Bien que ces méthodes permettent de concevoir des détecteurs de contours et de démontrer leur stabilité, elles ne peuvent suffire à vérifier objectivement les performances d'un détecteur dans des conditions réelles. Souvent, le résultat d'un algorithme sur une image réelle est présenté à la fin d'un article, pour permettre au lecteur d'en faire l'évaluation subjective.

Pour évaluer plus objectivement le résultat d'un algorithme sur une image réelle, il est possible d'utiliser une image de vérité-terrain des contours. Il existe depuis une trentaine d'années des méthodes permettant d'utiliser une telle image pour évaluer un détecteur, comme dans les travaux de Abdou et Pratt (1979). Ces méthodes sont longtemps restées peu utilisées, ceci étant dû au fait que la création manuelle d'une carte de contours est une tâche difficile.

Pour remédier à cette difficulté, des jeux de données des images de vérité-terrain ont été rendus publics, pour faire l'évaluation d'algorithmes de segmentation par région et de détection de contours (Heath *et al.*, 1997; Bowyer *et al.*, 2001; Martin *et al.* 2001; Grigorescu *et al.*, 2003).

Ces ressources permettent une évaluation plus complète des algorithmes. Cependant, vu la grande variété possible des conditions d'acquisition d'images, une question demeure : « quelles seront les

performances de ces algorithmes sur les images que l'utilisateur veut analyser ? » Chalmond *et al.* (2002) ont apporté des éléments de réponse à ce problème en montrant visuellement la capacité d'une banque d'indicateurs à identifier les zones d'ambiguïté et prédire la performance du détecteur Sobel sur une image aérienne infrarouge.

Pour demeurer à jour lors de l'apparition de nouveaux types d'images et de nouveaux algorithmes, un outil informatique capable d'apprentissage pourrait être une solution conviviale. Cet outil devrait pouvoir guider l'utilisateur lors du processus de création d'une carte, soit au moment du choix des données d'entrée et d'algorithmes de bas niveau, soit pour corriger partiellement le résultat d'un algorithme de bas niveau.

1.2 Problématique

Le détecteur de Canny est réputé être celui qui donne « les meilleurs résultats ». Un utilisateur pourrait vouloir utiliser un filtre de Canny bien paramétré. Or, il faut prendre garde à ce type d'approche. Heath *et al.* (1997) ont comparé les performances de plusieurs détecteurs de contours. Ils ont confirmé que, lorsque les paramètres du détecteurs de Canny sont adaptés à chaque image, ce dernier donne les meilleurs résultats parmi les filtres qu'ils ont étudiés. Cependant, ce même détecteur a donné les pires résultats lorsqu'il était utilisé avec les mêmes paramètres pour toutes les images.

Il est donc important, pour l'utilisateur recherchant la meilleure performance, d'investir du temps dans le paramétrage de la détection de contours. Or, le temps étant souvent limité, un utilisateur pressé apprécierait être guidé.

Lors de l'évaluation des performances d'un algorithme, les chercheurs s'assurent :

- qu'il soit suffisamment *sensible* pour détecter les contours et
- qu'il soit suffisamment *spécifique* pour ne pas réagir quand il n'y en a pas (Canny, 1986).

Cela constitue une évaluation fidèle du test. Cependant, à moins que les classes à séparer soient équiprobables, cette information ne décrit pas les performances sur de vraies données.

Les contours étant généralement minces et espacés les uns des autres, leur *prévalence* est peu élevée. Autrement dit, des régions sans contours occupent la majeure partie de l'image. Il est

possible, dans ces conditions, qu'un détecteur suppose, à tort, que plusieurs pixels contiennent des contours alors qu'en réalité, ces pixels appartiennent à des régions homogènes, sans contours.

Malgré les nombreuses erreurs du détecteur, le nombre de faux contours sera quand même largement inférieur à celui des pixels sans contours identifiés correctement. Ainsi, l'algorithme sera considéré comme étant spécifique. Dans ce scénario, pour souligner le fait que le nombre de pixels de faux contours est élevé par rapport au nombre pixels de vrais contours, l'algorithme sera qualifié d'imprécis.

Le degré de *confiance* qu'un usager accorde à une carte de contours dépend de la fiabilité des décisions utilisées pour la créer. Les décisions positives doivent correspondre à de vrais contours de l'image et les décisions négatives, à des zones sans contours. L'estimation de ces deux propriétés complémentaires se nomme *prédiction des performances*. En se basant sur les propriétés du voisinage d'un contour, il est possible, jusqu'à un certain point, de quantifier l'influence :

- de la structure et du type des contours;
- de leur contraste et leur dispersion;
- des propriétés de l'image telles que le niveau de bruit et la texture.

Il deviendrait alors possible d'étudier les écarts de performances attendues entre des images, des algorithmes, ou les régions d'une image. Notre problématique est de savoir s'il est possible de diminuer cette perte de temps associée au choix d'algorithme et de paramètres, en corrigeant automatiquement la carte. Nous espérons que cette thèse amène les utilisateurs à s'intéresser un peu plus au choix de paramètres et les chercheurs à s'intéresser aux méthodologies d'évaluation.

1.3 Objectifs de recherche

Cette thèse vise à mettre au point un algorithme/filtre d'amélioration de carte de contours. Ce dernier utilisera une banque d'indicateurs de « prédiction des performances » pour filtrer/atténuer les fausses alarmes avant le seuillage d'une carte de contours. Dans la présente approche, nous ajoutons un filtrage supplémentaire avant l'étape de seuillage.

La présente étude a donc pour objectifs spécifiques :

- de concevoir un procédé visant à utiliser un réseau de neurones pour améliorer les décisions de détecteurs de contours, en réduisant le nombre de pixels de fausses alarmes et de contours manqués;
- de développer un logiciel permettant d'appliquer ce procédé d'amélioration contours;
- de comparer l'efficacité de l'outil développé à celle d'autres méthodes de détection de contours;
- de montrer le potentiel de la méthode proposée pour améliorer la qualité des détecteurs de contours sur des images de télédétection et sur des images numériques en général;
- de mieux comprendre les facteurs de succès de la restauration de contours.

1.4 Hypothèses de recherche

Les hypothèses de recherche sont :

- la banque d'indicateurs de Chalmond, ou une banque d'indicateurs dérivés de ces derniers, permet de cartographier la présence ou l'absence de contours;
- les cartes de présence de contours permettent d'effectuer un traitement pouvant corriger les résultats fournis par un détecteur de contours conventionnel;
- le traitement obtenu est robuste au bruit;
- ce traitement est applicable à plusieurs types d'images;
- tous les indicateurs utilisés contribuent à l'indice de présence de contours. Ils sont tous nécessaires pour assurer la qualité du résultat.

1.5 Délimitation de l'étude

Au niveau de la chaîne de traitements, notre contribution porte sur la segmentation par détection de contours. Ce travail permet le raffinement d'une carte de contours utilisable par les algorithmes de classification qui doivent suivre. Nous ne tentons pas de prédire les performances des opérations de recalage et de filtrage pouvant précéder la segmentation.

Au sein de la détection de contours, trois étapes peuvent être considérées : le prétraitement de l'image en niveau de gris, le seuillage et les post-traitements de l'image binaire. Notre méthode ne

concerne ni le seuillage, ni le post-traitement. Elle est plutôt un prétraitement, un filtrage supplémentaire de la carte de contours flous avant le seuillage.

Étant donné la rareté des jeux d'images avec réalité du terrain pour les contours, nous n'avons touché que les domaines de bâti non dense sur des images IKONOS, des photographies aériennes d'une base militaire et de la reconnaissance au sol d'objets tirés de la vie quotidienne.

1.6 Plan de la thèse

Cette thèse est composée de sept chapitres. Après l'introduction, il est important de préparer convenablement cette recherche, comprendre les concepts impliqués dans la présente étude, mieux cibler la problématique et faciliter la compréhension des étapes méthodologique. Pour cela, le deuxième chapitre sera consacré à une revue de la littérature portant sur la détection de contours et la caractérisation des performances.

Le troisième chapitre décrit les données employées et explique les étapes méthodologiques de la thèse. Les résultats expérimentaux sont ensuite présentés dans le quatrième chapitre. L'interprétation est faite au cinquième chapitre. La discussion, qui occupe le chapitre six, situe les résultats dans un contexte plus large. Une conclusion et des recommandations viennent ensuite clore la thèse.

2. Cadre théorique

Dans ce chapitre, nous passerons en revue les concepts sur lesquels s'appuie la présente recherche. D'abord, pour la détection de contours, nous présentons des définitions, un historique, ainsi que les principales étapes de mise en oeuvre généralement employées. Nous traiterons ensuite de la *caractérisation des performances*. Pour ce faire, une section sera dédiée aux techniques d'*évaluation*, et une autre, à la *prédiction* des performances. Des documents supplémentaires sont répertoriés dans l'Annexe 1. Le chapitre se termine avec la description des considérations et prototypes de sélection d'images ayant menés à la réalisation de cette thèse.

2.1 Détection de contours

La *détection de contours* peut être utilisée comme étape préliminaire à l'identification des objets d'une scène et des relations spatiales entre ces objets. Un *contour* est une variation brusque des niveaux de gris, de la couleur ou de la texture de l'image. Les contours peuvent être situés à la frontière entre un objet et le fond de l'image ou entre deux objets qui se chevauchent. Ils peuvent aussi être causés par d'autres facteurs tels que l'ombre, des changements brusques dans l'orientation d'une surface, des changements dans les propriétés de réflectance, ou des variations d'illumination (Manjunath et Chellapa, 1993; Elder et Zucker, 1998; Nalwa et Binford, 1986).

Ces changements d'intensité sont une source riche d'information. Mathématiquement, des points dont la $n^{\text{ème}}$ dérivée dans une direction donnée est une fonction de Dirac sont des *discontinuités d'ordre n* . Les contours forment une représentation simplifiée des objets, se limitant à l'essentiel, sans donner de détails. La localisation des contours permet de déterminer la bordure des objets et de séparer ces objets en zones ayant des propriétés photométriques ou géométriques significativement distinctes. Ces contours peuvent être, par exemple, le pourtour d'une ville, la lisière d'une forêt, le rebord d'une excavation.

La représentation informatique de l'image étant une grille, il faut séparer les contours en fragments orientés, d'une longueur d'un pixel. Ces éléments de contours sont appelés *edgels* (Nalwa et Binford, 1986; Martin *et al.*, 2004). Sur l'image, les pixels de contours utiles à l'interprétation sont regroupés en structures linéaires d'un ou deux pixels de large. Nous appelons *courbe* un groupe de pixels de contours connexes. Le terme *trait* désigne un petit segment

rectiligne de contour. Les contours sont souvent séparés en quatre types : *marche/rampe*, *ligne*, *toit* et *jonction/coin*.

Les contours de type « marches » (*step edge*, en anglais) sont les plus communs. Ils se retrouvent entre deux objets ayant des niveaux de gris constants, mais différents l'un de l'autre. De profil, leur fonction d'intensité fait penser à une marche d'escalier. Ce sont des discontinuités d'ordre un. La majorité des algorithmes de détection de contours sont adaptés à ce type de contours. Si la démarcation est un plan incliné plutôt que parfaitement vertical, la discontinuité est parfois appelée *contour de type « rampe »*. Un tel contour peut être décrit par deux paramètres, la hauteur h et la pente θ (Pratt, 1991). Un détecteur devrait identifier un contour avec une largeur d'un pixel, au centre de la rampe (Parker, 1997).

Pour produire un *contour de type « ligne »*, un objet photographié doit être de résolution plus fine ou égale à celle du capteur. Son étendue se réduit pratiquement à la seule dimension de la longueur. Cela crée une courbe d'une largeur d'un ou deux pixels. Ce sont des discontinuités d'ordre zéro. Les contours de type « lignes » peuvent avoir plusieurs causes physiques, comme des égratignures ou du texte imprimé. Le contraste par rapport au voisinage permet de localiser un contour de type « ligne ». Si le contour est clair, il est appelé *contour de type « crête »* et s'il est sombre, *contour de type « vallée »*. Les contours de type « ligne » sont souvent trop étroits pour être détectés efficacement par les détecteurs de contours de type « marches » (Canny, 1986). Ces derniers sont portés à marquer faussement deux contours de type « marche », un de chaque côté du vrai contour de type « ligne » (Ziou et Tabbone, 1998). Les *contours de type « toit »*, moins étudiés par les chercheurs, sont des discontinuités d'ordre deux (Nalwa et Binford, 1986).

Les points de rencontre de plusieurs contours sont parfois appelés *jonctions* ou *coins*. Les détecteurs de coins peuvent servir à déterminer les points « intéressants » d'une image, utilisables pour la mise en correspondance avec d'autres images au contenu semblable (Lowe, 2004). Il y a des applications, notamment, en stéréoscopie, en reconstruction et en détermination de la structure à partir du mouvement (Forsyth et Ponce, 2003). Il existe plusieurs modèles de coins : en « L », en « T », en « V », en « X », en « Y », etc. (Ziou et Tabbone, 1998). Plusieurs détecteurs de contours échouent à ces endroits, où l'amplitude du gradient est plus faible et l'orientation, plus

difficile à estimer (Rothwell *et al.*, 1994). Selon le contexte, le mot contour, employé seul, désigne un contour de type « marche » ou bien un contour en général, peu importe son type.

Bien que les détails d'un objet soient riches en information, il est souvent utile de ne tracer que les frontières entre les objets. On appelle *profil*, ou encore *silhouette*, une courbe fermée, de largeur d'un pixel, représentant la limite entre un objet et le reste de l'image. La *bordure* est considérée comme faisant partie de l'objet, contrairement à l'ombre ou la surbrillance (Sun Microsystems, 2001). C'est une *bordure intérieure* si la ligne est superposée à l'objet. Sinon, c'est une *bordure extérieure*. Le mot anglais “*contour*” peut vouloir dire contour fermé, qu'il corresponde ou non à une frontière d'objet. On rencontre aussi “*closed contour*”.

Les contours ne sont cependant pas toujours fermés. Dans ce cas, ils sont composés de *fragments de contours* (Wang *et al.*, 2005). Ils peuvent être des courbes ouvertes si :

- le contraste estimé entre deux régions est faible et fluctue autour du seuil de détection de l'algorithme (Canny, 1986; Salotti *et al.*, 1996),
- un objet est formé de contours de type « lignes » (soit il est mince par rapport à la résolution du capteur, soit il contient beaucoup de détails de texture, appelés *hachures*),
- une partie d'un objet cache une autre partie du même objet,
- une partie de la frontière entre deux objets est rendue floue par l'ombre ou la lumière (Martin *et al.*, 2004).

Après la détection de contours, ces lignes sont à l'état d'ébauche. Dans une carte de contours, si les frontières des objets sont fragmentées, ou si la carte contient de petits faux contours dus au bruit, la segmentation est dite *incomplète*. Les contours doivent alors être nettoyés, amincis et complétés. Une fois les défauts corrigés, la carte devient une segmentation complète de l'image, par contours. Comme cette segmentation définit un partitionnement de l'image, elle est la représentation duale d'une *segmentation par région*. Dans ce contexte, on peut appeler *élément d'arête* un pixel situé à la frontière de deux régions différentes selon une propriété donnée, particulièrement si ces régions sont deux surfaces géométriques distinctes.

Une *carte de contours* est la représentation graphique des variations brusques des niveaux de gris ou de la texture d'une image. Cette représentation peut prendre plusieurs formes. La plus simple

consiste à marquer comme contours tous les pixels de l'image dans laquelle la variation des niveaux de gris est grande. Cela conduit souvent à des contours larges de deux pixels. Une technique complémentaire consiste à amincir systématiquement les contours à un pixel de large, localisant ainsi le côté des contours où l'intensité varie le plus.

Une dernière représentation ressemble beaucoup aux ébauches d'icônes préparées pour les interfaces graphiques. Ces ébauches sont constituées de lignes noires, pour assurer un affichage satisfaisant sur une grande gamme de couleurs de fond et de textures. Leur style est simplifié, facilitant ainsi l'interprétation du dessin. Les lignes sont limitées aux caractéristiques principales de l'objet représenté, notamment sa bordure intérieure. Ces lignes ont une largeur d'un pixel. Pour créer un effet de relief, de la surbrillance blanche est ajoutée. L'ombre noire est fournie par l'ébauche du graphique (Sun Microsystems, 2001). Cela ressemble à la convention utilisée par Yitzhaky et Peli (2003) pour encoder les contours : ils marquent systématiquement le côté sombre des contours en noir et le côté clair des contours en blanc, sur un fond gris. Une telle carte de contours contient une information équivalente à la phase du gradient.

Il est important de se rappeler qu'un contour théorique est de largeur nulle. Si les objets se terminaient aux frontières des pixels, ces contours théoriques passeraient entre les pixels. Cependant, il y aura toujours du flou ou du bruit dans l'image. Cette incertitude sur l'emplacement exact des contours est due à plusieurs facteurs :

- l'emplacement physique des objets n'est pas dicté par la grille de l'image;
- le capteur parfait n'existe pas;
- les scènes photographiées ont toutes une profondeur;
- la pénombre diffuse borde l'ombre (Elder et Zucker, 1998).

Le reste de cette section décrit plus en détail les étapes importantes de l'évolution historique des détecteurs de contours. Les sections 2.1.1 et 2.1.2 décrivent les fondements théoriques des techniques basées sur les dérivées première et seconde. Dans la section 2.1.3, des techniques d'appariement de formes sont décrites. La section 2.1.4 complète les précédentes en décrivant des techniques diverses, plus récentes. La section 2.1.5 décrit plus en détails la théorie, en situant la détection de contours au sein d'une chaîne complète de traitements de l'image.

2.1.1 Détection basée sur la dérivée première

Un des détecteurs de contours ayant le plus marqué le début de l'ère de la recherche en vision numérique est la croix de Roberts (1963). Son calcul commence par la détermination du gradient, via la convolution de l'image avec deux noyaux de dimension 2x2 (Figure 2.1). Il existe ensuite deux façons de compléter le calcul. La première utilise l'amplitude du gradient, c'est-à-dire la racine de la somme des carrés de ses composantes. La deuxième utilise une approximation : la somme des valeurs absolues des deux composantes du gradient. Dans les deux cas, les contours sont obtenus par un seuillage sur l'amplitude calculée.

L'opérateur de Roberts est intéressant pour sa rapidité de calcul, due à la petite taille des noyaux utilisés et à l'approximation de l'amplitude du gradient. Ce détecteur produit des contours fins, mais est sensible au bruit. Puisque les fenêtres coulissantes utilisées sont de taille paire, leur centre est entre 4 pixels. Le résultat est généralement décalé d'un demi pixel vers le haut et vers la gauche. La position des contours détectés est donc systématiquement biaisée.

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Figure 2.1 – Filtres de Roberts

En 1970, Prewitt (Figure 2.2) et Sobel (Figure 2.3) ont chacun proposé des filtres horizontaux et verticaux de dimension 3x3 (Umbaugh, 1998). Le filtre de Sobel donnant plus de poids aux pixels centraux, il est possible qu'il favorise la bonne localisation des contours, au prix d'une moins bonne élimination des discontinuités. Pour estimer le gradient de l'image, les filtres de Prewitt ou de Sobel sont convolués avec cette dernière. Les contours sont ensuite marqués par un seuillage sur l'amplitude du gradient. L'utilisation d'un noyau de dimension impaire a l'avantage de ne pas décaler la position des contours. Les trois coefficients de chaque côté des masques calculent en quelque sorte l'intensité moyenne du signal de part et d'autre du pixel central : le bruit est ainsi légèrement réduit, au prix d'une augmentation de l'incertitude sur la position.

Il est possible d'élargir les filtres de Prewitt et Sobel à des tailles de 5x5, 7x7, 9x9... Ces détecteurs permettent de contrôler le degré de filtrage. Le positionnement des contours est cependant affecté par la taille des plateaux rectangulaires entourant le pixel central. En présence

de contours bruités, ces détecteurs par « différence de boîtes » marquent plusieurs maximums locaux qu'il est difficile de séparer (Canny, 1986; Umbaugh, 1998).

Les détecteurs décrits plus haut considèrent les contours comme des maximums dans la dérivée première d'un signal. Si la réponse de l'opérateur le long d'un contour varie de part et d'autre du seuil, les contours identifiés seront fragmentés (Canny, 1986).

$$\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 0 & 0 & 0 \\ -1/3 & -1/3 & -1/3 \end{bmatrix} \quad \begin{bmatrix} 1/3 & 0 & -1/3 \\ 1/3 & 0 & -1/3 \\ 1/3 & 0 & -1/3 \end{bmatrix}$$

Figure 2.2 – Filtres de Prewitt

$$\begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 0 & 0 & 0 \\ -1/4 & -1/2 & -1/4 \end{bmatrix} \quad \begin{bmatrix} 1/4 & 0 & -1/4 \\ 1/2 & 0 & -1/2 \\ 1/4 & 0 & -1/4 \end{bmatrix}$$

Figure 2.3 – Filtres de Sobel

Bien que certaines innovations soient comparées au détecteur de Sobel, la majorité des innovations sont comparées au détecteur de Canny. Ce dernier est probablement le détecteur de contours le plus connu. Canny (1986) propose trois critères pour caractériser les performances d'un détecteur de contours :

- 1) *Bonne détection. Il devrait y avoir une faible probabilité de ne pas identifier un vrai contour, ainsi qu'une faible probabilité de marquer faussement un endroit sans contour. Comme ces deux probabilités sont des fonctions décroissantes monotones du rapport signal sur bruit du signal de sortie, ce critère équivaut à maximiser le rapport signal sur bruit.*
- 2) *Bonne localisation. Les points marqués comme contours par l'opérateur devraient être aussi près que possible du centre du vrai contour.*
- 3) *Une seule réponse par contour. Cela est implicite dans le premier critère; en effet, quand il y a deux réponses au même contour, l'une d'elles doit être considérée fausse. Cependant, la formulation mathématique du premier critère ne capture pas l'exigence d'un contour unique et cette exigence est rendue explicite.*

Traduit de : Canny (1986)

Selon l'auteur, le filtre pour isoler les contours de type « marche » de manière « optimale » peut être approximé par la dérivée première d'une gaussienne. Marr et Hildreth (1980) affirment aussi que le filtre gaussien est optimal, car il offre à la fois une bonne localisation spatiale et fréquentielle. Ce filtre avait été proposé plus tôt par Argyle (1971). Canny propose un détecteur

qui élimine d'abord les pixels ayant une valeur inférieure à celle de leurs voisins dans la direction du gradient. Après cette suppression des non-maximums, il suggère l'utilisation d'un seuillage double. Le premier seuillage consiste à éliminer les pixels dont l'amplitude du gradient est inférieure à une certaine valeur. Le deuxième seuillage concerne les courbes. Quand l'amplitude du gradient de chacun des pixels d'une courbe est inférieure à un deuxième seuil, défini comme la moitié du premier seuil, ce groupe de pixels est éliminé.

D'autres critères d'optimalité sont possible, selon le modèle de contours ou de bruit considéré. Le problème est souvent traité à une dimension (1D), puis étendu à deux dimensions (2D) par symétrie de révolution (Forsyth et Ponce, 2003). Le calcul de la dérivée est souvent fait après convolution par ce filtre isotrope. Un tel critère est la capacité à séparer les contours rapprochés (Nalwa et Binford, 1986).

Un autre critère d'optimalité est le rapport signal sur bruit « discriminant » qui considère toute réponse hors-centre comme du bruit (Ben-Arie et Rao, 1993). Cela revient à réduire le bruit tout en minimisant la différence entre l'image originale et l'image filtrée. Le filtre exponentiel est optimal pour cette tâche. Il est défini par l'équation :

$$g(x) = 0,5 \sigma \exp(-\sigma |x|) \quad (2.1)$$

où σ est la dispersion et x est la distance radiale. Le type de contours ainsi détecté a d'abord été nommé *contour de type « rampe »*, pour le distinguer des contours de type « marche » « parfaits » (Petrou et Kittler, 1991). Cet opérateur produirait de bonnes décisions, bien localisées (Moon *et al.*, 2002). Il permettrait aussi de conserver au maximum la résolution spatiale d'une image radar quand la réduction du bruit de chatoiement exige l'utilisation d'un noyau de convolution assez large pour englober plus d'un contour (Fjørtoft *et al.*, 1998).

2.1.2 Détection basée sur la dérivée seconde

Le laplacien ∇^2 d'un signal 2D est défini par l'équation :

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad (2.2)$$

Il a la particularité de changer de signe algébrique en présence d'un contour. Ses passages par zéro suivent des courbes fermées (Ziou et Tabbone, 1998). Le masque de convolution du

laplacien est donné dans la Figure 2.4. Comme le laplacien n'est pas directionnel, il y a diminution du rapport signal sur bruit, particulièrement au voisinage des jonctions (Smith et Brady, 1995; Rothwell *et al.*, 1994).

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Figure 2.4 – Masque laplacien

D'après Marr et Hildreth (1980), ainsi que Kass *et al.* (1988), un détecteur plus robuste au bruit est obtenu en définissant les contours comme étant les passages par zéro du laplacien d'un signal filtré par une gaussienne

$$G_{\sigma} * \nabla^2 I \quad (2.3)$$

où G_{σ} est un filtre gaussien, $*$ est l'opérateur de convolution et I est l'image originale. Une autre approche à la détection des contours est l'utilisation d'une approximation du laplacien de la gaussienne, soit la différence de gaussiennes ayant leur paramètre σ dans un rapport de 1,6 (Marr et Hildreth, 1980).

En présence de bruit, le laplacien du gaussien marque plus de points de contours que les détecteurs basés sur l'amplitude du gradient (Ziou et Tabbone, 1998). Si l'échelle de la détection est fine, c'est-à-dire si le lissage gaussien est léger, beaucoup de contours sont marqués (Lindeburg, 1996).

Selon l'application visée, un tel résultat peut ne pas être directement utilisable, particulièrement si les contours détectés sont très denses. L'opération peut alors être interprétée comme une première étape d'élimination de pixels sans contour. La carte des contours peut être raffinée par un détecteur basé sur le gradient. Son utilisation peut alors être limitée aux pixels n'ayant pas été rejetés auparavant. Ce type d'approche est parfois appelé classification par rejet de formes, pour la différencier de la reconnaissance de formes traditionnelle (Baker et Nayar, 1996). Il serait possible, pour faire un parallèle avec la médecine, de parler de test de dépistage de formes.

Poggio (1982) a suggéré, et Haralick (1984) a développé, un détecteur basé sur la dérivée seconde, orientée dans la direction du gradient :

$$\frac{\partial^2}{\partial x^2} \cos^2(\theta) + \frac{\partial^2}{\partial x \partial y} \sin(\theta) \cos(\theta) + \frac{\partial^2}{\partial y^2} \sin^2(\theta) \quad (2.4)$$

où x est position horizontale, y est la position verticale et θ est la direction du gradient. Freeman et Adelson (1991) ont éventuellement démontré formellement que les dérivées d'une image dans une direction quelconque peuvent être calculées par une somme pondérée de dérivées directionnelles dans des directions particulières.

2.1.3 Appariement de formes

Les approches décrites plus haut tentent de localiser les contours via les dérivées de l'image. Une autre approche de détection de contours est l'appariement de formes. Elle consiste à comparer une partie d'image à la forme d'un contour idéal, paramétré selon divers critères tels que son orientation, la différence d'intensité, la largeur... Hueckel (1973, 1974) décrit un contour comme une fonction à six paramètres, définie sur un domaine circulaire d'un rayon de $\sqrt{20}$ pixels compris dans un voisinage de 9x9. Le profil d'intensité de la pastille d'image est projeté dans le sous-espace de contours de type marche et de type ligne défini par la fonction. Les paramètres sont ajustés pour minimiser l'erreur quadratique entre l'image et la fonction. La partie du signal qui reste est assimilée au bruit. Si le bruit domine le signal de contours, l'opérateur signale l'échec de la reconnaissance. Sinon, un contour est reconnu avec les paramètres qui lui sont associés.

Kirsch, en 1971, et Robinson, en 1977, ont défini chacun huit filtres de taille 3x3 pour décrire la forme de contours de type « marche » (Umbaugh, 1998). Ces filtres correspondent aux huit directions d'une boussole (N, NE, E, SE, S, SO, O, NO). L'ensemble de filtres peut être produit en prenant un filtre de départ et en effectuant la rotation de ses coefficients par multiples de 45° (Figure 2.5). L'amplitude des contours est donnée par la valeur maximale obtenue en convoluant les huit masques avec l'image. La phase est donnée par l'orientation du masque ayant donné la valeur maximale (Klette et Zamperoni, 1994). Frei et Chen (1977) décomposent aussi le signal en fonctions de base : quatre fonctions pour les contours de type « marches » (Figure 2.6), quatre fonctions pour les contours de type « ligne » (Figure 2.7), une fonction pour la moyenne ou un signal plat (Figure 2.8).

$$\begin{bmatrix} -3 & -3 & +5 \\ -3 & 0 & +5 \\ -3 & -3 & +5 \end{bmatrix} \quad \begin{bmatrix} -3 & +5 & +5 \\ -3 & 0 & +5 \\ -3 & -3 & -3 \end{bmatrix}$$

Figure 2.5 – Filtres de Kirsch à 0° et 45°

$$\frac{1}{2+\sqrt{2}} \begin{bmatrix} +1 & +\sqrt{2} & +1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} \quad \frac{1}{2+\sqrt{2}} \begin{bmatrix} +1 & 0 & -1 \\ +\sqrt{2} & 0 & -\sqrt{2} \\ +1 & 0 & -1 \end{bmatrix}$$

$$\frac{1}{2+\sqrt{2}} \begin{bmatrix} 0 & -1 & +\sqrt{2} \\ +1 & 0 & -1 \\ -\sqrt{2} & +1 & 0 \end{bmatrix} \quad \frac{1}{2+\sqrt{2}} \begin{bmatrix} +\sqrt{2} & -1 & 0 \\ -1 & 0 & +1 \\ 0 & +1 & -\sqrt{2} \end{bmatrix}$$

Figure 2.6 – Filtres « contours » de Frei et Chen

$$\frac{1}{2} \begin{bmatrix} 0 & +1 & 0 \\ -1 & 0 & -1 \\ 0 & +1 & 0 \end{bmatrix} \quad \frac{1}{2} \begin{bmatrix} -1 & 0 & +1 \\ 0 & 0 & 0 \\ +1 & 0 & -1 \end{bmatrix} \quad \frac{1}{8} \begin{bmatrix} +1 & -2 & +1 \\ -2 & +4 & -2 \\ +1 & -2 & +1 \end{bmatrix} \quad \frac{1}{8} \begin{bmatrix} -2 & +1 & -2 \\ +1 & +4 & +1 \\ -2 & +1 & -2 \end{bmatrix}$$

Figure 2.7 – Filtres « lignes » de Frei et Chen

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Figure 2.8 – Filtre « moyenne » de Frei et Chen

2.1.4 Techniques diverses

La détection de contours peut être effectuée selon l'énergie locale dans le domaine fréquentiel. L'énergie est calculée à partir de la convolution de l'image par des couples de filtres en quadrature. Ces filtres symétriques et anti-symétriques, obtenus par le calcul de la transformée d'Hilbert, permettant d'extraire des informations complémentaires sur les contours, sans nécessairement aboutir à des dédoublements comme ceux qui pourraient être rencontrés lors de la combinaison des résultats d'un détecteur de marches et d'un détecteur de lignes (Ziou et Tabbone, 1998). Ces approches seraient particulièrement efficaces lorsque combinées à des indicateurs de texture (Martin *et al.*, 2004; Konishi *et al.*, 2003).

Plusieurs détecteurs de contours procèdent en combinant des détections obtenues à différentes

échelles, dans le but d'identifier les contours qui demeurent quand le niveau de détails de l'image varie (Rosenfeld, 1970; Bergholm, 1987). La plupart du temps, ces *détecteurs multi-échelles* se basent sur une série d'images, plus ou moins filtrées par une gaussienne. La dérivée première (Canny, 1986; Lindeburg, 1996), le laplacien (Marr et Hildreth, 1980), ou son approximation par une différence de gaussiennes (Lowe, 2004), sont utilisés pour identifier les discontinuités significatives. Ensuite, un seuillage adaptatif peut être fait, par exemple en comparant l'amplitude du gradient avec la somme de la moyenne et de l'écart-type des variations du voisinage d'un pixel (Salotti *et al.*, 1996). De leur côté, Elder et Zucker (1998) proposent un critère pour déterminer l'échelle minimale pour la dérivée seconde dans la direction du gradient, dans le but de limiter à 5 % de l'image les erreurs de détection dues au bruit du capteur.

Un *contour actif* est un modèle minimisant une fonction d'énergie définie sur le domaine de l'image. Quand cette fonction est bien choisie, le modèle sera attiré par les contours. Un contour actif recherche la meilleure solution dans un voisinage spatial, favorisant ainsi la bonne localisation des contours. C'est donc un outil qui permet de faire converger une courbe approximative vers une meilleure solution. Il peut être autonome ou guidé interactivement, notamment par le choix de points initiaux appropriés (Kass *et al.*, 1988). Certains snakes géométriques, par exemple ceux basés sur les courbes de niveaux, sont robustes au bruit et peuvent donc être utilisés sur des images radar. Il est alors possible de détecter plusieurs types d'entités, notamment les bateaux et les littoraux (Lesage et Gagnon, 2000).

Smith et Brady (1995) utilisent l'inverse du nombre de pixels d'intensité semblable à celle du pixel au centre d'une pastille; ils peuvent ainsi détecter les contours et les coins, avec une bonne capacité de localisation. Leur calcul est fait à l'intérieur d'un masque 7x7, sur une pastille de rayon $\sqrt{10}$.

D'autres techniques de détection de contours ont été étudiées, notamment les méthodes par réseaux de neurones (Manjunath et Chellapa, 1993), la recherche de l'orientation qui maximise la différence entre la distribution des pixels dans les deux moitiés d'un masque circulaire (Ruzon et Tomasi, 2001), la descente du gradient dans un espace combinant la grille spatiale et l'intensité des pixels (Comaniciu et Meer, 2002), l'apprentissage d'un modèle de contours (Konishi *et al.*, 2003).

2.1.5 Base théorique

Ce qui se dégage de l'historique présenté plus haut, c'est que les techniques de détection de contours les plus utilisées se basent sur la dérivée première de l'image. Les diverses méthodes de cette approche de détection de contours ont des étapes en commun. L'obtention d'une carte de contours nécessite toujours plusieurs étapes, les principales étant la différentiation et le seuillage de l'image. Le plus souvent, un filtrage précède ces deux opérations. Une fois les contours identifiés, certains détecteurs plus sophistiqués peuvent aussi amincir et parfois rallonger les contours afin d'obtenir des courbes fermées. Enfin, pour sauver du temps, certains détecteurs choisissent de ne pas traiter les zones homogènes de l'image, ces zones ayant une faible probabilité de contenir des contours.

2.1.5.A Délimitation de la région d'intérêt

Il serait inutile d'appliquer un détecteur à une zone exempte de contours. Il est donc souhaitable de vérifier sommairement si la région d'intérêt peut en contenir. Un opérateur visant à éliminer les zones non pertinentes est appelé *rejeteur de formes*. Son rôle peut être comparé à celui d'un test de dépistage médical.

Pour des raisons d'efficacité algorithmique, il n'est pas demandé au rejeteur d'éliminer toutes les zones sans contours. Il lui serait facile d'éliminer, par exemple, les zones de l'image où l'intensité est stationnaire. Le rejeteur doit minimiser le nombre de fausses éliminations, laissant au détecteur la responsabilité de classifier les zones complexes. Suite à ce prétraitement, il ne restera que les zones méritant d'être étudiées. Dans certains cas, la taille de ces zones sera petite par rapport à celle de l'image. Ainsi, la charge de travail du détecteur sera moins grande. Il pourra alors focaliser ses ressources sur les décisions plus difficiles (Baker et Nayar, 1996). Le rejet de formes ne doit pas être confondu avec l'élimination de formes, qui est un modèle de création de classifieur robuste au manque de données (Globerson et Roweis, 2006).

Un détecteur de contours donné peut ne pas être compatible aux conditions de l'image entière. Cibler les zones où il est apte à prendre des décisions correctes est un moyen de réduire le nombre de détecteurs à essayer avant de parvenir à une carte de contours satisfaisante. Cela permet aussi d'assurer une meilleure fiabilité au résultat final. Il ne serait pas approprié d'appliquer à une zone importante de l'image un détecteur qui a souvent tort quand il n'y voit pas

de contours. Il ne faudrait pas non plus se fier aux résultats d'un détecteur si, dans une région donnée, les contours tels qu'il les voit ne correspondent pas à de vrais contours. L'identification et l'élimination de zones mal adaptées à un détecteur peut se faire après la *prédiction des performances* du détecteur sur l'image. Nous décrivons la théorie de la prédiction des performances dans la section 2.3.

2.1.5.B Filtrage

En vision numérique, le mot *bruit* désigne une information, contenue dans l'image, que nous ne voulons pas utiliser, ou que nous ne savons pas utiliser (Forsyth et Ponce, 2003). En vue de mesurer les caractéristiques pertinentes, telles que l'amplitude du gradient de l'intensité, pour détecter les contours, il faut s'assurer que des effets non pertinents ne dégradent pas les mesures. Si un algorithme tente naïvement de situer les discontinuités d'intensité sur une image moyennement ou fortement bruitée, il identifiera à tort beaucoup de faux contours dus au bruit. Il est donc nécessaire de commencer par réduire le niveau de bruit contenu dans l'image (Torre et Poggio, 1984). Ainsi, les pixels bruités isolés influenceront peu l'image lissée. Cependant, comme le filtrage rend généralement l'image plus floue, il y a un compromis à faire entre la préservation de l'information et l'élimination du bruit (Moon *et al.*, 2002; Canny, 1986).

Pour une performance optimale, il est nécessaire d'employer un filtre adapté au type de bruit contenu dans l'image. La plus simple façon de déterminer le type de bruit et d'obtenir une estimation des paramètres du bruit est d'étudier les propriétés d'une zone homogène de l'image (Meer *et al.*, 1990). Les modèles de bruit les plus employés en télédétection sont le bruit additif gaussien pour les images optiques et le bruit multiplicatif gamma pour les images radar. La détection des contours dans une image radar peut être faite en prenant le rapport de la moyenne des valeurs dans les deux zones de part et d'autre d'un pixel central (Borghys *et al.*, 2002; Touzi *et al.*, 1988). Selon l'application, il peut être approprié d'éliminer les contours complexes dus aux textures fines (Martin *et al.*, 2004; Konishi *et al.*, 2003).

Quand un filtre, par exemple le filtre gaussien, effectue un moyennage local de l'image, le résultat est une version plus ou moins floue de l'image. Ainsi, un pixel bruité a peu d'effets sur l'image lissée. La taille du filtre affecte le résultat de l'algorithme. Un petit filtre produira des contours

bien localisés, mais bruités. Un grand filtre éliminera bien le bruit, mais rendra difficile la localisation précise des contours (Canny, 1986).

2.1.5.C Estimation de l'amplitude et de la phase du gradient de l'intensité

Un contour peut être orienté dans plusieurs directions. Les détecteurs de contours emploient plusieurs masques orientés pour détecter les contours. Le gradient de l'image, ∇I , est défini comme un vecteur composé de la dérivée partielle de l'image dans deux directions orthogonales :

$$\nabla I(x, y) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) \quad (2.5)$$

Une image réelle n'est pas une fonction lisse continue; c'est plutôt une fonction discrète, plus ou moins bruitée. La plupart des algorithmes utilisent un masque de convolution pour traiter rapidement l'image. Le gradient est donc approximé par une différence finie (Forsyth et Ponce, 2003). La convolution employée pour faire ce calcul peut inclure un filtrage, pour annuler les effets non pertinents. En effet, deux filtres, w_1 et w_2 , appliqués en séquence à une image I peuvent être remplacés par un seul filtre cumulatif w_{net} , défini comme la convolution des deux filtres individuels :

$$(I * w_1) * w_2 = I * (w_1 * w_2) = I * w_{net} \quad (2.6)$$

où $w_{net} = w_1 * w_2$ (Duda *et al.*, 2001). Il est donc commun de rencontrer un seul filtre pour effectuer le filtrage et le calcul du gradient.

Certaines contraintes permettent de mieux définir cette convolution. D'abord, la réponse du détecteur doit être nulle pour les régions homogènes (Canny, 1986). Il faut donc que :

$$0 = \sum_{i,j} K_{ij} \quad (2.7)$$

où K_{ij} sont les coefficients du masque de convolution.

De plus, afin d'éviter les débordements de capacité, l'intensité maximale de la réponse doit être égale à la valeur maximale permise par le type de donnée de l'image. Pour cela, la somme des coefficients positifs du masque et la somme des coefficients négatifs du masque, doivent toutes deux être de 1 ou moins, ce qui entraîne :

$$2 \leq \sum_{i,j} |K_{ij}| \quad (2.8)$$

Une fois le gradient calculé, son amplitude $\rho(x, y)$ est définie par :

$$\rho(x, y) = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \quad (2.9)$$

Sa phase, autrement son orientation ou sa direction, est donnée par :

$$\theta(x, y) = \arctan\left(\frac{\partial I}{\partial y} \div \frac{\partial I}{\partial x}\right) \quad (2.10)$$

Connaissant l'amplitude et la phase pour un pixel et ses voisins, il est possible, au besoin, de faire une interpolation pour estimer la position exacte de l'élément de contour à l'intérieur du pixel central (Rothwell *et al.*, 1994).

Il existe d'autres façons d'estimer l'amplitude et la phase du gradient. Par exemple, l'amplitude peut être estimée par le nombre de pixels ayant des niveaux de gris suffisamment différents de celui du pixel central d'un voisinage (Smith et Brady, 1995). Les opérateurs « compas » procèdent autrement pour estimer l'amplitude et la phase du gradient. Ils utilisent plusieurs masques, généralement 4 ou 8, pour calculer la dérivée partielle dans plusieurs directions autour du pixel central. Le plus grand résultat est considéré comme l'amplitude du gradient. La direction l'ayant produit est considérée comme la phase (Canny, 1986).

Le poids des lignes dans la carte d'amplitude du gradient peut créer une impression de relief. Sur cette carte, les zones floues donnent une impression d'éloignement et les lignes plus marquées, une impression de proximité. Bien que la variation de l'étalement des contours puisse être causée par une différence de profondeur dans la scène, elle peut aussi être due à des effets d'ombre ou de pénombre (Elder et Zucker, 1998).

Le rehaussement de contour est un traitement d'image visant à faciliter l'interprétation visuelle d'une image en augmentant le contraste entre les contours et le fond de l'image (Parker, 1997). Cela peut être fait, par exemple, en additionnant l'image originale et une image d'amplitude du gradient estimé.

2.1.5.D Seuillage

Une fois la dérivée calculée, l'étape suivante est d'appliquer un seuil, pour décider où les résultats suggèrent la présence de contours. Le résultat du seuillage est une classification binaire dans laquelle chaque pixel est étiqueté comme contour ou non-contour. Les pixels où l'amplitude du gradient dépasse un certain seuil sont étiquetés « contours ». Tous les autres sont considérés comme des pixels non-contours.

Le choix du seuil est problématique. Si le seuil est trop bas, une carte de contours peut contenir les contours des objets intéressants de l'image, mais contenir aussi un grand nombre de contours de texture qui ne sont pas reliés à la forme des objets. Si, pour éliminer la texture, le seuil est trop élevé, une carte de contours peut aussi contenir des contours incomplets. La qualité de l'identification d'objets dépend de ces facteurs (Ghosh et Petkov, 2005).

Il est difficile de déterminer un seuil générique qui fonctionne convenablement sur toutes les images. La distribution de l'amplitude du gradient peut être modélisée, par exemple par une distribution de Weibull (Ravid et Levabib, 1992; Lalonde *et al.*, 2001), ou une mixture de deux Gaussiennes (Xu *et al.*, 2006). Bien qu'aucune technique ne soit universellement acceptée, si une bonne stratégie de sélection de seuil existe, elle implique probablement l'écart-type de l'amplitude du gradient. C'est du moins ce que croient Voorhees et Poggio, tel que mentionné par Hancock et Kittler (1991).

Elder et Zucker (1998) croient que le seuil doit être choisi en fonction du niveau de bruit. Pour un même degré de lissage, le seuil doit être plus élevé quand le niveau de bruit est plus élevé. Le niveau de bruit peut être estimé automatiquement selon les zones homogènes de l'image (Meer *et al.*, 1990).

Le résultat du seuillage est une première forme de la carte de contours. Cette *ébauche*, grossière et encore imparfaite est appelée à être retravaillée. Les détails restent à affiner : il faut appliquer un post-traitement pour éliminer les faux contours dus au bruit, amincir les contours principaux et restaurer les contours incomplets. Les deux paragraphes qui suivent expliquent ces techniques de *restauration de contours*, l'amincissement et la fermeture.

2.1.5.E Amincissement

Après le seuillage, la carte de contours peut être bonifiée par un nettoyage. Pour supprimer les

faux contours dus au bruit, une technique souvent utilisée est de supprimer tous les segments de contours constitués d'un nombre de pixels inférieurs à une limite arbitraire. Par la suite, si les contours restants ont plus d'un pixel de large, ils peuvent être amincis.

Pour amincir les contours, il est possible d'utiliser la technique de suppression des non-maximums (Canny, 1986). Cette technique consiste à conserver seulement les pixels de contours pour lesquels l'amplitude du gradient correspond à un maximum dans la direction du gradient. Comme un pixel a huit voisins, il y a quatre axes de directions possibles.

Pour des raisons d'efficacité informatique, la suppression des non-maximums peut être effectuée avant le seuillage. Le résultat intermédiaire est alors appelé *toile des contours* (Hancock et Kittler, 1991).

D'autres techniques ont été développées pour l'amincissement des contours. Ces techniques tiennent compte du contenu du voisinage des éléments de contours. Par exemple, Stentiford et Mortimer (1983) utilisent des masques d'appariement de forme. Smith (1995) base plutôt son calcul sur le nombre de pixels de contours dans ce voisinage. Grigorescu *et al.* (2003) proposent d'atténuer la réponse centrale d'un détecteur de contours en fonction de sa réponse hors de sa surface de support. Cette technique, inspirée de la biologie, se nomme inhibition non classique d'unité sensorielle.

2.1.5.F Fermeture

Pour la reconnaissance de formes, il n'est pas forcément nécessaire que les contours soient complets (Moon *et al.*, 2002). Cependant, il est fortement souhaitable qu'ils soient le plus complet possible. Étant donné les imperfections des algorithmes actuels de détection d'éléments de contours, il existe un intérêt pour le développement d'algorithmes de restauration de contours, notamment la fermeture de contours (Chalmond *et al.*, 2001).

Les pixels de contours forment généralement des lignes continues. Si un contour est significatif, il comporte au moins une section qui se démarque bien de son entourage. Il comporte peut-être aussi une ou plusieurs sections moins claires. Un gradient de grande intensité suggère la présence d'un contour et un gradient de faible intensité, l'absence de contour. Ces deux états sont plausibles

quand un pixel a une valeur intermédiaire. En effet, il existe une plage de valeurs pour laquelle certains pixels sont des contours et d'autres ne le sont pas.

La fermeture de contours, parfois appelée arrimage, liaison, remplissage de brèches ou suivi de lignes, vise à regrouper, compléter et rendre spatialement continues les sections ou fragments appartenant à un même contour. Dans le cas de frontières fermées, le terme extraction de frontières est aussi approprié (Wang *et al.*, 2005).

Le *seuillage par hystérésis* (Canny, 1986) permet de récupérer les sections de contours moins contrastées, sans étiqueter faussement trop de pixels de bruit. Un de ses buts est d'augmenter la connectivité de la carte de contours (Hancock et Kittler, 1991). Le seuillage se base sur l'idée que les pixels de contours isolés sont plus souvent des faux positifs que les pixels connectés (Nalwa et Binford, 1986).

Tout d'abord, un seuil élevé permet de trouver les parties de contours les mieux marquées. À partir de ces amorces, le tracé des contours peut être allongé. Les pixels voisins des contours dans la direction du gradient sont inspectés. Si leur amplitude est supérieure à un seuil moins élevé, ils sont classés comme de nouveaux pixels de contour. Cette étape de relaxation est répétée récursivement sur les nouveaux contours, jusqu'à convergence. Canny utilise un rapport de 2 entre les deux seuils. Un rapport basé sur la prévalence *a priori* des contours peut aussi être utilisé (Hancock et Kittler, 1991).

Le seuillage par hystérésis tend à diminuer la fragmentation des contours. Cependant, la présence de bruit près des vrais contours peut conduire à la formation de plusieurs branches courtes, connectées aux vrais contours qui nécessitent un nettoyage supplémentaire (Sappa, 2006).

Un *modèle bayésien* peut aussi être employé pour faire le groupement de traits. Cela est rendu possible par l'estimation de la probabilité de trois événements pouvant interrompre un trait : une courbure douce, un coin, ou une interruption causée par une occlusion, de l'ombre ou une perte de contraste. Ces événements sont décrits en termes de distance entre les traits, d'angle entre les traits, de différence d'intensités moyennes et de différences de contrastes (Elder et Goldberg, 2002). Le modèle a été testé sur la détermination des bordures exactes de lacs dans des images

panchromatiques IKONOS et sur la recherche de frontières de régions de peau humaine dans des images naturelles (Elder *et al.*, 2003).

Le groupement de contours peut se baser sur des statistiques écologistes des indices de formes (lois de la Gestalt) : proximité, continuité, similarité de brillance et contraste (Wang *et al.*, 2005; Mahamud *et al.*, 2003; Elder et Goldberg, 2002). La fermeture peut aussi être faite par dilatation morphologique adaptative (Jiang, 2000). Pour les applications industrielles, la fermeture de contours peut être effectuée par un circuit électronique dédié (Hajjar et Chen, 1999).

L'amincissement et la fermeture constituent ensemble l'étape de restauration des contours. Cette étape est la dernière opération de bas niveau dans la chaîne de traitements. Elle a pour résultat un plan sommaire mettant en évidence les aspects principaux d'une scène. Cette *esquisse* n'est composée que de courbes, sans ombre ou effet artistique de relief. La détection des contours est alors complète. Bien que la détection de contours ne soit qu'une étape dans la chaîne de traitements, elle peut parfois suffire à décrire le contenu de l'image, si cette dernière n'est composée que d'objets linéaires. Cependant, de façon générale, il est nécessaire de pousser plus loin l'analyse de la scène.

2.1.5.G Représentations intermédiaires et de haut niveau

La carte de contours n'est qu'une représentation de bas niveau basée sur les pixels. Pour en faciliter l'interprétation, elle est souvent convertie en une représentation intermédiaire. Cette dernière peut être composée de *primitives* comme les lignes, les cercles, les ellipses, les contours de différents types à différentes échelles, ainsi que les linéaments. Le terme *linéament* désigne les lignes brutes, élémentaires, caractéristique d'une forme, repérables sur des photos, aériennes ou terrestres, ainsi que sur des images satellites. Cette structuration facilite la conversion à une représentation de haut niveau basée sur les objets (Xue et Wu, 2005).

Ainsi, les courbes issues de la détection de contours servent à passer au niveau du *croquis*. Le croquis est un dessin formé de grands traits servant à donner des points de repère. C'est une représentation grossière de la scène, qui doit être corrigée en regroupant les *primitives* en objets significatifs. Une région de l'image délimitée par un contour de configuration reconnaissable est appelée une *forme*. Si la scène contient, par exemple, des bâtiments, le regroupement de lignes pourrait donner des boîtes rectangulaires. L'emplacement, l'organisation et les relations entre les

objets constituent la *topologie* de la scène (Rothwell *et al.*, 1994). Après l'analyse de la topologie, il est possible de produire un *schéma*, c'est-à-dire une description réduite aux traits essentiels.

Lorsqu'une seule image est utilisée pour estimer la profondeur, le résultat, qui ne peut être présenté que sous l'angle de visée original, est nommé *croquis à 2,5D*, pour le distinguer d'un résultat obtenu avec un couple stéréo d'images. Il est intéressant de noter que, jusqu'à ce stade de traitement, la seule information utilisée dans l'analyse d'images concerne les contours. À moins d'indication contraire, les erreurs commises au départ sont propagées jusqu'à cette représentation intermédiaire (Kass *et al.*, 1988).

Le groupement perceptuel est l'opération consistant à extraire des relations significatives dans l'image, sans connaissance du contenu de l'image, et à faire le groupement d'*entités* dans le but d'obtenir des structures significatives de haut niveau. Les premières recherches en groupement perceptuel datent des années 1920 et ont été faites par les psychologues de l'école Gestalt. Les principes de groupement hiérarchiques qu'ils ont proposés englobent des concepts comme la proximité, la similarité, la continuité, la fermeture et la symétrie (Sappa, 2006). À cette étape du traitement, seules les caractéristiques pertinentes de l'objet sont retenues en vue de le modéliser avant sa classification. La classification, non décrite ici par souci de concision, permet d'obtenir une carte d'occupation du sol.

2.2 Théorie de l'évaluation des performances

Les performances d'un algorithme peuvent être étudiées selon deux aspects, soit l'*optimisation* et la *caractérisation*. L'*optimisation* des performances est faite lors du développement d'un algorithme ou lors d'un choix itératif des paramètres d'exécution d'un algorithme. Par exemple, la démarche de création d'un détecteur de contours optimal, comme celui de Canny (1986).

Le concept de *caractérisation des performances* découle du problème qu'il n'est pas nécessairement évident de choisir la démarche la mieux adaptée pour résoudre un problème de vision artificielle. Il s'intéresse à la question de savoir quelle technique utiliser et quand l'utiliser. Selon Clark et Courtney (2000), tant que ce problème n'est pas résolu, la vision demeure surtout un domaine de recherche. La caractérisation des performances comporte deux volets : l'*évaluation* et la *prédiction*.

L'évaluation des performances est faite en examinant *a posteriori* le résultat d'un algorithme. Par exemple, les études faites par Bowyer *et al.* (2001). La *prédiction* des performances consiste à évaluer la capacité d'un algorithme à fournir des résultats corrects sur une image donnée.

Afin de pouvoir déterminer si un traitement améliore la qualité d'une détection de contours, il est essentiel de pouvoir mesurer la qualité de cette détection avant et après le traitement. Ainsi, l'évaluation empirique d'algorithmes prend de plus en plus sa place comme domaine d'étude en vision numérique. Une évaluation favorable aide à convaincre les utilisateurs de la maturité d'un algorithme. La méthodologie de comparaison est en elle-même un problème de recherche, méritant une étude sérieuse. Le plus souvent, il est difficile de faire une étude théorique pour évaluer si un algorithme excelle sur une métrique de performances donnée (Hoover *et al.*, 1996).

Un protocole d'évaluation d'algorithmes graphiques devrait comprendre trois éléments : une méthodologie d'acquisition de vérité-terrain, une définition de correspondance et une métrique de performance. La métrique devrait être quantitative, normative, objective, et, autant possible, compatible avec l'évaluation humaine (WenYin et Dori, 1999; Hoover *et al.*, 1996). Le développement de méthodes d'évaluation des performances pour les algorithmes de vision numérique peut exiger un effort considérable, particulièrement lors de la création de jeux d'images jumelés à des vérités-terrain. L'exploration des espaces de paramètres peut aussi demander un temps de calcul considérable (Bowyer *et al.*, 2001).

Aucun algorithme de vision par ordinateur n'est parfait. En particulier, les algorithmes de segmentation par région peuvent commettre des erreurs de sous-segmentation, en combinant à tort des objets distincts. Ils peuvent aussi sursegmenter, en scindant à tort des objets de base en parties. Quant aux algorithmes de détection de contours, ils ont aussi leurs modes de défaillance. Ils peuvent produire un surplus de contours qui empêche d'identifier les contours pertinents ou d'interpréter l'image. Ils peuvent aussi omettre des contours essentiels à l'interprétation (Suetens *et al.*, 1992).

Les algorithmes les plus performants ne sont pas nécessairement les plus connus ou les plus récents. Il est souhaitable d'avoir des méthodes pour valider les performances des algorithmes de vision, pour comparer ces performances entre elles. Si les suppositions faites par un algorithme

sophistiqué s'avèrent fausses dans un contexte donné, il est possible que ces performances soient moins bonnes que celles d'un algorithme plus simple (Bowyer *et al.*, 2001).

La présente section décrit plusieurs techniques de mesure utilisées pour l'évaluation des performances d'un détecteur : le taux de rendement, la figure de mérite de Pratt, le taux de fausse alarme, le taux de rappel et le taux de précision. La présentation est complétée par une description de particularités reliées à l'évaluation des performances de détection de contours.

L'oeil humain est souvent considéré comme le meilleur juge pour évaluer une interprétation automatisée du contenu d'une scène. Ainsi, la plus simple méthode d'évaluation des performances est de comparer des exemples d'images provenant de différents détecteurs (Chalmond *et al.*, 2001).

La plupart du temps, une description subjective accompagne les exemples (Vinken *et al.*, 2000; Heath *et al.*, 1997; Salotti *et al.*, 1996). Cependant, si une telle évaluation est faite sans un protocole expérimental rigoureux, la précision d'une telle évaluation peut être mise en doute. De façon générale, il est recommandé d'utiliser une méthode quantitative d'évaluation, basée sur l'utilisation d'une vérité-terrain.

Soit une image matricielle indiquant la présence ou l'absence d'un signal, par exemple les contours. Lors de l'évaluation d'une telle carte-image, chaque pixel est comparé à une vérité-terrain et catégorisé comme vrai positif (VP), vrai négatif (VN), faux positif (FP), ou faux négatif (FN). Ces quantités permettent de déterminer si un traitement améliore la qualité d'une détection de contours, en mesurant la qualité de cette détection avant et après le traitement.

Généralement en télédétection, la classification est la dernière étape dans la production d'une carte d'occupation du sol. Pour optimiser une classification, il est nécessaire de maximiser le rendement, pourcentage d'identifications correctes sur l'ensemble des classes contenues dans l'image. Dans le cas d'une classification binaire, ce rendement est défini par :

$$(VP + VN) / (VN + FP + FN + VP) \quad (2.11)$$

Maximiser le rendement revient à maximiser la somme :

$$VP + VN \quad (2.12)$$

D'autres mesures sont utilisées pour évaluer les classifications, par exemple le coefficient Kappa (Carletta, 1995). Cependant, ce dernier ne semble pas utilisé par la communauté scientifique pour la détermination de la présence ou de l'absence d'un signal comme les contours. Son emploi rendrait difficile la comparaison avec d'autres études.

2.2.1 Utilisation d'une vérité-terrain

Dans la littérature, il est rapporté que l'évaluation des performances d'algorithmes de vision numérique manque souvent de rigueur. Souvent on compare des images traitées à partir de différents détecteurs sans expliquer le choix de seuil utilisé pour chacune. Il est alors difficile de juger si les images comparées correspondent aux meilleures images possibles pour chaque détecteur étudié. Les algorithmes de détection de contours sont souvent évalués sans image réelle.

Dans un contexte opérationnel, il est essentiel d'évaluer la fiabilité et les niveaux de performance d'un procédé. L'adoption des technologies de vision numérique par l'industrie nécessite l'évaluation des performances attendues de ces technologies (Courtney, 2000). De nombreux auteurs sont de plus en plus conscients de l'importance d'une bonne évaluation des performances (Klette *et al.*, 2000).

Cette évaluation peut être théorique ou expérimentale. L'évaluation théorique est une analyse mathématique accompagnant la description d'un algorithme dans un document scientifique. Ce type de démarche s'appuie sur un modèle de contours et de bruit présumément contenus dans les images. Les détecteurs ainsi obtenus sont souvent optimaux par rapport au modèle choisi (Canny, 1986; Petrou et Kittler, 1991; Sarkar et Boyer, 1991; Fjørtoft *et al.*, 1998; Moon *et al.*, 2002). Cependant, l'élégance conceptuelle et la sophistication mathématique des algorithmes ne sont pas nécessairement gages de performance d'un algorithme.

Dans la présente section, nous nous intéresserons aux évaluations expérimentales, basées sur l'analyse *a posteriori* d'images de contours. Ces évaluations peuvent être divisées en deux catégories selon qu'elles utilisent ou non une vérité-terrain.

Plusieurs méthodes d'évaluation expérimentale sans vérité-terrain existent. La plus simple consiste à fournir des exemples d'images provenant de différents détecteurs à l'utilisateur. La plupart du temps, une description subjective accompagne les exemples (Vincken *et al.*, 2000).

L'utilisateur est alors laissé à lui-même. Il doit faire une évaluation visuelle des exemples pour juger des différences et décider s'il est d'accord avec les conclusions.

Ces évaluations subjectives peuvent être formalisées (Bowyer *et al.*, 2001). Heath *et al.* (1997) ont créé une telle banque d'images et l'ont utilisée pour démontrer leur méthode d'évaluation. Elle consiste à présenter plusieurs cartes de contours à des sujets humains. Ces derniers doivent alors évaluer la qualité des cartes. Une carte de contours est jugée utile si elle permet d'identifier un objet situé au centre de l'image. L'analyse statistique des résultats permet ensuite de comparer entre elles les performances des algorithmes. La tâche étudiée par Heath *et al.* (1997) est la reconnaissance d'objets. Comme les résultats ont été faits sur une tâche de reconnaissance d'objets, les résultats sont seulement directement significatifs à cette tâche. Cette tâche est le but à long terme de la vision numérique.

L'utilisation d'une banque d'images commune facilite la comparaison des performances d'algorithmes (Clark et Courtney, 2000). Une telle évaluation est intéressante, car, par définition, les résultats sont en accord avec l'évaluation visuelle des cartes de contours. Dans un autre exemple sans vérité-terrain, Cho *et al.* (1997) utilisent une technique inspirée de l'appariement de formes et du rééchantillonnage pour évaluer les performances de cartes de contours.

Déterminer avec précision le positionnement de contours flous est un problème difficile (Elder et Zucker, 1996). Par exemple, un lissage prononcé est nécessaire pour détecter les contours d'ombre sans fragmentation. Cependant, un tel lissage affecte le positionnement des contours (Lindeberg, 1996). Ainsi, les méthodes sans vérité-terrain ne permettent pas d'identifier avec certitude un contour décalé de sa vraie position (Heath *et al.*, 1997). Ainsi, malgré l'existence de plusieurs techniques d'évaluation sans vérité-terrain, plusieurs auteurs, notamment Salotti *et al.* (1998), croient qu'une vérité-terrain préparée par un opérateur humain est la meilleure carte de contours disponible.

Impliquer l'humain dans un processus validation basé sur des images réelles ne permet pas seulement de mesurer les effets du bruit sur la sortie d'un détecteur. Lors de la création d'une vérité-terrain, l'humain détermine la finesse requise pour l'interprétation. L'évaluation subséquente permet de s'assurer que les contours candidats trouvés par un détecteur

correspondent aux contours utiles à l'interprétation de l'image. En d'autres termes, il devient alors possible de mesurer les faiblesses sémantiques d'un détecteur (Florack, 2000).

Les exigences de l'utilisateur final induisent des critères subjectifs de qualité. Ces critères peuvent être validés par un observateur humain qui effectuerait une visualisation d'images en temps réel dans des conditions contrôlées. Une telle évaluation visuelle pourrait être faite tant pour les résultats intermédiaires que les résultats finaux (Haralick *et al.*, 1994). La détection de contours est une tâche subjective. Deux humains qui traitent la même image avec la même intention peuvent ne pas voir les mêmes contours dans l'image. Bien qu'une telle démarche soit utilisable dans un contexte de recherche et développement, son exécution demande beaucoup de temps (Heath *et al.*, 1997).

Cette lourdeur rend intéressant le développement de méthodes automatiques pour évaluer sommairement les résultats intermédiaires. Lors du développement, on peut imaginer que ces méthodes simplifient l'optimisation d'algorithme. Elles pourraient faciliter le choix de paramètres. Les méthodes d'évaluation de performance basées sur des vérité-terrain viennent en partie combler ce besoin (Bowyer *et al.*, 2001; Kanungo, 1995; Martin *et al.*, 2004).

Les cartes avec le meilleur score ne sont pas nécessairement les plus plaisantes visuellement (Konishi, 2003). Cependant, une vérité-terrain permet de considérer la richesse et la diversité du contenu des images naturelles, ainsi que les besoins de l'application finale. Elle permet d'intégrer l'expertise de l'humain d'intervenir dans le processus d'entraînement et d'évaluation. Une vérité-terrain préparée par un humain est nécessairement subjective, mais c'est sûrement, à ce jour, le meilleur étalon disponible. Les vérités-terrain doivent être préparées avec soin, car il est essentiel qu'elles contiennent beaucoup moins de « bruit » que les algorithmes devant être évalués (Min *et al.*, 2004).

Les résultats obtenus par Martin *et al.* (2004) montrent que la localisation des contours par les humains est encore meilleure que celle obtenue par des algorithmes de détection de contours. Des variations sont observées quand deux sujets décrivent la même partie d'une scène avec des niveaux de détails différents. Pour ce qui est des petites erreurs de localisation sur l'emplacement des contours, il est possible d'adapter une mesure de performance pour qu'elle puisse les tolérer. Martin *et al.* (2004), ainsi que Bowyer *et al.* (1999) ont démontré indépendamment que les

vérités-terrain de contours sont généralement cohérentes entre elles. Avec un zoom de 2X ou 3X et un peu d'entraînement, pratiquement tous les contours peuvent être tracés à la précision spatiale d'un pixel (Martin, 2002). Cependant, une bonne vérité-terrain ne garantit pas, à elle seule, la qualité de l'évaluation (Swonger, 1997).

Les méthodes d'évaluation avec vérité-terrain ne sont pas nécessairement simples à mettre en œuvre. Déterminer la vraie position des contours dans une scène naturelle peut sembler très difficile. Selon Heath *et al.* (1997), les limites de ces méthodes sont leur dépendance à une vérité-terrain. Ainsi, elles dépendent d'images synthétiques ou d'images simples pour lesquelles il est relativement aisé de spécifier la vérité-terrain. Ces images ne capturent pas la complexité des vraies scènes; ainsi, aucune de ces méthodes ne mesure les performances des détecteurs de contours dans des conditions réalistes. C'est pourquoi plusieurs chercheurs emploient des images synthétiques et la vérité-terrain correspondante pour évaluer leurs algorithmes. Des méthodologies élaborées peuvent être mises au point, par exemple, pour analyser l'effet de la présence de bruit sur les résultats d'un algorithme (Kanungo *et al.*, 1995).

Cependant, comme les images synthétiques n'ont pas les mêmes propriétés statistiques que les images naturelles, la limite de leur applicabilité, voire leur pertinence, est souvent mise en doute (Bowyer et Phillips, 1998). Ainsi, les évaluations sont souvent complétées par des exemples subjectifs sur des images naturelles sans vérité-terrain.

2.2.2 Méthode de mise en correspondance

Il faut définir une méthode pour mettre en correspondance les pixels détectés et ceux de la vérité-terrain. Il est souhaitable dans cette optique de garder en tête les trois objectifs proposés par Canny (1986) pour la détection de contours :

- le taux élevé de détection (rappel, $R = VP/(VP+FN)$);
- le bon positionnement;
- la détection unique (précision, $P = VP/(VP+FP)$).

La méthode la plus simple est de comparer un à un les pixels détectés avec la vérité-terrain, pour déterminer le nombre de VP, FP et FN. Une telle approche n'est pas acceptable, parce qu'elle ne

tolère aucune erreur de positionnement. Ce faisant, elle pénalise beaucoup les cartes de contours produisant des frontières utilisables, mais légèrement décalées spatialement. Ces frontières décalées étant considérées à tort comme des FP, les cartes qui les contiennent obtiennent des évaluations inférieures à celles de cartes semblables qui contiendraient des contours fragmentés au lieu de décalés.

Il faut aussi considérer le fait que la vérité-terrain elle-même peut contenir de légères erreurs de positionnement. Ainsi, il est nécessaire de tolérer une légère erreur de positionnement, tout en pénalisant les détections multiples (Konishi *et al.*, 2003).

Selon la méthode de Salotti *et al.* (1996), un contour détecté correctement est un pixel marqué dans la vérité-terrain avec, dans la fenêtre 3x3 centrée sur lui, au moins un pixel détecté qui lui correspond. Un contour manquant est un pixel marqué dans la vérité-terrain, sans contour détecté dans la fenêtre 3x3 centrée sur lui. Un faux contour est un pixel marqué par le détecteur, sans véritable contour dans la fenêtre 3x3 centrée sur lui. Il est facile de déduire qu'un non-contour est un pixel qui n'est ni un contour détecté correctement, ni un contour manquant, ni un faux contour. L'utilisation d'une fenêtre de taille 3x3 définit implicitement une distance de tolérance de $\sqrt{2}$. Ces définitions permettent de compiler une matrice de confusion. Les auteurs s'en servaient pour estimer les erreurs de détection avec les quantités (1 - précision) et (1 - rappel). La méthode induit cependant un biais dû aux appariements multiples.

Martin *et al.* (2002) utilisent une distance de tolérance de deux pixels. Ils inspectent donc une partie des pixels dans une fenêtre 5x5. Ils regardent la mesure-F de chaque détecteur en fonction de la distance tolérée. La qualité assignée augmente avec la distance et les lignes sont parallèles. Ils en échantillonnent sur des voisinages 3x3, 5x5 et 7x7. La complexité de leur méthode d'appariement ne nous paraît pas justifiée, particulièrement après les essais effectués par Bowyer (2000), qui a observé très peu de différences entre diverses méthodes d'appariement.

2.2.3 Méthodes quantitatives d'évaluation avec vérité-terrain

La méthodologie d'évaluation doit être adaptée à l'application étudiée. En vision numérique, l'entropie conditionnelle, le coefficient d'information de Chernoff et le coefficient de Bhattacharyya sont parfois utilisés pour étudier la complémentarité de sources de données. L'information de Chernoff est optimale pour déterminer si un pixel est, ou n'est pas, un élément

de contours. La distance de Bhattacharyya est optimale pour distinguer un pixel de contour d'un pixel d'arrière-plan (Yuille et Coughlan, 2000). Ces deux tâches étant semblables, il n'est pas surprenant que Konishi *et al.* (2003) obtiennent des résultats comparables suite à l'une ou l'autre des mesures. Cependant, ces mesures sont utilisées lors du développement d'un détecteur.

En plus d'utiliser simplement un indice quantitatif, il est aussi important d'échantillonner l'espace paramétrique du détecteur évalué. Cela permet d'étudier le comportement du détecteur et d'identifier la meilleure performance qu'il est capable de fournir sur une image donnée. Bowyer *et al.* (2001) insistent pour que les chercheurs emploient plus souvent cette démarche. Pour ce faire, il est nécessaire de disposer de vérités-terrain sur plusieurs images réelles.

Il est possible d'utiliser le test de Kruskal-Wallis pour tester la significativité statistique des différences de performance entre plusieurs algorithmes (Nair *et al.*, 1995). L'analyse de la variance peut aussi être utilisée pour déterminer si les différences de moyennes entre les détecteurs sont significatives (Heath *et al.*, 1997; Borra et Sarkar, 1997). Lorsque que deux algorithmes sont comparés, un test des signes basé sur l'hypothèse d'une distribution binomiale peut être utilisé (Min *et al.*, 2004).

Lors de l'évaluation de la détection de contours, la qualité des décisions est plus souvent quantifiée, directement ou indirectement, avec le risque bayésien. Ce dernier est défini comme une moyenne pondérée des FN et des FP (Konishi *et al.*, 2003). Dans la littérature, un indice, ainsi que deux types de courbes semblent être plus souvent utilisés pour minimiser le risque de Bayes. Ce sont la figure de mérite de Pratt, les courbes de ROC (*“receiver operating characteristic”*, fonction d'efficacité de l'observateur) et les courbes de précision-rappel. L'utilisation de courbes permet de prendre en compte l'influence du seuil dans la méthode d'évaluation, en donnant une mesure des performances pour chaque valeur possible du paramètre de seuillage.

2.2.3.A Figure de mérite de Pratt

La figure de mérite de Pratt (Abdou et Pratt, 1979) est un indice souvent utilisé pour évaluer et comparer les détections de contours, notamment pour comparer la dégradation de leurs performances en présence de bruit (Umbaugh, 1998; Cho *et al.*, 1997). Elle est définie par :

$$M_{Pratt} = \frac{1}{\max(P(G), P(A))} \sum_{i=1}^{P(A)} \frac{1}{1 + \beta d^2}, \quad (2.14)$$

où :

- $P(G)$ est le nombre de pixels de contours dans l'image de vérités-terrain;
- $P(A)$ est le nombre de pixels de contours trouvés par le détecteur de contours;
- $\max(x, y)$ est la fonction maximum, $x + y + |x - y|$;
- β est une constante (adaptée au type d'image) pour pénaliser les contours décalés;
- d est la distance entre un contour présumé et le contour réel le plus près.

L'évaluation pénalisant les pixels selon la distance, elle ne fait pas intervenir les concepts binaires de « faux positifs » ou de « faux négatifs », sauf pour les deux cas non intéressants $\beta = 0$, où l'erreur de positionnement est toujours pardonnée et $\beta = +\infty$, où aucune erreur de positionnement n'est tolérée.

Quand la valeur β n'est pas nulle, l'indice est, en théorie, calculé sur un support infini. Pour tous les pixels détectés, il faut déterminer la distance euclidienne au pixel de vérité-terrain le plus près. En pratique, pour éviter des temps de calculs prohibitifs, une approximation par des masques de Chamfer peut être utilisée (Rothwell *et al.*, 1994).

La division par $\max(P(G), P(A))$ permet de pénaliser les contours trop larges ou manquants. Les contours trop larges sont généralement moins pénalisés que les contours manquants. Cela est souhaitable quand il existe des techniques pour amincir les contours. Une analyse théorique de la figure de mérite de Pratt est donnée dans l'Annexe 2.

2.2.3.B Courbes de ROC

Bowyer *et al.* (2001) ont utilisé des courbes de ROC (*receiver operating characteristic*, fonction d'efficacité de l'observateur), calculées à l'aide de vérités-terrain, pour comparer le résultat de différents détecteurs. Yitzhaky et Peli (2003) ont aussi utilisé les courbes de ROC, avec des estimations de la vérité-terrain. Les axes de ces courbes sont le taux de réussite et le taux de fausse alarme. Le taux de réussite (ou sensibilité, rappel) est un pourcentage exprimant le rapport entre le nombre d'objets pertinents retrouvés et le nombre total d'objets pertinents (Baeza-Yates et

Ribeiro-Neto, 1999). C'est donc la probabilité qu'un algorithme A détecte un signal ($A = 1$) quand un signal est présent ($G = 1$).

$$P(A=1|G=1)=VP/(VP+FN) \quad (2.15)$$

En évaluation des performances, la notion de sensibilité est intimement liée au choix des paramètres. Si les paramètres sont choisis pour rendre le détecteur très sensible, il aura tendance à trouver beaucoup de pixels de contour potentiels qui sont en fait attribuables au bruit. Si le détecteur n'est pas assez sensible, il peut manquer des contours valides (Umbaugh, 1998).

Le taux de fausse alarme (ou taux de déchet, taux de chute, “*fallout ratio*”) est la probabilité qu'un algorithme donne une réponse positive en l'absence de signal. Il est défini par :

$$FP/(VN+FP) . \quad (2.16)$$

Il correspond à la proportion de pixels d'arrière-plan faussement identifiés comme des contours. Les détecteurs peuvent aussi être comparés selon l'aire sous la courbe. L'optimisation des paramètres guidée par une courbe de ROC se fait en variant les paramètres de l'algorithme pour mettre des points sur la courbe. L'optimum est identifié en trouvant un point sur la courbe qui maximise le rappel et minimise le taux de fausse alarme.

Cependant, il n'est pas évident que les courbes ROC soient l'outil le plus approprié pour évaluer la détection de frontières (Grigorescu *et al.*, 2003; Martin *et al.*, 2004). En effet, il y a un problème avec l'utilisation du taux de fausse alarme, ou de son complément, la spécificité, définie comme la probabilité qu'un algorithme ne détecte pas de signal ($A = 0$) quand il n'y a pas de signal ($G = 0$),

$$P(A=0|G=0)=VN/(VN+FP) . \quad (2.17)$$

Ces deux quantités dépendent de la taille des pixels. Si la limite de résolution d'une image est augmentée d'un facteur n , le nombre de pixels augmente en n^2 . Le nombre de VN croît donc en $O(n^2)$. Cependant, puisque les frontières sont des lignes de largeur nulle, le nombre de VP (et de FP) croît en $O(n)$. Ainsi, le taux de fausse alarme varie selon la résolution de l'image, ce qui complique souvent les comparaisons entre différents types d'images.

2.2.3.C Courbes de précision-rappel

Les courbes de précision-rappel sont le deuxième type de courbe pouvant être utilisé pour minimiser le risque bayésien. Elles sont beaucoup utilisées dans le domaine de la recherche documentaire. Dans ce contexte, le taux de précision est la proportion de documents pertinents retrouvés par une requête. Une requête a un taux de précision de 1 si tous les documents qu'elle retourne sont pertinents. Le taux de rappel est la proportion des documents pertinents dans une base de données qui sont retrouvés par une requête. Une requête a un rappel de 1 si elle retourne l'ensemble des documents pertinents (Baeza-Yates et Ribeiro-Neto, 1999).

Dans le contexte du traitement de signal, pour éviter une propagation d'erreur hors contrôle, il faut utiliser des données suffisamment fiables (Brogaard et Ólafsdóttir, 1997). Les courbes de précision-rappel permettent de capturer le compromis entre la proportion de vrai signal requis pour permettre le succès d'une opération de plus haut niveau et la proportion de bruit tolérable (Martin, 2002), le « meilleur » détecteur de contour étant celui qui permet à l'application de haut niveau subséquente de performer le mieux (Shin *et al.*, 2001).

Haralick (1984) a utilisé la précision et le rappel pour évaluer son détecteur de contours basé sur la dérivée seconde orientée dans la direction du gradient. Sofka et Stewart (2003) ont aussi utilisé des courbes de précision/rappel pour évaluer les performances d'un détecteur de micro-anévrismes de la rétine. L'utilisation de la spécificité a été évitée, car elle requiert de définir des cibles non-micro-anévrisme, dont la sélection a été jugée très arbitraire et subjective. La précision et le rappel peuvent aussi être appelés, respectivement, taux de véracité et taux d'identification.

En recherche documentaire, la proportion de vrai signal détecté est quantifiée par le taux de rappel :

$$R = VP / (VP + FN) \quad (2.18)$$

Le silence est défini par :

$$S = 1 - R = FN / (VP + FN) \quad (2.19)$$

La proportion de signal détecté correspondant au vrai signal, nommée taux de précision, est définie par :

$$P = VP / (VP + FP) \quad (2.20)$$

Le bruit est défini comme :

$$B = 1 - P = FP / (VP + FP) \quad (2.21)$$

Le compromis entre la quantité de signal (fort taux de rappel) et la qualité du signal (fort taux de précision) est donné par la mesure-F. Cette dernière est définie comme une moyenne harmonique pondérée entre P et R :

$$F_{\alpha} = \frac{1}{\alpha P^{-1} + (1 - \alpha) R^{-1}} \quad (2.22)$$

La mesure-F maximale sur la courbe donne le seuil optimal pour un α donné, choisi en fonction des opérations de plus haut niveau. Pour minimiser le risque bayésien, en prenant un coût nul pour les bonnes décisions et un coût égal pour les deux types d'erreurs, il est possible d'employer la méthodologie de la mesure-F avec $\alpha = 0,5$, ce qui donne :

$$F_{\alpha=0.5} = \frac{2PR}{P+R} = \frac{2}{P^{-1} + R^{-1}} = \frac{2VP}{2VP + FP + FN} \quad (2.23)$$

Chaque point sur la courbe précision-rappel correspond à un seuil différent du détecteur de contours. Pour déterminer ces points, il est nécessaire de définir la manière dont la précision et le rappel seront mesurés. Pour chaque niveau de seuillage, la carte-image obtenue doit être comparée à une vérité-terrain, afin de déterminer le nombre de VP, FP et FN, qui permettront de calculer la précision, le rappel et, finalement, la mesure-F.

2.3 Théorie de la prédiction des performances

Avant d'appliquer un algorithme sur une image, nous ne savons pas s'il est le mieux adapté au contenu de cette image. Par ailleurs, les résultats de l'étude de Heath *et al.* (1997) suggèrent que, pour un type d'image donné, le détecteur de contours le mieux adapté peut varier d'une image à l'autre. La mise au point de techniques pour faciliter le choix d'algorithme ou d'image faciliterait sûrement l'adoption des techniques de vision numérique hors des milieux de recherche et développement (Clark et Courtney, 2000).

L'estimation des performances d'un détecteur de contours permettrait d'estimer si, vraisemblablement, son résultat pourra satisfaire aux exigences des étapes ultérieures de traitement. Dans une chaîne de traitement avec une approche ascendante, la modélisation des

erreurs permet de les corriger avant d'arriver à une représentation intermédiaire. Un détecteur adaptatif pourrait ajuster son comportement selon les propriétés du voisinage d'un pixel traité.

Selon Ziou et Tabbone (1998), pour définir un détecteur de contours, il est insuffisant de définir une marche à suivre pour le lissage, la dérivation et l'étiquetage. Il faut, en plus, donner une stratégie de choix d'échelle, ainsi qu'une description des contextes dans lesquels le détecteur peut être utilisé. Il faut donc une méthode de sélection d'algorithme, soit au niveau de l'image ou du pixel. Salotti *et al.* (1996) proposent de choisir un détecteur selon les prévalences des différents types de contours présents dans l'image.

Il y a longtemps que les chercheurs réfléchissent au problème de prédiction des performances, notamment dans le contexte d'algorithmes de segmentation par région (Rosenfeld, 1979). En principe, la prédiction des performances permet aussi de choisir le détecteur de contours le mieux adapté à une image. Elle permet d'assister l'utilisateur en l'aidant à mieux comprendre les conséquences d'un choix d'algorithme. De plus, au lieu de s'intéresser à la performance globale d'un test, il est possible de s'intéresser aux pixels qui composent l'image.

La prédiction des performances est déjà utilisée pour choisir une technique de restauration qui minimise l'erreur de la reconnaissance optique des caractères (Cannon *et al.*, 2003). Elle est aussi utilisée pour déterminer un seuil inférieur sur la probabilité d'identification correcte de cibles militaires dans des images radar. La prédiction est basée sur l'hypothèse que les erreurs dues à la localisation, à l'encombrement et au recouvrement entre les objets suivent une distribution uniforme. Selon Boshra et Bhanu (1999), l'utilisation de cette hypothèse rend souvent la prédiction trop optimiste. Dans les faits, les erreurs ne sont pas distribuées uniformément. Leur distribution peut varier d'un jeu de données à un autre. L'estimation de ces distributions pourrait avoir comme base une prédiction contextuelle des performances sur les étapes de traitement précédant celle de l'identification des cibles.

Mathématiquement, la prédiction des performances correspond à l'estimation, pour chaque pixel, du Coefficient de Prédicativité d'un test Positif (CPP) et du Coefficient de Prédicativité d'un test Négatif (CPN). Le CPP est défini par :

$$\text{CPP} = \text{VP}/(\text{VP}+\text{FP}) \quad (2.25)$$

et le CPN, par :

$$CPN = VN/(VN+FN) \quad (2.26)$$

Ainsi, le calcul du CPP à partir d'une matrice de confusion est le même que celui de la précision utilisée en évaluation des performances. Quand un algorithme est appliqué à chaque site s d'une image, l'image résultante, notée A , peut être combiné aux prédictions de performances pour calculer la confiance $C(s)$, selon la relation suivante :

$$C(s) = \begin{cases} CPN(s), & \text{si } A(s)=0 \\ CPP(s), & \text{si } A(s)=1 \end{cases} \quad (2.27)$$

Chalmond *et al.* (2001) ont proposé une banque d'indicateurs capables de prédire la performance du détecteur de Sobel sur une image aérienne infrarouge. Ils entraînent leurs indicateurs de performances sur les indices : filtres de Gabor, entropie de second ordre et rapport signal sur bruit.

L'entraînement supervisé d'un opérateur pouvant prédire les performances nécessite des données d'entraînement sous forme d'une réalité du terrain partielle, ou complète. Faute d'utiliser une image de réalité du terrain, Chalmond *et al.* (2001) ont classé des pixels dans les classes « non-contour, » « incertain » et « contour » selon que la valeur de l'amplitude du gradient de Sobel était presque nulle, près du seuil, ou près du niveau de saturation. La validation des cartes de performances peut se faire par une évaluation visuelle subjective (Chalmond *et al.*, 2001).

Chalmond *et al.* (2001) refusent d'utiliser une vérité-terrain sous prétexte que les contours spécifiés pourraient ne pas être cohérents avec le contexte. Cependant, leur façon de choisir des exemples d'entraînement en utilisant la carte d'amplitude du gradient de Sobel fait de cette dernière une vérité-terrain implicite.

En effectuant des seuillages prudents sur cette carte, ils parviennent à identifier des pixels où la probabilité d'avoir un contour est très grande; pour l'entraînement, ces pixels sont considérés comme des VPs. Ils peuvent aussi identifier des pixels où la probabilité de trouver un contour est très petite; pour l'entraînement, ces pixels sont considérés VNs. Sans réalité du terrain, les pixels restants sont dans une zone grise; Chalmond *et al.* (2001) n'ont pas cru possible de différencier les ensembles de points FNs et FPs.

Ainsi, Chalmond *et al.* (2001) regroupent les deux types d'erreurs (FN et FP) dans une même classe. Dans leur expérience, cette classe comprend surtout des contours isolés ou des contour à l'extrémité de contours. Ils proposent ensuite de modéliser la prédictivité positive du détecteur de Sobel en comparant la classe « erreur » avec la classe VP et de modéliser sa prédictivité négative en comparant la classe « erreur » avec la classe VN. Une telle simplification du problème empêche de faire la distinction entre la validité intrinsèque de l'algorithme de Sobel et sa validité prédictive.

2.4 Sélection d'images

Nous avons passé en revue les concepts de contours, évaluation des performances et prédiction des performances. La synthèse de ces éléments, ainsi que le souci de pouvoir créer des cartes de contours appropriées aux applications subséquentes, conduisent à une réflexion sur la sélection d'images.

Le choix du « meilleur » détecteur de contours dépend de l'image considérée (Bowyer *et al.*, 1999). Pour ce faire, il est nécessaire de s'assurer que les algorithmes utilisés soient bien adaptés à l'image traitée. Ajouter à un système d'information géographique la capacité de conseiller l'utilisateur à l'aide d'un module utilisant des techniques d'intelligence artificielle est une avenue prometteuse d'après les travaux de Vasconcelos *et al.* (2002). Ainsi avons-nous d'abord pensé résoudre le problème en développant un système à base de connaissances, basé sur des règles et sur des cas d'utilisation jumelés et des évaluations des performances. Un tel système aurait été capable d'expliquer ses choix d'images à l'utilisateur. Cependant, chaque ajout de règles au système nécessite la participation d'un expert thématique. De plus, la mise en place d'une telle approche aurait nécessité un système opérationnel de mise à jour de cartes avec un historique d'utilisation suffisamment vaste pour permettre l'extraction éventuelle de nouvelles règles de sélections ainsi que la prise de décision basée sur les cas rencontrés dans le passé.

N'ayant pas de telles données, il s'est avéré nécessaire de considérer une façon de faire basée sur la vision numérique, qui ne permet pas d'expliquer ses choix d'images, mais nécessite peu d'intervention humaine (Rousseau, 2003). Ainsi, nous avons développé un premier prototype logiciel (Figure 2.9) capable d'utiliser un réseau de neurones pour apprendre des fonctions de prédiction contextuelle des performances (Rousseau *et al.*, 2003; Rousseau, 2004). L'idée

première était d'avoir un banc d'essai pour évaluer les possibilités et les limites de la prédiction des performances pour effectuer un choix d'image et/ou d'algorithme.

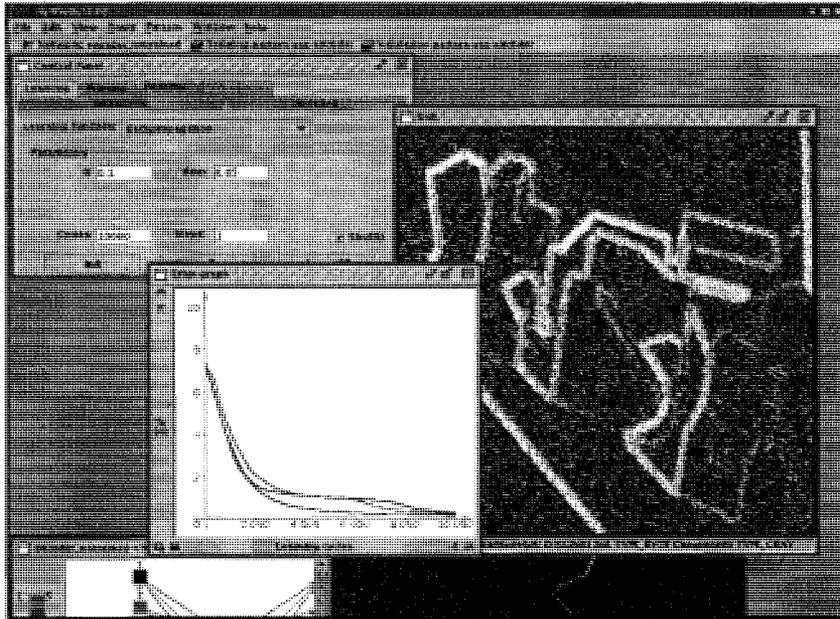


Figure 2.9 : Logiciel PerfPredict

Un des premiers constats fait était la faiblesse de la résolution spatiale du résultat. Cela rend difficile le développement d'une stratégie de choix d'image et/ou d'algorithme. Les causes probables sont la taille des fenêtres utilisées pour faire le calcul des indicateurs, ainsi que l'apprentissage demandé au réseau, qui doit modéliser une information dépendant à la fois le type de contours dans une image et la réponse d'un détecteur particulier à ce type de contours.

Malgré cela, le prototype permet de corriger partiellement une image contenant beaucoup de fausses alarmes suite à un mauvais choix de seuil, en considérant comme arrière-plan les pixels de contours pour lesquels la confiance est estimée à moins de 50 % (Figure 2.10).

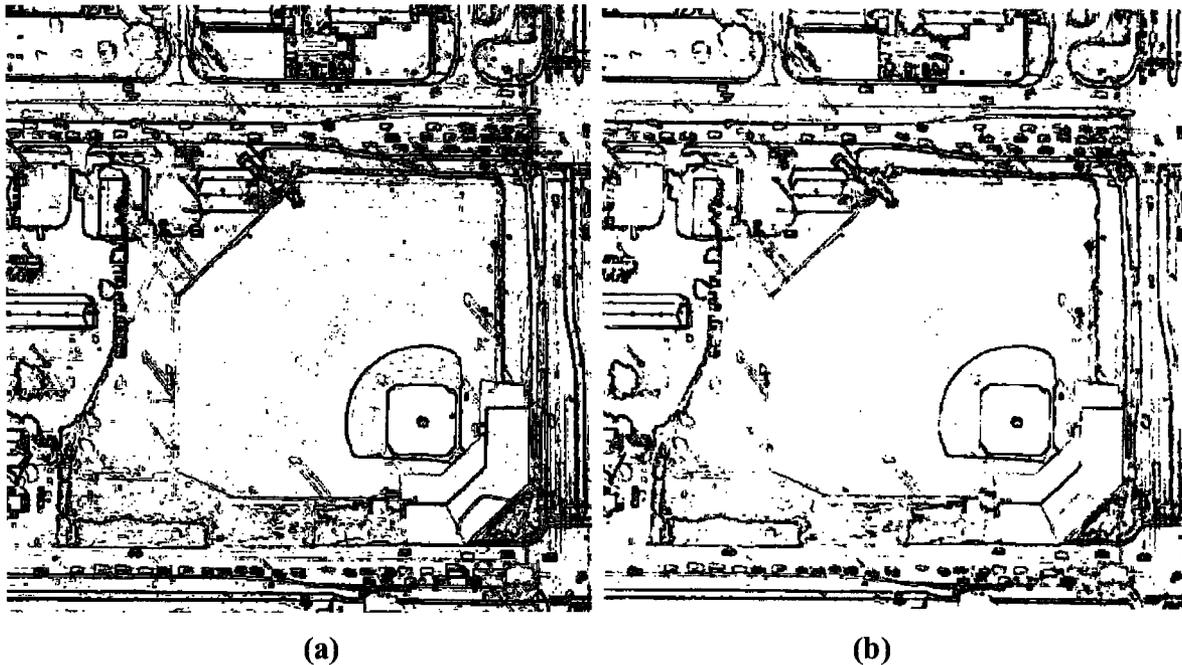


Figure 2.10 - Réduction de FP sur l'image « baseball »

(a) Algorithme de Prewitt ($F = 26,1 \% \pm 0,7 \%$),

(b) Algorithme de Prewitt corrigé ($F = 28,2 \%$).

Cependant, deux défauts nuisent à la mise en application de cette approche. Premièrement, la création d'un détecteur de contours adaptatif, basé sur la maximisation des performances prédites, nécessite la comparaison de plusieurs détecteurs. En pratique, la mise en place de cette approche est difficile. En effet, elle nécessite l'entraînement de plusieurs réseaux de neurones, chacun spécifique au jumelage d'un type d'image avec un détecteur de contours. De plus, l'utilisation d'un système basé sur cette approche nécessite, pour chaque choix d'algorithme et pour chaque détecteur de contours considéré, d'appliquer un réseau de neurones différent à l'image étudiée. Autre difficulté rencontrée, la communication avec la librairie JavaNNS nécessite l'utilisation de fichiers en format texte pour représenter les données, ce qui engendre un temps d'attente prohibitif, particulièrement lors de l'encodage. Il n'est donc pas évident de justifier le système selon des gains de temps à l'utilisation, en particulier s'il y a beaucoup d'images d'entrées à comparer.

Deuxièmement, l'utilisation du logiciel pour entraîner des réseaux de neurones sur différents détecteurs de contours laisse croire qu'il n'est pas possible, avec les indicateurs étudiés, d'obtenir

des cartes de prédiction des performances significativement différentes pour différents détecteurs de contours. Malgré ces deux limitations, l'outil prototype démontrait quand même une utilité certaine. Si un outil est suffisamment puissant pour identifier les zones d'ambiguïté et prédire la performance de détecteurs de contours, pourquoi ne pas s'en servir pour détecter les contours ou corriger un détecteur de contours? Il faudrait vérifier la possibilité d'utiliser l'outil pour détecter les contours, tel que suggéré par Chalmond et Graffigne (1999), ou pour restaurer des contours.

3. Méthodologie

Le logiciel développé dans le cadre de cette thèse se rapproche des travaux :

- de Konishi *et al.* (2003), en ce qui concerne l'utilisation de vérité-terrain pour l'entraînement,
- de Martin *et al.* (2004), pour le choix d'utiliser la mesure-F, ainsi que
- de Chalmond *et al.* (2001) pour le choix d'indicateurs.

Par contre, il est orienté vers la restauration de contours au lieu de leur détection. Dans ce chapitre, la méthodologie générale est suivie d'une description des données utilisées.

3.1 Méthodologie générale

Les étapes de notre travail sont :

- choix d'une méthode d'évaluation;
- détermination de cartes de base;
- choix d'indicateurs de présence de contours;
- entraînement supervisé;
- fusion de l'indice et des cartes de base;
- caractérisation de la robustesse au bruit;
- validation des résultats;
- analyse de sensibilité de la méthode développée.

La méthodologie fait l'objet de la Figure 3.1. Les cases numérotées concernent des manipulations. La rédaction de ce protocole expérimental répond à l'objectif initial de la thèse, celui de concevoir une méthode d'amélioration des décisions de détecteurs de contours.

L'approche implémentée dans le logiciel commence par l'utilisation des indicateurs de Chalmond pour entraîner un réseau de neurones réagissant à la présence de contours. L'information fournie par ce réseau est combinée à une carte de contours classique en vue d'en améliorer la qualité, notamment en réduisant les fausses alarmes. Le résultat de cette fusion est seuillé pour obtenir

une carte de contours optimisée. La Figure 3.2 donne une vue algorithmique du calcul utilisée pour restaurer les contours.

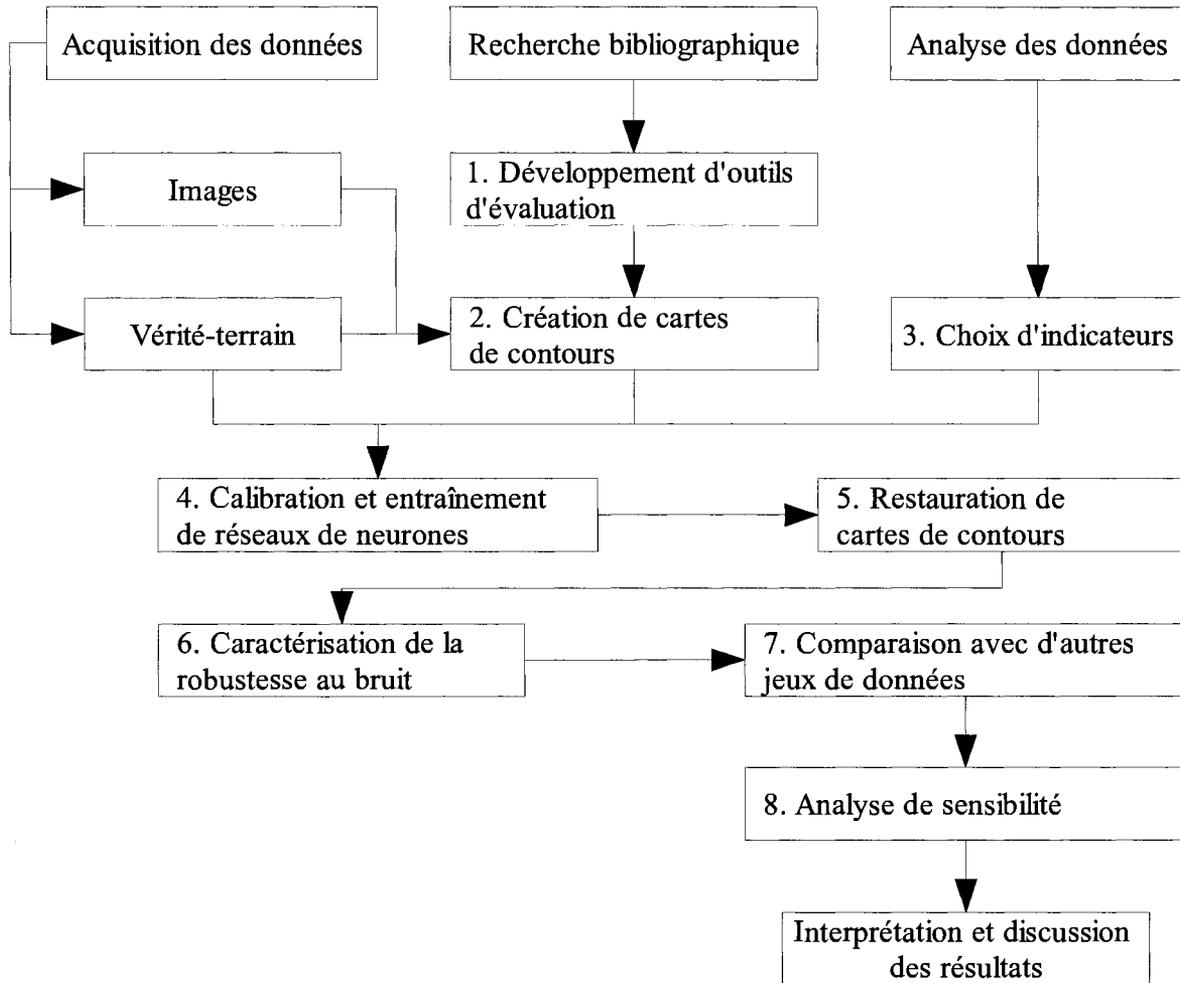


Figure 3.1 - Organigramme méthodologique

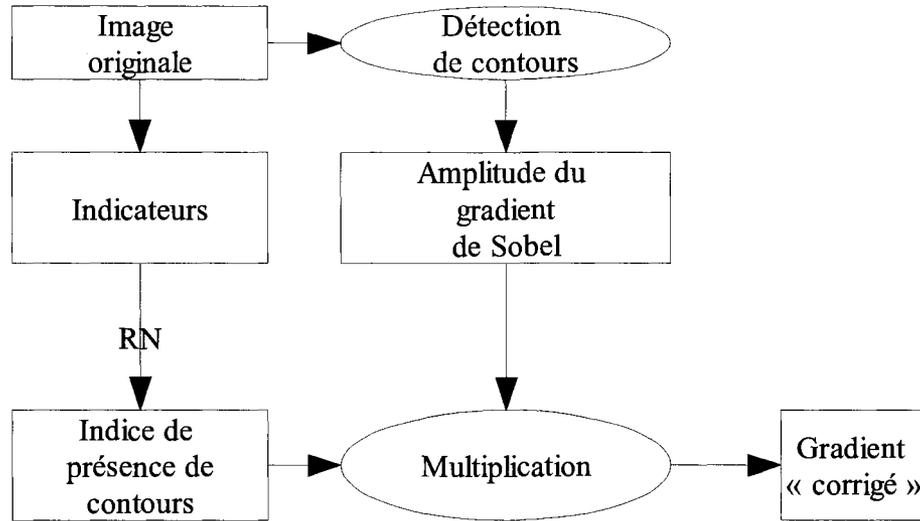


Figure 3.2 - Méthodologie du calcul de contours améliorés basé sur l'indice de présence de contours

3.1.1 Développement d'outils d'évaluation

Selon Salotti *et al.* (1996), l'utilisation d'un jeu de données basé sur des images naturelles permet une évaluation basée sur le plus réaliste des modèles de contours, soit les contours eux-mêmes. Selon eux, l'expertise humaine est l'outil le plus sophistiqué pour créer des vérités-terrains de contours.

Voici la méthode de mise en correspondance utilisée dans le présent travail. En principe, il y a peu de différence entre les diverses méthodes d'appariement possibles (Bowyer, 2000). Dans cet esprit, nous avons opté pour une méthode plus simple que celle utilisée par Martin *et al.* (2002).

Les pixels détectés sont examinés un à un pour voir s'ils correspondent à un pixel de vérité-terrain situé à une distance inférieure ou égale à deux pixels. Dans le cas d'une réponse positive, le pixel détecté et le pixel de la vérité-terrain sont considérés comme correspondants. Ils sont éliminés pour le reste du processus d'appariement. Si un pixel détecté correspond à plus d'un pixel de la vérité-terrain, celui qui est le moins éloigné est choisi. Un pixel détecté peut quand même correspondre à plusieurs pixels de la vérité-terrain, auquel cas les pixels de vérité-terrain concernés ne seront éliminés pour la distance considérée que s'ils correspondent à plusieurs pixels de contours détectés. L'algorithme détaillé est donné dans l'Annexe 3. À la fin du procédé, le résultat est le nombre de pixels vrais positifs, faux négatifs et faux positifs (VP, FN et FP) pour la carte de contours considérée.

Il faut aussi considérer le problème du choix d'une quantité pour résumer la matrice de confusion en une seule quantité. Plusieurs techniques de mesure ont été considérées, notamment le rendement et la mesure-F.

Le rendement n'est pas une façon appropriée d'optimiser la performance d'un détecteur de contours. Quand une classe occupe à elle seule la majeure partie de l'image, le rendement peut être élevé seulement à cause du hasard (Graham et Bull, 1998).

Les contours ont un faible taux de prévalence $P(G)$ (G pour "ground-truth", vérité-terrain), compris entre 5 % et 15 %. Ainsi, une image contient beaucoup plus de vrais négatifs que de vrais positifs. Dans un tel contexte, maximiser le rendement favorise les vrais négatifs. Sauf pour un petit seuil, où la quantité de FP sera très grande, le rendement choisi sera de l'ordre de :

$$1 - P(A) \approx 90 \% \quad (2.13)$$

où $P(A)$ est la proportion de pixels de contours détectés par un algorithme A . Le rendement de presque tous les seuils étant du même ordre, il peut être difficile de déterminer le meilleur seuil. Il y a un risque élevé de choisir un seuil trop grand.

Le résultat serait une image avec un faible $P(A)$, donc de faibles VP et FP. Très peu de pixels seraient identifiés comme contours. Ce phénomène est illustré dans l'Annexe 4. Bien qu'une telle image contienne peu de FP, le nombre élevé de FN la rend inutilisable. Suite à des essais sur des images, il apparaît que les outils conventionnels d'évaluation de classification ne conviennent pas à l'évaluation de détecteurs de contours. Il est donc nécessaire d'utiliser un critère autre que le rendement pour évaluer les cartes de contours.

La mesure-F offre quelques avantages intéressants (Nakache et Métails, 2005). Tout d'abord, lorsque la précision est égale au rappel, on obtient :

$$\text{précision} = \text{rappel} = \text{mesure-F} \quad (2.24)$$

Ensuite la moyenne harmonique baisse fortement lorsque l'un de ces paramètres est bas. Elle est majorée lorsque les deux paramètres sont proches tout en étant élevés. Ainsi, les algorithmes qui dégradent la précision ou le rappel au détriment de l'autre quantité sont pénalisés. Dans le cas $\alpha = 0,5$, on tend à trouver le maximum quand $FP \approx FN$. La mesure-F maximale est donc proche du point d'équilibre. Contrairement à la spécificité, la précision n'est pas assujettie au problème

de taille des pixels, car elle est normalisée par le nombre de VP au lieu du nombre de VN. D'une certaine manière, comme la précision et le rappel sont normalisés par VP, la mesure-F s'ajuste automatiquement à la prévalence des contours; il n'y a pas d'effet relié au débalancement entre VN et VP. De plus, la courbe de mesure-F est habituellement unimodale, justifiant un peu plus l'emploi de la mesure-F maximale comme descripteur de la performance (Martin, 2002).

Ainsi, notre choix de technique de mesure s'est arrêté sur la mesure-F. Afin de mieux valider les résultats obtenus, un intervalle de confiance est calculé pour la mesure-F. La méthode utilisée est celle de Samsa, telle que rapportée par Graham et Bull (1998). Dans notre notation, l'intervalle de confiance pour $\alpha = 0,05$ est

$$F \pm 1.96 \sqrt{F(1-F)/G} \quad (3.4)$$

où F est la mesure-F et G est le nombre de pixels de contours dans la vérité-terrain. Ce calcul est simple, car la méthode de Samsa suppose que FP = FN. Les intervalles obtenus sont larges comparés à ceux provenant d'autres méthodes, ce qui permet une comparaison plus conservatrice.

3.1.2 Création de cartes de contours

Avant de faire la restauration de contours, il faut déterminer la meilleure carte de contours capable d'être produite par un algorithme pour une image donnée. Ainsi, si la méthode de restauration de contours permet d'améliorer davantage la carte, l'efficacité de cette procédure s'en trouvera démontrée plus rigoureusement.

Il est nécessaire de refaire l'échantillonnage sur chaque image. En effet, l'utilisation de paramètres fixes peut carrément inverser les conclusions : dans l'expérience de Heath *et al.* (1997), l'algorithme de Canny est celui qui performe le mieux avec des paramètres adaptés à chaque image, mais qui a les moins bonnes performances lorsqu'il est utilisé avec des paramètres fixes.

La recherche de la meilleure carte de contours se fait en parcourant l'espace paramétrique de chaque détecteur et en sélectionnant, dans cet espace, la carte maximisant le critère de performance choisi. Plusieurs méthodes d'échantillonnage ont été proposées dans la littérature, par exemple par Bowyer *et al.* (2001), Shin *et al.* (2001) et Min *et al.* (2004).

Pour que cette exploration soit informatiquement réalisable, nous avons favorisé le choix de détecteurs possédant un nombre restreint de paramètres. Les détecteurs considérés dans la

présente étude ayant, au plus, deux paramètres, il a été convenu d'explorer complètement les espaces paramétriques au lieu de les échantillonner. Le choix d'algorithmes a aussi été guidé par le souci d'utiliser une variété de détecteurs, tous présentés dans des articles évalués par des pairs. Les algorithmes choisis sont ceux de Sobel, Canny, Argyle, Kirsch, Moon, ainsi que Frei-Chen.

L'algorithme de Sobel est intéressant pour sa simplicité, mais il donne des contours larges. Il produit des FPs en présence de bruit ou de texture et a de la difficulté à extraire les lignes fines. Nous l'avons utilisé, dans un premier temps, avec un seuillage simple de l'amplitude du gradient, ensuite avec un seuillage adaptatif basé sur l'écart à la moyenne dans une fenêtre coulissante de 17×17 .

L'algorithme de Canny (1986) est populaire car il donne des contours fins, peu bruités. Son paramétrage est plus compliqué que celui de Sobel, car il demande le choix d'un paramètre de lissage et de deux seuils. Pour ce détecteur, nous convenons d'effectuer le seuillage double avec des valeurs dont l'une vaut le double de l'autre, tel que suggéré dans la publication d'origine. Il nous faut aussi déterminer les masques pour le lissage. Nous utilisons plusieurs tailles de fenêtres : $N \in \{3, 5, 7, 9, \dots\}$. Par souci de simplicité numérique, nous avons choisi d'attribuer à chaque filtre un paramètre $\sigma = (N-1)/6$. Ainsi, par exemple, pour le cas $N = 3$, nous utilisons $\sigma = 1/3$. La précision induite par ce choix est comparable à celle des choix d'autres auteurs, par exemple (Rothwell, 1994).

Nous avons nommé « détecteur de Argyle » le détecteur défini par un filtrage gaussien suivi d'un seuillage simple de l'amplitude du gradient. Cet opérateur, suggéré dans (Argyle, 1971), équivaut à un détecteur de Canny sans post-traitement.

Les détecteurs de Kirsch, de Moon, ainsi que de Frei-Chen sont moins utilisés en pratique que les détecteurs cités plus haut. Ils sont mathématiquement différents des deux premiers détecteurs et permettent de valider nos résultats sur une plus grande variété d'algorithmes.

Le détecteur de contours « boussole » de Kirsch procède en convoluant l'image originale avec chacun des huit masques correspondant aux huit orientations possibles d'un modèle de contour sur un voisinage 3×3 . Pour chaque pixel, un indice de présence de contour est défini comme la réponse maximale sur les huit images de convolution. Ce détecteur produit des cartes semblables

à celles produites par le détecteur de Sobel, avec légèrement moins de fausses alarmes et une tendance à produire des contours plus fins.

Le détecteur de contours de Moon est un filtrage exponentiel suivi d'un seuillage sur l'amplitude du gradient. Notre implémentation applique un filtre exponentiel sur l'image et procède ensuite à un seuillage simple du gradient de Roberts (différences finies définies sur les diagonales d'une fenêtre 2x2). Pour le filtrage, nous utilisons des masques dont la largeur est un nombre impair de pixels, variant de $N = 3$ à $N = 15$.

Selon la librairie d'analyse d'images utilisée, le détecteur de Frei-Chen n'est pas toujours défini exactement de la même manière. Nous en avons relevé plusieurs. L'implémentation que nous utilisons est un seuillage sur un indice de présence de contours, défini par :

$$\sqrt{(I * E_1)^2 + (I * E_2)^2 + (I * E_3)^2 + (I * E_4)^2} \quad (3.5)$$

où * est l'opérateur de convolution, I est l'image originale, et E_1 , E_2 , E_3 , et E_4 sont les masques « contours » de Frei-Chen.

Il n'existe pas actuellement de méthode permettant de trouver les paramètres optimaux sans utiliser de vérité-terrain. Lors de son éventuelle mise au point, une telle méthode gagnerait à être intégrée à un algorithme de détection de contours. Pour choisir les seuils, nous avons exploré complètement l'espace paramétrique de chaque algorithme. Cette méthode évite au moins de biaiser les résultats en faveur d'un des algorithmes. (Heath *et al.*, 1997).

3.1.3 Choix d'indicateurs

Pour mettre au point l'indice de présence de contours, nous combinons plusieurs indicateurs. Pour bien caractériser les contours, cet indice doit pouvoir différencier les frontières des objets, de la texture et du bruit. Dans un premier temps, nous avons tenté d'utiliser les indices suggérés par Chalmond *et al.* (2001). Rappelons qu'il s'agit d'un filtre de Gabor symétrique à l'échelle $\sigma = 2$ selon quatre directions $\{\varphi_k, k = 1, 2, 3, 4\} = \{0, \pi/2, \pi/4, -\pi/4\}$, de l'entropie de deuxième ordre sur une fenêtre d'une largeur de 7 pixels et d'une estimation du rapport signal sur bruit à partir d'une fenêtre d'une largeur de 7 pixels. Le choix de ces indicateurs découle du souci de décrire les contours d'intensité de type « marches » (maximum de la dérivée première), les contours de texture et la qualité du signal.

Les filtres de Gabor sont utiles pour trouver les frontières entre les régions de différente intensité ou différente texture, à différentes échelles (Manjunath et Chellappa, 1993; Dunn *et al.*, 1995). En plus d'être sensibles aux contours, ils sont robustes aux variations d'illumination. Ils peuvent être utilisés comme indices de texture pour la recherche d'images (Manjunath *et al.*, 2000). Ils peuvent identifier des structures fines, et sont utilisés notamment, pour faire une restauration d'images d'empreintes digitales (Yang *et al.*, 2003). Les filtres symétriques sont sensibles aux contours de type « ligne » et les filtres antisymétriques, aux contours de type « marche ».

L'entropie renseigne sur l'hétérogénéité locale. Elle est faible sur une zone homogène, mais plus élevée sur contour de type « marche » ou de type « rampe ». Une zone très hétérogène contient probablement de la texture et ne devrait pas être classifiée comme « contours ».

Bien que l'hypothèse d'un bruit additif non-corrélé soit souvent faite, rien ne garantit que le bruit soit complètement indépendant du signal, notamment dans le cas où un éclairage non-uniforme sature le signal. Pour obtenir une caractérisation optimale du signal, une estimation de la variation locale du rapport signal sur bruit peut être bénéfique.

La complexité des données correspond de façon approximative au rapport signal sur bruit de l'image (Suetens *et al.*, 1992). Une image simple a un fort contraste et est localement homogène. Plus le rapport signal sur bruit est bas, plus il est nécessaire de lisser l'image et moins le positionnement des contours est précis (Ziou et Tabbone, 1998). S'il y a beaucoup de bruit, le pixel a peut-être avantage à être classé comme texture au lieu de contours (Walessa et Datcu, 2000).

Pour compléter la banque de filtres, nous proposons l'essai de plusieurs échelles. L'examen visuel des images aéroportées et de leurs vérités-terrain permet d'identifier plusieurs contours d'intensité de type « lignes ». Pour mieux caractériser ces motifs, il nous paraît pertinent d'utiliser un filtre de Gabor antisymétrique, comme l'ont fait Konishi *et al.* (2003).

3.1.4 Calibration et entraînement de réseaux de neurones

La régression logistique utilisée dans l'article original de Chalmond *et al.* (2001) est un moyen utile pour décrire la relation entre un ou plusieurs éléments de risques et une conséquence pouvant prendre deux valeurs. Par exemple, la relation entre l'âge, le sexe et le taux de mortalité.

Chaque coefficient de régression décrit l'importance de la contribution de chaque élément. Cependant, la complexité d'un tel modèle étant limitée à un coefficient par variable, il n'est pas possible de décrire les interactions entre les variables en utilisant la régression logistique.

Pour la présente étude, nous voulions utiliser un algorithme d'apprentissage n'ayant pas cette restriction. Ainsi, pour évaluer les capacités d'une combinaison d'indicateurs à détecter les contours, nous avons choisi d'utiliser un réseau de neurones. Ce dernier peut être considéré comme une généralisation de la régression logistique, avec une architecture plus élaborée. Comme il n'est pas évident d'extraire d'une telle architecture une description de l'effet des indicateurs sur la variable indépendante, l'emploi du réseau de neurones sera accompagné d'une analyse de sensibilité.

Pour effectuer l'entraînement du réseau, les indicateurs considérés sont présentés à l'entrée du réseau. La sortie désirée correspond à la vérité-terrain des contours. Bien que les réseaux de neurones ne garantissent pas de trouver la solution optimale, ils permettent d'arriver facilement à une solution satisfaisante.

L'architecture choisie ici est celle du Perceptron Multicouches, avec des synapses linéaires en entrée et tangente hyperbolique en sortie (LeCun *et al.*, 1998). L'apprentissage est fait avec l'algorithme de rétropropagation stochastique. Le lecteur intéressé à la théorie des réseaux de neurones peut consulter un des nombreux bons ouvrages sur le sujet, par exemple (Haykin, 1999). Nous n'expliquons ici que notre utilisation de cet outil.

Les paramètres de taux d'apprentissage η et la constante d'inertie μ sont fixés aux valeurs arbitraires de $\eta = 0,1$ et $\mu = 0,1$. Notre souci est d'atteindre des minimums locaux suffisamment petits, en accélérant un peu la vitesse de convergence, tout en conservant une stabilité dans la courbe de performance. Un réseau est considéré comme ayant convergé quand le taux absolu de changement de l'erreur moyenne quadratique par époque est suffisamment petite, c'est-à-dire 0,01 % pour cette expérience.

Pour accélérer la convergence lors de l'entraînement, les recommandations de LeCun *et al.* (1998) ont été suivies :

- utiliser l'apprentissage stochastique, c'est-à-dire mettre à jour les poids du réseau après la présentation de chaque exemple;
- faire varier aléatoirement l'ordre de présentation des exemples;
- utiliser une fonction sigmoïde de forme tangente hyperbolique;
- choisir des valeurs-cibles correspondants aux maximums de la dérivée seconde de la sigmoïde, soit ± 1 .

Les données d'entraînement sont séparées en trois groupes :

- un jeu d'entraînement;
- un jeu de validation;
- un jeu de test.

La différence entre les résultats attendus et les résultats obtenus s'appelle l'*erreur*. L'erreur sur le jeu d'entraînement est mesurée à chaque itération d'entraînement et sert à ajuster les poids du réseau. L'erreur jeu de validation est elle aussi mesurée à chaque itération. Quand le réseau a convergé, la configuration ayant la plus petite erreur de validation est conservée, pour éviter qu'il n'y ait surapprentissage. Le jeu de test est constitué des images non-présentées au réseau lors de l'entraînement. Ces images seront utilisées pour l'évaluation de la méthode. Elles permettront, lors de la prochaine étape de l'expérimentation, d'estimer la capacité du réseau à généraliser.

Nous avons utilisé « peu » de données pour l'entraînement et réservé la majorité des données pour les tests. Ce choix de partitionnement des données a été fait en considérant que, dans les faits, il est difficile d'obtenir une vérité-terrain de contours pour l'entraînement. Ainsi, pour effectuer le partitionnement des données, la moitié des pixels d'une image sont employés pour définir le jeu d'entraînement. L'autre moitié sert pour le jeu de validation. Ainsi, une image entière est utilisée pour paramétrer le réseau. Par la suite, toutes les autres images peuvent être employées pour des tests.

Lors de tout entraînement d'un réseau de neurone, il est nécessaire d'éviter le phénomène de surapprentissage. Si un réseau est entraîné pendant un trop grand nombre d'époques sur l'ensemble de données d'entraînement, ses performances sur cet ensemble de données seront

excellentes. Cependant, il sera trop spécialisé par rapport aux spécificités de cet ensemble et au bruit qu'il contient. Dans une telle situation, le réseau aura peu de pouvoir de prédiction sur les ensembles de données de validation et de test. Pour avoir un réseau qui généralise bien, il faut arrêter l'apprentissage quand l'erreur sur les données de validation est minimale.

Plusieurs stratégies sont utilisables pour éviter ce surapprentissage, notamment la sélection de modèle d'architecture de réseau, l'ajout de bruit, la décroissance de poids, l'apprentissage bayésien, la combinaison de réseaux et l'arrêt hâtif (Haykin, 1999; Sarle, 2002). Nous utiliserons l'arrêt hâtif, pour sa simplicité et le petit nombre de répétitions d'entraînements qu'il nécessite.

Pour appliquer la technique de l'arrêt hâtif, l'apprentissage par rétropropagation du gradient est effectué selon la technique usuelle. Après chaque époque, l'erreur sur les données de validation est mesurée. Le réseau est entraîné jusqu'à convergence de l'erreur sur les données d'entraînement. Bien que d'autres critères d'arrêt soient possibles, Prechelt (1998) rapporte qu'aucun n'a d'avantage significatif sur les autres. Après l'entraînement, c'est le réseau tel qu'il était à l'itération donnant l'erreur de validation minimale qui est conservé. Il est nécessaire d'utiliser « beaucoup » de neurones cachés pour éviter les minimums locaux et le sous-apprentissage (Sarle, 2002). La Figure 3.3 présente l'architecture du réseau.

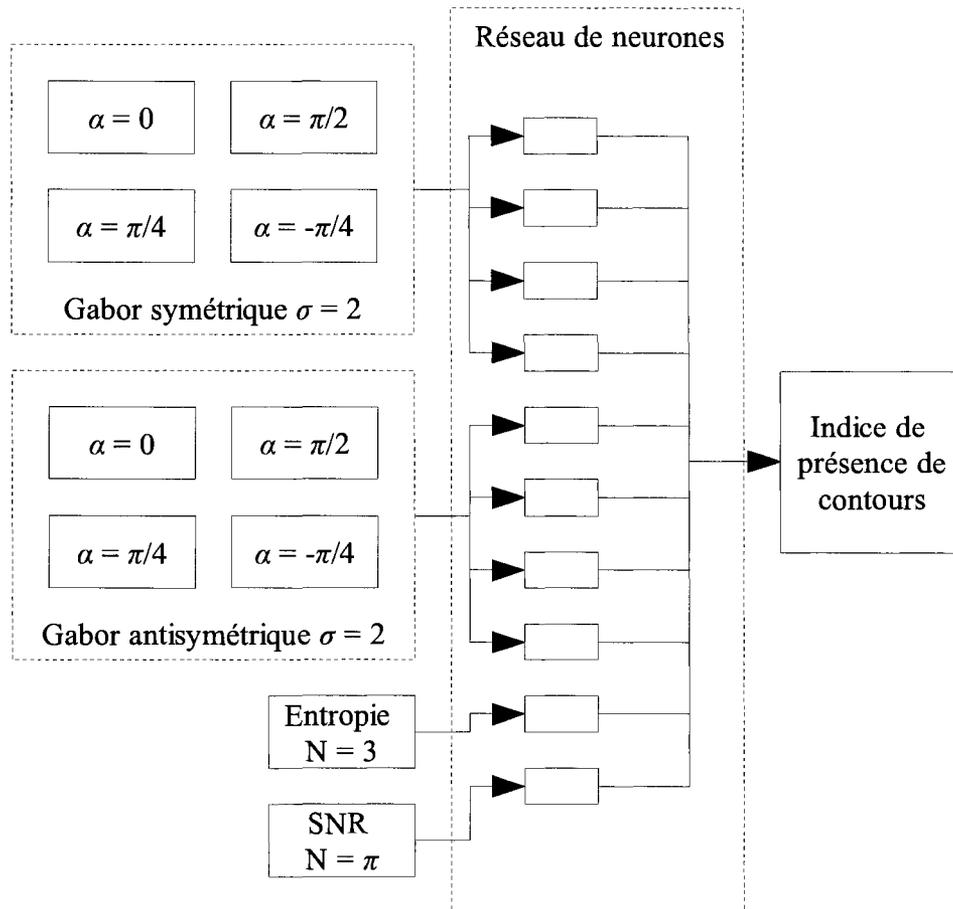


Figure 3.3 – Architecture du réseau de neurones

3.1.5 Restauration de cartes de contours

Avec cette étape, nous proposons de combiner l'information obtenue par réseau de neurones avec l'information « classique » sur les contours, plus précisément l'amplitude du gradient. Nous cherchons une méthode simple pour réduire le nombre de fausses alarmes dues à la texture tout en conservant les bordure des objets. Deux articles scientifiques ont retenu notre attention.

Rosenfeld (1970) s'est intéressé à la détection de contours par différence de moyenne mobile. Il a observé qu'en multipliant les images de contours à différentes échelles, le résultat était de meilleure qualité qu'aux échelles individuelles. Le produit est élevé seulement quand tous ses facteurs sont élevés. Cela a deux effets principaux, selon qu'une petite fenêtre ou une grande fenêtre est considérée.

Dans le cas d'une petite fenêtre de lissage, quand un point est décalé par rapport à un contour

quelconque, le résultat n'a pas une valeur élevée. Dans le produit, ce facteur limite l'effet de flou dû aux fenêtres plus grandes. Dans le cas d'une grande fenêtre de lissage, quand un point n'est pas rapproché spatialement d'un contour principal, le résultat n'a pas une valeur élevée. Dans le produit, ce facteur limite l'effet de faux contours dû aux fenêtres plus petites. Ainsi, puisque ces deux types d'erreurs sont évitées, les contours obtenus sont localisés plus précisément qu'aux échelles grossières et il y a moins de bruit qu'aux échelles fines.

Bao *et al.* (2005) se sont intéressés à la détection de contours basée sur le produit des réponses de la dérivée première d'une gaussienne à deux échelles. Les cartes de contours sont obtenues par un seuillage simple. La technique est évaluée analytiquement dans le cadre des deux critères de Canny jugés les plus utiles; le produit de ces deux critères a une valeur plus élevée pour la multiplication d'échelles que pour une échelle unique. Une évaluation empirique est aussi effectuée, d'abord sur une image synthétique, ensuite sur deux images terrestres. Les avantages observés sont les mêmes que dans le cas de Rosenfeld.

Ainsi, à l'instar de Rosenfeld (1970) et Bao *et al.* (2005), nous optons pour combiner nos cartes de contours en prenant la multiplication pixel par pixel des deux images. Pour appliquer la méthode au détecteur de Sobel, de Kirsch ou de Moon, la multiplication est faite avant le seuillage simple (Figure 3.2, précédemment). Dans le cas de l'algorithme de Sobel, d'autres essais sont faits avec un seuillage adaptatif basé sur l'écart à la moyenne dans une fenêtre coulissante de 17x17. Pour appliquer la méthode au détecteur de Canny, la multiplication pixel par pixel avec la sortie du réseau de neurones est effectuée avant l'étape de la suppression des non-maximums. Pour cette phase de validation, nous utilisons les dix imageries aéroportées.

3.1.6 Caractérisation de la robustesse au bruit

Aucune méthode n'est optimale dans toutes les conditions d'utilisation. Il est donc nécessaire de connaître la dégradation des performances de la méthode développée en présence de bruit. Déterminer la vraie position des contours dans une scène naturelle peut sembler très difficile. C'est pourquoi plusieurs chercheurs emploient des images synthétiques et la vérité-terrain correspondante pour évaluer leurs algorithmes. Plusieurs chercheurs optent pour l'utilisation d'une telle démarche, même si les résultats obtenus ne reflètent pas nécessairement le comportement des détecteurs de contours sur des images naturelles.

Le contenu des images synthétiques parfaitement contrôlé facilite la mise en place d'un protocole expérimental pour étudier, par exemple, la sensibilité des détecteurs de contours au niveau de bruit (Kanungo *et al.*, 1995). La qualité de résultats obtenus sera évalué avec la mesure-F, mais aussi avec la figure de mérite d'Abdou et Pratt (1979), qui est souvent utilisée pour estimer l'exactitude des détecteurs sur les contours dans les images bruitées. La valeur choisie pour la constante pénalisant les contours décalés est $\beta = 10$ (Gagnon et Jouan, 1997).

Nous utilisons aussi la mesure-F, comme pour les autres manipulations; cela permet de vérifier partiellement que l'optimisation par la figure de mérite de Pratt donne des résultats cohérents avec l'optimisation par la mesure-F.

Le détecteur le plus prometteur, utilisé avec et sans restauration, est considéré. Pour chaque cas, le seuillage donnant la meilleure figure de mérite de Pratt est déterminé expérimentalement. Cette opération est répétée sur des images synthétiques avec des rapports signal sur bruit variant entre 1 et 100.

Un essai est aussi effectué, sur les détecteurs de contours les plus prometteurs, pour la restauration d'images réelles bruitées, en ajoutant du bruit gaussien additif d'écart-type 25 à deux des images aéroportées, avant de leur appliquer la procédure de restauration.

3.1.7 Comparaison avec d'autres jeux de données

Pour assurer une meilleure validation de la méthode développée, il est souhaitable d'appliquer la technique développée à un autre domaine d'application et de comparer les résultats obtenus. Tout d'abord, il faut vérifier la capacité de la méthode à donner d'aussi bons résultats sur d'autres domaines d'application. Pour ce faire, nous appliquons la méthode de restauration à plusieurs images du jeu de photos d'objets urbains et domestiques. L'algorithme de détection de contours initiale utilisé est celui ayant donné les meilleurs résultats sur les images aéroportées. La validation est ensuite complétée en appliquant aussi la méthode à des images du jeu de données de segmentation/classification de Berkeley.

Ensuite, pour vérifier l'applicabilité de la méthode développée à l'imagerie satellite, tous les détecteurs de contours ayant démontré une amélioration sur les données aéroportées sont testés sur deux parties de scènes IKONOS.

3.1.8 Analyse de sensibilité

Une fois la méthode développée et validée, une question prend de l'importance, celle de savoir pourquoi la méthode réussie ou pas à fournir des résultats. Afin d'apporter des éléments de réponse, une analyse de sensibilité est indiquée. Dans un premier temps, il s'agit de produire les indicateurs individuellement et d'étudier leur capacité à détecter les contours, en évaluant la mesure-F maximale que chacun peut produire sur une image. Nous saurons ainsi si chacun des indicateurs est un bon filtre de détection. Il faut ensuite étudier la contribution de chacun des indicateurs au sein du réseau de neurone. Cela permet de déterminer s'il est nécessaire d'utiliser tous les signaux.

Pour l'étude de la capacité des indicateurs à détecter les contours, quelques ajustements sont nécessaires, afin de s'assurer d'avoir une seule bande de signal, avec de grandes valeurs numériques en présence de contours. Dans le cas de l'indicateur de Gabor, l'énergie isotrope a été utilisée comme estimateur de présence de contours. L'entropie a pu être utilisée sans modification. Il a été nécessaire d'inverser le signe du rapport signal à bruit, car une faible valeur de cet indicateur indique une forte probabilité de trouver un contour. Les indicateurs ont été utilisés, avec et sans fusion, pour détecter les contours sur l'image IKONOS du campus.

Par la suite, pour étudier l'influence des intrants sur le réseau de neurones, chaque combinaison de signaux en entrée sera inhibée, à tour de rôle, d'un facteur variant de 0 à 100%. Ces résultats permettront de déterminer la combinaison d'indicateurs sur laquelle le réseau a appris et de se faire une idée de la robustesse du réseau à une perturbation du signal en entrée.

3.2 Ensembles de données

L'entraînement et l'évaluation d'un détecteur de contours basé sur une approche statistique nécessitent l'utilisation de vérité-terrain. Selon Heath *et al.* (1997), spécifier une vérité-terrain de contours est très difficile, car les images naturelles contiennent beaucoup de détails. Ainsi, nous avons priorisé la réutilisation de jeux de données. La disponibilité limitée de telles interprétations a été un facteur déterminant dans le choix final.

Nous avons acquis le jeu de données mis au point par l'équipe de travail de Bowyer (2000) comprend des photos noir et blanc d'objets de la vie quotidienne et des photos aériennes d'une base militaire. Ces données ont été utilisées pour entraîner et tester le réseau de neurones.

Ensuite, pour mieux tester la stabilité du détecteur, nous utilisons aussi une image synthétique contenant un contour vertical unique. Une étude de faisabilité est aussi faite sur un jeu de données de segmentation d'images au sol (Martin *et al.*, 2001). Pour compléter la validation de l'applicabilité de notre méthode, nous avons préparé des vérités-terrains de contours pour deux imageries tirées de scènes IKONOS panchromatiques. Toutes les images de ces jeux de données ont un niveau de quantification de 8 bits/pixels.

3.2.1 Photos aériennes et au sol

Ce jeu de données comporte une série de 10 imageries tirées de photos aériennes d'une base militaire, ainsi qu'une série de 50 images d'objets au sol, tirés de la vie quotidienne (Figure 3.4). Ces images ont été préparées à l'Université de South Florida. Elles furent acquises par Bowyer *et al.* (2001) pour étudier l'évaluation de détections de contours par des humains. Ces derniers ont éliminé quelques images contenant des objets difficilement identifiables. Les images restantes contiennent des objets naturels et des objets fabriqués. Certains de ces objets sont très texturés, d'autres le sont moins.



Figure 3.4 - Exemples d'images du jeu de données de South Florida
(a) four à micro-ondes, (b) vérité-terrain associée

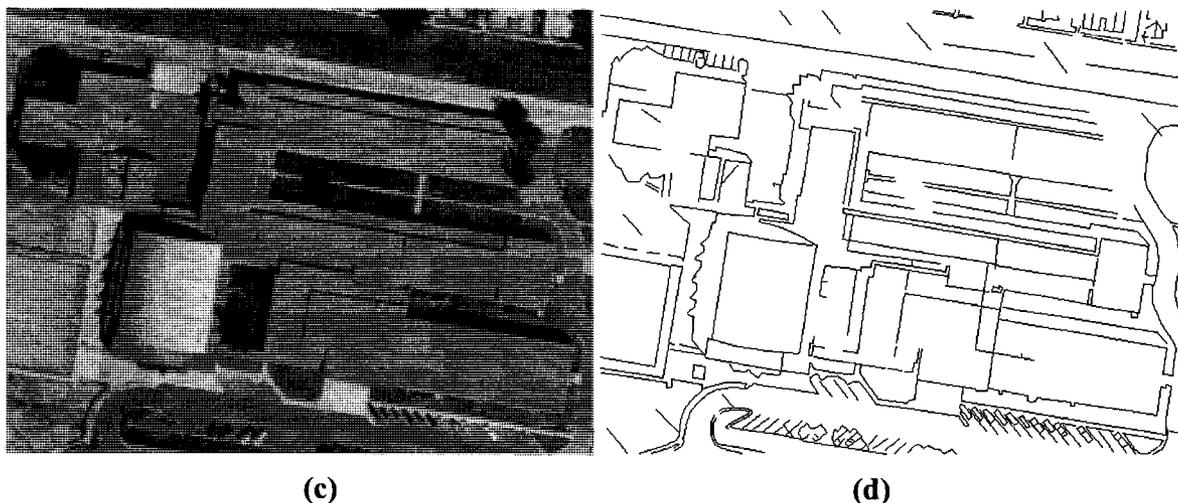


Figure 3.4 - Exemples d'images du jeu de données de South Florida (suite)
(c) école, (d) vérité-terrain associée.

Les photos ont été prises avec une caméra 35 mm sur film couleur avec une lentille de 50 mm. Les photos ont été scannées sur PhotoCD par un laboratoire commercial. Elles ont été extraites du CD en format 768x512, 24 bits/pixel. Elles ont été converties en tons de gris selon $0,299r + 0,587g + 0,114b$, où r est le niveau de rouge, g le niveau de vert et b le niveau de bleu.

Les images ont été rognées à un format d'environ 512x512. Finalement, le contraste et la brillance ont été ajustés pour bien paraître sur un écran d'ordinateur. La taille moyenne et la représentativité de ce jeu de données le rendent intéressant pour déterminer les indicateurs les plus prometteurs pour notre système.

Les contours associés aux images ont une largeur d'un seul pixel. Les contours flous n'étant pas inclus dans l'interprétation, la localisation des contours est précise (Konishi *et al.*, 2003). Bien que Bowyer *et al.* (2001) ne tiennent pas compte des zones texturées dans leur évaluations, nous choisissons de les considérer comme pixels d'arrière-plan, pour lesquels les détecteurs de contours ne devraient pas réagir. C'est aussi ce qu'ont fait Konishi *et al.* (2003). Suivant cette convention, les images aéroportées contiennent de 3,5 % à 6,3 % de pixels étiquetés « contour » et les images au sol en contiennent de 0,9 % à 7,6 %.

Hormis le bruit dû aux limites de la précision numérique, nous faisons l'hypothèse que les images sont contaminées par un bruit additif gaussien de moyenne nulle, dû au capteur. L'estimation

automatisée de la variance de ce bruit peut être faite avec la méthode de Meer *et al.* (1990). Nous avons adapté l'algorithme pour qu'il prenne en compte les images rectangulaires. Nous définissons le rapport signal sur bruit d'une image comme :

$$SNR_{image} = \sigma_{image}^2 / \hat{\sigma}_{bruit}^2 \quad (3.1)$$

Dans le jeu de données de South Florida, les images aériennes ont toutes un rapport signal sur bruit inférieur à 15. Dans les images d'objets au sol, 8 ont un rapport signal à bruit faible, compris entre 5 et 15. Un rapport signal à bruit élevé, compris entre 15 et 30, a été mesuré sur 19 images. Les 20 dernières images contiennent des zones parfaitement homogènes.

3.2.2 Images artificielles

Pour reprendre une expérience classique d'Abdou et Pratt (1979) visant à comparer les performances de détecteurs de contours, nous avons créé une image artificielle de 512x512 dont la moitié gauche est une région uniforme de niveau de gris 145 et la moitié de droite, 110. Ces valeurs ont été choisies afin d'éviter la saturation après l'ajout de bruit. Lors de leur étude sur le filtrage du chatoiement, Gagnon et Jouan (1997) utilisent plutôt les valeurs 200 et 50. Dans ce contexte, nous définissons le rapport signal sur bruit, SNR, par :

$$SNR = (h/\sigma)^2 \quad (3.2)$$

où h est la hauteur du contour et σ est l'écart-type du bruit. Du bruit gaussien a été additionné aux images afin d'obtenir des rapports signal sur bruit de 1, 5, 10, 20, 50 et 100, ces valeurs étant choisies pour faciliter la comparaison avec les publications antérieures (Figure 3.5). Pour ce faire, le filtre de « dispersion RVB » a été appliqué avec, respectivement, les valeurs 0,27; 0,12; 0,09; 0,06; 0,04 et 0,03 pour le paramètre « gris ». Les images artificielles seront utilisées pour évaluer l'effet du niveau de bruit sur la méthode développée.

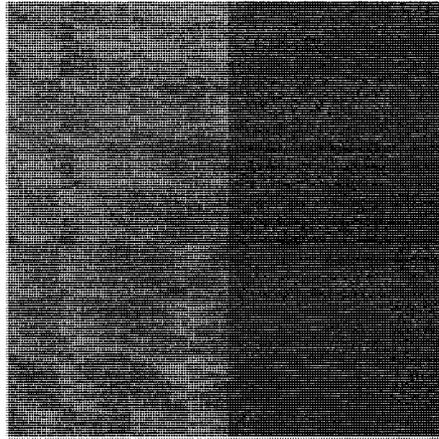


Figure 3.5 - Image artificielle, SNR = 5

3.2.3 Images de Berkeley

Ce jeu de données, comprenant 200 images d'entraînement et 100 images de tests, a été mis au point par Martin *et al.* (2001). Ce sont des photographies de scènes naturelles prises au sol dans les bandes visibles. Elles proviennent de la base de données de Corel et sont de dimension 481x321. Chaque image contient au moins un objet identifiable et a été segmentée par plusieurs humains pour créer une vérité-terrain composée. Ces différentes segmentations sont hautement cohérentes entre elles.

Selon Martin (2002), le passage au noir et blanc n'affecte pas significativement la capacité de sujets humains à séparer les images en régions cohérentes. Nous avons simplifié le problème en fusionnant les trois bandes selon l'espace de couleur du National Television System Committee (NTSC), défini par :

$$Y = 0,299r + 0,587g + 0,114b \quad (3.3)$$

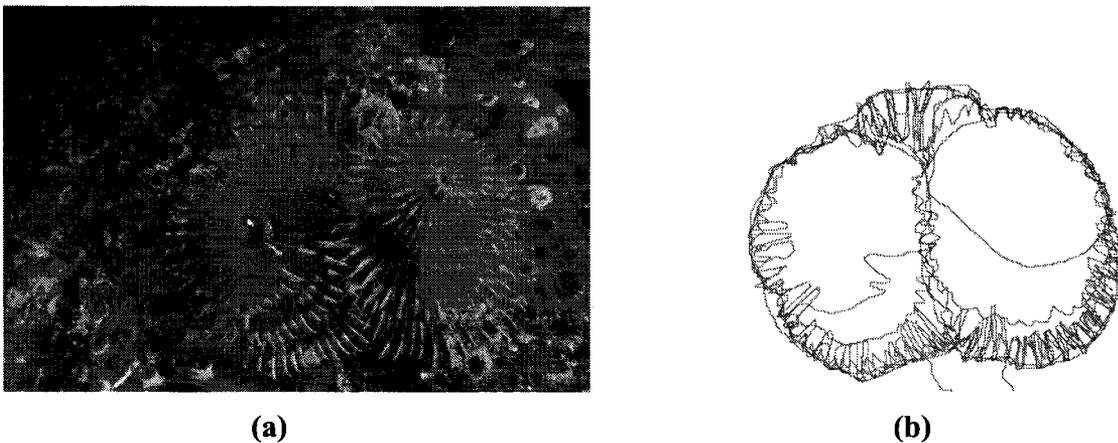
où Y est la luminance, r le niveau de rouge, g le niveau de vert et b le niveau de bleu, créant ainsi des images panchromatiques en noir et blanc (Konishi *et al.*, 2003).

Les vérités-terrain associées aux images de Berkeley contiennent des traits définissant les limites d'objets de haut-niveau. Ces objets sont composés de plusieurs segments de bas-niveau. Les détails des objets sont très souvent ignorés, ce qui n'est pas le cas pour les images de South Florida. Le jeu de données de Berkeley contient donc une interprétation implicite des objets de

l'image. Il décrit le résultat de groupements de segments en objets de haut-niveau de la scène. Les éléments qu'il décrit sont plus près de la classification que de la détection de contours.

Dans le jeu de données de Berkeley, la caractérisation du bruit selon la méthode de Meer *et al.* (1990) a permis d'identifier deux familles d'images bruitées. La première est composée d'images très bruitées, qui contiennent du sable granuleux ou de l'herbe texturée en gros plan. La seconde est composée d'images moyennement bruitées, avec une bonne séparation entre les zones de signal et les zones de bruit. Ces images ont été acquises sous l'eau, avec un faible éclairage. Le bruit dû au capteur et à la faible dynamique des niveaux de gris y est clairement visible.

Trois images « difficiles à traiter » ont été choisies dans ce jeu de données (Figure 3.6). La première est un banc de corail (12084). Sa vérité-terrain ignore de petits objets fortement contrastés. Elle contient 7,5 % de pixels étiquetés « contour » par au moins un interprète. La deuxième image, un aménagement paysager (86016), contient une texture granuleuse. Sa vérité-terrain indique la présence de sillons sur le sol. Elle contient 10,3 % de pixels étiquetés « contour » par au moins un interprète. La troisième image est une photo de pyramides (260058). Les contours des nuages y sont décrits comme contours, mais pas la texture du sable. Cette image contient 5,8 % de pixels étiquetés « contours » par au moins un interprète.



**Figure 3.6 - Images du jeu de données de Berkeley
(a) fond marin, (b) vérité-terrain associée**

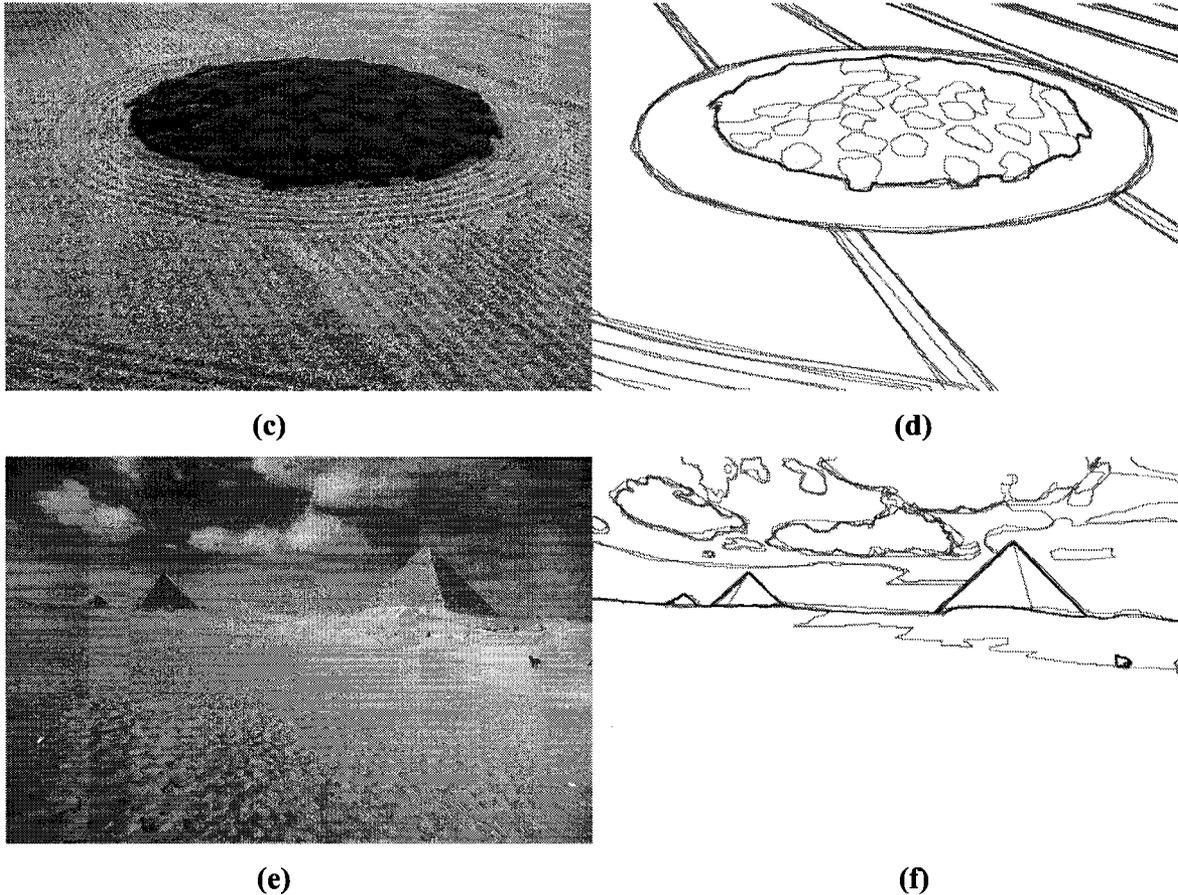


Figure 3.6 - Images du jeu de données de Berkeley (suite)
(c) aménagement paysager, (d) vérité-terrain associée,
(e) pyramides, (f) vérité-terrain associée.

3.2.4 Images IKONOS

Nous avons à notre disposition deux scènes IKONOS, une de Sherbrooke (Québec, Canada), l'autre de Saint-Hubert (Québec, Canada). Elles sont présentées dans les Figures 3.7 et 3.8. Pour la première, nous avons réutilisé une imagerie couvrant le campus principal de l'Université de Sherbrooke (Amani, 2006). Pour la seconde, nous avons sélectionné le site de l'Agence spatiale canadienne (512x512). Ces zones contiennent du bâti espacé.

Nous avons réalisé une vérité-terrain des contours pour chacune des images. Les contours saillants ont été retenus. Ils correspondent aux bâtiments, aux routes principales et aux ombres. Comme pour le jeu de données de South Florida, les contours flous ou de petite taille reliés à la

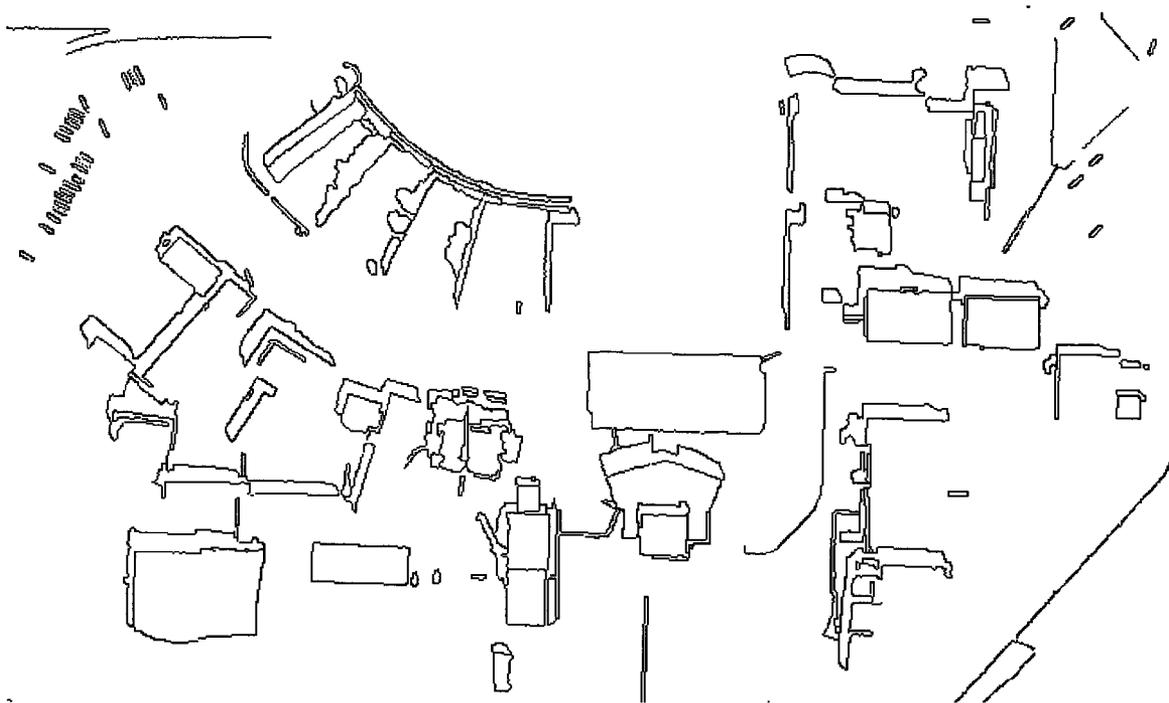
végétation ont été ignorés. L'image de l'Agence spatiale canadienne contient 5,2 % de pixels de contours; celle de l'Université de Sherbrooke en contient 3,8 %. Ces pourcentages sont comparables à ceux rencontrés dans le jeu de données de South Florida.



(a)

Figure 3.7 - Image IKONOS de l'Université de Sherbrooke

(a) image originale.



(b)

Figure 3.7 - Image IKONOS de l'Université de Sherbrooke (suite)

(b) vérité-terrain associée.

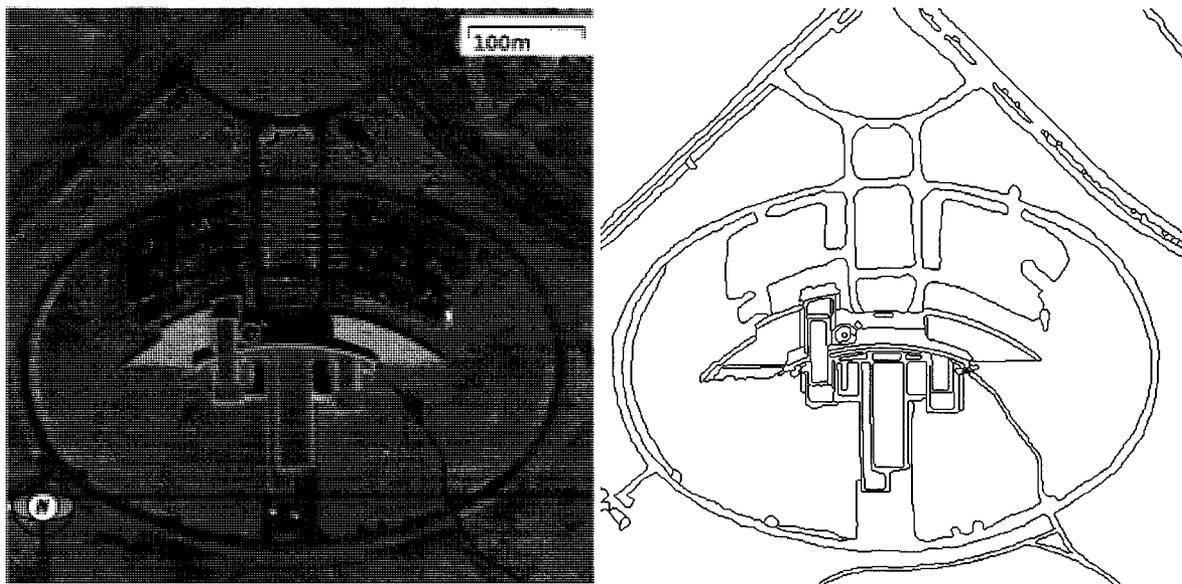


Figure 3.8 - Agence Spaticale Canadienne, image IKONOS et vérité-terrain associée.

3.3 Conclusion partielle

Les données utilisées pour tester notre algorithme sont très variées. La pertinence de cet algorithme est assurée par l'utilisation, dès le départ, de photos aériennes et terrestres pour l'entraînement supervisé. Les images artificielles de contours permettent ensuite de mieux étudier les propriétés du modèle implémenté et de confirmer la pertinence d'utiliser la mesure-F comme outil d'optimisation. Les images du jeu de données de Berkeley, par la différence dans la façon dont leur vérité-terrain est définie, permettent d'évaluer les limites de notre méthode. Finalement, les images IKONOS permettent de valider l'applicabilité de la méthode à des données satellites.

La méthode proposée permet d'utiliser un détecteur de contours classique, tout en tenant compte de la texture présente dans l'image. Cette approche est motivée par le désir de compenser les faiblesses de chaque approche : la sensibilité aux détails pour les méthodes classiques, et la difficulté à bien localiser les contours pour les méthodes utilisant la texture. Pour mieux quantifier le gain de performance que la nouvelle approche peut apporter, ses résultats sont comparés aux meilleures cartes de contours obtenues avec les méthodes classiques. Il faut en effet vérifier si les efforts d'un utilisateur final seraient mieux investis dans l'utilisation de la nouvelle méthode que dans le peaufinage des paramètres d'une méthode classique.

Les méthodes classiques sélectionnées pour créer les cartes de gradient de contours initiales ont été développées selon des paradigmes mathématiques différents. Elles incluent notamment les méthodes les plus utilisées en détection de contours. Collectivement, elles forment une bonne base pour évaluer la qualité de la méthode proposée.

4. Présentation des résultats

Le développement d'un logiciel pour l'amélioration de contours par l'emploi de réseaux de neurones est un résultat concret répondant au deuxième objectif de cette thèse (Figure 4.1). Il est décrit plus en détail dans l'Annexe 5. Ce chapitre décrit les ajustements nécessaires pour mettre au point un tel réseau de neurones, pour ensuite présenter en détail plusieurs aspects des performances obtenues lors de son utilisation.

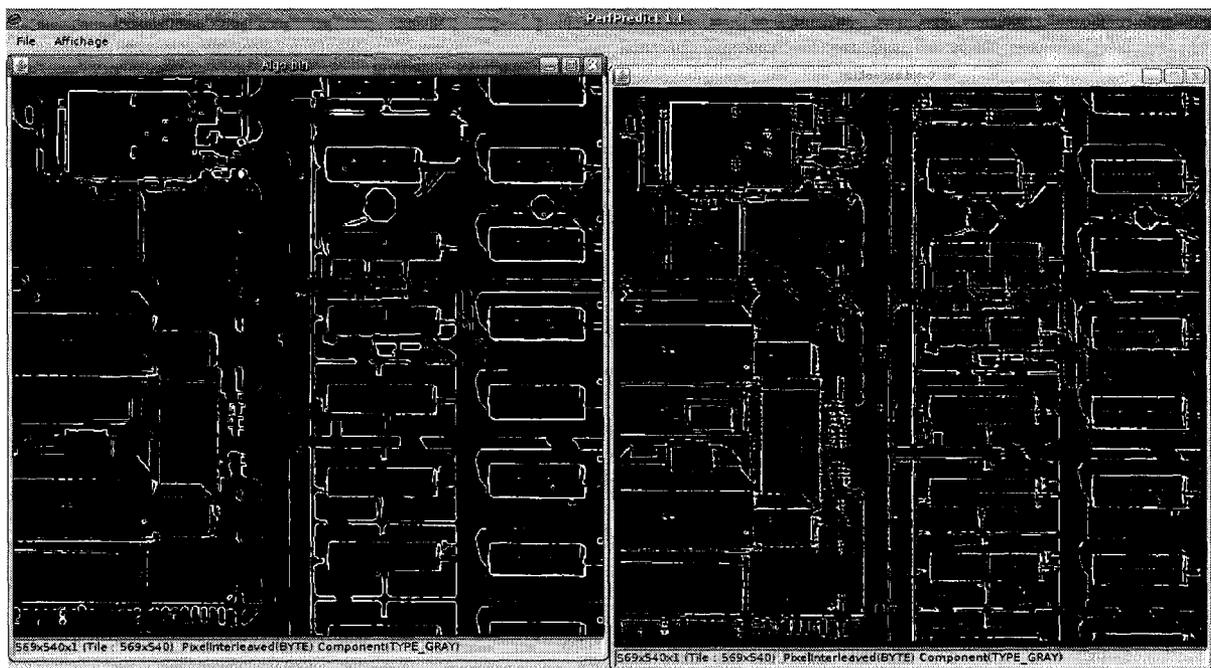
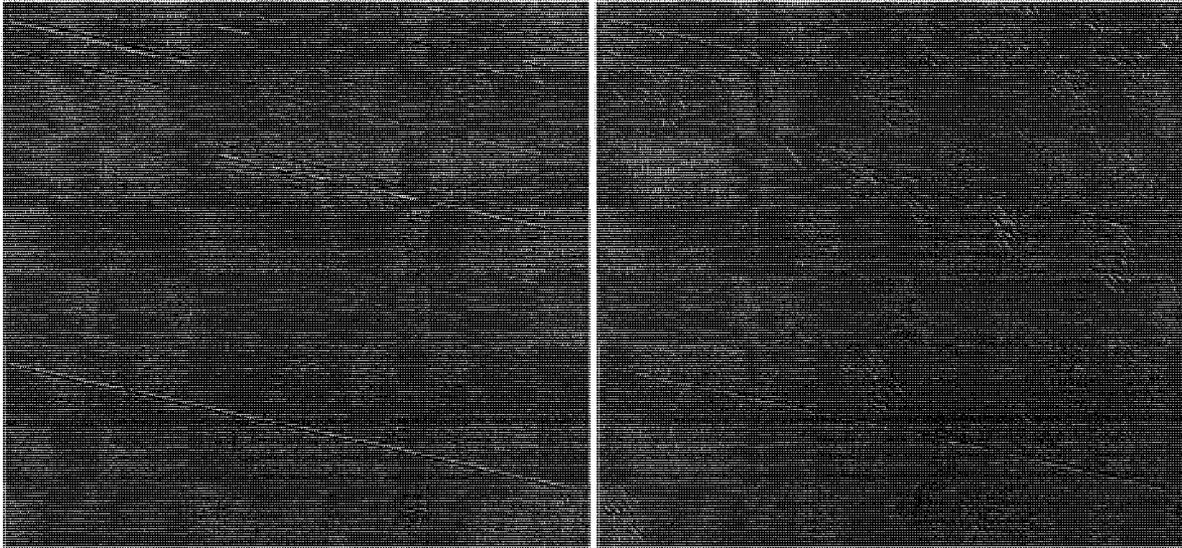


Figure 4.1 - Saisie d'écran du logiciel PerfAmel

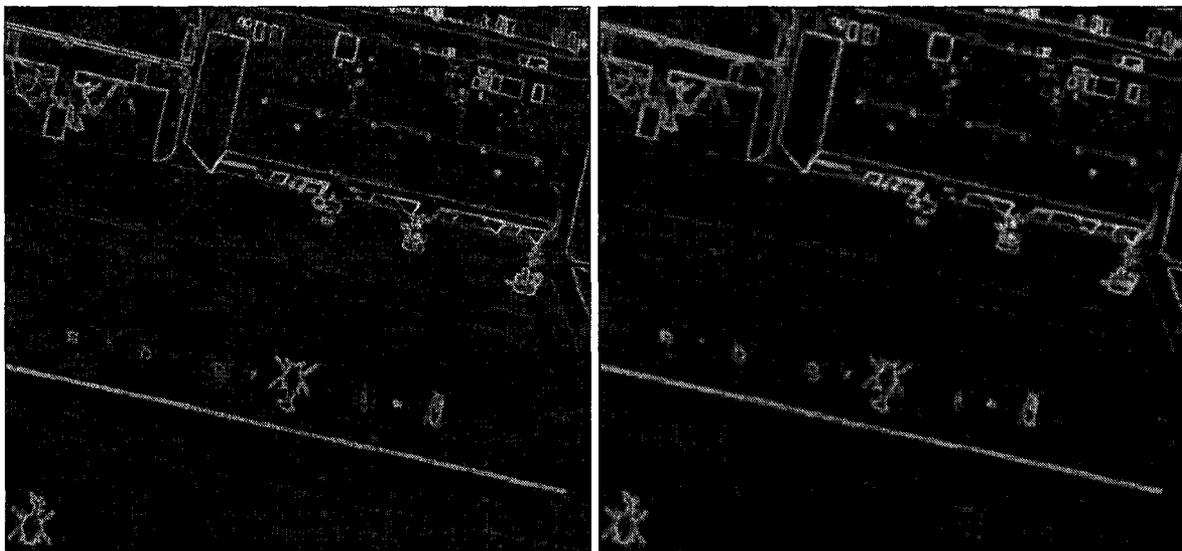
4.1 Mise au point du réseau de neurones

Suite à un premier entraînement avec les paramètres originaux de Chalmond (Gabor symétrique $\sigma = 2$, entropie $N = 7$, SNR $N = 7$, voir la Figure 4.2), les résultats n'étaient pas significatifs. Les images en sortie étaient bruitées et leur contraste était très faible, comme si le réseau ne voyait qu'une classe sur toute l'image (Figure 4.3).



(a)

(b)



(c)

(d)

Figure 4.2 - Indicateurs de Chalmond

(a) Gabor symétrique, $\sigma = 2$, $\varphi = +\pi/4$, (b) Gabor symétrique, $\sigma = 2$, $\varphi = +\pi/4$,
 (c) Entropie $N = 3$, (d) Entropie $N = 5$.

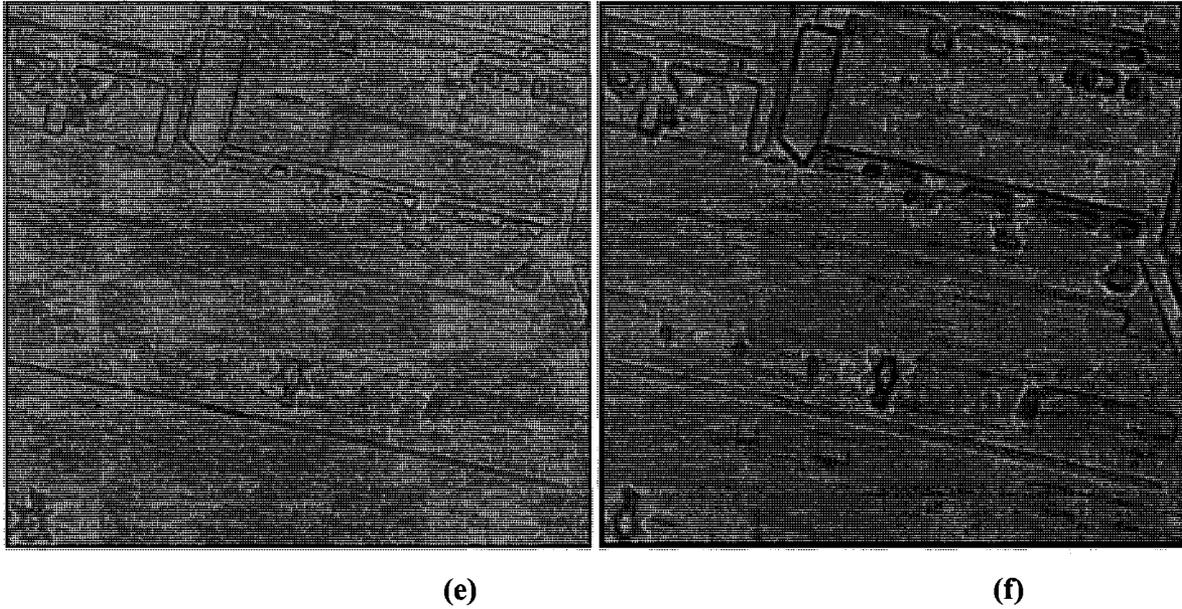


Figure 4.2 - Indicateurs de Chalmond (suite)
(e) Signal sur bruit $N = 7$, (f) Signal sur bruit $N = 11$.

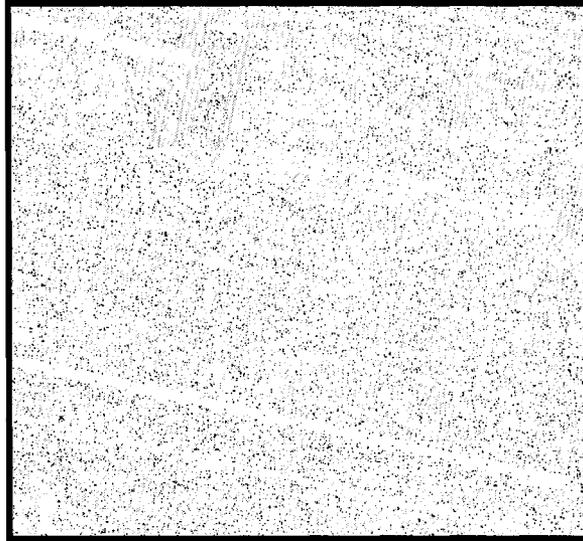


Figure 4.3 - Indice de présence de contours (faible contraste et bruit)

Les réseaux sont entraînés pour détecter les contours. Vu le petit nombre de pixels VP comparativement aux pixels VN, il s'est avéré nécessaire d'ajuster la composition des jeux d'entraînement pour éviter que les exemples VN dominent la fonction apprise. Tous les pixels VP de l'image sont utilisés pour l'entraînement. Un échantillonnage aléatoire de pixels VN est choisi

au début de l'entraînement. Les pixels VN situés à une distance inférieure ou égale à 2 pixels d'un VP ne sont pas utilisés. Les meilleurs résultats sont obtenus en prenant deux fois plus de pixels que le nombre de pixels VP. Lors d'une itération d'entraînement, autant de pixels VP que VN sont présentés au réseau; pour ce faire, les pixels VP sont utilisés deux fois.

Après cet ajustement des jeux d'entraînement, le réseau identifiait les contours sur l'image. Ces contours paraissaient larges sur les cartes de performances. Avec une réalité au sol, nous pouvons nous permettre d'être plus sévère sur le positionnement exact des contours.

Après examen visuel des images d'indicateurs, il apparaît souhaitable d'utiliser $N = 3$ comme échelle pour l'entropie. Cette échelle donne des contours de texture plus fins. Nous avons jugé souhaitable d'ajouter un filtre de Gabor antisymétrique, pour faciliter le rehaussement de lignes.

Les résultats obtenus avec la nouvelle combinaison d'indicateurs (Gabor symétrique et antisymétrique $\sigma = 2$, entropie $N = 3$, SNR $N = 7$) ont alors l'apparence de cartes de contours (Figure 4.4). Selon les tests effectués, utiliser un nombre de neurones cachés égal au nombre d'entrées du réseau, permet de limiter l'erreur due au modèle; en utiliser plus n'améliore pas la qualité des résultats.

Toutes les images produisent des résultats comparables. Cependant, il a été remarqué qu'un réseau de neurones entraîné sur l'image « Airfield » du jeu de photos aériennes donnait des résultats légèrement supérieurs à ceux des autres images du même groupe.

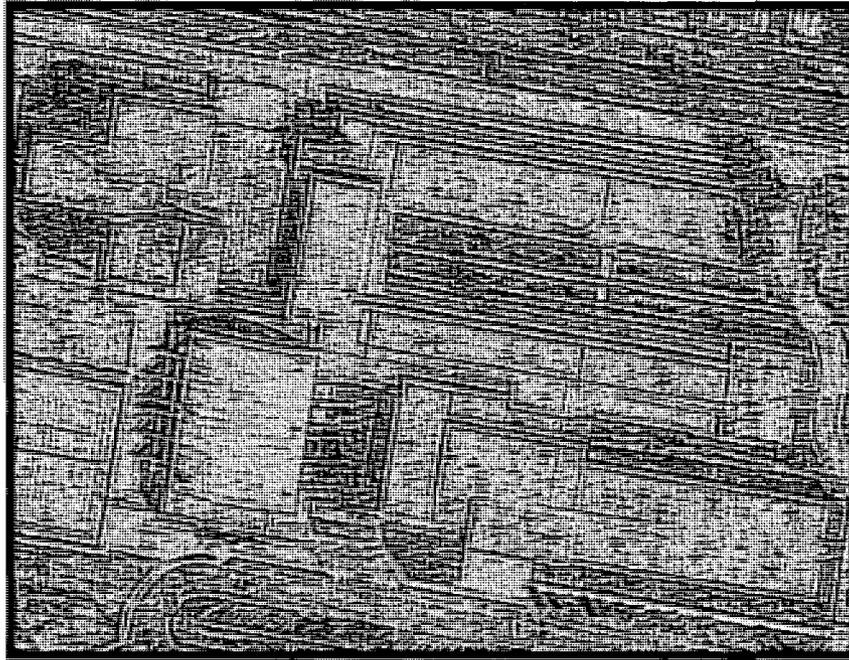


Figure 4.4 - Indice de présence de contours

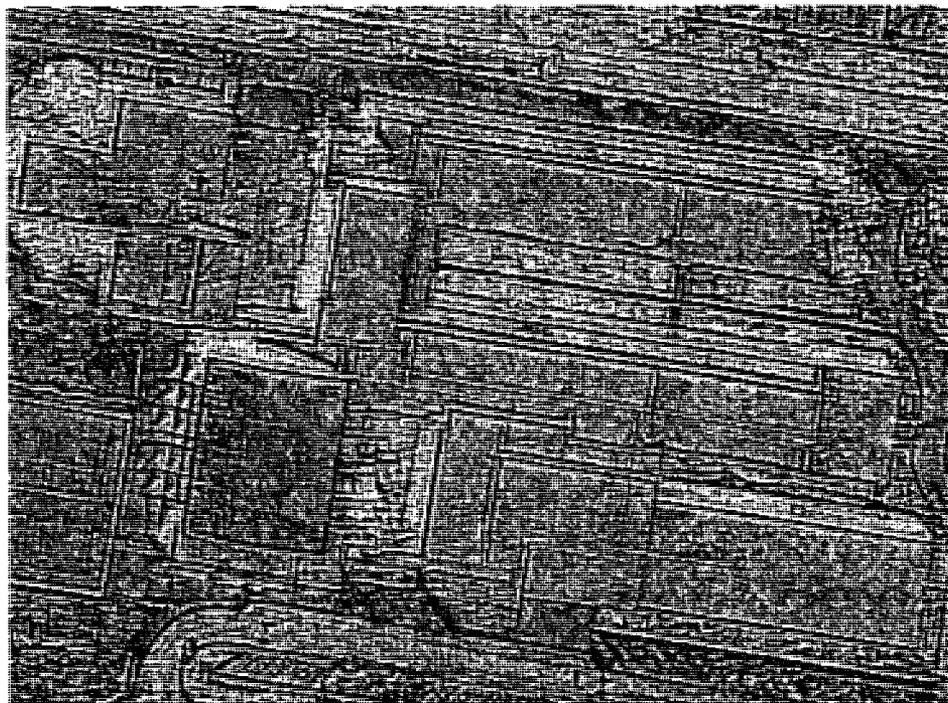
Même après optimisation de l'entraînement, les images générées par le réseau de neurones ne sont pas directement utilisables comme carte de contours. Pour obtenir des cartes utilisables, il est nécessaire de combiner les cartes créées par réseau de neurones avec des cartes classiques.

4.2 Restauration de cartes de contours

Dans cette section, nous décrivons les résultats obtenus lors des tentatives d'amélioration des détecteurs de Sobel, Kirsch, Moon, et Canny sur des images aériennes. La Figure 4.5 donne, à titre d'exemple, une image d'amplitude du gradient de Kirsch, avant et après multiplication par l'indice de présence de contours. Grâce à l'égalisation d'histogramme appliquée pour faciliter l'interprétation visuelle, un effet d'amincissement est facilement observé sur la deuxième image. La Figure 4.6 donne les deux mêmes images, après un seuillage global maximisant la mesure-F.



(a)



(b)

Figure 4.5 - Fusion de contours (niveaux de gris égalisés)
(a) Contours de Kirsch tels quels (b) Contours de Kirsch avec fusion

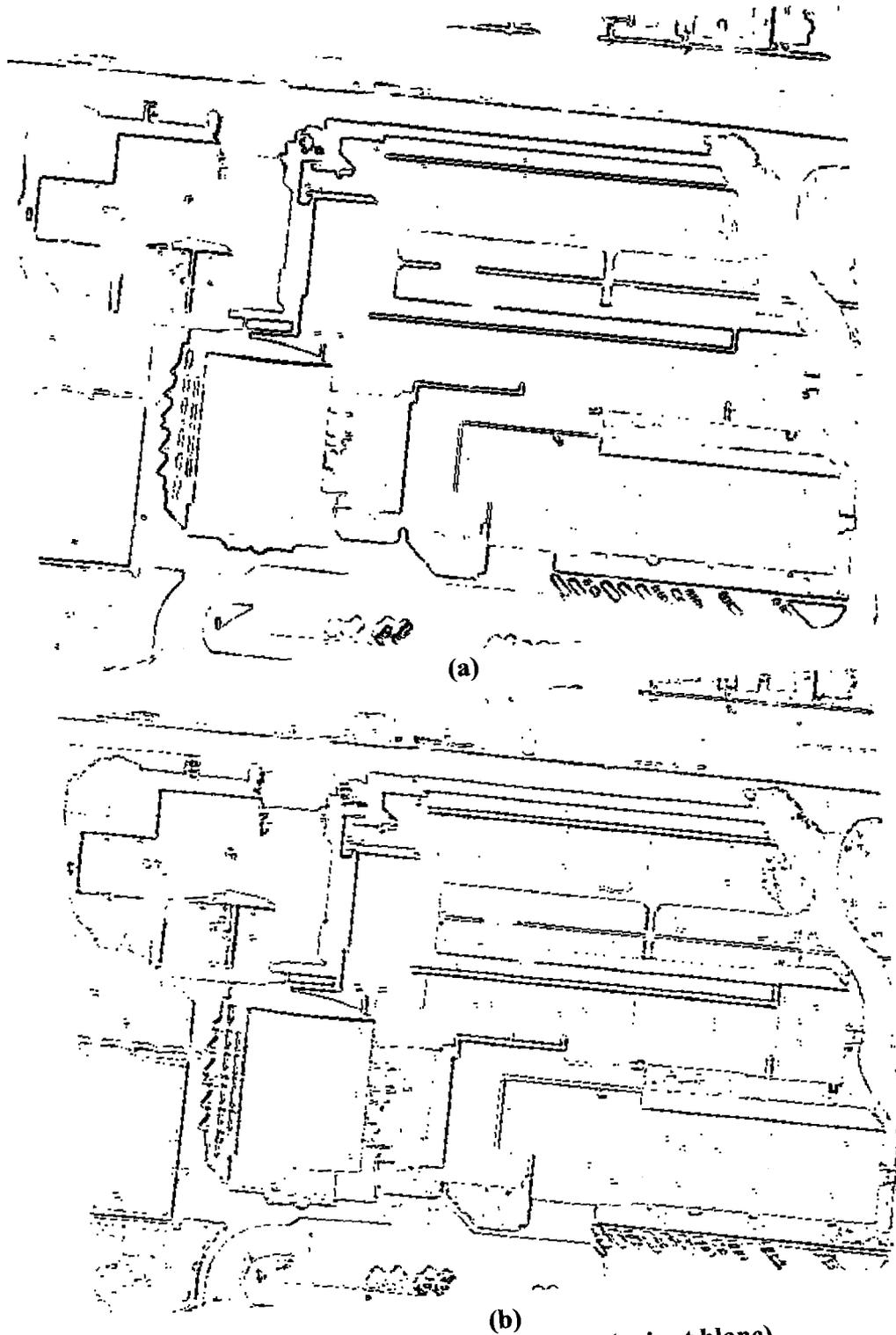


Figure 4.6 - Fusion de contours (noir et blanc)
(a) Contours de Kirsch tels quels (b) Contours de Kirsch avec fusion

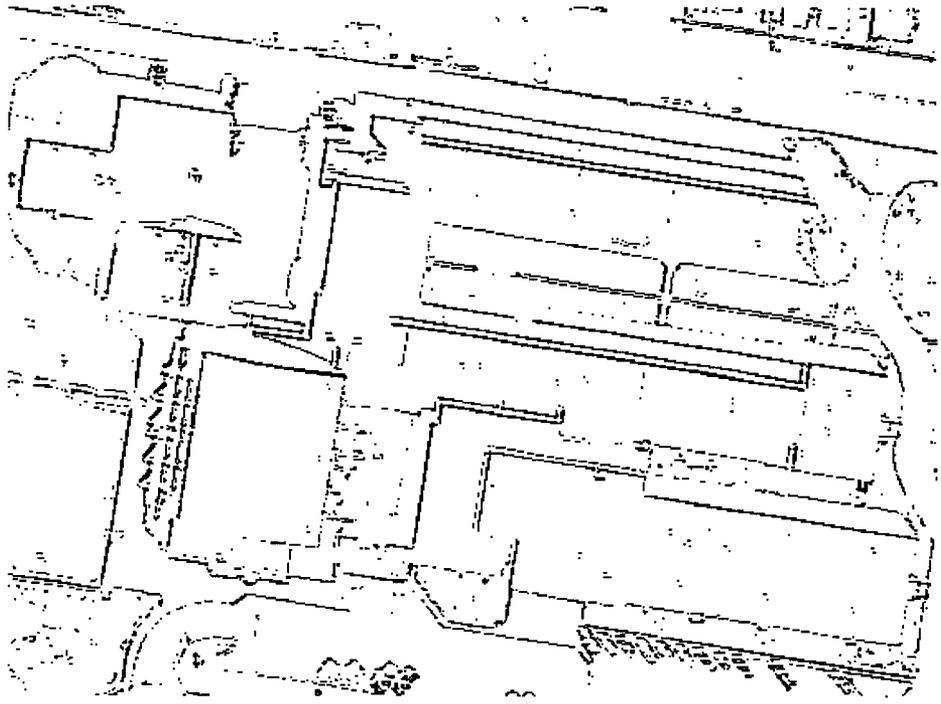
Les cartes obtenues par réseau de neurones ont été multipliées par celles obtenues par l'amplitude du gradient de Sobel. Les résultats obtenus ont une mesure-F systématiquement plus élevée que celle obtenue avec la méthode classique et le pourcentage d'amélioration médian est de 6,2 %. Le Tableau 1 présente ces résultats, accompagnés des résultats des détecteurs de Moon et Canny afin de faciliter la comparaison.

Tableau 1 - Mesure-F sur les images aériennes SouthFlorida (1)

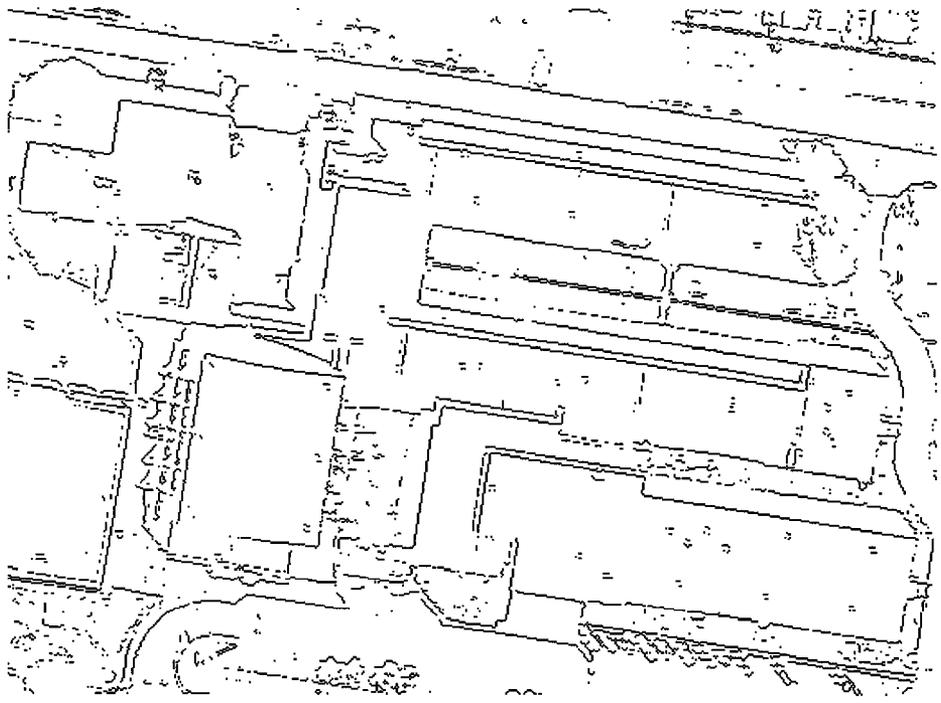
<i>Image</i>	<i>Sobel</i>	<i>Fusion</i>	<i>Différence</i>	<i>Moon (3x3)</i>	<i>Canny (1x1)</i>
airfield	41,5 % ±0,8 %	49,1 %	7,7 %	41,5 %	51,9 %
baseball	46,7 % ±0,8 %	54,2 %	7,5 %	50,4 %	59,3 %
building	59,9 % ±0,8 %	64,0 %	4,0 %	60,2 %	70,6 %
homes	43,7 % ±0,7 %	49,6 %	5,9 %	44,7 %	52,7 %
largebuilding	47,2 % ±0,7 %	52,7 %	5,4 %	49,2 %	58,9 %
mainbuilding	50,7 % ±1,0 %	57,1 %	6,4 %	52,3 %	61,0 %
pool_tennis	53,4 % ±0,8 %	58,0 %	4,6 %	54,5 %	64,2 %
school	55,7 % ±0,8 %	62,2 %	6,4 %	57,6 %	67,9 %
series	57,9 % ±0,7 %	65,3 %	7,4 %	60,1 %	75,3 %
wood	51,9 % ±1,0 %	55,7 %	3,8 %	54,2 %	61,2 %

Les résultats de Sobel et ceux de Sobel fusionnés sont tous deux améliorés par l'utilisation d'un seuillage adaptatif. La Figure 4.7 montre l'application, sur l'image « école », de l'algorithme de Sobel restauré, avec un seuillage global et avec un seuillage adaptatif. Comparé à l'algorithme classique avec seuillage adaptatif, l'algorithme restauré avec seuillage adaptatif conserve son avantage. L'application de la méthode de fusion à l'algorithme de type boussole de Kirsch et à l'algorithme de Frei-Chen donne aussi une amélioration (Tableau 2). Les pourcentages médians d'amélioration sont de 5,4% pour le détecteurs de Sobel adaptatif, 6.0 % pour Kirsch et 6,4 % pour Frei-Chen.

L'application de la méthode de fusion à l'algorithme de Moon donne des résultats équivalents à ce dernier. Les meilleurs résultats sont obtenus avec un filtre de dimension 3x3, bien adapté au faible niveau de bruit des images étudiées. Lors de l'application au détecteur de Canny, les résultats sont semblables au détecteur de Canny original, avec une légère détérioration.



(a)



(b)

Figure 4.7 - Stratégies de seuillage
(a) Contours de Sobel, seuillage global, (b) Contours de Sobel, seuillage adaptatif.

Tableau 2 - Mesure-F sur les images aériennes SouthFlorida (2)

<i>Image</i>	<i>Sobel Adapt.</i>	<i>Fusion</i>	<i>Diff.</i>	<i>Kirsch</i>	<i>Fusion</i>	<i>Diff.</i>	<i>Frei-Chen</i>	<i>Fusion</i>	<i>Diff.</i>
air	44,4 % ±0,8 %	52,2 %	7,8 %	41,7 % ±0,8 %	48,2 %	6,6 %	41,4 % ±0,8 %	49,0 %	7,6 %
baseball	49,0 % ±0,8 %	55,1 %	6,1 %	47,2 % ±0,8 %	53,1 %	5,8 %	46,4 % ±0,8 %	53,4 %	7,0 %
building	60,2 % ±0,8 %	64,6 %	4,5 %	60,2 % ±0,8 %	63,3 %	3,2 %	59,9 % ±0,8 %	64,1 %	4,2 %
homes	45,3 % ±0,7 %	50,7 %	5,4 %	43,7 % ±0,7 %	49,9 %	6,2 %	43,1 % ±0,7 %	49,5 %	6,4 %
large	50,7 % ±0,7 %	54,7 %	4,0 %	47,4 % ±0,7 %	52,2 %	4,8 %	47,0 % ±0,7 %	52,5 %	5,5 %
main	54,0 % ±1,0 %	59,1 %	5,1 %	50,7 % ±1,0 %	57,0 %	6,3 %	50,4 % ±1,0 %	56,8 %	6,4 %
pool	53,8 % ±0,8 %	58,9 %	5,1 %	53,6 % ±0,8 %	58,0 %	4,4 %	53,3 % ±0,8 %	58,1 %	4,8 %
school	57,2 % ±0,8 %	63,8 %	6,6 %	55,1 % ±0,8 %	61,6 %	6,5 %	55,6 % ±0,8 %	62,4 %	6,8 %
series	59,5 % ±0,7 %	67,5 %	7,9 %	58,2 % ±0,7 %	65,6 %	7,3 %	57,9 % ±0,7 %	65,5 %	7,6 %
wood	49,9 % ±1,0 %	55,4 %	5,5 %	51,8 % ±1,0 %	55,5 %	3,7 %	51,6 % ±1,0 %	55,7 %	4,1 %

4.3 Robustesse au bruit

La Figure 4.8 juxtapose la courbe de sensibilité au bruit du détecteur de Sobel avec celle du détecteur de Sobel avec fusion. Cette comparaison est faite avec la mesure-F et répétée avec la figure de mérite de Pratt ($\beta = 10$).

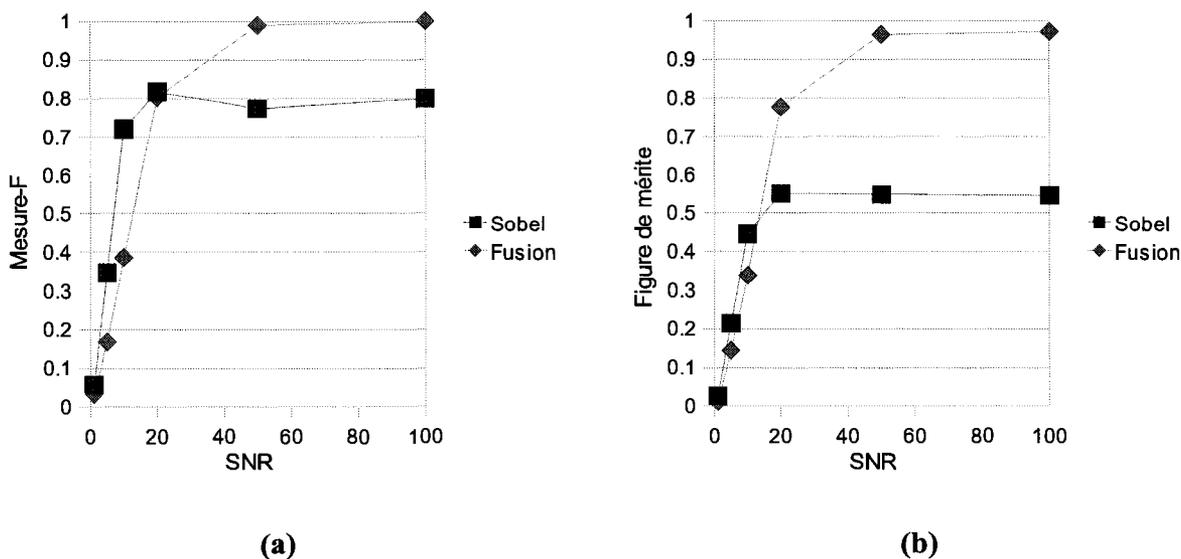


Figure 4.8 - Sensibilité au bruit

(a) Mesure-F, (b) Figure de mérite de Pratt ($\beta = 10$)

Des essais ont été effectués sur des images bruitées, avec les algorithmes les plus prometteurs, ceux de Sobel, Moon, Argyle et Canny. Dans un premier temps, nous avons ajouté du bruit gaussien additif d'écart-type 25 à l'image « Baseball ». Les résultats de la méthode de restauration sont présentés dans le Tableau 3. Dans un autre test, du bruit gaussien additif d'écart-type $\sigma = 25$ a été ajouté à l'image « Airfield ». Un réseau de neurones a été entraîné sur cette image. L'image « baseball » bruitée a été traitée avec ce réseau et l'algorithme de Moon. Le même réseau de neurones a été utilisé avec l'algorithme de Sobel pour traiter l'image « Baseball » non-bruitée (Tableau 4).

Afin de caractériser la façon dont le niveau de filtrage optimal est influencé par le niveau bruit, plusieurs résultats ont été rassemblés dans la Figure 4.9. Les résultats de l'algorithme de Canny et de Argyle y sont comparés, avec ou sans fusion, pour une plage de largeurs de fenêtre variant de $N = 1$ à $N = 15$ pixels.

Tableau 3 - Mesure-F sur l'image « baseball+bruit » SouthFlorida

<i>Algorithme</i>	<i>Original</i>	<i>Fusion</i>	<i>Commentaires</i>
Sobel	46,0 % $\pm 0,8$ %	43,5 %	Ajout de bruit
Moon 7x7	45,6 % $\pm 0,8$ %	41,9 %	Ajout de bruit
Moon 9x9	47,5 % $\pm 0,8$ %	43,1 %	Ajout de bruit
Moon 11x11	46,6 % $\pm 0,8$ %	45,5 %	Ajout de bruit
Moon 13x13	44,7 % $\pm 0,8$ %	46,4 %	Amincissement
Moon 15x15	43,0 % $\pm 0,8$ %	45,8 %	Amincissement
Canny 7x7	57,6 % $\pm 0,8$ %	52,0 %	Ajout de faux contours
Canny 11x11	60,6 % $\pm 0,7$ %	53,3 %	Ajout de faux contours
Canny 15x15	60,3 % $\pm 0,7$ %	50,2 %	Ajout de faux contours

Tableau 4 - Mesure-F après entraînement sur image bruitée

<i>Algorithme</i>	<i>Original</i>	<i>Fusion</i>	<i>Commentaires</i>
Moon 9x9 (image baseball bruitée)	47,6 % \pm 0,8 %	41,5 %	Ajout de bruit
Moon 13x13 (image baseball bruitée)	44,7 % \pm 0,8 %	45,5 %	Amincissement
Sobel (image baseball non bruitée)	46,7 % \pm 0,8 %	47,4 %	Amincissement

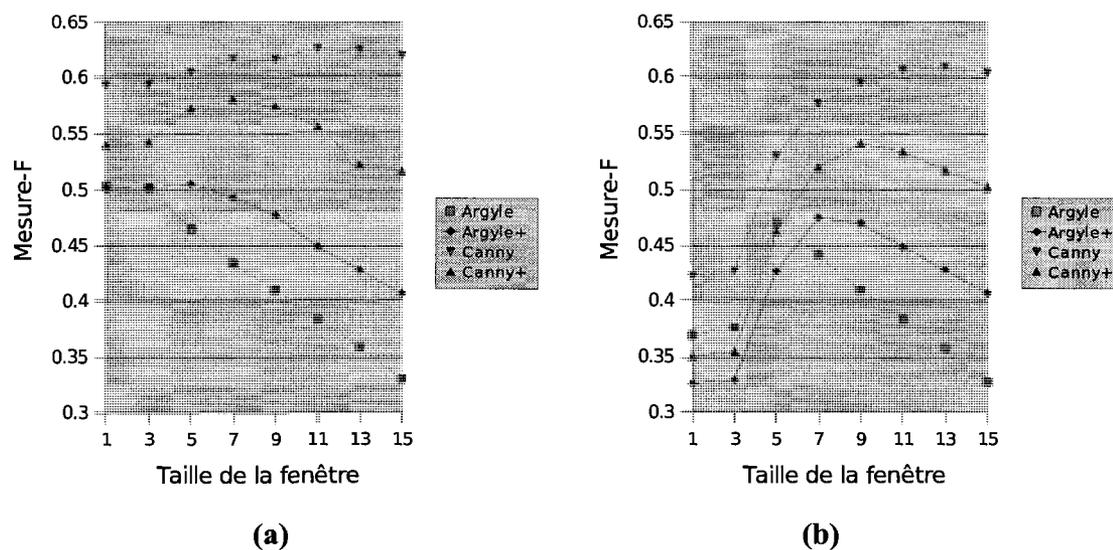


Figure 4.9 - Effet du lissage

(a) image « baseball », (b) image « baseball » et bruit gaussien d'écart-type 25.

4.4 Comparaison avec d'autres jeux de données

Pour la validation sur les images au sol, les tests sont effectués avec le détecteur qui a donné les meilleurs résultats sur les images aéroportées. Il s'agit du détecteur de Sobel. Le réseau de neurones entraîné sur les images aéroportées a été réutilisé. Les résultats sont présentés pour le seuillage global (Tableau 5) et le seuillage adaptatif (Tableau 6).

Tableau 5 - Mesure-F sur les photos au sol SouthFlorida (Seuillage simple)

<i>Image</i>	<i>Description</i>	<i>Sobel Simple</i>	<i>Fusion</i>	<i>Différence</i>
36	valise	54,7 % ±1,0 %	59,2 %	4,5 %
43	téléphone	60,1 % ±1,0 %	68,2 %	8,1 %
47	extincteur	55,4 % ±1,3 %	61,2 %	5,8 %
48	corbeille	61,2 % ±1,0 %	65,9 %	4,8 %
50	micro-onde	48,6 % ±0,9 %	57,9 %	9,3 %
56	motocyclette	51,4 % ±0,9 %	58,3 %	6,9 %
61	tomate	48,7 % ±1,3 %	50,9 %	2,3 %
62	chaudron	52,1 % ±1,2 %	60,1 %	7,9 %
101	fer à repasser	44,8 % ±0,7 %	54,6 %	9,8 %
103	rond de poêle	57,1 % ±0,8 %	62,0 %	4,9 %
143	ventilateur	57,2 % ±0,7 %	63,1 %	5,8 %
202	alligator	42,1 % ±1,2 %	39,0 %	-3,1 %
airplane	avion au sol	51,0 % ±0,8 %	58,7 %	7,7 %
block	bloc de béton	46,5 % ±1,1 %	53,6 %	7,1 %
coffee	cafetière	40,5 % ±0,9 %	46,3 %	5,8 %
cone	balise conique	39,4 % ±1,0 %	38,3 %	-1,1 %
egg	oeuf	59,5 % ±1,1 %	62,4 %	2,9 %
fountain	fontaine	45,5 % ±1,1 %	54,2 %	8,6 %
parking	parcomètre	47,3 % ±0,9 %	54,6 %	7,3 %
pitcher	pichet d'eau	52,7 % ±1,0 %	59,2 %	6,5 %
police	voiture	46,3 % ±0,9 %	49,7 %	3,4 %
stapler	agrafeuse	39,9 % ±1,1 %	42,6 %	2,7 %

Tableau 6 - Mesure-F sur les photos au sol SouthFlorida (Seuillage adaptatif)

<i>Image</i>	<i>Description</i>	<i>Sobel</i>	<i>Fusion</i>	<i>Différence</i>
56	motocyclette	52,6 % \pm 0,9 %	59,8 %	7,2 %
143	ventilateur	56,6 % \pm 0,7 %	62,9 %	6,3 %
airplane	avion au sol	54,4 % \pm 0,8 %	61,9 %	7,5 %
block	bloc de béton	51,8 % \pm 1,1 %	55,2 %	3,4 %
coffee	cafetière	44,0 % \pm 0,9 %	50,3 %	6,3 %
cone	balise conique	38,0 % \pm 1,0 %	38,0 %	0,0 % (non significatif)
egg	oeuf	59,6 % \pm 1,1 %	64,3 %	4,7 %
fountain	fontaine	51,2 % \pm 1,2 %	56,3 %	5,1 %
parking	parcomètre	53,5 % \pm 0,9 %	57,4 %	3,9 %
pitcher	pichet d'eau	56,4 % \pm 1,0 %	61,3 %	4,9 %
police	voiture	49,2 % \pm 0,9 %	50,7 %	1,5 %
stapler	agrafeuse	41,9 % \pm 1,1 %	45,9 %	4,0 %

Sur le jeu de données de Berkeley, les images « rehaussées » ont souvent une mesure-F légèrement inférieure à celle de l'image originale, même pour un réseau entraîné sur une image avant d'être appliquée sur celle-ci.

L'image « 260058/pyramides » obtient une mesure-F de 22,7 % pour l'algorithme de Sobel. Cependant, après une tentative de restauration, la meilleure mesure atteignable n'est que de 19,0 %.

Pour l'image de « 12084/fond marin » de Berkeley, la différence de mesure-F entre les cartes classiques et les cartes fusionnées est toujours inférieure à 0,8 %, autant pour l'algorithme de Sobel que celui de Moon, avec une largeur de fenêtre variant de $N = 3$ à $N = 15$ pixels.

Dans le cas de l'image « 86016/aménagement paysager », tous les algorithmes retrouvent le cercle central, mais aucun ne retrouve les sillons. La mesure-F est passée de 18,5 % à 18,0 % pour Sobel avec seuillage global et de 17,7 % à 18,9 % avec seuillage adaptatif. Afin de caractériser les performances de la restauration sur une image très bruitée, l'expérience sur la taille de fenêtres a été répétée sur cette image (Figure 4.10).

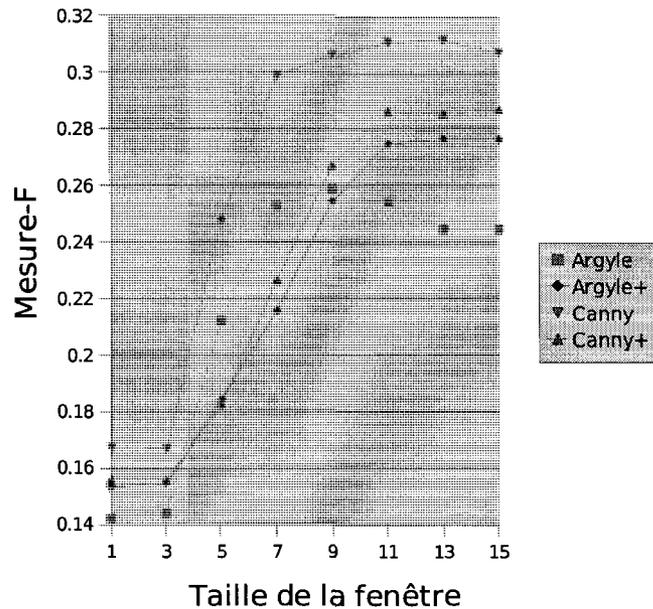


Figure 4.10 - Effet du lissage, image « aménagement paysager »

Sur les images IKONOS, vu le plus petit nombre d'images, nous avons testé tous les algorithmes qui ont pu être améliorés sur le jeu aéroporté. Pour ces images du campus de Sherbrooke et de l'Agence spatiale canadienne, les résultats obtenus en utilisant le réseau de neurones entraîné sur les images aéroportées sont fournis dans le Tableau 7. Les meilleurs résultats sont montrés dans les Figures 4.11 et 4.12.

Tableau 7 - Mesure-F sur les images IKONOS

<i>Algo.</i>	<i>Campus</i>	<i>Campus avec fusion</i>	<i>Diff.</i>	<i>Agence</i>	<i>Agence avec fusion</i>	<i>Diff.</i>
Sobel	45,7 % ±0,9 %	50,8 %	4,9 %	32,2 % ±0,9 %	40,9 %	8,7 %
Sobel Adapt.	45,7 % ±0,9 %	50,4 %	4,7 %	35,4 % ±0,9 %	42,2 %	6,8 %
Kirsch	43,7 % ±0,9 %	48,2 %	4,5 %	33,0 % ±0,9 %	40,1 %	7,1 %
Frei-Chen	44,7 % ±0,9 %	50,4 %	5,7 %	31,6 % ±0,9 %	40,0 %	8,4 %
Argyle 1x1	47,3 % ±0,9 %	48,6 %	1,3 %	35,1 % ±0,9 %	39,5 %	4,4 %

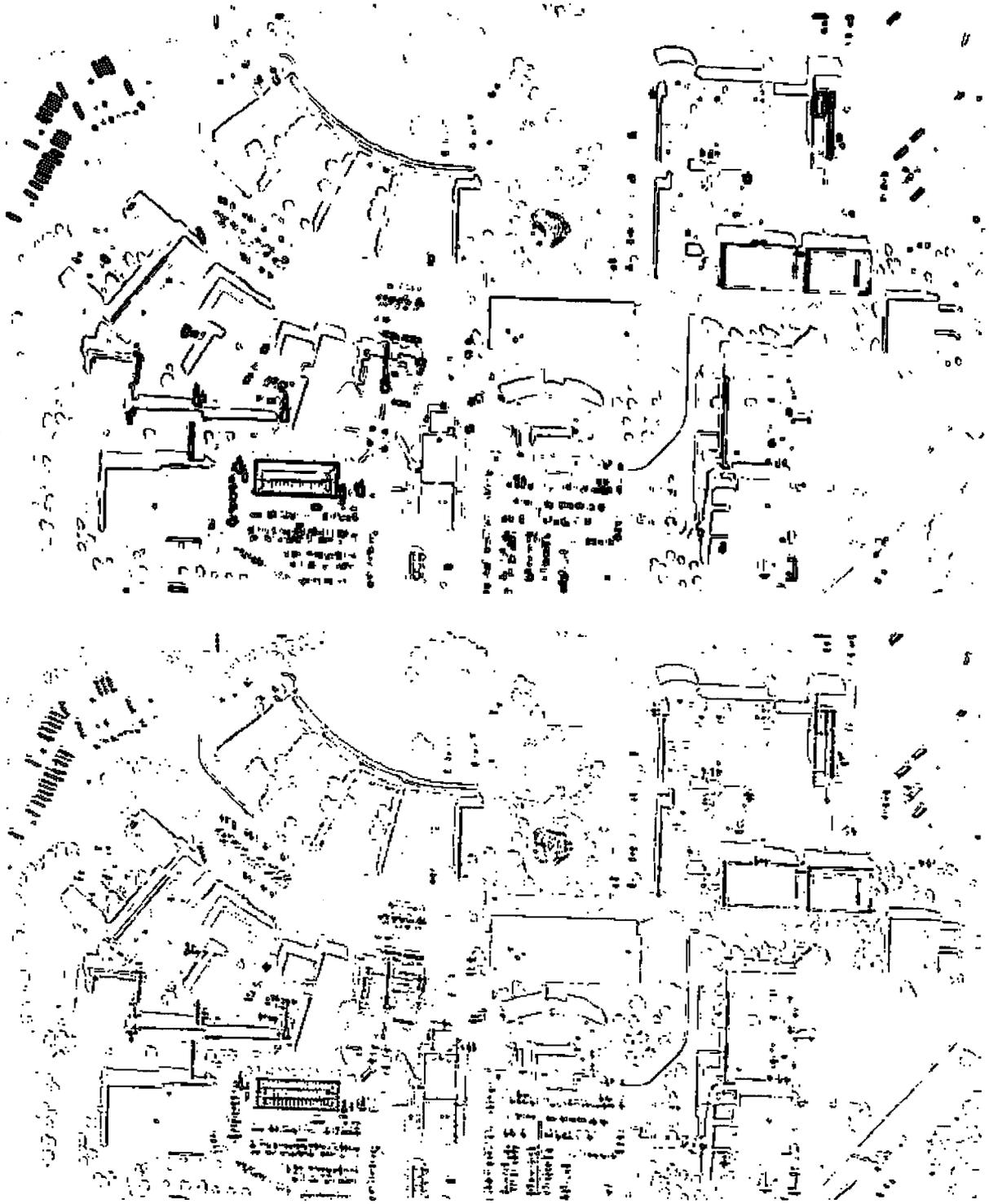


Figure 4.11 - Fusion sur l'image du campus (avant et après), détecteur de Sobel classique

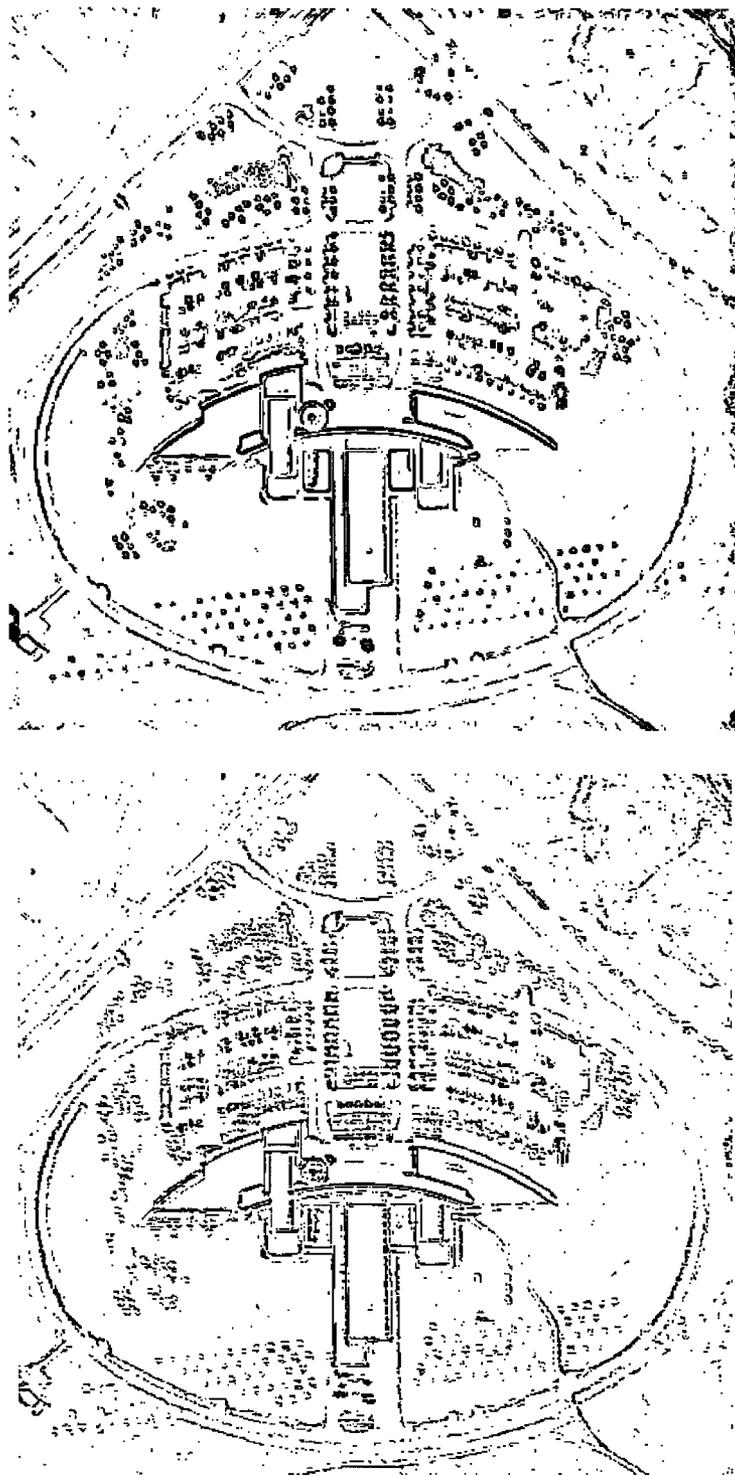


Figure 4.12 - Fusion sur l'image de l'Agence (avant et après), détecteur de Sobel adaptatif

4.5 Analyse de sensibilité

La qualité des contours obtenus par un détecteur « classique » est améliorée par la fusion. Cependant, quand les indicateurs sont utilisés comme des détecteurs de contours (Tableau 8), ils performant moins bien que le détecteur de Sobel ($45,7 \% \pm 0,9\%$).

Tableau 8- Indicateurs employés seuls pour détecter les contours

<i>Détecteur</i>	<i>Employé seul</i>	<i>Fusion</i>
Entropie	43,5 % $\pm 0,9$ %	43,2 % (NS)
Rapport signal sur bruit	38,9 % $\pm 0,9$ %	43,2 %
Filtre de Gabor	16,8 % $\pm 0,7$ %	38,6 %

Pour déterminer comment les indicateurs influencent la réponse du réseau de neurone, chaque combinaison d'indicateurs a été mise à zéro (Tableau 9). Les résultats obtenues avec combinaisons d'indicateurs qui contiennent le filtre de Gabor ne sont pas significativement différents de ceux obtenus avec le réseau complet.

Tableau 9- Combinaisons d'indicateurs employées pour la fusion

<i>Réseau</i>	<i>Mesure-F</i>
Réseau complet	50,8 % $\pm 0,9$ %
Gabor seul	50,9 % (NS)
Sans Entropie	50,8 % (NS)
Sans SNR	50,9 % (NS)
Sans Gabor	43,40 %
Entropie seule	45,7 %
SNR seul	37,4 %
Sans réseau	45,7 % $\pm 0,9$ %

Pour tester la robustesse du réseau, les signaux d'entrée ont été partiellement atténués. Sur la Figure 4.11, les deux courbes qui contiennent l'entropie sont légèrement plus élevées que les autres. Lorsque l'indicateur de Gabor est atténué, une détérioration est observée. À 50 % d'atténuation des indicateurs Gabor et Entropie, la diminution de la mesure-F n'est encore que de 0,67 %. Peu importe la combinaison d'indicateurs atténuée, il est possible d'atténuer le signal d'entrée de 75 % et d'améliorer la mesure-F en utilisant la fusion.

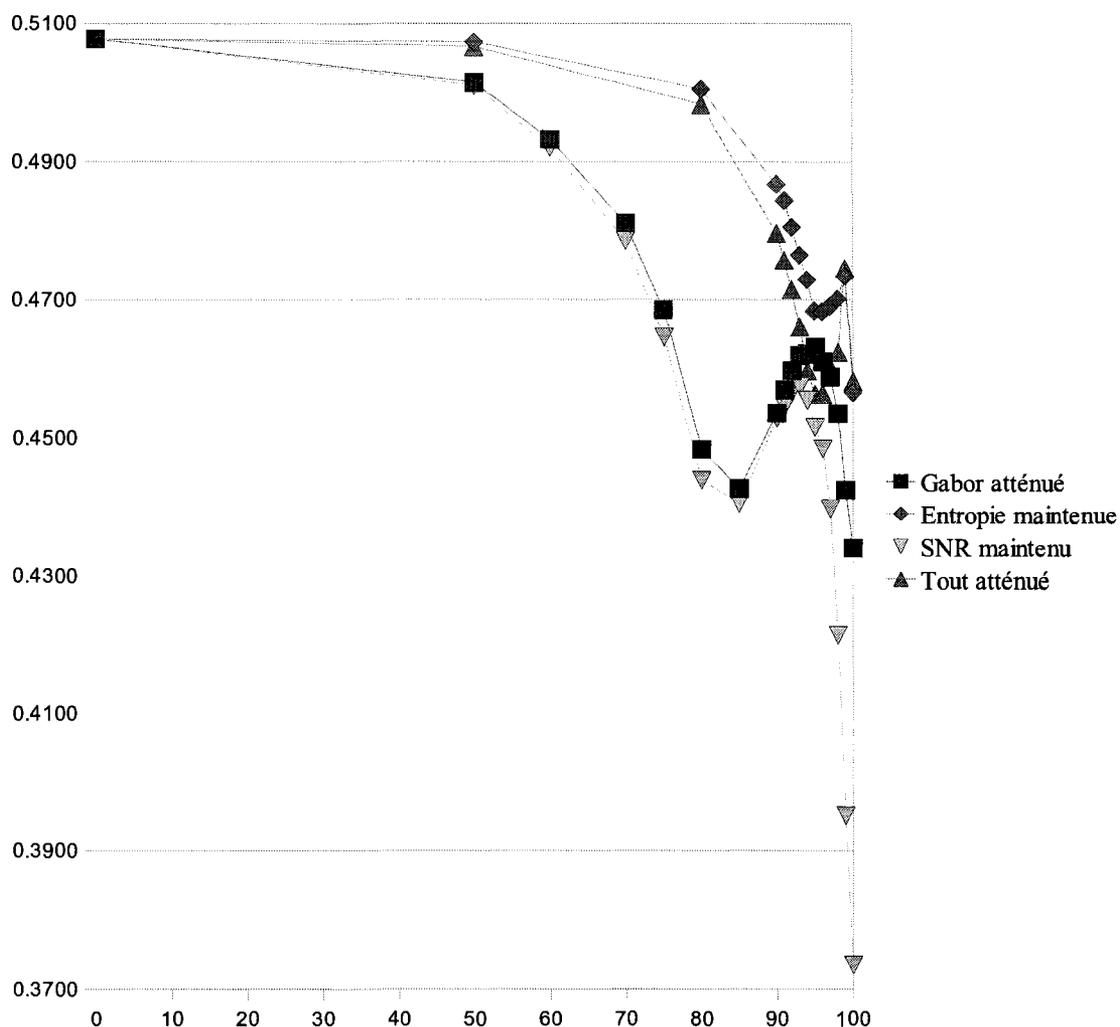


Figure 4.13 - Effet de l'atténuation du signal, image du campus

4.6 Conclusion partielle

L'application de notre protocole méthodologique a permis de développer et valider l'outil de restauration de contours. Après la mise au point du réseau de neurones, la restauration a été validée sur les données aéroportées. Les performances de la restauration en présence de bruit a ensuite été mesurée. Par la suite, les tests ont été répétés sur des images au sol et des imagerie IKONOS. Une analyse de sensibilité au niveau de signal en entrée a aussi été faite. Dans le prochain chapitre, une interprétation de ces résultats sera faite.

5. Interprétation des résultats

Les sections suivantes permettent de constater si les hypothèses de recherche sont confirmées ou infirmées. Elles seront l'occasion d'expliquer les observations faites lors de l'entraînement du réseau de neurones, d'interpréter les comportements attendus et observés lors de la restauration de contours, de commenter la façon dont la méthode étudiée réagit au bruit, de comparer les effets de la méthode sur divers types de données et, finalement, d'analyser l'effet de chaque indicateur sur l'indice de présence de contours.

5.1 Mise au point du réseau de neurones

Initialement, le réseau de neurones a été entraîné avec les indicateurs originaux de Chalmond, calculés sur un nombre égal d'exemples de « non-contours » et de « contours ». Tous les pixels de contours sont choisis, avec un échantillonnage aléatoire de pixels « non-contours ». Les images obtenues contenaient presque exclusivement des niveaux de gris élevés, indiquant une décision en faveur de la classe « contours ». Le contraste des images de sortie était faible. Même si la proportion entre les deux classes d'exemples était équilibrée, le réseau ne parvenait pas à les séparer. Apparemment, le nombre d'exemples utilisés était insuffisant.

Il semblerait que la classe « non-contours » contienne plus de variabilité que l'autre classe. Sa modélisation nécessite un plus grand nombre d'exemples. Pour la suite des essais, le nombre de pixels non-contours choisi a été doublé. Afin de conserver la parité entre l'influence des deux classes sur l'entraînement, le taux d'apprentissage associé aux pixels appartenant à la classe « contours » a été doublé.

Les cartes obtenues suite à cet entraînement contenaient des contours larges. Les contours larges observés peuvent être dus au fait que l'ensemble des indicateurs utilisés sont flous, car calculés sur une « grande » fenêtre. En réduisant la surface de support pour au moins un des indicateurs, les contours calculés rétrécissent.

Comparativement aux résultats obtenus sur l'ensemble des images, ceux obtenus en effectuant l'entraînement sur l'image « airfield » sont de qualité légèrement supérieure. Cette image contient beaucoup de contours anisotropes et saillants de type « ligne ». La grande majorité des contours qui y sont visibles sont représentés dans la vérité-terrain.

Ce n'est pas le cas de toutes les autres images, où les contours isotropes ou flous, difficiles à définir manuellement, sont souvent ignorés. Les images aéroportées du jeu de données de South Florida ont été acquises principalement dans le but de faire la cartographie automatique du bâti. Les contours de luminance contenus dans ces images ne forment pas nécessairement des frontières fermées. De plus, les frontières fermées ne correspondent pas toutes forcément à des objets pertinents à l'application finale. Ces différents niveaux d'évaluation créent des ambiguïtés dans la vérité-terrain associée à chaque image, notamment à cause de l'omission des contours de la cime des arbres. L'ajout d'images d'entraînement n'éliminerait pas une telle ambiguïté, qui est due à la nature même de l'application considérée. Il faudrait plutôt mieux définir le problème à résoudre au départ.

Bien que les cartes obtenues aient l'aspect de cartes de contours, cette interprétation est trompeuse. En effet, les éléments linéaires foncés qui y sont présents correspondent aux voisinages de contours et non aux contours eux-mêmes. Ainsi, chaque contour est représenté par trois lignes : une ligne foncée, une ligne pâle (un peu large) et une autre ligne foncée. Les cartes obtenus permettent de séparer les contours de leurs voisinages.

L'image contient aussi des régions pâles et des régions foncées (Figure 5.1). Les régions pâles correspondent à des zones homogènes de l'image, identifiées à tort comme des pixels de contours. Les régions foncées correspondent à des zones texturées de l'image. Dans ces zones texturées, l'image panchromatique originale varie très rapidement. Un détecteur basé sur l'amplitude du gradient y verrait, à tort, des contours. Ainsi, contrairement aux détecteurs basés sur la dérivée première, notre réseau de neurones réagit plus faiblement en présence de texture que sur un fond homogène.

Les cartes obtenues par réseau de neurones ont un aspect complémentaire aux images d'amplitude du gradient (voir Chapitre 4, Figures 4.4 et 4.5a). Elles inhibent le signal autour des contours, ainsi que dans plusieurs zones texturées. Le réseau entraîné selon notre méthodologie permet de générer des images d'une qualité suffisante pour satisfaire les exigences de la suite de la démarche.

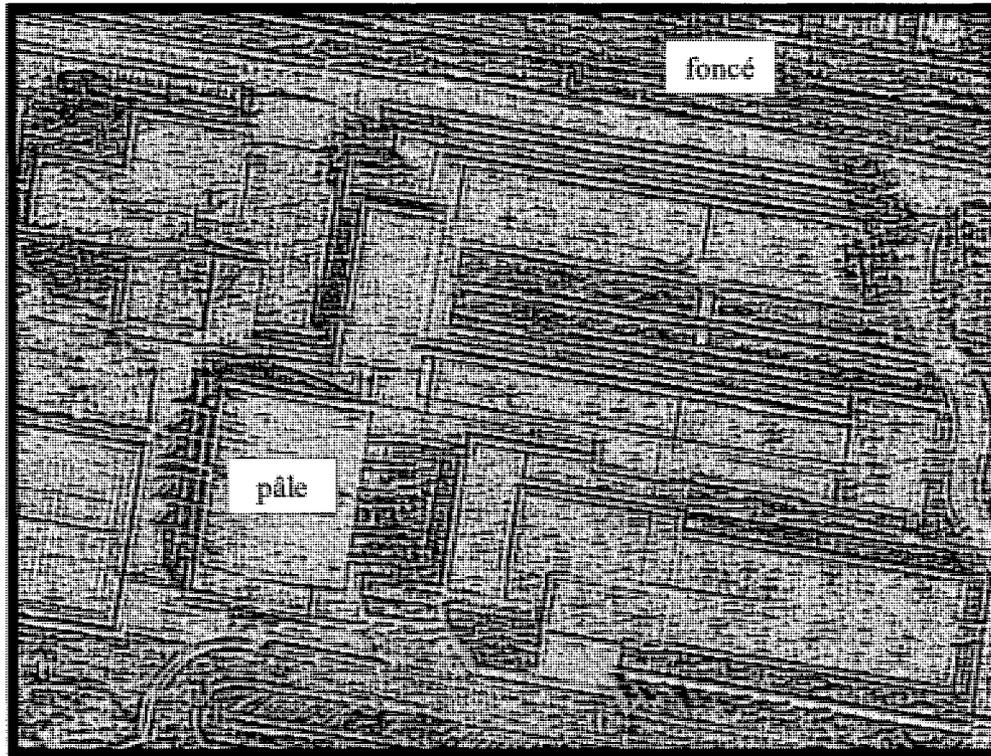


Figure 5.1 - Zone homogène (pâle) et zone texturée (foncé)

Le nombre d'exemples d'entraînement contenus dans une image est suffisant pour limiter l'erreur d'estimation, à condition de prendre en compte la variabilité intrinsèque à chaque classe en utilisant deux fois plus d'exemples « non-contours » que d'exemples « contours ». La localisation spatiale des éléments présents dans l'image finale semble limitée par les dimensions de la plus petite fenêtre employée lors du calcul des indicateurs.

La première hypothèse que cette recherche voulait vérifier est que la banque d'indicateurs de Chalmond, ou une banque d'indicateurs dérivés de ces derniers, permet de cartographier la présence ou l'absence de contours. Le résultat de l'expérience menée n'est pas utilisable à titre de détection de contours. Cependant, il contient une information qui complète celle contenue dans les cartes calculées avec un détecteur basé sur l'amplitude du gradient. Ainsi, nous considérons que l'hypothèse a été partiellement confirmée.

À cette étape de l'expérimentation, il est possible de fixer le choix de la structure du réseau et de la condition d'arrêt, ainsi que la méthode de sélection des données d'entraînement. Ces choix faits à l'avance peuvent simplifier l'utilisation de la méthode proposée par un non-expert. Dans la

section qui suit, nous vérifions si la complémentarité entre les deux types de cartes rend possible un prétraitement pouvant corriger une carte de contours flous avant le seuillage.

5.2 Restauration de cartes de contours

Les cartes de gradient et les images de contours sont représentées par des entiers en 8 bits. En principe, le résultat de la multiplication pixel à pixel utilisée pour faire la fusion devrait être codé sur 16 bits, mais il est renormalisé en 8 bits. Les pixels de l'image fusionnée auront une valeur inférieure ou égale à celle des pixels correspondants dans la carte de gradient. Cette valeur sera aussi inférieure ou égale à celle des pixels correspondants dans les images originales. Dans les faits, le niveau de gris moyen de l'image diminue. En général, les niveaux de gris des pixels de contours devraient diminuer peu et ceux des pixels d'arrière-plan devraient diminuer beaucoup. Ainsi, en principe, dans l'image fusionnée, le contraste entre les contours et leur voisinage est rehaussé.

Décrivons les conséquences espérées de la fusion sur les quatre types de pixels, soit VP, FP, FN et VN. Nous aborderons le cas d'un voisinage homogène et celui d'un voisinage texturé. Pour faciliter la compréhension de cette analyse, il peut être utile de se référer aux Figures 4.4 et 4.5a, dans le Chapitre 4, qui montrent une carte issue du réseau de neurones, une carte de gradient et une carte fusionnée.

5.2.1 Effets attendus

Dans une zone homogène, les VP ont une valeur élevée dans la carte de gradient, tout comme dans l'image issue du réseau de neurones. Dans la carte finale, ils devraient avoir une valeur élevée et demeurer des VP. Ils peuvent cependant devenir des FN si leur valeur est proche du seuil.

Dans une zone homogène, les FP sont causés par la présence de contours trop larges dans la carte de gradient. Les valeurs de chaque côté des contours étant faibles dans la carte issue du réseau de neurones, la valeur de ces pixels dans la carte finale sera plus faible que celle dans la carte de gradient. Plusieurs de ces pixels deviendront des VN dans la carte finale, ce qui amincira les contours. Un exemple de pixels VP et FP en zone homogène est le contour entre un bâtiment et son ombre. Ces contours sont amincis par la fusion.

Dans une zone homogène, sur la carte de gradient, les valeurs des pixels de FN sont légèrement sous le seuil. Cela est dû à un choix de seuil élevé. Un tel choix de seuil peut avoir été fait pour empêcher la formation excessive de FP les zones les plus texturées de l'image. Les pixels de FN correspondent à de « vrais contours ». Dans la carte issue du réseau de neurones, ils devraient avoir une valeur élevée. Ainsi, dans la carte finale, la valeur assignée aux pixels FN devrait être relativement élevée, ce qui peut permettre de restaurer, au moins partiellement, les contours. C'est ce qui se passe, par exemple, avec la limite du stationnement dans le coin supérieur gauche de l'image « école » (Figures 4.5 et 4.6, Chapitre 4).

Dans une zone homogène, les pixels VN ont une faible valeur dans la carte de gradient. S'ils sont situés près d'un vrai contour, ils ont aussi une faible valeur dans la carte issue du réseau de neurone. La valeur résultante dans la carte finale est très faible. Ces surfaces, par exemple des zones dégagées sur les toits de bâtiments, devraient conserver leur état de VN dans la carte finale. Si les pixels VN sont situés loin des vrais contours, ils ont une valeur élevée dans la carte issue du réseau de neurones. Dans la carte finale, leur valeur est inférieure, mais proche, de celle qu'elle avait dans la carte de gradient. Ils devraient conserver leur état de VN dans l'image finale, mais causent parfois des FP. Un exemple de ce phénomène est l'apparition sur l'image restaurée d'arbres au sol, absents des vérité-terrain et des cartes de contours basées sur le gradient.

Dans une carte de contours idéale, les contours sont minces et l'arrière-plan ne contient pas de bruit. En pratique, dans les zones homogènes, une carte de gradient contient des contours trop larges et peu de fausses alarmes dans l'arrière-plan. Suite à la fusion, un effet d'amincissement est facilement observé sur les contours. Cependant, quelques fausses alarmes sont ajoutées dans l'arrière-plan.

Dans une carte de contours idéale, les contours sont minces et l'arrière-plan est peu ou pas contaminé par du bruit. Cependant, en pratique, dans les zones texturées, les niveaux de gris des pixels d'une carte de gradient varient très rapidement spatialement. Plusieurs des maximums locaux qui y sont présents ne correspondent pas à des contours. Leur élimination par filtrage du gradient est associée à une perte de détails et de précision de localisation des contours. Le problème est d'autant plus grave si les éléments de texture sont d'une dimension proche de celle

des éléments utiles à l'interprétation. Sur de tels voisinages, une méthode de seuillage du gradient n'est pas satisfaisante pour identifier les contours.

Ayant décrit d'une façon générale les effets attendus de la restauration, il est souhaitable de voir, en pratique, à quel degré ces effets sont présents sur les images traitées. Dans la suite de cette section, nous décrivons les particularités des résultats obtenus lors des tentatives d'amélioration des détecteurs de Sobel, Kirsch, Moon, Canny sur des images aériennes.

5.2.2 Sobel global

Quand les cartes obtenues par réseau de neurones ont été multipliées par celles obtenues par l'amplitude du gradient de Sobel, les résultats obtenus ont une mesure-F systématiquement plus élevée que celle obtenue avec la méthode classique (Tableau 1). Les cartes corrigées contiennent des contours mieux localisés, plus fins que les contours de Sobel. Elles contiennent beaucoup moins de fausses alarmes dues au bruit et un peu plus de fausses alarmes dues à la détection de détails visuellement significatifs, mais absents de la vérité-terrain. Plusieurs contours sont allongés. Quelques contours déjà fragmentés dans la carte de Sobel sont parfois fragmentés un peu plus par l'effet d'amincissement, ce qui diminue la mesure-F. Les éléments conservés dans l'image sont plus souvent des contours que des éléments parasites.

L'image « airfield » ayant été utilisée pour faire l'entraînement du réseau, son taux d'amélioration de 7,7 % est biaisé. Les autres images montrent quand même des améliorations variant de 3,8 % à 7,5 %. Toutes les images sont améliorées. La moyenne des pourcentages d'amélioration sur les neuf images est de 5,7 %, l'écart-type de cet échantillon est de 1,4 %.

Le test des signes permet de vérifier si l'amélioration observée est vraiment significative. Soit l'hypothèse nulle est qu'il n'y a pas de vraie différence dans la performance moyenne entre un algorithme de base et un nouvel algorithme. Sous cette hypothèse nulle, chaque algorithme a une probabilité de 0,5 d'avoir une plus grande mesure-F pour un essai donné. Le nombre d'essais pour lesquels un algorithme génère une mesure-F plus grande que l'autre devrait suivre une distribution binomiale (Min *et al.*, 2004). En général, sous l'hypothèse nulle, le nouvel algorithme devrait avoir une plus grande mesure-F pour quatre ou cinq des neuf essais. Tout résultat hors de l'intervalle de trois à six supporte une différence significative au niveau $\alpha = 0,05$ entre les performances de l'algorithme de base et du nouvel algorithme. Selon ce raisonnement, quand

neuf essais sur neuf montrent une amélioration, comme c'est le cas pour l'algorithme de Sobel fusionné comparé à l'algorithme de Sobel classique, l'amélioration est significative.

5.2.3 Sobel adaptatif

La fusion proposée plus haut permet d'améliorer la qualité finale d'une carte de contours de Sobel. Un autre moyen simple d'améliorer une telle carte est de remplacer le seuillage global par un seuillage adaptatif. La carte obtenue par seuillage adaptatif contient des contours plus fins que celle obtenue par seuillage global. Elle contient aussi plus de détails, par exemple les objets sur les toits.

Nous comparerons maintenant les performances du détecteur adaptatif avec celles du détecteur global et du détecteur adaptatif avec fusion. L'efficacité du détecteur adaptatif avec fusion sera ensuite confrontée à celle du détecteur global avec fusion. Dans chacun de ces trois cas, la distribution de l'ampleur des effets observés sera résumée et la détermination de la significativité de ces effets sera faite.

Premièrement, étudions la différence entre l'algorithme de Sobel avec seuillage global et celui avec seuillage adaptatif. La moyenne des pourcentages d'amélioration sur les dix images est de 1,5 %, l'écart-type de cet échantillon est de 1,2 %. Le pourcentage d'amélioration varie de 0,3 % pour l'image « building », jusqu'à 3,5 % pour l'image « largebuilding ». Cependant, la mesure-F de l'image « woods » est détériorée de 2,0 %. Elle contient plusieurs détails, des arbres, qui ne sont pas identifiés comme contours dans la vérité-terrain, mais qui le sont dans la carte adaptative finale.

L'amélioration entre l'algorithme de Sobel avec seuillage global et avec seuillage adaptatif est-elle significative? Sous l'hypothèse nulle, le nouvel algorithme devrait avoir une plus grande mesure-F pour cinq des dix essais. Tout résultat hors de l'intervalle de trois à sept supporte une différence significative au niveau $\alpha = 0,05$ entre les performances de l'algorithme de base et du nouvel algorithme. Étant donné que neuf essais sur dix montrent une amélioration, la différence entre l'algorithme de Sobel avec seuillage et celui avec seuillage adaptatif est significative.

Deuxièmement, étudions la différence, pour un algorithme de Sobel avec seuillage adaptatif, entre ne pas utiliser la fusion et utiliser la fusion. L'image « airfield » montre une amélioration de

7,8 %, mais ce résultat est biaisé car l'image a été utilisée pour entraîner le réseau de neurones. La moyenne des pourcentages d'amélioration sur les neuf autres images est de 5,6 %, l'écart-type de cet échantillon est de 1,2 %. Le pourcentage d'amélioration varie de 4,0 % pour l'image « largebuilding », jusqu'à 7,9 % pour l'image « series ».

Pour un algorithme de Sobel avec seuillage adaptatif, l'amélioration entre la non-utilisation de la fusion et l'utilisation de la fusion est-elle significative? Sous l'hypothèse nulle, le nouvel algorithme devrait avoir une plus grande mesure-F pour quatre ou cinq des neuf essais. Tout résultat hors de l'intervalle de trois à six supporte une différence significative au niveau $\alpha = 0,05$ entre les performances de l'algorithme de base et du nouvel algorithme. Étant donné que neuf essais sur neuf montrent une amélioration, la différence entre l'algorithme adaptatif sans fusion et l'algorithme adaptatif avec fusion est significative.

Troisièmement, étudions la différence, pour un algorithme de Sobel avec fusion, entre le seuillage global et le seuillage adaptatif et fusion. L'image « airfield » montre une amélioration de 3,1 %, mais ce résultat est biaisé car l'image a été utilisée pour entraîner le réseau de neurones. La moyenne des pourcentages d'amélioration sur les neuf autres images est de 1,2 %; l'écart-type de cet échantillon est de 0,8 %. Le pourcentage d'amélioration varie de 0,6 % pour l'image « building », jusqu'à 2,2 % pour l'image « series ». Cependant, la mesure-F de l'image « woods » est détériorée de 0,3 %.

Pour un algorithme de Sobel avec fusion, l'amélioration entre seuillage global et seuillage adaptatif est-elle significative? Sous l'hypothèse nulle, le nouvel algorithme devrait avoir une plus grande mesure-F pour quatre ou cinq des neuf essais. Tout résultat hors de l'intervalle de trois à six supporte une différence significative au niveau $\alpha = 0,05$ entre les performances de l'algorithme de base et du nouvel algorithme. Étant donné que huit essais sur neuf montrent une amélioration, la différence est significative.

En résumé, tous les cas considérés comportent une amélioration significative. Les effets de la fusion et du changement de seuillage sont indépendants. L'amélioration de mesure-F qu'ils apportent peut être cumulée en appliquant les deux opérations (Figure 5.2).

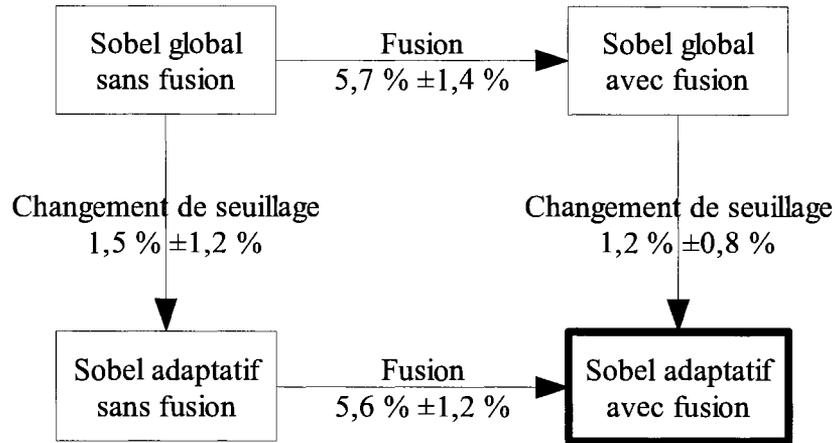


Figure 5.2 - Restauration de l'algorithme de Sobel

5.2.4 Kirsch

Visuellement, l'algorithme de Kirsch donne des résultats légèrement moins bruités que l'algorithme de Sobel. Quantitativement, la moyenne des pourcentages d'amélioration sur les dix images est de 0,1 %, l'écart-type de cet échantillon est de 0,3 %. Seules les images « school » et « woods » sont détériorées. Ainsi, la différence est significative au niveau $\alpha = 0,05$; sur les images aériennes, l'algorithme de Kirsch est mieux que l'algorithme de Sobel.

L'ajout de la fusion au détecteur de type boussole de Kirsch permet d'observer les mêmes effets que dans le cas du détecteur de Sobel avec seuillage global. L'image « airfield » ayant été utilisée pour faire l'entraînement du réseau, son taux d'amélioration de 6,6 % est biaisé. Les autres images montrent quand même des améliorations variant de 3,2 % pour l'image « building » à 7,3 % pour l'image « series ». La moyenne des pourcentages d'amélioration sur les neuf images est de 5,4 %, l'écart-type de cet échantillon est de 1,4 %. Toutes les images sont améliorées. La différence observée est donc significative au niveau $\alpha = 0,05$.

La fusion étant utilisée, est-il plus avantageux de se baser sur une carte de gradient de Sobel ou de Kirsch? L'image « airfield » montre une amélioration de 0,9 %, mais ce résultat est biaisé car l'image a été utilisée pour entraîner le réseau de neurones. La moyenne des pourcentages d'amélioration sur les neuf autres images est de 0,3 %, l'écart-type de cet échantillon est de 0,5 %. Seules les images « homes » et « series » sont détériorées. Ainsi, la différence est significative au niveau $\alpha = 0,05$; si la fusion est utilisée, il est préférable d'utiliser une carte de gradient Sobel

plutôt qu'une carte de Kirsch. La Figure 5.3 résume les comparaisons entre l'algorithme de Sobel et celui de Kirsch.

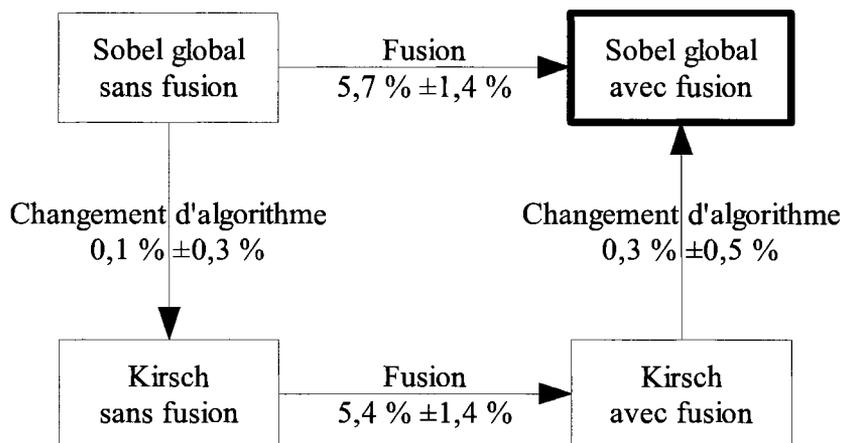


Figure 5.3 - Restauration de l'algorithme de Kirsch

5.2.5 Frei-Chen

Les contours déterminés par l'algorithme de Frei-Chen sont larges et bruités. Sur huit des dix images, la mesure-F est plus élevée pour l'algorithme de Sobel que pour l'algorithme de Frei-Chen. Sur les images « building » et « series », la mesure-F est la même. Ainsi, la différence est significative au niveau $\alpha = 0,05$. La moyenne de l'amélioration sur les dix images est de 0,2 %, avec un écart-type d'échantillon de 0,2 %.

Lors de l'ajout de la fusion au détecteur de Frei-Chen, les mêmes effets sont observés que dans le cas du détecteur de Sobel avec seuillage global. L'image « airfield » ayant été utilisée pour faire l'entraînement du réseau, son taux d'amélioration de 7,6 % est biaisé. Les autres images montrent quand même des améliorations variant de 4,2 % pour l'image « building » à 7,6 % pour l'image « series ». La moyenne des pourcentages d'amélioration sur les neuf images est de 5,9 %, l'écart-type de cet échantillon est de 1,3 %. Toutes les images sont améliorées. La différence observée est donc significative au niveau $\alpha = 0,05$.

Quand la fusion est utilisée, vaut-il mieux qu'elle se base sur le gradient de Sobel ou sur celui de Frei-Chen? Sur l'image « airfield », qui est biaisée, l'algorithme de Sobel a une mesure-F 0,1 % plus élevée que celui de Frei-Chen. Sur les autres images, la moyenne des changements, en passant de Frei-Chen à Sobel, est de 0,1 %, avec un écart-type sur l'échantillon de 0,3 %. Sur

quatre images, la mesure-F de l'algorithme de Sobel est supérieure à celle de l'algorithme de Frei-Chen. Sur quatre autres images, elle est inférieure. Sur l'image « woods », les deux algorithmes obtiennent la même mesure-F. Ainsi, il n'y a pas de différence significative entre les deux algorithmes. La Figure 5.4 résume les comparaisons entre l'algorithme de Sobel et celui de Frei-Chen.

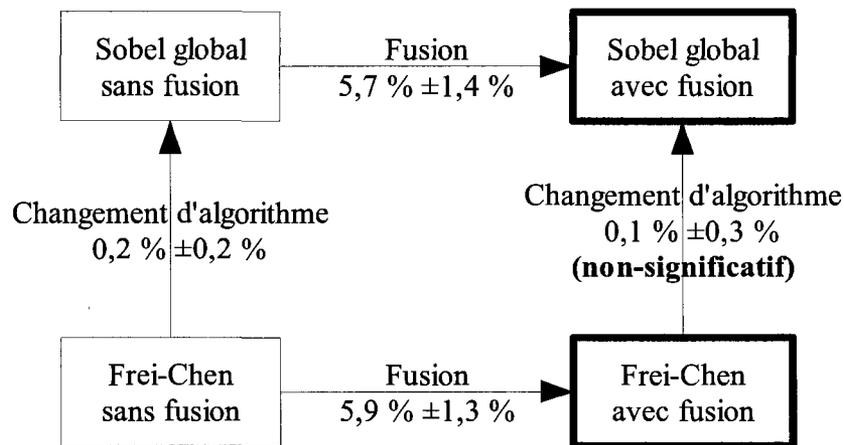


Figure 5.4 - Restauration de l'algorithme de Frei-Chen

5.2.6 Moon

Notre méthode appliquée à des cartes non-bruitées de gradient de l'algorithme de Moon a peu ou pas d'effet, selon l'échelle étudiée. L'application de la méthode de fusion à l'algorithme de Moon donne des résultats équivalents à ce dernier. Les résultats obtenus sont marginalement inférieurs ou supérieurs à ceux de l'algorithme de Moon original. Pour toutes les images, les meilleurs résultats sont obtenus avec un filtre de dimension 3x3, plutôt que 5x5 ou 7x7. Cela est dû au faible niveau de bruit des images étudiées.

Sur les algorithmes de Sobel, Kirsch et Frei-Chen, notre méthode corrige le positionnement des contours, tout en les amincissant. Or, les contours détectés par l'algorithme de Moon sont déjà bien positionés et minces. Le filtrage exponentiel utilisé par l'algorithme donne une forte pondération au pixel central, ajoutant ainsi moins de flou à l'image qu'un filtrage gaussien. Cela favorise la localisation précise des contours dans l'image. L'effet de réduction du bruit et de rétrécissement des contours par l'algorithme de Moon est semblable à ce qui se passe avec notre méthode. Cette dernière ne permet pas d'en améliorer les résultats.

Il faut souligner un fait hautement significatif : sur toutes les images, autant pour les algorithmes de Sobel global, Sobel adaptatif, Kirsch et Frei-Chen, toutes les cartes après fusion ont une mesure-F supérieure à la meilleure carte obtenue sur cette image avec le détecteur de Moon.

5.2.7 Canny

Aucune amélioration n'a pu être apportée à la performance du détecteur de Canny, sauf pour le cas biaisé de l'algorithme de Sobel adaptatif avec fusion sur l'image « airfield ». La légère dégradation observée lors de la modification de l'algorithme de Canny peut être due aux différences de positionnement entre la méthode de Canny et la nôtre. L'examen visuel des cartes montre que les maximums sont légèrement décalés lorsqu'on passe de la carte de Canny à la carte de Canny modifiée. Les contours modifiés sont légèrement aléatoires (Figure 5.5).

Les contours sont aussi plus larges que ceux de Canny. Cet effet pourrait être causé par la quantification appliquée par notre approche, à la fin de la fusion. Cette quantification réduit le nombre de valeurs possibles dans l'image fusionnée comparativement aux images sources. Ainsi, il y a une perte de contraste. Cela rend peut être plus difficile de faire la différence entre un élément de frontière et un élément près d'une frontière.

Dans certains cas, les contours sont fragmentés. Cette fragmentation se produit lors de l'amincissement par suppression des non-maximums. L'absence d'ajustement de la phase du gradient et la proportion fixe entre les deux seuils d'hystérésis pourraient être responsables de cet effet. Des solutions possibles sont l'élimination de petits contours parasites et l'augmentation du rapport entre les deux seuils.

Sur l'image « baseball », peu importe la taille de la fenêtre, les résultats obtenus avec l'algorithme de Canny sont toujours supérieurs à ceux obtenus par l'algorithme de Argyle. Les résultats de Canny semblent peu sensibles au niveau de lissage employé. Dans le cas du détecteur de Argyle l'application d'un filtrage à l'image détériore le résultat. La mesure-F est plus élevée sans lissage; elle prend alors une valeur de $50,4 \% \pm 0,8 \%$. Cependant, si la restauration est employée, le meilleur résultat est obtenu avec un lissage effectué avec une fenêtre d'une largeur de cinq pixels. La mesure-F vaut alors $50,5 \% \pm 0,8 \%$.

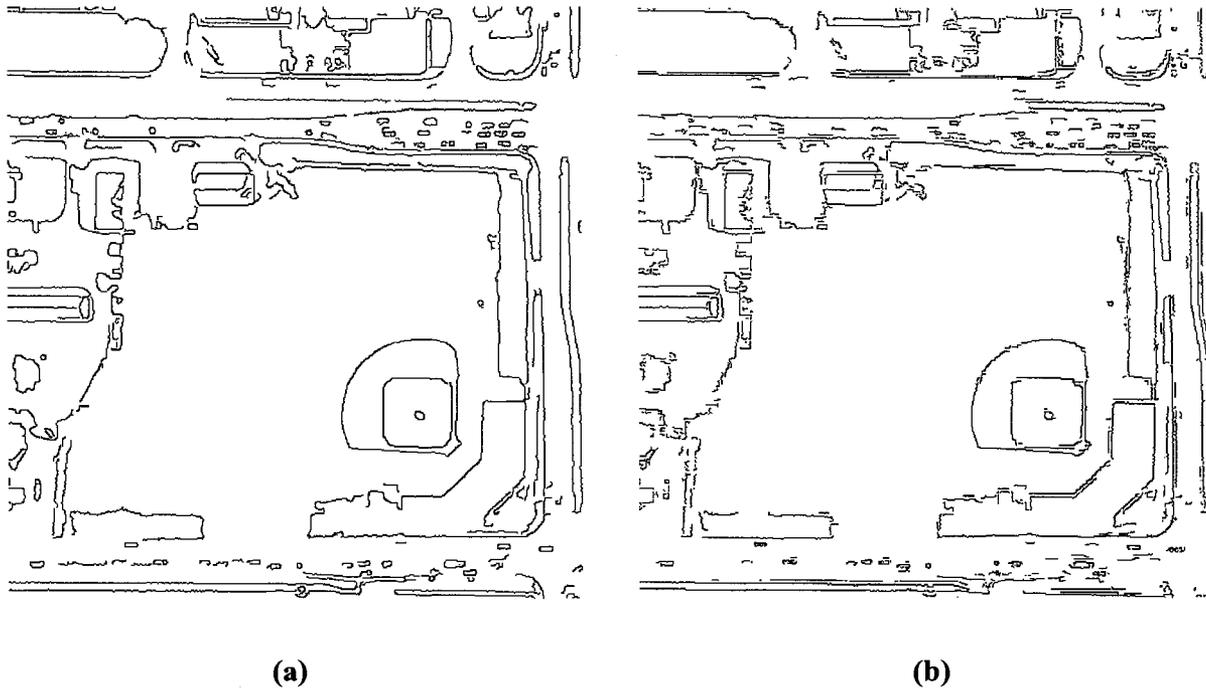


Figure 5.5 - Détérioration de la mesure-F après la fusion (algorithme de Canny)

(a) Canny avant fusion (N=11, F = 62,6%)

(b) Canny après fusion (N=11, F = 55,6%)

D'une façon générale, l'algorithme de Canny, l'algorithme de Moon et notre méthode ont des effets semblables. Grâce aux post-traitements qu'il utilise, le détecteur de Canny performe mieux que notre méthode.

5.2.8 Synthèse

Les résultats obtenus permettent de dégager quelques généralités. L'effet de restauration s'obtient avec un gradient de Sobel, Kirsch ou Frei-Chen. Les résultats obtenus avec Sobel sont significativement supérieurs à ceux obtenus avec Kirsch. Quand les résultats de Sobel sont comparés à ceux de Frei-Chen, il n'y a pas de différence significative. Considérant ces résultats et la simplicité de l'algorithme de Sobel, notre recommandation est d'utiliser ce dernier, plutôt que les deux autres, pour traiter des images aéroportées.

Dans la section ci-dessus, nous avons comparé les performances de notre outil avec celles de détecteurs tirés de la littérature. Ainsi, nous avons atteint le troisième objectif de la présente

thèse, qui est de comparer l'efficacité de l'outil développé à celle d'autres méthodes de détection de contours.

La deuxième hypothèse étudiée dans cette thèse était qu'une carte de présence de contours permet d'effectuer un traitement qui peut corriger les résultats fournis par un détecteur de contours conventionnel. Cette hypothèse a été vérifiée. Le choix de l'opérateur de fusion, ainsi que celui du type et de l'échelle des indicateurs donne des résultats suffisamment satisfaisants pour décider d'utiliser la méthode sans modification de ces paramètres.

Il est important de se rappeler que notre méthode, lorsqu'elle est utilisée pour rehausser l'algorithme de Sobel, permet d'obtenir une mesure-F supérieure à celle obtenue avec l'algorithme de Moon, avec ou sans fusion. Ceci est surprenant lorsqu'on considère que le détecteur de Moon est mathématiquement optimal pour la détection de contours de type « rampe ». Un tel résultat pourrait s'expliquer par le fait que les contours naturels ne correspondent pas exactement au modèle « rampe » et que le type de contours détecté par la méthode de fusion est plus près des contours naturels.

5.3 Robustesse au bruit

Des courbes de sensibilité au bruit ont été tracées pour l'algorithme de Sobel sans fusion et avec fusion. La mesure-F et la figure de mérite de Pratt ($\beta = 10$) ont toutes deux été employées. Les courbes ont toutes le même aspect général : quand le rapport signal sur bruit est faible ($SNR = 1$), leur valeur est proche de 0,0. Quand la proportion de signal augmente, la mesure de qualité augmente jusqu'à une valeur plateau. Pour l'algorithme de Sobel avec fusion, la courbe tracée selon la mesure-F est presque identique à celle tracée selon la figure de mérite de Pratt. Dans ces deux cas, le plateau a une valeur proche de 1,0.

Pour l'algorithme de Sobel sans fusion, la courbe de mesure-F forme son plateau à 0,80 et la courbe de figure de mérite, à 0,60. Bien que l'évaluation numérique diffère selon le critère employé, il faut souligner que l'optimisation selon les deux critères mène à un choix quasi-identique de paramètres de seuillage de l'amplitude du gradient. Cela est facilement expliqué en se référant à la définition des deux critères : chacun atteint sa valeur maximale lorsque le nombre de pixels FP est proche du nombre de pixels FN. Ainsi, en principe, le choix d'utiliser la mesure-F devrait mener aux mêmes conclusions que celui d'utiliser la figure de mérite de Pratt.

La comparaison des courbes de sensibilité suggère que l'emploi de la fusion est justifié quand le rapport signal sur bruit est supérieur ou égal à $SNR = 20$. Si le rapport signal à bruit avant la différentiation est inférieur à 20, les performances sont dégradées. Celles de notre détecteur sont plus affectées que celles du détecteur de Sobel.

Sur l'image « baseball » bruitée, les algorithmes de Sobel et de Moon avec un niveau moyen de lissage (7x7, 9x9, 11x11) produisent des contours bruités. Bien que la fusion parvienne à rallonger et amincir les contours, le bruit résiduel dans l'amplitude du gradient estimé est amplifié par la fusion, ce qui détériore la mesure-F. Pour l'algorithme de Moon avec un fort lissage (fenêtre de taille 13x13 et 15x15), l'estimation de l'amplitude du gradient est bien débruitée. Les contours ont pu être restaurés et le résultat final à ces échelles est meilleur que celui de l'algorithme de Moon employé seul. Les différences de mesure-F sont significatives à $\alpha = 0,05$.

Pour l'algorithme de Canny, il n'a pas été possible de restaurer de contours sur l'image bruitée. En particulier, avec la fenêtre 15x15, les contours élargis par le lissage gaussien sont plus larges que l'inhibition de signal de la méthode, ce qui crée de faux contours.

Quand le réseau est entraîné sur une image bruitée, l'efficacité de la restauration est réduite. Un effet d'amincissement des contours est quand même observé pour les algorithmes de Moon 13x13 employé sur une image bruitée et de Sobel employé sur une image non-bruitée. Cependant, d'après les intervalles de confiance de Samsa sur la mesure-F, les différences observées ne sont pas significatives à $\alpha = 0,05$.

Les résultats obtenus avec l'algorithme de Canny sont toujours supérieurs à ceux obtenus par l'algorithme de Argyle, peu importe le niveau de bruit affectant l'image. Sur l'image « baseball » sans bruit, les résultats de Canny semblent peu sensibles au niveau de lissage employé. Dans le cas de Argyle, l'application d'un filtrage à l'image détériore le résultat.

Sur l'image originale, le détecteur de Argyle sans restauration performe mieux sans lissage, avec dans ce cas une mesure-F de $50,4 \% \pm 0,8 \%$. Si la restauration est employée, le meilleur résultat est obtenu avec une largeur de fenêtre de cinq pixels, avec une mesure-F de $50,5 \% \pm 0,8 \%$.

Pour l'image bruitée, l'application d'un filtre est nécessaire pour tous les algorithmes. L'algorithme de Argyle original performe mieux sur une fenêtre d'une largeur de cinq pixel, à

47,0 % \pm 0,8 % de mesure-F. Avec restauration, la meilleure mesure-F est de 47,5 % \pm 0,8 % et est obtenue avec une fenêtre de lissage de sept pixels de largeur. Il n'y a pas de différence de mesure-F significative entre le meilleur résultat obtenu avec le détecteur de Argyle employé seul et le détecteur de Argyle « restauré ». Il semble que la technique de restauration requiert un peu plus de lissage pour bien performer.

Ainsi, pour rehausser une image bruitée, il est possible d'utiliser le même réseau de neurones que sur une image non-bruitée. Il faut cependant utiliser une carte de gradient lissée. L'algorithme de Moon, avec une taille de fenêtre adaptée au niveau de bruit, convient à la création d'une telle carte. Notre méthode parvient à améliorer les performances des contours de Moon, mais pas ceux d'un contour de Argyle. Cela est peut-être dû au fait que le filtrage exponentiel affecte moins le positionnement des contours que le filtrage gaussien.

La troisième hypothèse étudiée dans cette thèse était que la restauration de contours peut être effectuée sur des images bruitées. Notre méthode peut effectivement restaurer les contours sur une image bruitée; l'hypothèse est donc vérifiée. Cependant, il faut que l'image soit suffisamment débruitée par l'algorithme de filtrage précédant la restauration. Un réseau de neurones entraîné sur une image bruitée parvient quand même à restaurer les contours, mais le résultat est de moins bonne qualité que celui obtenu avec un réseau entraîné sur une image non-bruitée. Une stratégie de lissage avantageuse est la suivante :

- estimer le rapport signal à bruit de l'image (Meer *et al.*, 1990).
- Si le rapport est supérieur ou égal à 20, utiliser le détecteur de Sobel avec fusion.
- Sinon, utiliser un filtrage exponentiel pour augmenter le rapport signal à bruit à 20 ou plus avant d'effectuer la fusion.

5.4 Comparaison avec d'autres jeux de données

Sur les photos terrestres, selon les intervalles de Samsa, toutes les différences observées sont significatives à $\alpha = 0,05$, sauf pour l'image « cone » avec seuillage adaptatif, dont la mesure-F n'a pas été modifiée par la restauration. Dans l'image « 202/alligator », il y a une dégradation significative. Pourtant, dans les images « Cone » et « 202/alligator », l'effet d'amélioration de la localisation et de restauration de contours est présent, comme dans les autres images. Cependant,

la mesure-F est légèrement dégradée. Cela peut s'expliquer par le fait que la végétation, qui compose la majorité de ces images, est presque entièrement ignorée dans la vérité-terrain.

Lors d'utilisation du seuillage global, les images terrestres montrent des améliorations variant de 2,3 % pour l'image « 61/tomate » à 9,8 % pour l'image « 101/fer à repasser ». La moyenne des pourcentages d'amélioration sur les images testées est de 5,4 %, l'écart-type de cet échantillon est de 3,2 %. Sur les 22 images, deux seulement ne sont pas améliorées. La différence observée est donc significative au niveau $\alpha = 0,05$.

Avec le seuillage global, les images terrestres montrent des améliorations variant de 1,5 % pour l'image « police » à 1,5 % pour l'image « airplane ». Sur les 12 images, seule l'image « cone » n'est pas améliorée. La différence observée est donc significative au niveau $\alpha = 0,05$. La moyenne des pourcentages d'amélioration sur les images testées est de 4,6 %; l'écart-type de cet échantillon est de 2,2 %.

Les tests effectués sur le jeu de données de Berkeley se sont montrés non concluants, avec ou sans seuillage adaptatif. Beaucoup de détails sont omis dans les vérités-terrain associées à ces images.

Les rares améliorations sont parfois dues à une élimination de fausses alarmes aux bordures de l'image, là où un algorithme ne prend pas de décision à cause de la taille de la fenêtre de convolution qu'il utilise. C'est le cas du détecteur de Argyle sur l'image « 86016/aménagement paysager », la meilleure mesure-F, soit 25,9 %, est atteinte avec une fenêtre de largeur 9. Dans le cas du même algorithme avec restauration, la meilleure mesure-F est atteinte avec une fenêtre de largeur 13. Elle vaut 27,6 %. Bien que cela ne prouve pas qu'il y ait restauration, cet exemple est une confirmation supplémentaire de la tendance selon laquelle la méthode de restauration requiert plus de lissage qu'une détection de contours conventionnelle.

Sur les deux images IKONOS, les algorithmes de Sobel global, Sobel adaptatif, Kirsch, Frei-Chen et Argyle ont tous donné des améliorations significatives lors de la fusion. Ces résultats sont semblables à ceux obtenus sur les images aéroportées.

Tous les jeux de données sur lesquels une restauration a pu être faite – images aéroportées, terrestres et Ikonos – ont pu être restaurés en utilisant le réseau de neurones entraîné sur une

image aéroportée. Ainsi, non seulement la quatrième hypothèse – que la restauration peut être appliquée à plusieurs types d'images – a été vérifiée, mais les résultats suggèrent, en plus, qu'un même réseau de neurones peut convenir à plusieurs types de données. Cela laisse croire qu'il existe des similarités dans les contours de ces types d'images

5.5 Analyse de sensibilité

Les sections précédentes ont expliqué dans quelle mesure le réseau de neurones contribue à améliorer les performances d'un filtre classique. Pour mieux comprendre les causes de cette amélioration, il importe de distinguer l'effet de chaque indicateur sur la réponse du réseau.

L'entropie est le meilleur détecteur de contours parmi les indicateurs étudiés. Cependant, elle n'apporte qu'une faible amélioration à la qualité du résultat. C'est plutôt le filtre de Gabor qui domine le réseau, malgré sa faible capacité à identifier les limites de région aux échelles étudiées. Le filtre de Gabor est sensible aux éléments de texture; il permet d'améliorer la détection de contours basée sur le gradient des niveaux de gris. Quand l'indicateur de Gabor n'est pas atténué, le réseau n'est pas affecté.

Malgré le fait que l'indicateur SNR soit un détecteur donnant un résultat individuel meilleur que le filtre de Gabor, la présence du SNR dans le réseau détériore les performances de façon non-significative.

5.6 Conclusion partielle

Un réseau de neurones entraîné sur des indicateurs inspirés de ceux de Chalmond, même s'il ne permet pas de détecter les contours, peut fournir une information complémentaire au gradient de l'intensité. Cette information, lorsqu'elle est combinée au gradient, permet d'allonger et amincir les contours. La qualité de la carte obtenue est comparable à celle obtenue avec d'autres méthodes de pointe, comme l'algorithme de Argyle – une approximation du détecteur optimal de contours de type « marche » – et l'algorithme de Moon, qui est optimal pour les contours de type « rampe ». En présence de bruit, la méthode proposée nécessite l'utilisation d'une carte de gradient convenablement filtrée. La méthode développée fonctionne sur le domaine d'image sur lequel elle a été entraînée. En plus, il a été montré qu'un réseau de neurones entraîné sur les images aériennes permet de traiter les images satellites et terrestres. C'est surtout à cause du filtre de Gabor que la méthode parvient à différencier la texture des limites de régions.

6. Discussion des résultats

Cette section décrira plus en détails de quelles manières la méthodologie proposée est réellement plus utile qu'une méthode classique. Dans le cas de l'apprentissage automatique, les performances de l'algorithme sont influencées par le choix des indicateurs et par la technique d'entraînement. L'évaluation est influencée par la vérité-terrain sur laquelle est basé son calcul.

Au delà de l'atteinte des objectifs fixés au départ, il est important de réfléchir sur la position de cette thèse face aux autres travaux récents, de situer la qualité de nos résultats par rapport à ceux d'autres algorithmes.

Il faut donner un sens à cette thèse en répertoriant les nouveautés et l'originalité : déterminer d'éventuelles répercussions théoriques ou pratiques, nommer les avantages de notre méthode, s'intéresser à la portée de la thèse en rapport à la problématique. C'est l'occasion de présenter les limites, la généralisation possible des travaux et de faire l'inventaire de nouvelles questions de recherche.

6.1 Simplicité d'utilisation et économie de temps

Après une détection de contours effectué avec un détecteur classique, les contours détectés ne sont pas nécessairement des frontières de régions; ils sont généralement discontinus et peuvent contenir beaucoup de fausses alarmes. Les frontières de régions doivent être des courbes fermées. Pour obtenir de telles courbes, il est donc nécessaire d'appliquer des post-traitements, comme le suivi de contours, le remplissage de brèches, le lissage et l'amincissement. Cependant, ces opérations exigent beaucoup de temps de traitement. Un moyen de réduire ce temps est d'utiliser une technique de segmentation hybride (Fan *et al.*, 2001). Il est aussi possible de trouver des frontières plus complètes et plus fines, réduisant ainsi la complexité de la fermeture à effectuer. C'est ce que fait notre méthode.

Les premiers avantages se situent au niveau de la simplicité d'utilisation. La méthode proposée n'a pas trop de réglages internes compliqués; le paramétrage se limite au choix d'un seuil.

Quand un nouveau type d'image est disponible, le système doit apprendre à modéliser sa sensibilité aux caractéristiques connues. Pour ce faire, il suffit de lui fournir une image source jumelée à sa réalité du terrain. Le système s'en sert pour apprendre de lui-même. Après coup, il

peut améliorer les performances de n'importe quelle image du nouveau type, sans utiliser de réalité du terrain. Cet entraînement n'étant nécessaire qu'une fois sur un nouveau type d'image, il en résulte un gain de temps sur chaque utilisation subséquente de la méthode.

L'amélioration visuelle et quantitative apportée par la méthode devrait permettre à l'utilisateur final de fixer plus facilement le seuil de la détection de contours. Il pourrait s'en suivre un gain de temps significatif.

6.2 Adaptation de contours

En reconnaissance de formes, il est important de ne pas prétendre qu'un arrière-plan texturé contient des contours. Pour éviter cette erreur, le traitement est souvent divisé en deux étapes : la création d'une ébauche de carte de contours et son adaptation pour obtenir une carte détaillée. Pour créer une ébauche de carte, un détecteur spécifique aux contours est employé. Cette étape ne réagit pas à la texture, tout en identifiant les contours saillants. Ces contours sont souvent incomplets. Une deuxième détection, plus sensible, est alors utilisée pour compléter les contours. Pour mettre en place ces deux étapes, Phalke (2005) utilise deux algorithmes distincts, alors que Canny (1986) utilise un seuillage adaptatif.

Notre méthode se distingue par l'utilisation d'un seuillage global, comme dans les méthodes classiques de détection de contours. En principe, le paramètre de seuil devait permettre de sacrifier de la spécificité pour gagner de la sensibilité.

En pratique, lorsque notre méthode est appliquée à une image d'amplitude de gradient, cette dernière devient un peu moins spécifique aux contours, mais beaucoup plus sensible. Ce sont les conditions idéales pour compléter une ébauche de carte de contours. La fusion de notre indice de présence de contours avec la carte d'amplitude du gradient, effectuée avant le seuillage, est donc un prétraitement utile.

Récemment, Venkatesh *et al.* (2006) ont effectué l'adaptation de contours par réseau de neurones en employant l'approche des contours actifs. La chaîne de points de contrôle d'un contour actif est initialisée et mise à jour à l'aide de la sortie d'un algorithme de détection de contours appliqué à l'image. L'entraînement du réseau de neurones converge même si les contours ont une petite amplitude du gradient ou s'ils sont fragmentés. Cependant, lorsque le contour initial est éloigné

de la frontière réelle, il est nécessaire de prétraiter l'image de luminance en lui appliquant un filtrage gaussien. L'obligation d'utiliser une telle opération de « bas niveau » dans un algorithme de niveau intermédiaire est, selon les auteurs eux-mêmes, une limitation de l'algorithme.

6.3 Comparaison des effets observés à d'autres détecteurs

Sur les photos aériennes de South Florida, les essais effectués suggèrent que la méthode développée permet de maintenir ou d'améliorer la performance de l'algorithme de Moon, mais qu'elle ne permet pas d'améliorer l'algorithme de Canny. Bien que les performances obtenues soient inférieures à celles de l'algorithme de Canny, il demeure intéressant de se pencher sur les caractéristiques des contours issus de la méthode décrite ici. Suite à l'entraînement, le réseau de neurones acquiert la propriété d'inhiber le signal en bordure des contours de l'image. Ceci permet de mieux localiser le signal, un peu comme avec la suppression des non-maximums de Canny. Certains contours sont allongés, comme avec le seuillage par hystérésis. Le réseau de neurones inhibe aussi le signal dans plusieurs zones texturées ou bruitées.

Il y a aussi une ressemblance entre nos images et celles obtenues par Chalmond *et al.* (2001). La carte de prédictivité positive de Chalmond montre des contours fins et la carte de prédictivité négative de Chalmond réagit à toutes les discontinuités de l'image. Il est possible de reconnaître des tendances semblables dans nos images, où la réponse aux contours est fine avec une inhibition de la réponse latérale et où la réponse à plusieurs discontinuités de texture est elle aussi inhibée.

Konishi *et al.* (2003) observent des résultats semblables aux nôtres lorsqu'ils ajoutent un stade de groupement spatial à leur méthode statistique de détection de contours. Des propriétés similaires à l'hystérésis et la suppression des non-maximums émergent naturellement du processus d'apprentissage. Le groupement améliore significativement la qualité visuelle de nos résultats de détection de contours. Mais, paradoxalement, il ne donne qu'une petite amélioration au critère de performance. D'un autre côté, l'amincissement des contours n'est pas aussi efficace que la suppression des non-maximums employée par le détecteur de Canny. Ils attribuent cet effet à la quantification employée dans leur approche, qui peut créer des pixels voisins ayant des intensités de contours identiques. Il se peut qu'un phénomène semblable affecte la qualité de nos résultats.

Sur les Figures 4.9 et 4.10, notre méthode est appliquée au filtre de Argyle à plusieurs échelles.

Les échelles intermédiaires y sont les plus efficaces; un lissage insuffisant crée des fausses alarmes, alors qu'un lissage excessif détériore la localisation. Konishi *et al.* (1999), lorsqu'ils employaient une échelle unique, ont, eux aussi, obtenu des détections plus efficaces avec les échelles intermédiaires.

Nous avons obtenus les meilleurs résultats en réduisant le plus possible la taille des fenêtres de calcul. Martin (2002), Konishi *et al.* (1999) ainsi que d'Elder et Zucker (1998) atteignent eux aussi leurs meilleures cartes de contours lorsqu'ils utilisent la plus petite taille de fenêtre possible.

Nos échelles optimales sont ($\sigma_{\text{Gabor}}^* = 2$, $N_{\text{Ent}}^* = 3$, $N_{\text{SNR}}^* = 7$). Ces résultats chevauchent l'échelle optimale attribué au jeu de données de South Florida par Konishi *et al.* (1999), soit $\sigma_{\text{SF}}^* = 1$ ou $N_{\text{SF}}^* = 9$, tout en se distinguant de l'échelle qu'ils attribuent au jeu de données de Sowerby, soit $\sigma_{\text{Sow}}^* = 4$ ou $N_{\text{Sow}}^* = 9$. Chalmond *et al.* (2001) recommandent l'emploi de fenêtres de calcul d'une largeur $N_{\text{Chalmond}}^* \geq 9$ pour traiter une image aérienne infrarouge.

Les résultats que nous avons obtenus sur le jeu de données de Berkeley n'étaient pas concluants. Le fait que Martin (2002) trouve des $N_{\text{Grad}}^* = 13$ et $N_{\text{Texture}}^* = 33$ laisse croire que des résultats plus probants pourraient être obtenus en élargissant les fenêtres utilisées dans le calcul des indicateurs.

Nos meilleurs résultats sont obtenus avec une banque d'indicateurs d'une étendue de 2,6 octaves. Les indicateurs n'étaient pas tous utiles à la même échelle. Les résultats de Konishi *et al.* (1999), comme les nôtres, montrent que le mode multi-échelle est meilleur pour différencier les contours de texture des vraies frontières des objets. Leurs meilleurs résultats en mode multi-échelles utilisent une banque d'indicateurs d'une étendue de 2,0 octaves ($\sigma_{\text{Multi}}^* = \{1, 2, 4\}$).

6.4 Conséquences et limites dues aux choix méthodologiques

Les choix méthodologiques ont nécessairement des conséquences. Les paragraphes qui suivent en dressent un portrait général. L'exactitude de nos résultats pouvant être influencée par plusieurs facteurs, il est important d'en dresser un portrait. Nous nous intéresserons donc aux effets du choix des indicateurs, de l'entraînement, de la méthode de validation, de la vérité-terrain et du type d'images.

6.4.1 Choix des indicateurs

Conformément aux suppositions de Chalmond *et al.* (2001), il nous a été possible de trouver des indicateurs meilleurs que ceux utilisés dans leurs travaux. Conformément à l'opinion de Witten et Frank (1999), prendre la peine de faire un choix adéquat d'indicateurs était nécessaire pour rendre possible l'apprentissage du réseau de neurones au coeur de l'application développée.

En utilisant l'algorithme de Sobel comme modèle pour le comportement de ce même algorithme, Chalmond *et al.* (2001) biaisent leurs résultats. Il en va de même pour la nature des contours étudiés, qui, dans ce contexte, sont tous de type « marche ». Il est alors approprié d'utiliser seulement des filtres de Gabor symétriques. Cependant, nos vérités-terrain contiennent aussi des contours de type « ligne ». Il fallait s'attendre à constater une amélioration de la qualité de nos résultats lors de l'utilisation de filtres asymétriques conjointement aux filtres symétriques (Manjunath et Chellapa, 1993). Konishi *et al.* (1999) ont, eux aussi, utilisé des paires de filtres avec succès.

Les critères de Chalmond *et al.* (2001) ont été développés pour prédire les performances d'algorithmes de bas-niveau, en particulier celles du détecteur de contours de Sobel. Avec de légères modifications à la banque d'indicateurs, nous sommes parvenu à rehausser une carte de contours. Les cartes de Sobel ainsi rehaussées ont une mesure-F supérieure à celle du détecteur de Moon, optimal pour les contours de type « rampe ».

6.4.2 Entraînement

L'utilisateur final d'une application a des exigences pratiques particulières. Un algorithme ne saurait prétendre pouvoir lire les pensées d'un usager. Cependant, l'emploi de l'apprentissage automatique permet, dans une certaine mesure, de s'adapter aux préférences d'un usager. Lors de l'entraînement, l'utilisateur peut, via les exemples d'entraînement, spécifier le type de contours qu'il souhaite détecter.

En procédant de la sorte, nous nous affranchissons de l'emploi d'un modèle théorique de contours – nous ne prenons pas pour acquis que les contours sont de type « marche » ou de type « ligne ». Ainsi, en principe, l'algorithme tel que conçu détecte tous les contours spécifiés par la vérité-terrain et rien d'autre.

Un des avantages majeurs de ces méthodes, comme le disent Konishi *et al.* (2003) et Chalmond

et al. (2001), est que, après un premier entraînement supervisé, leur utilisation requiert peu ou pas de paramétrage. Dans le présent travail, nous avons constaté que lorsqu'une telle méthode est combinée avec une méthode classique de détection de contours, un nombre restreint de pixels d'entraînement suffit pour atteindre des résultats supérieurs à ceux obtenus avec la méthode classique employée seule.

Nous utilisons un classifieur de type perceptron multicouche, entraîné avec arrêt hâtif. Chalmond *et al.* (2001) utilisent une régression logistique et Konishi *et al.* (2003) emploient une méthode de seuillage multiple d'histogrammes multidimensionnels. Cette diversité entre les méthodes laisse croire que le choix des indicateurs a plus d'importance que le type de classifieur utilisé.

Pour faire notre entraînement, nous n'utilisons pas tous les pixels de l'image. Cela vérifie l'hypothèse de Chalmond *et al.* (2001) selon laquelle il n'est pas nécessaire de répertorier tous les contextes possibles pour entraîner l'algorithme, mais qu'il suffit d'avoir suffisamment d'exemples pour modéliser avec fiabilité la relation entre le contexte local et la sortie de l'algorithme (zones de contours et d'arrière-plan).

6.4.3 Vérité-terrain

Dans le présent travail, deux des trente deux images, « 202/alligator » et « cone », n'ont pas pu être améliorées significativement. On y décèle des contours dues à la végétation, mais absents de la vérité-terrain. Notre méthode ne parvient pas à éliminer complètement la texture d'arrière-plan causée par la végétation. Konishi *et al.* (1999) ont aussi rencontré cette difficulté et ont remarqué que les différences de performance entre les images est étroitement liée à la proportion de végétation qu'elles contiennent.

Ces derniers ont nommé « pommes pourries » les images pour lesquelles la carte de contours est de qualité significativement inférieure à celle de l'ensemble des images. Dans le cadre de leurs travaux, une mauvaise correspondance entre les cartes de contours et leur vérité-terrain a été observée pour cinq des cent trois images. Heath (1996) avait éliminé une image parmi les trente huit choisies pour sa thèse.

Ces trois échantillonnages ne sont pas significativement différents. Pour un échantillon de cent

trois éléments, tout taux de « pommes pourries » entre 2,5 % et 11 % n'est pas significativement différent des taux obtenus dans le travail de Konishi, de Heath ou dans le présent travail.

6.4.4 Type d'images

Visuellement, notre méthode performe mieux que les algorithmes classiques lorsque l'arrière-plan est encombré. La mesure-F vient confirmer cette observation. De leur côté, Konishi *et al.* (2003) constatent aussi une amélioration dans les zones texturées. Ils donnent une mesure quantitative des avantages apportés par les raffinements de leur algorithme en utilisant l'information de Chernoff et la distance de Bhattacharyya.

Nous pouvons réutiliser notre méthode telle quelle sur plusieurs types d'images. Le réseau entraîné sur les images aériennes de South Florida performe bien sur les images IKONOS et sur les images terrestres. Comme nous, Konishi *et al.* (2003) font une telle réutilisation entre les jeux de données de Sowerby et South Florida. Ils font remarquer qu'un tel comportement implique que la détection statistique de contours est robuste aux erreurs dans la vérité-terrain, dans la mesure où il aurait été impossible d'atteindre un tel niveau d'adaptation si les statistiques des contours étaient très sensibles aux critères de vérité-terrain plutôt différents utilisés dans leurs deux jeux de données. Chalmond *et al.* (2001) ont entraîné leur détecteur sur une image aérienne infrarouge et l'ont réutilisé sur une image optique.

6.5 Conclusion partielle

Il nous importait de savoir s'il est possible, en corrigeant automatiquement la carte, de diminuer la perte de temps associée au choix d'algorithme et de paramètres. Il est raisonnable de croire que les images « difficiles à traiter » induisent des pertes de temps plus importantes.

Selon Yuille et Coughlan (2000), ce niveau de difficulté peut être estimé en utilisant la théorie de l'information pour prendre en compte les événements rares. Le *paramètre d'ordre* ainsi déterminé quantifie la difficulté de plusieurs tâches de vision numérique, telles qu'identifier les éléments de contours parmi des éléments d'arrière-plan.

D'après leur modèle, la qualité des indicateurs utilisés détermine une borne inférieure sur la prévalence des contours détectables. Notre méthode peut être vue comme un prétraitement permettant d'adapter la carte de gradient au contenu local de l'image. Ainsi, moins les contours sont abondants dans une image, plus il serait souhaitable d'utiliser notre méthode.

7. Conclusion et recommandations

Après avoir comparé la présente étude aux travaux récents, il importe maintenant de mettre de l'avant les contributions de la recherche effectuée, de faire mention de ses limitations et de recommandations adaptées, puis de terminer en proposant des perspectives de recherche.

7.1 Recherche effectuée

Dans leur article sur la prédiction des performances de détecteurs de contours, Chalmond *et al.* (2001) n'abordent pas le problème de savoir comment exploiter les cartes de performance. Ces cartes seront utilisées conjointement avec une méthode de reconnaissance de formes, soit pour faire le rejet initial de formes par prédiction des performances, soit pour faire de la confirmation de formes, tel que suggéré par Phalke (2005). Dans une telle démarche, une détection préliminaire est faite par un détecteur « classique »; elle est ensuite complétée par un traitement additionnel. Le présent travail s'inscrit dans ce dernier type d'approche.

Chalmond *et al.* (2001) s'interrogent sur comment combiner une carte de performances à d'autres types de données. En réponse à ce questionnement, nous proposons de combiner la carte de performances avec la carte de gradient pour permettre des décisions plus exactes.

Devant la variété de détecteurs disponibles, il est souvent difficile de faire un choix. Bien que l'utilisateur base ce choix sur son expérience, l'utilisateur risque de procéder par tâtonnement, d'essayer plusieurs détecteurs avant de fixer son choix. Il peut aussi arriver que l'utilisateur choisisse un détecteur qui n'est pas celui qui est le mieux adapté à l'image, et perde du temps à ajuster les paramètres pour obtenir une carte de contours qui ne peut pas être la meilleure. Notre problématique est de savoir s'il est possible de diminuer cette perte de temps en corrigeant automatiquement la carte.

La présente étude a permis de développer un logiciel comprenant un réseau de neurones entraîné pour prédire la présence de contours. Ce réseau de neurones permet d'améliorer les décisions de détecteurs de contours, en réduisant le nombre de pixels de fausses alarmes et de contours manqués. L'efficacité cet outil est comparable à celle d'autres méthodes et en fait un choix intéressant pour la détection de contours. Grâce au logiciel développé, il est possible de réutiliser la méthodologie; cette dernière a permis d'opérationnaliser la méthode proposée.

Nos travaux ont permis d'améliorer la banque d'indicateurs de Chalmond, rendant possible le

prétraitement avant le seuillage de la carte de gradient. Cette restauration, effectuée au niveau de la carte en niveaux de gris plutôt que sur la carte binaire, permet d'obtenir des résultats de meilleure qualité que ceux fournis par un détecteur de contours conventionnel.

Lorsqu'employé conjointement à un filtrage exponentiel, ce détecteur est robuste au bruit et permet une meilleure localisation spatiale qu'un filtrage exponentiel employé seul. La méthode développée permet, lorsqu'elle est utilisée pour restaurer une carte de Sobel, d'obtenir un résultat de qualité entre le détecteur optimal pour les contours de type « rampe » et celui de Canny. Les contours corrigés sont plus fins, plus complets et mieux localisés que les contours originaux. Le choix d'un détecteur de contours pour traiter une image peut donc être considéré valide.

La possibilité de réutiliser le filtre, sans réentraînement est intéressante. La simplicité du paramétrage lors de l'utilisation est aussi un avantage. Ces deux facteurs répondent à un besoin de réduire le temps d'utilisation du logiciel.

Bien que nous considérons la présente méthodologie comme utile et assez générale, nous ne voulons pas prétendre qu'elle capture toutes les nuances possibles du problème de détection de contours. Son utilité pour une application donnée devra être déterminée au cas par cas, comme pour n'importe quelle autre méthode.

7.2 Limitations et recommandations

Les différences de qualité observées entre notre méthode et celle de Canny semblent être dues à l'utilisation, ou non, de post-traitements. Notre méthode propose un seuillage brut de l'image de gradient, alors que l'implémentation de Canny emploie un seuillage à hystérésis. Un post-traitement adapté aux particularités de notre méthode aurait avantage à être mis au point. Cette opération pourrait prendre la diminution de l'étendue des valeurs numériques des pixels suite à la fusion par multiplication.

Étant donné la rareté des jeux d'images avec vérité-terrain pour les contours, nous n'avons touché que les domaines de bâti non dense sur image IKONOS, de la photographie aérienne d'une base militaire et de la reconnaissance au sol d'objets tirés de la vie quotidienne. L'évaluation éventuelle de notre méthode sur d'autres domaines nécessitera la création de nouvelles de vérités-terrain.

Il paraît intéressant d'appliquer notre méthode neuronale à la restauration locale de contours,

guidée, par exemple, par un opérateur humain. Cela éviterait le problème d'ajout sporadique de bruit observé lorsque notre méthode est appliquée à l'image entière. Un plugiciel pourrait, être bâti pour réutiliser notre méthodologie, par exemple dans l'application de traitement d'images « GIMP ». Le développement de cet outil permettrait d'effectuer des retouches contextuelles.

7.3 Perspectives

Nous espérons que ce travail contribue à stimuler l'intérêt pour le développement d'algorithmes reliés à la préparation, ou la restauration d'une détection de contours. Pour le volet « préparation », il pourrait être intéressant, notamment, de développer un algorithme de rejet de formes, ou de dépistages de contours qui permettrait, grâce à une méthodologie inspirée des biostatistiques, d'éliminer certaines zones d'une image exemptes de contours. Un volet « post-traitement » pourrait aussi être ajouté; l'utilisation d'un algorithme d'amincissement adapté aux contours sinueux paraît intéressante. Pour remplacer la suppression des non-maximums de l'algorithme de Canny, il est peut-être avantageux d'employer un algorithme de ligne des partage des eaux (Roerdink et Meijster, 2001) ou l'amincissement de Stentiford et Mortimer (1983).

Le volet « restauration » correspond au développement éventuel d'une méthodologie permettant d'élargir la portée du présent travail. Une telle étude devrait utiliser des techniques de sélection d'indicateurs pour évaluer la pertinence d'une grande variété d'indicateurs de contours. Les travaux de Konishi *et al.* (2003) peuvent servir d'inspiration pour une telle recherche. La recherche sur la détection de contours de bas-niveau peut aussi être poussée plus loin en étudiant, par exemple, la fusion des deux meilleurs filtres « classiques » utilisés dans cette étude, soit le filtre exponentiel (Moon *et al.*, 2002) et le filtre gaussien (Argyle, 1971; Canny, 1986). Il serait aussi intéressant d'évaluer l'influence des indicateurs de texture de Haralick *et al.* (1973), les dérivées d'ordre 1 à 4 des filtres gaussiens (Iverson et Zucker, 1995), les moments statistiques (Auclair-Fortier, 1999; Petrou *et al.*, 2006), ainsi que les indices de Frei et Chen (1977). Différents classifieurs pourraient ensuite être comparés, notamment la méthode des k plus proches voisins ou un arbre de décision. Les résultats pourraient probablement être optimisés par l'utilisation d'un méta-algorithme tel que le *bagging*, le *stacking* ou le *boosting*. Pour ce qui est de la validation, nous recommandons une analyse de sensibilité pour déterminer les effets des variations dans la vérité-terrain.

8. Références

- Abdou, I.E., and Pratt, W.K. (1979-05) Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors. *Proceedings of the IEEE*, vol. 67, n° 5, p. 753-763.
- Altman, D.G., and Bland, J.M. (1994-06-11) Diagnostic tests. 1: Sensitivity and specificity. *BMJ (British Medical Journal)*, vol. 308, n° 6943, p. 1552.
- Altman, D.G., and Bland, J.M. (1994-07-09) Diagnostic tests 2: Predictive values. *BMJ (British Medical Journal)*, vol. 309, n° 6947, p. 102.
- Amani, M. (2006) Détection et exploitation d'ombre de bâti sur les images de très haute résolution spatiale (IKONOS) : application au milieu urbain (Sherbrooke). Thèse de doctorat, Université de Sherbrooke, Sherbrooke, 134 p.
- Argyle, E. (1971-02) Techniques for Edge Detection. *Proceedings of the IEEE*, p. 285-287.
- Auclair-Fortier, M.-F. (1999) Extraction de caractéristiques : contours multispectraux, contours de textures et routes. Mémoire de maîtrise, département de mathématiques et informatique, Université de Sherbrooke, Sherbrooke, 85 p.
- Baeza-Yates, R., and Ribeiro-Neto, B. (1999) *Modern information retrieval*. ACM Press, New York. 513 p.
- Baker, S., and Nayar, S.K. (1996) Pattern Rejection. In *Proceedings of Computer Vision and Pattern Recognition*, San Francisco.
- Bao, P., Zhang, L., and Wu, X. (2005-09) Canny Edge Detection Enhancement by Scale Multiplication. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, n° 9, p. 1485-1490.
- Ben-Arie, J., and Rao, K.R. (1993) A Novel Approach for Template Matching by Nonorthogonal Image Expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, n° 1, p. 71-84.
- Bergholm, F. (1987-06) Edge focusing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, p. 726-741.
- Borghys, D., Lacroix, V., and Perneel, C. (2002) Edge and Line detection in polarimetric SAR images. In *Proceedings of International Conference on Pattern Recognition*, p. 921-924.
- Borra, S. and Sarkar, S. (1997-11) A Framework for Performance Characterization of Intermediate Level Grouping Modules. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, n° 11, p. 1306-1312.
- Boshra, M., and Bhanu, B. (1999-06) Performance Prediction and Validation for Object Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, p. 380-386. <<http://citeseer.ist.psu.edu/boshra99performance.html>> (Consulté le 2007-08-08)
- Bowyer, K.W. (2000) Experiences with Empirical Evaluation of Computer Algorithms. p. 3-16, in Klette, R., Stiehl, H.S., Viergever, M., and Vincken, K.L. (éd.) *Performance Characterization in Computer Vision*. Kluwer, Netherlands, 315 p.

- Bowyer, K., Kranenburg, C., and Dougherty, S. (1999) Edge Detector Evaluation Using Empirical ROC Curves. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 23-25 June. vol. 1, p. 354-359.
- Bowyer, K., Kranenburg, C., and Dougherty, S. (2001) Edge Detector Evaluation Using Empirical ROC Curves. *Computer Vision and Image Understanding*, vol. 84, p. 77-103.
- Bowyer, K.W., Phillips, P.J. (1998-06), Overview of Work in Empirical Evaluation of Computer Vision Algorithms, *in* Bowyer, K.W., Phillips, P.J. (eds.) *Empirical Evaluation Techniques in Computer Vision*. Wiley-IEEE Computer Society Press, 262 p.
- Brogaard, S., and Ólafsdóttir, R. (1997-10) Ground-truths or Ground-lies? Environmental sampling for remote sensing application exemplified by vegetation cover data. *Lund Electronic Reports in Physical Geography*, n° 1, Department of Physical Geography, Lund University, Sweden.
- Cannon, M., Fugate, M., Hush, D.R., and Scovel, C. (2003-05) Selecting a restoration technique to minimize OCR error. *IEEE Transactions on Neural Networks*, vol. 14, n° 3, p. 478-490.
- Canny, J. (1986-11) A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, n° 6, p. 679-698.
- Carletta, J. (1995) Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, vol. 30, n° 11, 6 p.
- Chalmond, B., and Graffigne, C. (1999) A note on contextual performance prediction for image analysis algorithms, Technical Report 2000-30, CMLA, École Normale Supérieure, Cachan, France, 20 p. <<http://www.cmla.ens-cachan.fr/~chalmond/>> (Consulté le 2008-01-28)
- Chalmond, B., Graffigne, C., Prenat, M., and Roux, M. (2001) Contextual Performance Prediction for Low-Level Image Analysis Algorithms. *IEEE Transactions on Image Processing*, vol. 10, n° 7, p. 1039-1046.
- Cho, K., Meer, P., and Cabrera J. (1997) Performance Assessment Through Bootstrap. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, n° 11, p. 1185-1198. <<http://citeseer.ist.psu.edu/article/cho95performance.html>> (Consulté le 2007-08-06)
- Clark, A.F., and Courtney, P. (2000) Databases for Performance Characterization. p. 29-40, *in* Klette, R., Stiehl, H.S., Viergever, M., and Vincken, K.L. (éd.) *Performance Characterization in Computer Vision*. Kluwer, Netherlands, 315 p.
- Comaniciu, D., and Meer, P. (2002-05) Mean Shift: A Robust Approach Towards Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, n° 5, 18 p.
- Courtney, P. (2000) Evaluation and Validation of Computer Vision Algorithms : A perspective from a Vision Company. p. 17-28, *in* Klette, R., Stiehl, H.S., Viergever, M., and Vincken, K.L. (éd.) *Performance Characterization in Computer Vision*. Kluwer, Netherlands, 315 p.
- Duda, R.O., Hart, P.E., and Stork, D.G. (2000) *Pattern Classification*. Second Edition. John Wiley & Sons, New York. 654 p.
- Dunn, D., Higgins, W.E., Wakeley, J. (1995) Texture Segmentation Using 2-D Gabor Elementary Functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, n° 2, p.

130-149.

Elder, J.H. and Zucker, S.W. (1996) Space Scale Localization, Blur, and Contour-Based Image Coding. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, p. 27.

Elder, J.H. and Zucker, S.W. (1998-07) Local Scale Control for Edge Detection and Blur Estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, n° 7, p. 699-716.

Elder, J.H., and Goldberg, R.M. (2002) Ecological statistics of Gestalt laws for the perceptual organization of contours. Journal of Vision, vol. 2 n° 4, p. 324-353, <<http://journalofvision.org/2/4/5/>> (Consulté le 2007-07-30)

Elder, J.H., Krupnik, A., and Johnston, L.A. (2003-06) Contour Grouping with Prior Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, n° 6, p. 661-674.

Fan, J., Yau, D. K. Y., Elmagarmid, A. K., Aref, W. G. (2001) Automatic Image Segmentation by Integrating Color-Edge Extraction and Seeded Region Growing. IEEE Trans. on Image Processing, Vol. 10, No. 10, October 2001.

Fjørtoft, R., Lopès, A., Marthon, P., and Cubero-Castan, E. (1998) An Optimal Multiedge Detector for SAR Image Segmentation. IEEE Transactions on Geoscience and Remote Sensing, vol. 36, n° 3, p. 793-802.

Florack, L. (2000) Motion Extraction : An Approach Based On Duality and Gauge Theory. p. 69-80, *in* Klette, R., Stiehl, H.S., Viergever, M., and Vincken, K.L. (éd.) Performance Characterization in Computer Vision. Kluwer, Netherlands, 315 p.

Forsyth, D.A., and Ponce, J. (2003) Computer Vision : A Modern Approach. Prentice-Hall, 693 p.

Freeman, W.T., and Adelson, E.H. (1991-09) The Design and Use of Steerable Filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, n° 9, p. 891-906.

Frei, W., and Chen, C.-C. (1977) Fast Boundary Detection: A Generalization and a New Algorithm. IEEE Transactions on Computers, vol. C-26, n° 10, p. 988-998.

Gagnon, L., and Jouan, A. (1997) Speckle filtering of SAR images - A comparative study between complex-wavelet-based and standard filters. *In* Proceedings of SPIE Annual Meeting: Wavelet Applications in Signal and Image Processing V (SPIE #3169), San Diego.

Ghosh, A., and Petkov, N. (2005-11) Robustness of Shape Descriptors to Incomplete Contour Representations. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, n° 11, p. 1793-1804.

Globerson, A., and Roweis, S. (2006) Nightmare at Test Time: Robust Learning by Feature Deletion. *In* Proceedings of 23rd International Conference on Machine Learning, p. 353-360.

Graham P., and Bull, B. (1998) Approximate standard errors and confidence intervals for indices of positive and negative agreement. Journal of Clinical Epidemiology, vol. 51, n° 9, p. 763-771.

Grigorescu, C., Petkov, N., and Westenberg, M.A. (2003-07) Contour detection based on non-classical receptive field inhibition. IEEE Transactions on Image Processing, vol. 12, n° 7, p.

729-739.

Hajjar, A., and Chen, T. (1999-01) A VLSI Architecture for Real-Time Edge Linking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, n° 1, p. 89-94.

Hancock, E.R., and Kittler, J. (1991) Adaptive Estimation of Hysteresis Thresholds. *In Proceedings of IEEE Computer Vision and Pattern Recognition Conference*, p. 196-201.

Haralick, R.M., Shanmugam, K., and Dinstein, I. (1973) Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, n° 6, p. 610-621.

Haralick, R.M. (1984-01) Digital Step Edges from Zero Crossing of Second Directional Derivatives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, n° 1, p. 58-68.

Haralick, R., Cinque, L., Guerra, C., Levialdi, S., Weng, J., Huang, T.S., Meer, P., Shirai, Y., Draper, B.A., and Beveridge, J.R. (1994) Dialogue: Performance characterization in computer vision. *CVGIP: Image Understanding*, vol. 60, p. 245-265.

Haykin, S. (1999) *Neural Networks. A Comprehensive Foundation*. 2nd edition. Pearson Education, India, 823 p.

Heath, M.D. (1996) A Robust Visual Method for Assessing the Relative Performance of Edge Detection Algorithms. Master's thesis, University of South Florida. 132 p.

Heath, M.D., Sarkar, S., Sanocki, T., and Bowyer, K.W. (1997) A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, n° 12, p. 1338-1359.

Hoover, A., Jean-Baptiste, G., Jiang, X.Y., Flynn, P.J., Bunke, H., Goldgof, D.B., Bowyer, K., Eggert, D.W., Fitzgibbon, A., and Fisher, R.B. (1996-07) An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, n° 7, p. 673-689.

Hueckel, M.H. (1973-10) A Local Visual Operator Which Recognizes Edges and Lines. *Journal of the Association for Computing Machinery*, vol. 20, n° 4, p. 634-647.

Hueckel, M.H. (1974-04) Erratum: « A Local Visual Operator Which Recognizes Edges and Lines ». *Journal of the ACM*, vol. 21, n° 2.

Iverson, L.A., and Zucker, S.W. (1995-10) Logical/linear operators for image curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, n° 10, p. 982-996.

Jiang, X. (2000-11) An Adaptive Contour Closure Algorithm and Its Experimental Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, n° 11, p. 1252-1265.

Jiménez-Alaniz, J.R., Medina-Banuelos, V., and Yáñez-Suárez, O. (2006-01) Data-Driven Brain MRI Segmentation Supported on Edge Confidence and A Priori Tissue Information. *IEEE Transactions on Medical Imaging*, vol. 25, n° 1, p. 74-83.

Kanungo, T., Jaisimha, M.Y., Palmer, J., and Haralick, R.M. (1995-12) A Methodology for Quantitative Performance Evaluation of Detection Algorithms. *IEEE Transactions on Image Processing*, vol. 4, n° 12, p. 1667-1674.

- Kass, M., Witkin, A., and Terzopoulos, D. (1988) Snakes: Active Contour Models. *International Journal of Computer Vision*, p. 321-331.
- Klepko, R. (1997) Automated Edge Detection for RADARSAT Images. *In Proceedings of Geomatics in the Era of RADARSAT symposium.*
- Klette, R., Stiehl, H.S., Viergever, M., and Vincken, K.L. (réd.) *Performance Characterization in Computer Vision*. Kluwer, Netherlands, 315 p.
- Klette, R. and Zamperoni, P. (1996) *Handbook of image processing operators*. Wiley, 300 p.
- Konishi, S.M., Yuille, A.L., Coughlan, J. and Zhu, S.C. (1999-06) Fundamental Bounds on Edge Detection: An Information Theoretic Evaluation of Different Edge Cues. *In Proceedings of Computer Vision and Pattern Recognition*. Fort Collins, Colorado.
- Konishi, S., Yuille, A.L., Coughlan, J.M., and Zhu, S.C. (2003-01) Statistical Edge Detection: Learning and Evaluating Edge Cues. *IEEE Transactions on Pattern Analysis and machine Intelligence*, vol. 25, n° 1, p. 57-74.
- Laliberté, F., Gagnon, L., Sheng, Y. (2003-05) Registration and Fusion of Retinal Images: An Evaluation Study. *IEEE Transactions on Medical Imaging*, vol. 22, n° 5, p. 661-673.
- Lalonde, M., Gagnon, L., and Boucher, M.-C. (2001) Automatic visual quality assessment in optical fundus images. *In Proceedings of Vision Interface*, Ottawa.
- LeCun, Y., Bottou, L., Orr, G.B., and Müller, K.-R. (1998) Efficient backprop. *Neural Networks---Tricks of the Trade*, Springer Lecture Notes in Computer Sciences 1524, p. 5-50, [En ligne]. <<http://citeseer.ist.psu.edu/lecun98efficient.html>> (Page consultée le 2006-12-20)
- Lesage, F. and Gagnon, L. (2000) Experimenting Level Set-Based Snakes for Contour Segmentation in Radar Imagery. *In Proceedings of SPIE Aerosense: Visual Information Processing IX (SPIE #4041)*, Orlando.
- Lindeburg, T. (1996) Edge detection and ridge detection with automatic scale selection. Technical report ISRN KTH/NA/P{96/06}{SE, May 1996, Revised August 1998. *Int. J. of Computer Vision*, vol. 30, n° 2, 1998. Shortened version in *Proc. CVPR'96*, San Francisco, June 1996. 48 p.
- Lowe, D.G. (2004) Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, vol. 60, n° 2, p. 91-110.
- Mahamud, S., Williams, L.R., Thornber, K.K. and Xu, K. (2003-04) Segmentation of Multiple Salient Closed Contours from Real Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, n° 4, p. 433-444.
- Manjunath, B.S., and Chellapa, R. (1993-01) A Unified Approach to Boundary Perception: Edges, Textures, and Illusory Contours. *IEEE Transactions on Neural Networks*, vol. 4, n° 1, p. 96-107.
- Manjunath, B.S., Wu, P., Newsam, S., and Shin, H.D. (2000) A texture descriptor for browsing and similarity retrieval. *Signal Processing: Image Communication*. 10 p.
- Marr, D., and Hildreth, E. (1980) Theory of Edge Detection. *Proceedings of the Royal Society of London B*, vol. 207, p. 187-217.

- Martin, D. (2002) An Empirical Approach to Grouping and Segmentation, Ph.D. Dissertation, UC Berkeley, December 2002, 134 p.
- Martin, D.R., Fowlkes, C.C., and Malik, J. (2002-12) Learning to Detect Natural Image Boundaries Using Brightness and Texture. *In* Proceedings of NIPS, Vancouver.
- Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001) A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. *In* Proceedings of International Conference on Computer Vision. <<http://www.cs.berkeley.edu/projects/vision/bsds>> (Consulté le 2007-08-03)
- Martin, D.R., Fowlkes, C.C., and Malik, J. (2004-01) Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, n° 1, p. 1-20.
- Meer, P., Jolion, J.-M., and Rosenfeld, A. (1990-02) A Fast Parallel Algorithm for Blind Estimation of Noise Variance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, n° 2, p. 216-223.
- Mendoça, A.M., and Campilho, A. (2006-09) Segmentation of Retinal Blood Vessels by Combining the Detection of Centerlines and Morphological Reconstruction. *IEEE Transactions on Medical Imaging*, vol. 25, n° 9, p. 1200-1213.
- Min, J., Powell, M., and Bowyer, K.W. (2004-02) Automated Performance Evaluation of Range Image Segmentation Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 34, n° 1, p. 263-271.
- Moon, H., Chellappa, R., and Rosenfeld, A. (2002) Optimal Edge-Based Shape Detection. *IEEE Transactions on Image Processing*, vol. 11, n° 11, p. 1209-1226.
- Nair, D., Mitiche, A., and Aggarwal, J.K. (1995-10-23--1995-10-26) On Comparing the Performance of Object Recognition Systems. *In* Proceedings of IEEE International Conference on Image Processing, vol. 2, p. 631-634.
- Nakache, D., et Métais, E. (2005-05-24--2005-05-27) Évaluation: nouvelle approche avec juges. p. 555-570, *in* Compte rendu d'INFORSID 2005, Grenoble.
- Nalwa, V.S., and Binford, T.O. (1986-11) On Detecting Edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, n° 6, p. 699-714.
- Parker, J.R. (1997) *Algorithms for Image Processing and Computer Vision*. Wiley, 417 p.
- Petrou, M., and Kittler, J. (1991-05) Optimal Edge Detectors for Ramp Edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, n° 5, p. 483-491.
- Petrou, M., Kovalev, V.A., and Reichenbach, J.R. (2006-10) Three-Dimensional Nonlinear Invisible Boundary Detection. *IEEE Transaction on Image Processing*, vol. 15, n° 10, p. 3020-3032.
- Phalke, S., and Couloigner, I., (2004). Change Detection of Man made Objects using GIS data & Remotely-Sensed Imagery. *In* Proceedings of the 24th EARSeL Symposium: New Strategies for European Remote Sensing, Dubrovnik, Croatia, May 25-27, p. 191-198.

- Poggio, T. (1982-05) Visual Algorithms. MIT AI Memo n° 683, 28 p.
- Pratt, W.K. (1991) Digital image processing, 2nd ed. Wiley, New York, 698 p.
- Prechelt, L. (1998) Early Stopping - but when? p. 55-69, *in* Orr, G.B., and Müller, K.-R (Eds.) Neural Networks: Tricks of the trade, Lecture Notes in Computer Science 1524, Springer Verlag, Heidelberg, 432 p.
- Ravid, R., and Levabib, N. (1992-06) Maximum-likelihood CFAR for Weibull background. IEE Proceedings-F, vol. 139, n° 3, p. 256-264.
- Roberts, L. (1963-05-22) Machine Perception of Three-Dimensional Solids. Lincoln Laboratory Technical Report #315. <<http://www.lroberts.us/files/mach-per-3D-solids.html>> (Consulté le 2007-10-09)
- Roerdink, J.B.T.M., and Meijster, A. (2001) The Watershed Transform: Definitions, Algorithms and Parallelization Strategies , *Fundamenta Informaticae*, vol. 41, p. 187-228.
- Rosenfeld, A. (1970-05) A Nonlinear Edge Detection Technique. *Proceedings of the IEEE*, p. 814-817.
- Rothwell, C., Mundy, J., Hoffman, W. and Nguyen, V.-D. (1994-12) Driving Vision by Topology. Technical Report n° 2444, INRIA. <<http://citeseer.ist.psu.edu/article/rothwell94driving.html>> (Page consultée le 2006-20-12)
- Rousseau, K. (2003-04) Sélection semi-automatisée images-algorithmes appliquée à la cartographie - Phase I : Évaluation de l'approche par intelligence artificielle versus vision artificielle et apprentissage - Rapport d'étape. Montréal, CRIM. 47 p. [CRIM-03/04-01]
- Rousseau, K. (2004) PerfPredict v.1.0 User Manual. Montréal, CRIM. 22 p. [CRIM-04/03-04]
- Rousseau, K., Bénié, G. et Gagnon, L. (2003-10-14--2003-10-17) Prédiction des performances pour la détection de contours en télédétection. *In* Actes du 25^e Symposium canadien sur la télédétection (SCT) & 11^e Congrès de l'Association québécoise de télédétection (AQT). Montréal, Québec, Canada.
- Ruzon, M.A., and Tomasi ,C.(2001-11) Edge, Junction, and Corner Detection Using Color Distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, n° 11, p. 1281-1295.
- Salotti, M., Bellet, F., and Garbay, C. (1996-04) Evaluation of Edge Detectors: Critics and Proposal. *In* Proceedings of Workshop on Performance Characterization Vision Algorithms, Cambridge, U.K.
- Sappa, A.D. (2006-02) Unsupervised Contour Closure Algorithm for Range Image Edge-Based Segmentation. *IEEE Transactions on Image Processing*, vol. 15, n° 2, p. 377-384.
- Sarkar, S., and Boyer, K.L. (1991-11) On optimal infinite impulse response edge detection filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, n° 2, p. 1154-1171.
- Sarle, W.S. (2002-05-17) Neural Network FAQ, [En ligne]. <<ftp://ftp.sas.com/pub/neural/FAQ.html>> (Site consulté le 2008-01-29)
- Shin, M.C., Goldgof, D.B., Bowyer, K.W., and Nikiforou, S. (2001-08) Comparison of Edge

Detection Algorithms Using a Structure from Motion Task . IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, vol. 31, n° 4, p. 589 -601.

Smith, S.M. (1995) Edge Thinning Used in the SUSAN Edge Detector. Technical Report TR95SMS5, Oxford University, 4 p.

Smith, S.M., and Brady, J.M. (1995) SUSAN – A New Approach to Low Level Image Processing. Technical Report TR95SMS1c, Oxford University, 59 p. Also in (1997-05) International Journal of Computer Vision, vol. 23, n° 1, p.45-78.

Sofka, M., and Stewart, C.V. (2003-05) Retinal Vessel Centerline Extraction Using Multiscale Matched Filters, Confidence and Edge Measures. IEEE Transactions on Medical Imaging, vol. 25, n° 12, p. 1531-1546.

Stentiford, F.W.M. and Mortimer, R.G. (1983-01) Some new heuristics for thinning binary handprinted characters for OCR. IEEE Trans on Systems, Man, and Cybernetics, vol. 13, n° 1, p. 81-84.

Stuckens, J., Coppin, P. R., and Bauer, M. E.(2000) Integrating Contextual Information with per-Pixel Classification for Improved Land Cover Classification . Remote sensing of environment, vol. 71, p. 282-296.

Suetens, P., Fua, P., and Hanson, A.J. (1992-03) Computational Strategies for Object Recognition. ACM Computing Surveys, vol. 24, n° 1, p. 5-61.

Sun Microsystems (2001) Java™ Look and Feel Design Guidelines. 2nd edition, Addison-Wesley, 449 p.

Swonger, C.W.R. (1997-10) Machine Vision, The Automated Extraction of Information from Images. <http://scx2.com/fourth_level_machine.htm> Consulté le 1999-02-03.

Thepaut, O., Kpalma, K., Ronsin, J. (2000) Automatic registration of ERS and SPOT multisensor images in a data fusion context. Forest Ecology and Management, vol. 128, p. 93-100.

Torre, V., and Poggio, T. (1984-08) On Edge Detection. MIT AI Memo 768.

Touzi, R., Lopes, A., and Bousquet, P. (1988-11) A Statistical and Geometrical Edge Detector for SAR Images. IEEE Transactions on Geoscience and Remote Sensing, vol. 26, n° 6, p. 764-773.

Umbaugh, S.E. (1998) Computer Vision and Image Processing : a practical approach using CVIPtools. Prentice-Hall, Upper Saddle, New Jersey. ISBN 0-13-264599-8. 504 p.

Vasconcelos, M.J.P., Gonçalves, A., Catry, F.X., Paul, J.U., and Barros, F. (2002) A working prototype of a dynamic geographical information system. International Journal of Geographical Information Science, vol. 16, n° 1, p. 69-91.

Venkatesh, Y.V., Raja, S.K., and Ramya, N. (2006-04) Multiple Contour Extraction From Graylevel Images Using an Artificial Neural Network. IEEE Transactions on Image Processing, vol. 15, n° 4, p. 892-899.

Vincken, K.L., Koster, A.S.E., Graff, C.N. de, and Viergever, M.A. (2000) Model-Based Evaluation of Image Segmentation Methods. p. 299-311, in Klette, R., Stiehl, H.S., Viergever, M., and Vincken, K.L. (éd.) Performance Characterization in Computer Vision. Kluwer,

Netherlands, 315 p.

Voirin, Y. (2004) Élaboration d'un système à base de règles pour l'identification des zones perturbées en milieu forestier. Thèse de doctorat, Département de Géographie et Télédétection, Faculté des Lettres et Sciences Humaines, 170 p.

Walessa, M., and Datcu, M. (2000-09) Model-Based Despeckeling and Information Extraction from SAR Images. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, n° 5, p. 2258-2269.

Wang, S., Kubota, T., Siskind, J.M., and Wang, J. (2005-04) Salient Closed Boundary Extraction with Ratio Contour . *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, n° 4, p. 546-561.

WenYin, L. and Dori, D. (1999) Principles of Constructing a Performance Evaluation Protocol for Graphics Recognition algorithms. p. 97-106, *in Performance Characterization and Evaluation of Computer Vision Algorithms*, Kluwer.

Witten, I.H., and Frank, E. (1999-10-11) *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann. 416 p.

Xu, W., Jenkin, M., and Lespérance, Y. (2006) A Multi-Channel Algorithm for Edge Detection Under Varying Lighting. *In Proceedings of Computer Vision and Pattern Recognition*, p. 1885-1892.

Xue, X. and Wu, X. (2005-10) Directly Operable Image Representation of Multiscale Primal Sketch. *IEEE Transactions on Multimedia*, vol. 7, n° 5, p. 805-816.

Yang, J., Liu, L., Jiang, T., and Fan, Y. (2003) A modified Gabor filter design method for fingerprint image enhancement. *Pattern Recognition Letters*.

Yitzhaky, Y., and Peli, E. (2003-08) A Method for Objective Edge Detection Evaluation and Detector Parameter Selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, n° 8, p. 1027-1033.

Yuille, A.L., and Coughlan, J.M. (2000-02) Fundamental Limits of Bayesian Inference: Order Parameters and Phase Transitions for Road Tracking . *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, n° 2, p. 160-173.

Zhang, Q., Pavlica, G., Chena, W., Fräsera, R., Leblanca, S., and Cihlara, J. (2005) A semi-automatic segmentation procedure for feature extraction in remotely sensed imagery. *Computers & Geosciences*, vol. 31, p. 289-296.

Zhoua, Y., Starkeyb, J., Mansinhab, L. (2004) Segmentation of petrographic images by integrating edge detection and region growing. *Computers & Geosciences*, vol. 30, p. 817-831.

Ziou, D. and Tabbone, S. (1998) Edge Detection Techniques An Overview. *International Journal of Pattern Recognition and Image Analysis*, vol. 8, n° 4, p. 537-559.

Annexe 1 - Bibliographie thématique

Cet annexe donne une liste de documents pertinents à la thèse, mais non-cités dans le texte. Ils sont présentés ici pour apporter un complément à la revue de littérature. Il s'agit de manuels de référence, ainsi que d'autres documents sur la détection de contours, sur l'évaluation des performances et sur les indicateurs de présence de contours.

Manuels de référence

Angers, M. (1996) Initiation aux méthodes de recherche en sciences humaines. 2ème édition, CEC.

Cover, M. and Thomas, J.A. (1991) Elements of Information Theory. Wiley Series in telecommunication. John Wiley & Sons, Toronto, 542 p.

DeGroot, M.H., Schervish, M.J. (2002) Probability and Statistics, Third Edition. Addison-Wesley, Boston, 816 p.

Gosling, J., Joy B., Steele, G., and Bracha, G. (2005) The Java™ Language Specification, Third Edition. Addison-Wesley, Boston, 684 p.

Lehmann, E.L. (2006) Nonparametrics: Statistical Methods Based on Ranks. Springer-Verlag, New York, 480p.

Papoulis, A. (1991) Probability, Random Variables, and Stochastic Processes, 3rd Edition. McGraw-Hill, 661 p.

Pratt, W.K. (1978) Digital image processing. Wiley, New York, 750 p.

Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T. (1988) Numerical Recipes in C, The Art of Scientific Computing, Cambridge University Press, New York, 735 p.

Schoonjans, F. (2007-08-15) MedCalc manual, version 9.3.7.0. MedCalc Software, Mariakerke, Belgium. <<http://www.medcalc.be/manual/index.php>> (Consulté le 2007-09-11)

Schowengerdt, R.A. (1997) Remote Sensing. Models and Methods for Image Processing. Academic Press, 522 p.

Wahl, F.M. (1987) Digital Image Processing. Artech House, Boston, 183 p.

Zoubir, A. M. and Iskander, D. R. (2004) Bootstrap Techniques for Signal Processing. Cambridge University Press, 217 p.

Détection de contours

Bellon, O.R.P., and Silva, L. (2002-02) New Improvements to Range Image Segmentation by Edge Detection. IEEE Signal Processing Letters, vol. 9, n° 2, p. 43-45.

Denton, W.E., Duller, A.W.G., and Fish, P.J. (1995) Robust Boundary Detection for Skin Lesions. IEEE. 2 p.

Devernavy, F. (1995) A Non-Maxima Suppression Method for Edge Detection with Sub-Pixel Accuracy, Technical report no 2724, INRIA.

Elder, J. and Zucker, S. (1996b) Computing Computer Closure. In Proceedings of the Fourth

European Conference on Computer Vision, p. 399-412.

Hendry, A., Skingley, J., and Rye, A.J. (1988) Automated Linear Feature Detection and its Application to Curve Location in Synthetic Aperture Radar Imagery. p.1521-1524, in proceedings of IGARSS Symposium, Edinburgh, Scotland, 1988-09-13--1988-09-16.

Konishi, S., Yuille, A.L., and Coughlan, J.M. (2002) A Statistical Approach to Multi-Scale Edge Detection. In Proceedings of Generative-Model-Based Vision.

Lepage, R., and Crevier, D. (1994-10-31--1994-11-04) Multiresolution neural network for extraction of the primal sketch. In proceedings of SPIE's Conference on Intelligent Robots and Computer Vision XIII: Algorithms and Computer Vision, Boston. SPIE vol. 2353, p. 375-384.

Peli, E. (2002-01) Feature Detection Algorithm Based on a Visual System Model. Proceedings of the IEEE, vol. 90, n° 1, p. 78-93.

Perera, A., Tsai, C.-L., Flatland, R.Y., and Stewart, C.V. (2000) Maintining Valid Topology with Active Contours: Theory and Application. In proceedings of the Computer Vision and Pattern Recognition conference.

Thune, M., Olstad, B., and Thune, N. (1997) Edge Detection in Noisy Data Using Finite Mixture Distribution Analysis. Pattern Recognition, vol. 30, n° 5, p. 685-699.

Witkin, A.W. (1980-11) Shape from Contour. Massachusetts Institute of Technology, Artificial Intelligence Laboratory. Technical Report AI-TR-589, 106 p.

Évaluation des performances

Dougherty, S., Bowyer, K., Kranenburg, C. (1998) ROC curve evaluation of edge detector performance. *In* Proceedings of International Conference on Image Processing, 4-7 October. vol. 2, p. 525-529.

Imagery Resolution Assessments and Reporting Standards Committee (1996-03) Civil NIIRS Reference Guide. 40 p.

Johnson, T. (2007-07--2007-08) La force de persuasion des pourcentages. The Persuasive Power of Percentages. Santé canadienne, <<http://www.canadian-health.ca>> (Consulté le 2007-08-13)

Matei, B., Meer, P., and Tyler, D. (1998) Performance Assessment by Resampling: Rigid Motion Estimators. p. 72-95, in Bowyer, K.W. and Phillips, P.J. (éd.) Empirical Evaluation Techniques in Computer Vision, IEEE CS Press, Los Alamitos, CA.

Meer, P., Matei, B. and Cho, K. (2000) Input Guided Performance Evaluation. p. 115-124, in Klette, R., Stiehl, H.S., Viergever, M., and Vincken, K.L. (éd.) Performance Characterization in Computer Vision, Kluwer, Amsterdam.

Nguyen, T.B., and Ziou, D. (2000) Contextual and non-contextual performance evaluation of edge detectors. Pattern Recognition Letters, vol. 21, p. 805-816.

Null, N.B. and Bouzas, B.H. (2000) IQM Approach: Obtain Quality from Image Power Spectrum. MITRE. 8 slides. <<http://www.mitre.org/tech/mtf/>> (Page consultée le 2007-11-29).

Sanghavi, Darshak (2006-09-26 07:28 ET) Treat me? The Crucial Health Stat You've Never Heard Of. Slate, Medical Examiner.

Stone, M. (1974) Cross-validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society, B*, vol. 36. p. 111-147.

Yang, G. (2001) Detection of Micro-Aneurysms in Low-Resolution Color Retinal Images. M.Sc. DMI. 68 p. Yang (2001),

Zhao, Q., and Nishida, T. (1995) Using Qualitative Hypotheses to Identify Inaccurate Data. *Journal of Artificial Research*, vol. 3, p. 119-145.

Indicateurs de présence de contours

Aramini, M.J. (2002) Efficient Image Magnification by Bicubic Spline Interpolation, [En ligne]. <<http://members.bellatlantic.net/~vze2vrva/design.html>> (Page consultée le 2002-12-08)

Azencott, R.R., Wang, J.P., and Younes, L. (1997-02) Texture classification using windowed Fourier filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, p. 148-152.

Daugman, J. (1988) Complete discrete 2-D Gabor transform by neural network for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, n° 7, p. 1169-1179.

Lüdtke, N., and Wilson, R.C. (2003-07) A Mixture Model for Population Codes of Gabor Filters, *IEEE Transactions on Neural Networks*, vol. 14, n° 4, p. 794-803.

Tupin, F., Sigelle, M., and Maître, H. (2000-09) Definition of a spatial entropy and its use for texture discrimination, *International Conference on Image Processing, Vancouver*, p. 725-728.

Winter, A. (1998) Entropie et représentation multiéchelles pour l'analyse automatique d'images satellitaires ou aériennes à haute résolution. Thèse de Doctorat, Aérospatiale et ENST, février 1998, 267 p.

Annexe 2 - Analyse théorique de la figure de mérite de Pratt

Procédons maintenant à une brève analyse théorique de la figure de mérite de Pratt, dans le but de comparer son comportement à celui des autres méthodes d'évaluation, comme la mesure-F.

$$M_{Pratt} = \frac{1}{\max(P(G), P(A))} \sum_{i=1}^{P(A)} \frac{1}{1 + \beta d^2} \quad (A2.1)$$

Le minimum théorique $M_{Pratt} = \epsilon$ est atteint quand tous les pixels détectés sont éloignés de vrais pixels de contours.

Étudions maintenant le maximum théorique. Quand $\beta = 0$, les erreurs de positionnement ne sont pas pénalisées. La valeur de l'indice ne dépend que de $P(A)$ et $P(G)$. Elle est maximale et vaut 1 quand $P(A) = P(G)$. L'indice n'est pas informatif, car les contours détectés peuvent être situés n'importe où.

Quand $\beta > 0$, la distance pénalise. Le cas $\beta = +\infty$ correspond à la situation où seul un positionnement parfait est accepté. M_{Pratt} vaut alors $\min(\text{Précision}, \text{Rappel})$, où $\min(x, y)$ est la fonction minimum, $x + y - |x - y|$, où $|z|$ est la fonction valeur absolue. La figure de mérite atteint donc sa valeur maximale au point d'inflexion de la courbe précision-rappel.

Pour les valeurs réelles positives de β , la normalisation par $\max(P(G), P(A))$ entraîne trois cas différents de l'indice : $P(A) < P(G)$, $P(A) > P(G)$ et $P(A) \approx P(G)$.

Quand $P(A) < P(G)$ le maximum théorique est atteint quand certains pixels de la vérité-terrain ne sont pas détectés, mais que tous les pixels détectés sont situés à une distance nulle d'un vrai pixel. C'est la situation où la précision ($VP/(VP+FP)$) est égale à 1, mais où le rappel ($VP/(VP+FN)$) est inférieur à 1. Peu de bruit, beaucoup de silence.

Quand $P(A) > P(G)$, le maximum théorique est atteint quand tous les vrais pixels $P(G)$ sont détectés et bien positionnés, mais qu'il existe des fausses alarmes. Dans ce cas, le rappel vaut 1 et la précision est inférieure à 1. Peu de silence, beaucoup de bruit.

Quand $P(A) \approx P(G)$, le maximum théorique est atteint quand les contours détectés sont bien positionnés. Il y aura alors « peu de bruit » et « peu de silence. » C'est dans ce cas seulement que la précision et le rappel peuvent tous deux être près de 1. En principe, cela permet à l'indice

d'atteindre la vraie valeur maximale, pour laquelle la précision et le rappel ont la même grande valeur. La valeur exacte du maximum, entre 0 et 1, dépend de la valeur de β choisie.

L'analyse de la figure de mérite de Pratt suggère que, en principe, cet indice choisira comme maximum un point près du point d'inflexion de la courbe précision-rappel et que le paramètre β joue un rôle semblable à la limite de distance utilisée dans le calcul de la mesure-F.

Annexe 3 - Algorithme d'appariement de contours

Cet annexe décrit l'algorithme que nous avons utilisé pour mettre en correspondance deux cartes de contours avec une distance d'erreur tolérée de 2 pixels. La stratégie employée est de jumeler les pixels détectés avec un pixel de vérité-terrain qui soit le plus rapproché possible.

L'algorithme décrit est plus rapide que l'algorithme d'affectation de Munkres¹. Selon les tests effectués par Bowyer (2000) les différences entre les différents algorithmes d'appariement sont négligeables.

```
// X
// XXX
// XXOXX
// XXX
// X
SOIT PA=NOMBRE DE PIXELS A=1
SOIT PG=NOMBRE DE PIXELS G=1
SOIT ACC(G)=LISTE D'ACCUMULATEURS ASSOCIÉS À CHAQUE PIXEL G=1
SOIT TP=0;
POUR TOUTE POSITION P AVEC A(P)=1 {
  SI(G(P)=1) ALORS {
    A(P)=0, G(P)=0, TP++
  }
}
POUR DISTANCE D = 1, SQRT(2), 2 {
  POUR TOUTE POSITION P AVEC A(P)=1 {
    A(P)=0;
    SOIT N = NOMBRE DE G=1 À DISTANCE D DE P;
    POUR TOUTE POSITION Q À DISTANCE D DE P AVEC G(Q)=1 {
      ACC(G(Q)) += 1/N
    }
    POUR TOUT PIXEL G(P)=1 AVEC ACC(G(P))>1 {
      G(P)=0, TP++
    }
  }
}
SOMME = 0;
POUR TOUT G(P)=1 { SOMME += ACC(G(P)) }
TP+=SOMME;
SOIT FP=PA-TP;
SOIT FN=PG-TP;
SOIT TN=NBRE_PIXELS - TP - FP - FN
RETOURNE (TN, FP, FN, TP);
```

1 Ahuja, R., Magnanti, T., Orlin, J. (1993) Network Flows, Prentice Hall.

Annexe 4 - Rendement et mesure-F

L'image « École » du jeu de données aériennes de South Florida contient 5,5 % de contours (Figure A4.1). Pour le détecteur de Kirsch, le rendement optimal est de 95,7 % (Figures A4.2 et A4.3). Il est obtenu au seuil $\xi = 68$. Les contours sont incomplets et la mesure-F vaut 47,6 %. La meilleure mesure-F est de 55,1 %. Elle est atteinte au seuil $\xi = 50$. Les contours sont alors plus complets.

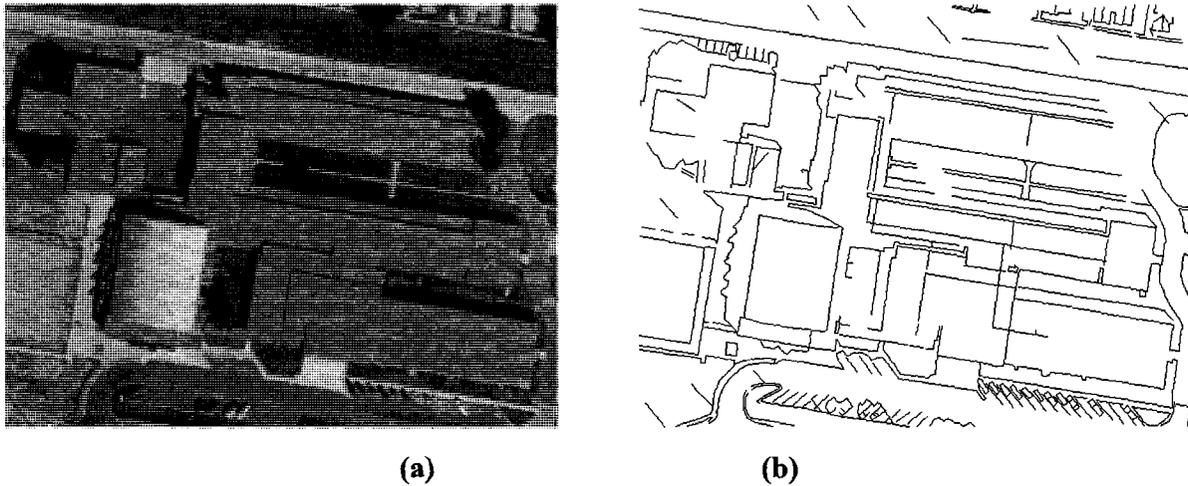


Figure A4.1 - (a) Image « école » du jeu de données aériennes de South Florida
(b) Vérité-terrain de contours

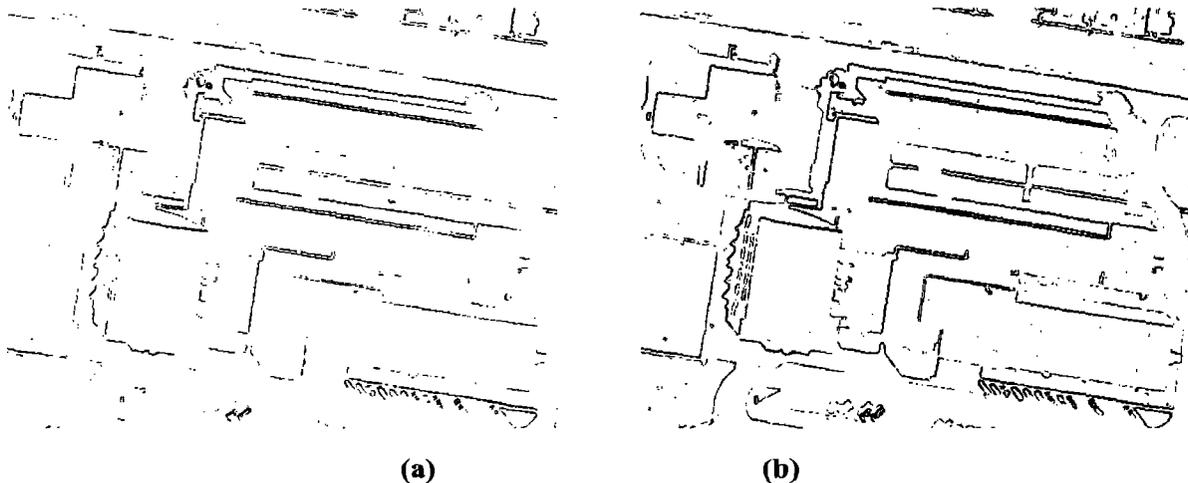
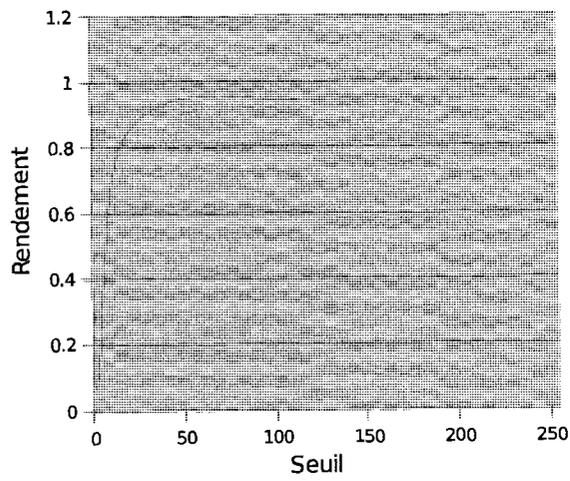
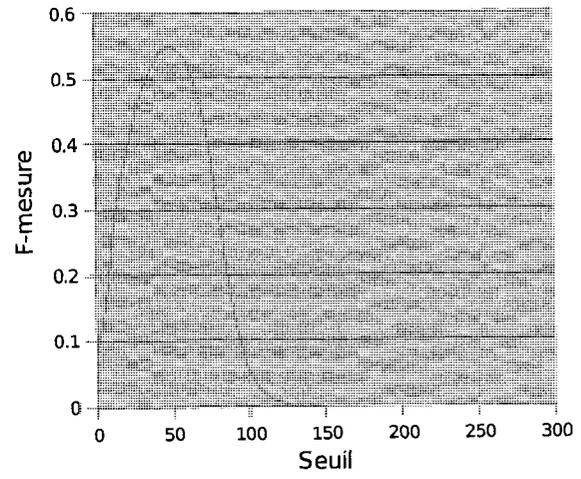


Figure A4.2 - (a) Contours de Kirsch maximisant le rendement
(b) Contours de Kirsch maximisant la mesure-F



(a)



(b)

**Figure A4.3 - (a) Rendement en fonction du seuil
(b) Mesure-F en fonction du seuil**

Annexe 5 - Utilisation du logiciel PerfAmel

Cette annexe explique comment utiliser le programme PerfAmel. Au démarrage, la première étape est d'ouvrir une image. Après ce choix, un dialogue vous demandera de sélectionner un algorithme de détection de contours à appliquer à l'image (Figure A5.1).

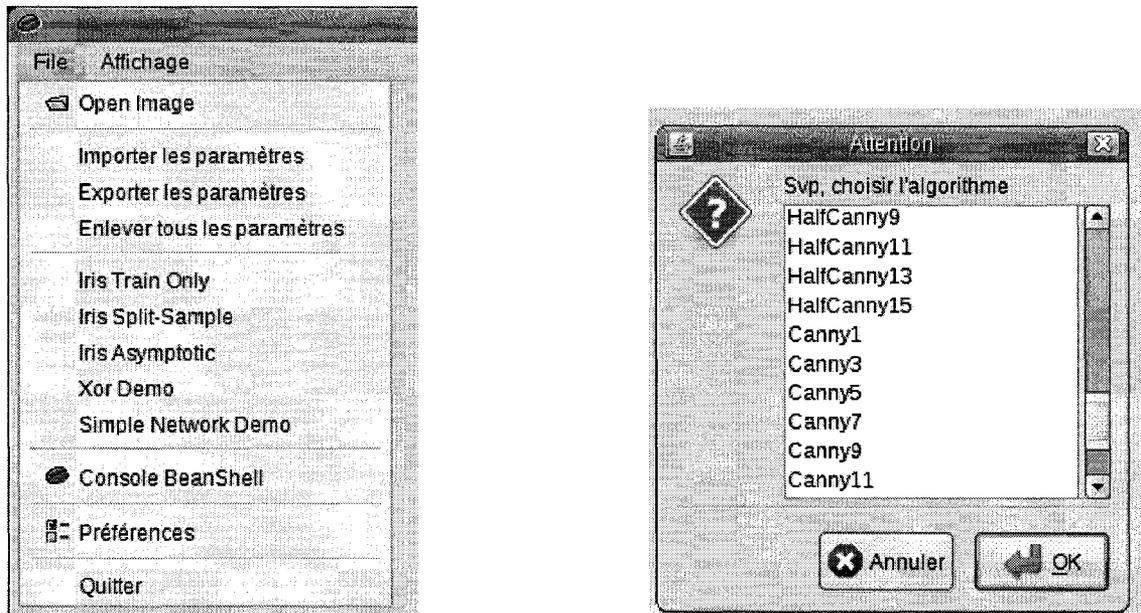


Figure A5.1 - Ouverture d'une image et choix d'un algorithme

Le logiciel calcule alors l'image d'amplitude de gradient de contours et la sauvegarde sous le nom `raw.tif` dans le répertoire de travail. Ensuite, le logiciel fait une recherche exhaustive pour déterminer les paramètres permettant d'optimiser les performances en fonction de la vérité-terrain associée à l'image (Figure A5.2). La meilleure carte de contours est alors affichée et sauvegardée sous le nom `bin.tif`.

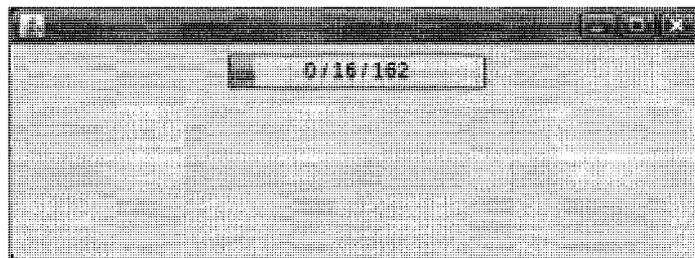


Figure A5.2 - Recherche exhaustive de paramètres

Pour l'étape de restauration, le logiciel commence par calculer les indicateurs qui serviront d'intrants au réseau de neurones. Ensuite, il vous demande alors si vous souhaitez entraîner un nouveau réseau ou réutiliser un réseau existant (Figure A5.3). Le logiciel procède ensuite automatiquement à l'entraînement ou au chargement d'un réseau. La carte d'indice de présence de contours est alors calculée et elle est multipliée avec la carte brute de l'algorithme classique. L'image obtenue est sauvegardée sous le nom `amel.tif`. Une recherche exhaustive est faite pour déterminer les meilleurs paramètres pour la restauration. Le résultat est affiché et sauvegardé sous le nom de `cleaned.tif`.

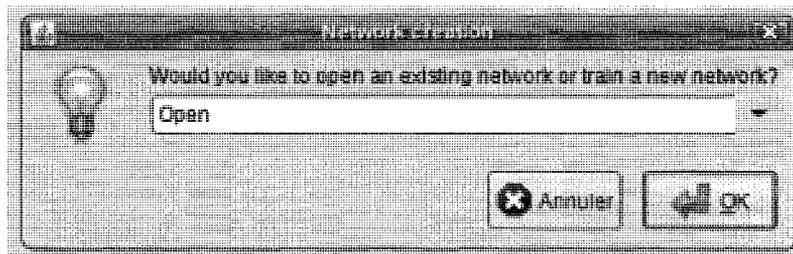


Figure A5.3 - Sélection d'un réseau de neurones