



**HAL**  
open science

# Détection de Contours et Diffusion Anisotropique dans les Images

Baptiste Magnier

► **To cite this version:**

Baptiste Magnier. Détection de Contours et Diffusion Anisotropique dans les Images. Traitement des images [eess.IV]. Université Montpellier II - Sciences et Techniques du Languedoc, 2011. Français. NNT: . tel-00780268

**HAL Id: tel-00780268**

**<https://theses.hal.science/tel-00780268>**

Submitted on 23 Jan 2013

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

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

UNIVERSITÉ MONTPELLIER II

N° attribué par la bibliothèque

--	--	--	--	--	--	--	--	--	--

## THÈSE

pour obtenir le grade de

**DOCTEUR de l'Université de thèse**

Spécialité : **Informatique**

préparée au laboratoire **LGi2P**

dans le cadre de l'École Doctorale **Information, Structure et Systèmes**

présentée et soutenue publiquement

par

**Baptiste Magnier**

le 12/12/2011

Titre:

**Détection de Contours et Diffusion Anisotropique  
dans les Images**

Directeur de thèse: **Philippe Montesinos**

Encadrant de proximité: **Daniel Diep**

Jury

M. Patrick Lambert,	Rapporteur
M. Frédéric Truchetet,	Rapporteur
M. William Puech,	Examineur
M. Khalifa Djemal,	Examineur
M. Philippe Montesinos,	Directeur de thèse

---

# Résumé

Ce mémoire de thèse est consacré à la détection anisotrope de contours et à la diffusion anisotropique dans les images qui utilisent des demi-filtres gaussiens. Ces demi-filtres parcourent toutes les orientations possibles autour d'un pixel avec un pas de discrétisation choisi. A partir de demi-filtres orientés de dérivées de gaussiennes anisotropes, une nouvelle approche de détection de bords dans les images couleurs a été développée. Puisque le filtre est allongé et fin, cette méthode permet d'extraire les contours des larges structures mais également des petits objets, même en présence de bruit. Les courbes gamma mettent en évidence des parties sombres ou sur-éclairées d'une image. L'utilisation combinée de ces demi-filtres et des courbes gamma apporte une robustesse supplémentaire au filtre puisqu'il est capable d'extraire les bords des objets, même dans les parties sous et sur-exposées d'une image. Le deuxième travail dans cette thèse concerne l'extraction de lignes de crêtes dans les images à partir de différence de deux demi-gaussiennes anisotropiques. Cette nouvelle méthode rend très précise la détection de ce type de contours même au niveau des lignes fortement courbées ou coudées. De plus, l'utilisation de demi-filtres apporte une information supplémentaire au niveau des jonctions qui correspondent aux maxima locaux. Enfin, le dernier thème de cette thèse est dédié à la diffusion anisotropique dans les images. Ainsi, trois nouveaux schémas de régularisation ont été développés dont deux pour supprimer les textures et un pour le débruitage. Ce dernier schéma donne des résultats intéressants, même dans des images en couleurs fortement bruitées.

**Mots clés :** Détection de contours, filtre anisotrope, gaussienne, filtre récursif, diffusion anisotropique.

# Abstract

This thesis is devoted to the anisotropic edge detection and the anisotropic diffusion in images. Based on Gaussian half filters, these half filters scan all orientations around a pixel with a defined step. From oriented derivatives of Gaussian half filters, an approach to edge detection in color images has been developed. Since the filter is long and thin, this method allows to extract anisotropically the contours of the large structures but also of small objects, even in the presence of noise. Gamma curves enhance dark or over-illuminated areas of an image. The combined use of these half filters and gamma curves provide an additional robustness to the method as it is able to extract edges of objects, even in areas under and over-exposed of an image. The second work in this thesis is concerned with the extraction of ridges in the images using the difference of two anisotropic half Gaussians. This new method is very accurate for roof edges even if they are strongly curved or bended. Moreover, the use of half filters provide additional information at level of the junctions which correspond to the local maxima. The final topic of this thesis is dedicated to the anisotropic diffusion in images. Thus, three new regularization schemes have been developed either for removing the textures or for denoising. This last scheme provides interesting results, even in highly noisy color images.

**Keywords :** Edge detection, anisotropic filter, gaussian, recursive filter, anisotropic diffusion.

# Remerciements

Dans un premier temps, je tiens à remercier Messieurs Frédéric Truchetet et Patrick Lambert d'avoir rapporter mon travail. Ces remerciements sont également dirigés vers Messieurs Khalifa Djemal et William Puech d'avoir assisté à ma soutenance.

Sans les conseils précieux de mes encadrants, je ne serais jamais venu à bout de cette thèse. C'est pourquoi ce paragraphe est consacré aux remerciements envers Philippe Montesinos pour tout ce qu'il a su m'expliquer et me guider, mais il n'est pas suffisant pour décrire ce qu'il a su apporter. En effet, ses conseils et sa disponibilité chaque jour m'ont été précieux, sa patience, sa pédagogie ainsi que sa motivation combinée à la mienne ont permis d'accomplir ce travail qui fut à la fois difficile et passionnant. Evidemment, Daniel Diep m'a apporté beaucoup avec un angle différent de celui de Philippe, sur des parties très pointues, donc je l'associe à ces remerciements.

Ce manuscrit a été relu par mes parents pour une correction orthographique, c'est pourquoi je les remercie d'avoir consacré du temps afin de m'aider pour cette tâche ingrate.

Il me tient particulièrement à cœur de remercier le LGi2P de m'avoir accueilli sans savoir si le changement de sujet de thèse allait fonctionner. Donc je remercie Michel Vasquez ainsi que Yannick Vimont, précédent directeur et actuel du laboratoire, qui m'ont encouragé depuis mon commencement avec Philippe. Cette thèse n'aurait pas abouti sans le travail des secrétaires du laboratoire nommées Claude Badiou et Valérie Roman. Pour différentes choses, je salue Isabelle Maurin au secrétariat. Je salue également tous les membres du laboratoire, tout particulièrement les thésards : Sihem Mallek, Sébastien Harispe, Afef Denguir, Benjamin Duthil, Mohameth-François Sy, Abdelhak Imoussaten et Mambaye Lo.

La personne qui m'a encouragé depuis le début, à qui je dois beaucoup est William Puech, je te remercie pour à la fois ton regard scientifique, ta motivation et ta disponibilité. Tu m'as toujours dit de ne pas lâcher, je n'ai rien lâché, merci pour tes conseils. J'espère que l'on ira voir d'autres matchs ensemble dans le futur et bonjour à toute la famille.

Au LIRMM, je salue tous les membres de l'équipe ICAR, mais tout particulièrement : Nicolas Tournier, Roseline Bénéière, Benoît Lange, Naveed Islam, Dalila Goudia, Loïc on The Wood, Mathieux Desoubieux, Azhar Hadmi, Alain Hassoun, Sarra Kouider, Auréline Quatrehomme, Methaq Gaata, Marc Chaumont et Olivier Strauss.

Je salue également les personnes rencontrées au LIRMM durant ma thèse : Bilal

Nehme, Kévin Loquin, Florence Jacquy, Khizar Hayat, Ashvin Sobee, Lotfi Chikh, Sébastien Krut, Milan Djilas, Bastien Durant, Antônio Padilha Lenari Bo, Fanny et Sébastien Le Floch, Chao Liu, Dingguo, Carla Aguiar, Sébastien Langagne, Olivier Leman, Mitsu, Nabil Zemiti et Jérémy Laforet Lala-lalala.

Dans des groupes de travail, workshop et conférences, j'ai rencontré différentes personnes que je salue par l'intermédiaire de ce paragraphe : Désiré Sidibé, David Hurych, Nan Luo, Camille Couprie, Cédric Demonceaux, Dimitar Doshkov, Eping Chen, Liansheng Wang, Jean-Charles Bazin, Baptiste Hemery, David Monnin, Lionel Bombrun et Simon Bozonnet.

D'autres personnes, qui n'ont aucun rapport avec ma thèse ont constitué une aide morale puisqu'il s'agit de mes amis d'enfance, d'adolescence et d'ailleurs. Donc je salue dans le désordre :

- ✓ Aude Zyg-Zyg, mon sac à patates, une partie de tennis comme en master ?
- ✓ Mathieu Martin, qui a su démarrer son scooter avec une perceuse,
- ✓ Christophe dit le Toulouzaing, à quand la prochaine descente de ski d'été ?
- ✓ Maxime Grenard dit Maxou, dommage qu'il n'y connaisse rien au foot,
- ✓ Nadir Maaroufi, j'espère discuter de tout ça prochainement,
- ✓ Pauline Monchy, à quand une bonne fête de la musique ?
- ✓ Marc-Antoine Souty, sa polo et son paquet de nouilles,
- ✓ Yohan Roisin, alors l'ASM ?
- ✓ Romain Duval, un concert ? un parc d'attractions ?
- ✓ Baptiste Arribe, tous ces délires à l'internat toujours présents dans nos têtes,
- ✓ Vincent Michel qui adore les sangliers,
- ✓ Séverine Delière, à qui je souhaite du bonheur et du courage,
- ✓ Simon et Charles Ingland, une virée dans les champs ou rechercher des fourmis ?
- ✓ Stéphanie Levailant, dit Ronnie, supportrice de VA maintenant ?
- ✓ David Languignon, à quand un ping pong ou une sortie en VTT ?
- ✓ Frédéric Larue, ancien colloc et désormais ami, allez les bleus !
- ✓ Guillaume Goubert, qui transporte une cuisinière sur son épaule : normal,
- ✓ Jérôme Le Borgne, dit Mamotte, se verra-t-on un jour sur Toulouse ?
- ✓ Maxime Chevailliot qui me doit encore une clef USB, et allez Lens !
- ✓ Laurent Dhieux, son poster et ses cheveux de Bob Marley,
- ✓ Cédric Chevailliot, SRFC's fan, pastonologue et parrain de notre petit boubou.

La fin des remerciements est consacrée à la partie *sentimentale*. J'embrasse très fort ma petite soeur Julie qui me manque beaucoup, avec une petite pensée pour David Verleye. Une grosse pensée également pour mes parents et mes grands parents. Je salue mes oncles et tantes, cousins et cousines.

Enfin, comment ne pas associer ce manuscrit avec mon petit boubou, Célian né le jour de la Saint Valentin, qui fut également un jour *dead line* et a assisté à la soutenance de son papa à l'âge de 10 mois. J'embrasse également sa petite maman, ma future femme Aurore Baudoux ♡.

# Table des matières

Résumé . . . . .	iii
Abstract . . . . .	iv
Remerciements . . . . .	v
Table des matières . . . . .	vii
<b>Introduction</b>	<b>1</b>
1 Contexte et motivations de la thèse . . . . .	1
2 Organisation de la thèse et contributions . . . . .	2
3 Remarques . . . . .	3
<b>1 Filtrage et Segmentation</b>	<b>5</b>
1 Introduction . . . . .	5
2 Quelques définitions . . . . .	5
3 Filtrage linéaire . . . . .	7
3.1 Filtrage spatial . . . . .	7
3.2 Fonctions de transfert en $z$ . . . . .	8
3.3 Transformée en $z$ inverse . . . . .	9
3.4 Les filtres à réponse impulsionnelle finie (RIF) . . . . .	9
3.5 Les filtres à réponse impulsionnelle infinie (RII) . . . . .	9
4 Opérateurs différentiels . . . . .	10
4.1 L'opérateur gradient . . . . .	10
4.2 L'opérateur Laplacien . . . . .	11
4.3 Tenseurs de gradient . . . . .	13
4.4 Tenseur de structure . . . . .	15
4.5 Tenseur de courbure . . . . .	16
5 Historique en détection de contours dans les images . . . . .	18
5.1 Introduction . . . . .	18
5.2 Les Masques . . . . .	21
5.3 Laplacien de gaussienne . . . . .	27
5.4 Différence de gaussienne . . . . .	29
6 Critères de Performances de Canny et Filtrage Optimal . . . . .	31
6.1 Les Approches Analytiques . . . . .	31
6.2 Le Filtre de Deriche . . . . .	34
6.3 Le Filtre Gaussien . . . . .	37
6.4 Le Filtre de Shen-Castan . . . . .	41
6.5 Extraction de lignes de crêtes . . . . .	42
7 Autres approches et filtres non linéaires . . . . .	46
7.1 Lissage et régularisation . . . . .	46

7.2	Détecteur de contours non linéaires . . . . .	47
8	Les Espaces Echelle et Equations Différentielles dans les Images . . . . .	50
8.1	Introduction . . . . .	50
8.2	Espace échelle linéaire . . . . .	51
8.3	Le modèle de Perona-Malik . . . . .	53
8.4	Axiomatisation de l'espace échelle . . . . .	56
8.5	Espaces échelle morphologiques . . . . .	57
8.6	Un lissage sélectif optimal . . . . .	59
8.7	Approches variationnelles . . . . .	61
8.8	Diffusion anisotrope d'images en couleurs . . . . .	63
8.9	Diffusion tensorielle . . . . .	65
9	Les Filtres Orientés . . . . .	69
9.1	Les filtres isotropes orientés . . . . .	69
9.2	Les filtres anisotropes orientés . . . . .	72
9.3	Les filtres de coins orientés . . . . .	74
9.4	Les filtres orientables de coin gaussien . . . . .	78
10	Conclusion . . . . .	78
<b>2</b>	<b>Extraction de Contours et Diffusion Anisotropique des Images</b>	<b>81</b>
1	Introduction . . . . .	81
2	Des demi-filtres orientés pour la détection de contours . . . . .	82
2.1	Introduction . . . . .	82
2.2	La dérivée d'un demi-filtre gaussien orienté . . . . .	83
2.3	Implémentation du filtre . . . . .	89
2.4	Performances du filtre anisotrope sur des contours virtuels . . . . .	91
2.5	Résultats sur des images réelles scalaires . . . . .	93
2.6	Adaptation du filtre à la couleur . . . . .	94
2.7	Résultats sur une image couleur . . . . .	97
2.8	Performance temporelle . . . . .	97
2.9	Détection anisotrope de contours utilisant les courbes gamma . . . . .	99
2.10	Résultats en utilisant les courbes gamma . . . . .	102
2.11	Evaluation du filtre de détection de bords . . . . .	105
2.12	Conclusion . . . . .	108
3	Différence de demi-gaussiennes anisotropes pour la détection de contours toits . . . . .	109
3.1	Introduction . . . . .	109
3.2	Un demi-filtre de lissage orienté . . . . .	110
3.3	Différence de deux demi-gaussiennes pour l'extraction de lignes de crêtes . . . . .	111
3.4	Résultats . . . . .	117
3.5	Evaluation de la Méthode . . . . .	120
3.6	Extraction de Courbes Perceptuelles . . . . .	122
3.7	Détection de Jonctions de Lignes de Crêtes . . . . .	126
3.8	Conclusion . . . . .	131
4	Régularisation des images par diffusion anisotropique . . . . .	133
4.1	Introduction . . . . .	133

4.2	Une détection de contours larges pour les principales structures de l'image . . . . .	135
4.3	Diffusion de l'image dans les deux directions correspondantes aux courbures du signal $s(\theta)$ . . . . .	138
4.4	Diffusion dans les deux directions des filtres de dérivation . . .	146
4.5	Diffusion dans les deux directions des filtres de dérivation avec contrôle de la diffusion . . . . .	151
4.6	Discussion . . . . .	166
4.7	Conclusion . . . . .	167
5	Conclusion . . . . .	168
<b>Conclusion Générale</b>		<b>171</b>
<b>3</b>	<b>ANNEXES</b>	<b>173</b>
1	Implémentation récursive des filtres de Deriche . . . . .	173
2	Approximation récursive du filtre gaussien et de ses dérivées . . . . .	174
3	Implémentation récursive des filtres de Shen-Castan . . . . .	177
4	Axiomatisation de l'analyse multi échelle . . . . .	177
5	Estimation de la pose d'une caméra omnidirectionnelle centrale . . .	180
<b>Bibliographie</b>		<b>185</b>



# Introduction

## 1 Contexte et motivations de la thèse

La segmentation des images occupe une place importante en traitement des images. Une image numérique est un objet représentant des objets ou un monde réel sous forme discrète, il faut donc analyser les images avec des opérateurs discrets. En général, les images sont capturées par des capteurs CMOS (*Complementary Metal Oxide Semiconductor*) ou CCD (*Charged-Coupled Device*). Aussi, des images peuvent être capturées par émission d'une onde et enregistrement de son écho (par exemple le laser, le radar, les ultrasons ou les rayons X). D'autres procédés mesurent une réaction particulière du milieu à capturer en fonction de l'excitation fournie (les plus connus sont l'Imagerie par Résonance Magnétique (IRM), la Tomographie par Emission de Positons (TEP) et la microscopie par balayage électronique). Toutes ces mesures, aussi différentes soient-elles produisent des effets indésirables. En effet, outre le fait d'échantillonner un monde continu vers un objet discret fini (une image rectangulaire la plupart du temps), les captures d'images introduisent de nombreux défauts, tels que des aberrations chromatiques, un manque de netteté, des artéfacts de couleurs, des effets de moiré, une forte présence de bruits statiques et dynamiques ou des déformations géométriques. Les défauts peuvent également intervenir au niveau des compressions des images ou vidéos. Tous ces problèmes rendent difficiles l'interprétation et l'analyse de scènes. L'analyse des images tente de décrire de façon quantitative ou qualitative les différents objets présents dans les données. Lorsqu'il s'agit d'images personnelles avec quelques défauts, il est assez aisé de modifier certains détails comme les yeux rouges ou une zone sous- ou surexposée de l'image en faisant varier les intensités des différentes couleurs. Cependant, dès que les images sont utilisées dans des applications telles que le domaine médical, l'aérien, l'astronomie, l'industriel, la surveillance, la restauration des images ou de films anciens, la segmentation a besoin d'être automatique ou semi-automatique. Par exemple, améliorer les images d'une vidéo une par une prend beaucoup de temps, il faut des mois voire des années pour débruiter un film.

Détecter les contours ou les régions permet de mieux analyser une scène capturée par une image. Les contours sont des éléments essentiels en analyse d'image, y compris pour la restauration. Il existe de nombreux algorithmes de détection de contours, seulement, plus l'image contient du bruit, moins les détecteurs de contours sont efficaces. Un détecteur de contours efficace permettrait de détecter les bords dans des régions perturbées par des défauts tout en restant précis au niveau du contour. Un contour est parfaitement détecté lorsque l'écart entre les frontières obtenues et les théoriques est nul. Les méthodes de détections de contours dans les images classiques ne sont pas adaptées lorsque le bruit dépasse une certaine fré-

quence ou intensité dans les images. La difficulté à détecter des contours dans des images perturbées par le bruit ou les textures a motivé ce travail. En effet, nous présenterons dans ce manuscrit une nouvelle approche de détection de bords permettant d'être robuste aux irrégularités de l'image tout en restant précis au niveau des contours. De plus, il existe plusieurs types de contours comme ceux de type marche, typiquement un changement brusque d'intensité lumineuse et les contours de type toit, comme une ligne blanche dans une image noire (ou inversement). Le problème est que les détecteurs de contours de type marche ne sont pas adaptés pour extraire ceux de type toit dans les images. Nous avons également développé une nouvelle méthode permettant d'extraire ce type de contours, même en présence de bruits.

Détecter les contours dans des images perturbées a également motivé un autre travail. Les filtres que nous avons développés pour la détection de bords et de lignes dans les images sont robustes aux bruits si bien qu'il est possible d'améliorer des images à l'aide de ces filtres. Afin de régulariser des images, les contours sont des éléments importants dans la préservation des principales structures donc plus le détecteur de bord est fiable, meilleure sera la restauration. Aussi, nous avons utilisé le détecteur de contours pour supprimer les textures et aussi restaurer les images par équation aux dérivées partielles.

## 2 Organisation de la thèse et contributions

Ce manuscrit est organisé en deux chapitres. Le premier est consacré à l'historique de travaux dans la littérature et le deuxième est centré sur les nouveaux opérateurs de détection de contours et de diffusion développés courant cette thèse.

Le chapitre 1 traite le filtrage des images, un historique de détection de contours, la régularisation des images par équations aux dérivées partielles ainsi que les filtres orientés. Nous soulignons les défauts des différentes approches tout en montrant de nombreux résultats.

Le chapitre 2 est divisé en trois parties majeures :

- Des demi-filtres gaussiens orientés utilisés en extraction de contours
- La différence de deux demi-filtres gaussiens orientés pour la détection de courbes
- La régularisation des images par diffusion anisotrope selon les directions obtenues par les demi-filtres

La première partie du second chapitre est consacrée à un nouveau filtre récursif pour la détection anisotrope de contours. La grande nouveauté est que le noyau utilisé se divise en deux parties, d'un côté, un demi-filtre estime le chemin entrant par un pixel et de l'autre, un deuxième demi-filtre estime le chemin sortant. Ces deux chemins sont en fait les directions d'un contour traversant un pixel. Nous avons également adapté ce filtre aux images vectorielles (i. e. les images en couleurs) puis nous avons comparé nos résultats avec de nombreuses approches ainsi que fourni des résultats quantitatifs. Ce travail a fait l'oeuvre d'une publication [MM10], une amélioration de cette méthode a été proposée afin de détecter les contours dans les zones sur ou sous-exposées de l'image et a également été publiée [MMD11a].

La seconde partie est dédiée à la détection de contours toits appelés lignes de crêtes. Afin de détecter les lignes de crêtes, les applications classiques estiment la courbure à la surface, ce qui rend en général des résultats perturbés par le bruit. Nous avons développé un nouveau filtre basé sur la différence de deux gaussiennes anisotropes qui permet d'extraire les lignes de crêtes même discontinues. Ce travail a été publié dans deux conférences internationales [MMD11c] [MDM]. Les résultats présentés ont été comparés avec d'autres méthodes et des résultats qualitatifs ont été apportés en fonction du niveau de bruit dans l'image. Aussi, ce filtre nous a permis d'extraire les jonctions de lignes de crêtes, ce qui a fait l'objet également d'une publication internationale [MMD11b].

Enfin, la troisième partie est consacrée à de nouveaux schémas de diffusion pour régulariser les images. Nous avons développé trois méthodes capable soit de supprimer la texture, soit de restaurer les images. Une première approche de diffusion de texture dans les images scalaires a été publiée dans une conférence internationale [MMD11d] puis l'adaptation aux images couleurs a fait également l'oeuvre d'une publication [MMD11e]. Un autre travail concernant la diffusion de textures en adaptant la méthode précédente avec le détecteur de contours développé auparavant a été développé et améliore la méthode précédente. Enfin, la dernière partie sur la restauration des images, qui contrôle la diffusion au niveau des coins et des contours contrairement aux méthodes de diffusion pour la suppression de textures. Cette approche a été valorisé dans une conférence internationale [MMD12a]. Nous avons toujours comparé nos différents résultats avec des méthodes existantes puis fourni des résultats quantitatifs et qualitatifs.

### 3 Remarques

Ce manuscrit rapporte des travaux développés pendant un an et demi puisque j'ai changé de sujet de thèse en 2010 et également de laboratoire. Pendant ce peu de temps, de nombreux thèmes ont été abordés et plusieurs travaux ont été développés, mais certaines études méritent d'être terminées. En conclusion de chaque partie, nous précisons d'éventuelles améliorations ou applications.

Mon premier sujet de thèse concernait la vision omnidirectionnelle. Mon travail consistait à déterminer une nouvelle méthode pour estimer l'inclinaison d'une caméra catadioptrique centrée par rapport au sol. Mes travaux ont débouché sur une publication dans une conférence internationale [MCS<sup>+</sup>10], il est présenté en annexe 5.



# Chapitre 1

## Filtrage et Segmentation

### 1 Introduction

Ce premier chapitre est consacré à l'état de l'art en filtrage des images. Les travaux présentés dans ce manuscrit s'inspirent de nombreuses méthodes précédemment développées. Nous avons également comparé nos approches avec celles de la littérature présentées dans ce présent chapitre.

Dans un premier temps nous introduirons dans la partie 2 quelques définitions qui permettront de mieux comprendre la suite de ce manuscrit. Puis, la partie 3 est consacrée à quelques propriétés sur le filtrage linéaire. Dans les parties suivantes, nous présenterons des opérateurs de gradient, laplacien et les notions de tenseur en partie 4. Les travaux présentés et détaillés dans les parties qui suivent sont nécessaires pour la compréhension et la situation de nos contributions. En effet, la partie 5 est dédiée à un historique en détection de contours puis à une détection de contours théorique optimale en partie 6. S'en suit une section dédiée à quelques détecteurs de contours non linéaires en partie 7. En partie 8, nous développerons les espaces échelle et les équations différentielles adaptées aux images dans la partie suivante. Enfin, la partie 9 de ce chapitre est dédiée aux filtres orientés qui introduira la majeure partie de nos travaux basés sur des filtres similaires.

La plupart de ces approches sont accompagnées immédiatement de résultats ou dans le chapitre suivant puisque nous comparons nos méthodes avec d'autres de la littérature.

### 2 Quelques définitions

#### Image numérique

Une image numérique est composée d'un tableau de valeurs appelées pixels. Le pixel est de forme carrée ou rectangulaire et vient des mots anglais *picture element*. Afin de créer une image, un capteur enregistre la lumière réfléchie par les objets observés et la transforme en tension électrique. Le résultat final est une matrice affectant à chaque pixel une valeur représentant une information lumineuse ne prenant qu'un nombre discret de valeurs.

Une image  $I$  est dite scalaire si elle ne comporte qu'un seul tableau de valeurs dont l'intensité est représentée par  $I(i_1, i_2)$  (image en niveaux de gris). Les différents

$i_{j, j \in \{1,2\}}$  correspondent aux coordonnées des pixels dans l'image. On représente par  $\Omega \subset \mathbb{R}^2$  le domaine spatial de l'image  $I$  de dimension 2. Si sa taille est  $n_1 \times n_2$ ,  $(n_1, n_2) \in \mathbb{N}^2$ , en posant  $N_{n_i} = \{0, 1, \dots, n_i - 2, n_i - 1\}$ , une image scalaire  $I$  peut être alors interprétée comme une fonction définie sur la grille  $N_1 \times N_2$  :

$$\begin{aligned} I : N_{n_1} \times N_{n_2} \subset \Omega &\rightarrow \mathbb{R} \\ (i_1, i_2) &\mapsto I(i_1, i_2) \end{aligned} \quad (1.1)$$

Une image vectorielle  $\mathbf{I}$  est composée de plusieurs images scalaires sur un domaine discret  $\Omega \subset \mathbb{R}^P$  de dimension  $P$  dont les valeurs se situent dans  $\mathbb{R}^d$ . Sa définition générale est donnée par :

$$\begin{aligned} \mathbf{I} : N_{n_1} \times N_{n_2} \times \dots \times N_{n_P} \subset \mathbb{R}^P &\rightarrow \mathbb{R}^d \\ (i_1, i_2, \dots, i_P) &\mapsto \mathbf{I}(i_1, i_2, \dots, i_P) \end{aligned} \quad (1.2)$$

$M$  représente le nombre de composantes d'une image vectorielle.

Dans la suite de ce manuscrit, nous travaillons exclusivement sur des images bi-dimensionnelles, nous noterons donc par  $I(x, y)$  l'intensité du pixel de coordonnées  $(x, y)$  de l'image  $I$ .

### Filtrage isotrope

Un filtre est dit isotrope lorsqu'il présente les mêmes propriétés dans toutes les directions.

### Filtrage anisotrope

Par opposition au filtrage isotrope, un filtre anisotrope possède des propriétés variant en fonction de la direction.

### Filtre linéaire

Si  $f$  est un filtre linéaire adapté aux images alors le même filtre est appliqué sur tous les pixels de l'image et :

$$(\lambda \cdot I + J) * f = \lambda \cdot I * f + J * f$$

où  $I$  et  $J$  sont deux images et  $\lambda \in \mathbb{R}$ , la linéarité se traduit par l'opération de convolution.

### Filtre causal et stable

Un filtre  $h$  est dit causal si et seulement si la réponse impulsionnelle est causale. Si  $h(n) = 0, \forall n < 0$ , une telle réponse impulsionnelle  $h$  est dite causale. Un filtre est stable si pour toute entrée  $x$  bornée, la sortie  $y$  est elle aussi bornée.

## Filtre séparable

Le filtre  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  est séparable si  $f(x, y) = f_a(x) \cdot f_b(y)$  avec  $f_a : \mathbb{R} \rightarrow \mathbb{R}$  et  $f_b : \mathbb{R} \rightarrow \mathbb{R}$ .

Les filtres bi-dimensionnels adaptés aux images sont dits séparables s'il est possible de les décomposer en deux filtres mono-dimensionnels appliqués successivement en horizontal puis en vertical (ou inversement).

## Seuillage dans les Images

Dans de nombreuses méthodes, afin d'extraire les contours ou les régions par exemple, l'étape finale consiste à seuiller le résultat des contours accentués. Il existe principalement deux types de seuillages, le seuillage simple ou le seuillage par hystérésis, couramment utilisés en détection de contours.

### Seuillage simple

Par seuillage simple, le but est de ne conserver que les pixels d'une image  $I$  dont l'intensité est au dessus d'un certain seuil  $\tau$ , ce qui crée une image binaire  $B$  :

$$B(x, y) = \begin{cases} 1 & \text{si } I(x, y) > \tau \\ 0 & \text{sinon} \end{cases}$$

où  $(x, y)$  représentent les coordonnées des pixels.

### Seuillage par Hystérésis

Le seuillage par hystérésis est un outil très utilisé en détection de contours. En effet, soit  $I$  une image contenant des pixels connexes en niveaux de gris (pour la détection de bords,  $I$  est souvent l'image des maxima locaux dans la direction du gradient). Dans le cas de pixels à valeur positives, en posant  $\tau_b$  et  $\tau_h$  respectivement les seuils bas et haut, une zone de pixels connexes sera détectée si la valeur d'au moins un de ces pixels est supérieur à  $\tau_h$  et tous les autres sont supérieurs à  $\tau_b$ . Notons que le seuillage par hystérésis peut s'appliquer pour des valeurs de pixels négatives à condition d'utiliser des seuils négatifs également.

## 3 Filtrage linéaire

### 3.1 Filtrage spatial

Filtrer signifie convoluer une image  $I(x, y)$  ( $(x, y) \in \mathbb{N} \times \mathbb{N}$ ) avec une fonction  $h(x, y)$  appelée réponse impulsionnelle (ou opérateur de convolution) du filtre. Ce filtrage correspond souvent à un lissage afin de régulariser l'image pour éviter des fausses détections de contours dues au bruit. Dans le cas continu, l'image filtrée est donnée par :

$$I_h(x, y) = h(x, y) * I(x, y). \tag{1.3}$$

Soit, dans le cas continu :

$$I_h(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h(u, v) \cdot I(x - u, y - v) \, dudv, \quad (1.4)$$

par commutativité, on obtient :

$$I_h(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h(x - u, y - v) \cdot I(u, v) \, dudv. \quad (1.5)$$

La dérivée d'un signal  $f$  convolué avec un filtre  $h$  est soit la dérivée du signal lui-même convolué avec le filtre, soit le filtre convolué avec la dérivée du signal :

$$(f * g)' = f' * g = f * g'. \quad (1.6)$$

Cette propriété signifie qu'il est possible de régulariser l'image en même temps que de la dériver à la condition que le filtre soit dérivable.

En posant  $\nabla I_h(x, y)$  gradient d'une image filtrée par  $h$ , il est calculé par :

$$\nabla I_h(x, y) = \nabla(I(x, y) * h(x, y)) \quad (1.7)$$

$$= \nabla I(x, y) * h(x, y) \quad (1.8)$$

$$= I(x, y) * \nabla h(x, y). \quad (1.9)$$

### 3.2 Fonctions de transfert en $z$

En traitement numérique du signal, la transformée en  $z$  permet de représenter un signal  $y$  possédant une infinité d'échantillons par une fonction complexe. Pour une suite  $y(n)$  donnée, elle s'écrit :

$$\mathcal{TZ}\{y(n)\} = \sum_{n=-\infty}^{+\infty} y(n) \cdot z^{-n}, \quad (1.10)$$

avec  $z \in \mathbb{C}$  tel que la série  $\mathcal{TZ}\{y(n)\}$  possède un domaine de convergence qui en général est un anneau centré sur l'origine de rayons  $R_1$  et  $R_2$  :  $R_1 < |z| < R_2$ , avec

$$R_1 = \lim_{k \rightarrow +\infty} |y(k)|^{1/k} \quad \text{et} \quad R_2 = \lim_{k \rightarrow +\infty} |y(-k)|^{-1/k}$$

La reconstruction du signal s'obtient par la transformation en  $z$  inverse.

Soit un filtre possédant un signal d'entrée  $x(n)$  et un signal de sortie  $y(n)$ , la fonction de transfert du filtre est définie comme le rapport des transformées en  $z$ . La fonction de transfert permet l'étude des propriétés asymptotiques et de stabilité du filtre.

### 3.3 Transformée en $z$ inverse

La transformée en  $z$  inverse est donnée par :

$$y(n) = \mathcal{TZ}^{-1}\{\mathcal{TZ}\{y(n)\}\} = \frac{1}{2\pi j} \oint_C \mathcal{TZ}\{y(n)\} \cdot z^{n-1} dz \quad (1.11)$$

où  $C$  représente un chemin fermé parcouru dans le sens anti-horaire et appartenant entièrement au domaine de convergence.

En pratique, la transformée en  $z$  inverse est peu utilisée mais on utilise souvent la propriété suivante :

$$\mathcal{TZ}^{-1}\{z^k \cdot \mathcal{TZ}\{y(n)\}\} = y(n+k). \quad (1.12)$$

### 3.4 Les filtres à réponse impulsionnelle finie (RIF)

Un filtre causal à réponse impulsionnelle finie de longueur  $N$  est un filtre numérique caractérisé par une réponse obtenue par convolution de la réponse impulsionnelle avec les valeurs du signal d'entrée. La réponse impulsionnelle d'un filtre RIF est stable et de durée finie qui dépend du nombre de coefficients du filtre.

Le RIF est décrit par l'équation suivante où  $x$  représente les valeurs du signal d'entrée et  $y$  les valeurs du signal de sortie :

$$y[n] = \sum_{k=0}^N b_k \cdot x[n-k]. \quad (1.13)$$

La réponse impulsionnelle de ce filtre est de longueur finie puisqu'elle est constituée de la suite des coefficients  $b_{i, i \in \{0, \dots, N\}}$ . Une propriété intéressante des filtres RIF est d'apporter un retard constant au signal lorsque les coefficients sont symétriques. Dans ce cas précis, le filtre est appelé filtre à phase linéaire.

Le filtre RIF est l'un des deux types de filtre numérique linéaire avec le filtre à réponse impulsionnelle infinie (filtre RII).

### 3.5 Les filtres à réponse impulsionnelle infinie (RII)

Un filtre causal à réponse impulsionnelle infinie correspond à une structure qui boucle sur elle-même se caractérisant par une réponse basée sur les valeurs du signal d'entrée ainsi que les valeurs antérieures de cette même réponse.

Le filtre RII est décrit par l'équation suivante où  $x$  représente les valeurs du signal d'entrée et  $y$  les valeurs du signal de sortie :

$$y[n] = \sum_{k=0}^{m-1} b_k \cdot x[n-k] - \sum_{k=1}^n a_k \cdot y[n-k]. \quad (1.14)$$

La réponse impulsionnelle de ce filtre est de longueur infinie.

Le problème d'implémentation d'un filtre récursif est de déterminer les coefficients réels  $a_k$  et  $b_k$  de la fonction de transfert caractérisant l'équation 1.14 et qui sont de la forme :

$$\mathcal{TZ}\{y(n)\} = \frac{\sum_{k=0}^{m-1} b_k \cdot z^{k-1}}{1 + \sum_{k=1}^n a_k \cdot z^{-k}} \quad (1.15)$$

La structure récursive permet, avec un faible nombre de coefficients, de réaliser des fonctions de transfert performantes avec un gain de temps de calcul. Cependant, la fonction de transfert d'un filtre RII est très sensible aux erreurs sur les coefficients et cela d'autant plus que son degré est élevé. D'autre part, elle comporte un risque d'instabilité, la stabilité sera garantie si les pôles de la fonction de transfert (équation 1.15) sont tous inférieurs strictement à 1.

## 4 Opérateurs différentiels

### 4.1 L'opérateur gradient

En considérant l'image dans un repère orthogonal (Oxy) tel que (Ox) désigne l'axe horizontal et (Oy) l'axe vertical, le gradient en un pixel  $(x, y)$  est un vecteur en deux dimensions caractérisé par son amplitude  $\nabla I(x, y)$  et sa direction  $\eta(x, y)$ . L'amplitude est directement liée à la quantité de variation locale des niveaux de gris et correspond à la pente de la surface image dans la direction de la plus forte pente comme schématisé en figure 1.1 (a). La direction du gradient est orthogonale à la direction du contour du point considéré. La méthode la plus simple pour estimer un gradient  $G_\theta(x, y)$  dans la direction  $\theta$  choisie est de calculer la variation monodimensionnelle en un point  $(x, y)$  de l'image :

$$G_\theta(x, y) = (I * D_\theta)(x, y) \quad (1.16)$$

où  $D_\theta$  représente l'opérateur de dérivation dans la direction  $\theta$ .

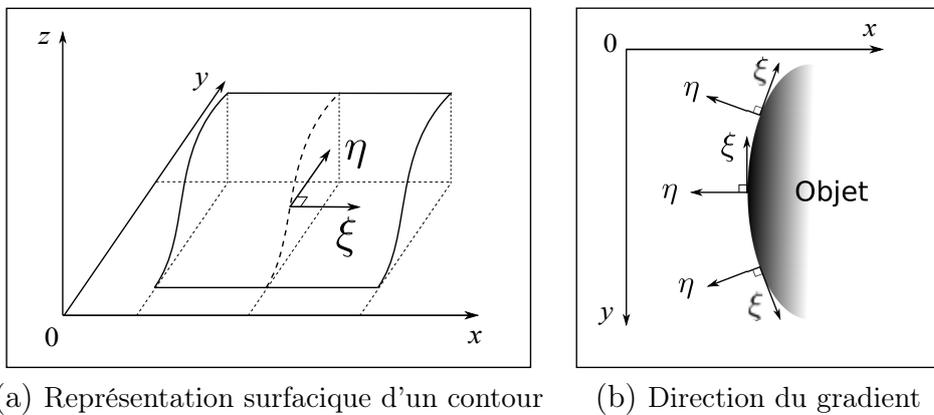


FIGURE 1.1 – Schéma d'un contour dans une image. (a)  $\eta$  représente la pente de la surface image dans la direction de la plus forte pente. L'axe  $z$  représente l'intensité lumineuse dans l'image. (b) Directions  $\eta$  et  $\xi$  en plusieurs points de contours.

La plupart des méthodes calculant le gradient appliquent l'équation 1.16 dans les deux directions horizontale et verticale de l'image :

$$I_x(x, y) = \frac{\partial I}{\partial x}(x, y), \quad (1.17)$$

$$I_y(x, y) = \frac{\partial I}{\partial y}(x, y). \quad (1.18)$$

Donc le gradient est donné par :

$$\nabla I = (I_x(x, y), I_y(x, y))^t. \quad (1.19)$$

Son module  $\|\nabla I(x, y)\|$  correspond à la plus forte pente de la surface image, il permet de quantifier l'importance du contour mis en évidence, c'est-à-dire l'amplitude du saut d'intensité relevé dans l'image est calculé par :

$$\|\nabla I(x, y)\| = \sqrt{(I_x^2(x, y) + I_y^2(x, y))}. \quad (1.20)$$

Sa direction appelée  $\eta(x, y)$  correspondant à la direction la plus forte dans l'image permet de déterminer l'arête présente dans l'image.

$$\eta(x, y) = \arctan \frac{I_y(x, y)}{I_x(x, y)}. \quad (1.21)$$

En effet, la direction du gradient  $\eta(x, y)$  est orthogonale à celle du contour appelée  $\xi(x, y)$  et les contours sont extraits en calculant les maxima locaux dans la direction  $\eta$ . La figure 1.1 (b) schématise les deux directions  $\eta$  et  $\xi$  par rapport à des points de contours.

## 4.2 L'opérateur Laplacien

En détection de bords, les méthodes de second ordre utilisent les passages par zéro du laplacien (figure 1.2) de l'image  $\Delta I$  qui correspondent aux contours :

$$\Delta I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}, \quad (1.22)$$

qui peut également être noté :

$$\Delta I = I_{xx} + I_{yy}, \quad (1.23)$$

avec  $I_{xx} = \frac{\partial^2 I}{\partial x^2}$  et  $I_{yy} = \frac{\partial^2 I}{\partial y^2}$ . Le laplacien se décompose aussi de la manière suivante :

$$\Delta I = I_{\xi\xi} + I_{\eta\eta}, \quad (1.24)$$

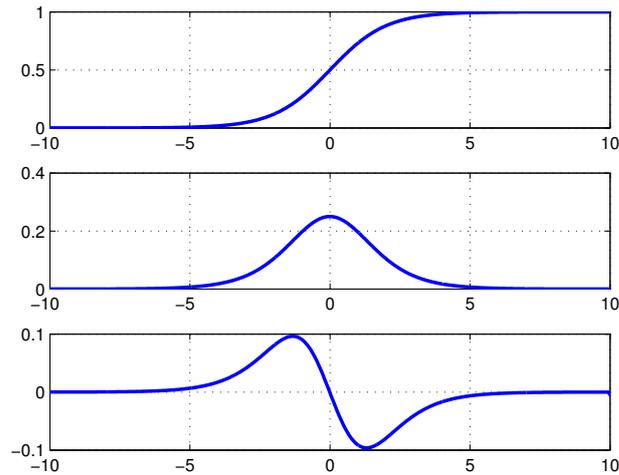


FIGURE 1.2 – Détection de contours en une dimension. De haut en bas : un signal comportant un contour, sa dérivée première où la maximum correspond au contour, sa dérivée seconde où le passage par zéro correspond également au contour.

il apparait un terme de courbure  $I_{\xi\xi} = \text{div}(\nabla I) = K \cdot \|\nabla I\|$  qui perturbe la détection de contours.  $K$  est appelé la courbure des isophotes<sup>1</sup> [KR82].  $I_{\xi\xi}$  et  $I_{\eta\eta}$  sont définis tels que :

$$\begin{cases} I_{\eta\eta} = \frac{I_{xx} \cdot I_x^2 + 2 \cdot I_x \cdot I_y \cdot I_{xy} + I_{yy} \cdot I_y^2}{I_x^2 + I_y^2} \\ I_{\xi\xi} = \frac{I_{xx} \cdot I_y^2 - 2 \cdot I_x \cdot I_y \cdot I_{xy} + I_{yy} \cdot I_x^2}{I_x^2 + I_y^2} \end{cases} \quad (1.25)$$

Dans le cas discret comme une image, l'opérateur  $\frac{\partial^2}{\partial x^2}$  peut être approximé par :

$$\Delta_x^2 f = \frac{\partial^2 f}{\partial x^2} = \Delta_x \Delta_x f = f(i+1, j) - 2f(i, j) + f(i-1, j). \quad (1.26)$$

Une approximation du laplacien peut donc être représentée par les masques  $M_1$  ou  $M_2$  présentés en figure 1.3, mais il en existe d'autres.

Cependant l'estimation de la dérivée seconde de l'image est très sensible aux bruits puisqu'une très petite variation de l'image conduit à des petits gradients localement maxima. Ces maxima locaux du gradient conduisent donc à des passages par zéro du laplacien, d'où un très grand nombre de détections de contours.

### Extraction des points de contours en utilisant le laplacien

Dans le cas d'une approche dérivée seconde, on dispose donc de la valeur du Laplacien en chaque pixels de coordonnées  $(x, y)$  de l'image originale ou préalablement lissée  $I$ . Si l'extraction de contours par le calcul du Laplacien était directe,

1. Une isophote ou ligne d'iso-intensité est une ligne d'intensité lumineuse constante tracée sur la surface image.

0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1
Masque $M_1$			Masque $M_2$		

FIGURE 1.3 – Masques d’approximation du laplacien.  $M_1$  nécessite un coefficient de  $\frac{1}{4}$  et  $M_2$  un coefficient de  $\frac{1}{8}$ .

il suffirait de sélectionner les pixels  $P$  de coordonnées  $(x, y)$  tels que  $\Delta I(x, y) = 0$ . Cependant, comme l’image de détection de contours en utilisant l’approximation du Laplacien est relativement bruitée, les pixels détectés sont ceux où le laplacien change de signe. La dernière étape consiste à seuiller l’image ainsi obtenue afin d’éliminer les points de trop faible gradient. Concrètement, l’extraction de ces passages par zéro s’effectue en trois étapes :

1. détermination d’une image de polarité  $I_p$  :

$$\begin{cases} I_p(x, y) = 0 & \text{si } \Delta I(x, y) > 0 \\ I_p(x, y) = 1 & \text{si } \Delta I(x, y) \leq 0 \end{cases}$$

2. calcul d’une image binaire  $I_z$  correspondant à la détection des passages par zéro dans  $I_p$  :

$$\begin{cases} I_z(x, y) = 1 & \text{si } (x, y) \text{ correspond soit à une transition 0-1 ou 1-0 dans } I_p \\ I_z(x, y) = 0 & \text{sinon} \end{cases}$$

3. L’image  $I_z$  est multipliée par l’image de maximisation du gradient dans la direction du gradient afin d’obtenir des contours de largeur un pixel. Enfin, les contours sont extraits par un seuillage simple ou par hystérésis.

### 4.3 Tenseurs de gradient

En analyse vectorielle et en géométrie différentielle, les tenseurs sont une généralisation multidimensionnelle de l’algèbre linéaire permettant de généraliser les scalaires et les vecteurs.

#### Tenseur en niveaux de gris

Le tenseur appelé monospectral  $\mathbf{M}(x, y)$  est un opérateur du champ d’orientation d’une image en niveaux de gris à partir des dérivées de l’image en  $X$  et en  $Y$  respectivement  $I_x(x, y)$  et  $I_y(x, y)$  en tout pixel de coordonnées  $(x, y)$  :

$$\mathbf{M}(x, y) = \begin{pmatrix} I_x^2(x, y) & I_x(x, y)I_y(x, y) \\ I_x(x, y)I_y(x, y) & I_y^2(x, y) \end{pmatrix}. \quad (1.27)$$

Cette matrice est de rang un, donc elle possède une valeur propre nulle. En posant,  $\lambda_{max}(x, y)$  et  $\lambda_{min}(x, y)$  respectivement les deux valeurs propres maximales et minimales de  $\mathbf{M}(x, y)$  :

$$\lambda_{max}(x, y) = I_x^2(x, y) + I_y^2(x, y) \quad (1.28)$$

$$\lambda_{min}(x, y) = 0, \quad (1.29)$$

donc (voir aussi partie 4.1 de ce chapitre)

$$\lambda_{max}(x, y) = \|\nabla I(x, y)\|^2. \quad (1.30)$$

Le tenseur monospectral permet d'estimer le gradient d'une image, certaines EDP (voir partie 8 de ce chapitre) sont basées sur ce tenseur [Nag83].

### Tenseur multispectral

Etant donné une image  $\mathbf{I}$  composée d'un nombre  $d$  de canaux. Souvent,  $\mathbf{I}$  est une image en couleurs. On note par  $I^i$  la  $i^{\text{ème}}$  composante de  $\mathbf{I}$  ( $1 \leq i \leq d$ ). Dans le cas des images classiques composées d'un canal rouge, vert et bleu ( $d = 3$ ),

$$\mathbf{I}(x, y) = \begin{pmatrix} I^1(x, y) = R(x, y) \\ I^2(x, y) = V(x, y) \\ I^3(x, y) = B(x, y) \end{pmatrix}$$

où  $R, V$  and  $B$  sont les canaux de l'image (Rouge, Vert, Bleu).

En détection de contours, prendre en compte les différents canaux de couleur en filtrant les différents canaux puis sommer les résultats peut mener à des résultats indésirables. En effet, la somme des directions peut s'annuler, ce qui amène à la perte des contours. Di Zenzo a proposé dans [DZ86] d'estimer, par une approche aux moindres carrés, le gradient le plus représentatif sur des images composées de plusieurs canaux. Cette méthode consiste à combiner les gradients directionnels pour les intégrer dans un tenseur multispectral symétrique  $\mathbf{T}$  :

$$\mathbf{T} = \begin{pmatrix} \mathbf{I}_x^2 & \mathbf{I}_x \mathbf{I}_y \\ \mathbf{I}_x \mathbf{I}_y & \mathbf{I}_y^2 \end{pmatrix} \quad (1.31)$$

$$= \begin{pmatrix} \left( \sum_{i=1}^d I_x^i \cdot I_x^i \right) & \left( \sum_{i=1}^d I_x^i \cdot I_y^i \right) \\ \left( \sum_{i=1}^d I_x^i \cdot I_y^i \right) & \left( \sum_{i=1}^d I_y^i \cdot I_y^i \right) \end{pmatrix} \quad (1.32)$$

où  $\mathbf{I}_x$  et  $\mathbf{I}_y$  sont des approximations des gradients horizontaux et verticaux. Les images  $\mathbf{I}_x$  et  $\mathbf{I}_y$  peuvent être calculées à l'aide des filtres de Deriche, de Shen-Castan, de la dérivée d'une gaussienne ou encore par les masques de convolution que nous allons détailler dans les parties suivantes de ce chapitre. Pour les images couleur composées des canaux rouge, vert et bleu, le tenseur multispectral couleur revient à :

$$\mathbf{T} = \begin{pmatrix} (R_x^2 + V_x^2 + B_x^2) & (R_x R_y + V_x V_y + B_x B_y) \\ (R_x R_y + V_x V_y + B_x B_y) & (R_y^2 + V_y^2 + B_y^2) \end{pmatrix}. \quad (1.33)$$

Le tenseur de multispectral décrit la variation de l'image dans une direction autour du point considéré. Di Zenzo estime par l'équation suivante l'orientation dominante, qui est égale à l'orientation dans l'image avec le maximum de changement de couleur :

$$\eta = \frac{1}{2} \arctan \left( \frac{2 \cdot \mathbf{I}_x \mathbf{I}_y}{\mathbf{I}_x^2 - \mathbf{I}_y^2} \right). \quad (1.34)$$

Cependant, les valeurs propres du tenseur permettent de déterminer les pentes de la surface image par maximisation de la somme des carrés pour chaque pixel de l'image, elles sont données par :

$$\lambda_1 = \frac{1}{2} \left( \mathbf{I}_x^2 + \mathbf{I}_y^2 + \sqrt{(\mathbf{I}_x^2 - \mathbf{I}_y^2)^2 + (2 \cdot \mathbf{I}_x \mathbf{I}_y)^2} \right) \quad (1.35)$$

$$\lambda_2 = \frac{1}{2} \left( \mathbf{I}_x^2 + \mathbf{I}_y^2 - \sqrt{(\mathbf{I}_x^2 - \mathbf{I}_y^2)^2 + (2 \cdot \mathbf{I}_x \mathbf{I}_y)^2} \right). \quad (1.36)$$

En effet, le vecteur propre associé à la valeur propre  $\lambda_1$  indique également l'orientation dominante souvent notée  $\eta$  et le vecteur propre associé à  $\lambda_2$  donne l'orientation de la pente au sens des moindres carrés. Cette orientation est perpendiculaire à celle prédominante, soit l'orientation tangente au contour notée  $\xi$ . Aussi, différentes combinaisons des valeurs propres  $\lambda_{i,i \in \{1,2\}}$  sont utilisées dans les équations de diffusions. Ainsi, la norme de cohérence  $\sqrt{\lambda_1 - \lambda_2}$  est utilisée dans [SR96] et [Wei]. Dans [TD02], l'auteur propose l'utilisation de l'extension de la norme du gradient  $\sqrt{\lambda_1}$ . Enfin, la somme  $\lambda_1 + \lambda_2$ , qui permet d'extraire les contours, décrit la variation totale et

$$\sqrt{\lambda_1 + \lambda_2} = \sqrt{\text{trace}(\mathbf{T})},$$

les auteurs de [BC98] l'incorporent dans leur schéma de diffusion. Le cas  $d = 1$  revient au tenseur monospectral.

#### 4.4 Tenseur de structure

La représentation du tenseur de structure est une matrice  $J_\rho(\nabla \mathbf{I}_\sigma)$  d'autocorrélation représentant la variation locale d'une image  $\mathbf{I}$  composée d'un nombre  $d$  de canaux en un point de coordonnées  $(x, y)$ . La mesure d'autocorrélation est estimée à partir des dérivées premières de  $\mathbf{I}$  (calculées sur un support gaussien  $G_\sigma$  d'écart type  $\sigma$ ) appelées :

$$\mathbf{I}_{\sigma,x} = \frac{\partial(G_\sigma * \mathbf{I})}{\partial x} \quad \text{et} \quad \mathbf{I}_{\sigma,y} = \frac{\partial(G_\sigma * \mathbf{I})}{\partial y}.$$

$J_\rho(\nabla \mathbf{I}_\sigma)$  est ensuite obtenu par :

$$J_\rho(\nabla \mathbf{I}_\sigma) = \begin{pmatrix} G_\rho * \mathbf{I}_{\sigma,x}^2 & G_\rho * \mathbf{I}_{\sigma,x} \mathbf{I}_{\sigma,y} \\ G_\rho * \mathbf{I}_{\sigma,x} \mathbf{I}_{\sigma,y} & G_\rho * \mathbf{I}_{\sigma,y}^2 \end{pmatrix} \quad (1.37)$$

$$= \begin{pmatrix} \left( G_\rho * \sum_{i=1}^d I_{\sigma,x}^i \cdot I_{\sigma,x}^i \right) & \left( G_\rho * \sum_{i=1}^d I_{\sigma,x}^i \cdot I_{\sigma,y}^i \right) \\ \left( G_\rho * \sum_{i=1}^d I_{\sigma,x}^i \cdot I_{\sigma,y}^i \right) & \left( G_\rho * \sum_{i=1}^d I_{\sigma,y}^i \cdot I_{\sigma,y}^i \right) \end{pmatrix} \quad (1.38)$$

où  $I_{\sigma,x}^i$  et  $I_{\sigma,y}^i$  sont respectivement la dérivée en  $X$  et en  $Y$  de la  $i^{\text{ème}}$  composante de  $\mathbf{I}$  ( $1 \leq i \leq d$ ) calculées sur un support gaussien  $G_\sigma$  d'écart type  $\sigma$ . La convolution par la gaussienne d'écart type  $\rho$  joue le rôle de la sommation et renseigne sur la taille de la fenêtre dans laquelle l'information d'orientation est moyennée. Dans le cas d'une image en niveau de gris ( $d = 1$ ),  $J_\rho(\nabla \mathbf{I}_\sigma)$  revient à convoluer le tenseur monospectral  $\mathbf{M}$  avec une gaussienne d'écart type  $\rho$  :  $J_\rho(\nabla \mathbf{I}_\sigma) = G_\rho * \mathbf{M}$ . Les valeurs propres de  $J_\rho(\nabla \mathbf{I}_\sigma)$  associées déterminent le contraste local dans les directions propres.

Le tenseur de structure s'utilise également afin de calculer des points d'intérêt [HS88], cependant, pour éviter le calcul de ces valeurs propres de  $J_\rho(\nabla \mathbf{I}_\sigma)$  avec  $d = 1$ , Schmid et al. [SMB00] proposent d'estimer les maxima locaux de l'opérateur  $H$  suivant :

$$H = \text{Det}(J_\rho(\nabla \mathbf{I}_\sigma)) - k \cdot \text{Trace}^2(J_\rho(\nabla \mathbf{I}_\sigma)).$$

où  $k$  est une variable fixée généralement telle que  $k = 0.04$ . Le tenseur de structure couleur permet de calculer les points de Harris précis en couleur [MGD98].

## 4.5 Tenseur de courbure

Les crêtes sont les maxima dans une image  $I$  en niveau de gris, les vallées sont les minima (voir partie 6.5 de ce chapitre). En effet, en considérant la surface image  $S$ , dans l'espace cartésien, l'ensemble des points des coordonnées  $(x, y, I(x, y))$  sont définis par l'équation :

$$\vec{S}(x, y) = \begin{pmatrix} x \\ y \\ I(x, y) \end{pmatrix}. \quad (1.39)$$

Soit  $S_T(x, y) = \{\vec{S}_x(x, y), \vec{S}_y(x, y)\}$  le plan tangent de la surface  $\vec{S}(x, y)$  en chaque point où :

$$\vec{S}_x(x, y) = \frac{\partial \vec{S}(x, y)}{\partial x} = \begin{pmatrix} 1 \\ 0 \\ I_x(x, y) \end{pmatrix}. \quad (1.40)$$

$$\vec{S}_y(x, y) = \frac{\partial \vec{S}(x, y)}{\partial y} = \begin{pmatrix} 0 \\ 1 \\ I_y(x, y) \end{pmatrix}. \quad (1.41)$$

où  $I_x$  et  $I_y$  sont les dérivées partielles de  $I$  selon respectivement l'axe des  $x$  et des  $y$ .

Les crêtes et les vallées sont données par les points où les valeurs de  $\vec{S}(x, y)$  sont les maxima ou respectivement les minima dans une direction donnée.

Cette direction est la normale à la courbe tracée par la crête ou la vallée au point  $(x, y)$ . La première et la seconde forme fondamentale  $F_1$  et  $F_2$  des surfaces paramétrées peuvent être utilisées pour calculer les courbures principales et les directions principales de courbure en chaque point de la surface  $\vec{S}(x, y)$ .

Dans la base  $\{\vec{S}_x(x, y), \vec{S}_y(x, y)\}$ ,  $F_1$  est définie par :

$$F_1 = \begin{pmatrix} \|\vec{S}_x(x, y)\|^2 & \vec{S}_x(x, y) \cdot \vec{S}_y(x, y) \\ \vec{S}_x(x, y) \cdot \vec{S}_y(x, y) & \|\vec{S}_y(x, y)\|^2 \end{pmatrix} \quad (1.42)$$

$$= \begin{pmatrix} 1 + I_x^2(x, y) & I_x I_y(x, y) \\ I_x I_y(x, y) & 1 + I_y^2(x, y) \end{pmatrix}. \quad (1.43)$$

$$(1.44)$$

$F_1$  permet de calculer la longueur d'une courbe dans la base  $\{\vec{S}_x(x, y), \vec{S}_y(x, y)\}$ .

En définissant  $\vec{N}(x, y)$  la normale à la surface  $\vec{S}(x, y)$  en tout point  $(x, y)$  :

$$\vec{N}(x, y) = \frac{\vec{S}_x(x, y) \wedge \vec{S}_y(x, y)}{\|\vec{S}_x(x, y) \wedge \vec{S}_y(x, y)\|} \quad (1.45)$$

$F_2$  est définie par :

$$F_2 = \begin{pmatrix} \vec{N}(x, y) \cdot \vec{S}_{xx}(x, y) & \vec{N}(x, y) \cdot \vec{S}_{xy}(x, y) \\ \vec{N}(x, y) \cdot \vec{S}_{xy}(x, y) & \vec{N}(x, y) \cdot \vec{S}_{yy}(x, y) \end{pmatrix} \quad (1.46)$$

$$= \vec{N}(x, y) \cdot \begin{pmatrix} \vec{S}_{xx}(x, y) & \vec{S}_{xy}(x, y) \\ \vec{S}_{xy}(x, y) & \vec{S}_{yy}(x, y) \end{pmatrix} \quad (1.47)$$

$$= \frac{\vec{S}_x(x, y) \wedge \vec{S}_y(x, y)}{\|\vec{S}_x(x, y) \wedge \vec{S}_y(x, y)\|} \cdot \begin{pmatrix} \vec{S}_{xx}(x, y) & \vec{S}_{xy}(x, y) \\ \vec{S}_{xy}(x, y) & \vec{S}_{yy}(x, y) \end{pmatrix} \quad (1.48)$$

où  $\vec{S}_{xx}(x, y)$ ,  $\vec{S}_{yy}(x, y)$  et  $\vec{S}_{xy}(x, y)$  représentent les dérivées secondes de  $\vec{S}(x, y)$  selon l'axe des  $x$  et des  $y$ .

Donc, après développement et factorisation :

$$F_2 = \frac{1}{\sqrt{1 + I_x^2 + I_y^2}} \cdot \begin{pmatrix} I_{xx}^2(x, y) & I_x I_y(x, y) \\ I_x I_y(x, y) & I_{yy}^2(x, y) \end{pmatrix} \quad (1.49)$$

$$= \frac{1}{\sqrt{1 + I_x^2 + I_y^2}} \cdot \mathcal{H}(x, y) \quad (1.50)$$

où  $\mathcal{H}(x, y)$  est la matrice hessienne de l'image composée des dérivées secondes de l'image  $I_{xx}^2(x, y)$ ,  $I_x I_y(x, y)$  et  $I_{yy}^2(x, y)$ .  $F_2$  permet de calculer les courbures dans la base  $\{\vec{S}_x(x, y), \vec{S}_y(x, y)\}$ .

L'endomorphisme de Weingarten dont la matrice  $W = F_1^{-1} F_2$  donne l'expression des courbures de la surface image dans le plan image :

$$W = F_1^{-1} F_2 = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix}. \quad (1.51)$$

Les vecteurs propres maximaux et minimaux de  $W$  indiquent les directions des courbures maximales et minimales et les courbures principales de l'image :

$$\vec{v}_1 = \begin{pmatrix} w_{12} \\ k_1 - w_{11} \end{pmatrix} \quad \vec{v}_2 = \begin{pmatrix} w_{12} \\ k_2 - w_{11} \end{pmatrix}. \quad (1.52)$$

Les courbures principales correspondent aux valeurs propres  $k_1$  et  $k_2$  et les directions principales aux vecteurs propres  $\vec{v}_1$  et  $\vec{v}_2$  de  $W$ .

La matrice hessienne  $\mathcal{H}(x, y)$  seule peut être utilisée pour estimer la courbure maximale [Ste98] [Lin98]. Les intersections des deux plans définis par la normale  $\vec{n}(x, y)$  à la surface  $\vec{S}$  et les vecteurs propres ( $\vec{V}_1, \vec{V}_2$ ) en chaque pixel  $(x, y)$  de l'image forment sur la surface  $\vec{S}$  les deux courbes de la surface appelées  $\mathcal{C}_1(x, y)$  et  $\mathcal{C}_2(x, y)$ . Toutefois, au niveau du sommet d'une ligne de crête, le plan tangent à cette ligne de crête est parallèle au plan (Oxy), donc la matrice  $F_1$  devient la matrice identité et  $W = \mathcal{H}$ . Alors la matrice hessienne seule permet de calculer les lignes de crêtes, cependant, lorsque la ligne de crête est en pente, la matrice  $F_1$  est différente de la matrice identité, dans ces cas, il faudrait utiliser  $W$ .

## 5 Historique en détection de contours dans les images

### 5.1 Introduction

Les contours dans une image représentent souvent les bords de différents objets ou des caractéristiques physiques des objets (discontinuités de réflectance) ou d'une scène observée telles que les ombres, les textures ou les matières absorbantes de lumière. Comme illustré en figure 1.4, nous pouvons définir différents types de contours dans les images. Les contours dans une image constituent des indices riches, au même titre que les points d'intérêts [TM08]. En effet, les résultats de détection de bords bénéficient à une gamme très large d'applications telles que la segmentation, l'amélioration d'image, la reconnaissance d'objets ou de personnes, le morphing, l'enregistrement, la compression, l'extraction, le tatouage, etc. L'extraction de contours est un processus qui détecte la présence et la localisation des bords constitués par de brusques variations d'intensités de couleurs (ou de luminosités) d'une image.

Il existe deux grands types de contours, marche ou rampe (en haut de la figure 1.4) qui peuvent être extraits en général avec un opérateur d'ordre un et ceux de type toit ou pointe (en bas de la figure 1.4) qui sont détectés avec un opérateur d'ordre deux.

Dans le cas idéal, le résultat de l'application d'un détecteur de contours dans une image devrait conduire à un ensemble de courbes connectées qui indiquent les limites des objets. Toutefois, il n'est en général pas possible d'obtenir ces bords idéaux à partir d'images réelles qui sont souvent complexes. Les contours extraits ne sont souvent pas connectés ou manquants (jonctions, ombres, occultations, ...), ce sont les faux négatifs, ainsi que les faux indésirables (par exemple à l'intérieur d'une texture) appelés faux positifs, qui conduisent tous à compliquer la tâche d'extraction des contours des objets.

Une propriété importante d'une méthode de détection de bords est sa capacité à extraire la ligne de contours avec précision et une orientation correcte. La détection

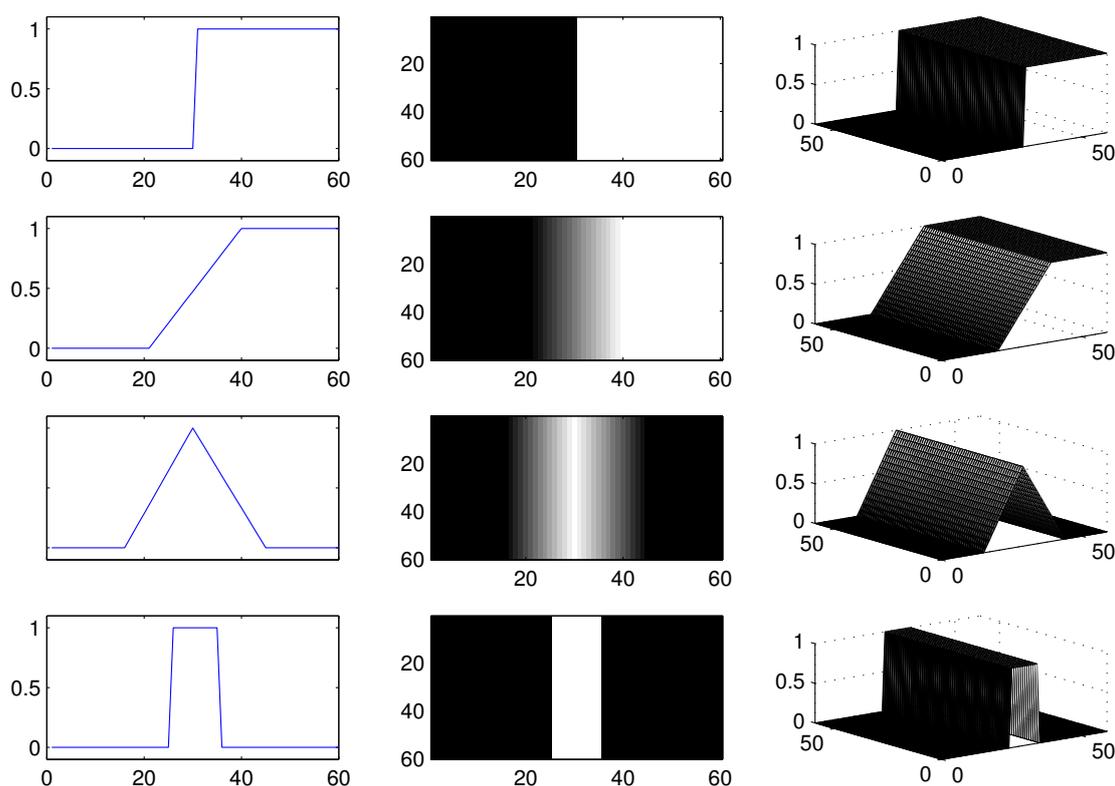


FIGURE 1.4 – Les différents types de contours dans les images. De gauche à droite : représentation d'un contour en une dimension, bord vertical dans une image et représentation en trois dimensions du contour (*les axes des abscisses et des ordonnées représentent les coordonnées des pixels et l'axe de cote l'intensité lumineuse des pixels*). De haut en bas : contour marche, rampe, toit et point.

de contours repose en principe sur l'étude des dérivées de la fonction d'intensité lumineuse de l'image. En effet, les extrema locaux du gradient d'intensité lumineuse permettent de calculer la présence d'un bord dans l'image. Il en va de même avec les passages par zéro en utilisant le laplacien. Plusieurs méthodes permettent de déterminer le gradient ou le laplacien d'une image.

En règle générale, une méthode de détection de bord du premier ordre (c'est à dire utilisant la dérivée première) peut être décomposée en trois étapes. Nous allons détailler différentes approches du premier ordre qui vont des masques de convolution aux filtres récurrents. Pour toutes ces approches, la première étape consiste à réduire le bruit l'image. La figure 1.5 illustre différents types de contours contenant du bruit. Afin d'obtenir de meilleures performances de détection de bord, le bruit d'image doit être atténué autant que possible comme montré en figure 1.6. Le contour idéal est celui de type marche qui est représenté par la fonction échelon ou *Heaviside* :

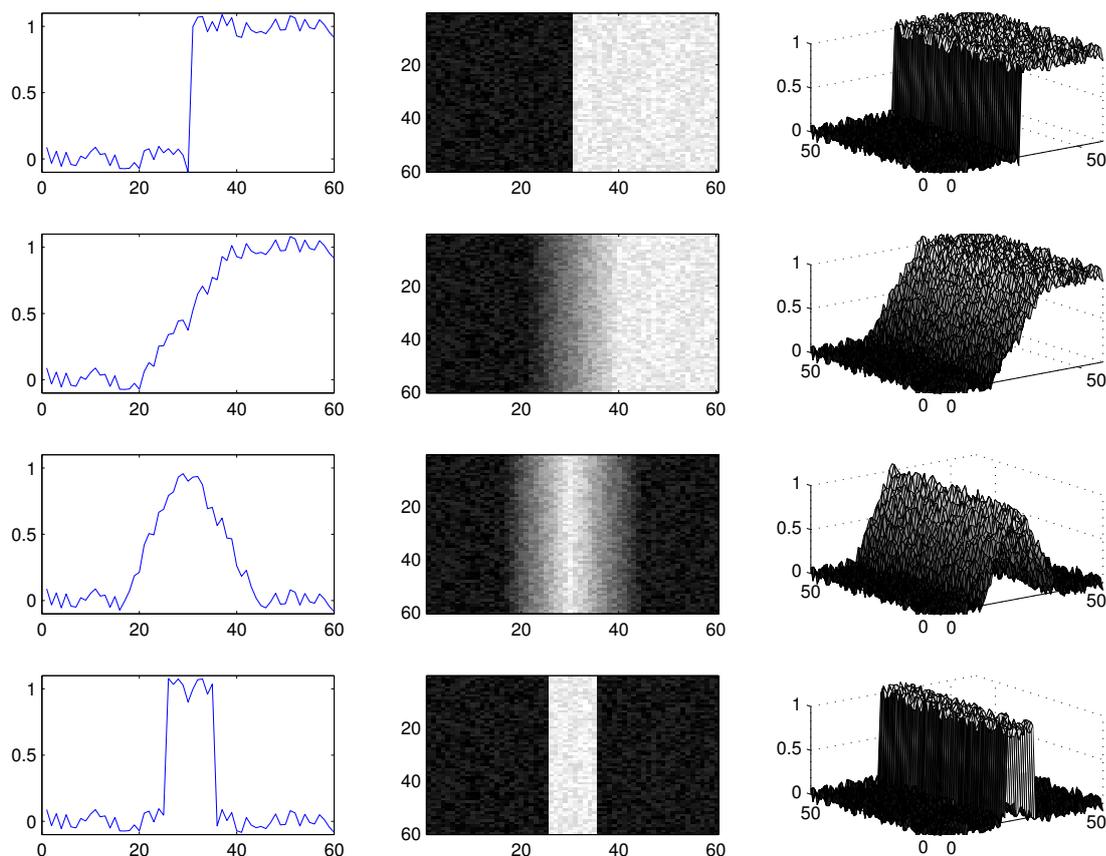


FIGURE 1.5 – Les différents types de contours dans les images avec un bruit additif uniforme aléatoire.

$$E(t) = \begin{cases} 0 & \text{si } t < 0 \\ 1 & \text{si } t \geq 0. \end{cases} \quad (1.53)$$

Un signal comportant un contour en  $t = 0$  ayant un saut d'intensité d'amplitude  $A$  contenant un bruit blanc gaussien additif  $n(t)$  de variance  $\sigma^2 = E(n^2(t))$  peut alors être modélisé par :

$$x(t) = A E(t) + n(t). \quad (1.54)$$

Même si le contour idéal est de forme marche, dans la réalité, l'éclairage, les mouvements ou même l'échantillonnage de l'image apportent du flou aux contours. Bourennane et al. [BGPT02] ont ainsi modélisé les contours flous par l'équation suivante :

$$C(t) = \begin{cases} 1 - \frac{e^{-s \cdot x}}{2} & \text{si } t \geq 0 \\ \frac{e^{-s \cdot x}}{2} & \text{si } t < 0 \end{cases} \quad \text{avec } s > 0. \quad (1.55)$$

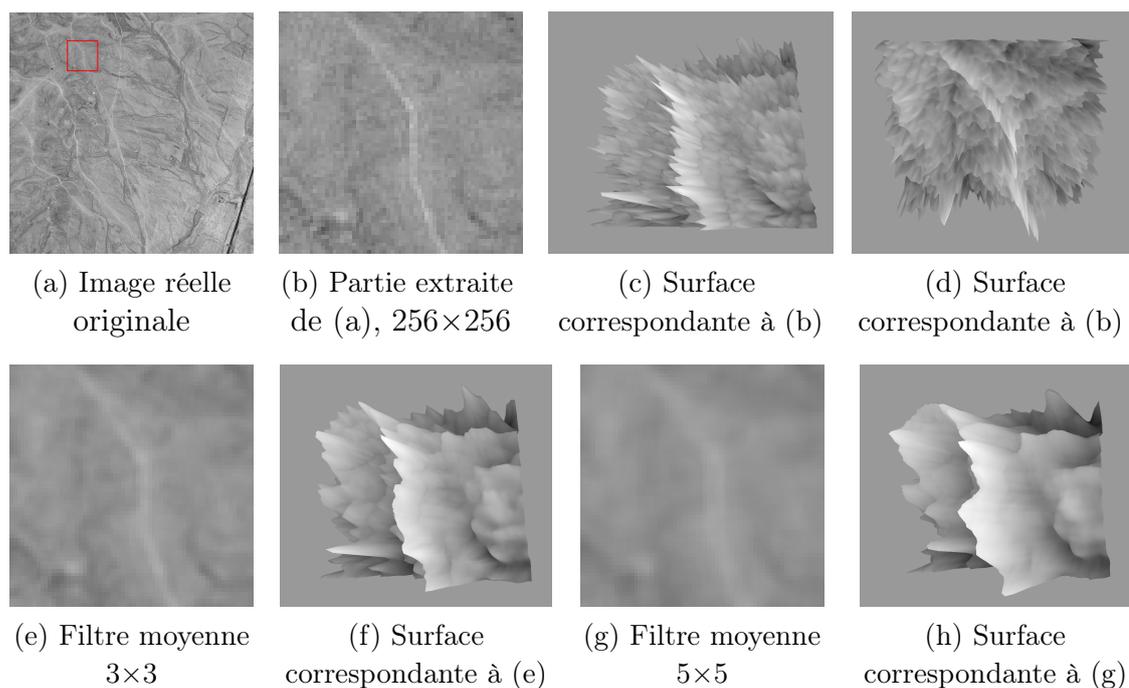


FIGURE 1.6 – Représentation 3D d’une image avec et après lissage avec un filtre moyenne.

Généralement, les contours sont modélisés avec un bruit additif. Toutefois, les contours ont une extension spatiale et sont moins affectés que le bruit par le processus de régularisation, ce qui permet de les détecter en ignorant le bruit. Cette réduction du bruit est habituellement obtenue en effectuant un filtrage passe-bas parce que le bruit est normalement un signal à haute fréquence (un exemple de lissage est présenté en figure 1.6). Cependant, les bords peuvent éventuellement être atténués par ce processus, parce qu’ils sont aussi des signaux haute fréquence. Par conséquent, le paramètre du filtre passe bas est couramment ajusté pour trouver un meilleur compromis entre la réduction du bruit et préservation de l’information bords. Dans la deuxième phase, un filtre de type dérivation lissé est habituellement employé pour trouver les bords. Cela revient à un filtre passe bande. Dans la dernière étape, un processus de localisation de bords est réalisé afin d’identifier les bords réels, qui se distinguent de ceux des réponses similaires causés par le bruit et ainsi éviter les fausses détections.

Nous allons également étudier d’autres processus capables d’extraire un autre type de contours appelés contours toits ou crêtes qui sont calculés à partir des dérivées du second ordre.

Les masques, les filtres isotropiques du premier et second ordres ainsi que les filtres orientés vont être détaillés dans les parties suivantes avec des résultats pour illustrer ces différentes approches.

## 5.2 Les Masques

Les images filtrées sont calculées par convolution de l’image d’entrée avec des masques directionnels approximant le gradient de l’image dans différentes directions par différences finies.

### Dérivée à gauche et dérivée à droite

$\begin{array}{ c c } \hline -1 & 1 \\ \hline \end{array}$	$\begin{array}{ c } \hline 1 \\ \hline -1 \\ \hline \end{array}$
(a) Approximation de la dérivée première en $x$	(b) Approximation de la dérivée première en $y$

FIGURE 1.7 – Masques de base pour l’approximation du gradient pixel à pixel.

Les premières approches de la dérivée première à gauche ont donc consisté à approximer les dérivées par différences finies pixel à pixel :

$$I_x(x, y) = I(x, y) - I(x - 1, y), \quad (1.56)$$

$$I_y(x, y) = I(x, y) - I(x, y - 1), \quad (1.57)$$

ou par un masque  $3 \times 1$  sur trois pixels :

$$I_x(x, y) = \frac{1}{2} (I(x + 1, y) - I(x - 1, y)), \quad (1.58)$$

$$I_y(x, y) = \frac{1}{2} (I(x, y + 1) - I(x, y - 1)). \quad (1.59)$$

Les masques les plus intuitifs (équations 1.56 et 1.57) à mettre en oeuvre sont deux masques à deux éléments comme montrés en figure 1.7. Très proche de ce dernier, l’estimation de la dérivée sur trois pixels comme en équations 1.58 et 1.59 peut également se calculer sous forme de masques illustrés en figure 1.8. L’approximation de la dérivée de l’image  $\|\nabla I\|$  est ensuite estimée en utilisant l’équation 1.20 et sa direction  $\eta$  d’après l’équation 1.21. La figure 1.9 illustre l’approximation du gradient avec les masques ainsi décrits.

$\begin{array}{ c c c } \hline -1 & 0 & 1 \\ \hline \end{array}$	$\begin{array}{ c } \hline -1 \\ \hline 0 \\ \hline 1 \\ \hline \end{array}$
(a) Approximation de la dérivée première en $x$	(b) Approximation de la dérivée première en $y$

FIGURE 1.8 – Masque de base pour l’approximation du gradient sur trois pixels. Ces masques nécessitent un coefficient de normalisation de  $\frac{1}{2}$  afin d’obtenir les dérivées exactes.

### Masques de Roberts

Le masque de Roberts proposé en 1965 [Rob65] permet de calculer un gradient selon des axes orientés à 45 degrés.

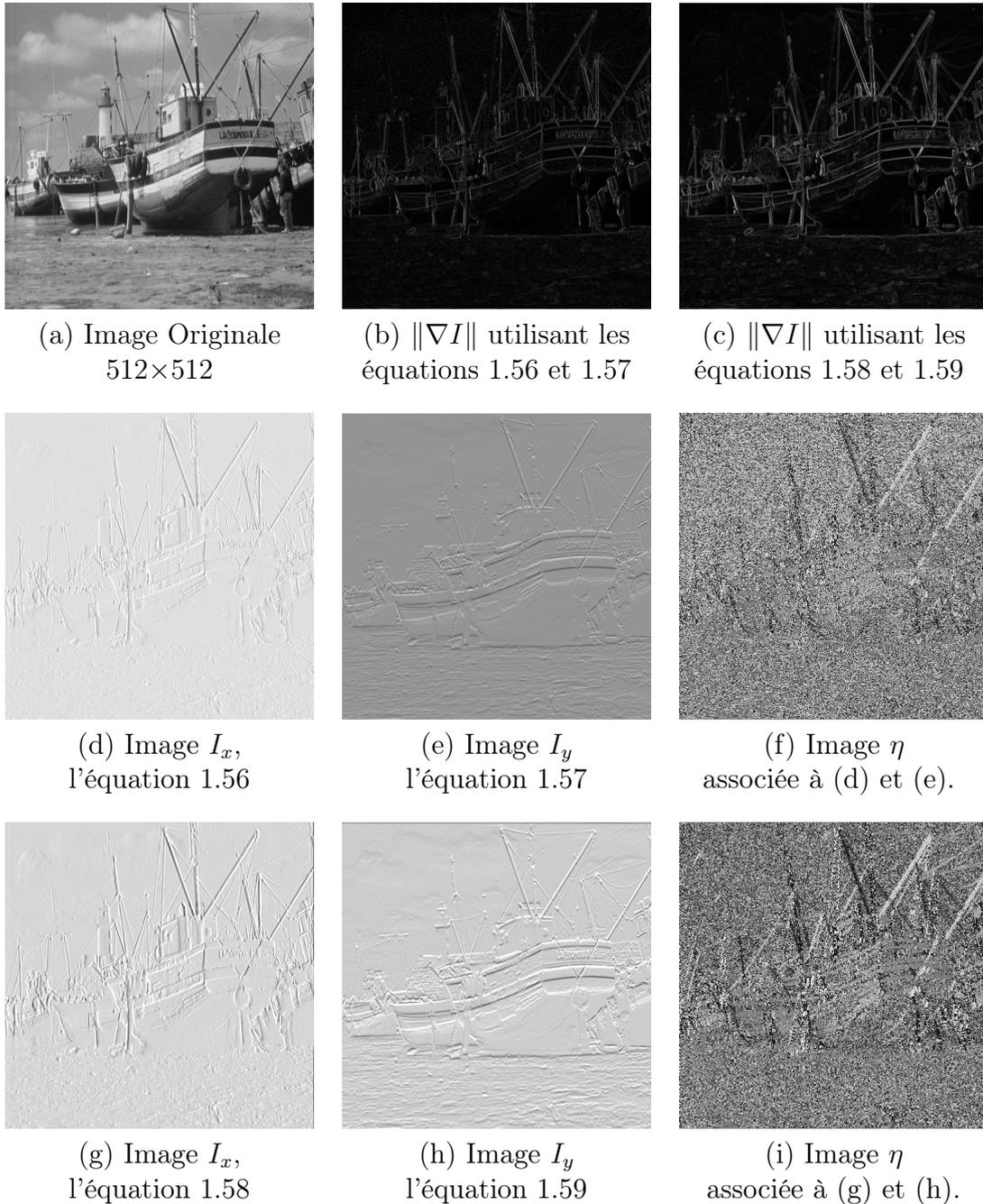


FIGURE 1.9 – Approximation du gradient de base en utilisant les masques  $[-1 \ 1]$  ou  $[-1 \ 0 \ 1]$  horizontaux et verticaux. Toutes ces images sont normalisées.

$$\begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array} \quad \begin{array}{|c|c|} \hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array}$$

(a)  $H_1$       (b)  $H_2$

FIGURE 1.10 – Masques de Roberts  $H_1$  et  $H_2$  orientés à 45 degrés.



(a) Image Originale 512×512      (b)  $\|\nabla I\|$  de Roberts

FIGURE 1.11 – Approximation du gradient de Roberts, image normalisée.

Ces masques sont illustrés en figure 1.10 et la direction du gradient est donnée par :

$$\eta(x, y) = \arctan \frac{I(x, y) * H_2}{I(x, y) * H_1} + \frac{\Pi}{4}. \tag{1.60}$$

Un résultat d'estimation de la dérivée d'une image est disponible en figure 1.11. Toutefois, les approximations du gradient présentées jusqu'à présent ont une forte sensibilité au bruit en raison de la petite taille des masques utilisés. D'autres masques plus larges ont ainsi été proposés afin de rendre le filtrage moins sensible au bruit.

### Masques de Prewitt

Les filtres 3×3 introduits par Prewitt [Pre70] pour extraire des contours ont une précision inférieure aux masques précédents car le masque est plus large (figure 1.12).

$$\begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

(a)  $H_1$       (b)  $H_2$

FIGURE 1.12 – Masques de Prewitt :  $H_1$  pour la dérivée en  $X$  et  $H_2$  pour la dérivée en  $Y$ .


 (a) Image Originale 512×512      (b)  $\|\nabla I\|$  de Prewitt

FIGURE 1.13 – Approximation du gradient de Prewitt.

Cependant, les contours ainsi obtenus sont généralement plus fiables que ceux obtenus avec les masques  $2 \times 2$  ou  $1 \times 2$  puisqu'ils sont moins sensibles au bruit comme l'illustre la figure 1.13. Le calcul du gradient s'effectue dans une direction, tandis qu'un lissage s'effectue dans la direction orthogonale. De plus, ils sont également centrés sur un pixel et non entre des pixels (filtres à phase nulle<sup>2</sup>). La valeur de la somme du produit du masque par les valeurs des pixels correspondant est retenue. Puis le masque est tourné de 90 degrés autour de son centre et la même mesure est répétée. En notant  $I_x$  et  $I_y$  ces deux mesures, leur norme est alors calculée comme en équation 1.20 afin de constituer la mesure du module du gradient en ce point ainsi que sa direction associée par la formule de l'équation 1.21.

### Masques de Sobel

Le filtre de Sobel [Sob70] illustré en figure 1.14 est très similaire à celui de Prewitt. Ce masque est le plus utilisé dans les applications industrielles ou robotiques, nécessitant des contraintes temps-réel (un résultat est disponible en figure 1.15).

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

(a)  $H_1$ 
(b)  $H_2$

 FIGURE 1.14 – Masques de Sobel :  $H_1$  pour la dérivée en  $X$  et  $H_2$  pour la dérivée en  $Y$ .

### Convolution de l'image avec un masque $3 \times 3$

Les opérateur de Prewitt et de Sobel peuvent se décomposer en un masque de gradient en  $X$  et un de lissage en  $Y$ . De plus, les filtres  $H_1$  et  $H_2$  sont séparables :

<sup>2</sup>. Les phases des filtres ont été présentées dans [MD97a]

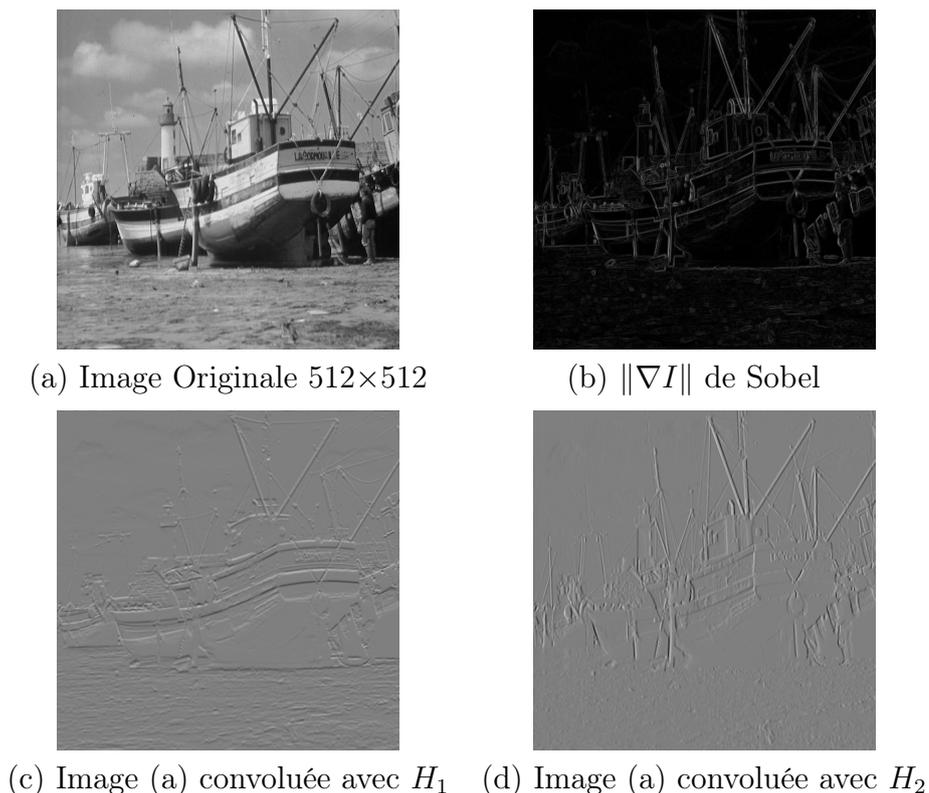


FIGURE 1.15 – Approximation du gradient de Sobel.

$$\begin{aligned} H_1 &= D_{1X} * L_{1Y}, \\ H_2 &= L_{1X} * D_{1Y}, \end{aligned}$$

avec  $D_{1X} = [-1 \ 0 \ 1]$ ,  $L_{1Y} = [1 \ c \ 1]$ ,  $D_{1Y} = [-1 \ 0 \ 1]$ ,  $L_{1X} = [1 \ c \ 1]$  ( $c = 1$  pour Prewitt et  $c = 2$  pour Sobel). Donc l'approximation de la dérivée en  $X$ ,  $I_x(x, y) = H_1 * I(x, y)$  et celle en  $Y$ ,  $I_y(x, y)$  peut se décomposer de la même manière avec  $H_2$ .

### Les masques boussole

Les opérateurs dits boussole consistent à filtrer l'image avec 8 masques directionnels  $3 \times 3$  appelés  $G_k$ ,  $k = 1, 8$ . L'orientation du contour est alors donnée par le filtre dont la réponse  $G$  est la plus importante :

$$G(x, y) = \max_{k=1,8} G_k(x, y). \quad (1.61)$$

Il existe de nombreux masques de ce type, dont celui de Kirsch [Kir71] et de Robinson [Rob77] présentés en figures 1.16 et 1.17. Aussi, les masques de Prewitt et Sobel peuvent également être implémentés avec des permutations circulaires, il en existe d'autres et de différentes tailles ( $5 \times 5$  [NB80],  $7 \times 7 \dots$ ) qui ont été étudiés [CD85] [Dav75]. Deux résultats des masques de Kirsch et de Robinson sont présentés en figure 1.18.

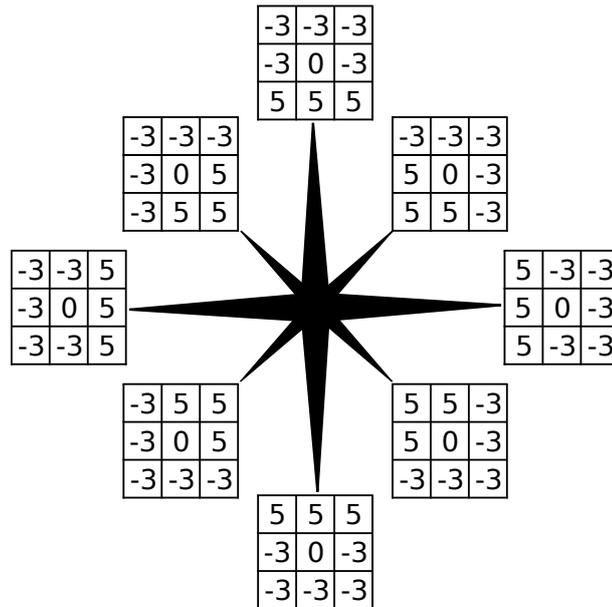


FIGURE 1.16 – Opérateurs gradient boussole avec les masques de Kirsch.

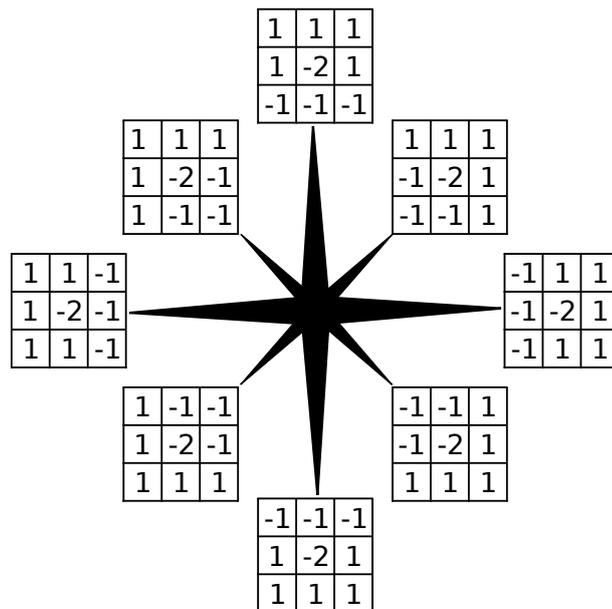


FIGURE 1.17 – Opérateurs gradient boussole avec les masques de Robinson.

### 5.3 Laplacien de gaussienne

Marr et Hildreth ont proposé dans [MH80] de convoluer l'image  $I$  avec une gaussienne  $G_\sigma$  d'écart type  $\sigma$  afin de régulariser l'image avant d'appliquer un opérateur laplacien. Cette méthode est couramment appelée LoG (*Laplacian of Gaussian*).

En une dimension,  $G_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$ , pour un signal  $s$  donné, le LoG revient à :

$$\Delta(s * G_\sigma)(x) = (\Delta s) * G_\sigma(x) = s * (\Delta G_\sigma(x)), \quad (1.62)$$



(a) Image Originale 512×512      (b)  $G$  de Kirsch      (c)  $G$  de Robinson

FIGURE 1.18 – Approximation du gradient de Kirsch et Robinson.

où

$$\Delta G_\sigma(x) = \frac{\partial^2}{\partial^2 x} G_\sigma(x) = \frac{x^2 - \sigma^2}{\sqrt{2\pi} \cdot \sigma^5} e^{-\frac{x^2}{2\sigma^2}}.$$

En deux dimensions,  $G_\sigma(x, y)$  s'écrit :

$$G_\sigma(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

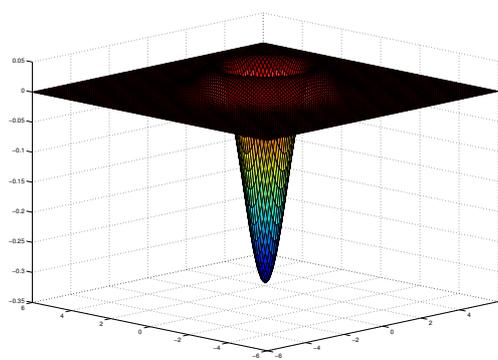
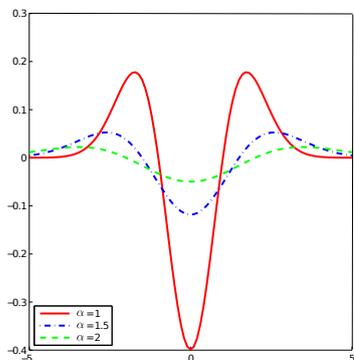
Dans le cas continu,  $\Delta G_\sigma(x, y)$  s'écrit en un pixel de coordonnées  $(x, y)$  :

$$\Delta G_\sigma(x, y) = \frac{\partial^2}{\partial^2 x} G_\sigma(x, y) + \frac{\partial^2}{\partial^2 y} G_\sigma(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{\sqrt{2\pi} \cdot \sigma^5} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1.63)$$

Dans le cas discret,  $\Delta G_\sigma(x, y)$  s'écrit :

$$\Delta G_\sigma(r) = (C_{2a} + C_{2b} \cdot (x^2 + y^2)) e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (1.64)$$

où  $(C_{2a}, C_{2b})$  représentent des coefficients de normalisation de manière à obtenir un filtre de moyenne nulle.



(a) Laplacien avec différents  $\sigma$       (b) Laplacien en 2 dimensions,  $\sigma = 1$

FIGURE 1.19 – Laplacien de gaussienne.

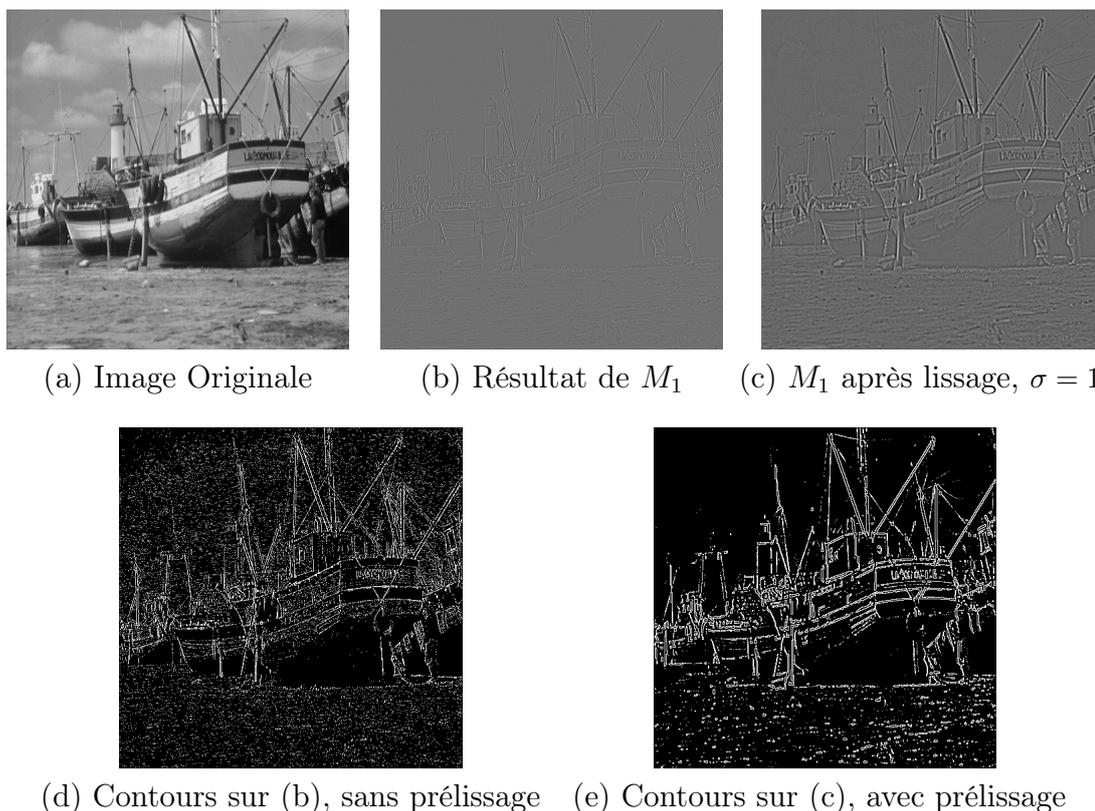
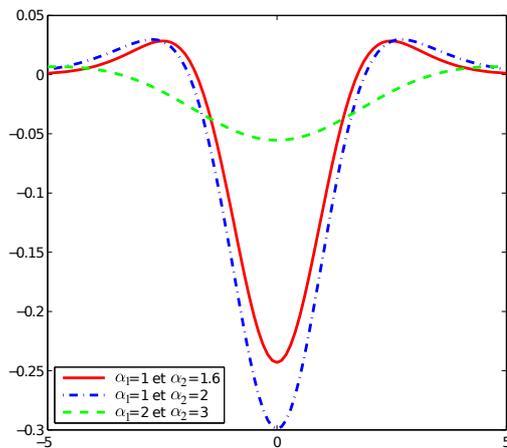


FIGURE 1.20 – Résultat du masque  $M_1$  avec et sans image pré-lissée (images normalisées).

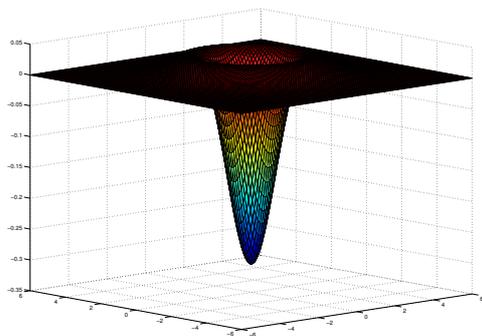
La variance de la gaussienne (voire figure 1.21) limite l’extension du domaine fréquentiel de l’image d’entrée et atténue le bruit avant d’appliquer le laplacien. La méthode revient donc à détecter les passages par zéro du laplacien de l’image préfiltrée par une gaussienne. Cependant, un filtre de variance trop large dégrade le signal contenu dans l’image et ne permet plus la détection des structures fines ce qui crée des faux négatifs. A l’inverse, si la variance est trop faible, le bruit ne sera pas suffisamment filtré et donc sera détecté en tant que contours, créant ainsi des faux positifs.

## 5.4 Différence de gaussienne

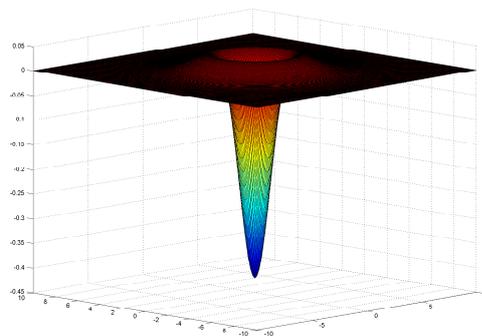
Un autre filtre très utilisé et très proche du LoG est le DoG (*Difference of Gaussians*), soit un filtrage par différence de deux gaussiennes dont les écarts-type sont appelés  $\sigma_1$  et  $\sigma_2$  sont de valeurs différentes. Souvent, le rapport entre les deux écarts types est de 1.6, ce qui approxime le laplacien (voire figure 1.21). L’image est convoluée d’une part par deux gaussiennes d’écart type différents, puis l’image de DoG est formée par différence des deux images convoluées comme montré en figure 1.22. Enfin, les contours sont évalués par une méthode de passages par zéros.



(a) DoG en une dimension avec différents  $\sigma_1$  et  $\sigma_2$



(b)  $\sigma_1 = 1$  et  $\sigma_2 = 1.6$



(c)  $\sigma_1 = 1$  et  $\sigma_2 = 2.5$

FIGURE 1.21 – Différence de gaussiennes en 2 dimensions.



(a) Image Originale



(b)  $\sigma_1 = 1$  et  $\sigma_2 = 1.6$



(c)  $\sigma_1 = 1$  et  $\sigma_2 = 2.5$

FIGURE 1.22 – Détection de contours utilisant les DoG (images normalisées).

## 6 Critères de Performances de Canny et Filtrage Optimal

Les approximations des dérivations des images présentées précédemment consistent à convoluer l'image par des masques de petites dimensions (généralement  $3 \times 3$ ) et sont basées sur une modélisation assez simple d'un contour idéal. Ces approches sont donc dépendantes de la taille des objets traités, elles sont aussi très sensibles au bruit. De plus, ces approches sont empiriques. Un autre type d'approche repose sur la définition de critères d'optimalité de la détection de contours; ces critères débouchant sur des filtres de lissage optimaux. Torre et Poggio [TP86] ont montré qu'il est nécessaire de régulariser avant d'appliquer une dérivation. D'après Canny [Can86], analytiquement, les performances d'un détecteur de bords se caractérisent essentiellement par trois critères d'optimalité :

1. détection : l'opérateur doit donner une réponse maximale au voisinage d'un contour et minimiser les fausses réponses, même en présence de faibles contours sur l'image d'entrée,
2. localisation : le contour doit être localisé avec précision, c'est à dire minimiser la distance entre les points détectés et le vrai contour,
3. réponse unique : un contour unique doit provoquer une réponse unique de l'opérateur d'extraction, ce qui assure une seule détection pour un contour et ainsi évite les effets de rebond.

Cette approche a permis une bien meilleure compréhension des conditions d'une bonne détection de contours.

### 6.1 Les Approches Analytiques

Dans le cas monodimensionnel, l'optimisation des trois critères proposés permet de définir le filtre linéaire optimal pour la détection de contours de type marche sous l'hypothèse d'un bruit blanc additif (de moyenne nulle). Il s'agit de détecter le contour en convoluant le signal par un filtre dont la réponse impulsionnelle est  $h(x)$ . Cela permet d'obtenir une valeur maximum en sortie lorsqu'un contour est présenté en entrée.

Pour Canny, le contour idéal est celui de type marche qui est représenté par la fonction échelon (*heaviside*) en équation 1.54.

La moyenne des échantillons est nulle et les échantillons sont décorrélés. La sortie  $y(t)$  s'écrit comme la convolution de l'entrée  $x(t)$  par la réponse impulsionnelle du filtre recherché  $h(t)$  :

$$y(t) = x(t) * h(t) = \int_{-\infty}^{+\infty} x(v)h(t - v) dv. \quad (1.65)$$

$$y(t) = \int_{-\infty}^{+\infty} A h(t - v) E(t) dv + \int_{-\infty}^{+\infty} n(v) h(t - v) dv \quad (1.66)$$

$$= \int_{-\infty}^t A h(t - v) dv + \int_{-\infty}^{+\infty} n(v) h(t - v) dv \quad (1.67)$$

$$= S(t) + N(t). \quad (1.68)$$

### Critère de bonne décision

L'accentueur  $h(t)$  a été choisi de telle façon à maximiser le rapport signal sur bruit ( $RSB$ ) de l'image filtrée aux points de contours, soit maximiser au point  $t = 0$  :

$$RSB = \frac{\text{amplitude du signal}}{\sqrt{\text{variance du bruit}}} \quad (1.69)$$

$$= \frac{S(0)}{\sqrt{E(n^2(0))}} \quad (1.70)$$

$$= \frac{\int_{-\infty}^0 A h^2(t) dt}{n(0) \sqrt{\int_{-\infty}^{+\infty} h^2(t) dt}}. \quad (1.71)$$

ce qui conduit à maximiser la quantité :

$$\Sigma(h) = \frac{\int_{-\infty}^0 h(t) dt}{\sqrt{\int_{-\infty}^{+\infty} h^2(t) dt}}. \quad (1.72)$$

### Critère de bonne localisation

Canny cherche de plus une localisation optimale du contour, par minimisation de l'écart quadratique moyen entre le contour détecté et le contour réel. Ceci conduit donc à maximiser :

$$\Lambda(h) = \frac{|h'(0)|}{\sqrt{\int_{-\infty}^{+\infty} h'^2(t) dt}}. \quad (1.73)$$

### Critère d'unicité de la réponse

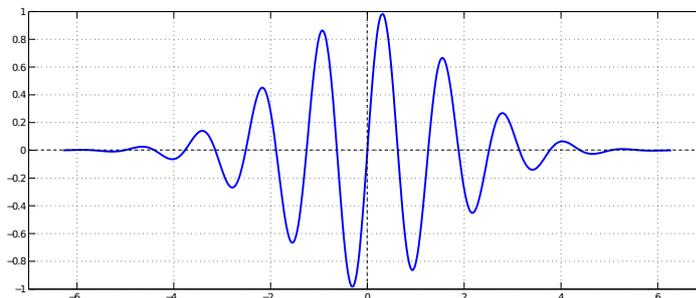


FIGURE 1.23 – Filtre contenant plusieurs maxima locaux secondaires.

Enfin, l'opérateur de réhaussement trouvé ne doit pas détecter plusieurs maxima locaux pour un seul contour comme le filtre illustré en figure 1.23. Le filtre trouvé doit être antisymétrique sans maxima secondaires. Ceci s'exprime à l'aide d'une estimation de la densité des passages par zéro de la fonction  $h'$  appelée  $D$  qui doit rester fixée à une certaine valeur donnée  $k$  :

$$D = \frac{\pi}{2} \sqrt{\frac{\int_{-\infty}^{+\infty} h''^2(t) dt}{\int_{-\infty}^{+\infty} h'^2(t) dt}} = k. \quad (1.74)$$

La résolution du système revient à maximiser le produit  $\Sigma\Lambda$  sous la contrainte  $D = k$ . La solution à ce problème est fourni par l'équation différentielle suivante :

$$2h(t) - 2\lambda_1 h''(t) + 2\lambda_2 h^{(4)}(t) + \lambda_3 = 0 \quad (1.75)$$

avec  $(\lambda_1, \lambda_2, \lambda_3) \in \mathbb{R}^3$ , dont la solution générale est de la forme :

$$h(t) = a_1 e^{\alpha t} \cos(\omega t) + a_2 e^{\alpha t} \sin(\omega t) + a_3 e^{-\alpha t} \cos(\omega t) + a_4 e^{-\alpha t} \sin(\omega t). \quad (1.76)$$

avec  $(\alpha, \omega, a_1, a_2, a_3, a_4) \in \mathbb{R}^6$ .

Cherchant l'opérateur  $h(t)$  à réponse impulsionnelle finie (RIF) défini sur  $[-W, W]$  comme montré en figure 1.24 et présentant une pente  $S$  à l'origine, les conditions imposées aux limites sont :

$$h(0) = 0, h(W) = 0, h'(0) = S, h'(-W) = 0, \quad (1.77)$$

ce qui permet de déterminer les coefficients  $a_1, a_2, a_3$  et  $a_4$ . Comme  $h(t)$  est une fonction impaire, la solution est étendue aux négatifs avec  $h(t) = -h(-t)$ . La solution trouvée étant loin d'être simple, elle peut cependant être approximée par la dérivée première d'une gaussienne avec comme critère de Canny  $\Sigma\Lambda = 0.92$  et  $k = 0.51$ .

Demigny [Dem02] revisite les expressions des trois critères de Canny pour la détection de bord afin de déterminer un filtre optimal en menant un raisonnement dans le domaine discret directement.

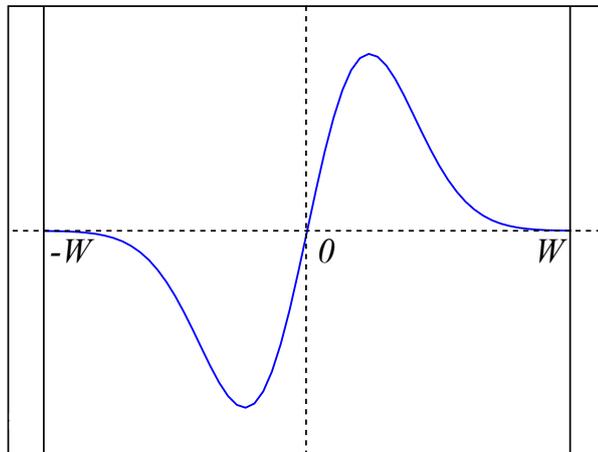


FIGURE 1.24 – Filtre dérivateur défini sur  $[-W, W]$ .

## 6.2 Le Filtre de Deriche

Deriche a généralisé dans [Der87] les idées de Canny détaillées précédemment en proposant une solution exacte avec un coefficient bien meilleur  $\Sigma\Lambda = 2$ . Seules les conditions aux limites sont différentes par rapport à la solution proposée par Canny [Can86] :

$$h(0) = 0, h(+\infty) = 0, h'(0) = S, h'(+\infty) = 0. \quad (1.78)$$

On obtient alors :  $a_1 = a_2 = a_3 = c = 0$ . D'où le filtre de dérivation optimal :

$$h(t) = a_4 \cdot e^{-\alpha|t|} \sin(\omega t). \quad (1.79)$$

Cependant, le filtre proposé possède une réponse impulsionnelle infinie (filtre RII).

Les résultats obtenus en évaluant pour cet opérateur les différentes intégrales intervenant dans le calcul des critères de performance sont :

$$\Lambda = \sqrt{2\alpha} \quad \text{et} \quad \Sigma = \sqrt{\frac{2\alpha}{\alpha^2 + \omega^2}}. \quad (1.80)$$

L'évolution des valeurs de  $\omega$  permet de rechercher où le produit  $\Sigma\Lambda$  est maximum. Pour cela, en posant  $\alpha = m\omega$ , on obtient différentes valeurs de  $k$  selon le  $m$  donné :

$m$	$\lambda$	$\Sigma$	$\Sigma\Lambda$	$k$
$m \gg 1$	$\sqrt{2\alpha}$	$\sqrt{2\alpha}$	2	0.44
$m \ll 1$	$\sqrt{2\alpha}$	$1/m$	$2m$	1
$m = 1$	$\sqrt{2\alpha}$	$\sqrt{1/\alpha}$	$\sqrt{2}$	0.58
$m = \sqrt{3}$	$\sqrt{2\alpha}$	$\sqrt{3/2\alpha}$	$\sqrt{3}$	0.5

Le cas  $m = \sqrt{3}$  montre que pour des valeurs de  $k$  identiques, l'opérateur de Deriche présente un indice de performance près de 90% meilleur que la première dérivée d'une gaussienne.

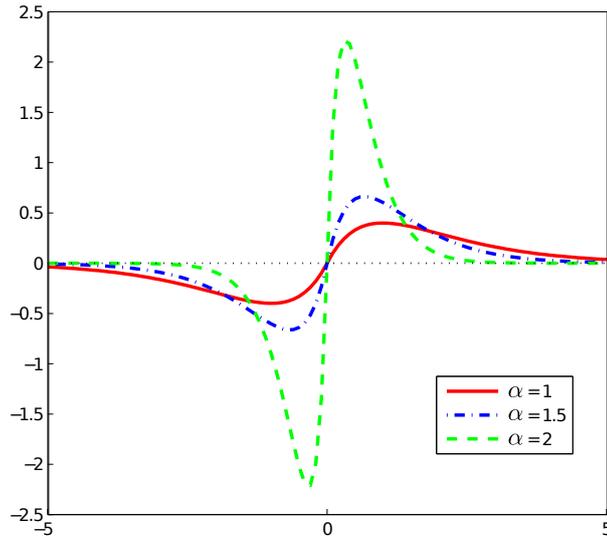
Le cas  $m = 1$  montre que pour une même valeur de  $k$ , l'opérateur de Deriche présente un indice amélioré de 25% par rapport à l'opérateur optimal de Canny qui donne  $\Sigma\Lambda = 1, 12$ .

Le cas  $m \ll 1$  présente une réponse idéale mais la valeur de  $\Sigma\Lambda = 2m$  devrait être largement inférieure à 1.

Enfin, le cas  $m \gg 1$  correspondant à la limite de l'opérateur de Deriche pour  $\omega$  tendant vers 0 présente le meilleur indice de performance  $\Sigma\Lambda$ . Comme  $m \gg 1$ ,  $\omega \approx 0$ , donc  $\sin(\omega t) \approx \omega t$ . Cette limite correspond à l'opérateur de dérivation  $\mathcal{D}(t)$  qui présente le meilleur compromis, donné par :

$$\mathcal{D}(t) = C_1 \cdot u \cdot e^{-\alpha|t|} \quad (1.81)$$

avec


 FIGURE 1.25 – Filtre de dérivation de Deriche avec différents paramètres  $\alpha$ .

$$C_1 = \frac{(1 - e^{-\alpha})^2}{e^{-\alpha}}. \quad (1.82)$$

Par intégration du filtre de dérivation est calculé le filtre de lissage

$$\mathcal{L}(t) = C_0 \cdot (\alpha |t| + 1) e^{-\alpha |t|}. \quad (1.83)$$

avec  $C_0$  choisi de façon à avoir un filtre de lissage normalisé :

$$\sum_{n=-\infty}^{+\infty} \mathcal{L}(n) = 1, \quad (1.84)$$

d'où

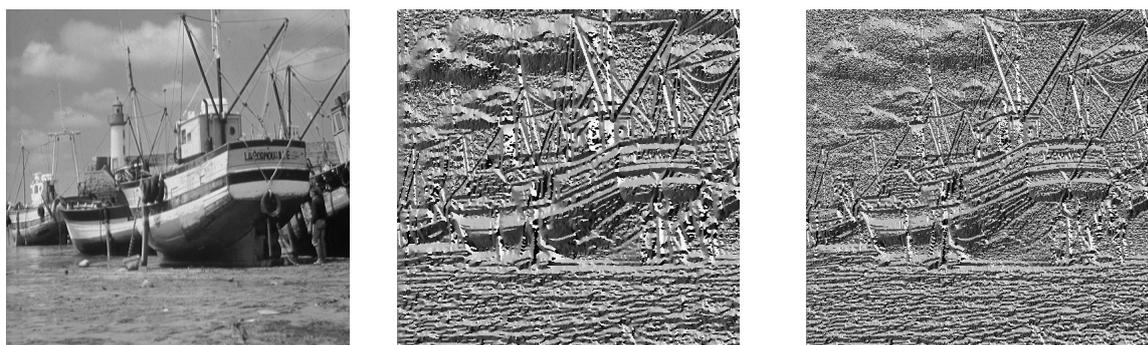
$$C_0 = \frac{(1 - e^{-\alpha})^2}{1 + 2 \cdot \alpha \cdot e^{-\alpha} - e^{-2 \cdot \alpha}}. \quad (1.85)$$

Le facteur  $\alpha$  définit donc la largeur du filtre, le compromis entre la détection et la localisation. Comme illustré en une dimension en figure 1.25, plus  $\alpha$  est grand, plus le filtre est fin. Les contours sont localisés précisément mais cela rend aussi le filtre sensible au bruit. Plus la valeur de  $\alpha$  est petite, moins le filtre est sensible au bruit mais plus les contours sont délocalisés.

L'implémentation récursive des filtres de Deriche est présentée en annexe 1.

La figure 1.26 illustre l'estimation des dérivées en  $x$  et en  $y$  ainsi que celle du gradient et de sa direction sur une image.

Il est intéressant de noter que dans [BGPT02], Bourennane et al. ont généralisé le filtre de Deriche pour les contours de type rampe. Ils obtiennent ainsi un filtre récursif d'ordre 3 qui améliore le filtre de Deriche concernant le rapport signal sur bruit.



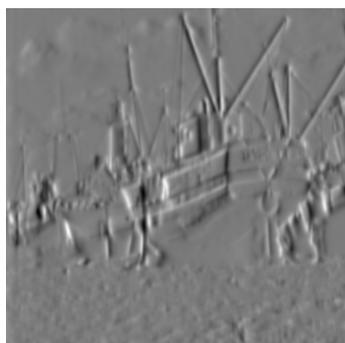
(a) Image Originale

(b)  $\eta$  avec  $\alpha = 1.5$

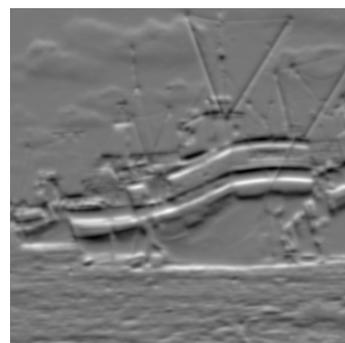
(c)  $\eta$  avec  $\alpha = 3$



(d)  $\|\nabla I\|$  avec  $\alpha = 0.5$ ,



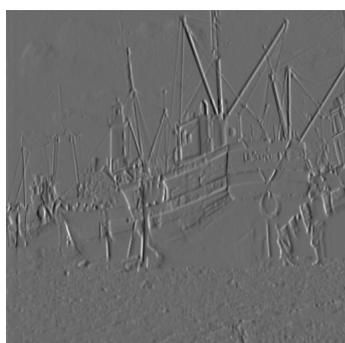
(e) Image  $G_x$  avec  $\alpha = 0.5$ ,



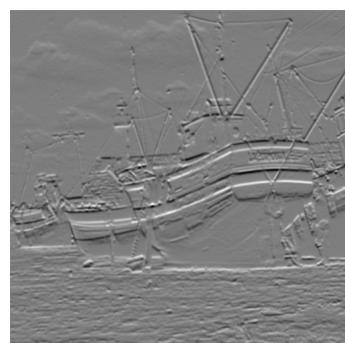
(f) Image  $G_y$  avec  $\alpha = 0.5$ ,



(g)  $\|\nabla I\|$  avec  $\alpha = 1.5$ ,



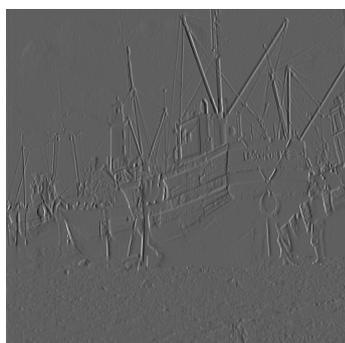
(h) Image  $G_x$  avec  $\alpha = 1.5$ ,



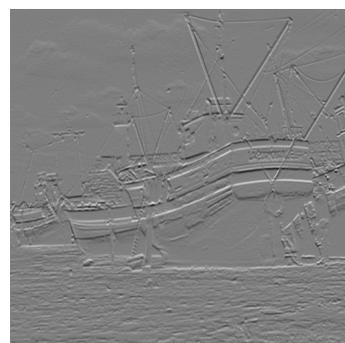
(i) Image  $G_y$  avec  $\alpha = 1.5$ ,



(j)  $\|\nabla I\|$  avec  $\alpha = 3$ ,



(k) Image  $G_x$  avec  $\alpha = 3$ ,



(l) Image  $G_y$  avec  $\alpha = 3$ .

FIGURE 1.26 – Approximation du gradient de Deriche avec différents paramètres  $\alpha$ . Toutes ces images sont normalisées.

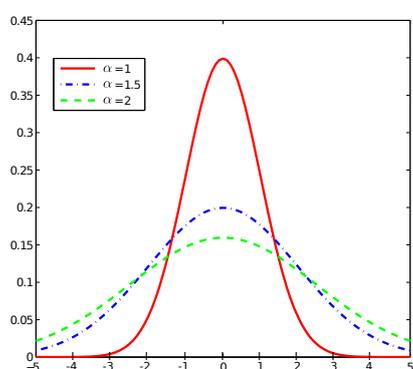
### 6.3 Le Filtre Gaussien

La dérivée première d'une gaussienne constitue l'approximation de Canny [Can86] pour le filtre optimal défini plus haut. Plus généralement, le lissage d'un signal s'effectue par convolution avec un filtre de lissage. Pour obtenir la dérivée, le signal est convolué avec la dérivée première d'un filtre de lissage, il en va de même avec la dérivée seconde.

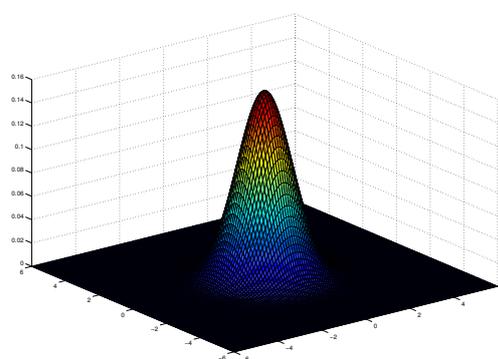
Dans le cas continu, le filtre de lissage gaussien s'écrit :

$$g_0(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (1.86)$$

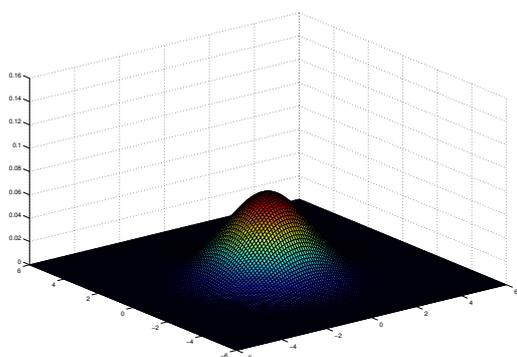
où  $\sigma$  représente l'écart type de la gaussienne.



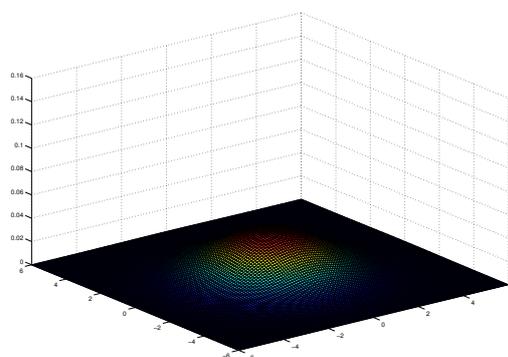
(a)  $g_0(x)$ , différents  $\sigma$



(b)  $G_0(x, y)$  avec  $\sigma = 1$



(c)  $G_0(x, y)$  avec  $\sigma = 1.5$



(d)  $G_0(x, y)$  avec  $\sigma = 2$

FIGURE 1.27 – Fonction gaussienne avec différentes valeurs de  $\sigma$ .

Ce filtre n'est pas récursif donc son implémentation représente une complexité algorithmique élevée, surtout pour une valeur de  $\sigma$  importante, qui correspond à un fort lissage de l'image.

Dans le cas continu, en deux dimensions, ce filtre s'écrit :

$$G_0(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}} \quad (1.87)$$

La figure 1.27 représente la fonction  $G_0(x, y)$  avec différents paramètres  $\sigma$ . Ce filtre présente une propriété d'isotropie en deux dimensions, il fournit des réponses identiques pour les contours obliques et les contours parallèles aux axes de l'image. En effet, en coordonnées polaires, en posant  $r^2 = x^2 + y^2$ , on obtient :

$$G_0(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{r^2}{2\sigma^2}} \quad (1.88)$$



FIGURE 1.28 – Image convoluée avec l'approximation de gaussienne  $h_0$  en utilisant différents écarts types.

La dérivée première  $g_1$  du filtre gaussien  $g_0$  s'écrit :

$$g_1(x) = \frac{-x}{2\pi \cdot \sigma^3} \cdot x \cdot e^{-\frac{x^2}{2\sigma^2}}. \quad (1.89)$$

Dans le cas discret,  $c_1$  correspond à un coefficient de normalisation tel que :

$$\sum_{n=-\infty}^{+\infty} n \cdot g_1(n) \quad \text{et} \quad g_1(x) = c_1 \cdot x \cdot e^{-\frac{x^2}{2\sigma^2}}. \quad (1.90)$$

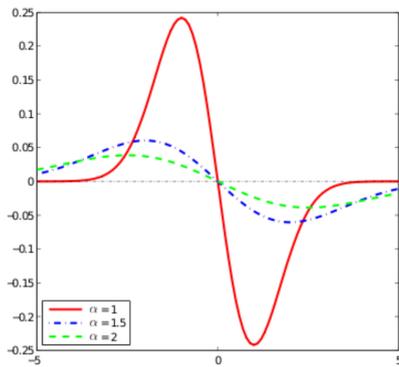
Le produit du filtre de lissage  $g_0$  et celui de dérivation  $g_1$  constitue le filtre de dérivation en deux dimensions (illustré en figure 1.29) :

$$G_1(x, y) = g_1(x) \cdot g_0(y) = C_1 \cdot x \cdot e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (1.91)$$

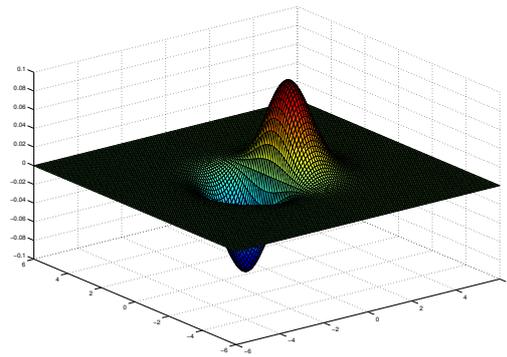
où  $C_1$  représente un coefficient de normalisation, ce qui revient à lisser dans une direction avec une gaussienne et dériver dans la direction perpendiculaire à l'aide de la dérivée d'une gaussienne. Il faut noter que l'écart type  $\sigma$  de la gaussienne et de sa dérivée doivent être identiques pour que le filtre soit isotrope.

Deriche a proposé une implémentation récursive du filtre gaussien et de ses dérivées dans [Der92], elle est présentée en annexe 2. Ce filtre est de la forme :

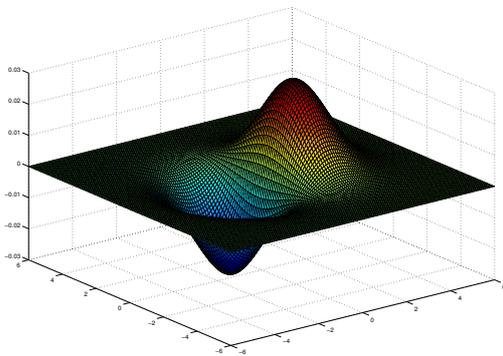
$$h_i(x) = \beta_i \left( \left( a_0 \cos\left(\frac{\omega_0}{\sigma}x\right) + a_1 \sin\left(\frac{\omega_0}{\sigma}x\right) \right) e^{-\frac{b_0}{\sigma}x} + \left( c_0 \cos\left(\frac{\omega_1}{\sigma}x\right) + c_1 \sin\left(\frac{\omega_1}{\sigma}x\right) \right) e^{-\frac{b_1}{\sigma}x} \right) \quad (1.92)$$



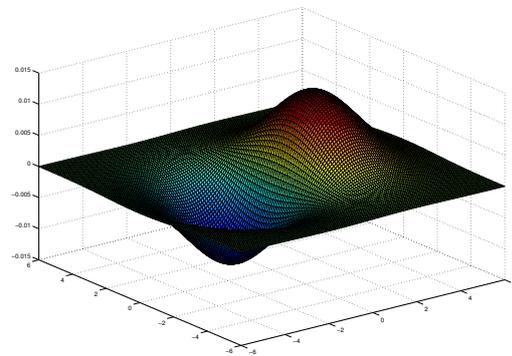
(a)  $g_1(x)$ , différents  $\sigma$



(b)  $G_1(x, y)$  avec  $\sigma = 1$



(c)  $G_1(x, y)$  avec  $\sigma = 1.5$



(d)  $G_1(x, y)$  avec  $\sigma = 2$

FIGURE 1.29 – Dérivées de la gaussienne avec différentes valeurs de  $\sigma$ .

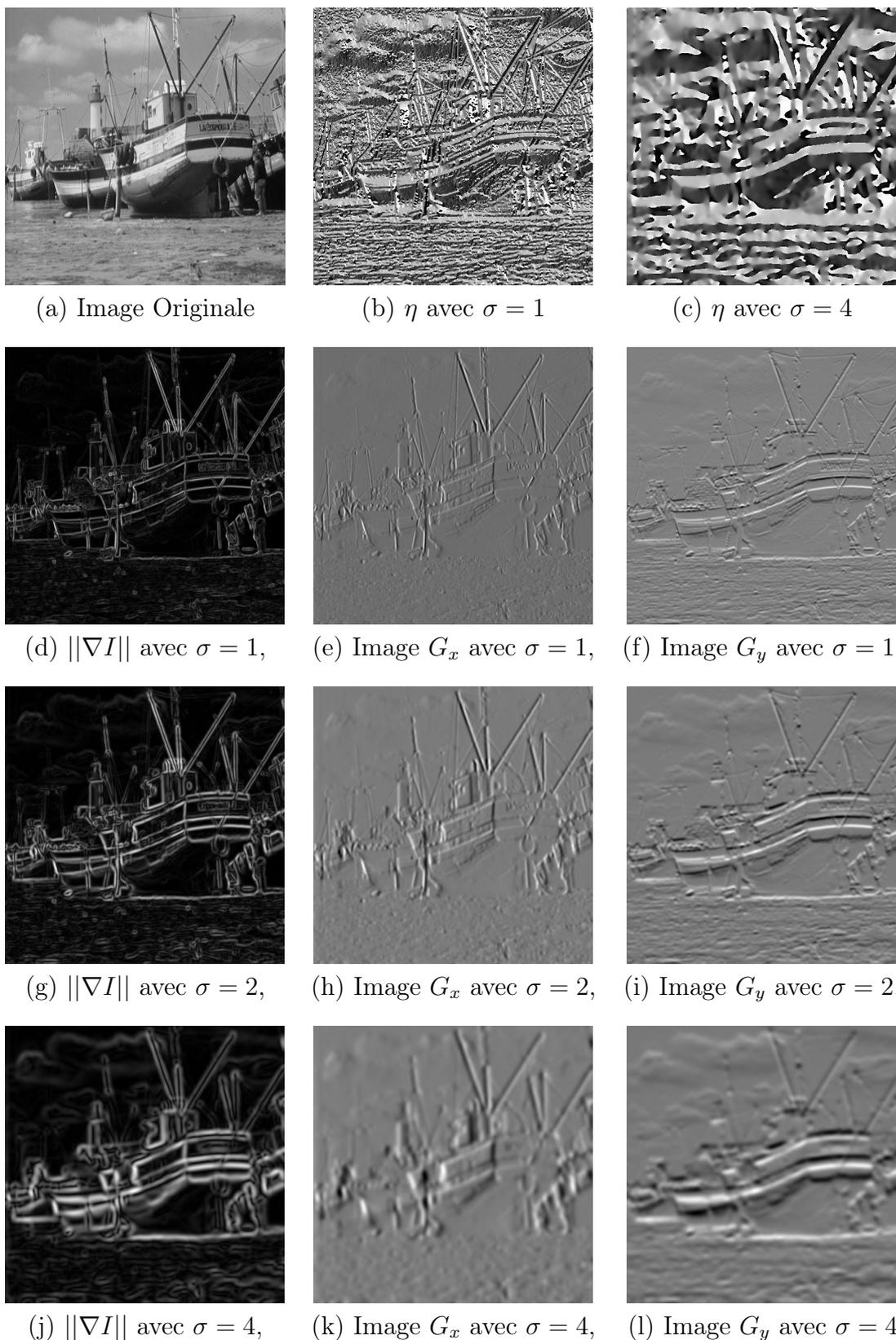


FIGURE 1.30 – Approximation du gradient Gaussien avec différents paramètres  $\sigma$ . Toutes ces images sont normalisées.

où  $i \in \{0, 1, 2\}$  dépend de l'ordre de la dérivée,  $\beta_0 = \frac{0.4}{\sigma}$  représente un coefficient de normalisation pour  $h_0$ , 1 sinon et  $\{a_i, b_i, c_i, \omega_i\}_{i \in \{0,1,2\}} \in \mathbb{R}^{12}$ .

Pour une image  $I$ , l'approximation du gradient  $\|\nabla I\|$  est calculée en convoluant l'image  $I$  telle que  $G_x(x, y) = h_0(x) \cdot h_1(y) * I(x, y)$  et  $G_y = h_0(y) \cdot h_1(x) * I(x, y)$  afin d'obtenir l'estimation des dérivées en  $x$  et en  $y$  de  $I$  appelées respectivement  $G_x$  et  $G_y$ . Puis le gradient est calculé par

$$\|\nabla I(x, y)\| = \sqrt{G_x^2(x, y) * I(x, y) + G_y^2(x, y) * I(x, y)},$$

sa direction associée par  $\eta(x, y) = \arctan\left(\frac{G_y(x,y)*I(x,y)}{G_x(x,y)*I(x,y)}\right)$  pour le pixel de coordonnées  $(x, y)$ . Lisser une image avec l'approximation de la gaussienne revient à convoluer l'image avec  $h_0(x) \cdot h_0(y)$  en chaque pixel. La figure 1.28 illustre l'approximation d'un lissage gaussien dans une image. Des exemples d'estimation de gradients sont disponibles en figure 1.30. Si le paramètre  $\sigma$  n'est pas le même pour le filtre en  $x$  et celui en  $y$ , le détecteur devient anisotrope.

## 6.4 Le Filtre de Shen-Castan

Shen et Castan [SC92] ont proposé un opérateur basé sur les mêmes critères analytiques que ceux proposés par Canny incluant la détection et la localisation. Les filtres obtenus sont assez similaires dans la pratique puisqu'ils sont également basés sur le filtre exponentiel. Comme il n'est pas dérivable à l'origine, il ne se prête pas à une comparaison avec le critère de Canny. Le filtre calcule un lissage optimal par modélisation de la frontière avec un échelon d'amplitude  $A$  noyé dans un bruit blanc stationnaire additif de moyenne nulle. Le filtre de lissage de Shen-Castan s'écrit :

$$F(x) = C_0 \cdot e^{-\alpha|x|} \tag{1.93}$$

avec  $C_0$  choisi de telle façon à obtenir un filtre normalisé dans l'espace discret.

### Calcul des coefficients de normalisation :

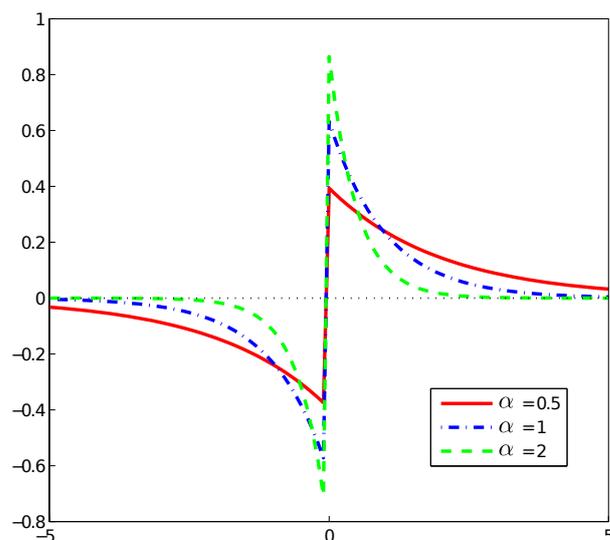
Afin d'obtenir un filtre de lissage normalisé, il faut que le filtre  $F$  vérifie :

$$\sum_{-\infty}^{+\infty} F(n) = 1 \tag{1.94}$$

d'où :

$$C_0 = \frac{1 - e^{-\alpha}}{1 + e^{-\alpha}}.$$

Le filtre de dérivation  $\mathcal{S}$  est calculé par normalisation de la dérivée de  $F$  :


 FIGURE 1.31 – Filtre de dérivation de Shen et Castan avec différents paramètres  $\alpha$ .

$$\mathcal{S}(x) = \begin{cases} C_1 e^{-\alpha x} & \text{si } x > 0 \\ 0 & \text{si } x = 0 \\ -C_1 e^{\alpha x} & \text{si } x < 0 \end{cases} \quad (1.95)$$

$C_1$  est choisi de manière à obtenir un filtre de dérivation  $\mathcal{S}$  normalisé :

$$\sum_{n=-\infty}^{+\infty} n \cdot \mathcal{S}(n) = 1 \quad (1.96)$$

d'où

$$C_1 = \frac{(1 - e^{-\alpha})^2}{2 \cdot e^{-\alpha}}.$$

Le paramètre  $\alpha$  définit la “largeur” du filtre : plus  $\alpha$  est proche de 0, plus le lissage effectué par le filtre est important comme illustré en figure 1.31.

La discontinuité d'ordre 1 au point 0 du filtre de Shen permet d'éviter une délocalisation importante des contours dans l'image lissée, même avec des valeurs faibles de  $\alpha$ . En raison de la forme “pointue” de  $\mathcal{S}$ , il localise très précisément les contours, cependant sa discontinuité en 0 le rend sensible aux bruits.

Les filtres de lissage et de dérivation de Shen et Castan s'implémentent récursivement d'ordre 1 et sont présentés en annexe 3.

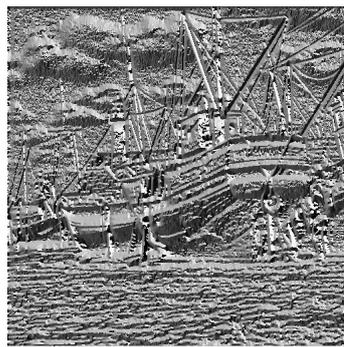
La figure 1.32 illustre l'estimation du gradient sur une image avec différents paramètres  $\alpha$ .

## 6.5 Extraction de lignes de crêtes

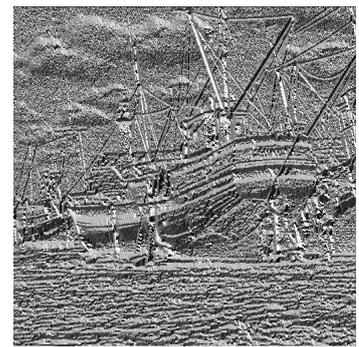
Les crêtes et les vallées représentent les points où les intensités des pixels de l'image  $I$  sont maxima ou respectivement minima dans une direction donnée (illus-



(a) Image Originale



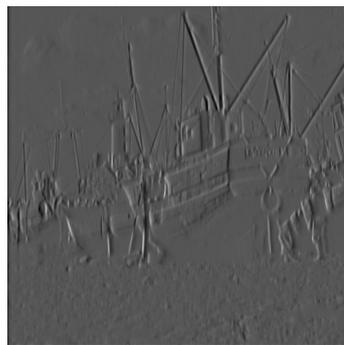
(b)  $\eta$  avec  $\alpha = 1$



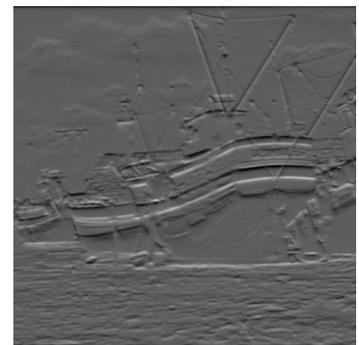
(c)  $\eta$  avec  $\alpha = 2$



(d)  $\|\nabla I\|$  avec  $\alpha = 0.5$ ,



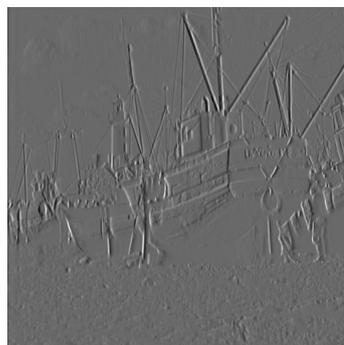
(e) Image  $G_x$  avec  $\alpha = 0.5$ ,



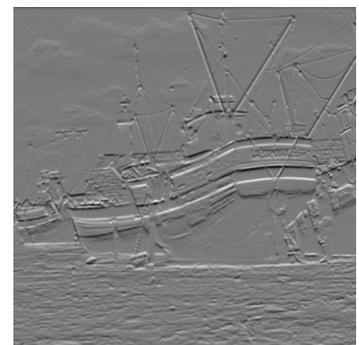
(f) Image  $G_y$  avec  $\alpha = 0.5$ ,



(g)  $\|\nabla I\|$  avec  $\alpha = 1$ ,



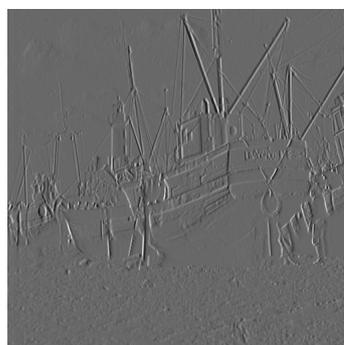
(h) Image  $G_x$  avec  $\alpha = 1$ ,



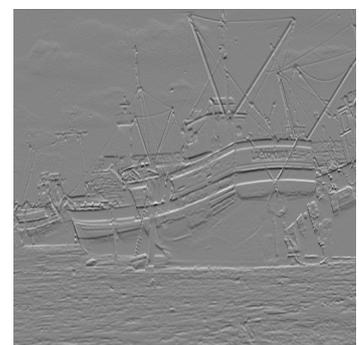
(i) Image  $G_y$  avec  $\alpha = 1$ ,



(j)  $\|\nabla I\|$  avec  $\alpha = 2$ ,



(k) Image  $G_x$  avec  $\alpha = 2$ ,



(l) Image  $G_y$  avec  $\alpha = 2$ .

FIGURE 1.32 – Approximation du gradient de Shen avec différents paramètres  $\alpha$ . Toutes ces images sont normalisées.

tration d'une crête et d'une vallée en figure 1.33). Dans le cas idéal (figure 1.4), les crêtes de type pointe peuvent être modélisées par la fonction porte d'équation :

$$\Pi(t) = \begin{cases} 1 & \text{si } |t| \geq 0 \\ 0 & \text{si } |t| < 0 \end{cases} \quad \text{avec } a > 0, \quad (1.97)$$

celles de type toit par la fonction triangle :

$$T(t) = \begin{cases} -\frac{t}{a_1} + 1 & \text{si } a_1 \geq t \geq 0 \\ -\frac{t}{a_2} + 1 & \text{si } 0 \geq t \geq a_2 \\ 0 & \text{sinon,} \end{cases} \quad (1.98)$$

ou encore, en tenant compte d'un certain flou, par :

$$C(t) = e^{-a \cdot t} \quad \text{avec } a > 0. \quad (1.99)$$

Les détecteurs de contours classiques ou anisotropes ne sont pas adaptés à la détection de lignes de crêtes puisqu'il calculent la pente des contours et vont donc détecter deux contours de part et d'autre d'une crête ou d'une vallée comme illustré en figure 1.33.

En convoluant l'image avec la dérivée seconde de la gaussienne en deux dimensions, puis en supprimant les non maxima dans la direction qui maximise la variance permet de détecter les lignes de crêtes [Zio91]. Cependant, cette approche est très sensible au bruit dans les images, créant ainsi de fausses détections.

Comme décrit dans la partie 4.5, l'endomorphisme de Weingarten et sa matrice  $W$  peuvent être utilisés pour extraire les lignes de crêtes. En effet, les courbures principales et les directions principales correspondent respectivement aux valeurs propres  $k_1$  et  $k_2$  et aux vecteurs propres  $\vec{v}_1$  et  $\vec{v}_2$  de  $W$  (voir équation 1.51). La ligne de crête correspond au maximum de la courbure maximale dans la direction de la courbe maximale.

L'extraction des lignes de crêtes peut être décomposée en différentes étapes :

1. Calcul des dérivées partielles  $I_x$ ,  $I_y$ ,  $I_{xx}$ ,  $I_{yy}$  et  $I_{xy}$  jusqu'à l'ordre 2 de l'image  $I$ .

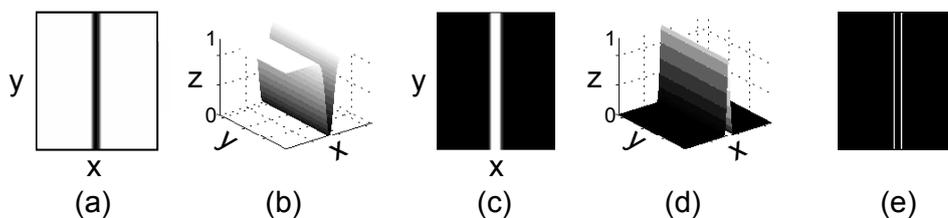


FIGURE 1.33 – Crête et vallée dans une image. (a) Une vallée dans une image. (b) Représentation de la surface d'une vallée. (c) Crête dans une image. (d) Représentation de la surface d'une crête. (e) Détection de contour de l'image (a) ou (c). Pour (b) et (d), l'axe des  $z$  correspond à l'intensité lumineuse des pixels.

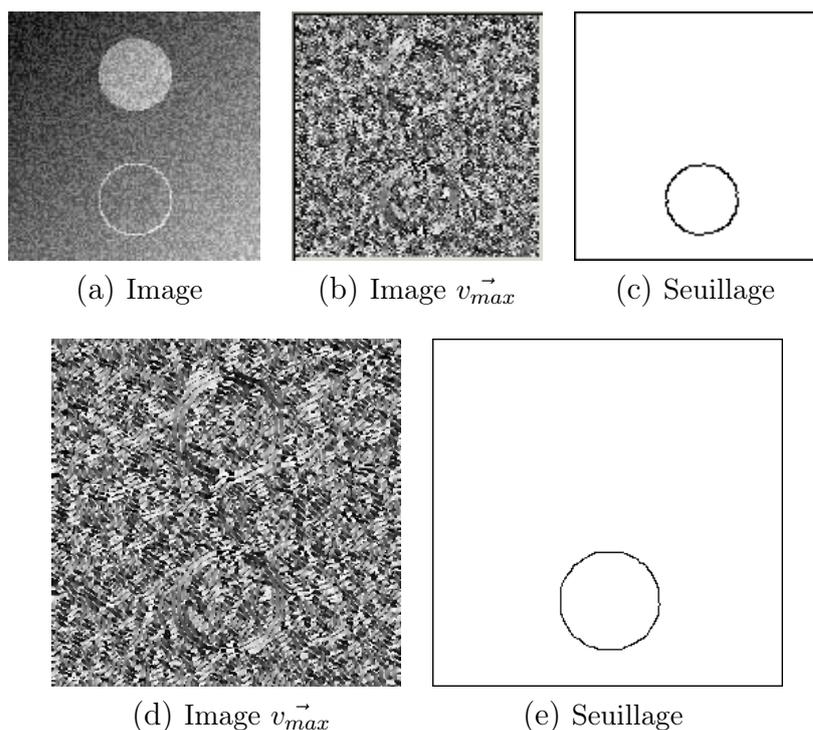


FIGURE 1.34 – Exemple d’extraction de lignes de crêtes sur une image avec un bruit gaussien additif  $128 \times 128$ . Pour (c) et (e) :  $\sigma = 1$ ,  $\tau_b = 0.6$  et  $\tau_h = 0.9$  Extraction sous pixellique de lignes de crêtes.

2. Estimation des deux courbures principales  $k_1$  et  $k_2$  et des deux directions principales de courbure  $\vec{v}_1$  et  $\vec{v}_2$  (équation 1.52).
3. Détermination des points où la valeur absolue de la courbure maximum

$$k_{max}(x, y) = \max(|k_1(x, y)|, |k_2(x, y)|)$$

en un pixel de coordonnées  $(x, y)$  est maximale dans la direction principale correspondante  $v_{max}$  :

$$v_{max}(x, y) = \arg \max_{\vec{v}_1(x, y), \vec{v}_2(x, y)} (|k_1(x, y)|, |k_2(x, y)|).$$

Les dérivées de l’images peuvent être calculées en utilisant les approximations des dérivées de la gaussienne [AMM97] [Ste98] [Lin98]. L’écart type de la gaussienne  $\sigma$  permet d’adapter la détection en fonction de la largeur des lignes de crêtes désirées. Une analyse multi-échelle [AMM97] [Ste98] [Lin98] permet d’extraire les lignes de crêtes de différentes épaisseurs en itérant en chaque pixel de l’espace échelle. Montesinos et Datteny [MD97b] ont développé une approche de détection de lignes de crêtes sous pixellique, qui rend la détection plus robuste. La figure 1.34 montre un résultat de l’extraction de crête dans une image bruitée en utilisant les dérivées secondes de l’image [AMM97] ainsi que la méthode sous-pixellique [MD97b] qui obtient un résultat plus précis.

Dans [MAM95] [HM95], les auteurs ont utilisé les dérivées troisièmes  $I_{xxx}$ ,  $I_{yyy}$ ,  $I_{xxy}$  et  $I_{xyy}$  pour obtenir un meilleur résultat. Cependant, utiliser les dérivées d'ordre 3 complique la segmentation ainsi que le filtrage et rend la méthode plus instable.

## 7 Autres approches et filtres non linéaires

Il existe également des opérateurs non linéaires pour réduire le bruit et la détection de contours dans les images. Nous en décrivons quelques uns qui sont régulièrement utilisés dans le traitement des images.

### 7.1 Lissage et régularisation

#### Le filtre median

Le filtre median est un filtre de lissage non linéaire qui consiste simplement à remplacer un pixel par la valeur médiane de ses voisins définie sur la fenêtre de taille  $n \times n$  centrée sur le pixel ( $n \in \mathbb{N}$ ) [BRLM90]. Ce filtre régularise l'image en supprimant les petits détails tout en préservant les contours de type marche.

#### Le filtre bilatéral

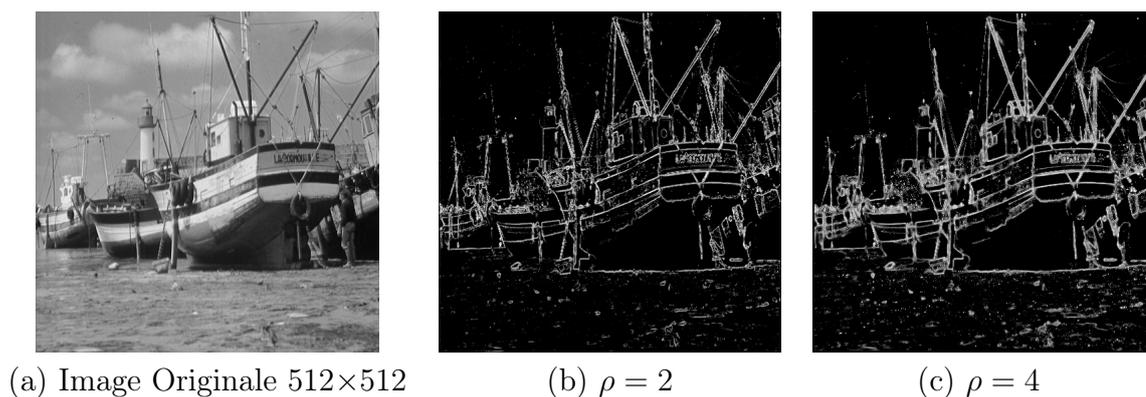
Le filtre bilatéral est une méthode bien connue en restauration des images [TM98] [PKT09]. Il permet de filtrer le bruit dans les zones d'intensités uniformes, tout en préservant les principales structures des objets. Sa formulation simple contribue à sa popularité. Il est défini comme une moyenne pondérée des pixels voisins dans un masque carré centré sur un pixel. Le filtre bilatéral prend en compte la différence de valeur du pixel central avec ses voisins pour préserver les bords tout en lissant. L'idée est alors de combiner des informations d'intensité en se basant à la fois sur leur proximité spatiale et sur leur similarité photométrique. En effet, pour que l'intensité d'un pixel puisse en influencer une autre, le pixel ne doit pas seulement se situer proche du pixel central, mais aussi avoir une valeur similaire. L'intensité d'un pixel filtré est remplacée par la moyenne des combinaisons non-linéaires des intensités des pixels dans un masque carré, centré sur le pixel traité et de taille  $\mathcal{S} = (n+1) \times (n+1)$  ( $n \in \mathbb{N}$ ). Les poids utilisés dans le calcul de la moyenne varient en fonction des similarités photométriques et de la position des pixels dans le masque par rapport au pixel courant. Cela permet un filtrage local sélectif qui propage l'information d'intensité sans traverser les discontinuités. La version la plus populaire utilise des poids gaussiens, comme dans l'expression discrétisée suivante :

$$\mathcal{B}\{I(p)\} = \frac{1}{\mathcal{W}(p)} \cdot \sum_{q \in \mathcal{S}} G_{\sigma_s}(\|p - q\|) \cdot G_{\sigma_r}(|I(p) - I(q)|) \cdot I(q), \quad (1.100)$$

avec  $\mathcal{W}(p)$  coefficient de normalisation tel que :

$$\mathcal{W}\{p\} = \sum_{q \in \mathcal{S}} G_{\sigma_s}(\|p - q\|) \cdot G_{\sigma_r}(|I(p) - I(q)|). \quad (1.101)$$

où :


 FIGURE 1.35 – Réponses  $R(\vec{r}_0)$  de l’algorithme de SUSAN avec  $t = 20$ .

- $I(p)$  représente l’intensité d’un pixel de coordonnées  $p$  ( $p \in \mathcal{S}$ ),
- $\sigma_s$  et  $\sigma_r$  sont les écarts types respectivement des gaussiennes  $G_{\sigma_s}$  et  $G_{\sigma_r}$ . Le paramètre  $\sigma_s$  est fixé en fonction de la taille du masque et  $\sigma_r$  définit la similarité photométrique (il évolue en fonction du niveau du bruit, plus la valeur de  $\sigma_r$  est importante, plus le filtre est passe bas),
- $\|p - q\|$  est la distance euclidienne entre les pixels de coordonnées  $p$  et  $q$ ,
- $\mathcal{S}$  représente le domaine spatial de forme carré des positions  $q$  possibles.

Pour les images en couleur, la distance euclidienne des intensités lumineuses des différents canaux n’est pas une bonne estimation de la couleur perçue. Le filtre bilatéral permet de combiner plusieurs bandes de couleur appropriées. Il mesure des distances euclidiennes photométriques entre les pixels dans un autre espace couleur [TM98].

Nous avons comparé nos résultats de régularisation des images notamment avec ceux obtenus avec ce filtre.

## 7.2 Détecteur de contours non linéaires

### Le détecteur de contours SUSAN

L’algorithme de SUSAN (*Smallest Univalve Segment Assimilating Nucleus*) ne nécessite aucune dérivée [SB97]. En effet, pour un masque circulaire de rayon  $\rho$  centré sur un pixel  $\vec{r}_0$ , l’intensité du pixel central du masque appelé *nucleus* permet de définir une zone où les pixels ont la même intensité que  $\vec{r}_0$ . Cette zone nommée USAN (*Univalve Segment Assimilating Nucleus*) permet de reconnaître les contours et les coins, les coins étant détectés plus fortement que les contours.

La zone USAN est maximale lorsque le masque circulaire est situé dans une région homogène. La taille de l’USAN diminue lorsque le masque s’approche des contours. Les contours correspondent à une zone USAN moins importante qu’au niveau d’une zone homogène. Au niveau des coins, la zone USAN est encore moins importante, ce qui correspond à un minimum local.

Le nombre de pixels de même intensité que le *nucleus* situés à l’intérieur du masque circulaire de centré au point  $\vec{r}_0 = (x_0, y_0)$  est défini par :

$$n(\vec{r}_0) = \sum_{\vec{r}} e^{\left(\frac{I(\vec{r})-I(\vec{r}_0)}{t}\right)^6} \quad (1.102)$$

avec  $\vec{r} = (x, y)$  représentant la position de n'importe quel autre point que le *nucleus* du masque circulaire.  $I(\vec{r}_0)$  correspond à l'intensité du pixel considéré.  $t$  est un seuil sur l'intensité des pixels correspondant au minimum de contraste de détermination des contours.

Ensuite,  $n(\vec{r}_0)$  est comparé avec le seuil géométrique appelé  $g$  pour réponse initiale des contours  $R(\vec{r}_0)$  :

$$R(\vec{r}_0) = \begin{cases} g - n(\vec{r}_0) & \text{si } n(\vec{r}_0) < g \\ 0 & \text{ailleurs.} \end{cases} \quad (1.103)$$

En effet, si le *nucleus* est positionné dans un coin alors la taille de la zone USAN sera moins de deux fois plus petite que la surface du masque circulaire et sera un minimum local.

Afin de déterminer les maxima locaux, les directions des contours sont estimés par les moments de USAN où rapport  $\frac{\overline{(y - y_0)^2}(\vec{r}_0)}{\overline{(x - x_0)^2}(\vec{r}_0)}$  estime l'orientation du contour, avec :

$$\begin{cases} \overline{(x - x_0)^2}(\vec{r}_0) = \sum_{\vec{r}} (x - x_0)^2 \cdot e^{\left(\frac{I(\vec{r})-I(\vec{r}_0)}{t}\right)^6} \\ \overline{(y - y_0)^2}(\vec{r}_0) = \sum_{\vec{r}} (y - y_0)^2 \cdot e^{\left(\frac{I(\vec{r})-I(\vec{r}_0)}{t}\right)^6} \end{cases} \quad (1.104)$$

## Le gradient morphologique

La morphologie mathématique représente un cadre intéressant pour la détection de contours [Mat75] [HSZ87]. Le gradient morphologique est calculé par la différence pixel à pixel des niveaux de gris de l'image originale  $I$  transformée par une dilatation d'une part et par une érosion d'autre part.

Soit  $S(x, y)$  un élément structurant de l'image centré sur un pixel  $I(x, y)$  de coordonnées  $(x, y)$ . L'élément structurant peut être un carré de taille  $n \times n$  ( $n \in \mathbb{N}^+$ ) ou un losange, un cercle... L'érosion  $E(x, y)$  permet de calculer la valeur minimale d'intensité pour tous les pixels à l'intérieur de  $S$  et correspond à l'opération :

$$E(x, y) = \text{Min}\{I(a, b) \mid (a, b) \in S(x, y)\}.$$

Au contraire, la dilatation  $D(x, y)$  estime la valeur maximale d'intensité pour tous les pixels à l'intérieur de  $S$  et correspond à l'opération :

$$D(x, y) = \text{Max}\{I(a, b) \mid (a, b) \in S(x, y)\}.$$

La gradient morphologique estimé revient à :

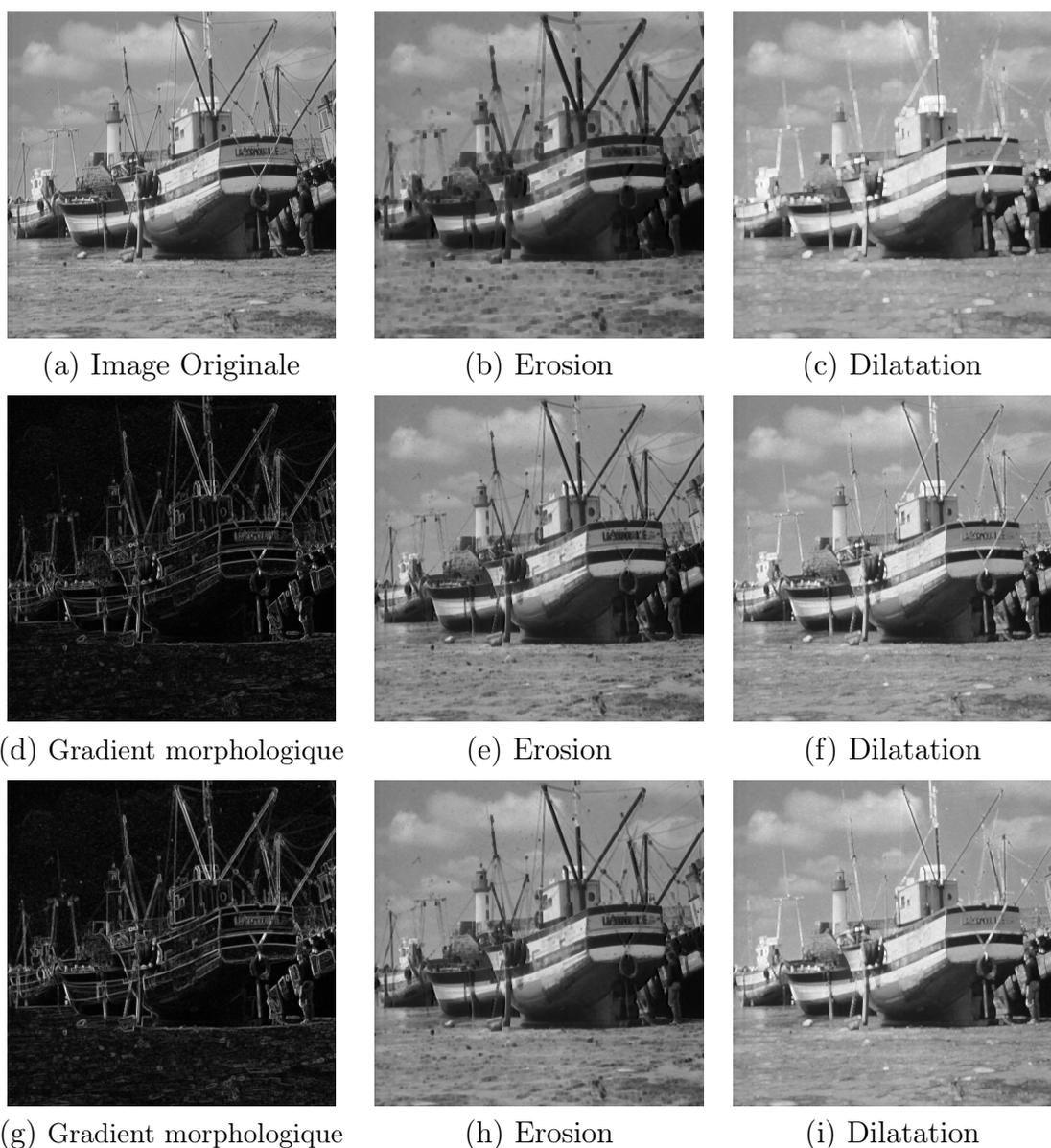


FIGURE 1.36 – Estimation du gradient morphologique. Un élément structurant carré  $5 \times 5$  est utilisé pour (b) et (c). Le gradient estimé en (d) à partir de (e) et (f) est calculé à partir d'un structurant à 4 connexités. Celui estimé en (g) à partir de (h) et (i) est calculé à partir d'un structurant carré  $3 \times 3$ . Le résultat présenté en (c) est moins perturbé par la texture mais l'élément structurant à 4 connexités détecte moins bien les contours obliques que le carré. Les images (d) et (g) sont normalisées.

$$\|\nabla I(x, y)\| = \frac{1}{2} (D(x, y) - E(x, y)). \quad (1.105)$$

La taille des contours ainsi estimés dépend de la largeur de l'élément structurant ainsi utilisé. Cependant, le gradient morphologique n'a pas de direction associée à chaque pixel, donc il est impossible d'extraire les maxima dans la direction du gradient. Concernant les images texturées ou contenant du bruit, les fonctions  $E(x, y)$  et  $D(x, y)$  vont dilater ou éroder des points indésirables, ce qui perturbera l'estima-

tion du gradient. Plus l'élément structurant  $S$  est large, plus les points indésirables vont être mis en valeur. Les figures 1.36 (b) et (c) illustrent une dilatation ainsi qu'une érosion avec un élément structurant carré de taille  $5 \times 5$ , les petits objets ressortent nettement. Donc, si  $S$  est trop grand, les contours estimés vont être larges et les petits objets ainsi que le bruit seront amplifiés.

## 8 Les Espaces Echelle et Equations Différentielles dans les Images

### 8.1 Introduction

La régularisation des images est un problème majeur en vision par ordinateur notamment pour la détection de contours et le débruitage. Dans la littérature, il existe un nombre non exhaustif de méthodes de régularisation d'images, parmi lesquelles les approches de débruitage par moyennes (filtre médian [BRLM90] ou bilatéral [TM98]), les filtres à fenêtres adaptatives [KHEK76] [NM79], les filtres de moyennes non locales (connu sous le nom de *Non-Local means* [BCM<sup>+</sup>06]) ou encore les méthodes utilisant des dictionnaires de patches [MBP<sup>+</sup>09]. Quelques résultats de débruitages sur une image réelle sont présentés en figure 1.37.

Aussi, les équations aux dérivées partielles ont développé un fort intérêt pour résoudre ce problème [AK06] [Ter04] [LDA<sup>+</sup>97] [DF95]. En effet, les EDP permettent d'appliquer un lissage important aux images tout en ne gardant que les principales

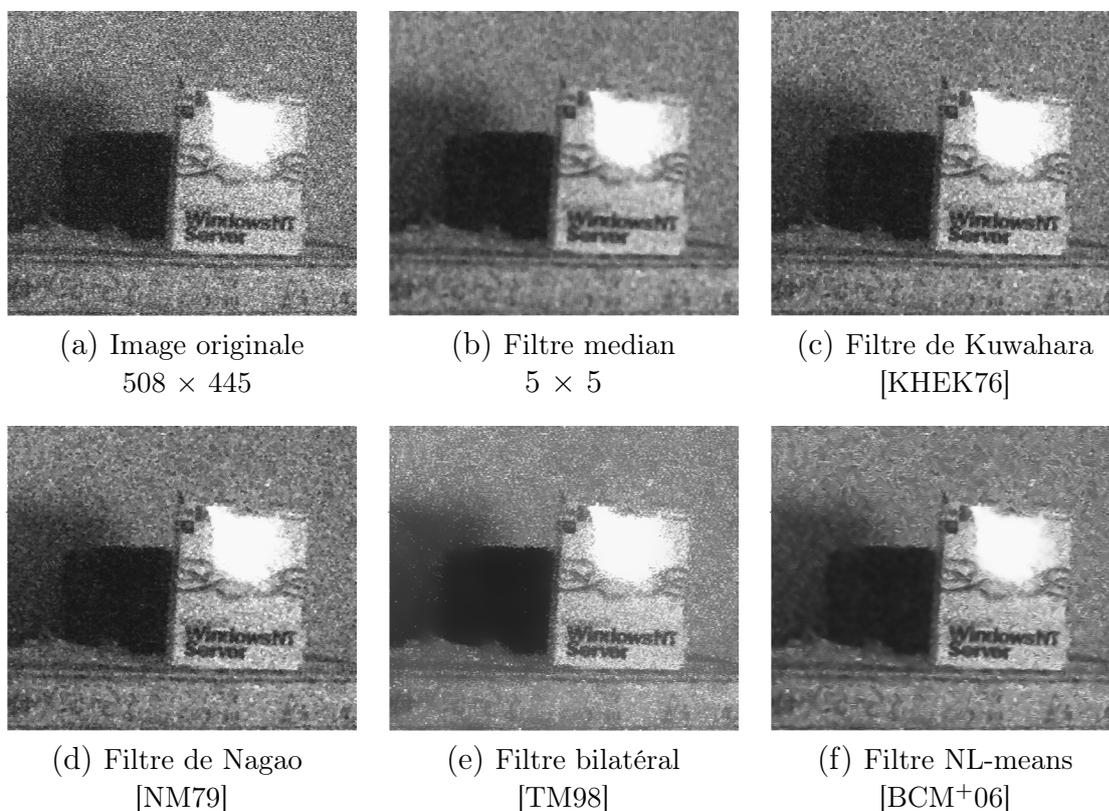


FIGURE 1.37 – Quelques résultats de méthodes de débruitages sur une image réelle.

structures telles que les contours ou les coins. L'idée est d'introduire une variable additionnelle  $t$  liée au degré de lissage qui peut être interprétée comme une valeur de temps. Plus  $t$  est grand, plus le lissage sera fort, il faut donc limiter la valeur de  $t$  en fonction du niveau du bruit dans l'image sinon la majorité des structures seront difusées.

Afin de régulariser une image scalaire  $I$ , le modèle général d'évolution de l'image est de la forme :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) &= F \left( I_0, \frac{\partial I}{\partial x}(x, y, t), \frac{\partial I}{\partial y}(x, y, t), \frac{\partial^2 I}{\partial x^2}(x, y, t), \frac{\partial^2 I}{\partial y^2}(x, y, t), \frac{\partial^2 I}{\partial x \partial y}(x, y, t) \right) \\ I(x, y, 0) &= I_0(x, y) \end{cases}$$

où  $I_0$  représente l'image originale et  $F$  une fonction de contrôle de la diffusion dépendant de  $I_0$  et de ses dérivées spatiales du premier et second ordre [CM98].

## 8.2 Espace échelle linéaire

Une théorie de la représentation multi-échelle est de représenter les signaux à de multiples échelles. Les structures fines sont ainsi successivement supprimées en associant un paramètre d'échelle à chaque niveau de la représentation multi-échelle.

Witkin [Wit83] a proposé la première représentation de l'espace échelle (*scale-space*) d'une fonction. Pour un signal  $s : \mathbb{R} \rightarrow \mathbb{R}$  donné, une représentation linéaire de l'espace échelle de  $s$  est un autre signal  $L$  :

$$\begin{aligned} L : \mathbb{R}^+ \times \mathbb{R} &\rightarrow \mathbb{R} \\ (t, x) &\mapsto h(x, t) * s(x) \end{aligned} \tag{1.106}$$

avec  $h : \mathbb{R}^+ \times \mathbb{R} \rightarrow \mathbb{R}$  un noyau de convolution, où  $*$  représente le produit de convolution.

Il est essentiel pour la famille de l'espace échelle que la représentation à une échelle grossière constitue une simplification des représentations du signal à des échelles plus fines. Witkin préconise une convolution de  $s$  par une gaussienne  $g_\sigma$ , les extrema locaux des signaux en une dimension s'amenuisent au cours d'une augmentation de la valeur de l'écart type de la gaussienne  $\sigma$  [Wit83].

En deux dimensions, afin d'obtenir des images à échelle de plus en plus haute, il suffit de convoluer l'image avec un noyau gaussien d'écart type croissant [Koe84] [Lin93]. En notant  $t$  le facteur d'échelle tel que  $\sigma = \sqrt{2t}$ , la fonction  $g_{\sqrt{2t}} : \mathbb{R}^2 \rightarrow \mathbb{R}$  s'écrit :

$$g_{\sqrt{2t}}(x, y) = \frac{1}{4\pi t} e^{-\frac{x^2+y^2}{4t}}. \tag{1.107}$$

Il a été remarqué par Koenderink [Koe84] l'équivalence entre la solution de l'EDP de diffusion de la chaleur [Bre83] pour un temps  $t$  et la convolution de l'image  $I$  avec un noyau gaussien d'écart type  $\sqrt{2t}$ . L'équation de la chaleur est une équation de diffusion isotrope qui peut se mettre sous la forme parabolique linéaire suivante :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) &= \frac{\partial^2 I}{\partial x^2}(x, y, t) + \frac{\partial^2 I}{\partial y^2}(x, y, t) \\ I(x, y, 0) &= I_0(x, y) \end{cases} \tag{1.108}$$

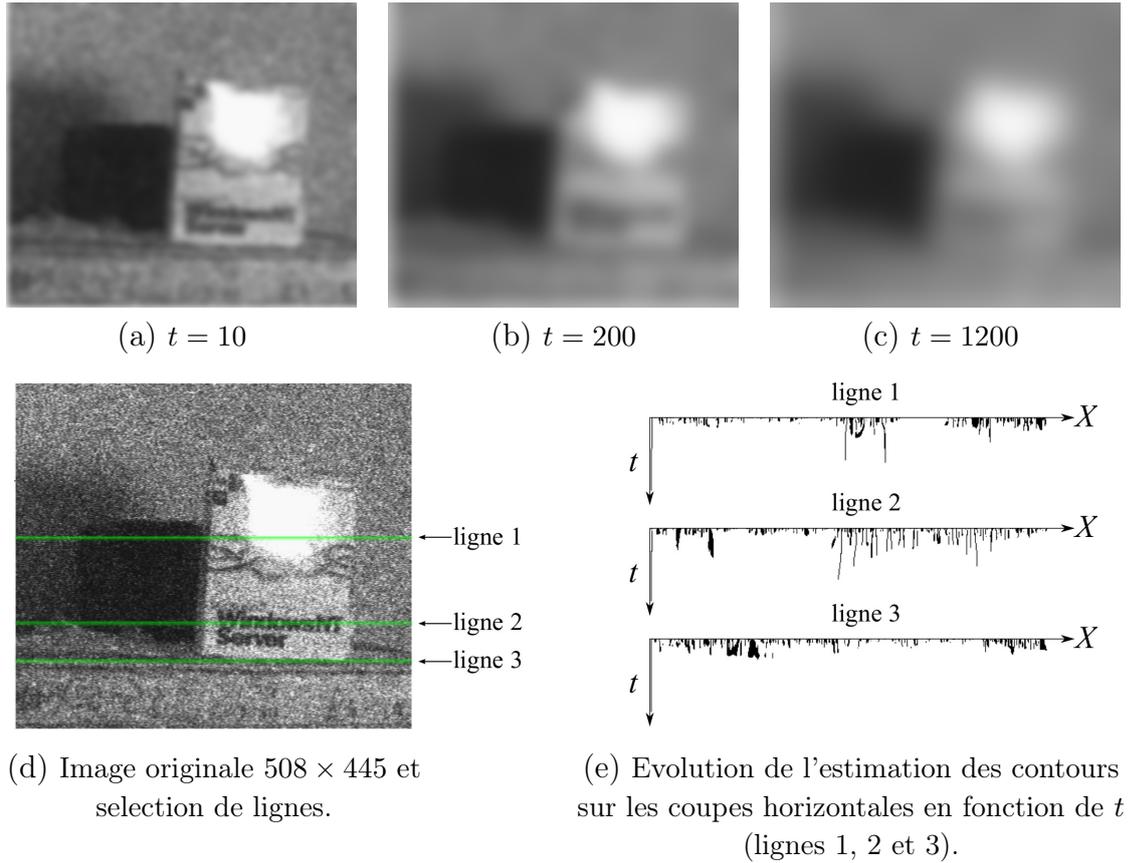


FIGURE 1.38 – Diffusion isotropique dans une image et évolution de la détection de contours. En (e),  $0.5 < t < 50$ , lorsque les traits ne sont pas fins, cela correspond à des contours horizontaux.

cette équation peut être également écrite sous la forme divergence :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = \operatorname{div}(\nabla(I(x, y, t))) \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.109)$$

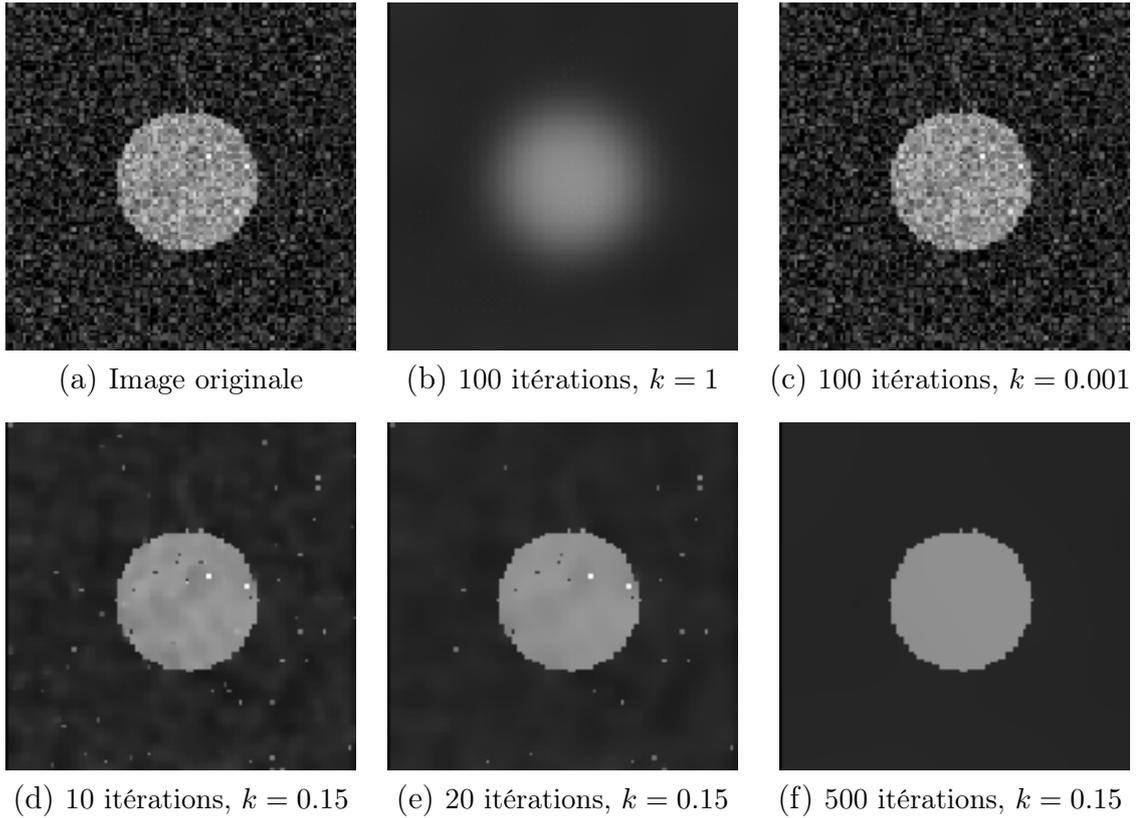
ou encore en utilisant le laplacien de l'image  $\Delta I(x, y, t)$  :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = \Delta I(x, y, t) \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.110)$$

où  $\Delta I = I_{xx} + I_{yy} = I_{\xi\xi} + I_{\eta\eta}$  (voir partie 4.2 de ce chapitre).

Ce schéma de diffusion permet de régulariser les images et supprimer le bruit. Cependant, comme illustré en figure 1.38, la diffusion par la convolution de l'image avec une gaussienne d'écart type trop important mélange les valeurs des pixels, ce qui détruit également les principales structures telles que les contours ou les lignes de crêtes.

Dans la suite de ce manuscrit, nous allons alléger la notation à l'échelle  $t$  en appelant,  $I = I(x, y, t)$  et  $\|\nabla I\| = \|\nabla I(x, y, t)\|$ .


 FIGURE 1.39 – Diffusion de Perona-Malik avec la fonction  $c_1$ .

### 8.3 Le modèle de Perona-Malik

Afin de résoudre les problèmes liés à la diffusion isotrope, les premières idées en diffusion anisotrope ont été introduites par Perona et Malik [PM90]. Contrairement au lissage classique isotrope, ils introduisent la détection des contours dans le filtrage lui-même. Ils proposent une diffusion forte dans les régions à faible gradient (régions homogènes) et une diffusion faible au voisinage des contours. En ré-écrivant l'équation de la chaleur (équation 1.109), Malik et Perona introduisent dans l'opérateur divergence une fonction  $c(\cdot)$  décroissante de la norme du gradient  $\|\nabla I(x, y, t)\|$  et d'un paramètre  $k$  ( $k > 0$ ) qui va atténuer la diffusion au voisinage des contours. Deux fonctions  $c(\cdot)$  appelées  $c_1(\cdot)$  et  $c_2(\cdot)$  [PM90] ont été proposées initialement :

$$c_1(\|\nabla I\|) = e^{(-\frac{\|\nabla I\|}{k})^2} \quad (1.111)$$

ou encore

$$c_2(\|\nabla I\|) = \frac{1}{1 + \left(\frac{\|\nabla I\|}{k}\right)^2} \quad (1.112)$$

L'espace échelle qui formalise cette idée est alors décrit par l'EDP suivante :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = \operatorname{div}(c(\|\nabla I\|) \nabla I) \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.113)$$

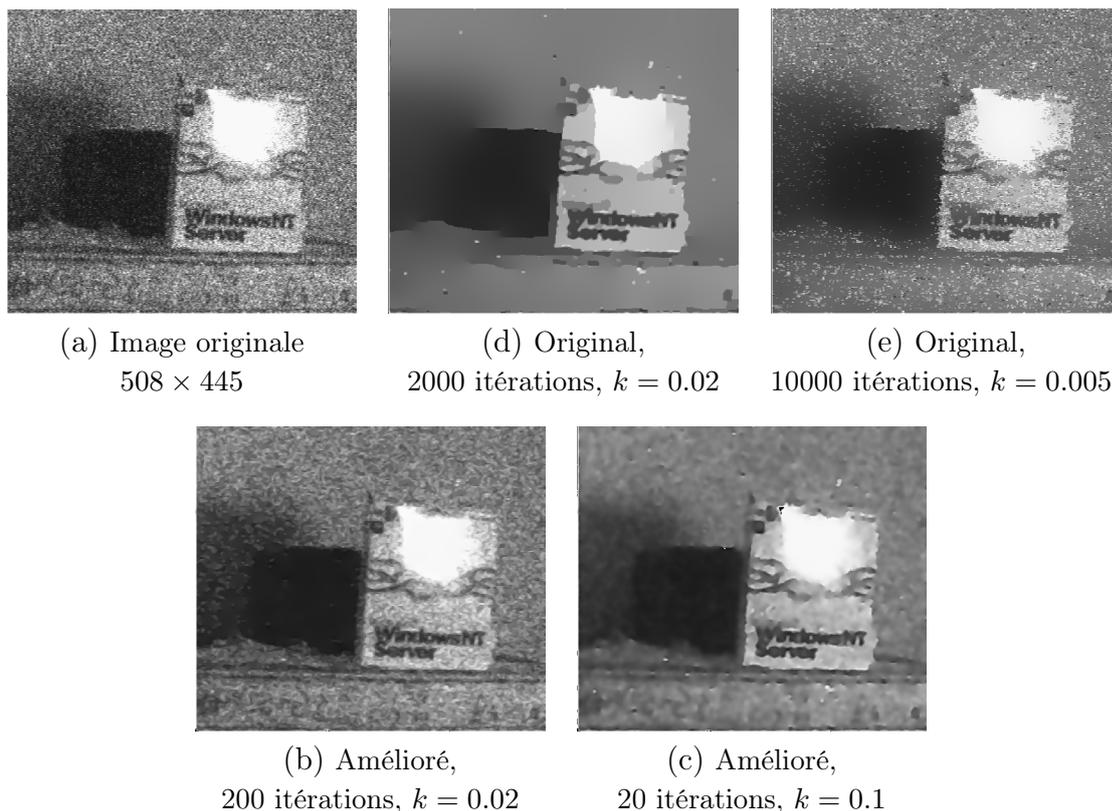


FIGURE 1.40 – Comparaison entre l’algorithme original de Perona-Malik et la version améliorée avec l’estimation d’un gradient gaussien ( $\sigma = 1$ ), ce qui permet de conserver les contours importants.

En développant l’opérateur divergence en terme de dérivées secondes directionnelles, en notant  $c'(\cdot)$  la dérivée de la fonction  $c(\cdot)$ , nous obtenons :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = (c(\|\nabla I\|) + \|\nabla I\| \cdot c'(\|\nabla I\|)) I_{\eta\eta} + c(\|\nabla I\|) I_{\xi\xi} \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.114)$$

où  $\eta$  représente la direction du gradient et  $\xi$  la direction perpendiculaire au gradient, i.e. la direction tangente au contour. Les images  $I_{\eta\eta}$  et  $I_{\xi\xi}$  désignent respectivement les dérivées de l’image suivant  $\eta$  et  $\xi$ .

En utilisant la fonction  $c_1$ , l’équation 1.114 peut également s’écrire sous la forme :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = c_\xi I_{\xi\xi} + c_\eta I_{\eta\eta} \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.115)$$

avec :

$$\begin{cases} c_\xi = e\left(-\frac{\|\nabla I\|^2}{k^2}\right) \\ c_\eta = \frac{(k^2 - 2\|\nabla I\|^2)}{k^2} \cdot e\left(-\frac{\|\nabla I\|^2}{k^2}\right) \end{cases}$$

L’équation de diffusion suit les comportements suivants :

- lorsque le paramètre  $k$  est proche de zéro, la diffusion est inhibée, en revanche lorsque  $k$  augmente, la diffusion tend à devenir isotrope (ce comportement est identique pour chacune des deux fonctions proposées  $c_1(\cdot)$  et  $c_2(\cdot)$ ).
- pour des autres valeurs du paramètre  $k$ , chaque pixel de l'image va être lissé anisotropiquement et avec des coefficients de lissage différents :
  - Si  $\|\nabla I\|$  est faible, alors la diffusion est isotrope.
  - Si  $\|\nabla I\|$  est fort, alors la diffusion est inhibée.

L'équation 1.113 s'écrit sous la forme discrète avec  $I_{i,j}^t$  le pixel de l'image de coordonnées  $(i, j)$  au temps de diffusion  $t$  :

$$\begin{aligned}
 I_{i,j}^{t+1} = & I_{i,j}^t \\
 & + \delta_t \left( c \left( |I_{i-1,j}^t - I_{i,j}^t| \right) \cdot |I_{i-1,j}^t - I_{i,j}^t| \right) \\
 & + \delta_t \left( c \left( |I_{i+1,j}^t - I_{i,j}^t| \right) \cdot |I_{i+1,j}^t - I_{i,j}^t| \right) \\
 & + \delta_t \left( c \left( |I_{i,j-1}^t - I_{i,j}^t| \right) \cdot |I_{i,j-1}^t - I_{i,j}^t| \right) \\
 & + \delta_t \left( c \left( |I_{i,j+1}^t - I_{i,j}^t| \right) \cdot |I_{i,j+1}^t - I_{i,j}^t| \right)
 \end{aligned} \tag{1.116}$$

où  $0 \leq \delta_t \leq \frac{1}{4}$  pour rendre le processus de diffusion stable.

Ce modèle présente plusieurs inconvénients en plus de contrôler la diffusion avec des éléments par différences finies (équation 1.116). En effet, la diffusion sera l'inhibée dans le cas de vrais contours mais aussi pour les fausses détections dues au bruit (lorsque le bruit est très fort, il sera réhaussé).

Aussi, la fonction  $c_1$  peut mener à un coefficient négatif  $c_\eta$ , ce qui peut conduire à une équation de la chaleur inverse dans certains cas (elle peut donc présenter des problèmes d'instabilité numérique), ce qui rend le modèle instable. Le coefficient  $c_\eta$  s'écrit :

$$c_\eta = \left( c_1(\|\nabla I\|) + \|\nabla I\| c_1'(\|\nabla I\|) \right) = \frac{(k^2 - 2\|\nabla I\|^2)}{k^2} \exp\left(-\frac{\|\nabla I\|^2}{k^2}\right)$$

Nous obtenons alors une équation de la chaleur inverse pour certaines valeurs du gradient soit lorsque :

$$\|\nabla I\| > \frac{k}{\sqrt{2}} \quad \text{alors} \quad c_\eta < 0,$$

nous reviendrons sur ces problèmes d'instabilité en présentant d'autres fonctions de la littérature dans la partie 8.7 de ce chapitre.

Outre les forts gradients rehaussés qui peuvent être des points de bruit, cette méthode souffre également au niveau des transitions à forte pente (contours de type rampe). Celles-ci évoluent en segments linéaires séparés par des sauts, ce phénomène d'instabilité est nommé *effet d'escalier*<sup>3</sup>.

Pour répondre à ces problèmes, Catté et al. ont proposé dans [CLMC92] de modifier légèrement ce modèle au niveau de l'estimation du gradient dans l'équation 1.113, afin d'assurer l'existence et l'unicité de la solution. Le module du gradient

---

3. Dans [CMSSX10] les auteurs ont proposé de remplacer la valeur de  $k$  initialement constante dans la fonction  $c_1(\cdot)$  par une autre fonction contenant un indicateur de contours appelé aussi courbure de différence qui permet de distinguer une rampe pour un contour [CMS<sup>+</sup>10].

$\|\nabla I\|$  est alors estimé à partir de dérivées gaussiennes.

L'EDP améliorée est donnée par :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = \operatorname{div} [ c( \|\nabla(G_\sigma \star I)\| ) \cdot \|\nabla I\| ] \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.117)$$

où la fonction  $G_\sigma(x, y)$  est une gaussienne d'écart-type  $\sigma$ .

Cette amélioration a permis de rendre le problème bien posé, de montrer l'existence et l'unicité d'une solution. Elle a ainsi amélioré la convergence des schémas numériques pour toute fonction image de  $L^2$  quelle que soit la fonction  $c(\cdot)$  régulière vérifiant  $c(x) \geq 0$ .

Malheureusement cette équation améliorée n'a toujours pas d'interprétation géométrique satisfaisante avec la présence d'un terme hybride combinant le gradient et son estimée gaussienne dans la divergence. De plus, ce modèle n'est pas stable lorsque le paramètre  $\sigma$  de la gaussienne est proche de 0.

Malgré ces problèmes théoriques difficiles, ces EDP donnent des résultats intéressants. Les figures 1.39 et 1.40 présentent des résultats obtenus avec ces EDP (méthode de Perona-Malik originale et l'approche améliorée) sur des données synthétiques et réelles. Ces résultats ont été obtenus avec la fonction de contrôle  $c(u) = c_1(u)$  en faisant varier le paramètre  $k$  et le nombre d'itérations.

## 8.4 Axiomatisation de l'espace échelle

Une généralisation de l'analyse multi-échelle sous forme d'axiomes architecturaux a été développée dans [AGLM92] et [AGLM93] assurant la stabilité d'une analyse multi-échelle. Nous présentons également les propriétés d'invariance caractérisant une analyse. Cette axiomatisation va constituer une théorie générale regroupant le filtrage linéaire classique isotrope (i.e. le filtrage gaussien), le filtrage anisotrope non linéaire et la morphologie mathématique (érosion, dilatation, analyse en courbes de niveaux). Une analyse multi-échelle sera représentée par une EDP.

L'analyse multi-échelle consiste en la génération d'une suite d'images  $I(x, y, t)$  où chaque image apparaît comme une version lissée d'une image originale  $I_0(x, y)$ , à une échelle d'analyse  $t$ . En conséquence, elle définit une famille de transformations  $(T_t)_{t \geq 0}$  appliquées à l'image originale :  $I(x, y, t) = I_0(x, y)$ , et qui conduit à la suite d'images :  $I(x, y, t) = (T_t(I_0))(x, y)$  et au système suivant :

$$\begin{cases} I(x, y, t) = (T_t(I_0))(x, y) \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.118)$$

où  $T_t$  est un opérateur permettant de passer de l'image initiale  $I(0, x, y)$  à l'image  $I(x, y, t)$  au "temps"  $t$ .

Pour qu'une telle analyse soit stable, il faut que l'opérateur  $(T_t)_{t \geq 0}$  soit causal. Autrement dit, d'un point de vue perceptuel, cette analyse doit correspondre à un processus de vision "raisonnable" au sens physique du terme. D'un point de

vue formalisation mathématique, elle doit vérifier certains axiomes architecturaux, détaillés par la suite, jusqu'à obtenir l'unicité de l'opérateur  $(T_t)_{t \geq 0}$ .

A titre d'exemple, l'équation de la chaleur (équation 1.110) modélise une analyse multi-échelle où la gaussienne (équation 1.107) correspond à notre opérateur  $(T_t)$ . Le laplacien de l'image noté  $\Delta I$  quant à lui correspond à l'opérateur infinitésimal permettant de passer de l'instant  $t$  à l'instant  $t + dt$ .

### Axiomes architecturaux et théorème fondamental

Les axiomes principaux détaillés en annexe 4 sont les suivants :

- Axiomes de récursivité et de structure pyramidale
- Axiomes de comparaison et de comparaison locale
- Axiome de régularité

Les espaces multi-échelle non-linéaires issus de cette axiomatisation sont détaillés dans la partie suivante.

## 8.5 Espaces échelle morphologiques

### Espace d'échelle morphologique euclidien

Si une analyse multi-échelle vérifie les axiomes architecturaux et possède à la fois les propriétés d'invariance euclidienne et d'invariance morphologique alors la solution de cette analyse est solution de viscosité de l'EDP suivante :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = \|\nabla I\| \cdot F\left(\operatorname{div}\left(\frac{\nabla I}{\|\nabla I\|}\right), t\right) \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.119)$$

avec  $F(s, t)$  fonction non décroissante par rapport à la première variable. Cette analyse multi-échelle est connue sous le nom d'espace d'échelle morphologique euclidien (*Euclidean Morphological Scale Space (EMSS)*).

Un cas particulier intéressant consiste à utiliser la fonction  $F(s, t) = s \cdot t$ . L'équation 1.119 devient :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = t \cdot \|\nabla I\| \cdot \operatorname{div}\left(\frac{\nabla I}{\|\nabla I\|}\right) \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.120)$$

Après changement d'échelle du temps [LDA<sup>+</sup>97], en posant  $t' = \frac{1}{2}t^2$ , nous obtenons :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = \|\nabla I\| \cdot \operatorname{div}\left(\frac{\nabla I}{\|\nabla I\|}\right) \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.121)$$

où  $\operatorname{div}\left(\frac{\nabla I}{\|\nabla I\|}\right)$  est la courbure euclidienne des isophotes. L'équation 1.121 peut s'écrire aussi sous la forme :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = I_{\xi\xi} \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.122)$$

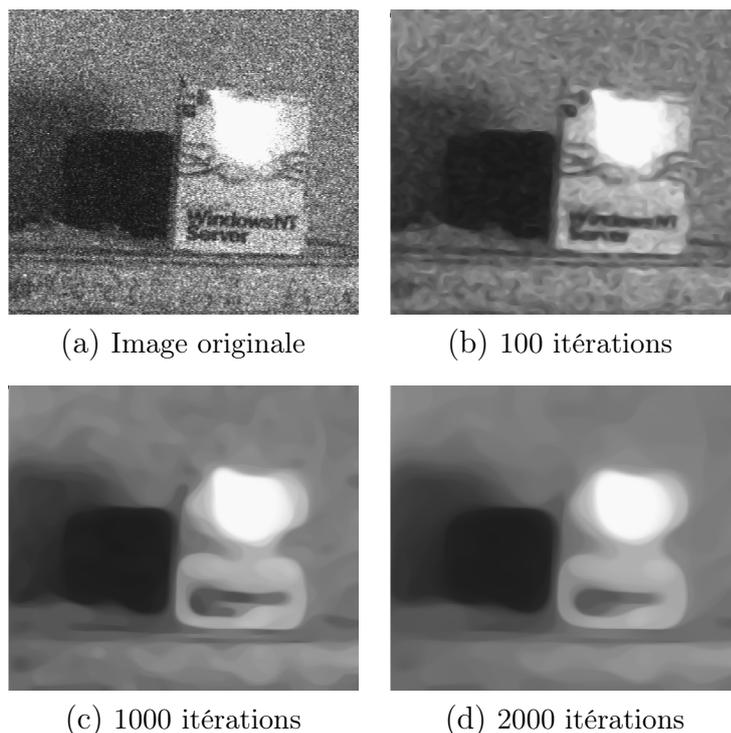


FIGURE 1.41 – Résultats d'une MCM avec différentes itérations.

avec :

$$I_{\xi\xi} = \left( \frac{\nabla I^\perp}{\|\nabla I\|} \right)^t \cdot \nabla^2 I \cdot \left( \frac{\nabla I^\perp}{\|\nabla I\|} \right)$$

où  $\xi$  représente la direction perpendiculaire au gradient dans le plan image. Cette analyse connue sous le nom de *Mean Curvature Motion (MCM)* fait varier les lignes de niveaux de l'image dans la direction du gradient avec une vitesse proportionnelle à leur courbure euclidienne. La figure 1.41 présente des résultats obtenus à l'aide de cette EDP, les contours sont préservés mais les lignes de niveaux se déplacent avec une vitesse qui dépend de leur courbure euclidienne. Les formes dans l'image ont donc tendance à devenir circulaires.

Un autre cas particulier intéressant, lorsque la fonction  $F(s, t) = \pm 1$ . L'équation 1.119 devient :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = \pm \|\nabla I\| \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.123)$$

Dans ce cas, cette analyse multi-échelle correspond aux deux opérateurs de base de la morphologie mathématique que sont l'érosion et la dilatation. En effet si l'on discrétise le temps de la manière suivante :

$$\frac{\Delta I}{\Delta t} = \frac{I^{t+h} - I^t}{\Delta t} = \pm \|\nabla I\|$$

Nous obtenons alors :

$$I^{t+h} = I^t \pm \Delta t \cdot \|\nabla I\|$$

$I^t \pm \Delta t \cdot \|\nabla I\|$  est alors un developpement limité à l'ordre un de  $I$  dans la direction du gradient,  $I^{t+h}$  est alors le max ou le min de  $I^t$  dans une boule de rayon  $\Delta t$ .

### Espace d'échelle morphologique affine

Une analyse multi-échelle satisfaisant à la fois les axiomes de structure pyramidale, de comparaison locale, de régularité et d'invariance affine et morphologique est unique et est représentée par l'EDP suivante :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = \|\nabla I\| \left( t \cdot \operatorname{div} \left( \frac{\nabla I}{\|\nabla I\|} \right) \right)^{\frac{1}{3}} \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.124)$$

Avec ce schéma de diffusion, les courbes de niveaux dans les images évoluent de manière invariante affine (de nombreux détails sont disponibles dans [DF95] et [LDA<sup>+</sup>97]). Cette analyse multi-échelle, appelée espace d'échelle morphologique affine (*Affine Morphological Scale Space (AMSS)*), permet de diffuser les objets dans les images tout en gardant leurs formes.

## 8.6 Un lissage sélectif optimal

Alvarez, Lions et Morel [ALM92] abordent le problème un peu différemment de Malik et Perona. Dans une image  $I_0(x, y)$ , au lieu d'inhiber complètement la diffusion au voisinage des arêtes, l'idée est de diffuser uniquement dans la direction parallèle aux contours. Cela permet de régulariser tant que possible l'image des deux côtés de chaque arête. Ils proposent alors l'EDP parabolique non linéaire suivante :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = g(\|G_\sigma * \nabla I\|) \cdot \left[ (1 - h(\|\nabla I\|)) \Delta I + h(\|\nabla I\|) \cdot \|\nabla I\| \cdot \operatorname{div} \left( \frac{\nabla I}{\|\nabla I\|} \right) \right] \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.125)$$

avec  $G_\sigma$  une gaussienne d'écart type  $\sigma$ . Lorsque  $\|G_\sigma * \nabla I\|$ , l'estimation du gradient, est normalisée,  $g(\cdot)$  et  $h(\cdot)$  sont deux fonctions telles que :

$$g(\|G_\sigma * \nabla I\|) = \begin{cases} 1 & \text{si } \|G_\sigma * \nabla I\| \rightarrow 0 \\ 0 & \text{si } \|G_\sigma * \nabla I\| \rightarrow 1 \end{cases}$$

$$h(\|\nabla I\|) = \begin{cases} 0 & \text{si } \|\nabla I\| \leq e \\ 1 & \text{si } \|\nabla I\| \geq 2e \end{cases}$$

où  $e$  est un coefficient numérique petit souvent de l'ordre de 0.02, il joue le rôle d'un seuil sur les valeurs du gradient. Par exemple, le choix des fonctions  $g(\cdot)$  et  $h(\cdot)$  peut être :

$$g(\|G_\sigma * \nabla I\|) = \exp \left( - \left( \frac{\|G_\sigma * \nabla I\|}{k} \right)^2 \right)$$

$$h(\|\nabla I\|) = \begin{cases} 0 & \text{si } \|\nabla I\| \leq e \\ (\|\nabla I\| - e)/e & \text{si } e < \|\nabla I\| < 2e \\ 1 & \text{si } 2e \leq \|\nabla I\| \end{cases}$$

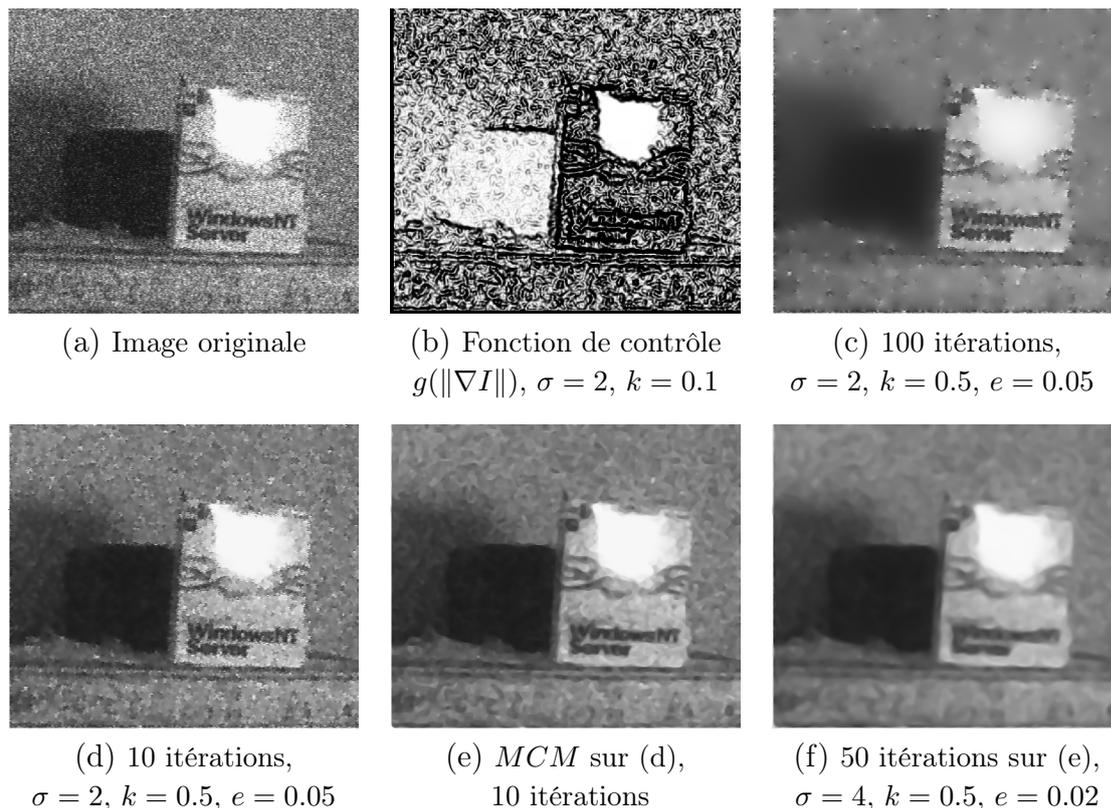


FIGURE 1.42 – Résultat du lissage sélectif optimal. Pour (d), l’objectif est de préserver les contours à gauche de la boîte.

Pour faire le lien avec l’équation 1.115, l’équation 1.125 peut s’écrire sous la forme :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = g(\|G * \nabla I\|) [I_{\xi\xi} + (1 - h(\|\nabla I\|)) \cdot I_{\eta\eta}] \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.126)$$

Contrairement au modèle de Perona-Malik, le choix de la fonction  $h(\cdot)$  toujours positive garantit la stabilité du modèle. Ce modèle donne de bons résultats et semble être un modèle limite parmi les autres méthodes de débruitage à base d’EDP. D’une part, il répond aux différentes exigences de la restauration d’images (l’élimination du bruit et la conservation des contours). D’autre part, d’un point de vue théorique, ce modèle représente une version stable de la restauration par approche variationnelle décrit par l’équation 1.115 et bien justifié par l’existence d’une solution de viscosité de l’EDP 1.125. De plus ce modèle a une bonne interprétation géométrique car la diffusion est isotrope et maximale loin des contours lorsque la valeur du gradient est faible. Aussi, elle s’oriente et s’affaiblit au voisinage des contours pour s’annuler aux points de contours. Toutefois, lorsque le bruit est trop fort, il faut augmenter la valeur de l’écart type de la gaussienne  $\sigma$ , ce qui délocalise les principales structures et diffuse les petits objets comme montré en figure 1.42 (f). Des résultats obtenus avec ce modèle sont présentés à la figure 1.42. Le but poursuivi ici est de régulariser l’image tout en conservant les deux contours à gauche de la boîte, le filtrage est donc plus contraint que pour l’exemple du modèle de Perona et Malik. Pour cette

raison le résultat de la figure 1.42 (d) est donc plus bruité que celui de la figure 1.40 (e). Cependant, quelques itérations de diffusion avec le schéma *MCM* permettent de lisser le bruit sans trop lisser les contours principaux de l'image (figure 1.42 (e)).

## 8.7 Approches variationnelles

Régulariser les images peut être effectué en minimisant une énergie fonctionnelle, cette énergie est une mesure globale des variations locales de l'image. L'objectif est de supprimer les petites variations en préservant les principales structures de l'image.

En recherchant l'image  $I$  définie sur  $\Omega \subset \mathbb{R}^2$  (un rectangle), une image  $I_0$  peut être régularisée telle que l'énergie  $E(I)$  soit minimale :

$$E(I) = \int_{\Omega} \Phi(\|\nabla I\|) d\Omega, \quad (1.127)$$

l'image  $I_{reg}$  régularisée sera donc :

$$I_{reg} = \arg \min_I \left( \int_{\Omega} \Phi(\|\nabla I\|) d\Omega \right), \quad (1.128)$$

avec  $\Phi : \mathbb{R} \rightarrow \mathbb{R}$  une fonction décroissante telle que le processus soit stable et préserve les structures importantes de l'image. Les équations d'Euler-Lagrange associées à la fonctionnelle en équation 1.127 apportent les conditions nécessaires pour la minimisation de l'énergie (voire [DF95] pour plus de détails). L'EDP résultante est de la forme :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = \operatorname{div} \left( \frac{\Phi'(\|\nabla I\|)}{\|\nabla I\|} \cdot \nabla I \right) \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.129)$$

L'équation 1.129 peut alors s'écrire sous la forme :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = c_{\xi} I_{\xi\xi} + c_{\eta} I_{\eta\eta} \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.130)$$

avec :

$$\begin{cases} c_{\xi} = \frac{\Phi'(\|\nabla I\|)}{\|\nabla I\|} \\ c_{\eta} = \Phi''(\|\nabla I\|) \end{cases} \quad (1.131)$$

L'image est alors lissée respectivement dans les directions  $\xi$  et  $\eta$  sous contrôle de  $c_{\xi}$  d'une part et de  $c_{\eta}$  d'autre part. Cependant, afin de préserver les contours et de diffuser isotropiquement dans les zones faibles de gradient, tout en garantissant la stabilité et la convergence, un certain nombre de propriétés doivent être vérifiées pour la fonction  $\Phi$  :

- Il est nécessaire que les paramètres  $(c_{\eta}, c_{\xi})$  soient toujours positifs pour éviter l'équation de la chaleur inverse, soient  $\Phi'(\|\nabla I\|) \geq 0$  et  $\Phi''(\|\nabla I\|) \geq 0$ , la fonction  $\Phi$  doit donc être convexe.

- La diffusion doit être isotrope dans les régions à faible gradient, donc :

$$\lim_{\|\nabla I\| \rightarrow 0} \frac{\Phi'(\|\nabla I\|)}{\|\nabla I\|} = \lim_{\|\nabla I\| \rightarrow 0} \Phi''(\|\nabla I\|) = \Phi''(0) > 0$$

- Les points à fort gradient ne doivent pas être flous, les contours doivent être lissés majoritairement dans les directions des isophotes plutôt que dans les directions perpendiculaires aux contours :

$$\left\{ \begin{array}{l} \lim_{\|\nabla I\| \rightarrow +\infty} \Phi''(\|\nabla I\|) = 0 \\ \lim_{\|\nabla I\| \rightarrow +\infty} \frac{\Phi'(\|\nabla I\|)}{\|\nabla I\|} = 0 \\ \lim_{\|\nabla I\| \rightarrow +\infty} \frac{\Phi''(\|\nabla I\|)}{\frac{\Phi'(\|\nabla I\|)}{\|\nabla I\|}} = 0 \end{array} \right. \quad (1.132)$$

Ces propriétés sont suffisantes pour restaurer des images dégradées en préservant les principales structures. Le choix de la direction de diffusion dépend de la fonction de régularisation  $\Phi$ . Plusieurs fonctions ont été proposées dans la littérature qui mènent à différentes méthodes de régularisation, certaines sont données dans le tableau suivant :

Nom des auteurs	$\Phi(x)$	$\Phi'(x)/x$	$\Phi''(x)$
Perona-Malik [PM90]	$-\frac{k^2}{2} \left( e^{-(x/k)^2} - 1 \right)$	$e^{-(x/k)^2}$	$\left( 1 - 2 \cdot \frac{x^2}{k^2} \right) \cdot e^{-(x/k)^2}$
Perona-Malik [PM90]	$\frac{k^2}{2} \log \left( 1 + \frac{x^2}{k^2} \right)$	$\frac{1}{1+x^2/k^2}$	$\frac{k^2(k^2-x^2)}{(k^2+x^2)^2}$
Geman-McClure [GM85]	$\frac{(s/k)^2}{1+(s/k)^2}$	$\frac{2 \cdot k^2}{(k^2+x^2)^2}$	$-\frac{2 \cdot k^2(-k^2+3x^2)}{(k^2+x^2)^3}$
Tikhonov [Tik63]	$x^2/2$	1	1
Green [Gre90]	$\log(\cosh(x))$	$\frac{\tanh(x/k)}{k \cdot x}$	$k^{-2} (\cosh(x/k))^{-2}$
Rudin et Osher [ROF92]	$x$	$1/x$	0
Aubert [CBFAB94]	$\sqrt{1 + (s/k)^2} - 1$	$\frac{1}{k^{-2} \sqrt{\frac{k^2+x^2}{k^2}}}$	$\frac{ k }{(k^2+x^2)\sqrt{k^2+x^2}}$

Comme les trois premières fonctions ne sont pas convexes, afin que l'EDP soit stable, il faut la condition  $s \leq \frac{k^2}{2}$  pour la première et la troisième fonctions et la condition  $s^2 \leq k^2$  pour la deuxième fonction.

Concernant le choix  $\Phi(u) = u^2/2$  [Tik63], l'équation 1.130 se ramène à l'équation de la chaleur classique (diffusion isotrope) :

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = I_{\xi\xi} + I_{\eta\eta} \\ I(x, y, 0) = I_0(x, y) \end{cases}$$

Finalement, dans le cas où la fonction  $\Phi(u)$  donne des coefficients  $(c_\eta, c_\xi)$  tous deux positifs, l'EDP parabolique de l'équation 1.130 peut être interprétée comme une opération de lissage anisotrope selon les deux directions  $\eta$  et  $\xi$ .

## 8.8 Diffusion anisotrope d'images en couleurs

Des modèles de diffusion pour régulariser les images à partir d'EDP ont été étendus aux images couleurs [TD02]. Les algorithmes de diffusions pour les images scalaires ne peuvent pas être appliqués à chaque plan  $I^i$  d'une image couleur  $\mathbf{I}$  ( $1 \leq i \leq d$ ) un par un. Il faut donc prendre en compte la corrélation entre les différents canaux.

Dans une image scalaire, le gradient indique la direction de la plus importante variation d'intensité. Concernant une image couleur, comme chaque composante de l'image apporte une information différente, il est difficile d'établir quelle information est la plus importante. Le tenseur multispectral [DZ86], détaillé dans la partie 4.4 de ce chapitre apporte une information générale sur une image couleur. En effet, ses valeurs propres maximales et minimales ( $\lambda_+$ ,  $\lambda_-$ ) définissent les variations de l'image dans les directions des vecteurs propres associés.

Plusieurs normes de gradient  $\mathcal{N}(\mathbf{I})$  ont ainsi été définies.

Blomgren et Chan [BC98] ont introduit la variation totale (*total variation* notée  $TV$ ) pour la couleur. Ils calculent la variation de l'image à partir de la racine carrée de la trace du tenseur multi spectral, d'où  $\mathcal{N}(\mathbf{I}) = \sqrt{\lambda_+ + \lambda_-}$ . Cependant le schéma de diffusion proposé dans [BC98] diffuse les intensités de l'image avec différents poids pour chacun des canaux, ce qui fait apparaître des couleurs indésirables.

En effet, prenons le cas du schéma de la  $MCM$  faisant varier les lignes de niveaux de chaque canal de l'image dans la direction du gradient avec une vitesse proportionnelle à leur courbure euclidienne (voir partie 8.5 de ce chapitre). Si la diffusion s'effectue dans les directions  $\xi^i$  associées à chaque canal  $I^i$  de  $\mathbf{I}$  alors l'EDP est donnée par :

$$\begin{cases} \frac{\partial I^i}{\partial t}(x, y, t) = I_{\xi^i \xi^i}^i \\ I^i(x, y, 0) = I_0^i(x, y) \end{cases} \quad (1.133)$$

La diffusion gardera les lignes de niveaux dans les différents canaux. Cependant, comme chaque composante de l'image couleur est traitée séparément, l'image couleur ainsi diffusée obtiendra des couleurs mélangées comme illustré en figure 1.43 (b).

La solution est de diffuser chaque canal dans la même direction  $\xi$  afin de garder les mêmes lignes de niveaux vectorielles :

$$\begin{cases} \frac{\partial \mathbf{I}}{\partial t}(x, y, t) = \mathbf{I}_{\xi\xi} \\ I^i(x, y, 0) = I_0^i(x, y) \end{cases} \quad (1.134)$$

La figure 1.43 (c) présente un résultat de  $MCM$  vectoriel, on remarque effectivement que les couleurs restent cohérentes, il n'y a pas de mélange bleu/jaune contrairement au cas non vectoriel. La valeur  $\xi$  est estimée à partir du tenseur de structure mais ne permet pas de conserver les coins, comme le schéma de diffusion  $MCM$  sur les images scalaires. Différentes normes  $\mathcal{N}(\mathbf{I})$  existent, elles permettent de contrôler la diffusion afin de mieux préserver les principales structures.

Sapiro et Ringach proposent dans [SR96] la norme  $\mathcal{N}(\mathbf{I}) = \sqrt{\lambda_+ + \lambda_-}$  dans l'équation de diffusion suivante pour la régularisation d'images couleurs :

$$\begin{cases} \frac{\partial \mathbf{I}}{\partial t}(x, y, t) = g\left(\sqrt{\lambda_+ - \lambda_-}\right) \cdot \mathbf{I}_{\xi\xi} \\ I^i(x, y, 0) = I_0^i(x, y) \end{cases} \quad (1.135)$$

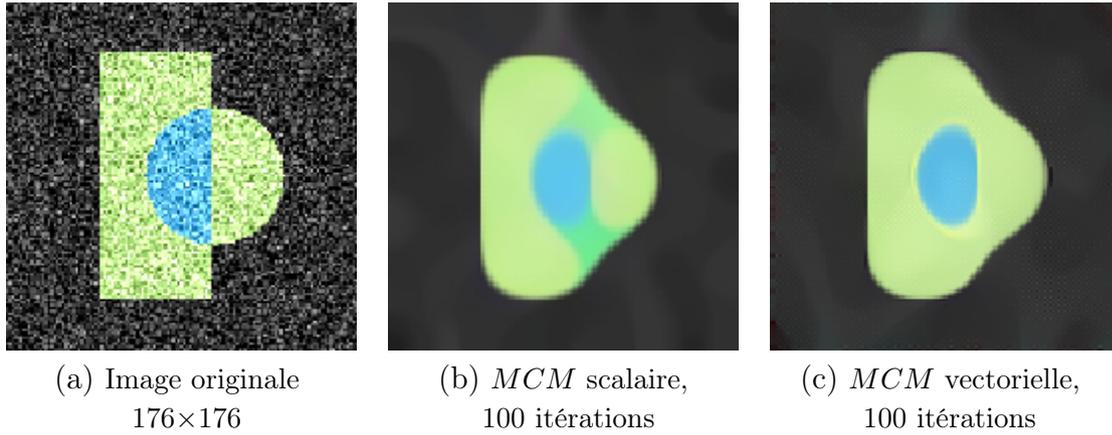


FIGURE 1.43 – Résultat d’une *MCM* couleur et comparaison avec la méthode ne prenant pas compte de l’information vectorielle.

soit par composante  $I^i$  de  $\mathbf{I}$  :

$$\begin{cases} \frac{\partial I^i}{\partial t}(x, y, t) = g(\mathcal{N}(\mathbf{I})) \cdot \mathbf{I}_{\xi\xi} \\ I^i(x, y, 0) = I_0^i(x, y) \end{cases} \quad (1.136)$$

avec :

$$g(\mathcal{N}(\mathbf{I})) = \begin{cases} 1 & \text{si } \mathcal{N}(\mathbf{I}) \rightarrow 0 \\ 0 & \text{si } \mathcal{N}(\mathbf{I}) \rightarrow \infty \end{cases}$$

La fonction  $g(\cdot)$  peut être par exemple :

$$g(t) = \exp\left(-\frac{t}{k}\right)^2$$

L’information vectorielle a donc bien été prise en compte dans l’équation 1.136 par  $g(\mathcal{N}(\mathbf{I}))$ . La direction de diffusion  $\xi$  correspond au vecteur propre du tenseur multi spectral associé à la valeur propre faible  $\lambda^-$ . Au voisinage des contours et des coins, la diffusion est pratiquement inhibée lorsque  $\lambda_+ = \lambda_-$ , ce qui permet de préserver les principales structures de l’image. Toutefois, dans les régions homogènes, la diffusion s’effectue dans la direction tangente au contour  $\xi$ , donc le bruit n’est pas totalement lissé.

Tschumperlé et Deriche ont proposé dans [TD02] d’étendre le modèle de restauration étudié par Kornprobst et al. [KDA97]. Ces méthodes [KDA97] [TD02] combinent filtre de chocs<sup>4</sup>, lissage sélectif ainsi qu’un terme d’attache aux données.

4. Afin de supprimer le flou dans une image, cela revient à effectuer une diffusion inverse de la chaleur, cependant, ce processus est hautement instable et amplifie fortement le bruit. Oscher et Rudin [OR90] proposent un modèle à base d’EDP décrit par l’équation hyperbolique suivante :

$$\begin{cases} \frac{\partial I}{\partial t}(t, x, y) = -\text{sign}(I_{\eta\eta}) |\nabla I| \\ I(0, x, y) = I_0(x, y) \end{cases} \quad (1.137)$$

Cette équation apporte une dilataion dans les zones où l’image est maximale (i.e. le laplacien est négatif) et une érosion dans les zones où l’image est minimale (i.e. le laplacien est positif).

En effet, le terme de diffusion élimine le bruit, celui de choc rehausse les contours afin de supprimer le bruit [OR90] et celui d'attache aux données préserve certains détails au cours du processus de diffusion. Alvarez et Mazzora avaient proposé une EDP qui tient compte des deux premiers termes [AM94]. L'approche variationnelle pour restaurer les images en couleurs [TD02] tient compte de l'analyse vectorielle de Di'Zenzo [DZ86] pour déterminer la direction de diffusion ainsi que la variation locale vectorielle. En effet, ils utilisent la norme  $\mathcal{N}(\mathbf{I}) = \sqrt{\lambda_+}$  correspondant à la valeur de variation maximum dans la direction du vecteur propre associé à  $\lambda_+$ . Contrairement aux normes utilisées dans les travaux précédents ( $\sqrt{\lambda_+ + \lambda_-}$  ou  $\sqrt{\lambda_+ - \lambda_-}$ ), celle-ci n'accorde pas plus d'importance à certains coins.

Ainsi, dans [TD02], les auteurs ont proposé l'équation de diffusion suivante :

$$\begin{aligned} \frac{\partial \mathbf{I}}{\partial t}(x, y, t) = & \alpha_a \cdot (\mathbf{I} - \mathbf{I}_0) + \\ & \alpha_d \cdot (g_1(\mathcal{N}(\mathbf{I}_\sigma)) \cdot \mathbf{I}_{\eta\eta} + \mathbf{I}_{\xi\xi}) + \\ & \alpha_r \cdot (g_1(\mathcal{N}(\mathbf{I}_\sigma)) - 1) \text{sign}(G_\rho * \mathbf{I}_{\eta\eta}) \cdot \|\nabla \mathbf{I}\| \end{aligned} \quad (1.138)$$

où  $\mathcal{N}(\mathbf{I}_\sigma) = \sqrt{\lambda_+} = \|\nabla \mathbf{I}\|$ ,  $\mathbf{I}(x, y, 0) = \mathbf{I}_0(x, y)$ ,  $\mathbf{I}_\sigma$  représente l'image convoluée avec une gaussienne  $G_\sigma$  d'écart type  $\sigma$ .

Similairement au schéma de diffusion de Sapiro [SR96], les fonctions  $g_{1,2}(\cdot)$  peuvent être choisies telles que :

$$g_{1,2}(\mathcal{N}(\mathbf{I})) = \begin{cases} 1 & \text{si } \mathcal{N}(\mathbf{I}) \rightarrow 0 \\ 0 & \text{si } \mathcal{N}(\mathbf{I}) \rightarrow \infty \end{cases}$$

soient par exemple  $g_{1,2}(\mathcal{N}(\mathbf{I})) = \exp\left(-\frac{\mathcal{N}(\mathbf{I})^2}{k}\right)$ .

Les paramètres  $\alpha_{i, i \in \{a, d, r\}}$  permettent de pondérer les différentes opérations présentes dans ce modèle [KDA97] [TD02]. En effet,  $\alpha_a$  est le paramètre pondérant l'influence du terme d'attache aux données,  $\alpha_d$  du terme de diffusion et  $\alpha_r$  du terme de réaction.

L'information vectorielle a bien été prise en compte au niveau des deux directions de diffusion  $\eta$  et  $\xi$ , et aussi au niveau de la norme  $\mathcal{N}(\mathbf{I})$ . Pourtant, l'image est diffusée toujours dans la direction tangente aux contours ce qui mène à rogner les coins. De plus, le terme de réaction (filtre de chocs) peut conduire à des instabilités numériques. Toutefois, les résultats obtenus avec cette approche sont très intéressants et restaurent correctement les images.

## 8.9 Diffusion tensorielle

Afin de diffuser majoritairement dans la direction parallèle aux contours, un tenseur de structure peut être utilisé dans l'intention de déterminer les directions dominantes de diffusion pour une image scalaire ou vectorielle. En effet, les tenseurs de diffusion définissent un formalisme de base et procurent un cadre plus général. Weickert propose ainsi d'utiliser ces tenseurs qui se présentent sous la forme de matrices dont les vecteurs propres définissent les directions d'action et les valeurs propres l'intensité des actions.

L'objectif de la diffusion tensorielle est de permettre de définir des comportements différents en fonction de l'anisotropie de la zone, et de la direction dominante locale. Le besoin se fait alors sentir d'employer des outils plus sophistiqués que le

vecteur gradient. Weickert [Wei] [Wei99] propose ainsi d'utiliser des tenseurs de diffusion. Ces tenseurs se présentent sous la forme de matrices dont les vecteurs propres dirigent les directions et les valeurs propres l'intensité de régularisation. Ces formulations relient la définition d'un champ de tenseurs  $\mathbf{T}$  qui imposent la direction pour le lissage. Ces schémas de diffusion sont généralement utilisés afin d'augmenter la continuité des structures filaires.

Soit  $\mathbf{I}$  une image et  $I^i$  ( $1 \leq i \leq d$ ) sa  $i^{\text{ème}}$  composante. Tout d'abord, le tenseur de structure  $\mathbf{G}_\rho$  de l'image  $\mathbf{I}$  est calculé :

$$\mathbf{G}_\rho(x, y) = \left( \sum_{i=1}^d \nabla I_\sigma^i(x, y) \cdot (\nabla I_\sigma^i(x, y))^T \right) * G_\rho$$

où  $G_\rho$  représente un support gaussien d'écart type  $\rho$  et  $\nabla I_\sigma^i(x, y)$  l'estimation du gradient avec un noyau gaussien d'écart type  $\sigma$  de la  $i^{\text{ème}}$  composante de  $\mathbf{I}$  au pixel de coordonnées  $(x, y)$  (voir partie 4.4 de ce chapitre pour plus de détails).

Ensuite les valeurs propres de  $\mathbf{G}_\rho$  appelée  $(\lambda^+, \lambda^-)$  et les vecteurs propres associés  $(\theta^+, \theta^-)$  sont calculés en chaque pixel de coordonnées  $(x, y)$ . Les vecteurs  $\theta^+$  et  $\theta^-$  représentent des vecteurs unitaires de  $\mathbb{R}^2$  dont les directions indiquent respectivement les variations locales des maxima et des minima des intensités de l'image donnée par  $\lambda^+$  et  $\lambda^-$  [DZ86]. Ces valeurs et ces vecteurs permettent de calculer le champ de tenseur  $\mathbf{T}$  définissant le comportement du lissage local :

$$\mathbf{T}(x, y) = f^-(\lambda^-, \lambda^+) \theta^- (\theta^-)^T + f^+(\lambda^-, \lambda^+) \theta^+ (\theta^+)^T. \quad (1.139)$$

Les fonctions  $f^- : \mathbb{R}^2 \rightarrow \mathbb{R}$  et  $f^+ : \mathbb{R}^2 \rightarrow \mathbb{R}$  définissent la force du lissage désiré le long des deux directions  $(\theta^+, \theta^-)$  au pixel de coordonnées  $(x, y)$ . Plusieurs choix de fonctions  $(f^-, f^+)$  sont possibles mais pour la régularisation des images, il faut lisser majoritairement dans la direction  $\theta^-$  que  $\theta^+$ . En revanche, dans les régions homogènes, la matrice  $\mathbf{T}$  est proche de la matrice identité multipliée par un coefficient  $\alpha$ , donc la diffusion doit être isotropique. Ainsi, les conditions sur les fonctions  $(f^-, f^+)$  sont :

- $f^- \ll f^+$  pour  $\lambda^- \ll \lambda^+$  (diffusion anisotropique)
- $f^- \approx f^+$  pour  $\lambda^- \approx \lambda^+ \approx \alpha$  (diffusion isotropique).

La régularisation de l'image est effectuée avec l'évolution d'une EDP faisant intervenir soit un opérateur de divergence, soit de trace de matrice.

### Divergence basée sur des EDP

Weickert a proposé dans [Wei] [Wei99] l'EDP de divergence suivante pour lisser anisotropiquement  $\mathbf{I}$  à l'aide des vecteurs propres de  $\mathbf{T}$  :

$$\begin{cases} \frac{\partial I^i}{\partial t}(x, y, t) &= \operatorname{div} [\mathbf{T} \nabla I^i] \\ I^i(x, y, 0) &= I_0^i(x, y) \end{cases} \quad (1.140)$$

Le champ de tenseur  $\mathbf{T}$  est le même pour tous les canaux  $I^i$  ( $1 \leq i \leq d$ ) de  $\mathbf{I}$ . Ce schéma de diffusion unifie des méthodes de régularisation sous la contrainte [PM90] [ALM92]. Cependant lorsque l'on développe l'opérateur de divergence :

$$\operatorname{div} [\mathbf{T} \nabla I^i] = \operatorname{trace} [\mathbf{T} \mathcal{H}^i] + (\nabla I^i)^T \cdot \operatorname{div} [\mathbf{T}], \quad (1.141)$$

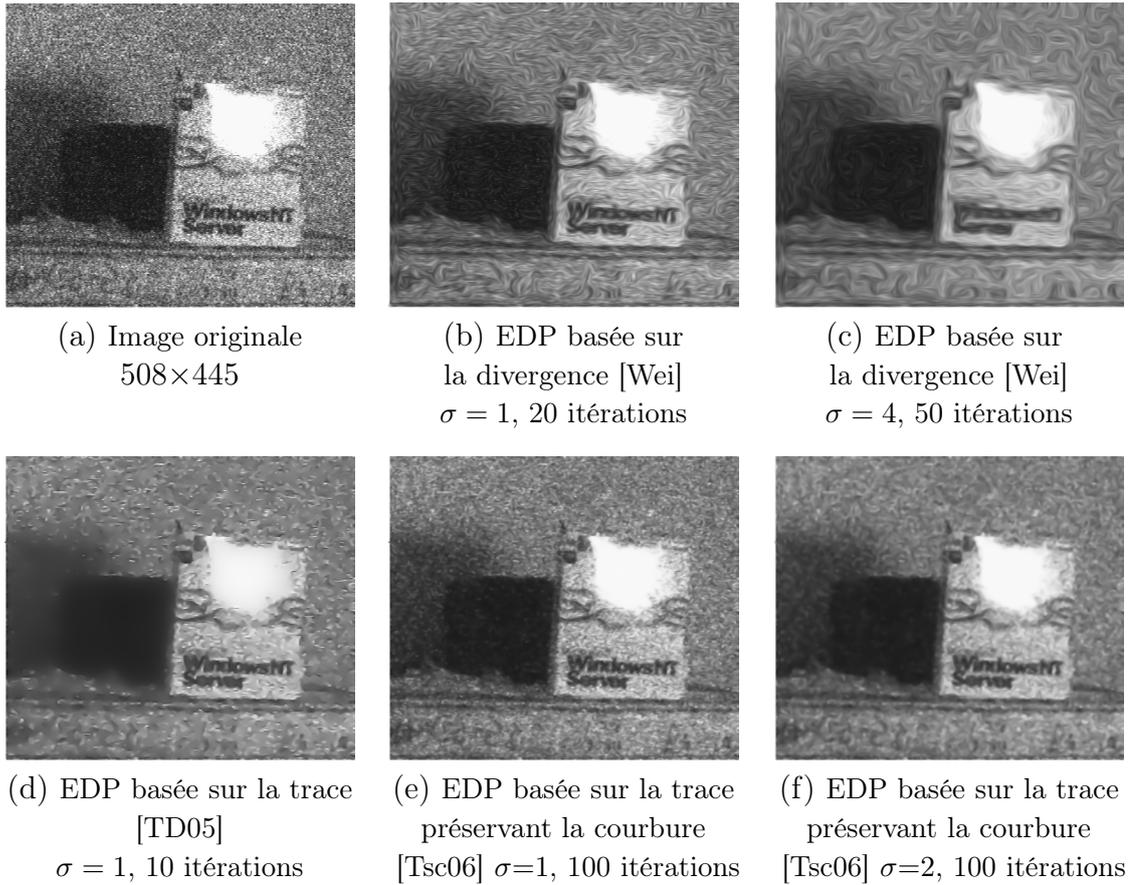


FIGURE 1.44 – Comparaison entre les EDP basées sur la divergence, la trace et la diffusion tensorielle préservant la courbure.

Dans les régions homogènes, le lissage de l'image sera isotropique. Par contre, au niveau des structures, le lissage sera moins important dans la direction du gradient.

Tschumperlé [TD05] [Tsc06] a remarqué que le terme  $(\nabla I^i)^T \cdot \text{div} [\mathbf{T}]$  peut perturber le lissage dirigé par le premier terme trace  $[\mathbf{T}\mathcal{H}^i]$  correspondant à un lissage dirigé localement par les éléments propres de  $\mathbf{T}$ .  $\mathcal{H}^i$  représente la matrice hessienne de la composante  $I^i$  de  $\mathbf{I}$  ( $1 \leq i \leq d$ ). Par conséquent, l'équation 1.140 ne va pas lisser l'image  $\mathbf{I}$  avec les mêmes directions des vecteurs propres de  $\mathbf{T}$ ,  $\theta^-$  et  $\theta^+$ . Cela a pour effet de ne pas diffuser anisotropiquement dans le sens des contours et donc à perdre des structures dans l'image.

### Trace basée sur des EDP

Afin de respecter la diffusion locale de  $\mathbf{T}$ , Tschumperlé a proposé dans [TD05] une autre formulation unificatrice basée sur la trace de l'opérateur et la matrice hessienne :

$$\begin{cases} \frac{\partial I^i}{\partial t}(x, y, t) = \text{trace}[\mathbf{T}\mathcal{H}^i] \\ I^i(x, y, 0) = I_0^i(x, y) \end{cases} \quad (1.142)$$

D'après l'équation 1.139, ce schéma de diffusion revient à :

$$\begin{cases} \frac{\partial I^i}{\partial t}(x, y, t) &= f^-(\lambda^-, \lambda^+) \cdot \frac{\partial^2 \mathbf{I}}{\partial (\theta^-)^2} + f^+(\lambda^-, \lambda^+) \cdot \frac{\partial^2 \mathbf{I}}{\partial (\theta^+)^2} \\ I^i(x, y, 0) &= I_0^i(x, y) \end{cases} \quad (1.143)$$

Ce formalisme permet la conception d'un lissage spécifique qui respecte mieux les propriétés naturelles de régularisation que le modèle de diffusion à partir de la divergence.

Cependant, au niveau des structures courbées telles que les coins, ce schéma de diffusion aura tendance à arrondir les coins, tout comme le schéma de *MCM*.

Dans [Tsc06], Tschumperlé propose un nouveau modèle d'EDP permettant la préservation de la courbure en ajoutant un nouveau terme au schéma de diffusion de l'équation 1.142 :

$$\begin{cases} \frac{\partial I^i}{\partial t}(x, y, t) &= \text{trace}[\mathbf{T}\mathcal{H}^i] + (\nabla I^i)^T \cdot \mathbf{J}_{\theta^-} \\ I^i(x, y, 0) &= I_0^i(x, y) \end{cases} \quad (1.144)$$

avec  $\mathbf{J}_{\theta^-}$  la matrice jacobienne de l'estimation de la direction de l'isophote  $\theta^-$ , en posant  $\theta^- = (u \ v)^T$  :

$$\mathbf{J}_{\theta^-} = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix}.$$

Le terme  $(\nabla I^i)^T \cdot \mathbf{J}_{\theta^-}$  permet de lisser l'image  $\mathbf{I}$  le long de  $\theta^-$  et dépend du champ de vecteurs  $\theta^-$ .

La figure 1.44 compare les trois approches détaillées dans cette partie sur une image réelle très bruitée. Le but est de lisser le bruit tout en préservant les principales structures et en conservant autant que faire se peut les petits objets. L'approche EDP basée sur la divergence [Wei] [Wei99] ne permet pas de supprimer le bruit (figures 1.44 (b) et (c)) puisqu'il est trop fort et considéré comme un objet. Ce schéma de diffusion est adapté aux structures fines. Puisque le bruit est très fort dans cette image, le bruit est alors lissé de façon anisotropique comme si le processus recherchait des chemins à travers le bruit, d'où cet effet fibreux. Lorsque l'écart type de la gaussienne augmente, comme en figure 1.44 (c), cela arrondit les coins, diffuse les petits objets mais lisse toujours le bruit anisotropiquement. Les approches basées sur la trace [TD05] [Tsc06] préservent mieux les objets mais lorsque le bruit est trop fort, la diffusion a tendance à donner un effet fibreux au bruit dans l'image (figures 1.44 (d), (e) et (f)). La méthode préservant la courbure [Tsc06] préserve effectivement les principales structures mais également le bruit fort puisque dans cette zone se forme une courbure importante (figure 1.44 (e)). Il faut alors augmenter l'écart type de la gaussienne, ce qui rend les contours des objets flous comme en figure 1.44 (f).

Les méthodes de diffusion tensorielles permettent de régulariser les images basées sur les champs de tenseurs. Ce type de diffusion met en évidence les structures fines mais lorsque le bruit est trop élevé, ces approches ne sont pas adaptées car le paramètre de lissage doit être augmenté, ce qui diffuse les petits objets.

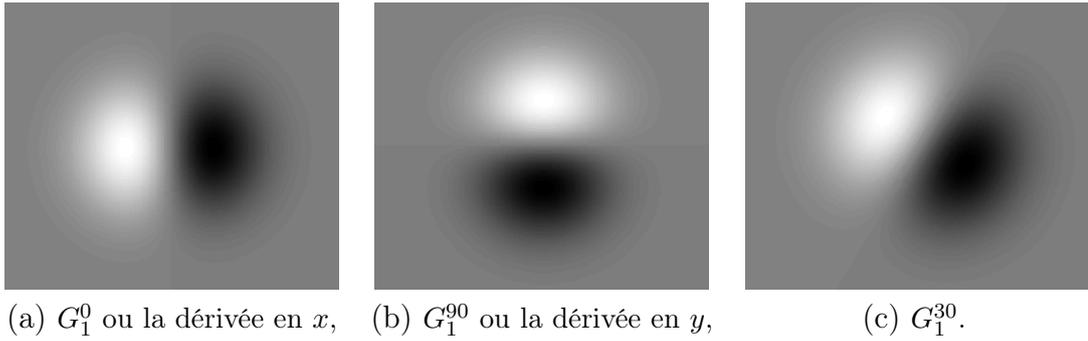


FIGURE 1.45 – Exemple de filtres gaussiens  $G_1^\theta$  orientés.

## 9 Les Filtres Orientés

Les méthodes utilisant la réponse d'un banc de filtres orientés estiment une orientation du contour bien plus précise que les méthodes fondées sur les dérivées de filtres isotropes. Du fait de leurs orientations multiples, ces filtres sont capables de calculer plusieurs orientations pour les points de jonctions ou de coins. L'orientation retenue correspond à la réponse maximum d'un banc où chaque filtre résulte de la rotation d'un filtre de base, isotrope ou anisotrope.

### 9.1 Les filtres isotropes orientés

Les filtres orientables ou "*steerable filters*" sont une autre classe des méthodes de bases de dérivation. Le principe des filtres orientables est de convoluer l'image avec un jeu de filtres de dérivées gaussiennes d'orientations discrètes d'angle  $\theta$  comprises entre 0 et 360 degrés.

#### Définition

Un filtre  $h$  est dit orientable d'un nombre de  $N$  angles si la convolution d'une version orientée de ce filtre  $h_\theta$  avec une fonction  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  peut être exprimée comme suit pour tout  $\theta$  :

$$f(x, y) * h_\theta(x, y) = \sum_{k=1}^N \sum_{i=0}^k b_{k,i}(\theta) f(x, y) * \left( \frac{\partial^{k-i}}{\partial x^{k-i}} \frac{\partial^i}{\partial y^i} g(x, y) \right), \quad (1.145)$$

avec  $b_{k,i}(\theta)$  fonction d'interpolation [FA91] et  $g(x, y)$  une fonction à fenêtre isotrope qui pourrait être la gaussienne.

Ces filtres ont été introduits par Freeman et Adelson et sont construits à partir des deux dérivées partielles d'une gaussienne en deux dimensions  $G_\sigma$  d'écart type  $\sigma$  dans une direction quelconque  $\theta$ . Soit  $G_1^{0^\circ}(x, y)$  la dérivée en  $x$  de la fonction  $G_\sigma$  et  $G_1^{90^\circ}(x, y)$  la dérivée en  $y$  :

$$\begin{cases} G_1^{0^\circ}(x, y) = \frac{\partial G_\sigma(x, y)}{\partial x} \\ G_1^{90^\circ}(x, y) = \frac{\partial G_\sigma(x, y)}{\partial y}. \end{cases} \quad (1.146)$$

Soit  $G_1^\theta$  la dérivée de la fonction  $G_\sigma$  dans la direction  $\theta$  :

$$G_1^\theta(x, y) = \frac{\partial G_\sigma R_\theta}{\partial x},$$

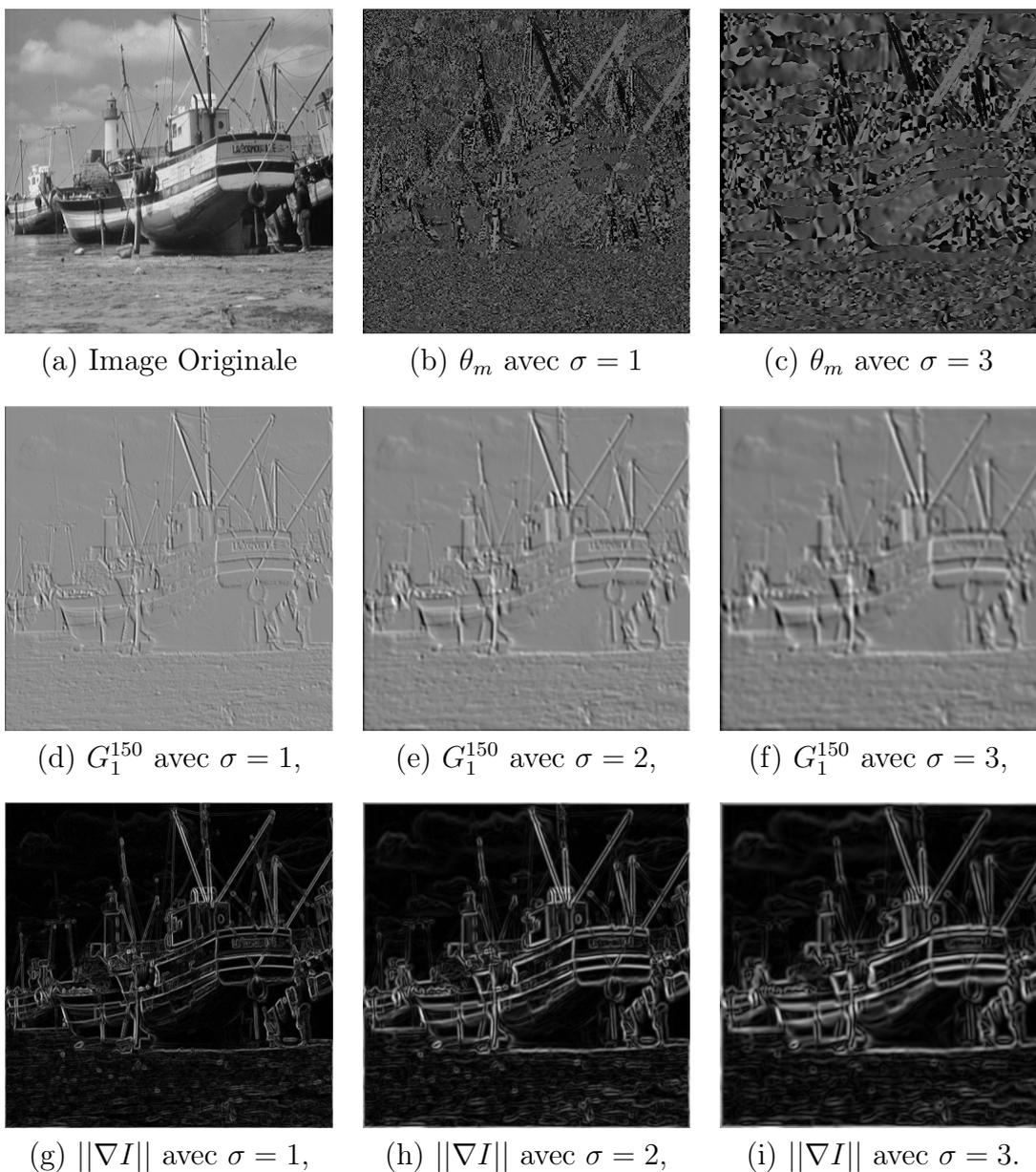


FIGURE 1.46 – Approximation du gradient en utilisant les filtres orientés isotropes. Le filtre de convolution est la gaussienne, différents écarts types sont utilisés. Toutes ces images sont normalisées.

avec  $R_\theta$  la matrice de rotation d'angle  $\theta$ . Freeman et Adelson ont démontré dans [FA91] que la dérivée directionnelle  $G_1^\theta$  d'orientation  $\theta$  peut être générée par une combinaison linéaire d'une rotation des filtres de base  $G_1^{0^\circ}$  et  $G_1^{90^\circ}$  :

$$G_1^\theta(x, y) = \cos(\theta) G_1^{0^\circ}(x, y) + \sin(\theta) G_1^{90^\circ}(x, y). \quad (1.147)$$

La convolution de l'image  $I$  par un noyau orienté (illustré en figure 1.45) s'écrit :

$$(I * G_1^\theta)(x, y) = \cos(\theta) (I * G_1^{0^\circ})(x, y) + \sin(\theta) (I * G_1^{90^\circ})(x, y). \quad (1.148)$$

La magnitude du gradient retenue notée  $\|\nabla I(x, y)\|$  (plusieurs exemples sont disponibles en figure 1.46) et l'orientation du contour en un pixel est celle où les produits de convolution de l'image  $I$  avec et les filtres  $G_1^\theta$  est maximum en valeur absolue :

$$\|\nabla I(x, y)\| = \max_{\theta \in [0, 360[} |(I * G_1^\theta)(x, y)|, \quad (1.149)$$

$$\theta_m = \arg \max_{\theta \in [0, 360[} |(I * G_1^\theta)(x, y)|. \quad (1.150)$$

Une large classe de filtres peut être orientée, les gaussiennes directionnelles en deux dimensions comme les fenêtres circulaires multipliées par un polynôme. Les filtres orientables par combinaisons linéaires des réponses d'un ensemble de filtres de base ne s'appliquent pas seulement aux contours. Aussi, les crêtes ou les vallées sont détectées avec des filtres pairs, elles peuvent donc être estimées avec la dérivée deuxième ou quatrième d'une gaussienne, qui sont également orientables.

Jacob et Unser [JU04] ont conçu une famille optimisée de filtres orientés en fonction des critères de Canny pour la détection de contours et de lignes de crêtes. Les filtres sont construits à partir de combinaisons linéaires de dérivées partielles de noyaux gaussiens. Comme Freeman et Adelson ont démontré que les dérivées de noyaux gaussiens sont orientables, l'approche théorique proposée par Jacob et Unser entre également dans ce cadre. Toutefois, lorsque l'on tente d'augmenter le rapport signal sur bruit du filtre, le filtre doit devenir le plus allongé possible, au prix d'un plus grand nombre de filtres de base (dérivée troisième et cinquième pour les contours et dérivée quatrième pour les lignes de crêtes comme illustré en figure

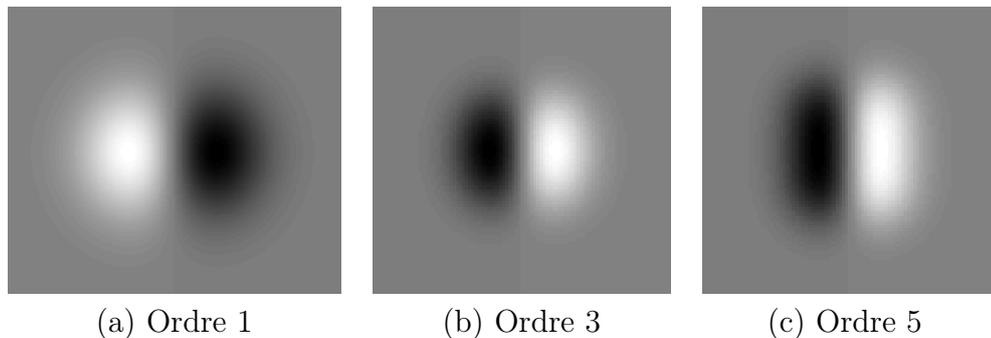


FIGURE 1.47 – Détecteur de contours en  $x$  de Jacob et Unser.

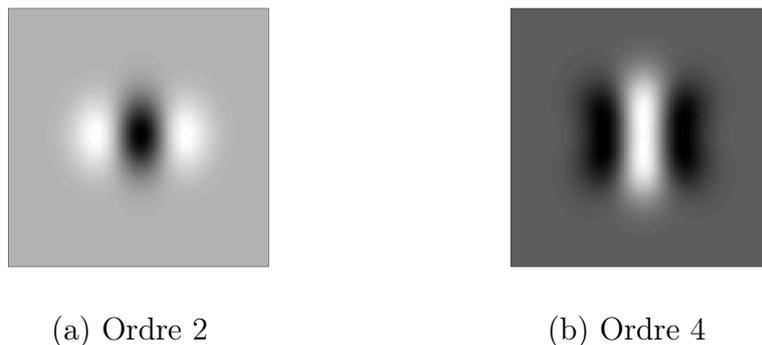


FIGURE 1.48 – Détecteurs de lignes de crêtes. (a) Dérivée seconde en  $X$ , utilisée pour les détecteurs de crêtes et de vallées classiques. (b) Noyaux de Jacob et Unser pour la détection de lignes de crêtes. Ce noyau qui est une combinaison linéaire de filtres jusqu'à l'ordre 4 est plus long que les noyaux classiques d'ordre 2.

1.47). Le problème revient à optimiser les coefficients utilisés dans la combinaison linéaire des dérivées partielles constituant le filtre. Même si les méthodes du troisième et cinquième ordre prennent plus de temps de calcul que les filtres orientables classiques, les résultats sont intéressants. Les différences entre les résultats du troisième et cinquième ordre sont minimales visuellement, toutefois d'après les auteurs, une meilleure optimisation pourrait améliorer considérablement l'efficacité de calcul pour la dérivée cinquième.

Les estimateurs de lignes de crêtes basés sur la décomposition en valeurs propres de la matrice hessienne  $\mathcal{H}(x, y)$  peuvent également être interprétés en termes de filtres orientés. En utilisant l'angle de rotation  $\theta$ , il suffit de rechercher le maximum en chaque pixel de coordonnées  $(x, y)$  de la fonction :

$$\mathcal{H}_\theta(x, y) = (\cos(\theta), \sin(\theta))^t \mathcal{H}(x, y) (\cos(\theta), \sin(\theta)). \quad (1.151)$$

Le vecteur propre associé à la valeur propre absolue maximale de  $\mathcal{H}_\theta(x, y)$  indique l'angle d'orientation de la ligne de crête au niveau du pixel de coordonnées  $(x, y)$ .

Pour la détection de lignes de crêtes, Jacob et Unser [JU04] ont également conçu des filtres orientés construits à partir de combinaisons linéaires de dérivées partielles jusqu'au quatrième ordre de noyaux gaussiens. Comme illustrés en figure 1.48, ces filtres d'ordre 4 sont plus allongés que ceux à l'ordre 2 (utilisés classiquement), ce qui les rend moins sensibles au bruit dans les images.

Au lieu d'utiliser la combinaison linéaire des dérivées partielles d'un filtre, Perona a proposé d'augmenter la sélectivité angulaire en construisant des filtres anisotropes d'ordre 1 et 2 lissant davantage dans la direction de leur orientation.

## 9.2 Les filtres anisotropes orientés

Le concept de noyaux déformables introduits par Perona [Per95] augmente de façon significative la sélectivité angulaire des filtres orientables. En effet, les filtres anisotropes lissants davantage dans la direction de leur orientation sont les dérivées partielles de la fonction  $G_{\sigma_x \sigma_y}(x, y)$  :

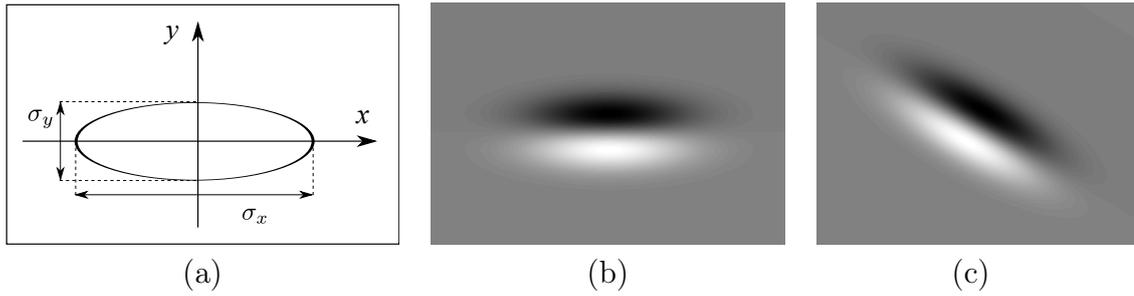


FIGURE 1.49 – Noyau  $G_{\sigma_x \sigma_y}(x, y)$  et  $G'_{\sigma_x \sigma_y}(x, y)$  tel que  $\sigma_x/\sigma_y = 3$ . (a) Noyau déformable. (b)  $G'_{\sigma_x \sigma_y}(x, y)$  original (c)  $G'_{\sigma_x \sigma_y}(x, y)$  orienté à  $30^\circ$ .

$$G_{\sigma_x \sigma_y}(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{x^2}{2\sigma_x^2}} e^{-\frac{y^2}{2\sigma_y^2}}. \quad (1.152)$$

Une autre notation est proposée en équation 2.1 pour un filtre anisotrope dirigé vers une certaine orientation  $\theta$  dans le repère image.

Afin de lisser d'avantage dans la direction du filtre, Perona utilise la fonction  $G_{\sigma_x \sigma_y}(x, y)$  avec un rapport  $\frac{\sigma_x}{\sigma_y} = 3$  (illustré en figure 1.49).

Les contours sont détectés en calculant la dérivée première (par exemple ici en  $x$ ) de ce filtre :

$$G'_{\sigma_x \sigma_y}(x, y) = \frac{\partial G_{\sigma_x \sigma_y}(x, y)}{\partial x} = \frac{-x}{\pi\sigma_x\sigma_y} e^{-\frac{x^2}{2\sigma_x^2}} e^{-\frac{y^2}{2\sigma_y^2}}, \quad (1.153)$$

puis le filtre est tourné à différentes orientations et la direction obtenant l'énergie la plus forte est retenue.

Ces filtres orientés sont approximés pour une échelle par une décomposition en valeurs singulières dans une base de fonctions séparables en  $x - y$ . Toutefois, ces filtres ne peuvent plus être décomposés comme une somme finie de filtres de base comme l'ont démontré Freeman et Adelson pour les filtres isotropes [FA91]. De plus, ces filtres sont très coûteux en temps de calcul puisqu'ils ne sont pas implémentés récursivement. En effet, afin d'extraire les contours, la réponse retenue est celle où l'énergie d'un filtre orienté est maximale pour un pixel. Ce filtre appartient à une banque de filtres qui sont chacun convolués à l'image. Perona propose également une base à la fois multi-échelle et multi-orientation [Per92]; mais seulement les réponses où l'énergie est maximale sur toutes les orientations sont retenues, les coûts calculatoires en sont augmentés. Même si le temps de calcul est important en utilisant ce type de filtres, ils sont utilisés notamment pour la caractérisation de texture [LM01].

La détection de structures orientées à partir d'un filtrage anisotrope gaussien a été implémentée de façon récursive par Guesebroek et al. [GSvdW02]. La figure 1.50 montre l'estimation d'un gradient avec des filtres anisotropes gaussiens, les structures linéaires ressortent mieux que les petits objets ou la texture. Le grand intérêt de ce travail est le gain en temps de calcul, puisque les contours ou les lignes de crêtes ne sont plus estimés à partir d'une base de filtres.

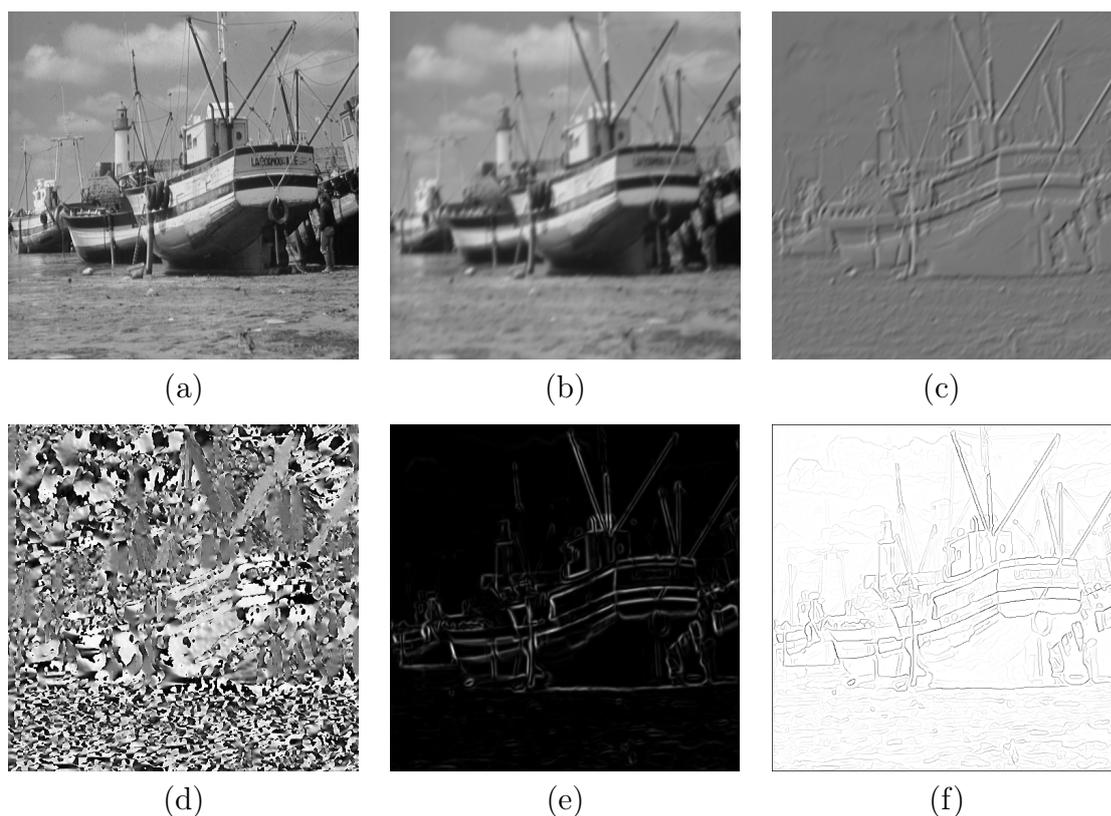


FIGURE 1.50 – Image lissée et dérivée avec l'orientation  $\theta = 35^\circ$  et estimation de gradient en utilisant les filtres gaussiens anisotropes orientés de paramètres  $\sigma_x = 3$ ,  $\sigma_y = 1$  et  $\Delta\theta = 5^\circ$ . (a) Image originale. (b) Image lissée à  $35^\circ$ . (c) Approximation du gradient à  $35^\circ$  avec la dérivée première du filtre anisotrope (équation 1.153). (d) Image d'angle  $\eta$ . (e) Estimation de l'image de gradient. (f) Maxima du gradient dans la direction  $\eta$ . Les images (c), (d), (e) et (f) sont normalisées.

Enfin, Knossow et al. [KvdWHR07] ont adapté le filtre gaussien anisotrope à la couleur à partir de tenseurs (voir partie 4.3 de ce chapitre).

### 9.3 Les filtres de coins orientés

Les filtres orientables anisotropes fournissent une estimation de l'orientation modulo 180 degrés, ce qui convient à des structures linéaires [FA91] [Per92] [GSvdW02] dans le cas des contours ou des lignes de crêtes. Cependant ces filtres ne sont pas adaptés pour l'estimation de plusieurs directions telles que les jonctions ou les coins. La raison principale vient du fait que la symétrie des filtres orientés se situe autour du point central des masques de convolution (figure 1.49). Cela accentue le phénomène de recouvrement et empêche alors une étude pertinente de toutes les différentes directions (i.e. de 0 à  $360^\circ$ ) à partir du point central comme illustré en figure 1.51 (d), (e) et (f).

Les filtres de coins orientés (appelés "*wedges steerable filters*") introduits par Simoncelli et Farid [SF96] sont des masques asymétriques qui étudient l'orientation dans différentes directions centrés sur un pixel. La base de ces filtres est constituée d'éléments identiques avec des orientations différentes. Ces filtres, comme pour les filtres anisotropes orientés présentés dans la partie précédente sont décomposables

en une partie angulaire  $h_e(\theta)$  (filtre pair) et une partie radiale  $h_o(\theta)$  (filtre impair) commune aux deux masques. Ces fonctions sont déterminées de manière à maximiser la sélectivité angulaire de la paire de filtres. Seule la phase entre les deux filtres varie (90 degrés) et une transformation de Hilbert permet de passer de l'une à l'autre.

Ces fonctions angulaires consistent en une somme pondérée de fonctions harmoniques :

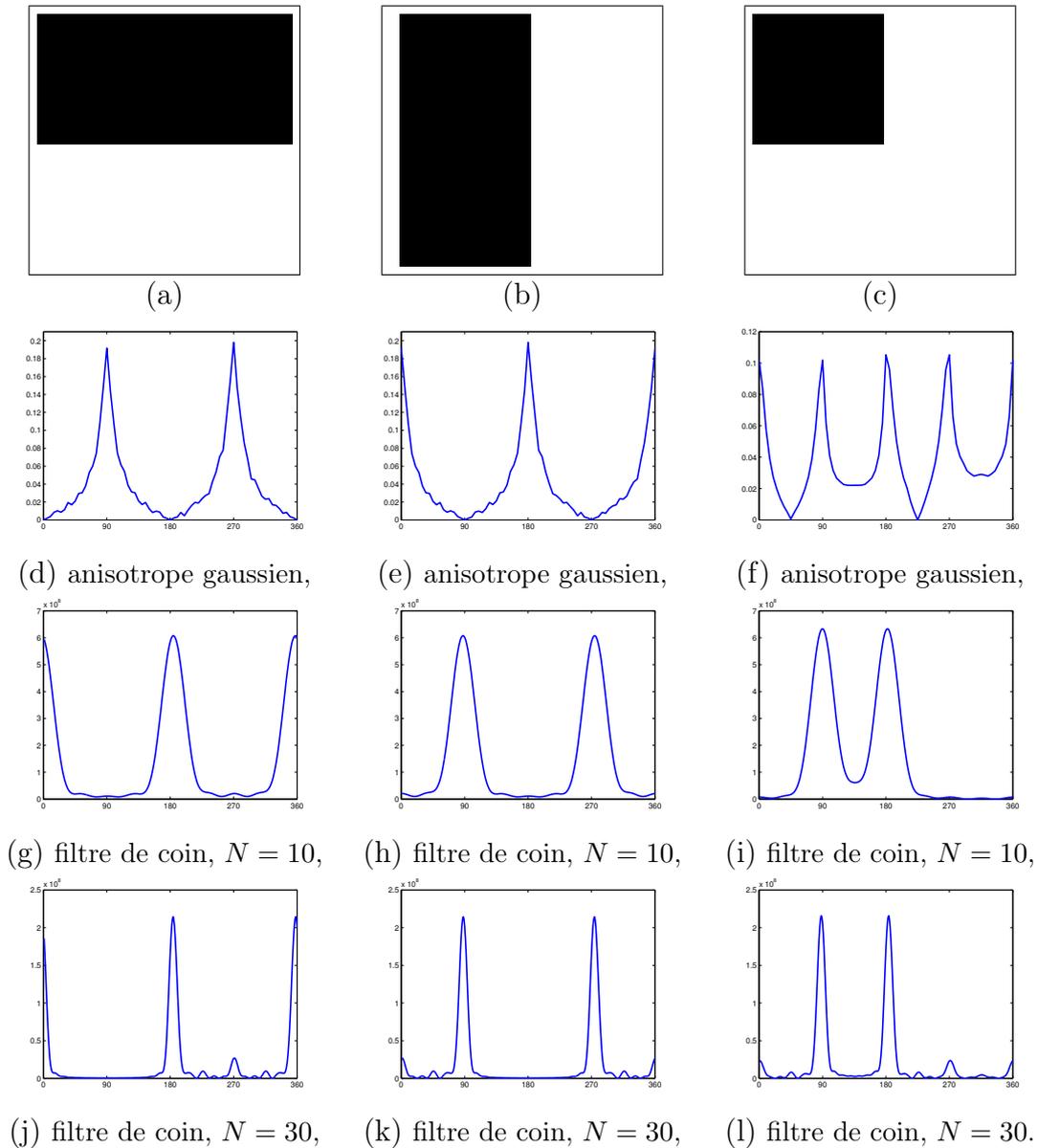


FIGURE 1.51 – Comparaison des énergies orientées obtenues à partir du détecteur de contours gaussien anisotrope [Per92] [GSvdW02] et du détecteur de coins orienté [SF96] sur un contour horizontal, vertical et un coin. L'axe des ordonnées indique l'énergie estimée et celui des abscisses représente l'orientation du filtre en degrés. Les deux filtres tournent dans le sens anti-horaire, l'origine du filtre gaussien anisotrope orienté est verticale tandis que celle du filtre de coin orienté est horizontale vers la droite comme illustré en figure 1.53 (a) et (e).

$$h_e(\theta) = \sum_{n=1}^N w_n \cdot \cos(n\theta) \cdot f(x, y), \quad (1.154)$$

$$h_o(\theta) = \sum_{n=1}^N w_n \cdot \sin(n\theta) \cdot f(x, y), \quad (1.155)$$

Le nombre de filtres de base nécessaire est de  $2N$ , soit  $N$  filtres pairs et  $N$  filtres impairs. Plus  $N$  augmente, plus le nombre de masques de convolution augmente et plus le filtre devient étroit comme illustré en figure 1.53.

Les coefficients  $w_n$  sont déterminés de telle manière à maximiser la sélectivité du filtre, ce qui revient à minimiser le critère  $P(\vec{w})$  suivant :

$$P(\vec{w}) = \int \lambda^2(\theta) (h_e^2(\theta) + h_o^2(\theta)) d\theta,$$

où  $\vec{w} = \{w_n, n \in \{1, N\}\}$  et  $\lambda(\theta)$  est une fonction strictement croissante (par exemple  $\lambda(\theta) = \theta$  simplifiera la minimisation).

$P(\vec{w})$  peut être réécrit sous forme matricielle :

$$P(\vec{w}) = \vec{w} [C^t \Lambda^t \Lambda C + S^t \Lambda^t \Lambda S] \vec{w}, \quad (1.156)$$

où  $\Lambda = \{\lambda_{ii} = 2\pi i/m\}$  est une matrice diagonale contenant les valeurs de  $\lambda(\theta)$ , avec  $i \in \{1, N\}$ .  $C$  et  $S$  sont des matrices définies en fonction de l'échantillonnage de la valeur  $\theta$  en  $m$  points par :

$$C = \{C_{ij} = \cos(2\pi ij/m)\}, \quad (1.157)$$

$$S = \{S_{ij} = \sin(2\pi ij/m)\}, \quad \text{avec } (i, j) \in \{1, N\}^2. \quad (1.158)$$

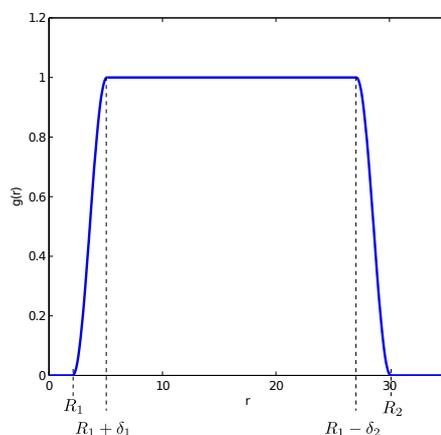


FIGURE 1.52 – Fonction  $g(r)$  proposée par Simoncelli et Farid [SF96] avec  $R_1 = 2$ ,  $R_2 = 30$ ,  $\delta_1 = 3$  et  $\delta_2 = 3$ .

La fonction radiale  $g(r)$  peut maintenant être combinée à partir de  $h_e(\theta)$  et de  $h_o(\theta)$ . Donc  $g(r)$  est choisie sans affecter le caractère d'orientabilité du filtre. Une fonction radiale  $g(r)$  peut être choisie unitaire sur une couronne  $(R_1 + \delta_1, R_2 - \delta_2)$ .  $R_1$  correspond au rayon intérieur et  $R_2$  au rayon extérieur. La fonction  $g(r)$  est choisie de manière à décroître lentement vers 0 au centre et à l'extérieur du masque :

$$g(r) = \begin{cases} 0, & r < R_1 \\ \frac{1}{2} [1 - \cos(\pi(r - R_1)/\delta_1)], & R_1 \leq r < R_1 + \delta_1, \\ 1, & R_1 + \delta_1 \leq r < R_2 - \delta_2, \\ \frac{1}{2} [1 + \cos(\pi(r + \delta_2 - R_2)/\delta_2)], & R_2 - \delta_2 \leq r < R_2, \\ 0, & R_2 \leq r. \end{cases} \quad (1.159)$$

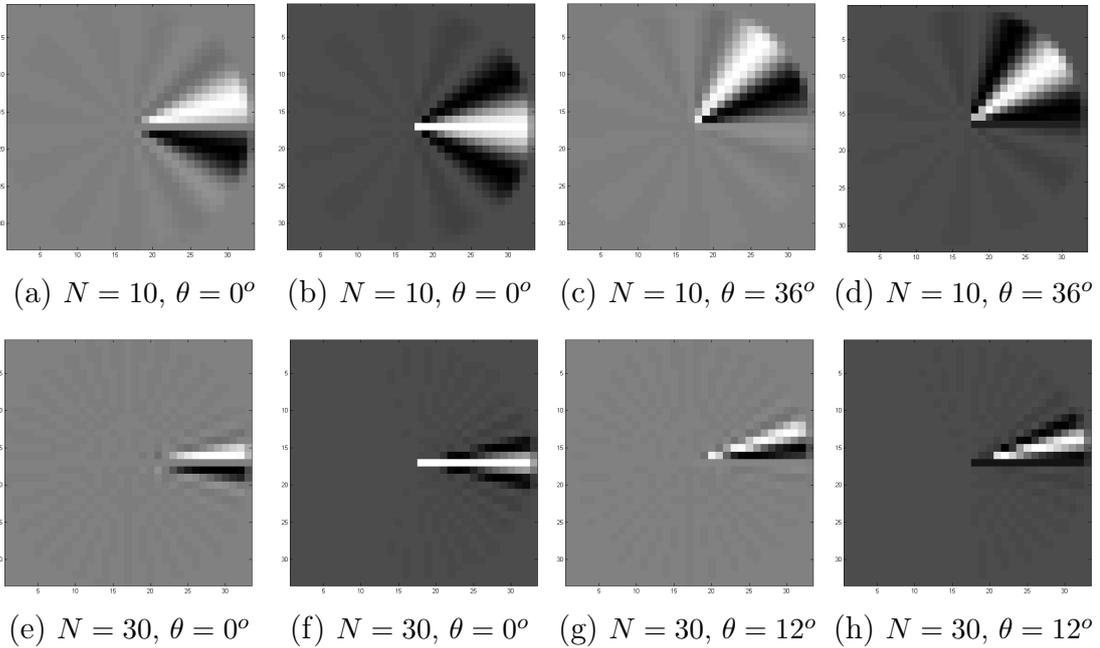


FIGURE 1.53 – Exemple de paires de filtres orientés de coins.

La fonction radiale permet de contrôler l'échelle d'analyse des filtres. Un exemple de fonction  $g$  est présenté en figure 1.52. Les fonctions d'interpolation réalisent l'interpolation comme une combinaison linéaire des réponses des filtres de base  $h_e(\theta)$  et  $h_o(\theta)$ . La figure 1.53 présente deux paires de filtres de coins orientés à différentes orientations et différentes valeurs de  $N$ . Les maxima locaux indiquent les orientations des contours ou des crêtes pour un pixel. Même si ces filtres permettent d'obtenir une énergie orientée dans différentes directions, centrés sur un pixel comme illustré en figure 1.51, ils utilisent des masques de convolution de grande taille. Par exemple dans la figure 1.53, les masques utilisés sont de taille  $33 \times 33$ , ce qui rend les calculs plus importants que des filtres récursifs. Aussi, les filtres ont une forme éparse, ce qui rend la détection des orientations des contours peu précise. Le fait d'augmenter le nombre de filtres (i.e. choisir une valeur de  $N$  plus grande) va permettre de calculer des orientations plus précises (figure 1.51 (j) (k) et (l)) au prix d'un temps de calcul plus important. De plus, ces filtres ne fournissent pas une estimation très locale. En effet, concernant les filtres gaussiens isotropes [FA91] et les filtres anisotropes [Per92]

[GSvdW02], les pixels au centre du masque sont les plus influençables pour le calcul de l'énergie. Ainsi, pour les filtres de coins orientés, tous les pixels à une distance entre  $R_1$  et  $R_2$  du centre du masque dans la zone non nulle du filtre possèdent le même poids et est supérieur à l'influence du pixel central. Ce pixel et ses voisins à une distance inférieure à  $R_1$  du centre du masque n'ont aucune influence pour l'estimation de l'orientation (le nombre de voisins dépend de la valeur de  $R_1$  et influe sur la fonction  $g(r)$  en équation 1.159).

## 9.4 Les filtres orientables de coin gaussien

Yu et al. [YDS01] ont également proposé une approche à base de filtres asymétriques afin d'améliorer la résolution angulaire de la méthode présentée précédemment en modifiant notamment l'interpolation réalisée.

Le filtre proposé est issu d'un produit d'une fonction radiale  $g$  et d'une gaussienne  $G_0$  à petit support. La fonction  $g$  est similaire à la méthode proposée par Simoncelli et Farid [SF96], elle définit un support identique à une couronne bornée par  $R_1$  et  $R_2$ .

La réponse angulaire  $S(\theta)$  est calculée à partir d'une moyenne de l'intensité lumineuse de l'image autour du pixel étudié dans le support angulaire mais l'utilisation de la gaussienne évite d'obtenir un signal trop brut contrairement à l'approche de Simoncelli et Farid [SF96]. L'analyse directe du signal  $S(\theta)$  obtenu permet d'extraire les orientations des lignes de crêtes et des vallées qui correspondent à des pics dans le signal. Les orientations des contours peuvent être étudiées à partir de la dérivée  $DS(\theta)$  de ce même signal, les maxima locaux indiquent les orientations des contours ou des crêtes pour un pixel. Toutefois,  $S(\theta)$  et  $DS(\theta)$  ne peuvent pas constituer une paire de filtres en quadrature car  $DS(\theta)$  n'est pas une transformée de Hilbert de  $S(\theta)$ .

Tout comme la méthode de Simoncelli et Farid [SF96], la méthode d'orientation de coins de Yu et al. [YDS01] apporte une information intéressante au niveau des coins. Toutefois concernant cette approche comme celle de Simoncelli et Farid, les pixels au centre du masque n'ont pas de poids pour le calcul de l'énergie.

## 10 Conclusion

Ce chapitre, dédié aux travaux présentés dans la littérature est nécessaire afin de situer nos contributions détaillées ci-après. Ainsi, nous commençons par présenter le filtrage linéaire indispensable en traitement des images avec notamment les filtres récursifs et non récursifs.

Dans la majeure partie de nos travaux, nous utilisons la détection de contours dans les images, c'est pourquoi nous avons détaillé les opérateurs gradient et laplacien ainsi que différents tenseurs avant de rappeler l'évolution des détecteurs de bords dans les images. Nous avons également détaillé les approches analytiques pour un filtre optimal en détection de contours en présentant le filtre gaussien récursif que nous allons utiliser dans nos approches. Aussi, les différentes approches de détection de contours sont accompagnées de résultats sur des images réelles, ce qui permet de visualiser les qualités et les défauts de ces détecteurs. Nous présentons également les méthodes de filtrage pour la détection de contours de type toit qui sont également

accompagnés de résultats. Dans nos algorithmes de détection de contours détaillés dans le chapitre suivant, nous comparons nos résultats avec ceux explicités dans ce présent chapitre.

La partie suivante est consacrée aux espaces échelle et aux EDP. Parmi nos contributions sont détaillées des approches en diffusion des images par EDP. Nous avons ainsi présenté différentes approches de régularisation des images par diffusion anisotropique, avec leurs qualités et leurs défauts. Ces méthodes sont accompagnées de résultats, nous présenterons également d'autres résultats dans le chapitre dédié à nos contributions.

Finalement nous terminons ce chapitre en détaillant les filtres orientables qui nous ont inspirés grandement pour la majeure partie de nos contributions. Nous avons explicité leurs différents défauts pour avancer nos améliorations.



# Chapitre 2

## Extraction de Contours et Diffusion Anisotropique des Images

### 1 Introduction

Ce chapitre est dédié aux travaux abordés dans cette thèse.

Le premier sujet présenté en partie 2 concerne la détection anisotrope de contours et plus précisément une détection basée à partir de deux demi-noyaux gaussiens qui permettent d'estimer en chaque pixel le chemin entrant et sortant d'un contour. Nous obtenons ainsi un détecteur de contours qui reste précis au niveau des bords des objets et également des coins. De nombreux résultats ont été présentés et comparés à d'autres méthodes de détection de bords puis une amélioration a été proposée en calculant les contours sur les images corrigées par les courbes gamma. Cela permet d'estimer les contours dans les zones sombres ou sur-éclairées de l'image (comme dans l'ombre ou les contre jours). Nous avons apporté des résultats quantitatifs à nos approches, ce qui permet d'estimer la robustesse de nos méthodes de détection anisotropes de contours. Les résultats quantitatifs ainsi que ceux sur des images réelles permettent de valider notre approche.

La partie suivante est dédiée à la détection anisotrope de contours de type toit (partie 3). La nouveauté de cette approche réside dans l'utilisation de différences de demi-gaussiennes anisotropes. Cette approche permet d'extraire les lignes de crêtes, même lorsqu'elles sont courbées et dans des images très bruitées. Nous avons testé cette approche sur des images variées de différents types et, avons apporté ici aussi des résultats quantitatifs, validant ainsi notre méthode.

La dernière partie de ce chapitre (partie 4) est consacrée à une nouvelle méthode de diffusion anisotropique des images. Classiquement, les méthodes de diffusion anisotropiques lissent les images dans le sens des contours mais dans une direction unique tangente au contour. Ici, nous avons développé un algorithme qui diffuse dans les deux directions des contours. L'approche proposée procède en deux étapes, premièrement elle classe les pixels appartenant à des régions ou des contours puis dans un deuxième temps régularise les images en lissant anisotropiquement dans les directions des bords et isotropiquement ailleurs. Cette méthode permet de conserver les coins et les petits objets. Ainsi nous l'avons utilisée pour supprimer la texture dans les images et l'avons adaptée pour la restauration des images. La comparaison avec d'autres approches et les résultats statistiques ainsi apportés nous permettent

de montrer la supériorité de notre approche.

## 2 Des demi-filtres orientés pour la détection de contours

### 2.1 Introduction

Les méthodes de détection de contours diffèrent en fonction du type de filtres de lissage appliqués, elles diffèrent également pour le type de filtre utilisé dans l'approximation du gradient (plusieurs de ces méthodes sont détaillées dans le premier chapitre [SC92] [FA91] [Der92] [Per92] [GSvdW02] [SF96] [SB97]). Ziou et Tabbone réfèrent également un certain nombre de filtres dans [ZT98]. Le détecteur de contours présenté dans ce chapitre est motivé par les applications ayant besoin d'un *bon* détecteur de bords [Can86], produisant des points précis et réellement informatifs tout en détectant le moins possible de faux négatifs. Par exemple, dans le domaine de la recherche ou reconnaissance d'objets dans des scènes naturelles, ces objets peuvent présenter des formes variées et des aspects colorimétriques différents. Pour de tels cas, la majorité des méthodes locales utilisant les points d'intérêts et des invariants couleurs s'avère inefficace.

Dans beaucoup d'applications, la détection de contours est un point central puisqu'elle peut produire des informations géométriques. Dans la plupart des cas, l'utilisation des détecteurs de contours classiques [Can86] [SC92] [Der87] [Der92] [SB97] ne mène pas directement aux bords des objets ; par conséquent les contours doivent être recherchés parmi un certain nombre de points de bords ainsi détectés. De plus, les jonctions et les coins ne sont pas bien détectés et mal localisés par des méthodes classiques. En effet, la réponse du filtre est plus importante à coté du coin (plus exactement à l'intérieur du coin) que sur le coin lui-même et la régularisation de contours proches induit également une délocalisation [LTM07].

Pour remédier à ces différents problèmes, Freeman et Adelson [FA91] ont introduit les filtres orientables qui peuvent être dirigés à des orientations spécifiques à l'aide d'une combinaison linéaire de filtres isotropes. Les techniques utilisant le filtrage anisotrope gaussien ont été introduites par Perona [Per92] et implémentées récursivement par Geusebroek et al. [GSvdW02] puis en couleur par Knosow et al. [KvdWHR07]. Ces approches sont capables de détecter correctement les structures linéaires. Concernant les méthodes de filtrage anisotrope [Per92] [GSvdW02] [KvdWHR07], la robustesse envers le bruit dépend fortement des deux paramètres de lissage du filtre (ces paramètres correspondent aux deux écarts types de la fonction gaussienne en deux dimensions). Si ces paramètres croissent, la détection est moins sensible au bruit mais les petites structures et petits objets seront considérés comme du bruit et donc ignorés. Par conséquent, la précision des points détectés décroît fortement au niveau des pixels de coins et pour les objets ayant des bords non linéaires. Les filtres de coins orientés apportent une meilleure précision au niveau des coins et jonctions [SF96] [YDS01]. Toutefois, ces filtres sont trop évasés, le ou les pixels centraux n'ont aucune influence et leur mise en oeuvre est trop lente puisque ce ne sont pas des filtres récursifs.

Dans cette section, nous présentons un nouveau détecteur de contours anisotrope capable de détecter les jonctions et les coins. Il est composé de deux filtres

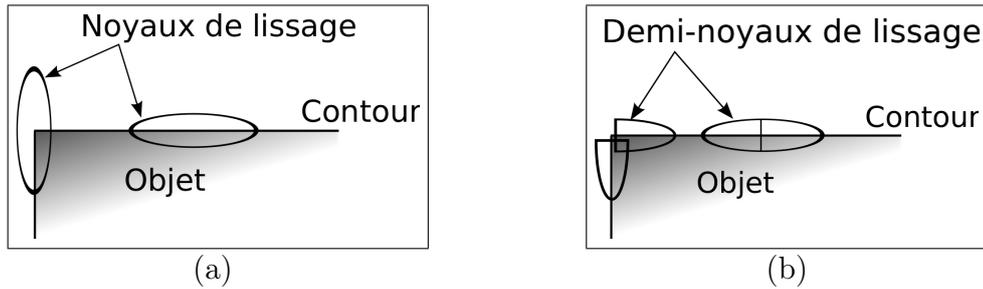


FIGURE 2.1 – Noyaux gaussiens anisotropes sur les portions linéaires de contours et des coins. (a) Noyau anisotrope complet. (b) Demi-noyaux gaussiens anisotropes.

allongés dérivateurs orientés dans deux directions différentes. Ce détecteur de bords impliquant des filtres récursifs calcule un opérateur de contours construit avec une maximisation selon deux directions locales de la réponse du filtre.

Nous avons également amélioré la détection de contours en utilisant les courbes gamma sur l'image originale [Poy96], les contours sont détectés à partir de trois images : l'image originale et deux images corrigées. Les résultats ainsi obtenus permettent de détecter des contours dans les zones sous ou sur-exposées de l'image, tout en restant robuste au bruit.

## 2.2 La dérivée d'un demi-filtre gaussien orienté

Comme exposé dans le chapitre 1 en section 9.2, la détection anisotrope de contours se calcule à partir d'images lissées avec une banque de noyaux de dérivées de gaussiennes anisotropes orientées. Nous pouvons également extraire les contours après avoir lissé l'image originale avec un noyau gaussien anisotrope orienté :

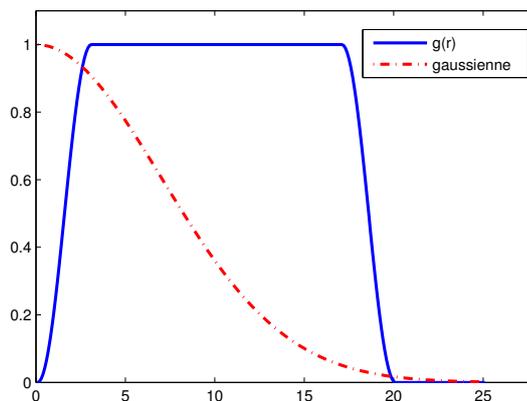


FIGURE 2.2 – Comparaison entre la fonction gaussienne et  $g(r)$  (équation 1.159). Dans le voisinage de l'origine, les points sont faiblement pondérés contrairement à la gaussienne. Inversement, les points plus éloignés ont la même influence alors qu'avec la gaussienne, l'effet diminue avec la distance. Dans cette illustration, la valeur de l'écart type de la gaussienne utilisée est de 7 pour une longueur de filtre similaire.

$$g_{(\mu,\lambda)}(x, y, \theta) = C \cdot e^{-\begin{pmatrix} x & y \end{pmatrix} R_\theta^{-1} \begin{pmatrix} \frac{1}{2\mu^2} & 0 \\ 0 & \frac{1}{2\lambda^2} \end{pmatrix} R_\theta \begin{pmatrix} x \\ y \end{pmatrix}} \quad (2.1)$$

où  $C$  est un coefficient de normalisation,  $R_\theta$  une matrice de rotation horraire d'angle  $\theta$ ,  $(x, y)$  les coordonnées d'un pixel et  $(\mu, \lambda)$  sont les deux paramètres de l'écart type de la gaussienne.  $R_\theta$  est également la matrice de passage pour le changement de base du filtre dans la direction  $\theta$  :

$$R_\theta = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}.$$

Par convolution de l'image  $I$  avec ces noyaux orientés, nous obtenons une collection d'images lissées  $I_\theta$  qui peuvent être dérivées le long de l'axe des  $X$  et des  $Y$  afin d'obtenir les dérivées directionnelles anisotropes  $I_{\theta X}$  et  $I_{\theta Y}$ . Ces dérivées peuvent être combinées dans un tenseur d'orientation [GSvdW02] puis le gradient estimé avec la plus grande valeur propre du tenseur.

Comme expliqué en introduction de ce chapitre ,en section 9.2 du chapitre 1 et également schématisé en figure 2.1 (a), le détecteur anisotrope de contours ci-dessus est très performant pour des portions linéaires de contours mais au niveau des coins, la magnitude du gradient décroît. Aussi, la robustesse au bruit décroît lorsque les deux paramètres  $\lambda$  et  $\mu$  décroissent. Inversement, les petits objets sont considérés comme du bruit et ignorés si ces paramètres croissent.

La solution la plus simple est d'utiliser des filtres orientés de coins estimant les chemins croisant chaque pixel dans plusieurs directions comme exposé en section 9.3 du chapitre 1 [SF96] [YDS01]. Cependant ces filtres sont très évasés et ne permettent pas la détection précise de points de contours. De plus, comme illustré en figure 2.2, les pixels centraux sont faiblement pondérés et ceux plus éloignés ont la même

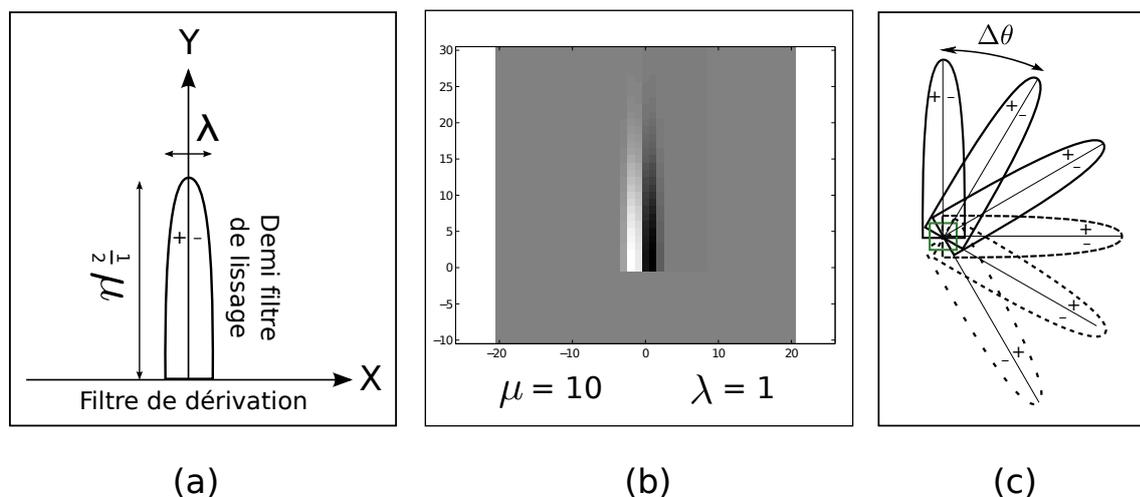


FIGURE 2.3 – Un demi-filtre gaussien orienté : (a) Opérateur de contours (*Filtre de dérivation de long de l'axe des X et demi-filtre de lissage de long de l'axe des Y*). (b) Exemple de filtre discrétisé. (c) Filtres orientés avec un angle de  $\Delta\theta$ .

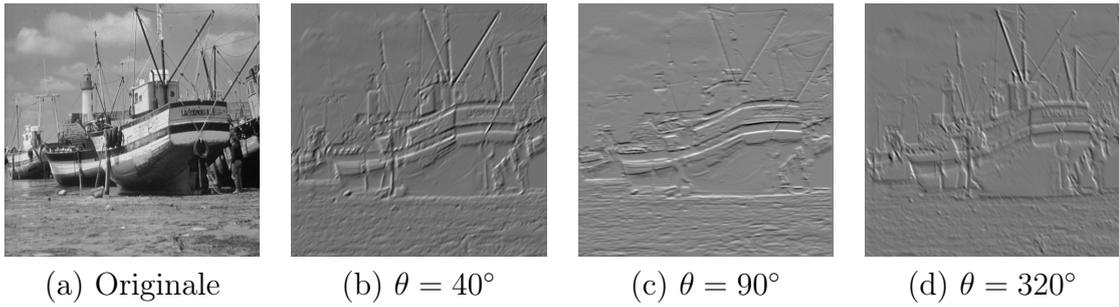


FIGURE 2.4 – Image et ses dérivées normalisées à différentes orientations  $\theta$ , avec  $\mu = 5$  et  $\lambda = 1$ .

influence contrairement à la gaussienne qui est une solution optimale pour les critères de Canny [Can86].

Nous proposons d'utiliser des demi-filtres orientés issus de l'équation 2.1, l'idée est de "couper" en deux parties le noyau gaussien, ce qui revient à obtenir deux demi-noyaux dans deux directions (figure 2.1 (b)). Comme nous n'avons besoin seulement que de la partie causale du filtre comme illustré en figure 2.3 (a) et (b) (nous reviendrons dans la section suivante à l'implémentation de ce filtre), cette opération correspond à la fonction échelon (ou marche)  $H$  :

$$G_{(\mu,\lambda)}(x, y, \theta) = H \left( R_\theta \begin{pmatrix} x \\ y \end{pmatrix} \right) \cdot g_{(\mu,\lambda)}(x, y, \theta). \quad (2.2)$$

Ainsi, nous obtenons un filtre dépendant seulement de  $\theta$  (voir figure 2.3(c)). Par convolution de l'image  $I$  avec ces noyaux rotatifs, (voir Figure 2.3(b)), nous obtenons une collection d'images lissées directionnelles et dérivées  $I_\theta = I * G_{(\mu,\lambda)}(\theta)$ .

Pour un temps de calcul optimum, nous procédons dans un premier temps à la rotation de l'image dans des orientations discrètes de 0 à 360 degrés et d'incrément  $\Delta\theta = 1, 2, 5$ , ou 10 degrés (tout dépend de la précision angulaire désirée). Les images tournées sont ensuite dérivées par le même filtre en équation 2.2 qui lui, reste fixe et qui contient toujours les mêmes paramètres  $\lambda$  et  $\mu$ , ce qui revient à appliquer le filtre suivant :

$$\mathcal{G}_{(\mu,\lambda)}(x, y) = H(-y) \cdot \frac{\partial g_{(\mu,\lambda)}(x, y, 0)}{\partial x} \quad (2.3)$$

$$= C_1 \cdot H(-y) \cdot x \cdot e^{-\left( \frac{x^2}{2\lambda^2} + \frac{y^2}{2\mu^2} \right)} \quad (2.4)$$

où  $C_1$  représente un coefficient de normalisation.

Comme l'image est tournée et non le filtre, l'implémentation du filtrage peut être utilisée par approximation récursive du filtre gaussien [Der92] (voir section 6.3 du premier chapitre). Dans un second temps, nous appliquons une rotation inverse de l'image et obtenons une banque de  $360/\Delta\theta$  images (quelques exemples d'images dérivées par ce demi-filtre à différentes orientations sont présentées en Figure 2.4).

En effet, en chaque pixel de coordonnées  $(x, y)$ , un filtre de dérivation est appliqué pour obtenir une information dérivative appelée  $\mathcal{Q}(x, y, \theta)$  en fonction de l'orientation  $\theta$  :

$$\mathcal{Q}(x, y, \theta) = I_{R_\theta} * \mathcal{G}_{(\mu, \lambda)}(x, y) \quad (2.5)$$

$$= I_{R_\theta} * C_1 \cdot H(-y) \cdot x \cdot e^{-\left(\frac{x^2}{2\lambda^2} + \frac{y^2}{2\mu^2}\right)} \quad (2.6)$$

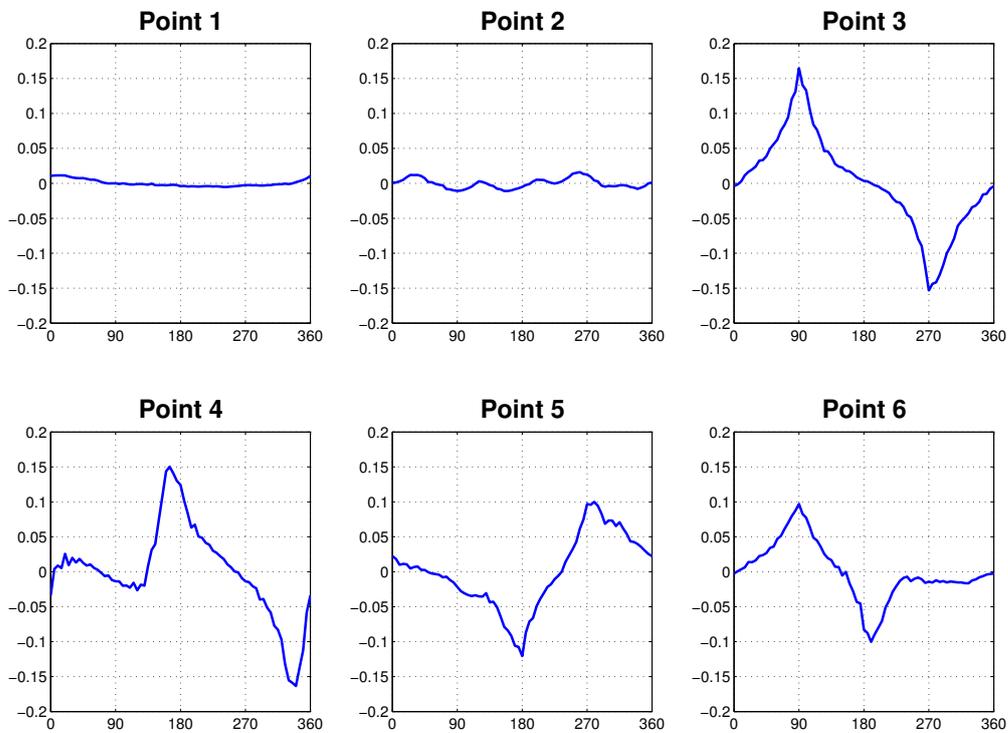
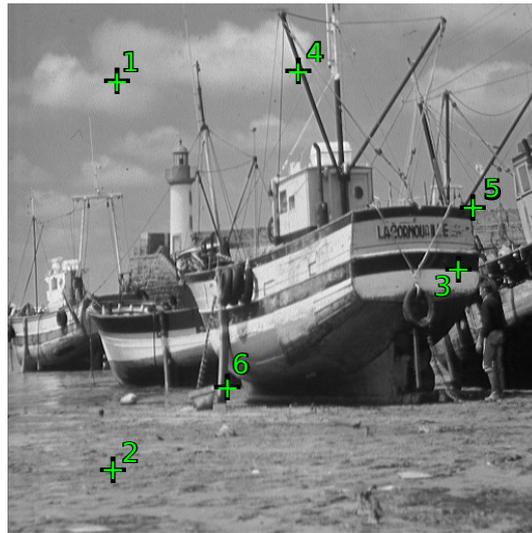


FIGURE 2.5 – Sélection de points dans l'image du haut et les fonctions  $\mathcal{Q}(x, y, \theta)$  associées avec  $\Delta\theta = 5^\circ$ ,  $\mu = 5$  et  $\lambda = 1$ . L'axe des X représente l'orientation du filtre et l'axe des Y indique la réponse du filtre.

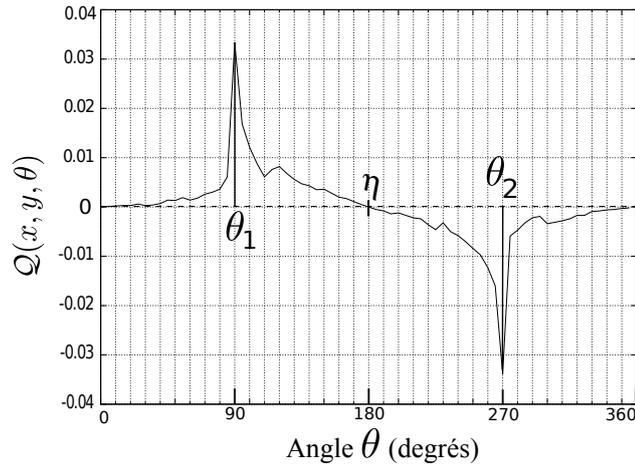


FIGURE 2.6 – Exemple de fonction  $\mathcal{Q}(x, y, \theta)$ .

où  $I_{R_\theta}$  correspond à la rotation de l'image originale d'orientation  $\theta$ .

Comme schématisé en figure 2.7,  $\mathcal{Q}(x, y, \theta)$  représente la pente d'une ligne issue d'un pixel dans la direction  $\theta$ , i.e. la direction perpendiculaire à cette même ligne.

Quelques exemples de fonctions  $\mathcal{Q}(x, y, \theta)$  sont montrés à partir d'une sélection de différents types de points en figure 2.5. Dans la direction des différents contours, le filtre est performant et cela se traduit par des pics positifs ou négatifs dans la fonction  $\mathcal{Q}(x, y, \theta)$ . En effet, les points 3 et 4 appartiennent à deux contours respectivement oblique et horizontal, les deux pics dans chacun des signaux  $\mathcal{Q}(x, y, \theta)$  indiquent l'orientation du contour en ces deux points. Il en va de même pour les points appartenant à des coins (les points 5 et 6), les pics indiquent également les directions des contours formant un chemin passant par ces pixels. En revanche, les pixels appartenant à une région homogène (voir point 1), le signal  $\mathcal{Q}(x, y, \theta)$  ne contient aucun pic. En ce qui concerne les points de texture (point 2), le signal est relativement stochastique mais son amplitude reste faible comparée aux points de contours.

Maintenant, obtenir un gradient  $\|\nabla I\|$  et sa direction associée  $\eta$  devient un problème d'optimisation par recherche de maxima extrêmement simple. Tout d'abord,

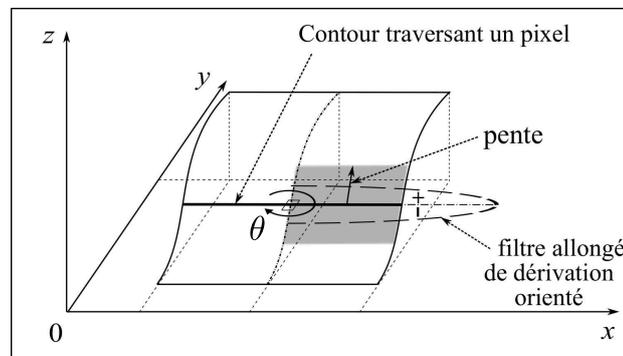


FIGURE 2.7 – Estimation de la pente autour d'un pixel. L'axe des  $z$  représente l'intensité lumineuse.

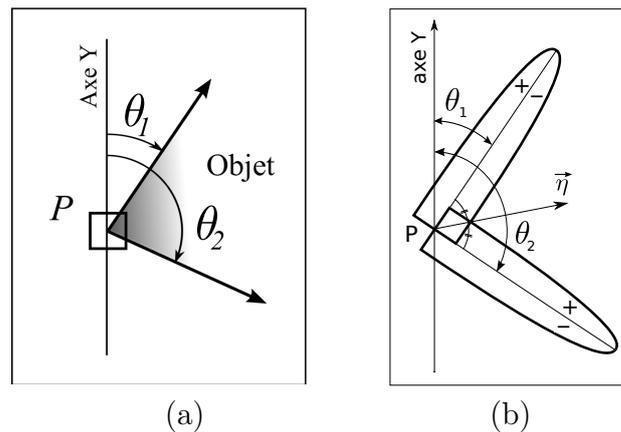


FIGURE 2.8 – Calcul de  $\eta(x, y)$  à partir de  $\theta_1$  et  $\theta_2$ . (a)  $\theta_1$  et  $\theta_2$  correspondent aux deux directions d'un contour d'objet au niveau d'un pixel  $P$ . (b) L'angle  $\eta$  est la bissectrice de  $\theta_1$  et  $\theta_2$ .

pour chaque pixel  $P$  de coordonnées  $(x, y)$ , nous calculons les extrema locaux de la fonction  $\mathcal{Q}(x, y, \theta)$  aux directions  $\theta_1$  et  $\theta_2$  (illustrés en figure 2.6).  $\theta_1$  et  $\theta_2$  définissent une courbe traversant le pixel (avec un chemin entrant et un chemin sortant). Puis ces deux extrema (le maximum et le minimum) sont combinés ensemble pour maximiser  $\|\nabla I\|$  :

$$\theta_1 = \arg \max_{\theta \in [0, 360[} (\mathcal{Q}(x, y, \theta)), \quad (2.7)$$

$$\theta_2 = \arg \min_{\theta \in [0, 360[} (\mathcal{Q}(x, y, \theta)), \quad (2.8)$$

$$\|\nabla I\| = \mathcal{Q}(x, y, \theta_1) - \mathcal{Q}(x, y, \theta_2). \quad (2.9)$$

Nous estimons simplement  $\eta$  (voir figure 2.8 et 2.6 ) par la bissectrice formée par l'angle entre  $\theta_1$  et  $\theta_2$  :

$$\eta = \frac{\theta_1 + \theta_2}{2}.$$

Après avoir calculé  $\|\nabla I\|$  et  $\eta$ , les contours peuvent être extraits en calculant les maxima locaux de  $\|\nabla I\|$  dans la direction  $\eta$  suivis par un seuillage par hystérésis.

La figure 2.9 illustre le calcul de différents gradients  $\|\nabla I\|$  et d'angles  $(\theta_1, \theta_2, \eta)$  avec différents paramètres  $\lambda$  et  $\mu$ . Même en choisissant un filtre allongé (i.e. la valeur de  $\mu$  est grande) et fin (i.e. paramètre  $\lambda < 2$ ), les contours sont détectés précisément au niveau des bords et des coins des objets :

- un paramètre  $\mu$  relativement long (i.e.  $\mu > 5$ ) dans le sens du lissage permet d'atténuer l'effet du bruit qui pourrait délocaliser la détection de bords,
- l'utilisation d'un paramètre  $\lambda < 2$  pour la dérivée localise avec précision les contours (voir figures 2.9 (j) et (l)).

Nous montrerons différents résultats dans les sections 2.7 et 2.9 puis nous adapterons ce filtre aux images vectorielles. Enfin, nous testerons la robustesse de notre détecteur de contours en fonction du niveau de bruit dans l'image.

### 2.3 Implémentation du filtre

Concernant la détection de contours dans les images, il a été établi [Can86] que la gaussienne et ses dérivées forment un filtre optimal dans le sens rapport signal sur bruit (RSB), localisation et contour unique (voir section 6 du chapitre 1). Grâce à leur invariance euclidienne et l'émergence de la théorie de l'espace échelle, ces filtres ont reçu une grande attention en détection de bords. Pourtant, plusieurs autres filtres basés sur des exponentielles préservant aussi des structures optimales ont été étudiés pour la détection de contours [Der87], [SC92]. Ces filtres ne présentent pas d'invariance isotropique (contrairement à la gaussienne [Can86] [MH80]), cependant ils sont intéressants pour leur faible complexité calculatoire. Le filtre de Shen-Castan peut être implémenté récursivement à l'ordre 1 [SC92], celui de Deriche à l'ordre 2 [Der87] tandis que le filtre approximant la gaussienne l'est lui à l'ordre 4 [Der92].

Dans notre cas, premièrement nous ne sommes pas concernés par la propriété isotropique du filtre. Deuxièmement, le meilleur RSB du filtre gaussien par rapport aux filtres exponentiels n'est pas fondamental comme nous allons utiliser un écart type de la gaussienne  $\mu$  très large dans le sens du lissage. Dans tous les cas, qu'il s'agisse du filtre gaussien ou exponentiel, ce lissage important supprimera le bruit. Nous pouvons utiliser l'un de ces trois filtres pour le lissage [Der87], [SC92] [Der92], tout dépend de la complexité désirée. Au contraire, pour la dérivation, nous allons privilégier la première dérivation de la gaussienne d'écart type  $\lambda$  considérant son RSB.

Il est bien connu que pour implémenter un filtre récursif, nous avons besoin de décomposer la fonction du filtre en une partie causale et une anti-causale. Ici, contrairement aux implémentations décrites dans [Der87], [SC92] [Der92] (voir annexes 1, 2 et 3), nous avons besoin de demi-filtres. La décomposition du filtre doit être symétrique dans la mesure où nous souhaitons compter pour moitié le pixel central dans le chemin entrant et dans le chemin sortant. Le but étant d'assurer la normalisation de la composition des deux filtres. Le filtre de lissage est donc décomposé de la manière suivante :

$$F(x) = F^+(x) + F^-(x)$$

où :

$$F^+(x) = \begin{cases} F(x) & \text{if } x > 0 \\ \frac{1}{2}F(x) & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (2.10)$$

et

$$F^-(x) = \begin{cases} 0 & \text{if } x > 0 \\ \frac{1}{2}F(x) & \text{if } x = 0 \\ F(x) & \text{if } x < 0. \end{cases} \quad (2.11)$$

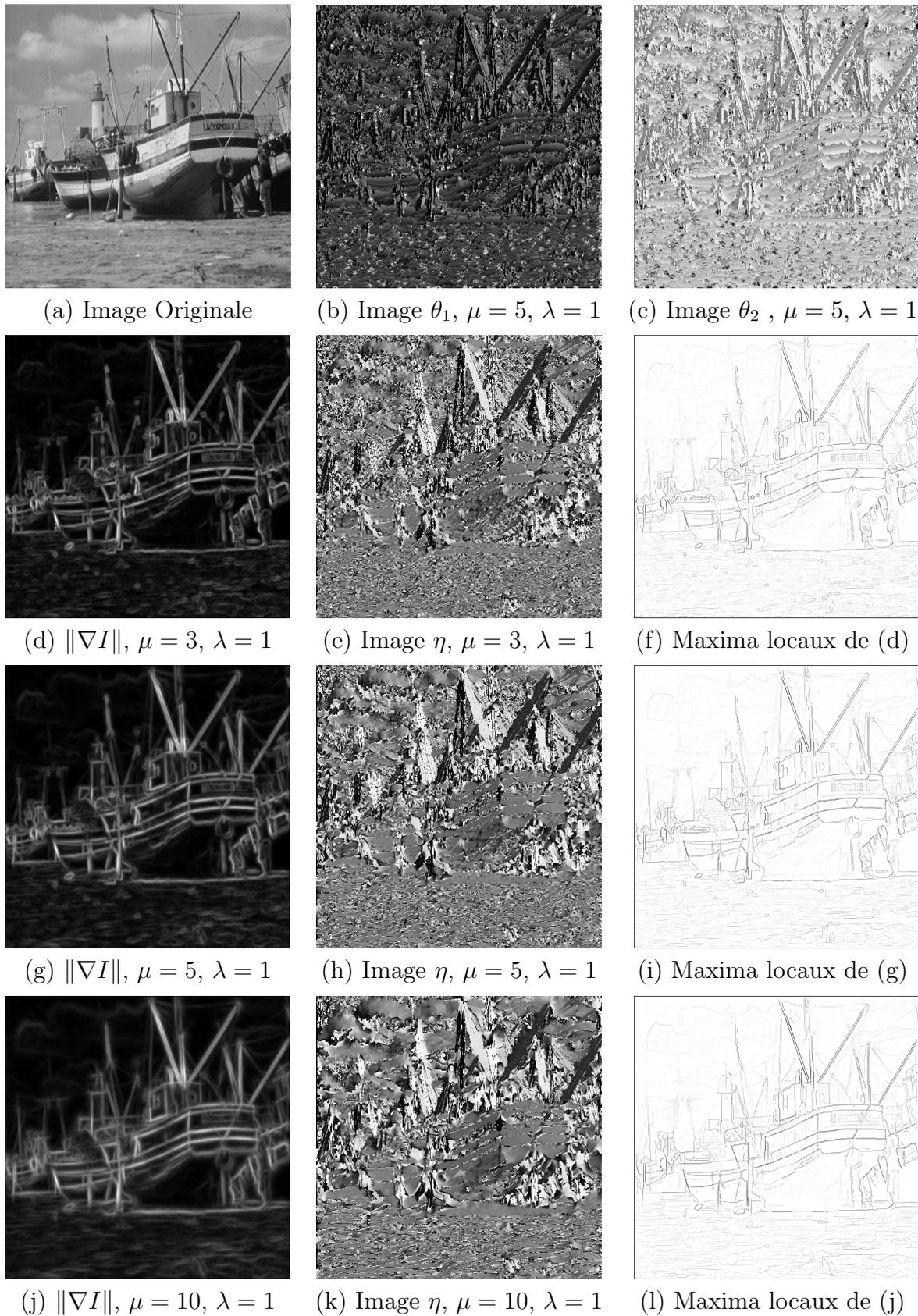


FIGURE 2.9 – Estimation du gradient  $\|\nabla I\|$  et des angles  $\theta_1$ ,  $\theta_2$  et  $\eta$  avec différents paramètres  $\mu$  et  $\lambda$ . Toutes ces images sont normalisées et  $\Delta\theta = 5^\circ$ .

Seul le filtre de l'équation 2.10, représentant le demi-filtre sera implémenté, la partie anti-causale étant assurée par la rotation du filtre. Si le filtre choisi est le filtre gaussien, les paramètres considérés seront  $\mu$  pour le filtre de lissage et  $\lambda$  pour celui de dérivation (équation 2.2 et figure 2.3). Notons que dans le cas de l'utilisation du filtre de Shen-Castan comme filtre de lissage, l'écart type utilisé sera  $\lambda = 1/\sqrt{2\alpha}$ .

## 2.4 Performances du filtre anisotrope sur des contours virtuels

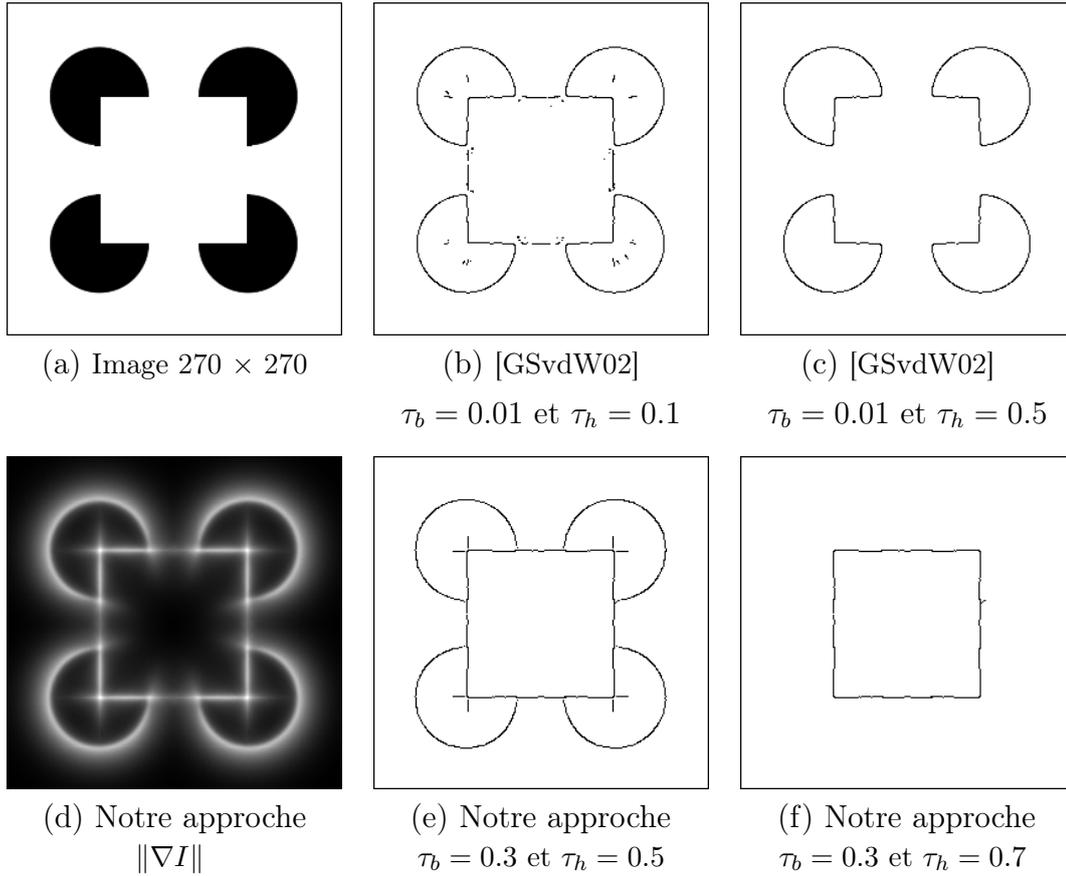


FIGURE 2.10 – Extraction de contours d'un carré perceptuel. Résultat de notre détecteur de contours anisotrope en (d), (e) et (f) et comparaison avec la méthode de [GSvdW02]. L'image en (d) est normalisée.

Certains contours ne sont pas explicitement dessinés sur une image, notre système de perception est capable de deviner la présence de contours dits perceptuels. Cette détection de contours discontinus est réellement utile dans les images réelles puisque les bords des objets peuvent être flous à certains endroits ou perturbés par un bruit ou une ombre. Dans ce paragraphe, nous démontrons que notre détecteur de contours anisotrope est capable de reconstituer ces contours. En effet, dans les figures 2.10 et 2.11, il existe un seuillage par hystérésis bas  $\tau_b$  et haut  $\tau_h$  tels que les contours virtuels soient détectés avec notre détecteur anisotrope.

Dans la figure 2.10, nous avons utilisé les paramètres  $\mu = 25$ ,  $\lambda = 1$  et  $\Delta\theta = 5^\circ$ . Nous avons comparé nos résultats avec ceux du détecteur de contours anisotrope gaussien [GSvdW02] en utilisant les mêmes paramètres.

La figure 2.10 (d) indique que notre détecteur de contours anisotrope est capable de détecter des points de contours virtuels et la figure 2.10 (e) montre que les contours perceptuels peuvent être extraits. Par contre le détecteur de contours anisotrope gaussien [GSvdW02] n'est pas adapté à ce type de contours virtuels puisqu'au niveau des coins formés par les 3/4 de disques et les carrés perceptuels, les orientations ne sont pas correctes donc le carré devient discontinu, même avec un seuil très bas ( $\tau_b = 0.01$ ).

Le résultat de notre approche est rendu possible en utilisant un filtre relativement long ( $\mu = 25$ ). Même si notre détecteur est performant pour détecter les 3/4 de cercles, cependant, notre filtre est plus performant pour les contours linéaires, donc en prenant un seuil haut  $\tau_h = 0.7$ , il est possible de ne détecter seulement que le carré (figure 2.10 (f) ). Afin d'obtenir un filtre encore plus performant, il faudrait que les demi-noyaux soient de forme "banane" [KPP], mais ce filtre perdrait alors sa récursivité, d'où une implémentation coûteuse.

Nous pouvons aussi noter dans la figure 2.10 (d) que les coins ressortent plus que les contours, il serait possible de les extraire avec un simple seuillage sur cette image.

L'avantage de notre détecteur de contours anisotrope est que sa réponse dépend de deux directions. Il est donc possible d'extraire des contours linéaires virtuels comme dans la figure 2.10 mais aussi des contours contenant des coins comme illustré en figure 2.11.

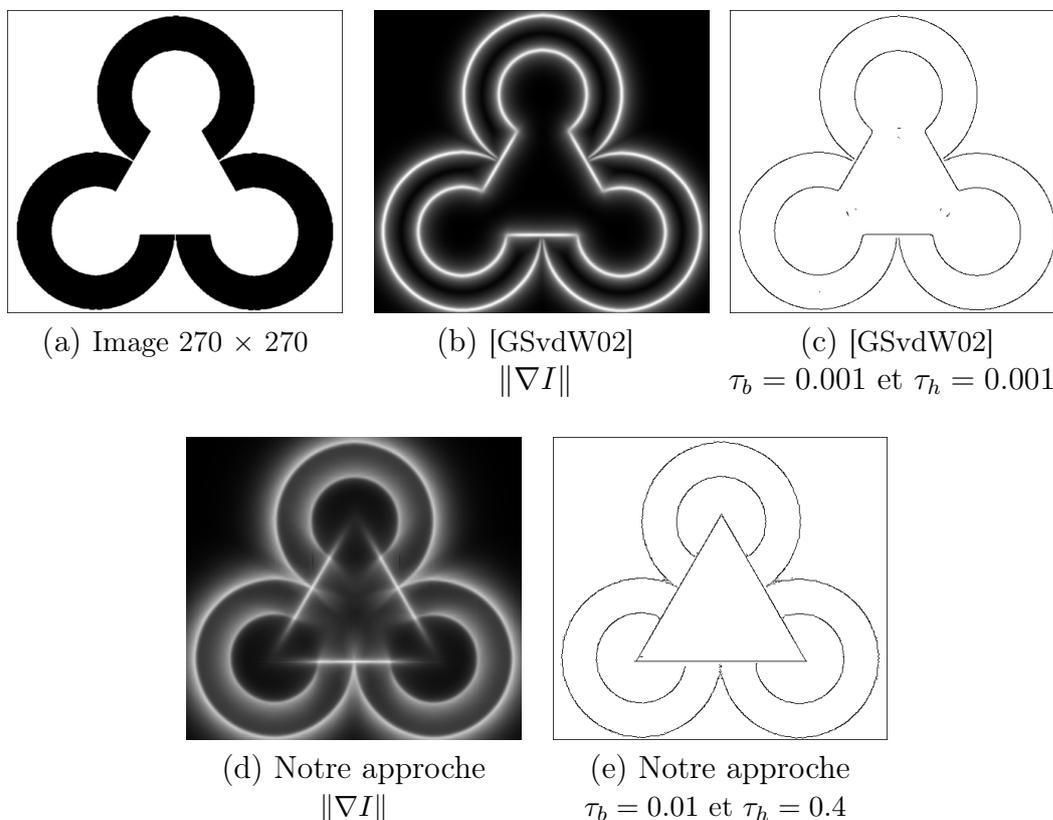


FIGURE 2.11 – Extraction de contours d'un carré perceptuel. Résultat de notre détecteur de contours anisotrope en (d) et (e) et comparaison avec la méthode de [GSvdW02]. Les images en (b) et (d) sont normalisées.

En effet, tout comme le carré perceptuel, en utilisant un filtre long (ici  $\mu = 30$ ), il existe un seuillage bas pour extraire parfaitement le triangle et ses coins. Le détecteur anisotrope gaussien n'est pas adapté à ce type de contours virtuels puisque qu'il n'est composé que d'un seul noyau orienté. L'image de l'estimation des gradients en est la parfaite illustration (figure 2.11 (b) ), il n'y a pas de gradient détecté au niveau des extrémités du triangle perceptuel. De plus, les directions des contours ne sont pas optimales au niveau des coins. Nous pouvons le remarquer des coins formés par des cercles, ces derniers ne sont pas extraits correctement, ils sont localisés vers l'intérieur du coin, ce qui n'est pas le cas avec notre détecteur. Donc, si un gradient avait été détecté au niveau du triangle virtuel, l'extraction de contours dans la direction du gradient n'aurait pas été correcte pour ces points ; tout comme le carré en figure 2.10, il y aurait eu des traits discontinus contrairement aux résultats obtenus avec notre détecteur.

## 2.5 Résultats sur des images réelles scalaires

Les images réelles en niveau de gris sont souvent composées d'objets de différentes tailles et de textures sur ou autour de ces mêmes objets.

Les différents résultats montrés en figure 2.12 ne sont pas trop perturbés par la texture, mis à part la plage sur le bas de l'image. Les contours extraits à l'aide d'un seuillage par hystérésis sont présentés en fonction des meilleurs contours visuellement désirés (bord des objets) avec le moins possible de contours indésirables (bruit, texture, réflexions spéculaires...). Les détecteurs de contours isotropes<sup>1</sup> [SC92] [Der87] [Der92] et notre détecteur anisotrope extraient les petits objets contrairement au détecteur anisotrope gaussien [GSvdW02] qui détecte seulement très bien les structures linéaires. Ainsi, nous pouvons remarquer notamment que le détecteur anisotrope gaussien de contours possède des difficultés pour extraire les mats du bateau de gauche.

Lorsque les images sont composées majoritairement de texture comme en figure 2.13 (a), les résultats des détecteurs de contours isotropes sont pollués par les contours de texture et la tour n'est pas détectée entièrement. En revanche les détecteurs de contours anisotropes extraient la tour entièrement. Les figures 2.13 (f) et (g) illustrent que les gradients calculés par notre détecteur de contours anisotrope au niveau de la tour est plus important que celui au niveau des branches. Cela montre qu'il existe un seuil permettant d'extraire entièrement la tour tout en restant précis au niveau des bords pour un minimum de contours indésirables dus aux contours de la texture (figures 2.13 (h) et (i) ).

Les résultats dans les figures 2.12 et 2.13 montrent que notre détecteur anisotrope de bords est capable d'extraire aussi bien les contours linaires perturbés par des textures que les petits objets. Le fait d'avoir construit un filtre fin et long basé sur deux noyaux orientés indépendants permet de garder la précision des détecteurs de bords isotropes avec un écart type petit (par exemple  $\sigma < 1$  dans le cas de la gaussienne) tout en gardant les atouts du détecteur anisotrope gaussien.

---

1. Ces filtres sont une approximation de filtres isotropes, nous les appelons isotropes afin de les différencier avec les filtres anisotropes orientés présentés en partie 9.2 du premier chapitre.

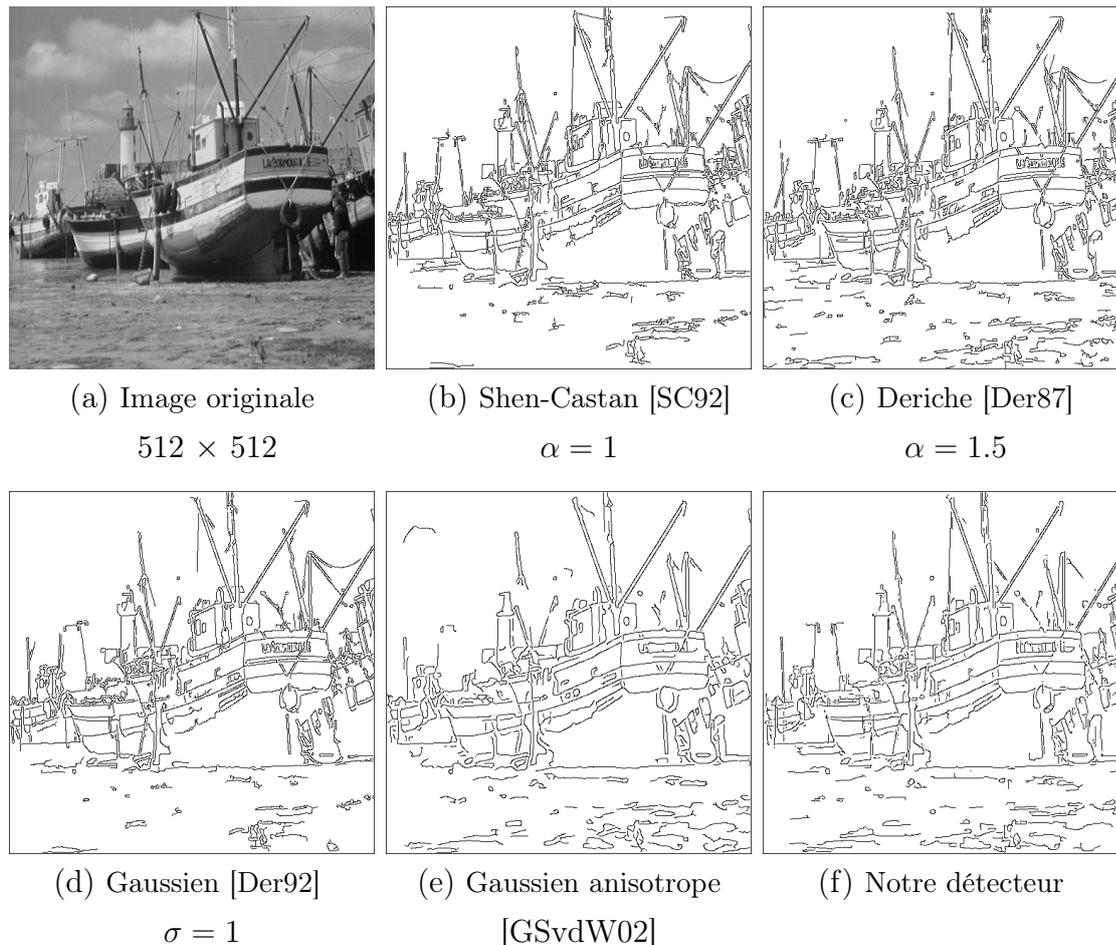


FIGURE 2.12 – Extraction de contours sur une image réelle. Résultat de notre détecteur de contours anisotrope en (f) et comparaison avec d’autres méthodes. Les paramètres utilisés pour notre détecteur de contours anisotrope et le détecteur gaussien anisotrope sont  $\mu = 5$ ,  $\lambda = 1$  et  $\Delta\theta = 5^\circ$ . Nous avons utilisé un seuillage par hystérésis pour extraire les contours : (b)  $\tau_b = 0.02$  et  $\tau_h = 0.08$ , (c)  $\tau_b = 0.02$  et  $\tau_h = 0.05$ , (d)  $\tau_b = 0.02$  et  $\tau_h = 0.08$ , (e)  $\tau_b = 0.1$  et  $\tau_h = 0.2$ , (f)  $\tau_b = 0.05$  et  $\tau_h = 0.12$ .

## 2.6 Adaptation du filtre à la couleur

Une image couleur  $\mathbf{I}$  est composée de plusieurs canaux comme déjà abordé dans le chapitre 1. Détaillé dans le chapitre 1 en section 4.4, le tenseur de structure permet une extension des détecteurs de contours classiques aux images vectorielles [DZ86]. Similairement, Knossow et al. ont développé le détecteur anisotrope de bords gaussien en couleur [KvdWHR07].

Le détecteur anisotrope de contours décrit en partie 2.2 de ce chapitre peut être implémenté sur des images scalaires mais il peut être adapté séparément aux différents plans formant une image couleur. En effet, comme ce filtre ne calcule pas de dérivées dans les directions  $X$  et  $Y$ , nous n’avons pas besoin d’utiliser un tenseur pour étendre la détection de contours aux images couleurs. Alors, nous calculons la plus grande valeur du gradient  $\|\nabla\mathbf{I}\|$  en utilisant la norme max qui est la norme la plus simple pour la couleur. En notant  $d$  le nombre de canaux d’une image couleur  $\mathbf{I}$  et  $I^i$  sa  $i^{me}$  composante ( $1 \leq i \leq d$ ), son gradient  $\|\nabla\mathbf{I}\|$  est calculé en chaque pixel

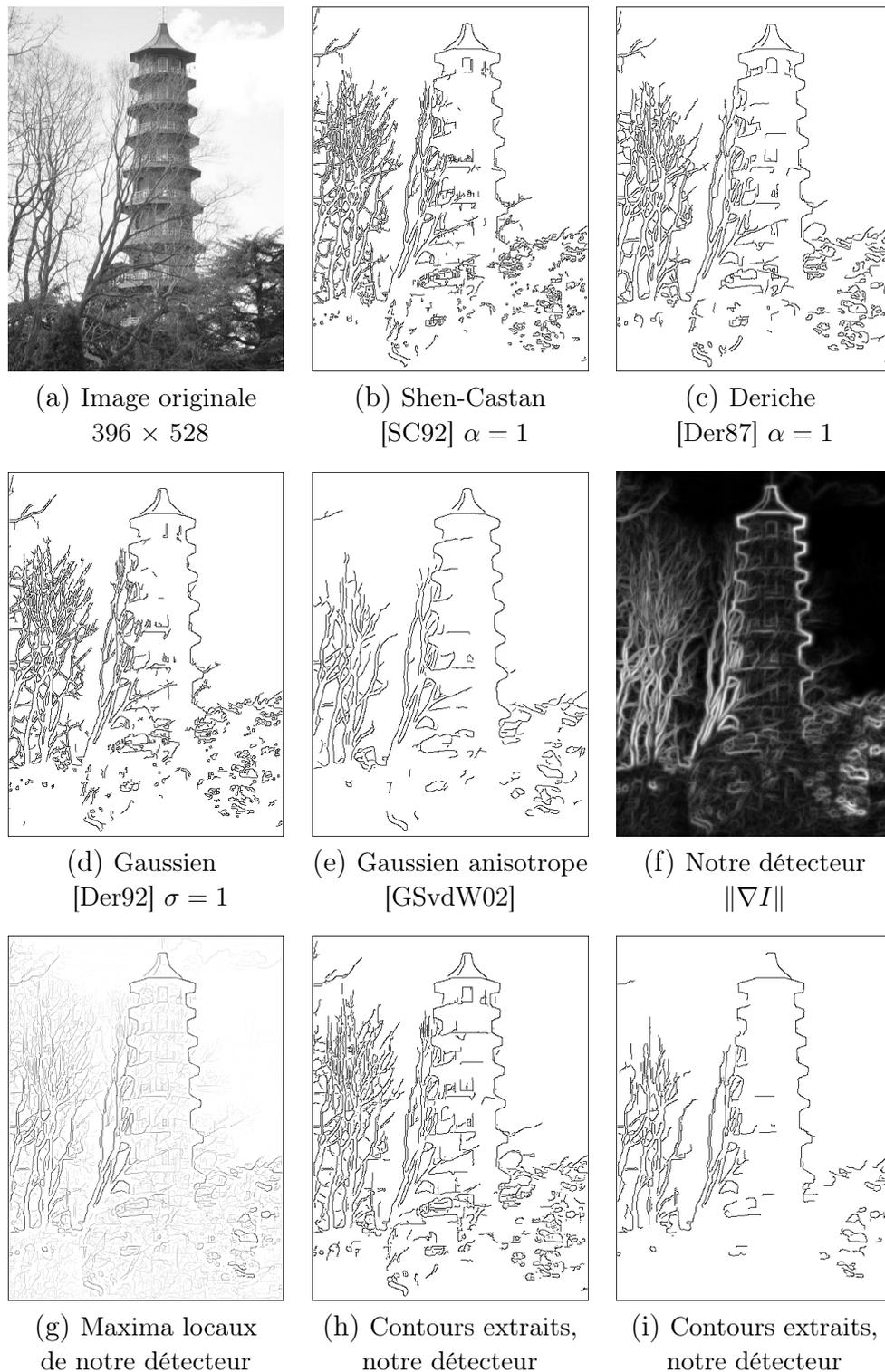


FIGURE 2.13 – Extraction de contours sur une image réelle avec une texture de branches. Les paramètres utilisés pour notre détecteur de contours anisotrope et le détecteur gaussien anisotrope sont  $\mu = 5$ ,  $\lambda = 1$  et  $\Delta\theta = 5^\circ$ . La figure en (f) illustre le gradient estimé avec notre détecteur et celle en (g) le calcul des maxima locaux (images normalisées). Nous avons utilisé un seuillage par hystérésis pour extraire les contours : (b)  $\tau_b = 0.05$  et  $\tau_h = 0.1$ , (c)  $\tau_b = 0.02$  et  $\tau_h = 0.05$ , (d)  $\tau_b = 0.05$  et  $\tau_h = 0.1$ , (e)  $\tau_b = 0.2$  et  $\tau_h = 0.3$ , (h)  $\tau_b = 0.1$  et  $\tau_h = 0.15$ , (i)  $\tau_b = 0.2$  et  $\tau_h = 0.26$ .

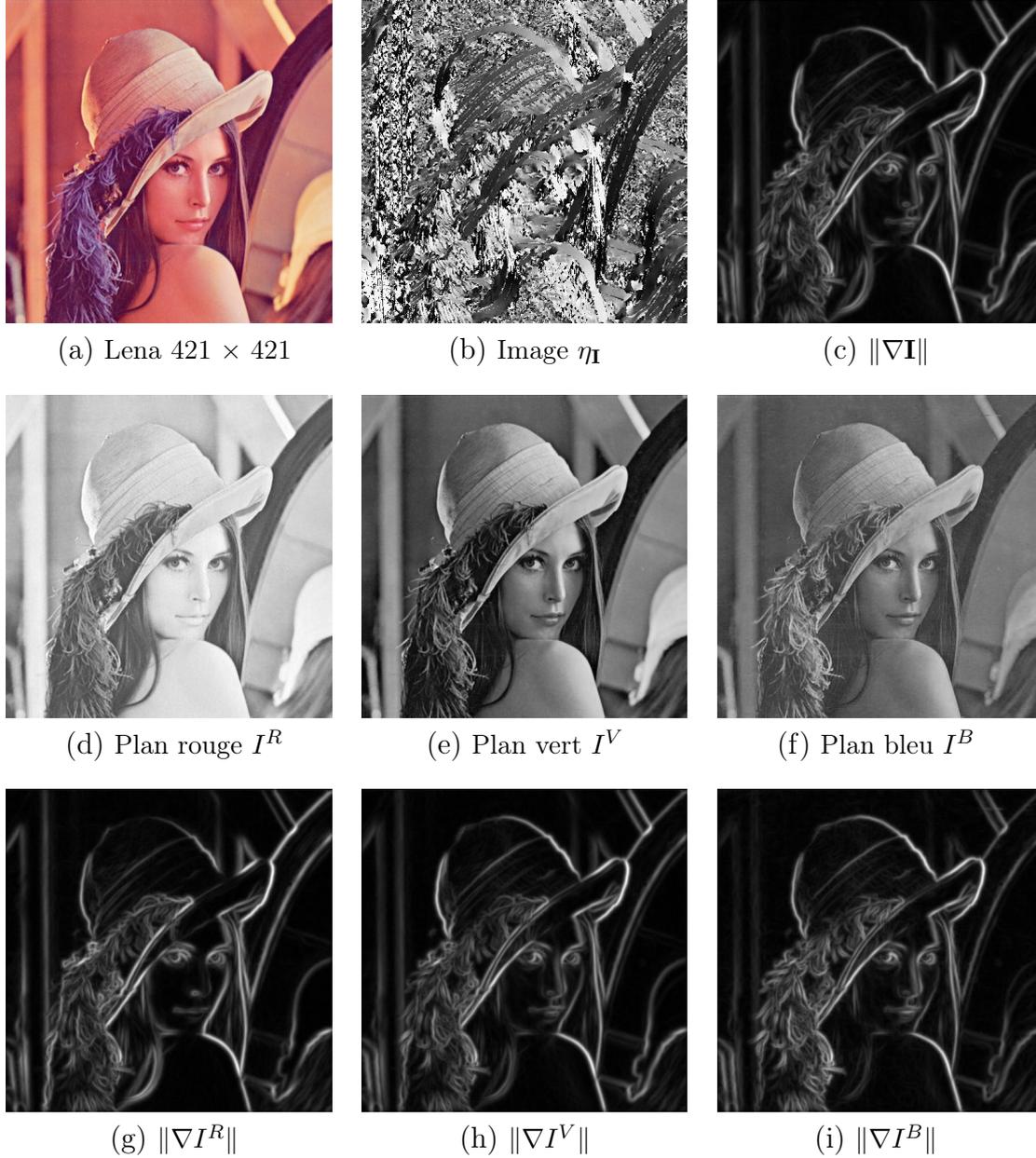


FIGURE 2.14 – Estimation du gradient avec notre détecteur de contours anisotrope sur les différents canaux d'une image réelle couleur. Cette image contient 3 plans : rouge, vert et bleu notés  $(I^R, I^V, I^B)$ . Les paramètres utilisés pour notre détecteur de contours anisotrope et le détecteur gaussien anisotrope sont  $\mu = 5$ ,  $\lambda = 1$  et  $\Delta\theta = 5^\circ$ . Les images (b), (c), (g), (h) et (i) sont normalisées.

de coordonnées  $(x, y)$  par :

$$\|\nabla \mathbf{I}(x, y)\| = \max_{1 \leq i \leq d} (\|\nabla I^i(x, y)\|), \quad (2.12)$$

et

$$\eta_{\mathbf{I}}(x, y) = \arg \max_{\eta^i(x, y), 1 \leq i \leq d} (\|\nabla I^i(x, y)\|), \quad (2.13)$$

où  $\eta^i(x, y)$  est la direction du gradient du pixel  $(x, y)$  associé au canal  $i$  de  $\mathbf{I}$ . Après cette étape, les contours peuvent être extraits en calculant les maxima locaux de  $\|\nabla \mathbf{I}(x, y)\|$  dans la direction  $\eta_{\mathbf{I}}(x, y)$ . La figure 2.14 illustre l'estimation du gradient et de sa direction  $\eta_{\mathbf{I}}$  sur une image couleur et ses différents canaux.

Dans le paragraphe suivant, nous montrons un résultat sur l'image Lena et nous le comparons avec celui d'autres méthodes. Ensuite, nous détaillerons une méthode qui améliore la détection de contours pour les images réelles en utilisant les courbes gamma. Nous comparerons les résultats avec la méthode présentée jusqu'à maintenant puis nous évaluerons ces approches avec des résultats quantitatifs.

## 2.7 Résultats sur une image couleur

Nous avons appliqué notre détecteur sur l'image bien connue de Lena (figure 2.15) et comparé notre résultat avec les détecteurs de contours classiques ainsi que le détecteur de contours anisotropique gaussien.

Contrairement au détecteur anisotrope de contours gaussien, notre filtre est très précis pour les petits objets ou détails (yeux, bouche, cheveux...) comme les approches classiques isotropes. Cependant, notre détecteur de contours est plus performant que les détecteurs classiques pour détecter les contours des structures à l'arrière plan, contrairement aux détecteurs de contours classiques. L'extracteur de contours anisotropique gaussien est performant pour détecter les contours en arrière plan mais les détails de Lena ne sont pas extraits.

Nous montrerons d'autres résultats dans les sections suivantes, dans lesquelles nous améliorons cette méthode.

## 2.8 Performance temporelle

Le processus est divisé en deux étapes : dérivation puis estimation du gradient. Au niveau de l'étape de dérivation, les images dérivées sont calculées et écrites sur le disque dur. Pour l'estimation du gradient, les images dérivées sont rechargées, pour estimer les deux directions du contour en chaque pixel, puis le gradient et sa direction sont écrits sur le disque. Seulement l'opération de dérivation est parallélisée sur les 8 processeurs.

Le code de notre détecteur anisotrope est implémenté en C++ sous Linux. Pour tester notre détecteur avec une dérivée et un demi-lissage gaussiens, nous avons utilisé une plateforme composée d'un processeur de huit coeurs tournant à 2.6 GHz. Pour analyser une image de  $256 \times 256$  pixels en niveaux de gris avec une discrétisation de 5 degrés, pour une performance temporelle de 1.3 secondes.



FIGURE 2.15 – Extraction de contours sur une image couleur réelle. Les paramètres utilisés pour notre détecteur de contours anisotropique et le détecteur gaussien anisotropique sont  $\mu = 5$ ,  $\lambda = 1$  et  $\Delta\theta = 5^\circ$ . Nous avons utilisé un seuillage par hystérésis pour extraire les contours : (b)  $\tau_b = 0.01$  et  $\tau_h = 0.05$ , (c)  $\tau_b = 0.01$  et  $\tau_h = 0.05$ , (d)  $\tau_b = 0.001$  et  $\tau_h = 0.1$ , (e)  $\tau_b = 0.01$  et  $\tau_h = 0.2$ , (f)  $\tau_b = 0.01$  et  $\tau_h = 0.12$ .

Une amélioration temporelle pourrait être apportée en partie avec la parallélisation de l'estimation du gradient puis en évitant de sauvegarder et de recharger les images dérivées. L'espace mémoire pour le processus entier est petit et le détecteur n'est pas limité par la taille de l'image. Par exemple, les contours d'une image de taille  $1000 \times 1000$  peuvent être facilement estimés avec une plateforme de 2 GB de RAM.

Une autre amélioration de la méthode serait de réduire le temps pour la rotation des images. En effet, il existe des approches qui estiment la rotation des images récursivement [UTY95] [CVDV08], ce qui apporterait un gain de temps de calcul.

## 2.9 Détection anisotrope de contours utilisant les courbes gamma

### Correction gamma

La correction gamma est le nom d'une opération non linéaire qui modifie les intensités des pixels dans une image [Poy96]. En effet, dans le cas d'images sur-exposées, les corrections gamma peuvent être utilisées pour assombrir les images et inversement pour des images sous exposées, elles peuvent mener à éclaircir les images. Des idées similaires sont utilisées pour les images HDR (*High Dynamic Range images*) [FLW02] mais pas en détection de contours. La correction gamma est définie comme suit :

$$I_{out}(P) = (I(P))^\gamma, \quad \gamma > 0 \quad \text{et} \quad I(P) \in [0, 1], \quad (2.14)$$

où  $I(P)$  et  $I_{out}(P)$  sont respectivement l'entrée et la sortie des intensités normalisée des pixels. La figure 2.16 montre le comportement des courbes gamma en fonction de différentes valeurs de  $\gamma$ . Une valeur de  $\gamma > 1$  assombrira l'image alors qu'une valeur  $\gamma < 1$  l'éclaircira. Un exemple de correction gamma sur une image est présenté en figure 2.17.

### Détection de contours anisotrope utilisant les courbes gamma

La figure 2.17 présente le problème de la détection de contours sans correction gamma. En effet, le résultat de gradient obtenu  $\|\nabla I\|$  montre que les gradients verticaux sont plus forts que les horizontaux. Cependant, avec les corrections gamma, les contours horizontaux deviennent plus clairs. Pour solutionner ce problème, nous proposons un nouveau détecteur de contours anisotrope utilisant les courbes gamma nommé ANEG (*ANisotropic Edge detector using Gamma correction*).

La solution est d'estimer un gradient avec la méthode de détection de contours anisotrope présentée en section 2.2 sur les trois images suivantes :  $I$ ,  $I^\gamma$  et  $I^{1/\gamma}$  puis de combiner ces trois informations obtenues.  $I$  représente l'image originale scalaire,  $I^\gamma$  représente  $I$  corrigée avec le paramètre  $\gamma$  et  $I^{1/\gamma}$  est  $I$  corrigée avec le paramètre  $1/\gamma$  ( $\gamma > 1$ ). Ces trois images  $I$ ,  $I^\gamma$  et  $I^{1/\gamma}$  sont considérées séparément comme trois

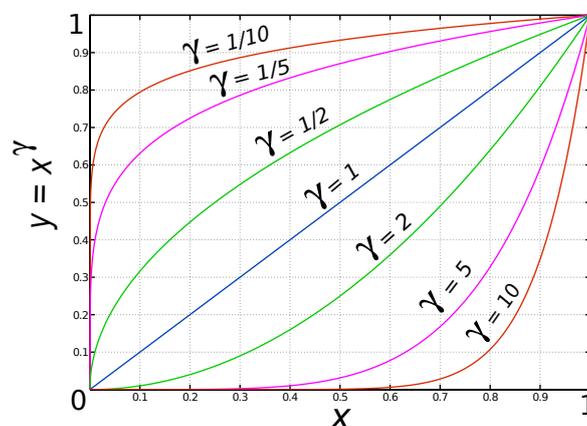


FIGURE 2.16 – Courbes gamma où  $X$  représente l'intensité d'un pixel normalisée.

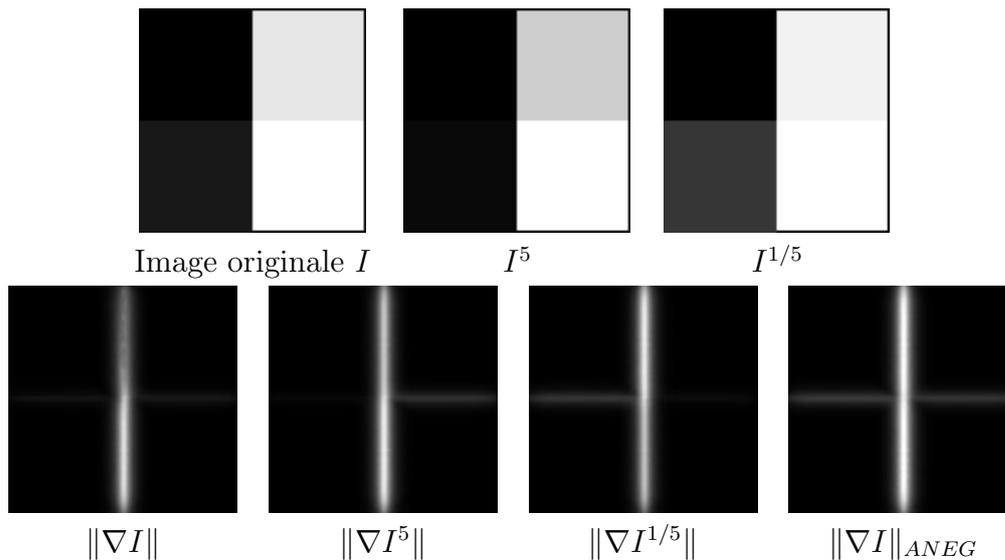


FIGURE 2.17 – Principe de l’ANEG : estimation d’un gradient sur 3 images  $I$ ,  $I^\gamma$ ,  $I^{1/\gamma}$  et sélection de la valeur du gradient la plus élevée appelée  $\|\nabla I\|_{ANEG}$ . Le contour horizontal est détecté plus aisément.

images en niveaux de gris. Puis similairement à notre approche adaptée à la couleur détaillée en partie 2.6 de ce chapitre, nous sélectionnons la plus grande valeur de gradient  $\|\nabla I\|_{ANEG}$  (illustré en figure 2.17). Nous les combinons à travers la formule suivante ;

$$\|\nabla I(x, y)\|_{ANEG} = \max(\|\nabla I(x, y)\|, \|\nabla I^\gamma(x, y)\|, \|\nabla I^{1/\gamma}(x, y)\|). \quad (2.15)$$

Dans la figure 2.18 (c), le pixel appartient à un bord d’un objet qui peut être considéré comme sur-exposé. Dans ce cas précis, d’après l’équation 2.15,  $\|\nabla I\|_{ANEG} = \|\nabla I^\gamma\|$ , avec  $\gamma = 5$ .

Les corrections gamma permettent de mieux distinguer et estimer les contours mais elles changent aussi les valeurs de pixels de bruits. Toutefois, notre détecteur de contours anisotrope est suffisamment allongé (paramètre  $\mu$ ) pour ne pas être perturbé par les corrections gamma au niveau du bruit. En effet, en présence de bruit, la valeur de  $\|\nabla I\|_{ANEG}$  est peu modifiée car contrairement à un pixel appartenant à un contour, (figure 2.18 (c)), les extrema locaux de  $\mathcal{Q}(x, y, \theta)$  ne sont que très peu amplifiés (voir figure 2.18 (b)).

Après cette étape, les contours sont extraits en calculant les maxima locaux de  $\|\nabla I\|_{ANEG}$  dans la direction  $\eta_{ANEG}$  :

$$\eta_{ANEG}(x, y) = \arg \max_{\eta_I(x, y), \eta_{I^\gamma}(x, y), \eta_{I^{1/\gamma}}(x, y)} (\|\nabla I(x, y)\|, \|\nabla I^\gamma(x, y)\|, \|\nabla I^{1/\gamma}(x, y)\|). \quad (2.16)$$

### Adaptation de l’ANEG aux images couleurs

Les corrections gamma peuvent être appliquées aux images couleur. En notant  $d$  le nombre de canaux d’une image couleur  $\mathbf{I}$  et  $I_i$  sa  $i^{me}$  composante ( $1 \leq i \leq d$ ),

nous appliquons la correction gamma sur les différents canaux  $I^i$  de l'image  $\mathbf{I}$ . De ces différents plans, nous obtenons trois groupes de trois images :  $\mathbf{I}$ ,  $\mathbf{I}^\gamma$  et  $\mathbf{I}^{1/\gamma}$ . D'après les équation 2.12 et 2.13 :

$$\|\nabla \mathbf{I}^\gamma(x, y)\| = \max_{1 \leq i \leq d} (\|\nabla I_i^\gamma(x, y)\|), \quad (2.17)$$

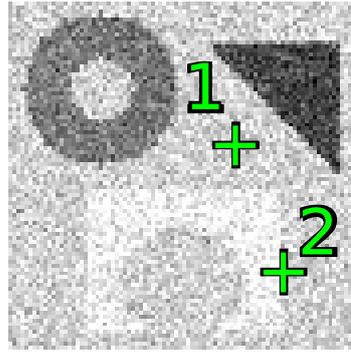
$$\|\nabla \mathbf{I}^{1/\gamma}(x, y)\| = \max_{1 \leq i \leq d} (\|\nabla I_i^{1/\gamma}(x, y)\|), \quad (2.18)$$

et

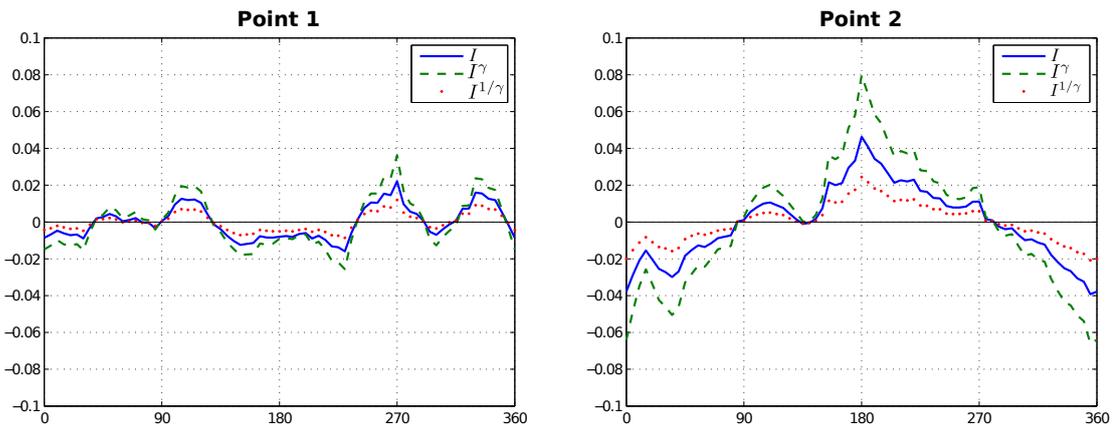
$$\eta_{\mathbf{I}^\gamma(x,y)} = \arg \max_{\eta_i^\gamma(x,y), 1 \leq i \leq d} (\|\nabla I_i^\gamma(x, y)\|), \quad (2.19)$$

$$\eta_{\mathbf{I}^{1/\gamma}(x,y)} = \arg \max_{\eta_i^{1/\gamma}(x,y), 1 \leq i \leq d} (\|\nabla I_i^{1/\gamma}(x, y)\|). \quad (2.20)$$

où  $\eta_i^\gamma(x, y)$  est la direction du gradient du pixel  $(x, y)$  associé au canal  $i$  de  $\mathbf{I}^\gamma$ .



(a) Sélection de points dans une image synthétique bruitée



(b) Pixel dans une région homogène

(c) Point de contour d'un objet sur-exposé

FIGURE 2.18 – Fonction  $\mathcal{Q}(x, y, \theta)$  utilisant les corrections gamma ( $\gamma=5$ ) de deux points sélectionnés en figure (a) d'une image synthétique bruitée. Les paramètres du filtre pour obtenir  $\mathcal{Q}(x, y, \theta)$  sont  $\mu=5$ ,  $\lambda=1$  and  $\Delta\theta=5^\circ$ .

Finalement, d'après les équations précédentes et l'équation 2.15, nous obtenons l'estimation d'un gradient  $\|\nabla\mathbf{I}(x, y)\|_{ANEG}$  d'une image vectorielle  $\mathbf{I}$  :

$$\begin{aligned} \|\nabla\mathbf{I}(x, y)\|_{ANEG} &= \max(\|\nabla\mathbf{I}(x, y)\|, \|\nabla\mathbf{I}^\gamma(x, y)\|, \|\nabla\mathbf{I}^{1/\gamma}(x, y)\|), \\ \eta_{ANEG}(x, y) &= \arg \max_{\eta_{\mathbf{I}}(x, y), \eta_{\mathbf{I}^\gamma}(x, y), \eta_{\mathbf{I}^{1/\gamma}}(x, y)} (\|\nabla\mathbf{I}(x, y)\|, \|\nabla\mathbf{I}^\gamma(x, y)\|, \|\nabla\mathbf{I}^{1/\gamma}(x, y)\|). \end{aligned} \quad (2.21)$$

Maintenant, les contours utilisant les courbes gamma sont extraits en calculant les maxima locaux de  $\|\nabla\mathbf{I}\|_{ANEG}$  dans la direction  $\eta_{ANEG}$ .

## 2.10 Résultats en utilisant les courbes gamma

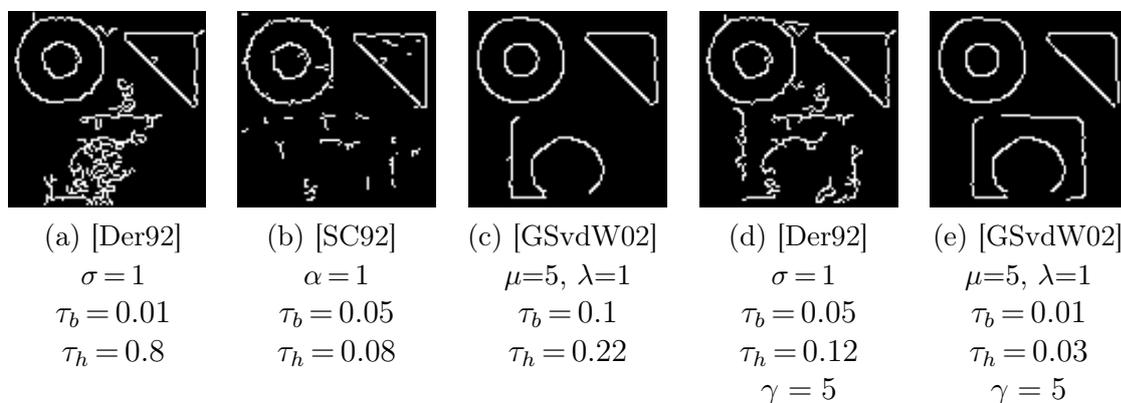


FIGURE 2.19 – Détection de contours de la figure 2.20(a) avec différentes méthodes : (a) et (d) gaussien isotrope, (b) Shen-Castan, (c) et (e) anisotrope gaussien. Nous avons appliqué les corrections gamma pour les images (d) et (e). Pour (c) et (e) :  $\Delta\theta = 5^\circ$ .

Dans nos résultats, nous utilisons  $\Delta\theta = 5^\circ$  pour tous les filtres anisotropes [GSvdW02] [KvdWHR07]. Dans le cas d'images vectorielles, nous utilisons le même paramètre  $\mu$  et  $\lambda$  pour tous les canaux de l'image et toutes les images corrigées par les paramètres  $\gamma$ . Le paramètre  $\gamma$  (et respectivement  $1/\gamma$ ) est le même pour tous les plans de l'image. D'après nos test,  $\gamma = 5$  est suffisant pour détecter les contours dans les parties de l'image sous et sur-exposées. Les seuils  $\tau_b$  et  $\tau_h$  sont choisis manuellement en fonction du maximum de contours désirés et du minimum de contours indésirables.

De façon générale, le choix des paramètres  $\mu$  et  $\lambda$  est effectué en fonction de la taille de l'image originale et la longueur des contours que nous souhaitons détecter. En effet, si nous avons besoin de détecter des petits détails ou des bords dans une petite image, le paramètre  $\mu$  ne doit pas être très élevé (par exemple  $\mu = 5$ ), alors que pour des longues structures,  $\mu$  doit être plus important (par exemple  $\mu = 10$ ). Pour sa part, le paramètre  $\lambda$  correspond à la largeur du filtre et ne doit pas être trop large puisqu'il est lié à la précision désirée au niveau du contours (en général  $\lambda = 1$ ).

Le premier résultat sur une image en niveaux de gris est présenté en figure 2.20, avec  $\mu = 5$  et  $\lambda = 1$ . Le résultat de l'ANEG est moins bruité que celui n'utilisant

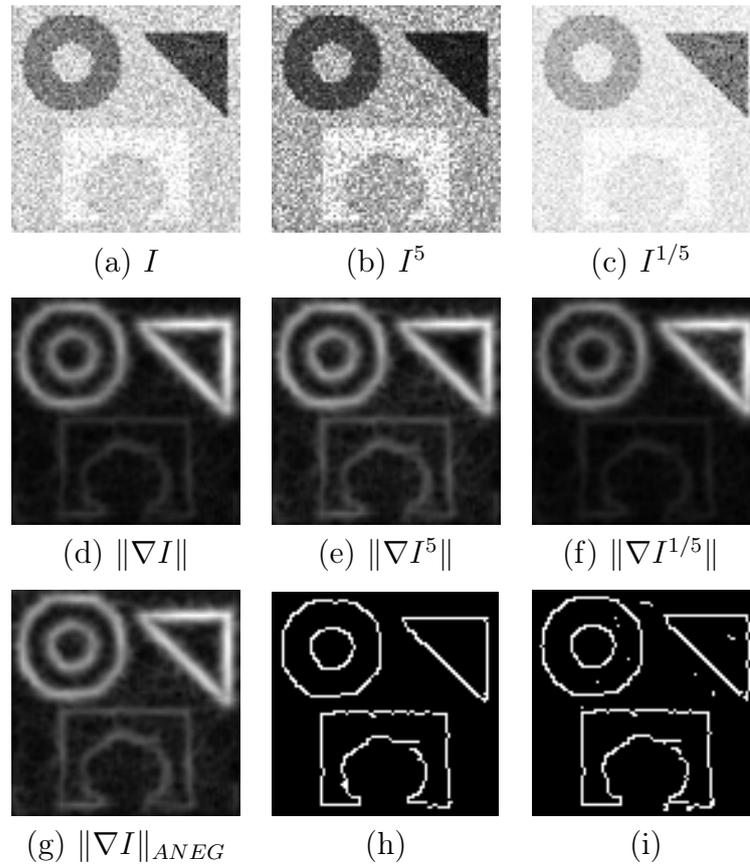


FIGURE 2.20 – Résultat du détecteur ANEG avec  $\mu = 5$ ,  $\lambda = 1$  (a) Image originale  $I$   $128 \times 128$ , (h) Contours extraits avec l'ANEG,  $\tau_b = 0.15$  et  $\tau_h = 0.2$ . (i) Résultat sans l'aide des courbes gamma (contours extraits de (a)),  $\tau_b = 0.08$  and  $\tau_h = 0.1$ .

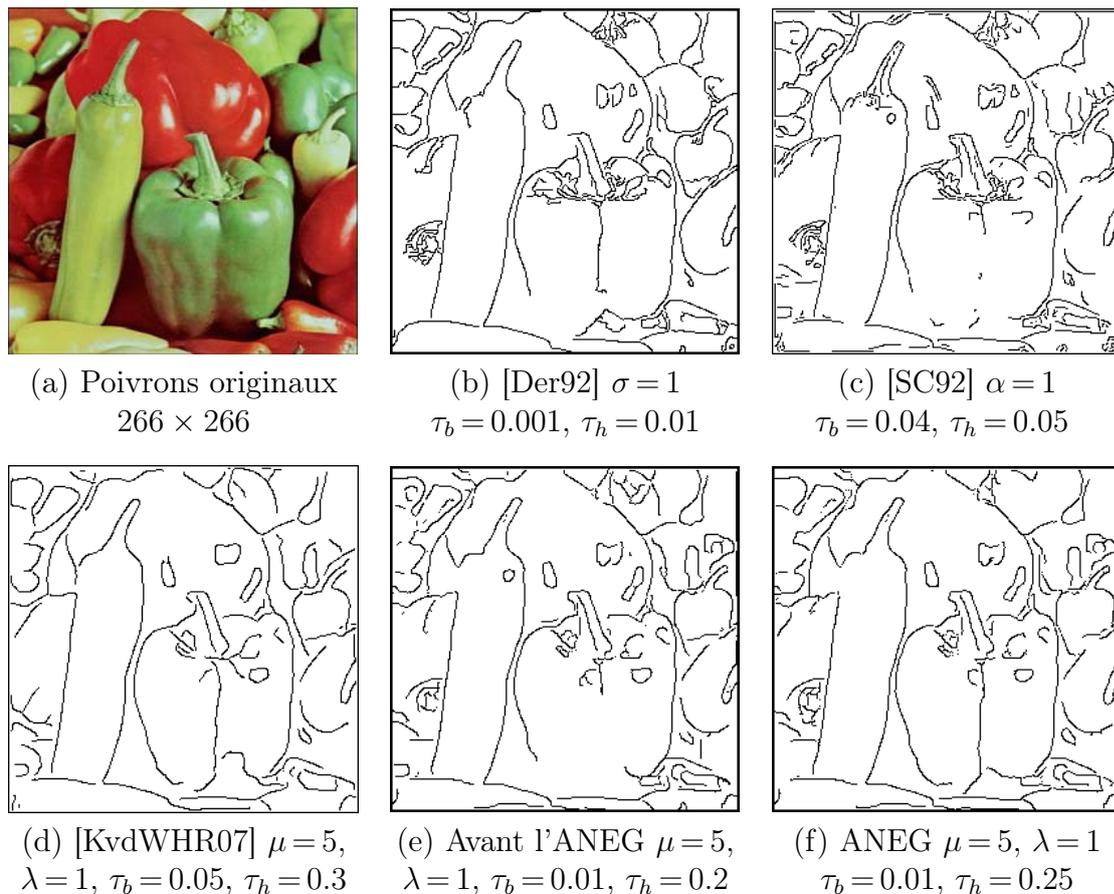


FIGURE 2.21 – Resultats sur une image couleur naturelle.

pas les courbes gamma. Donc l'ANEG semble plus robuste au bruit que le détecteur original. De plus, notre résultat est plus continu que celui du détecteur de Shen-Castan [SC92], le gaussien isotrope [Der92] et du filtre de Deriche [Der92]. Il est également plus précis que l'anisotrope gaussien [GSvdW02] (voire figure 2.19(a) (b) (c)). La figure 2.19(c) et (e) montre que les corrections gamma suivies par une maximisation n'est pas adaptée aux détecteurs classiques ou anisotropes.

Dans les résultats présentés en figure 2.21 (f), les contours détectés sont plus continus que ceux obtenus avec d'autres approches illustrés en figures 2.21(b), (c), (d) et (e). Cela se remarque spécialement au niveau du poivron vert au milieu de l'image qui est correctement détecté par l'ANEG. Le détecteur gaussien anisotrope (figures 2.21(d) ) détecte mal le bas du poivron sur la gauche mais surtout il est défaillant pour extraire les petits détails de l'image.

Dans la dernière image (figure 2.22(f) ), le but est d'extraire les contours de la voiture en obtenant des résultats le moins possible pollués par la texture des plantes. Les détecteurs de contours classiques [SC92] [Der92] et anisotropes [KvdWHR07] ne détectent pas la voiture ou très mal parce que la texture perturbe les résultats. Le résultat de notre détecteur de contour anisotrope est meilleur mais l'ANEG apporte plus de détails dans les zones de l'image sous et sur-exposées (spécialement les roues) sans pour autant créer des contours indésirables à l'intérieur de la texture.

Ces résultats nous satisfont grandement par rapport à notre approche initiale où

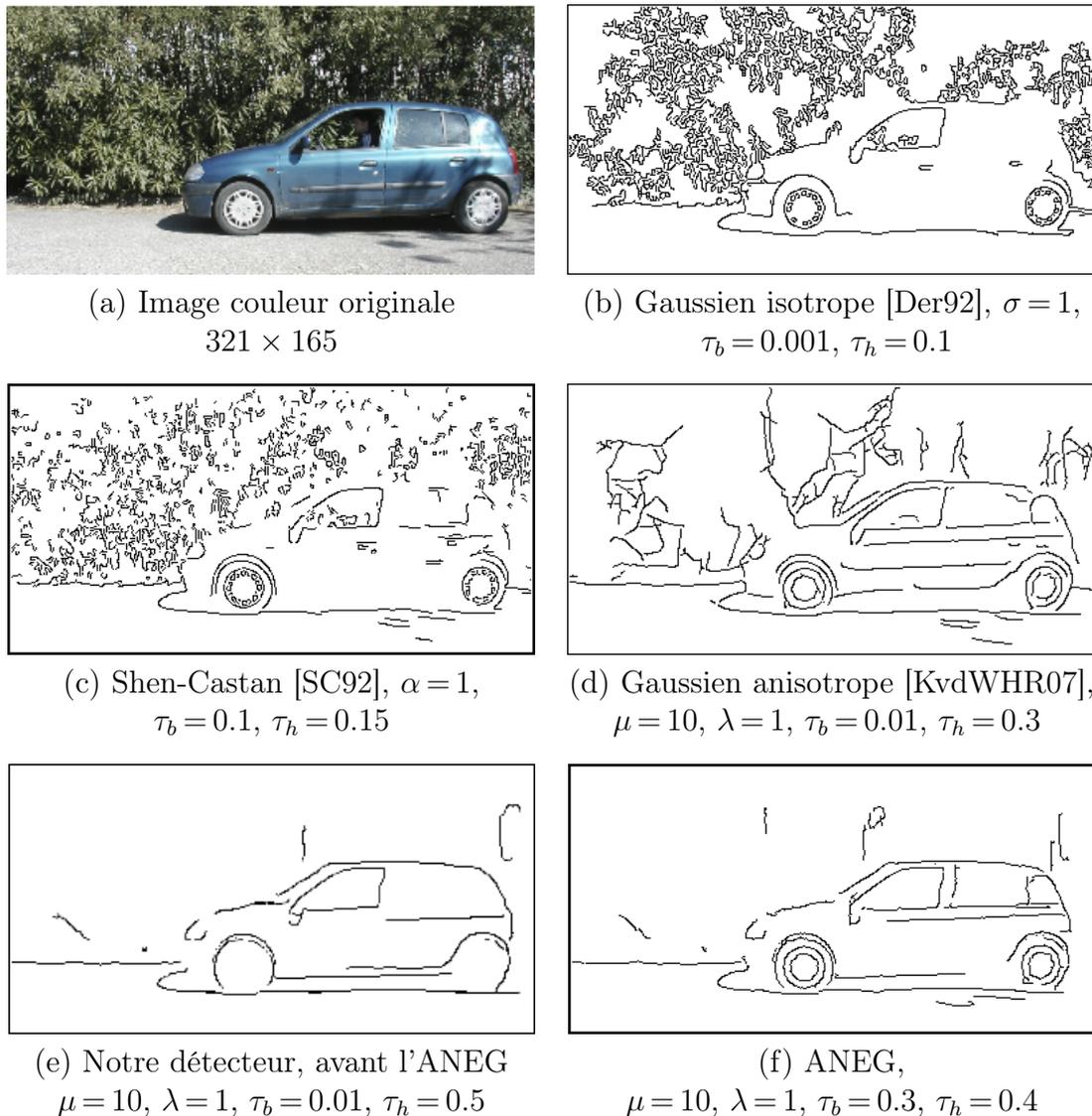


FIGURE 2.22 – Résultats sur une image couleur naturelle composée d'un objet et de textures.

détecter des contours en insérant les courbes gamma sur l'image originale semblait loin d'être évidente. Les courbes gamma font évoluer les valeurs des pixels, même ceux du bruit et de la texture, cependant, le filtre de dérivation est suffisamment allongé et fin pour lisser le bruit tout en restant précis aux niveaux des contours.

## 2.11 Evaluation du filtre de détection de bords

Afin de réaliser des résultats quantitatifs, nous avons conduit un nombre de tests avec des images synthétiques composées d'un carré noir (illustré en figure 2.23). Dans notre test, nous avons effectué la détection de contours et comparé le résultat avec l'image originale, pixel par pixel. Nous obtenons alors une erreur quantifiée par la différence entre les deux images. Nous analysons l'effet de bruit blanc uniforme additif sur l'image originale en utilisant la formule suivante :

$$I_m = (1 - L)I_0 + L \cdot I_N,$$

où  $I_0$  représente l'image originale,  $I_N$  une image de bruit aléatoire et  $I_m$  l'image bruitée résultante.

Comme attendu, le nombre d'erreurs croît monotoniquement avec le niveau du bruit  $L$ . Les courbes affichées en figures 2.24 représentent en (a) le nombre de pixels vrais négatifs (pixel de contours non détectés) et en (b) le nombre de faux positifs (contour détecté n'existant pas), qui constituent tous deux une erreur. Les figures 2.24 (c) et (d) montrent des détails de (a) et (b). Pour chaque approche, nous avons également calculé le nombre de coins détectés en figure 2.24 (e) parmi les 4 coins du carré présenté en figure 2.23 (a). Les paramètres des différentes approches sont :

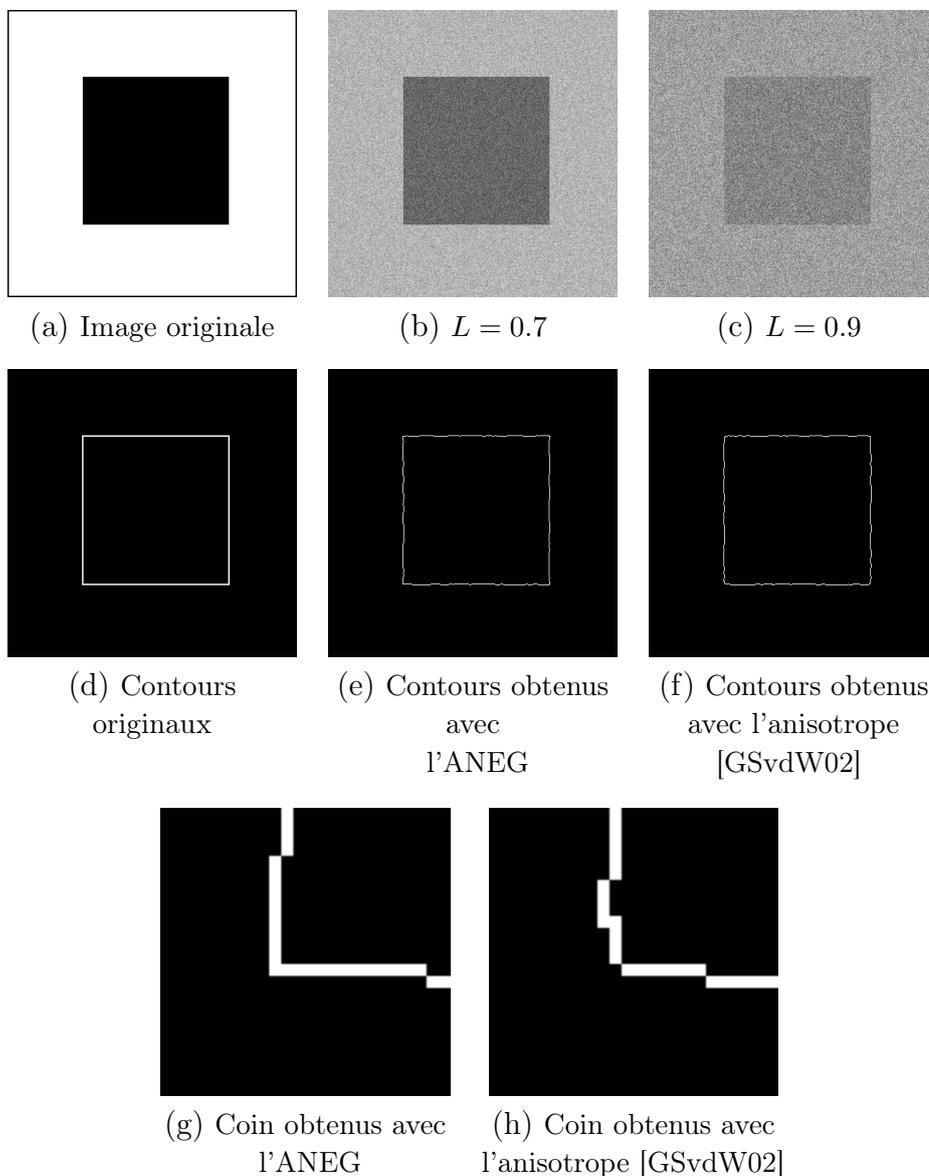


FIGURE 2.23 – Images  $400 \times 400$  avec différents niveaux de bruit  $L$  et comparaison avec le détecteur anisotrope gaussien [GSvdW02].

- $\alpha = 1$  pour le filtre de Shen-Castan [SC92]
- $\sigma = 1$  pour le filtre isotrope gaussien [Der92],
- $\mu = 1$ ,  $\lambda = 10$  et  $\Delta\theta = 5^\circ$  pour le filtre anisotrope gaussien [GSvdW02],
- $\mu = 1$ ,  $\lambda = 10$  et  $\Delta\theta = 5^\circ$  pour les demi-filtres anisotropes gaussiens  $\mathcal{Q}(x, y, \theta)$  (équation 2.6),
- $\mu = 1$ ,  $\lambda = 10$ ,  $\Delta\theta = 5^\circ$  et  $\gamma = 5$  pour l'ANEG.

Les résultats sont comparés après seuillage par hystérésis, où les contours obtenus sont les plus proches des contours désirés avec le moins de bruit possible.

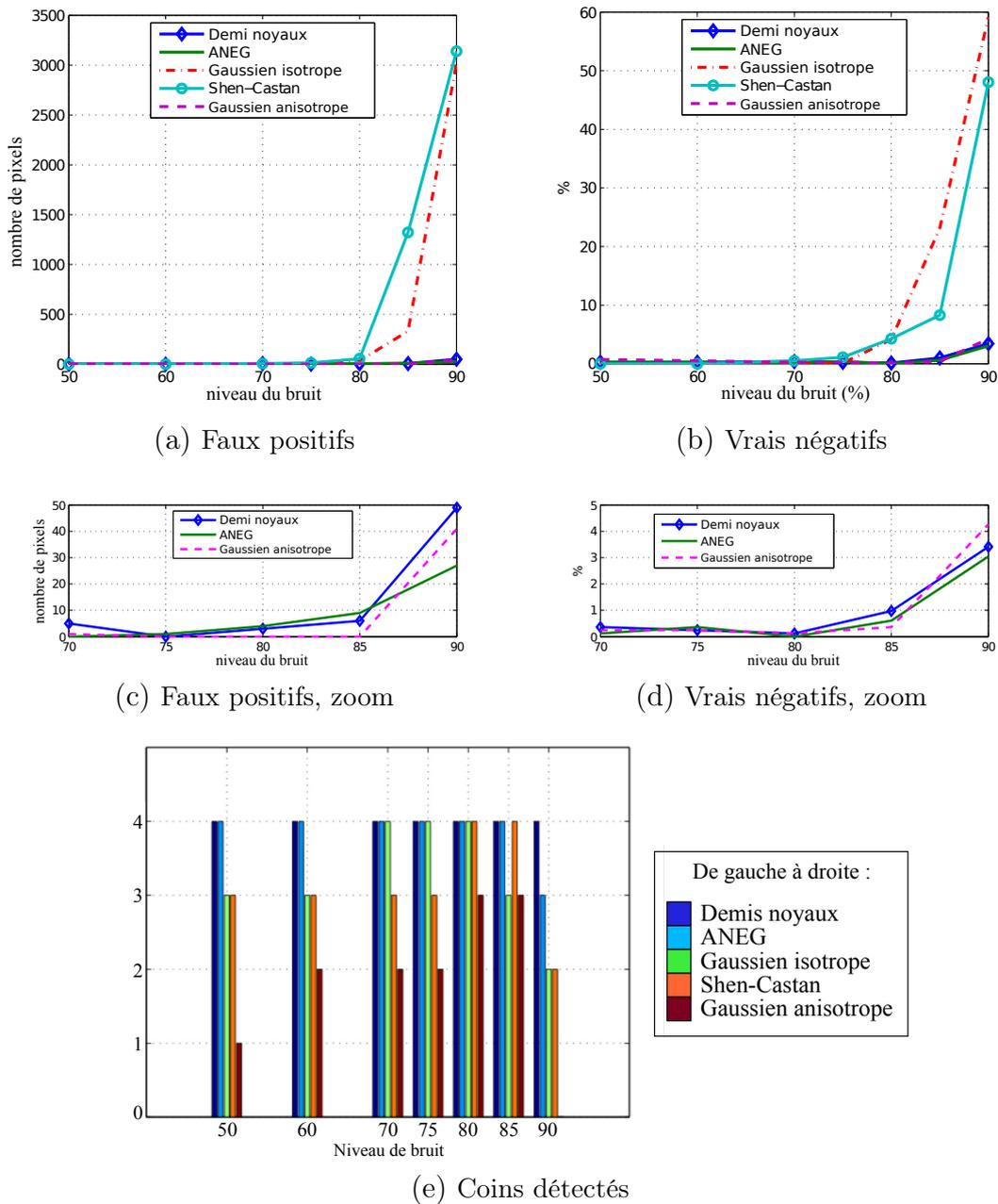


FIGURE 2.24 – Influence du niveau du bruit sur différentes méthodes de détection de contours.

Comparé aux méthodes isotropes de Shen-Castan [SC92] et du gaussien [Der92], les trois méthodes de détection anisotropes de contours sont performantes et montrent une grande robustesse au bruit. Cependant, comme illustré en figure 2.24 (e), les approches des demi-noyaux gaussiens et l'ANEG sont meilleures que le gaussien anisotrope pour détecter les coins [Per92] [GSvdW02]. Les figures 2.23 (g) et (h) illustrent la détection d'un coin avec l'ANEG et avec le gaussien anisotrope. La détection de contours en utilisant les noyaux anisotropes gaussiens arrondit les coins des contours, la plus forte réponse du filtre se situe à l'intérieur même du coin, cela revient au même problème que le détecteur isotrope de contour gaussien.

Les détections de contours avec les demi-noyaux gaussiens et l'ANEG produisent des résultats satisfaisants, même en présence de bruit fort. Cette évaluation a été effectuée sur des images synthétiques, les résultats présentés dans les parties précédentes montrent également la supériorité de nos approches par rapport aux filtres composés d'un seul noyau gaussien [Der92] [Per92] [GSvdW02].

## 2.12 Conclusion

Nous avons proposé dans cette partie un détecteur de contours anisotrope, précis et robuste, pour les images scalaires ou vectorielles. Il est formé par deux demi-noyaux orientés de lissage. Ce filtre est capable d'extraire les contours de structures linéaires mais aussi de petits objets puisque ses deux demi-noyaux lui permettent de rechercher un chemin entrant et un chemin sortant dans deux différentes directions pour chaque pixel de l'image considérée. A l'aide d'un fort lissage dans les directions des contours, le détecteur est peu sensible au bruit, il est capable d'extraire les contours même pour des images très bruitées. De plus, il reste localisé au niveau des contours car il lisse peu dans la direction perpendiculaire aux contours.

Ce détecteur a été amélioré en utilisant les courbes gamma, les gradients sont estimés sur trois images différentes : l'image originale et deux images corrigées. Le gradient retenu est celui dont la valeur est maximale. Les corrections gamma n'introduisent pas de contours indésirables mais permettent d'extraire des bords dans les zones sous ou sur-exposées de l'image même en présence de bruit.

Le filtre présenté est implémenté récursivement, ce qui lui permet d'avoir une performance temporelle intéressante. Cependant, en optimisant le code et en parallélisant sur un GPU (surtout concernant la rotation des images avant d'appliquer notre filtre de dérivation [UTY95] [CVDV08]), nous pourrions obtenir un résultat en temps réel, ce qui serait intéressant pour des applications robotiques ou industrielles par exemple.

Une idée pour extraire des contours dans les images floues serait l'adaptation de ce filtre en multi-échelle [Per92]. En effet, l'échelle optimale du détecteur de bords, dépend de l'échelle équivalente du contour et la meilleure localisation n'est pas forcément dans l'échelle la plus fine [LTD05]. La meilleure représentation en terme de point de contour doit être donc repérée dans chaque échelle.

Finalement, ce détecteur est performant au niveau des coins, nous pensons étendre dans les temps prochains notre méthode à la détection de coins et de jonctions [MDA07] [MFA11] en utilisant les deux directions et les valeurs de notre filtre. Nous pouvons également imaginer des potentielles applications telles que la détection de points d'intérêts et la mise en correspondance. Cette dernière application a déjà

fait l'objet d'un travail dans notre laboratoire [PMD12] où suite à une détection de points de Harris [HS88] [SMB00] [MGD98] dans deux images différentes, les points d'intérêts ont été appariés suivant les valeurs des fonctions  $\mathcal{Q}(x, y, \theta)$  (équation 2.6). Lorsque deux points d'intérêts dans deux images différentes obtiennent deux signaux similaires, ils sont mis en correspondance.

## 3 Différence de demi-gaussiennes anisotropes pour la détection de contours toits

### 3.1 Introduction

Les crêtes et les vallées sont formées par les points où l'intensité lumineuse des pixels en niveaux de gris est un extremum local dans une direction donnée. Comme détaillé dans la section 6.5 du chapitre 1, cette direction est la normale à la courbe en cet extremum [DC76]. Les lignes de crêtes correspondent à des structures importantes de l'image. Les crêtes et les vallées sont attachées mais non limitées aux routes dans les images aériennes [Men03] [LML<sup>+</sup>00] ou aux vaisseaux sanguins dans les images médicales [AMM97] [KQ04].

Les détecteurs de contours classiques échouent pour détecter les lignes de crêtes (voir section 6.5 du chapitre 1). Cependant, les contours peuvent être calculés pour détecter des lignes droites en utilisant une transformée de Hough [Hou62]. Ainsi, Zhou et al. [ZBSA06] estiment les bords à l'aide d'un détecteur anisotrope de contours [Per92] [GSvdW02] créant deux contours droits de part et d'autre de la ligne de crête. La transformée de Gabor permet alors de détecter des lignes. Cependant, cette méthode est seulement adaptée pour détecter des lignes droites [EMED08].

Il existe plusieurs catégories d'algorithmes de segmentation pour détecter les lignes de crêtes, le lecteur curieux pourra se référer à [KQ04] et [Lac04]. Parmi ces méthodes, la reconnaissance de formes, les techniques de filtrage (par exemple la géométrie différentielle comme détaillée en section 6.5 du chapitre 1 [AMM97] [MD97b]), les approches de contours actifs (snakes) [Lin98] [LML<sup>+</sup>00], ou les approches de suivi.

Les méthodes de filtrages sont bien adaptées pour l'extraction de lignes de crêtes puisqu'elles sont capables de lisser le bruit et amplifier l'information des lignes de crêtes en calculant la courbure de la surface [MAM95] [AMM97] [Ste98] [Lin98] [Tra06] [Zio91]. Toutefois, les résultats obtenus par ces approches peuvent présenter un nombre important de fausses détections puisque le filtrage passe haut utilisé pour la dérivée seconde est sensible au bruit.

Dans cette partie, nous présentons un filtre orienté capable de détecter les crêtes et les vallées. Notre détecteur de lignes de crêtes implémente un filtrage linéaire directionnel anisotrope par le moyen de la différence de deux demi-filtres de lissage orientés (plus précisément deux différences de deux demi-gaussiennes orientées). Puis, en utilisant une maximisation ou une minimisation directionnelle de la réponse du filtre, les crêtes ou respectivement vallées sont extraites. Ces directions correspondent à l'orientation d'une ligne de crête ou d'une jonction de différentes lignes de crêtes. Contrairement à plusieurs approches d'extraction de lignes de crêtes, cet algorithme est très performant, même au niveau de crêtes ou de vallées très coudées puisqu'il est composé de deux demi-noyaux de lissage orientés dans deux directions différentes. Finalement, comme le détecteur de contour présenté dans la partie 2.2

de ce présent chapitre, la détection de lignes de crêtes ainsi détaillée est peu sensible au bruit puisque le filtre est composé d'un fort lissage dans les directions de la ligne de crête.

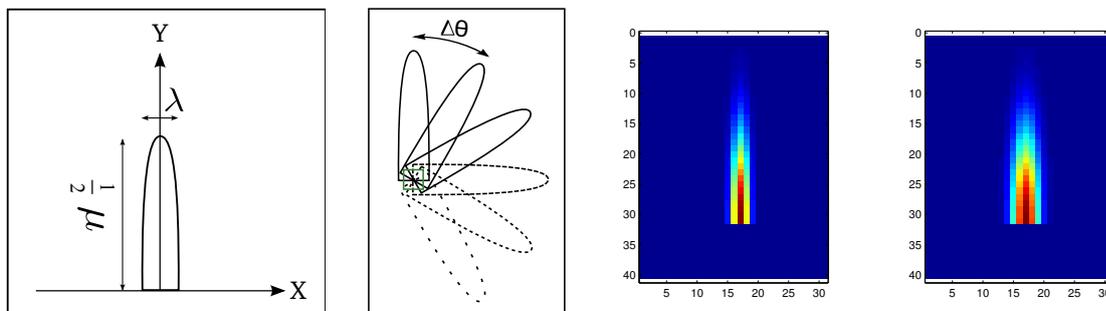
### 3.2 Un demi-filtre de lissage orienté

Dans notre méthode de détection de lignes de crêtes, nous utilisons un filtre de lissage orienté pour construire un signal  $s$  qui est une fonction de l'angle de rotation  $\theta$  et de la réponse du filtre en chaque pixel. Comme montré dans la partie 2.2 de ce chapitre, lisser avec des demi-filtres orientés signifie que l'image est lissée avec une banque de demi-noyaux gaussiens orientés d'angle  $\theta$  :

$$G_{(\mu,\lambda)}(x, y, \theta) = C.H \left( R_\theta \begin{pmatrix} x \\ y \end{pmatrix} \right) e^{-\begin{pmatrix} x & y \end{pmatrix} R_\theta^{-1} \begin{pmatrix} \frac{1}{2\mu^2} & 0 \\ 0 & \frac{1}{2\lambda^2} \end{pmatrix} R_\theta \begin{pmatrix} x \\ y \end{pmatrix}} \quad (2.22)$$

où  $C$  représente un coefficient de normalisation,  $R_\theta$  une matrice de rotation d'angle  $\theta$ ,  $(x, y)$  représentent les coordonnées du pixel considéré et  $(\mu, \lambda)$  les écarts types du filtre gaussien. Nous n'avons seulement besoin de la partie causale du filtre (illustré en figure 2.25(a), (c) et (d)), donc nous "coupons" le noyau de lissage au milieu, cette opération correspond à la fonction marche ou échelon  $H$ . Par convolution avec ces noyaux orientés, (voir figure 2.25(b)), nous obtenons une collection d'images lissées orientées  $I_\theta = I * G_{(\mu,\lambda)}(\theta)$ .

Comme pour le filtre de dérivation de la partie 2.2 de ce chapitre, nous procédons dans une première partie à la rotation de l'image dans différentes directions avec une incrémentation  $\Delta\theta$  de 0 à 360 degrés. Puis, le filtre de lissage qui est fixe est appliqué et une rotation inverse de l'image est effectuée pour construire une banque de  $360/\Delta\theta$  images lissées  $I_\theta$  (quelques exemples d'images  $I_\theta$  sont disponibles en figure 2.26). Dans une orientation donnée, le bruit est totalement lissé tandis qu'une courbe dirigée dans la direction du filtre de lissage ressort nettement.



(a) Filtre de lissage (b) Filtres orientés (c)  $\mu = 10 \lambda = 1$  (d)  $\mu = 10 \lambda = 1.5$

FIGURE 2.25 – Un filtre de lissage de paramètres  $\mu$  et  $\lambda$ . Exemple de filtre discret en (c) et (d).

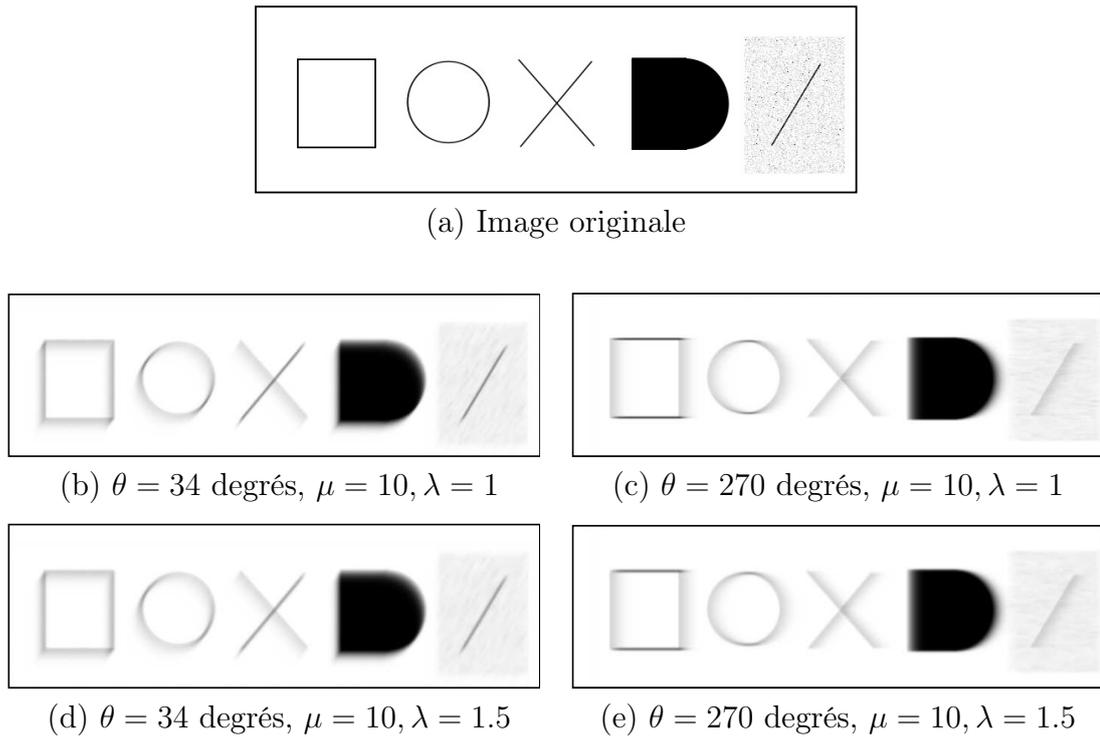


FIGURE 2.26 – Image lissée avec différents paramètres à différentes orientations.

### 3.3 Différence de deux demi-gaussiennes pour l'extraction de lignes de crêtes

#### Différence de deux demi-filtres de lissage orientés

Comme présenté en figure 2.27, nous souhaitons estimer en chaque pixel une dérivée seconde lissée de l'image le long d'une courbe traversant ce pixel. En une dimension, la dérivée seconde d'un signal peut être estimée en utilisant l'opérateur de différence de gaussiennes appelé DoG. Ainsi illustré en figure 2.28, par convolution avec une DoG, les impulsions d'un signal 1D apparaîtront toujours sous forme de pics tandis que les signaux de type porte seront complètement déformés et de moyenne nulle. Concernant notre problème en deux dimensions, nous devons juste

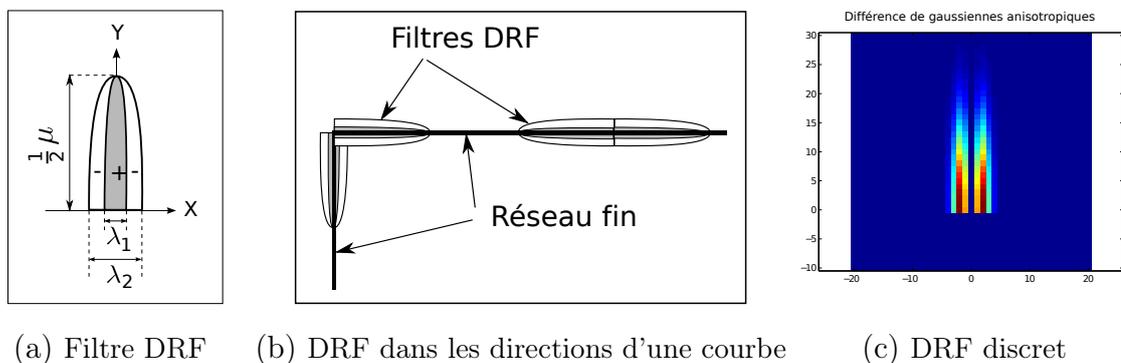


FIGURE 2.27 – Descriptions du filtre DRF. Pour (c)  $\mu = 10, \lambda = 1$  et  $\lambda = 1.5$ .

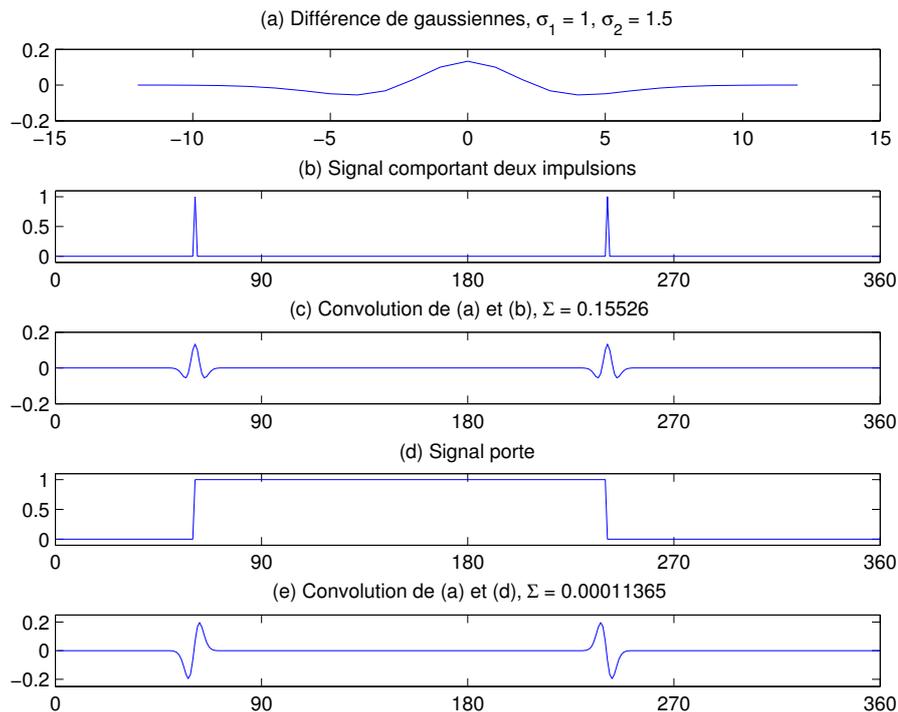


FIGURE 2.28 – Différence de gaussienne en une dimension convoluée avec des signaux 1D. La valeur  $\Sigma$  correspond à la somme des deux valeurs maximales et des deux valeurs minimales du signal convolué et  $(\sigma_1, \sigma_2)$  aux deux écarts types des gaussiennes.

appliquer deux filtres avec deux paramètres différents  $\lambda$  et le même  $\mu$  afin d'obtenir les dérivées directionnelles (deux filtres de lissage discrétisés sont présentés en figure 2.25 (c) et (d)). Puis, nous calculons la différence de ces deux filtres pour obtenir l'information de dérivée seconde de lissage désirée dans les directions de la courbe comme schématisé en figure 2.27(b). Nous appelons ce filtre DRF (*Difference of Rotated Half Smoothing Filters*).

### Classification des pixels

En appliquant par convolution le filtre DRF à chaque pixel de l'image (par la technique de rotation d'image expliquée précédemment), nous obtenons pour chaque

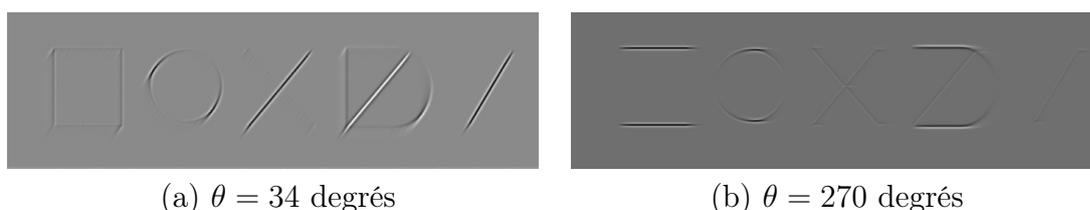
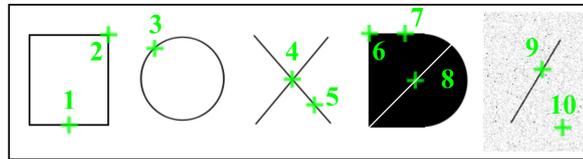


FIGURE 2.29 – Résultat du filtre DRF dans différentes orientations  $\theta$  avec les paramètres suivants :  $\mu = 10$ ,  $\lambda_1 = 1$  et  $\lambda_2 = 1.5$  (images normalisées).

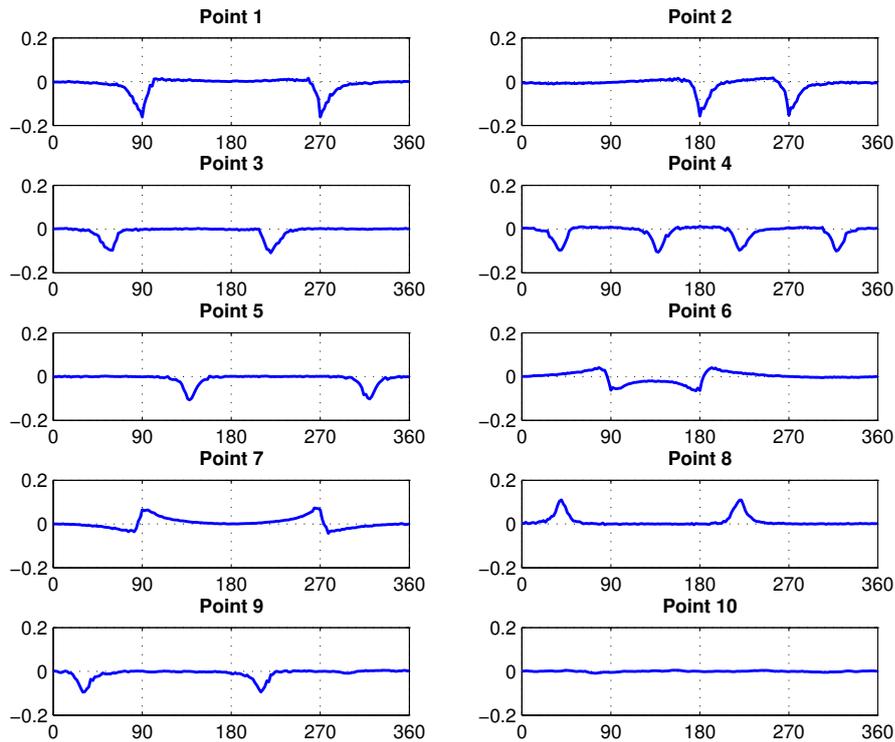
pixel un signal qui correspond à  $360/\Delta\theta$  analyses dans toutes les directions. Deux résultats du filtre DRF sont disponibles en figure 2.29, les crêtes ressortent positivement lorsque le filtre est orienté dans la direction de la courbe (ligne blanche) et négativement dans le cas des vallées (ligne noire). Notre idée est alors de caractériser les pixels qui appartiennent à une crête ou une vallée.

En posant  $D(x, y, \theta)$  le signal obtenu en un pixel  $P$  situé en  $(x, y)$ ,  $D(x, y, \theta)$  est une fonction de la direction  $\theta$  tel que :

$$D(x, y, \theta) = G_{(\mu, \lambda_1)}(x, y, \theta) - G_{(\mu, \lambda_2)}(x, y, \theta), \quad (2.23)$$



(a) Sélection de points



(b) Signaux  $D(x, y, \theta)$  associés aux points de (a)

FIGURE 2.30 – Exemples de fonctions  $D(x, y, \theta)$  sur une sélection de points en utilisant les paramètres  $\mu = 10, \lambda_1 = 1, \lambda_2 = 1.5$ . Dans (b), l'axe des  $x$  correspond à la valeur de  $\theta$  (en degrés) et l'axe des  $y$  à  $D(x, y, \theta)$ .

où  $\mu$ ,  $\lambda_1$  et  $\lambda_2$  correspondent aux écarts-types des différents filtres gaussiens. La figure 2.30 illustre quelques exemples de  $D(x, y, \theta)$  sur une sélection de pixels.

Nous définissons un opérateur de crête et de vallée  $\Sigma(x, y)$  par l'équation suivante :

$$\Sigma(x, y) = D(x, y, \theta_{M_1}) + D(x, y, \theta_{M_2}) + D(x, y, \theta_{m_1}) + D(x, y, \theta_{m_2}) \quad (2.24)$$

où  $(\theta_{M_1}, \theta_{M_2})$  sont les directions des maxima locaux de la fonction  $D$  et  $(\theta_{m_1}, \theta_{m_2})$  les directions de minima locaux (un exemple d'image  $\Sigma$  est disponible en figure 2.32(a)). Les conditions de détection de ligne de crête sont les suivantes :

- si  $\Sigma(x, y) > \Sigma_{th}$ , le pixels  $P$  appartient à une crête,
- si  $\Sigma(x, y) < -\Sigma_{th}$ , le pixels  $P$  appartient à une vallée,

avec  $\Sigma_{th} > 0$ .

Sur une vallée typique, (par exemple le point 1 dans la figure 2.30), le signal du pixel au minimum d'une vallée contient au moins deux pics négatifs. Dans le cas d'une crête, (par exemple le point 7 de la figure 2.30), le signal du pixel au maximum d'une crête contient au moins deux pics positifs. Ces pics étroits correspondent à deux directions d'une courbe (un chemin entrant par le pixel et un chemin sortant). Dans le cas d'une jonction, le nombre de pics correspondent au nombre de lignes de crêtes au niveau du pixel de la jonction (point 4 dans la figure 2.30). Nous obtenons également une information des deux directions pour les lignes courbées (illustré au point 2 de la figure 2.30). Toutefois, au niveau d'un contours, la valeur absolue de  $\Sigma$  est proche de 0 puisque la valeur absolue de  $D$  au niveau de  $\theta_{M_1}$ ,  $\theta_{M_2}$ ,  $\theta_{m_1}$  et  $\theta_{m_2}$  sont proches mais de signes différents (voir point 6 et 7 de la figure 2.30). Finalement, le filtre lisse assez fortement pour que  $D$  soit proche de 0 en présence de bruit en dehors d'une ligne de crête ou d'un contour (illustré au point 10 de la figure 2.30), c'est pourquoi notre détecteur DRF n'est pas sensible au bruit.

Nous pouvons également utiliser le paramètre  $\Sigma_{th}$  pour le seuillage par hystérésis.

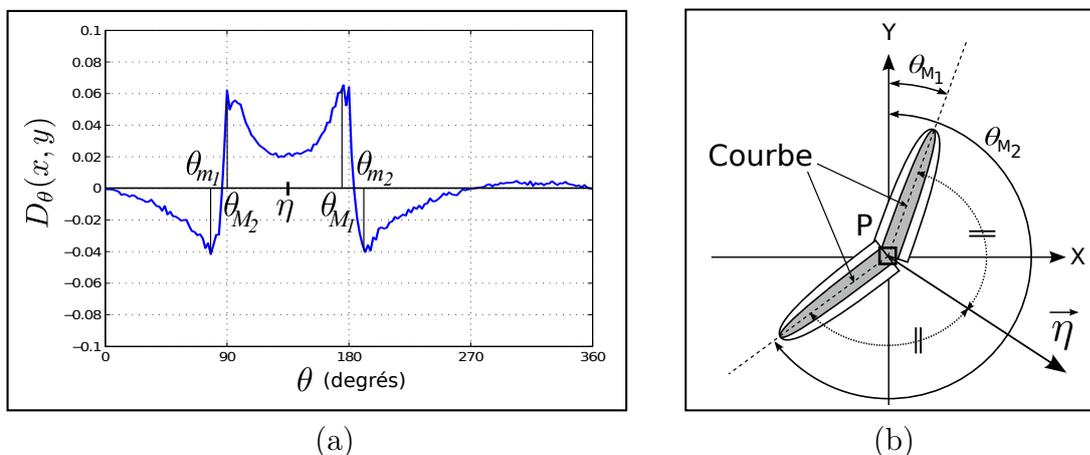


FIGURE 2.31 – Estimation de la direction  $\eta$  (cas  $\Sigma(x, y) > \Sigma_{th}$ ). (a) Calcul de  $\eta$  à partir de  $\theta_{M_1}$  et  $\theta_{M_2}$ . (b)  $\eta$  correspond à la direction perpendiculaire à la ligne de crête au niveau du pixel  $P$ .

### Extraction de crêtes et de vallées

Une fois  $\Sigma(x, y)$  calculé, nous estimons simplement  $\eta(x, y)$  (voir figure 2.31(a) et (b)) par :

$$\begin{cases} \eta(x, y) = (\theta_{M_1} + \theta_{M_2})/2, & \text{si } \Sigma(x, y) > \Sigma_{th} \\ \eta(x, y) = (\theta_{m_1} + \theta_{m_2})/2, & \text{si } \Sigma(x, y) < -\Sigma_{th}. \end{cases}$$

A partir de  $\Sigma(x, y)$  et  $\eta(x, y)$  (schéma en figure 2.31(b)), les lignes de crêtes peuvent être facilement extraites en calculant les maxima locaux de  $\Sigma(x, y)$  dans la direction  $\eta(x, y)$  (dans le cas d'une crête et les minima pour la détection d'une vallée, des exemples des différentes étapes sont visibles en figure 2.32(c) et (d)).

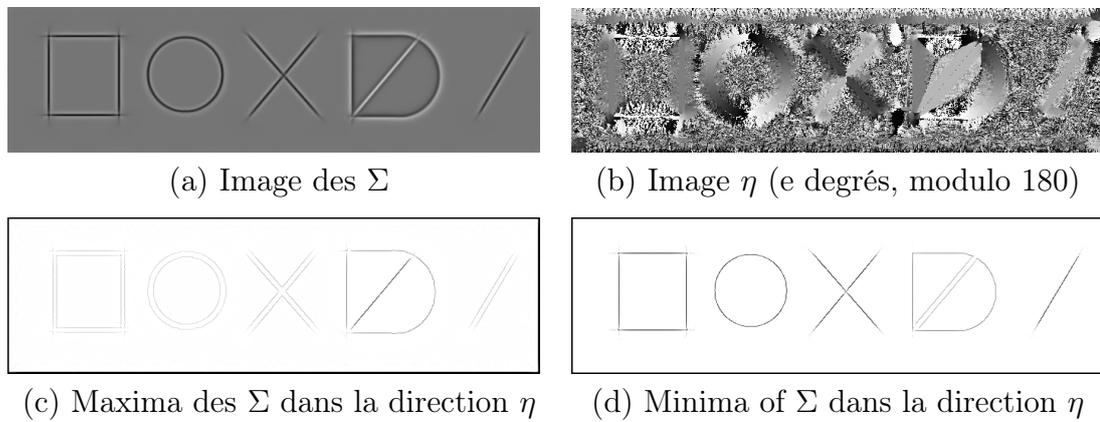


FIGURE 2.32 – Exemple des différentes étapes pour l'extraction de lignes de crêtes dans l'image présentée en figure 2.26 (a). Les paramètres utilisés sont  $\Delta\theta = 2$  (degrés),  $\mu = 10$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$ . Toutes ces images sont normalisées.

Dans la partie suivante sont exposés quelques résultats, en comparant avec d'autres approches, puis nous avons évalué la méthode et montré que le DRF peut détecter des courbes perceptuelles.

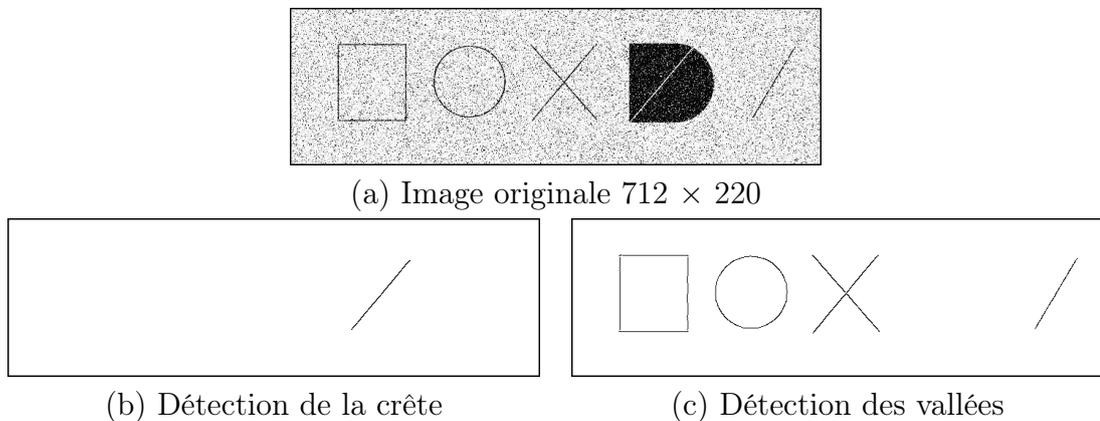
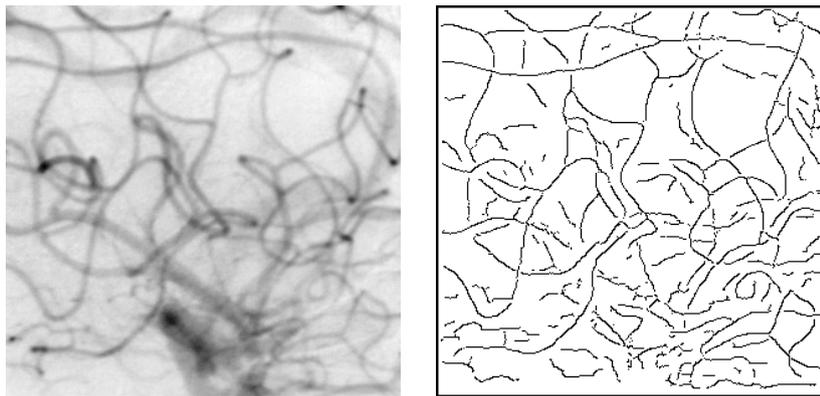
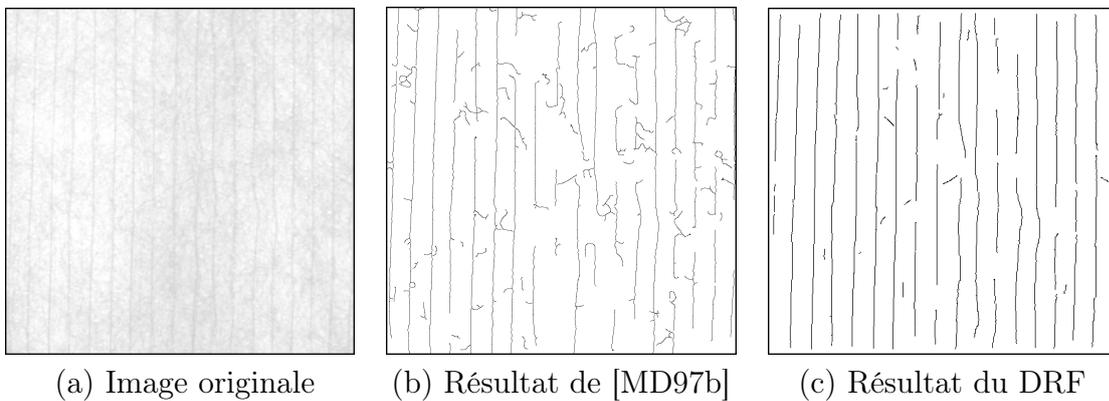


FIGURE 2.33 – Détection sur une image synthétique bruitée avec les paramètres suivants  $\Delta\theta = 2^\circ$ ,  $\mu = 10$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$ ,  $\tau_b = 0.03$  et  $\tau_h = 0.08$ .



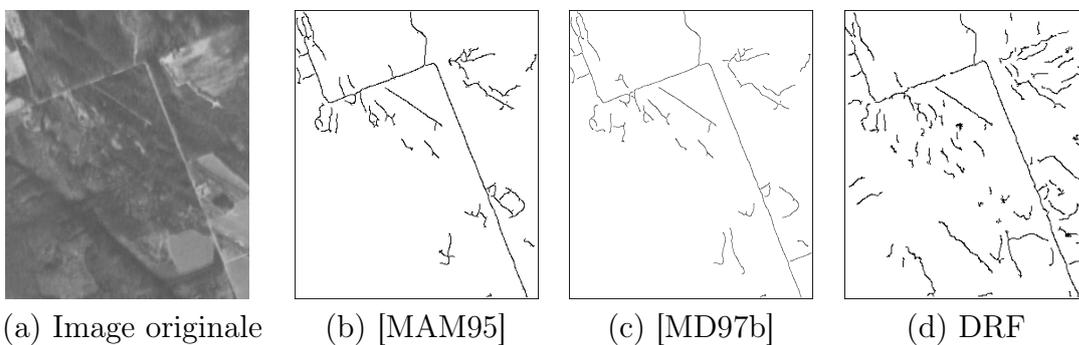
(a) Image originale 312×312 (b) Détection des vallées

FIGURE 2.34 – Extraction des vaisseaux sanguin dans une image de cerveau.  $\Delta\theta = 5^\circ$ ,  $\mu = 5$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$ ,  $\tau_b = 0.0001$  et  $\tau_h = 0.005$ .



(a) Image originale (b) Résultat de [MD97b] (c) Résultat du DRF

FIGURE 2.35 – Détection des filigranes dans un papier correspondant à des vallées dans une image. (b)  $\sigma = 1.5$ ,  $\tau_b = 0.5$  et  $\tau_h = 0.8$ . (c)  $\Delta\theta = 5^\circ$ ,  $\mu = 10$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$ ,  $\tau_b = 0.001$  et  $\tau_h = 0.008$ .



(a) Image originale (b) [MAM95] (c) [MD97b] (d) DRF

FIGURE 2.36 – Détection des crêtes dans une image satellite  $277 \times 331$ . Pour (a) :  $\sigma = 1.5$ ,  $\tau_b = 0.5$  et  $\tau_h = 0.7$ . Pour (c) :  $\sigma = 1.5$ ,  $\tau_b = 0.5$  et  $\tau_h = 0.7$ . Pour (d) :  $\Delta\theta = 2$ ,  $\mu = 3$ ,  $\lambda_1 = 1.33$ ,  $\lambda_2 = 2$ ,  $\tau_b = 0.002$  et  $\tau_h = 0.01$ .

### 3.4 Résultats

Nous présentons des résultats obtenus à la fois sur une image synthétique et des images réelles en utilisant le DRF.

#### Résultats sur une image synthétique

La figure 2.33 montre le résultat de notre détecteur de lignes de crêtes sur une image synthétique bruitée. Les crêtes sont aussi bien correctement extraites que les vallées, alors que le DRF n'est ni perturbé par les contours de l'objet noir ni le bruit. Après les résultats, nous présentons une évaluation de la robustesse de notre

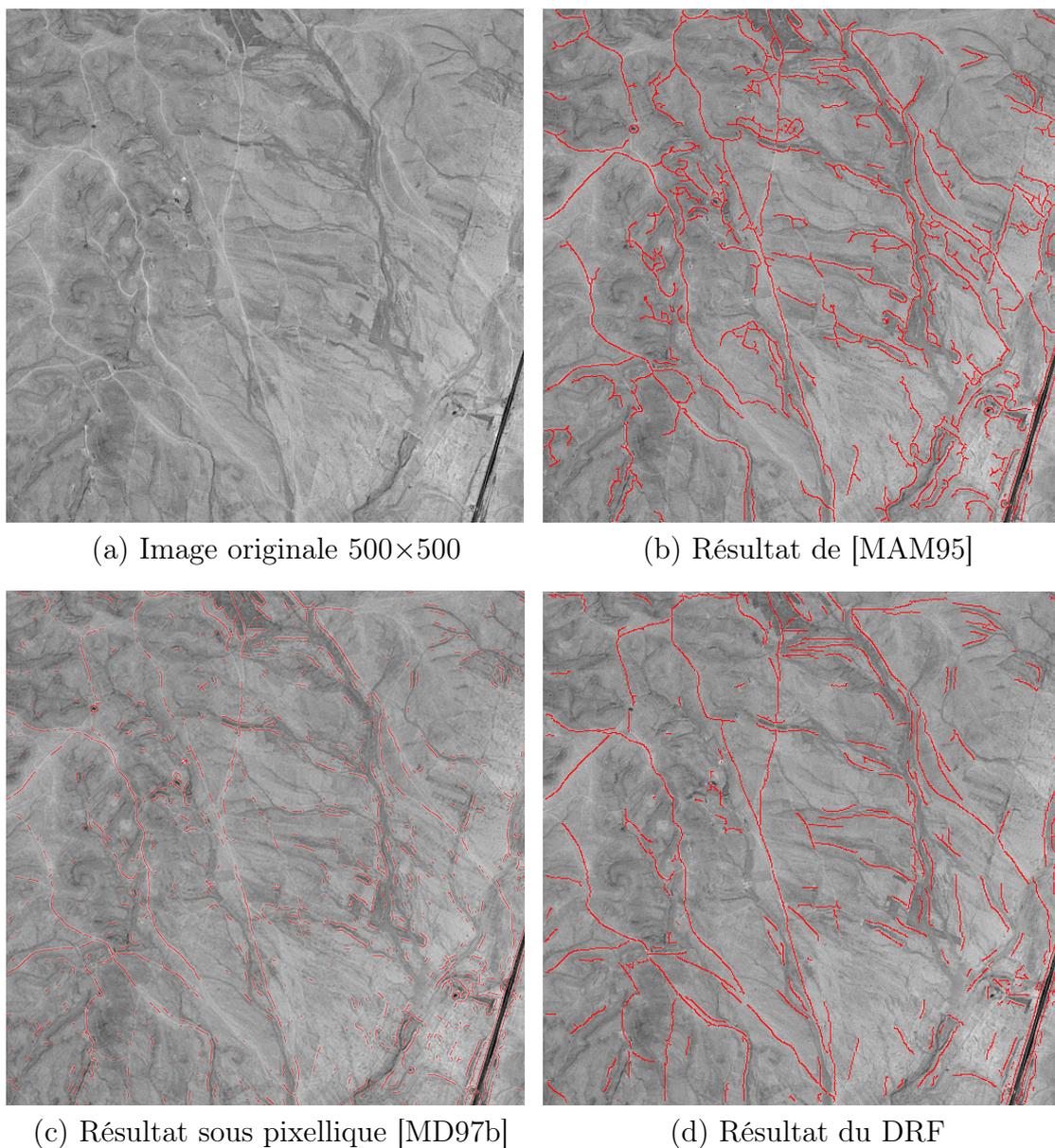


FIGURE 2.37 – Détection de crêtes dans une image satellite. Pour (b),  $\sigma = 2$ ,  $\tau_b = 0.5$  et  $\tau_h = 0.7$ . Pour (c),  $\sigma = 1.5$ ,  $\tau_b = 0.55$  et  $\tau_h = 0.65$ . Pour (d) :  $\Delta\theta = 5^\circ$ ,  $\mu = 10$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$ ,  $\tau_b = 0.02$  et  $\tau_h = 0.06$ .

détecteur en fonction du niveau de bruit dans l'image.

### Resultats sur des images réelles

Nous avons testé notre détecteur sur plusieurs images réelles différentes et comparé la DRF avec les approches décrite dans [MAM95] et [MD97b].

Dans la première image médicale, figure 2.34, le but est d'extraire les vaisseaux sanguins. Cette image n'est pas polluée par le bruit, cependant les vallées sont coudées mais le DRF les extrait facilement de l'image.

Dans la figure 2.35, le but est d'extraire les filigranes verticaux. Comme l'image est très bruitée, la tâche d'extraire les vallées causées par les filigranes est très difficile avec les méthodes classiques [MAM95]. Même les approches sous-pixelliques ne parviennent pas à extraire correctement les lignes de crêtes dans cette image, comme illustré en figure 2.35 (b). Cependant, le DRF est performant, les contours extraits sont moins bruités avec notre méthode que l'approche sous-pixellique [MD97b].

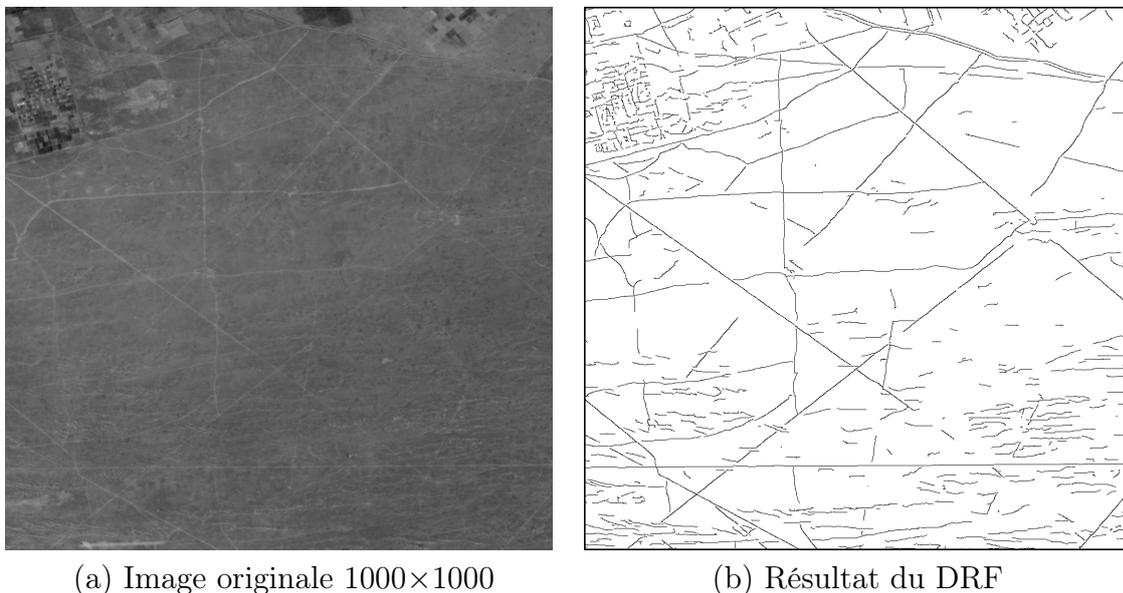


FIGURE 2.38 – Détection de crêtes dans une image satellite du désert.  $\Delta\theta = 5^\circ$ ,  $\mu = 10$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$ ,  $\tau_b = 0.01$  et  $\tau_h = 0.0025$ .

Les routes apparaissent souvent comme des crêtes ou des vallées dans les images satellites. Dans la figure 2.36, les routes extraites sont clairement visibles contrairement à l'image en figure 2.37 où le DRF détecte les crêtes même lorsqu'elles sont très coudées. De plus, notre détecteur est performant au niveau des jonctions. Nous avons comparé notre détecteur avec les approches présentées dans [MAM95] et [MD97b], les résultats montrent clairement la supériorité de notre approche. Dans la figure 2.38, les crêtes sont floutées puisqu'il s'agit de routes dans le désert, toutefois le DRF est capable d'extraire la plupart des routes.

Les deux dernières images montrent l'efficacité de notre détecteur face au bruit. Le but dans la figure 2.39 est de détecter le filament. Nous avons comparé notre méthode avec 4 autres approches existantes, dont deux orientées. La méthode d'extraction de crêtes sous-pixellique donne un résultat très propre, cependant, les crêtes

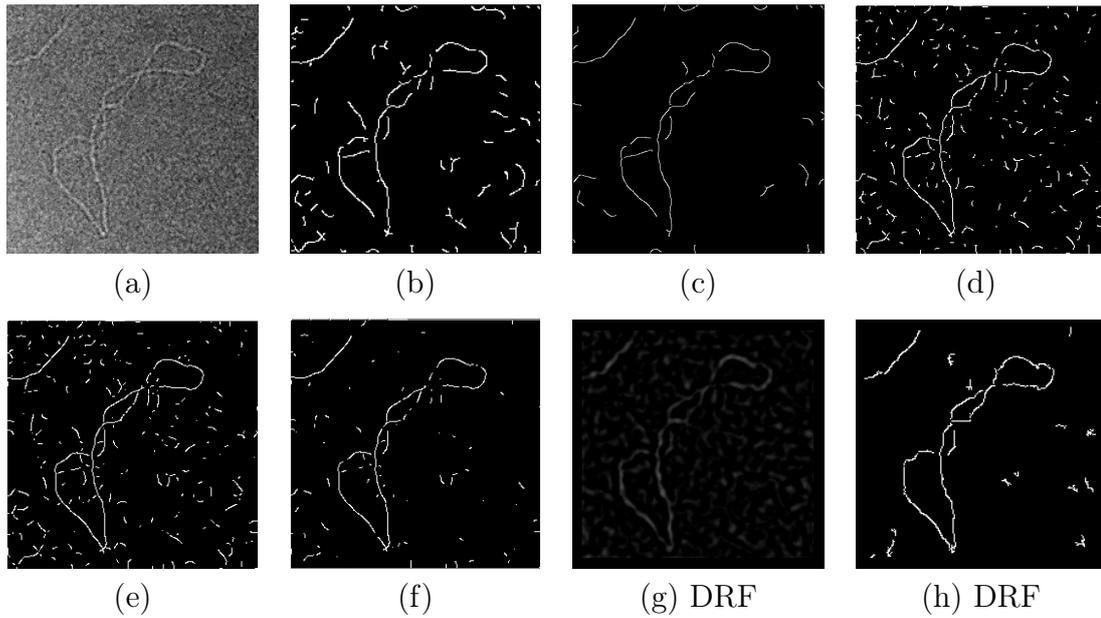


FIGURE 2.39 – Détection d’un filament d’ADN dans une image médicale avec différentes méthodes. (a) Image originale  $205 \times 205$ . (b) Détection de crêtes avec le tenseur de courbure [MAM95],  $\sigma = 3$ . (c) Détection sous pixelique de crêtes avec le tenseur de courbure [MD97b],  $\sigma = 3$ . (d) Détection de crêtes en utilisant la dérivée seconde orientée [FA91],  $\sigma = 3$ . (e) et (f) Détection de crêtes en utilisant la dérivée quatrième orientée [JU04],  $\sigma = 3$ . La différence entre (e) et (f) est que le seuil de (f) est plus fort, ce qui permet de supprimer du bruit, mais les crêtes du filament deviennent discontinues. (g) Crêtes détectées avec le DRF,  $\mu = 5$ ,  $\lambda_1 = 2$ ,  $\lambda_2 = 3$ . (h) Crêtes extraites avec le DRF.

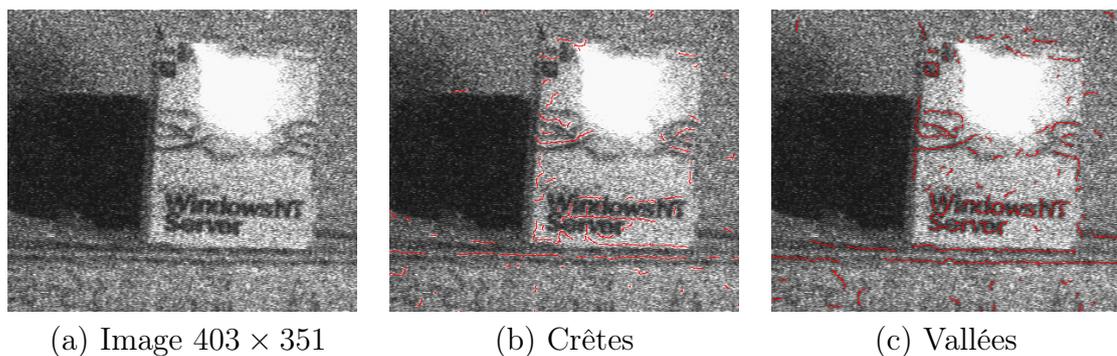


FIGURE 2.40 – Détection de crêtes et de vallées (en rouge) dans une image bruitée.  $\Delta\theta = 5^\circ$ ,  $\mu = 5$ ,  $\lambda_1 = 2$ ,  $\lambda_2 = 3$ ,  $\tau_b = 0.01$  et  $\tau_h = 0.03$ .

détectées formant le filament ne sont pas continues. Les méthodes orientées obtiennent des résultats intéressants [JU04]. Toutefois, le filament n’est pas totalement extrait lorsque l’on essaie d’éliminer les fausses détections dues au bruit contenu dans l’image originale ; contrairement à notre méthode qui est peu perturbée par de fausses détections.

Enfin, dans la figure 2.40, nous avons testé la détection des crêtes et des vallées

du DRF. Ce résultat nous satisfait grandement puisque notre approche est capable de détecter à la fois les courtes vallées créées par les lettres et les crêtes entre ces mêmes lettres tout en détectant les autres lignes de crêtes de l'image. De plus, le bruit dans cette image n'affecte pas la détection. La partie suivante est dédiée à l'évaluation du DRF en fonction du niveau de bruit dans l'image originale.

### 3.5 Evaluation de la Méthode

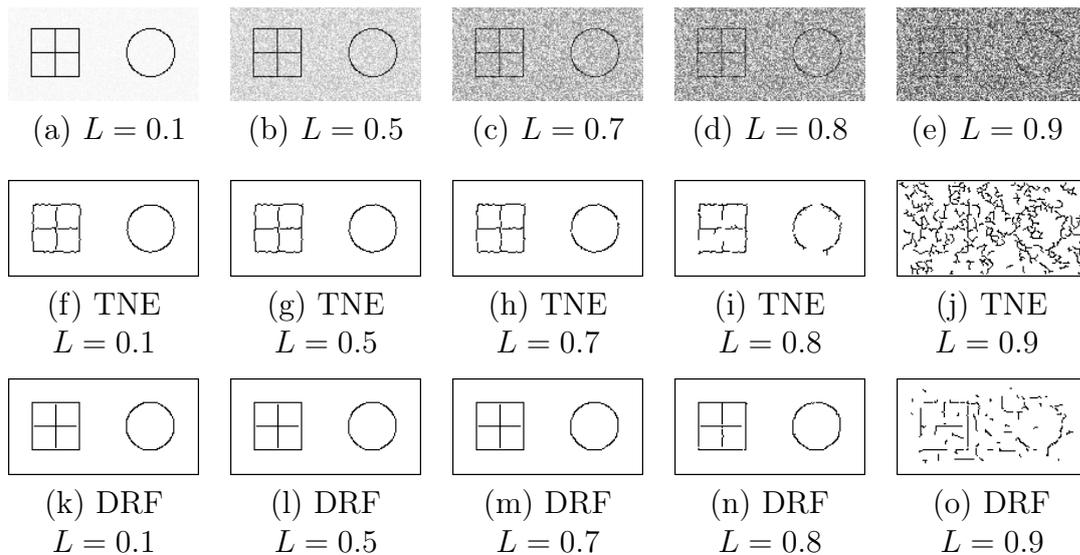


FIGURE 2.41 – Images  $160 \times 80$  avec différents niveaux de bruit  $L$  et extraction des vallées avec la méthode TNE et le DRF.

Afin d'évaluer le DRF, nous avons utilisé des résultats quantitatifs, nous avons également procédé à quelques tests avec des images synthétiques incluant des lignes de crêtes de largeur 1 pixel. La figure 2.41 montre un exemple d'une image bruitée à différents degrés  $L$  contenant quelques vallées correspondant à un carré et un cercle.

Dans notre test, nous estimons les vallées et nous comparons le résultat pixel par pixel avec l'image originale. Nous obtenons alors une erreur quantifiée en calculant la différence entre les deux images. Le tableau 2.1 illustre un certain nombre d'erreurs pour les paramètres suivants  $\lambda_1 = 1$ ,  $\lambda_2 = 1,5$  et  $\mu = 5$  pour chaque image. Dans cet exemple, nous remarquons que 346 pixels sur les 372 qui construisent la figure ont été correctement détectés alors que 26 pixels sont manquants. Comme 34 pixels ont été détectés comme indésirables, il y a donc au total  $60 = 34 + 26$  erreurs de détections pour le niveau de bruit  $L = 0.5$ .

TABLE 2.1 – Matrice montrant les erreurs de détection pour  $L = 0.5$ .

	Détection positive	Détection négative
Pixel vrai	346	26
Pixel faux	34	12394

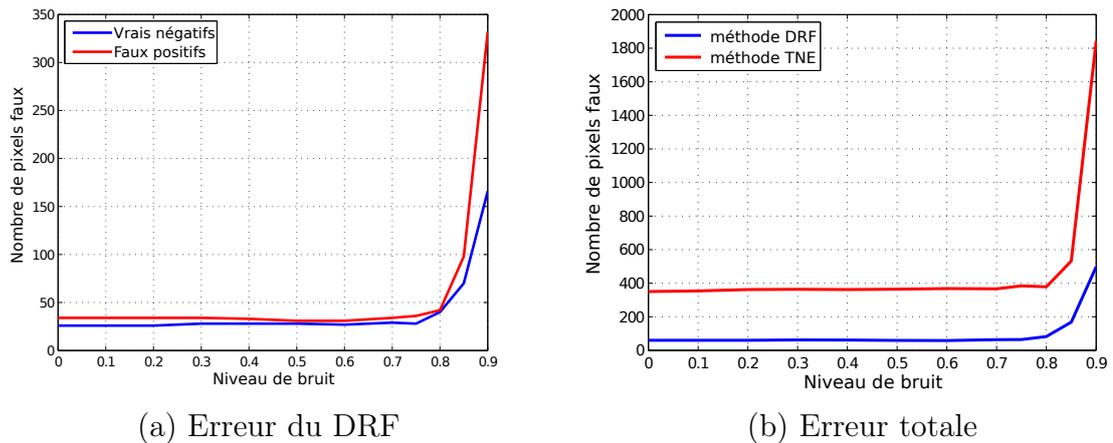


FIGURE 2.42 – Evaluation de l’erreur de notre approche. Comparaison de l’erreur totale avec la méthode TNE [MAM95].

**Influence du bruit.** Nous analysons l’effet d’un bruit blanc additif uniforme sur l’image originale en utilisant la formule suivante :

$$I_m = (1 - L)I_0 + L \cdot I_N, \quad L \in [0, 1],$$

où  $I_0$  correspond à l’image originale,  $I_N$  une image d’un bruit uniforme aléatoire et  $I_m$  l’image bruitée de sortie construite à partir de  $I_0$  et  $I_N$  en fonction d’un coefficient  $L$  tel que  $0 \leq L \leq 1$ . Deux courbes ont été affichées en figure 2.42(a), le nombre de pixels vrais négatifs et le nombre de faux positifs qui constituent tous les deux une erreur. Comme les courbes l’indiquent, le nombre d’erreurs croît monotoniquement avec le niveau de bruit  $L$ . Pour un niveau de bruit bas ( $L < 0.8$ ), des petites variations de l’erreur sont causées par l’effet d’échantillonnage : les lignes dans l’image sont projetées sur une grille carrée et binarisées, générant quelques imprécisions de quantifications. En particulier, le dessin d’un cercle peut différer d’une détection à l’autre, menant cependant à des représentations perceptuelles équivalentes.

Finalement, le nombre d’erreur reste relativement faible même lorsque le niveau du bruit est élevé, montrant ainsi la robustesse du DRF.

**Comparaison avec une autre méthode.** Dans un second temps, nous avons comparé les résultats du DRF avec la méthode dans [MAM95] appelée TNE (*Thin Nets Extraction*). Le nombre total d’erreurs, i.e. les faux positifs + les vrais négatifs ont été tracés en figure 2.42(b). Les deux méthodes montrent la même évolution de robustesse face au bruit, mais le filtre DRF est clairement plus performant que la méthode TNE. Le bruit délocalise les positions des maxima dans la méthode TNE, ce qui produit une erreur puisque les lignes de crêtes sont détectées avec une distance de 1 pixel. Concernant le DRF, le fort lissage dans les directions de chaque ligne de crête permet de ne pas délocaliser la détection (voire figure 2.31(b)).

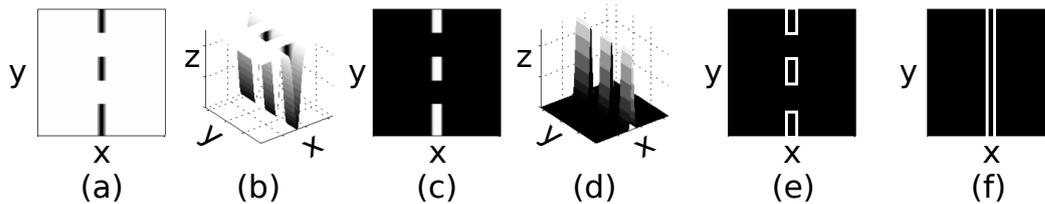


FIGURE 2.43 – Crête et vallée discontinues dans une image scalaire. (a) Vallée discontinue dans une image. (c) Crête discontinue dans une image. (b) et (d) Représentation de la surface respectivement de (a) et (c). *L'axe des z correspond à l'intensité des pixels en niveaux de gris.* (e) Détection de contours classique sur l'image (a). (f) Détection de contours anisotropique sur l'image (a).

### 3.6 Extraction de Courbes Perceptuelles

#### Saillance et Groupement Perceptuel

En vision par ordinateur, l'organisation perceptuelle a souvent été utilisée pour l'extraction de courbes dans les images [Alq98]. De nombreux travaux adressent le problème de qualité visuelle qualifiant les courbes qui est souvent référencé comme la saillance [AM98] [BL06] [GM96] [HH93] [SU88]. Dans cette partie, nous revisitons le groupement perceptuel et la saillance avec le point de vue du filtrage anisotrope (le filtre DRF). Contrairement à de nombreux travaux concernant l'organisation perceptuelle, l'approche proposée par le DRF n'est pas seulement valable sur les lignes binaires de l'image; le DRF est aussi capable de segmenter directement les images complexes comme les images aériennes vues plus haut.

Comme remarqué dans le chapitre 1, partie 6.5, les méthodes classiques de détection de bords ne sont pas adaptées pour l'extraction de lignes de crêtes dans les images puisqu'elles produisent deux contours de part et d'autre de la ligne de crête. Pour les lignes discontinues, le résultat des méthodes de détection de bords classiques sera des petits carrés ou rectangles (un exemple est disponible en figure 2.43 (e)). Le résultat des méthodes de détections de contours anisotropiques comme vu dans la partie 2.2 de ce chapitre ou par exemple [GSvdW02], sera deux lignes parallèles (illustré en figure 2.43 (f)).

Les méthodes classiques de détections de lignes de crêtes [MAM95] n'extraient

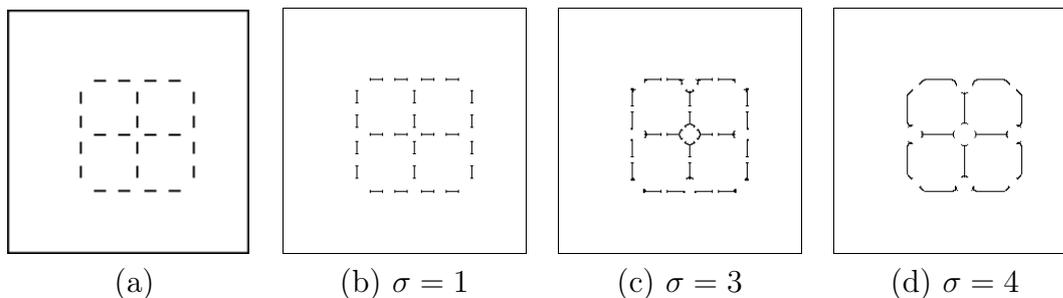
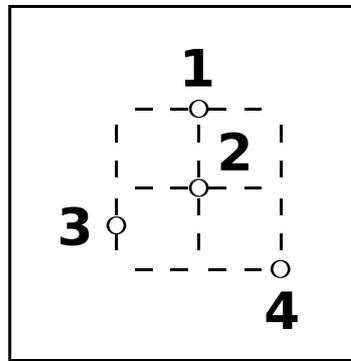


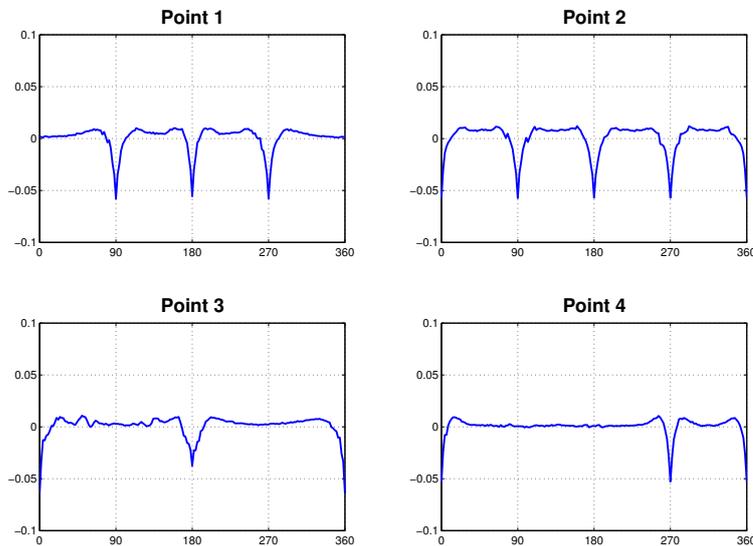
FIGURE 2.44 – Détection des vallées en utilisant les méthodes classiques [MAM95], il est impossible d'extraire les courbes perceptuelles. (a) Image originale  $120 \times 120$ . (b)  $\tau_b = 0.5$  et  $\tau_h = 0.8$ . (c)  $\tau_b = 0.7$  et  $\tau_h = 0.8$ . (d)  $\tau_b = 0.8$  et  $\tau_h = 0.8$ .

pas les courbes perceptuelles, sauf lorsque le paramètre de lissage  $\sigma$  croît, ce qui ne permet plus la détection de coins (illustré en figure 2.44).

**Un filtre adapté aux lignes de crêtes discontinues** Le DRF détaillé dans ce chapitre permet d'extraire les lignes de crêtes discontinues puisqu'il est formé de deux noyaux fins allongés qui lissent suffisamment l'information pour pouvoir extraire des crêtes ou des vallées perceptuelles. En effet, le calcul de  $\Sigma$  en équation 2.24 pour les pixels appartenant à des lignes, coins ou jonctions perceptuelles permet d'extraire les lignes de crêtes. La figure 2.45 illustre que les pics sont toujours visibles dans le signal  $D(x, y, \theta)$  (voir équation 2.23).



(a) Sélection de points dans une image  $120 \times 120$



(b) Signal  $D(x, y, \theta)$  associé aux points de (a)

FIGURE 2.45 – Comportement du filtre sur des points de courbes perceptuelles. Les points 1 et 2 correspondent à des pixels de jonctions perceptuelles, le point 3 à une ligne perceptuelle et le point 4 à un coin perceptuel.

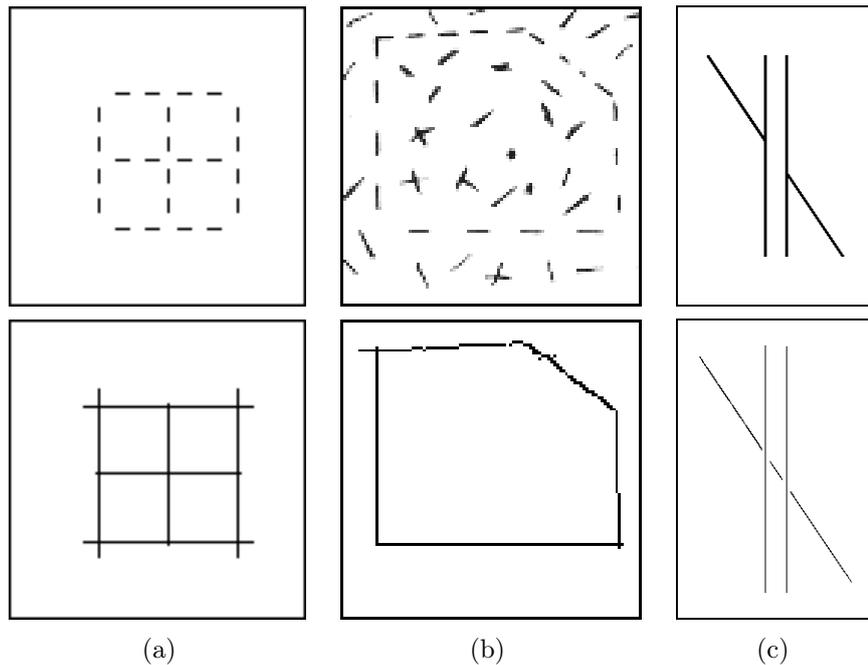


FIGURE 2.46 – Détection de vallées perceptuelles. De haut en bas : image originale et détection de vallées. (a) Cinq carrés perceptuels ( $118 \times 120$ ). Détection des vallées perceptuelles avec  $\Delta\theta = 2^\circ$ ,  $\mu = 15$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$ ,  $\tau_b = 0.02$  et  $\tau_h = 0.12$ . (b) Pentagone perceptuel ( $100 \times 100$ ). Détection des vallées perceptuelles avec  $\Delta\theta = 2^\circ$ ,  $\mu = 10$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$ ,  $\tau_b = 0.01$  et  $\tau_h = 0.05$ . (c) Illusion de Poggendorff ( $210 \times 315$ ). Détection des vallées perceptuelles avec  $\Delta\theta = 2^\circ$ ,  $\mu = 30$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$ ,  $\tau_b = 0.0$  et  $\tau_h = 0.04$ .

## Résultats

Nous présentons des résultats obtenus à la fois sur des images contenant des courbes perceptuelles et des contours discontinus obtenus à partir d'une image bruitée.

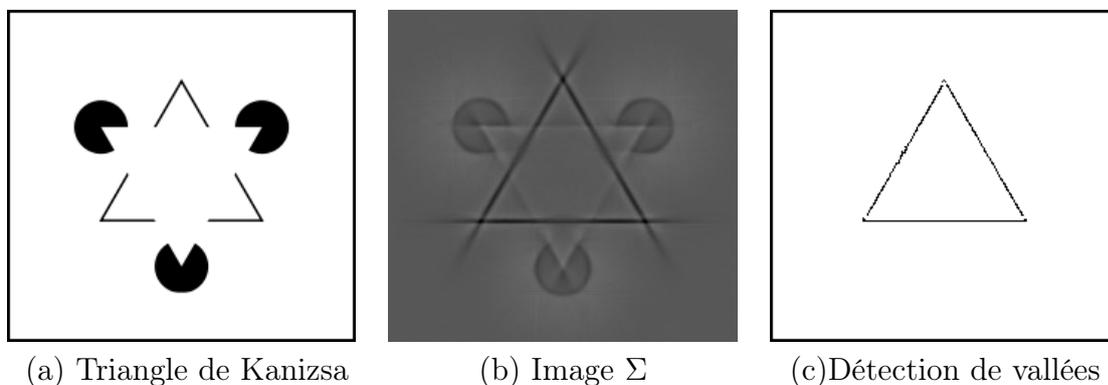


FIGURE 2.47 – Détection de vallées perceptuelles sur le triangle de Kanizsa ( $230 \times 220$ ).  $\Delta\theta = 2^\circ$ ,  $\mu = 20$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$ ,  $\tau_b = 0.01$  et  $\tau_h = 0.06$ .

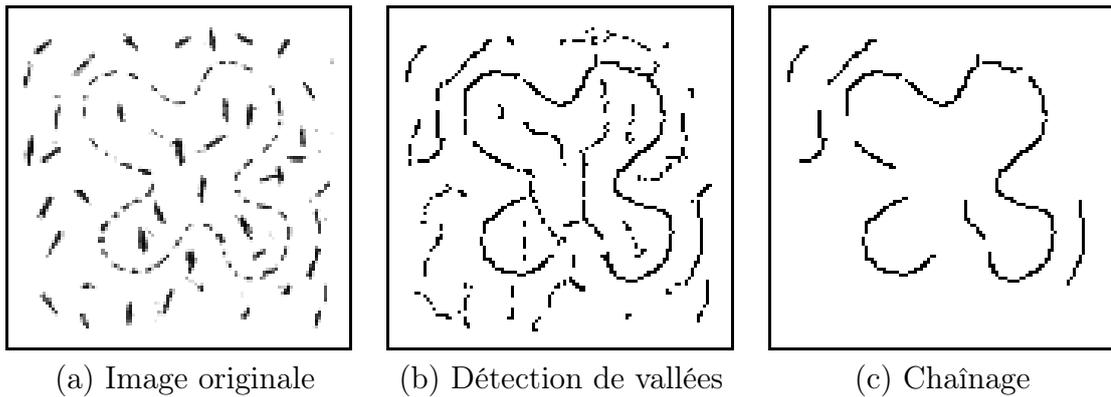


FIGURE 2.48 – Détection de vallées perceptuelles. (a) Image ( $100 \times 100$ ). (b) Résultat de la détection de vallées perceptuelles  $\Delta\theta = 2^\circ$ ,  $\mu = 5$ ,  $\lambda_1 = 1.5$ ,  $\lambda_2 = 2.25$ ,  $\tau_b = 0.001$  et  $\tau_h = 0.01$ . (c) Chaînage sur (b) avec une taille minimale de la chaîne égale à 12 pixels.

**Résultat sur des courbes perceptuelles.** Dans le premier groupe d’images de la figure 2.46, le détecteur DRF est capable de compléter les lignes discontinues,

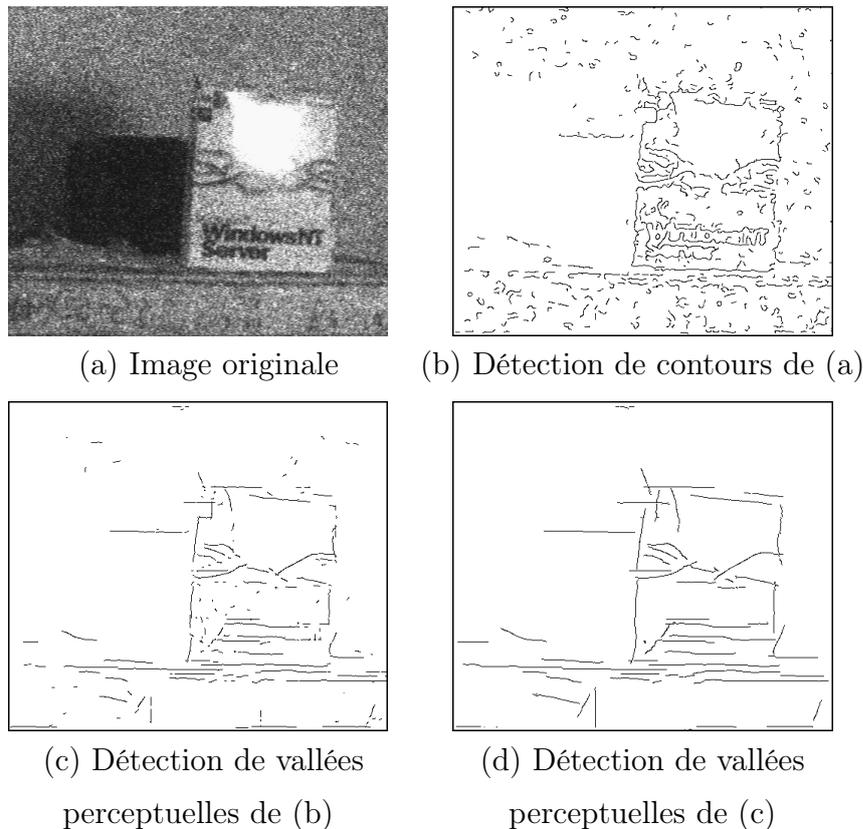


FIGURE 2.49 – Détection de vallées à partir de contours. (a) Image originale bruitée ( $500 \times 437$ ). (b) Détection isotropique de contours [Der92],  $\sigma = 2$ ,  $\tau_b = 0.03$  et  $\tau_h = 0.04$ . (c) Détection de vallées perceptuelles de (b),  $\Delta\theta = 2^\circ$ ,  $\mu = 20$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$ ,  $\tau_b = 0.02$  et  $\tau_h = 0.025$ . (d) Détection de vallées perceptuelles de (c),  $\Delta\theta = 2^\circ$ ,  $\mu = 20$ ,  $\lambda_1 = 2$ ,  $\lambda_2 = 3$ ,  $\tau_b = 0.01$  et  $\tau_h = 0.02$ .

même pour les points de coins . Nous pouvons noter que le bruit causé par les petits segments de la figure 2.46 (b) ne perturbe pas l'extraction de courbes perceptuelles.

Le filtre DRF a été testé sur l'illusion classique triangle de Kanizsa en figure 2.47. Juste perçu par l'oeil humain, le triangle équilatéral peut être reconstitué par le DRF en utilisant un paramètre  $\mu$  suffisamment large. Pourtant, une valeur de  $\mu$  large ne génère pas de mauvaise détection.

Dans la figure 2.48, la difficulté est d'extraire la forme chromosomique puisque les petits objets perturbent la perception. Le DRF extrait correctement la forme et après un chaînage, nous sommes capables de reconstruire presque entièrement le chromosome sans présenter trop de fausses courbes perceptuelles même au niveau des courbes coudées.

Clairement, notre méthode s'aligne bien à la perception humaine et peut être utilisée efficacement pour l'analyse et l'interprétation des images [Low84].

**Résultat sur des contours perceptuels.** Les résultats de détection de contours peuvent renvoyer des contours incomplets ou faux, spécialement dans le cas d'images bruitées comme illustré en figure 2.49 (b). Comme le filtre DRF est performant dans le cas de courbes perceptuelles, les contours peuvent être améliorés. Après deux itérations, il reste seulement les contours des objets [Low84].

### 3.7 Détection de Jonctions de Lignes de Crêtes

#### Classification des pixels

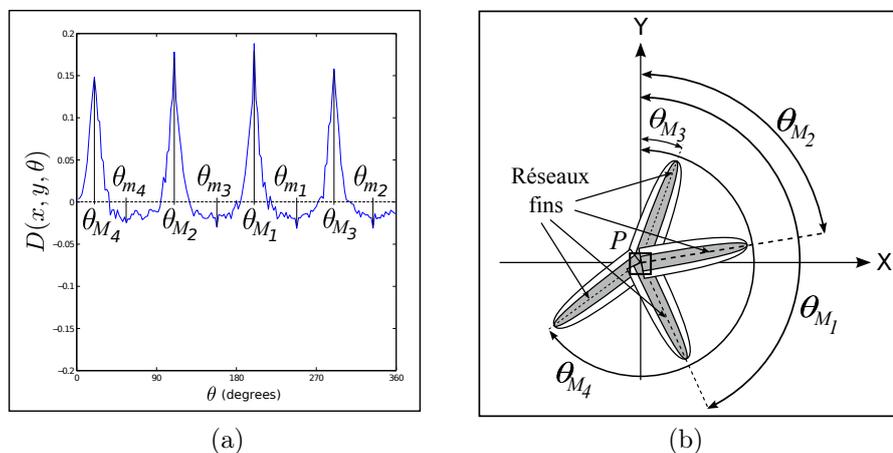


FIGURE 2.50 – (a) Exemple de  $\theta_{M_i}$  et  $\theta_{m_i}$  en fonction de  $D$  au niveau d'une jonction de crêtes. (b)  $\theta_{M_i}$  correspond aux directions des crêtes ( $i \in \{1, 2, 3, 4\}$ ).

Dans cette partie, nous proposons une nouvelle méthode de détection de jonctions de lignes de crêtes basée sur la différence de deux demi-filtres de lissage anisotrope gaussiens. Cette approche est similaire à celle de détection de lignes crêtes présentée dans ce chapitre sauf qu'elle est basée sur un seuillage spatial et non sur la recherche de maxima dans une direction précise.

Nous utilisons les mêmes noyaux que le DRF : la différence de deux demi-filtres gaussiens de mêmes longueurs  $\mu$  mais de largeurs différentes ( $\lambda_1, \lambda_2$ ) pour calculer une valeur  $D(x, y, \theta)$  en fonction de la direction du filtre  $\theta$  (équation 2.23).

Nous définissons un opérateur de jonctions de crêtes/vallées  $J$  en un pixel de coordonnées  $(x, y)$  par l'expression suivante :

$$J(x, y) = \sum_{k=1}^4 (D(x, y, \theta_{M_k}) + D(x, y, \theta_{m_k})), \quad (2.25)$$

où  $\theta_{M_i}$  correspondent aux directions des maxima locaux de la fonction  $D$  et  $\theta_{m_i}$  aux directions des minima locaux, avec  $i \in \{1, 2, 3, 4\}$ . Les valeurs de  $\theta_{M_i, i \in \{1, 2, 3, 4\}}$  sont rangées en ordres décroissant et celles des  $\theta_{m_i, i \in \{1, 2, 3, 4\}}$  en ordre croissant. Le filtre orienté aura une forte réponse (positive ou négative dans le cas d'une crête ou d'une vallée) dans la direction d'une ligne de crête. Comme les jonctions sont formées d'au moins trois chemins entrants mais que la plupart des jonctions sont l'intersection de quatre lignes de crêtes (souvent des carrefours en X), nous calculons les quatre maxima et minima locaux les plus importants.

Les conditions de la détection sont les suivantes :

$$\text{si } \begin{cases} J(x, y) > J_{th} \\ D(x, y, \theta_{M_1}) < 2 \cdot D(x, y, \theta_{M_3}) \end{cases} \quad (2.26)$$

le pixel  $P$  appartient à une jonction de crêtes,

$$\text{si } \begin{cases} J(x, y) < -J_{th} \\ D(x, y, \theta_{m_1}) > 2 \cdot D(x, y, \theta_{m_3}) \end{cases} \quad (2.27)$$

le pixel  $P$  appartient à une jonction de vallées, où  $J_{th} > 0$ . La figure 2.51(a) illustre le résultat de  $J$  sous la condition de l'équation 2.27 avec deux jeux de paramètres différents. Sur une jonction de vallée typique, le pixel contiendra au moins trois pics étroits négatifs. Pour une jonction de crêtes, le signal du pixel contiendra

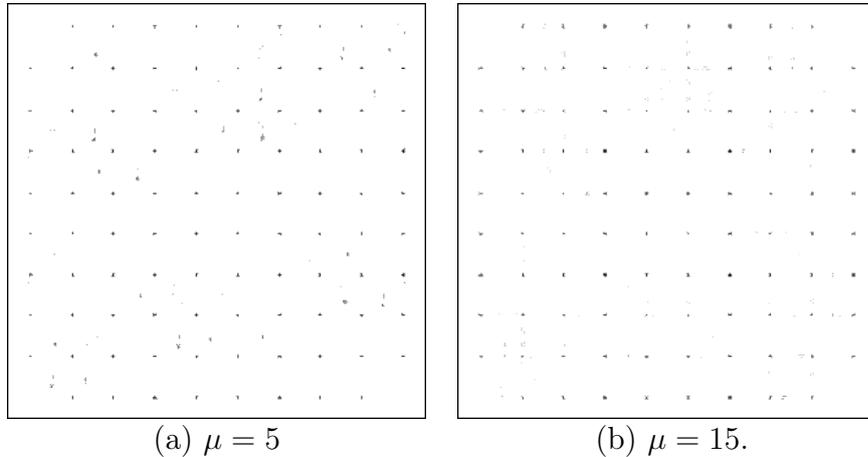


FIGURE 2.51 – Estimation des  $J$  avec la condition  $D(x, y, \theta_{m_1}) > 2 D(x, y, \theta_{m_3})$  de l'équation 2.27 sur l'image 2.52 (a) avec le paramètre  $\mu$  différent,  $\Delta\theta=2^\circ$ ,  $\lambda_1=1$ ,  $\lambda_2=1.5$  Ces images sont normalisées.

au moins trois pics positifs. Ces pics étroits correspondent aux directions des lignes de crêtes, alors que le nombre de pics correspondent au nombre de crêtes ou de vallées au niveau de la jonction. La deuxième condition  $|D(x, y, \theta_{M_1})| < |2 \cdot D(x, y, \theta_{M_3})|$  dans les équations 2.26 et 2.27 signifie que le troisième plus haut pic (en valeur absolue) est suffisamment conséquent pour ne pas être causé par le bruit ou d'autres structures. De par cette condition, ce troisième pic est associé à une ligne de crête dans la direction  $\theta_{M_3}$  ou  $\theta_{m_3}$ . Cette condition est totalement arbitraire mais elle est nécessaire afin d'extraire les jonctions.

Cependant, au niveau d'un contour, la valeur absolue de  $J$  est proche de 0 car les valeurs absolues de  $D$  au niveau de  $\theta_{M_1}$ ,  $\theta_{M_2}$ ,  $\theta_{m_1}$  et  $\theta_{m_2}$  sont proches. Enfin, comme le filtre lisse beaucoup dans la direction des lignes de crêtes, il est robuste au bruit.

	4				2		1	9
			3	5	1		8	6
3	1			9	4	7		
	9	4						7
2							8	9
		9	5	2			4	1
4	2		1	6	9			
1	6		8					7

(a) Image originale  
320×320

	4				2		1	9
			3	5	1		8	6
3	1			9	4	7		
	9	4						7
2							8	9
		9	5	2			4	1
4	2		1	6	9			
1	6		8					7

(b) Toutes les jonctions, même les nombres,  
 $\mu = 5, \lambda_1 = 1, \lambda_2 = 1.5$

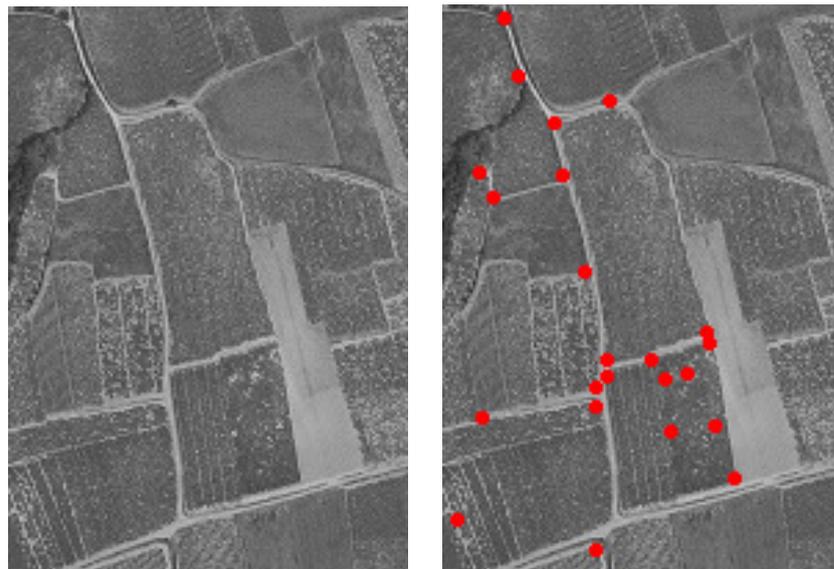
	4				2		1	9
			3	5	1		8	6
3	1			9	4	7		
	9	4						7
2							8	9
		9	5	2			4	1
4	2		1	6	9			
1	6		8					7

(c) Jonctions des lignes,  
 $\mu = 10, \lambda_1 = 1, \lambda_2 = 1.5$ .

	4				2		1	9
			3	5	1		8	6
3	1			9	4	7		
	9	4						7
2							8	9
		9	5	2			4	1
4	2		1	6	9			
1	6		8					7

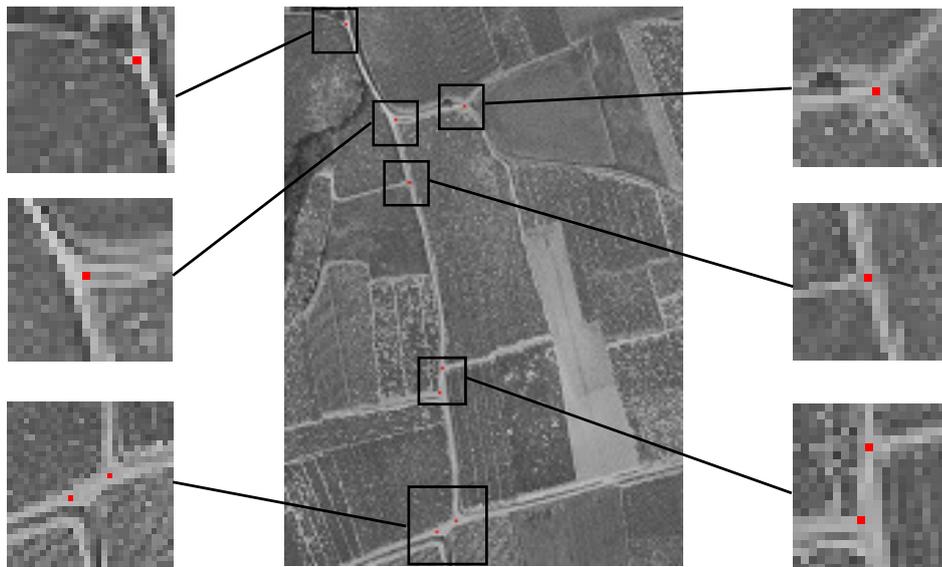
(d) Jonctions des lignes larges,  
 $\mu = 15, \lambda_1 = 2, \lambda_2 = 3$ .

FIGURE 2.52 – Détection de jonctions sur une grille de sudoku (en rouge).  $\Delta\theta = 2^\circ$ . Seuillage spatial pour (a)  $J_{th} = 0.25$ , pour (b)  $J_{th} = 0.3$ , pour (c)  $J_{th} = 0.35$ . Les jonctions détectées ont été dilatées pour une meilleure visualisation.



(a) Image originale  $150 \times 200$

(b) Détecteur de Harris,  
 $\sigma = 1$  [SMB00].



(c) Notre résultat (en rouge) superposé sur (a)

FIGURE 2.53 – Détection de jonctions de routes dans une image aérienne Pour (c) :  $\Delta\theta = 2^\circ$ ,  $\mu = 10$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$  et  $J_{th} = 0.05$ .

### Extraction des jonctions de lignes de crêtes

Une fois  $J(x, y)$  calculé et l'équation 2.26 ou 2.27 vérifiée (illustré en figure 2.51), nous procédons simplement à l'estimation locale spatiale des maxima de  $J$  pour une jonction de crêtes et des minima de  $J$  pour les vallées.

## Résultats

Nous présentons des résultats à la fois sur une image synthétique et sur une image réelle. Selon le jeu de paramètre proposé pour les demi-noyaux anisotropes de gaussiennes, les résultats ne sont pas les mêmes.

**Résultats sur une image synthétique :** Dans le premier résultat illustré en figure 2.52, le but est de détecter les jonctions d'une grille de sudoku. Toutefois notre méthode est capable de détecter les jonctions à l'intérieur des nombres 1, 4, 6 et 9 selon le jeu de paramètres proposé. Notre algorithme est aussi capable d'extraire les jonctions de toutes les lignes mais aussi uniquement les lignes larges avec un jeu de paramètres adaptés  $\lambda_1$ ,  $\lambda_2$  et  $\mu$ . En effet, plus les noyaux vont être courts (paramètre  $\mu$ ), plus les jonctions détectées vont être celles de lignes courbées (par exemple les chiffres). Tout comme la détection de lignes de crêtes, la largeur de ces noyaux (les paramètres  $\lambda_{i, i \in \{1,2\}}$ ) définit les jonctions dont la largeur des lignes est désirée. Plus les paramètres  $\lambda_{i, i \in \{1,2\}}$  vont croître, plus les lignes vont être larges (illustré en figure 2.52 (d)). Il est également important de choisir une discrétisation d'angle relativement fine (ici  $\Delta\theta = 2^\circ$ ) afin d'obtenir des pics étroits et les plus haut possibles.

**Résultats sur une image réelle :** Nous avons testé notre algorithme sur une image aérienne contenant des routes et des vignes. Le but est d'extraire les jonctions de routes. Notre approche est capable d'extraire correctement les carrefours de routes sans être perturbée par les textures parallèles des vignes, contrairement aux méthodes de coins classiques [HS88] [SMB00] [KR82] qui détectent plusieurs points avec un écart de localisation par rapport aux jonctions de lignes de crêtes lorsqu'elles ne sont pas perturbées par la texture.

## Evaluation des résultats

Afin d'évaluer notre méthode avec des résultats quantitatifs, nous avons conduit un nombre de tests avec les images synthétiques de largeur un pixel pour les vallées.

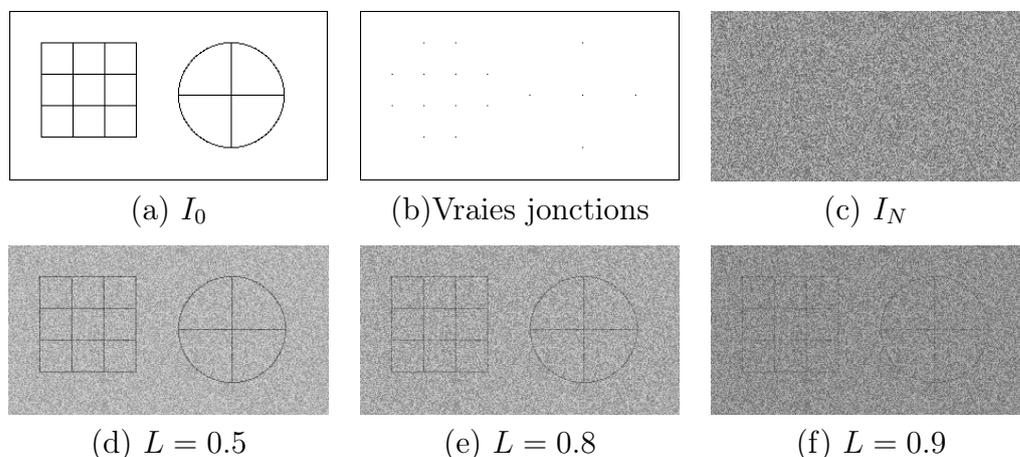


FIGURE 2.54 – Images  $300 \times 160$  with different levels of noise  $L$ .

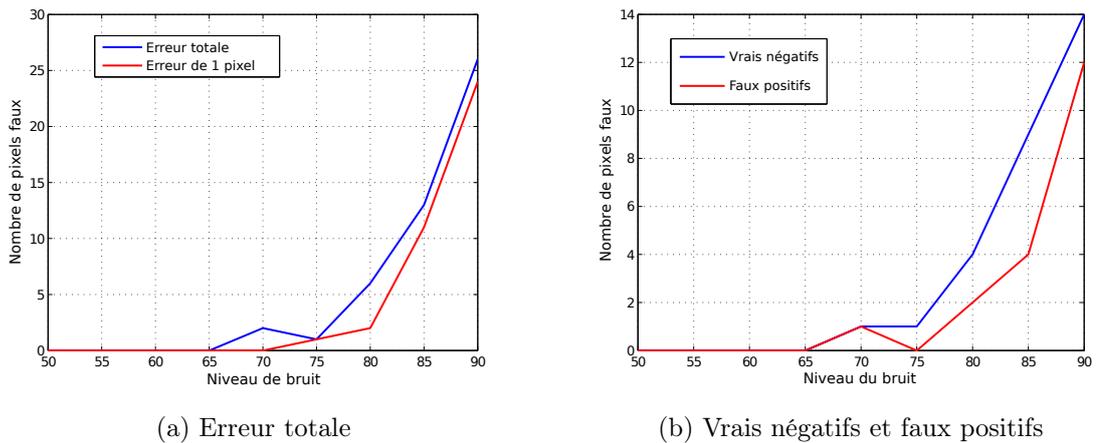


FIGURE 2.55 – Evaluation de l’erreur de notre approche.

La figure 2.54(b) montre un exemple des jonctions de vallées avec une image simple composée de carrés et d’un cercle contenant une croix.

Dans notre test, nous détectons les jonctions des vallées et comparons à l’image de vérité terrain, pixel par pixel. Nous obtenons une erreur quantifiée en faisant la différence entre les deux images. Nous avons utilisé les paramètres suivants pour toutes les images de ce test :  $\lambda_1 = 1$ ,  $\lambda_2 = 1.5$  et  $\mu = 10$ .

**Influence du bruit :** Tout comme pour évaluer l’influence du bruit pour le DRF en partie 3.5 de ce chapitre, nous analysons l’effet d’un bruit blanc uniforme additif sur une image originale. En construisant une image bruitée  $I_m$  en fonction d’un coefficient  $L$  ( $0 \leq L \leq 1$ ) par la formule  $I_m = (1 - L)I_0 + L \cdot I_N$  où  $I_0$  est l’image originale,  $I_N$  une image d’un bruit aléatoire uniforme, nous estimons les jonctions des différentes images  $I_m$ . Les jonctions ainsi détectées sont comparées avec celles de la vérité terrain afin de calculer les erreurs. Comme indiqué en figures 2.55 (a) et (b), le nombre d’erreurs croît en fonction du niveau du bruit. Le nombre d’erreurs est relativement faible, même pour un niveau de bruit élevé ( $L = 0.8$ ), montrant ainsi la bonne robustesse de notre filtre. Ce filtre est composé d’un fort lissage dans la direction des lignes de crêtes tout en restant très étroit, ce qui permet de ne pas délocaliser les la détection de jonctions (illustré en figure 2.50 (b)).

### 3.8 Conclusion

Dans cette partie, nous avons présenté une nouvelle méthode de détection de lignes de crêtes. Cette approche est basée sur la différence de deux demi-filtres orientés linéaires de lissage puis d’une maximisation ou minimisation locale pour extraire respectivement les crêtes et les vallées. Notre algorithme est capable de détecter les lignes de crêtes même lorsqu’elles sont fortement coudées. De plus, comme le filtre est formé de deux demi-filtres de lissage orientés, notre approche estime les deux orientations d’une ligne de crête et les deux principales directions d’une jonction. Aussi, le fort lissage dans la direction de la ligne de crête rend la méthode fortement robuste au bruit. Cette approche a été testée avec succès sur des types d’images va-

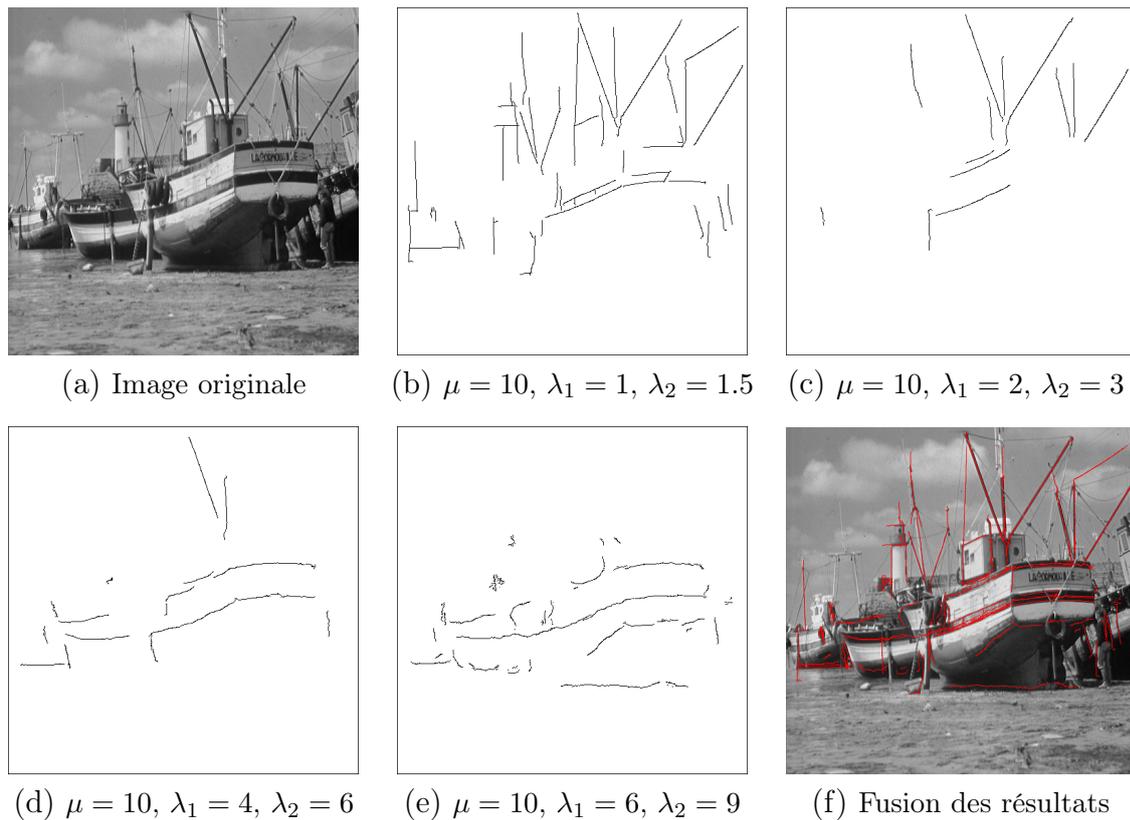


FIGURE 2.56 – Extraction des vallées à différentes échelles avec  $\Delta\theta = 2^\circ$  et réunion par addition de toutes les vallées extraites projetées sur l'image originale dans (f).

riées présentant des problèmes difficiles pour des méthodes de détections classiques de lignes de crêtes. Par ailleurs, les noyaux composants le filtre sont assez longs pour détecter les lignes de crêtes discontinues même coudées contrairement aux méthodes classiques. L'avantage de cet algorithme est qu'il fonctionne aussi bien sur les images binaires qu'en niveaux de gris. Il peut par exemple détecter directement les routes dans une image satellite, tout comme reconstruire des contours binaires issus d'une image bruitée. Les deux directions d'une courbe ainsi extraite peuvent apporter une information importante qui pourrait être utilisée en diffusion des images, comme pour la reconstruction de réseaux fins discontinus ou dans des images bruitées.

La détection de crêtes ou de vallées ainsi présentée avec la réponse maximale de deux noyaux de lissages dans deux directions différentes permet également d'estimer les jonctions de lignes de crêtes. Tout comme la méthode de détection de lignes de crêtes, cette approche est peu sensible au bruit, une application potentielle serait de détecter les carrefours dans des images satellites routières.

Une amélioration du DRF serait de l'utiliser en multi-échelle afin d'extraire les lignes de crêtes de différentes largeurs [AMM97] [Lin98]. La figure 2.56 illustre la détection de vallées à différentes échelles. En faisant varier la largeur du filtre seulement (les paramètres  $\lambda_i, i \in \{1, 2\}$ ), une méthode de sélection automatique pourrait être développée en cherchant les maxima (pour les crêtes) ou les minima (pour les vallées) dans la direction  $\eta$  correspondante à la valeur de  $|\Sigma|$  la plus forte. Enfin, comme les lignes des crêtes correspondent à des maxima (ou minima) dans les images, il faut

draît supprimer les crêtes ne correspondant pas à des minima du gradient [AMM97] en utilisant le détecteur anisotrope de contours (détaillé en partie 2.2 de ce chapitre). Une application éventuelle de cette approche serait de détecter les personnes dans les images, les bras et les jambes doivent être détectés avec un filtre moins large que celui pour le corps ou le torse. Aussi, sur des images contenant des arbres, il serait possible d'extraire les branches de différentes tailles et ainsi les reconstruire en trois dimensions.

## 4 Régularisation des images par diffusion anisotro- pique

### 4.1 Introduction

Dans cette partie, afin de régulariser les images, nous nous sommes intéressés dans un premier temps à la suppression de texture dans les images. Il est difficile d'éliminer la texture et de préserver les principales structures [BLMV10]. Nous avons ainsi développé deux méthodes de diffusion de texture qui permettent de régulariser les images en préservant les bords des principaux objets. Comme cette méthode conservait les contours dans les images, nous l'avons améliorée pour la restauration des images, même très fortement bruitées.

L'analyse de texture dans les images est un domaine de recherche actif depuis plus de quarante ans [Har05] [EL99] puisque les textures rendent la segmentation des images difficiles. La texture est liée à la distribution spatiale ou statistique des niveaux de gris ou des intensités colorimétriques [KJB96]. Elle contient des informations importantes concernant l'arrangement structurel des surfaces des images et leurs relations avec leur environnement direct. Il est facile pour un observateur humain de reconnaître la texture [Jul81], toutefois, il devient difficile de la définir précisément et de l'analyser automatiquement [PD02]. Cette difficulté est reflétée par le grand nombre de définitions différentes de la texture. Dans les parties suivantes de ce manuscrit, nous n'analysons pas la texture mais simplement essayons de l'identifier et puis de la diffuser anisotropiquement telle qu'elle apparaisse comme une région homogène.

La détection de contours dans les images naturelles ou contenant beaucoup de textures a motivé ce travail. En effet, l'extraction des contours avec les méthodes classiques en utilisant un petit filtre (par exemple une valeur de l'écart type de la gaussienne  $\sigma = 1$ ), les contours des textures vont polluer le résultat final (voir figure 2.13). Dans le cas contraire, un écart type trop large perdra la précision des contours et spécialement les coins.

Afin de préserver les principales structures de l'image, des filtres à fenêtres adaptatives peuvent être utilisés. Ainsi, les filtres de Kuhawara [KHEK76] et de Nagao [NM79] calculent la valeur moyenne des voisinages d'un masque  $5 \times 5$  et conservent celle qui a la plus petite variance. Le filtre bilatéral est souvent utilisé pour lisser les images et préserver les contours [TM98] [PKT09]. Il est défini comme une moyenne pondérée des pixels voisins dans un masque carré centré sur un pixel (ce filtre est détaillé en partie 7.1 du chapitre précédent). Il permet un filtrage local sélectif qui propage l'information d'intensité en préservant les discontinuités.

Buades et al. [BLMV10] ont développé une approche supprimant la texture en préservant les bords. En effet, cet algorithme détermine si un pixel appartient à une zone homogène ou à une texture par calcul de la variation locale totale dépendant d'une gaussienne d'écart type  $\sigma$  faisant office d'échelle. Si le pixel appartient à une partie texturée de l'image, sa variation locale totale sera forte. Si le paramètre d'échelle  $\sigma$  est petit, les textures fines seulement seront supprimées. Cependant, avec un paramètre  $\sigma$  élevé, les textures plus marquées seront effacées mais également les petits objets ainsi que les contours.

En restauration des images par EDP [AK06] [DF95], la détection de contours est souvent utilisée pour détecter les bords des images afin de contrôler un processus de diffusion. Ainsi, dans les régions homogènes, la diffusion est isotropique et au niveau des points de bords, la diffusion est inhibée [PM90] ou anisotropique [ALM92]. Néanmoins, il est difficile de distinguer entre le bruit, la texture et les petits objets qui ont besoin d'être préservés par le processus de diffusion.

En restauration d'images couleurs, nous avons présenté dans le chapitre 1, partie 8.8, un certain nombre de schémas de diffusion utilisant des normes de gradients couleurs [DZ86] [SR96] [TD02], la divergence [Wei99] ou la trace [TD05] [Tsc06] du tenseur de structure afin de contrôler le processus de diffusion. Aussi, toutes les composantes d'une image couleur ne doivent pas être diffusées indépendamment pour ne pas perdre la diffusion couplée [BC98].

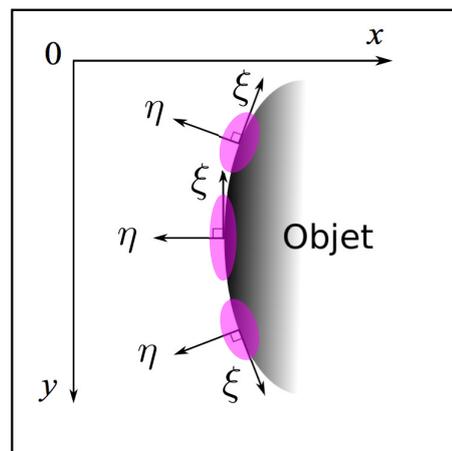


FIGURE 2.57 – Diffusion anisotropique pour des pixels de contours. Plus le gradient est fort, plus la diffusion est anisotropique et tangentielle au contour, les ellipses de couleur magenta représentent ce phénomène.

Dans les parties suivantes, nous présentons un demi-filtre rotatif capable de détecter les textures dans les images en niveaux de gris puis une classification pour les textures en images couleurs. Puis nous introduisons une nouvelle méthode de diffusion qui contrôle la diffusion près des contours et les coins. En particulier, notre détecteur produit deux directions différentes pour les bords, permettant de préserver les coins. Ces informations permettent une diffusion anisotropique dans ces deux directions contrairement aux méthodes de diffusion classiques et récentes où une seule direction est considérée : celle du contour appelée  $\xi$  [ALM92] [AK06] [Wei99] [TD05] [Tsc06] (la direction perpendiculaire au contour est également considérée pour la diffusion,

mais négligeable concernant les points de fort gradient comme illustré dans le schéma de la figure 2.57). En effet, au lieu de diffuser dans une seule direction tangente au contour (modulo  $180^\circ$ ), notre schéma de diffusion lisse les contours dans deux directions  $(\xi_1, \xi_2)$  correspondantes aux deux directions d'un contour (un chemin entrant et un chemin sortant en chaque pixel de contour).

Nous commencerons par présenter un demi-filtre rotatif lissant, capable de détecter les textures. Puis nous introduirons une nouvelle classification de pixels utilisant une banque d'images filtrées. Notre schéma de diffusion sera alors introduit avant d'étendre notre processus de diffusion aux images vectorielles. Par la suite nous allons améliorer cette approche en introduisant le détecteur de contours anisotrope présenté en partie 2.2 de ce chapitre, qui permet de diffuser les textures tout en préservant plus précisément les bords. Enfin, nous présenterons également un nouveau schéma de restauration d'images fortement bruitées avec des fonctions de contrôle au niveau des contours et des coins qui préserve les principales structures de l'image ainsi que les coins.

Les résultats présentés sont comparés à d'autres méthodes de régularisation d'images avec ou sans EDP. Dans la dernière partie, nous apporterons des résultats quantitatifs afin de montrer la supériorité de notre approche dans le cas d'images fortement bruitées.

## 4.2 Une détection de contours larges pour les principales structures de l'image

Dans notre méthode de détection de contours larges, en chaque pixel de l'image originale, nous utilisons les demi-noyaux gaussiens orientés  $G_{(\mu,\lambda)}(x, y, \theta)$  de l'équation 2.2 afin de construire un signal  $s$ . Les variables  $(x, y)$  représentent les coordonnées du pixel considéré et  $(\mu, \lambda)$  les écarts types du filtre gaussien orienté d'angle  $\theta$ . Ce signal est une fonction de l'angle de rotation  $\theta$  et de la réponse du filtre en chaque pixel. Par convolution avec ces noyaux orientés, comme pour notre détection de contours toits, nous obtenons une collection d'images lissées orientées  $I_\theta = I * G_{(\mu,\lambda)}(\theta)$ . La figure 2.58 illustre une image contenant plusieurs textures lissée à différentes orientations avec  $G_{(\mu,\lambda)}(x, y, \theta)$ .

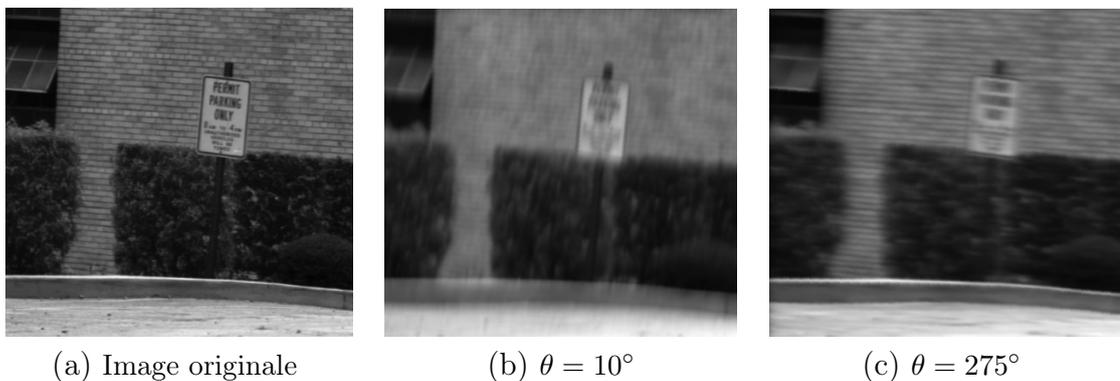
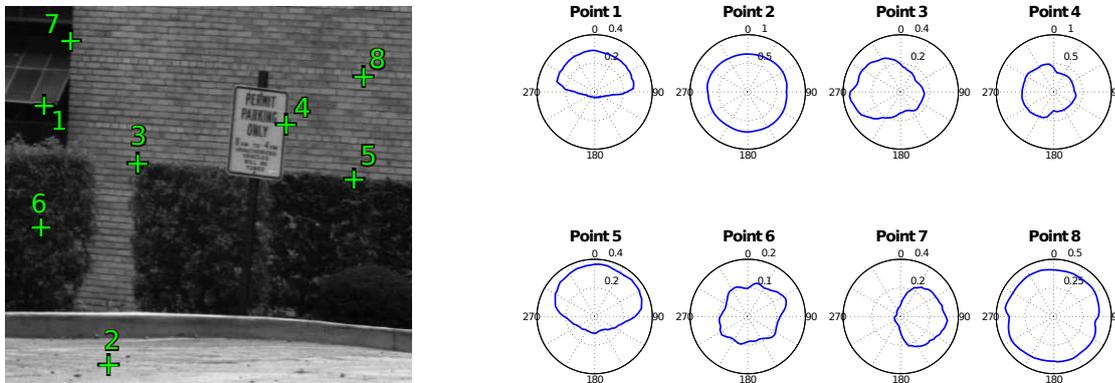


FIGURE 2.58 – Image  $420 \times 395$  lissée à différentes orientations avec  $\mu = 10$  et  $\lambda = 1$ .

Un signal pour chaque pixel



(a) Sélection de points sur une image texturée (en vert)

(b) Représentation polaire des  $s(\theta)$  pour chaque point de (a)

FIGURE 2.59 – Sélection de points et leur signal associé,  $\mu = 10$ ,  $\lambda = 1$  et  $\Delta\theta = 5^\circ$ .

L'application du filtre orienté produit un signal  $s$  en chaque pixel d'une image en niveaux de gris. Le signal du pixel est une fonction unique  $s(\theta)$  de l'orientation d'angle  $\theta$ . La figure 2.59(b) est un exemple de plusieurs fonctions  $s$  mesurées en 8 points situés sur l'image de la figure 2.59(a). Chaque tracé de la figure 2.59(b) représente la fonction  $s(\theta)$  en coordonnées polaires d'un point particulier. En chaque pixel, à partir de ces signaux, nous extrayons le descripteur qui discriminera contours et différentes régions. Dans le cas d'un pixel dans une région homogène ou de texture très fine, la fonction  $s(\theta)$  sera constante (illustré en figure 2.59, point 2). Au contraire, dans une région texturée, la fonction  $s(\theta)$  sera aléatoire (comme les points 6 et 8 de la figure 2.59, la figure 2.60 illustre le signal  $s$  du point 8 en coordonnées cartésiennes). Si un pixel est situé entre plusieurs régions homogènes différentes, la fonction  $s(\theta)$  contiendra plusieurs zones *plates* i.e. la dérivée sera nulle (comme le point 1 de la figure 2.59(a) ou illustré aussi en figure 2.61 pour une représentation cartésienne). Dans le cas où le pixel est situé entre une région homogène et une région texturée,

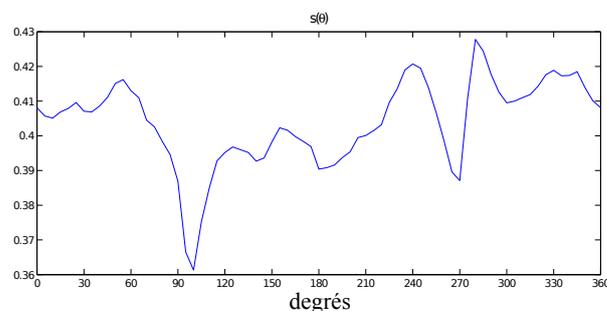


FIGURE 2.60 – Représentation cartésienne de  $s(\theta)$  au point 8 correspondant à un pixel de région texturée.

la fonction  $s(\theta)$  contiendra une seule région plate, (comme le point 7 de la figure 2.59). Cependant, si la région contient une texture fine,  $s(\theta)$  sera alors une fonction constante à cause du filtre de lissage  $G_{(\mu,\lambda)}(x, y, \theta)$  et la région sera donc considérée comme une région homogène. Nous sommes alors capables de classifier un pixel situé entre deux régions homogènes ou texturées, comme les points 3 et 5 de la figure 2.59.

### Détection de zones plates dans $s(\theta)$

L'idée principale pour analyser un signal polaire est de détecter les zones plates significatives, qui correspondent à des régions homogènes ou de textures très fines de l'image. La figure 2.61 (a) montre le signal  $s(\theta)$  extrait en au point 1 de la figure 2.59 (a). Ce point particulier est situé à la limite de deux régions. Après lissage la dérivée  $s_\theta(\theta)$  est calculée et représentée sur la figure 2.61 (b). A partir de  $s_\theta(\theta)$ , les zones plates sont détectées comme des intervalles (i.e. secteurs angulaires), avec une dérivée nulle, i.e. un ensemble de valeurs excédant un seuil en amplitude donné  $s_{th}$ . Ce seuil  $s_{th}$  est représenté par les lignes pointillées horizontales sur la figure 2.61 (b). La direction médiane, notée  $\eta$ , correspond à la direction orthogonale au contour. Nous notons ces intervalles  $\alpha_i$  avec  $\alpha_1$  le plus grand secteur angulaire,  $\alpha_2$  le second etc. Nous considérons que nous détectons une zone plate dans  $s(\theta)$  lorsque

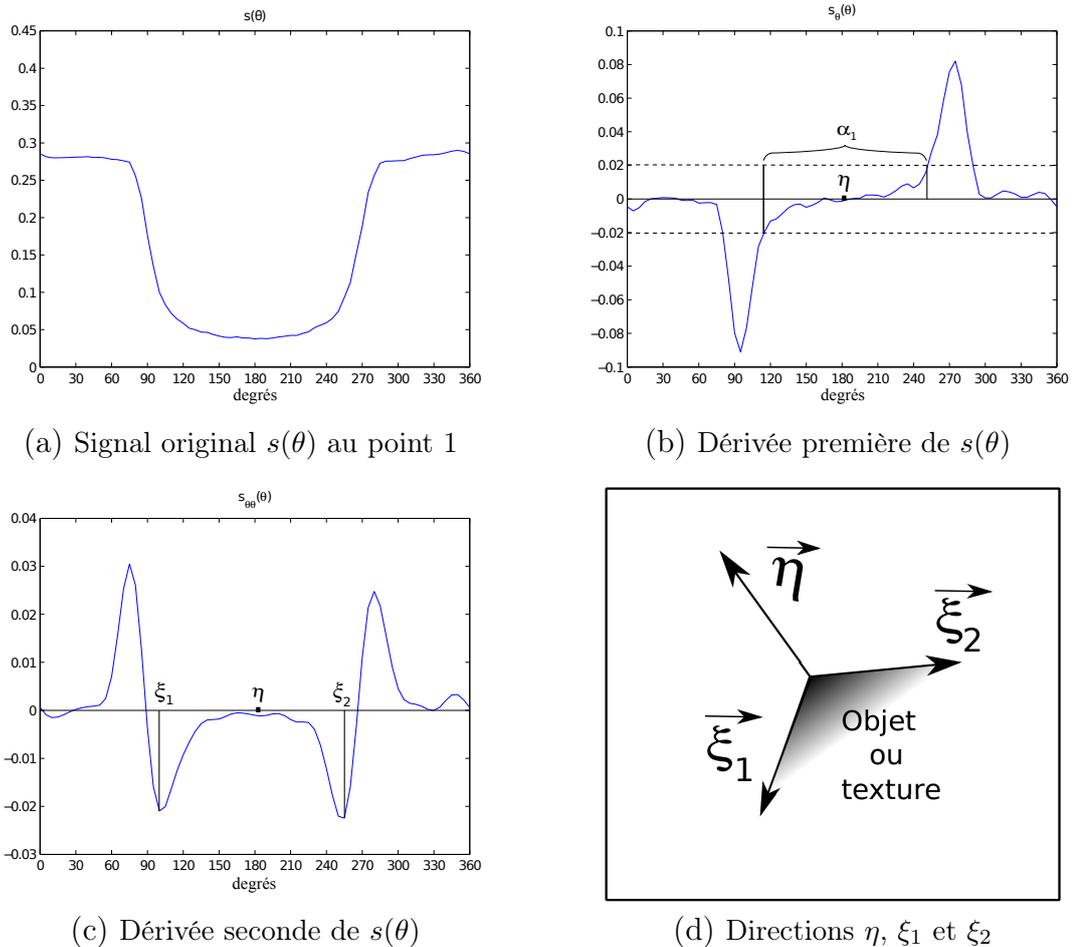


FIGURE 2.61 – Détection de zones plates sur  $s(\theta)$ .

$30^\circ < \alpha_1 < 360^\circ$ . Lorsque  $\alpha_1 = 360^\circ$ , le pixel appartient à une zone homogène ou de texture très fine. Si  $30^\circ > \alpha_1$ , le pixel est soit dans de la texture, soit dans proche d'un contour mais pas suffisamment afin d'être considéré dans une zone de contour.



FIGURE 2.62 – Régions où  $s(\theta)$  contient des zones plates.  $\mu = 10$ ,  $\lambda = 1$  et  $\Delta\theta = 5^\circ$ .

Les régions noires dans la figure 2.62 montrent où les zones plates ont été détectées dans les signaux  $s(\theta)$  de l'image présentée en figure 2.58(a) et correspondent à une détection de contours larges. Donc cette image sera lissée anisotropiquement dans les régions noires de la figure 2.62 et isotropiquement dans les régions blanches.

A partir de la dérivée seconde  $s_{\theta\theta}(\theta)$ , nous pouvons extraire les directions  $\xi_1$  et  $\xi_2$  qui délimitent les zones plates détectées. Les orientations  $\xi_{i, i \in \{1,2\}}$  sont calculées comme les directions de la courbure maximale ou minimale (figure 2.61 (c)). Comme illustré en figure 2.61 (d), les orientations  $\xi_{i, i \in \{1,2\}}$  représentent les deux directions d'un contour traversant le pixel considéré (une direction entrante et une direction sortante). Ces directions vont être utilisées dans un schéma de diffusion anisotropique qui sera présenté dans la partie suivante. Cependant, nous avons uniquement utilisé la dérivée sur signal  $s_\theta(\theta)$  pour améliorer le schéma de diffusion de la partie 4.4 ainsi que pour débruiter des images (partie 4.5).

### 4.3 Diffusion de l'image dans les deux directions correspondantes aux courbures du signal $s(\theta)$

#### Un nouveau schéma de diffusion dans deux directions

Comme introduit au début de cette partie, nous souhaitons créer un processus de lissage capable de supprimer la texture, préserver les contours et lisser les régions homogènes. Similairement à de nombreux schémas de restaurations qui peuvent être interprétés d'un point de vue géométrique, nous allons définir un processus de diffusion qui tiendra compte de la classification de pixel établie dans la partie précédente.

Ainsi, nous souhaitons lisser isotropiquement à l'intérieur des régions homogènes et texturées alors que nous désirons diffuser anisotropiquement sur et proche des contours. Afin d'éviter de "rogner" les coins comme de nombreux schémas de diffusions présentées dans la chapitre 1, nous souhaitons minimiser cet effet en considérant les deux directions  $\xi_1$  et  $\xi_2$  provenant de notre processus de classification de pixels de contours.

Ce nouveau schéma de diffusion peut maintenant être décrit par l'EDP suivante :

$$\begin{cases} I_t &= F(I_0)\Delta I_t + (1 - F(I_0))\frac{\partial^2 I_t}{\partial \xi_1 \partial \xi_2} \\ I(0) &= I_0 \end{cases} \quad (2.28)$$

où :

- $t$  représente le temps de diffusion,
- $I(0)$  l'image originale,
- $I_t$  l'image diffusée au temps  $t$ ,
- $F(I_0)$  la fonction de contrôle ( $F(I_0) = 1$  lorsqu'il s'agit d'une région homogène ou texturée,  $F(I_0) = 0$  dans une région de contours).

L'implémentation de ce schéma de diffusion est discutée en partie 4.6.

### Classification de zones plates et schéma de diffusion pour les images vectorielles

TABLE 2.2 – Classification de zones plates  $\alpha_{j,\{1 \leq i \leq d\}}^i$ .

Taille de $\alpha_j^i$	$v^i$	type
$\frac{4}{3}\pi < \alpha_1^i < \frac{3}{2}\pi$	2	angle aigu / coin
$\pi < \alpha_1^i < \frac{4}{3}\pi$	3	angle obtus
$\frac{2}{3}\pi < \alpha_1^i < \pi$	4	bord de ligne
$\frac{2}{3}\pi < \alpha_1^i$ et $\frac{2}{3}\pi < \alpha_2^i$	5	contour
$\frac{1}{6}\pi < \alpha_1^i < \frac{2}{3}\pi$	1	coin
autre	0	autre

Dans cette partie, nous considérons  $I^i$  le  $i^{\text{ème}}$  canal d'une image couleur  $\mathbf{I}$  composée de  $d$  canaux ( $1 \leq i \leq d$ ). Les images couleurs que nous avons utilisées possèdent trois canaux ( $i = 3$ ) : rouge, vert et bleu notés respectivement  $R$ ,  $V$  et  $B$ .

Notre objectif est d'estimer  $\xi_1^{\text{couleur}}$  et  $\xi_2^{\text{couleur}}$  de chaque canal de couleur  $I^i$  dans un premier temps (un exemple de chaque canal est présenté en figure 2.64). Dans un

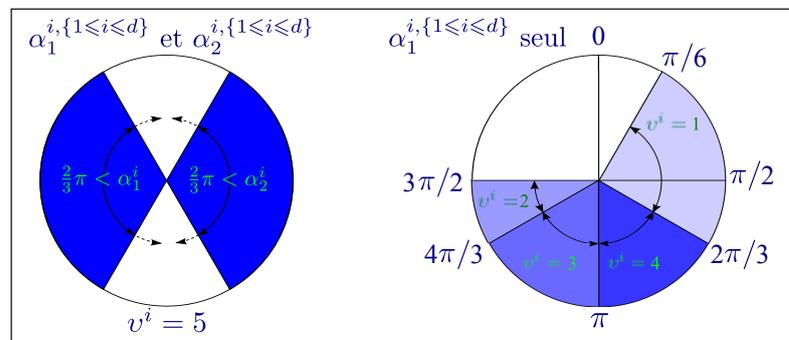


FIGURE 2.63 – Représentation colorimétrique de la classification des zones plates  $\alpha_{j,\{1 \leq i \leq d\}}^i$ . Plus le score de  $v^i$  est important, plus la couleur est sombre.

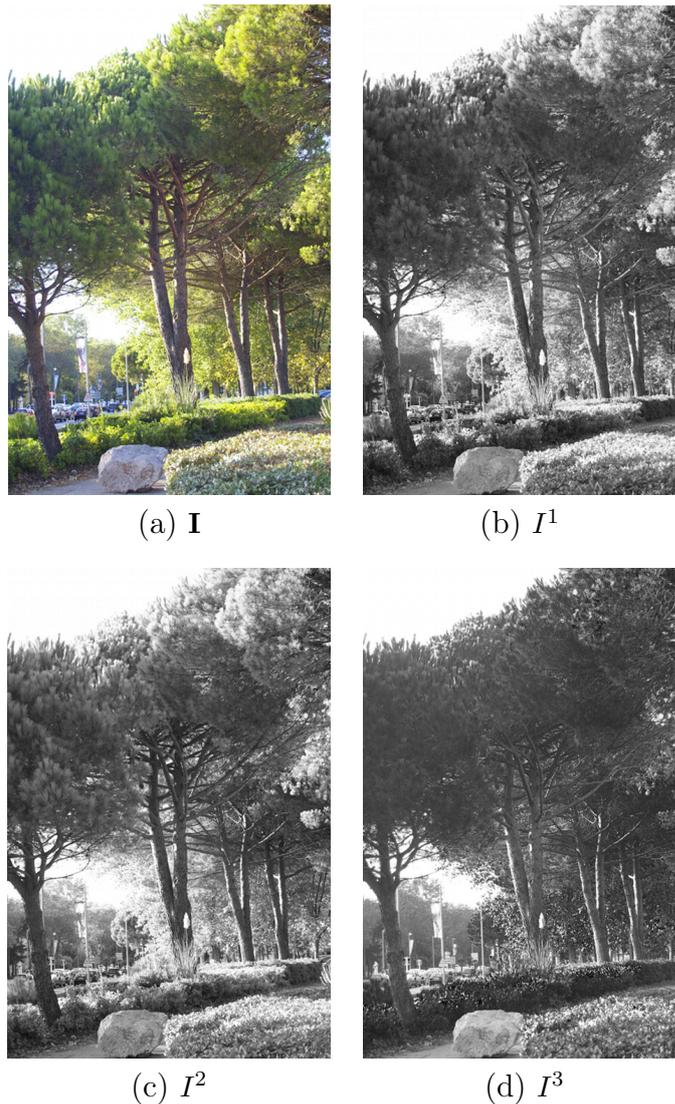


FIGURE 2.64 – Image originale couleur et ses trois composantes  $R$ ,  $V$  et  $B$ .

second temps, nous souhaitons lisser chaque  $I^i$  en utilisant notre schéma de diffusion présenté dans la partie précédente avec  $\xi_1 = \xi_1^{couleur}$  et  $\xi_2 = \xi_2^{couleur}$ .

Premièrement, nous appliquons notre détection de contours larges présentée en partie 4.2 de ce chapitre pour chaque composante  $I^i$  de  $I$ . Ainsi, nous calculons les plus grandes zones plates notées respectivement  $\alpha_1^i$  pour la plus grande et  $\alpha_2^i$  pour la seconde plus grande. Deuxièmement, nous attribuons un score  $v^i$  dépendant de  $\alpha_1^i$  et  $\alpha_2^i$ . Cette classification est présentée dans le tableau 2.2 et représentée sous forme de diagramme circulaire en figure 2.63.

Les pixels situés près des contours obtiendront un score fort tandis qu'à une certaine distance des bords, ils recevront un score plus faible. De plus, les angles obtus  $\alpha_1^i$  recevront un score plus fort que les angles aigus qui peuvent être causés par la texture ou le bruit.

Nous présentons également cette classification sur les figures 2.64 (b), (c) et (d) dans les figure 2.65 (a), (b) et (c) sur des images scalaires (la valeur 0 de notre classification correspond à la couleur noire et la valeur 5 à la blanche).

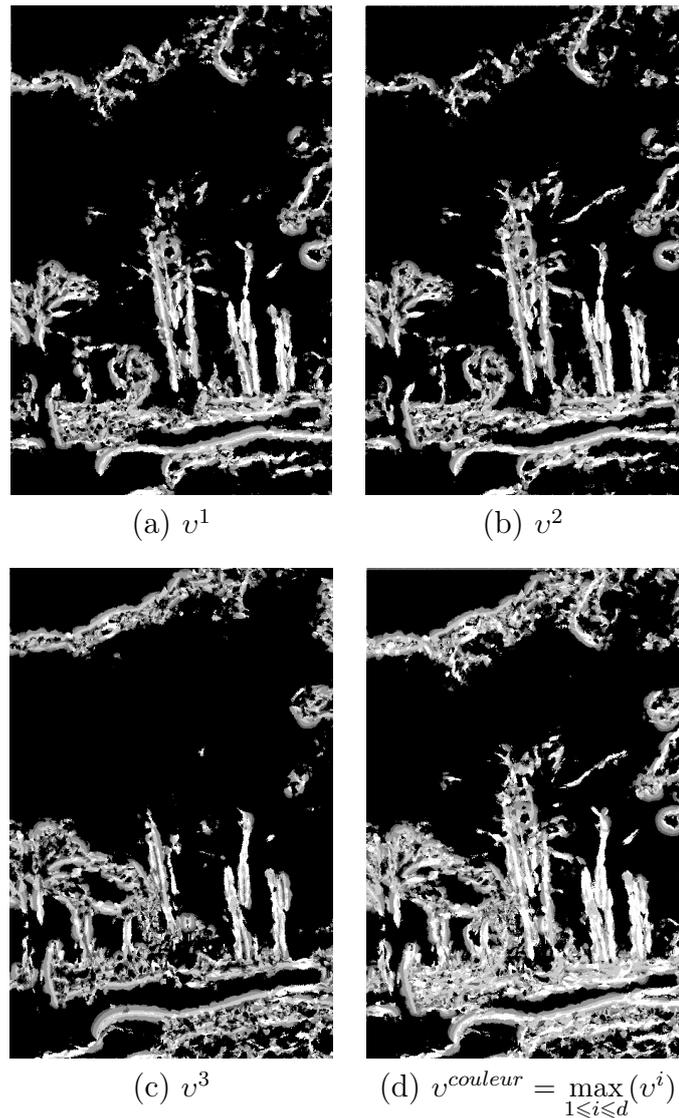


FIGURE 2.65 – Détection des zones plates avec  $v^i$ ,  $i = \{1, 2, 3\}$ .

Troisièmement, à partir du maximum des  $v^i$  sont calculées les directions  $\xi_1^{couleur}$  et  $\xi_2^{couleur}$  suivant l'équation :

$$\begin{cases} \xi_1^{couleur} = \arg \max_{\xi_1^i, 1 \leq i \leq d} (v^i), \\ \xi_2^{couleur} = \arg \max_{\xi_2^i, 1 \leq i \leq d} (v^i). \end{cases} \quad (2.29)$$

La figure 2.65 (d) montre en niveaux de gris le résultat de

$$v^{couleur} = \max_{1 \leq i \leq d} (v^i).$$

Contrairement aux méthodes de diffusion présentées au chapitre 1, partie 8.8 et 8.9, nous n'utilisons ni une norme  $\mathcal{N}(\mathbf{I})$ , ni un tenseur multi spectral [DZ86] dans notre schéma de diffusion pour les images vectorielles. Au contraire, nous diffusons

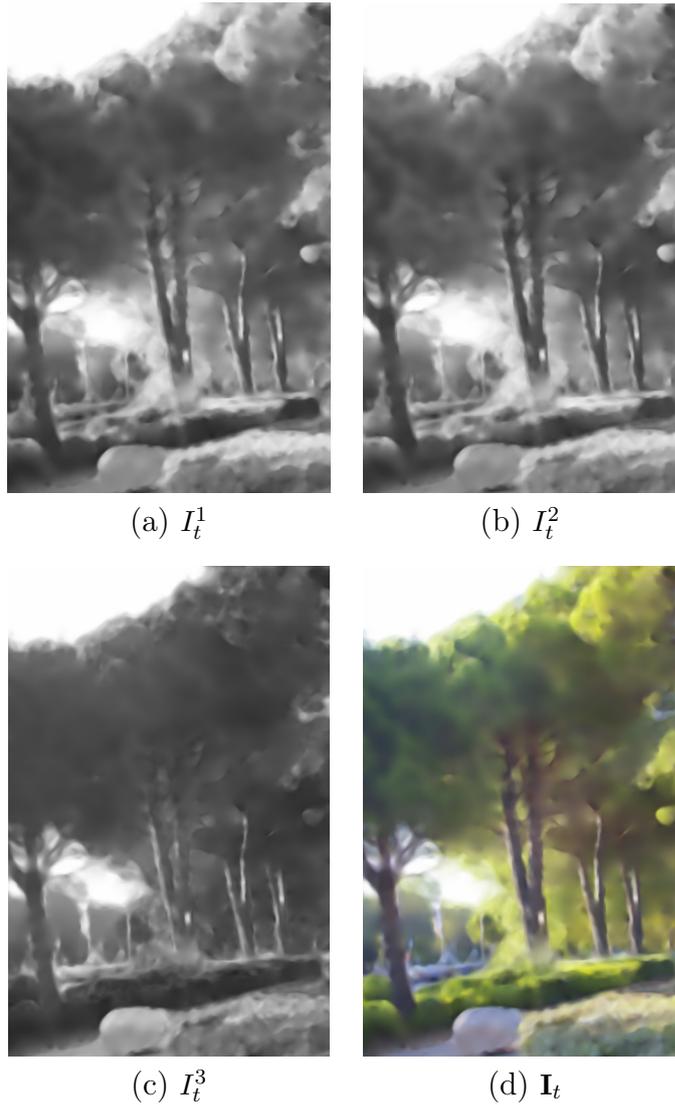


FIGURE 2.66 – Diffusion de chaque composante  $I^i$  de  $\mathbf{I}$  et fusion des résultats après 100 itérations dans  $\mathbf{I}_t$ .

chaque canal de l'image  $I^i$  en utilisant notre schéma de diffusion décrit précédemment avec l'équation suivante :

$$\begin{cases} I_t^i &= F(\mathbf{I}_0)\Delta I_t^i + (1 - F(\mathbf{I}_0))\frac{\partial^2 I_t^i}{\partial \xi_1^{couleur} \partial \xi_2^{couleur}} \\ \mathbf{I}(0) &= \mathbf{I}_0 \end{cases} \quad (2.30)$$

où  $\mathbf{I}(0)$  représente l'image couleur initiale.

Finalement, à partir de chaque  $I_t^i$  (figure 2.66 (a), (b) and (c)), nous pouvons synthétiser une image couleur  $\mathbf{I}_t$  (figure 2.66 (d)), ce qui revient au schéma de diffusion suivant :

$$\begin{cases} \mathbf{I}_t &= F(\mathbf{I}_0)\Delta \mathbf{I}_t + (1 - F(\mathbf{I}_0))\frac{\partial^2 \mathbf{I}_t}{\partial \xi_1^{couleur} \partial \xi_2^{couleur}} \\ \mathbf{I}(0) &= \mathbf{I}_0 \end{cases} \quad (2.31)$$

## Résultats

Nous présentons des résultats de notre schéma de diffusion sur des images réelles en niveaux de gris et en couleur ainsi que des résultats de différentes autres approches. La première image de taille  $1000 \times 1000$  est une photographie aérienne de vignes. L'objectif est de séparer les vignes des routes. Nous avons utilisé notre détecteur avec  $\mu = 5$  et  $\lambda = 1$  et un angle discrétisé  $\Delta\theta = 5^\circ$ .

Le résultat obtenu complète l'objectif fixé puisque les vignes sont diffusées et les routes ne sont pas perdues. Parmi les résultats présentés, notons que le filtre bilatéral [TM98] n'est pas adapté pour diffuser les textures puisque certaines parcelles ne sont pas du tout diffusées. Si les paramètres du filtre changent afin d'obtenir le résultat désiré, les routes seront diffusées. Il en va de même pour le filtre de Perona-Malik [PM90]. L'approche d'Alvarez et al. [ALM92] donne un meilleur résultat, les routes sont toujours visibles, cependant certaines parcelles ont fusionné. Les approches par masques de Kuwahara [KHEK76] et Nagao [NM79] permettent de supprimer la majorité des stries présentes dans les textures de vignes et de conserver les routes. Cependant, ces approches ne sont pas adaptées pour régulariser complètement l'image tout en gardant les principales structures de celle-ci. L'approche *MCM* [CLMC92] régularise l'image mais arrondit les coins. La diffusion tensorielle de Weickert [Wei] arrondit également les coins tout en apportant un aspect fibreux au niveau des stries formées par les vignes. L'approche tensorielle de Tschumperlé et Deriche [TD05] n'est pas adaptée non plus puisqu'elle rend flou certaines parties de l'image, moins d'itérations provoquerait le même type de résultat que la diffusion tensorielle de Weickert. Le résultat de Tschumperlé préservant la courbure [Tsc06] est très satisfaisant, cependant notre schéma en figure 2.67 (l) diffuse mieux certaines parcelles.

L'objectif dans la figure 2.68(a) est de détecter le chemin tout en diffusant les autres régions. Après 500 itérations,  $\mu = 5$ ,  $\lambda = 1.5$ ,  $s_{th} = 0.2$  et  $\Delta\theta = 5^\circ$ , notre méthode a diffusé les arbres et les herbes. La figure 2.68(k) illustre les textures qui ont été diffusées avec notre méthode, ainsi nous pouvons remarquer les différentes branches, pierres et feuilles qui ont été supprimées par notre méthode.

Nous avons comparé notre résultat avec ceux des différentes approches de la *MCM* [CLMC92], du filtre bilatéral [TM98], d'un filtre de textures [BLMV10] et de deux méthodes tensorielles [TD05] [Tsc06]. La *MCM* et le filtre bilatéral ne sont pas du tout adaptés à ce type d'image puisque certains détails dans les textures ressortent. Les méthodes tensorielles ne sont pas adaptées non plus puisque certains détails dans la texture sont toujours visibles tout en rendant flou la droite du chemin. Un trop grand nombre d'itérations pour ces deux approches fusionnera les herbes à la droite de l'image avec le chemin. Aussi, en augmentant la valeur de l'écart type de la gaussienne  $\sigma$ , les détails à l'intérieur de la texture seront lissés mais le chemin sera diffusé avec le reste de l'image également.

La comparaison de notre algorithme avec le filtre de texture est intéressante [BLMV10]. Une valeur de  $\sigma = 6.5$  est nécessaire pour supprimer la texture mais comme indiquée en introduction de cette section, les bords des objets sont supprimés et ici le but est de séparer le chemin des plantes. Cette approche n'est donc pas adaptée à ce type d'images.

La détection de contours sur l'image diffusée (figure 2.69) permet de reconnaître le chemin contrairement à la détection de contours sur l'image originale avec le même

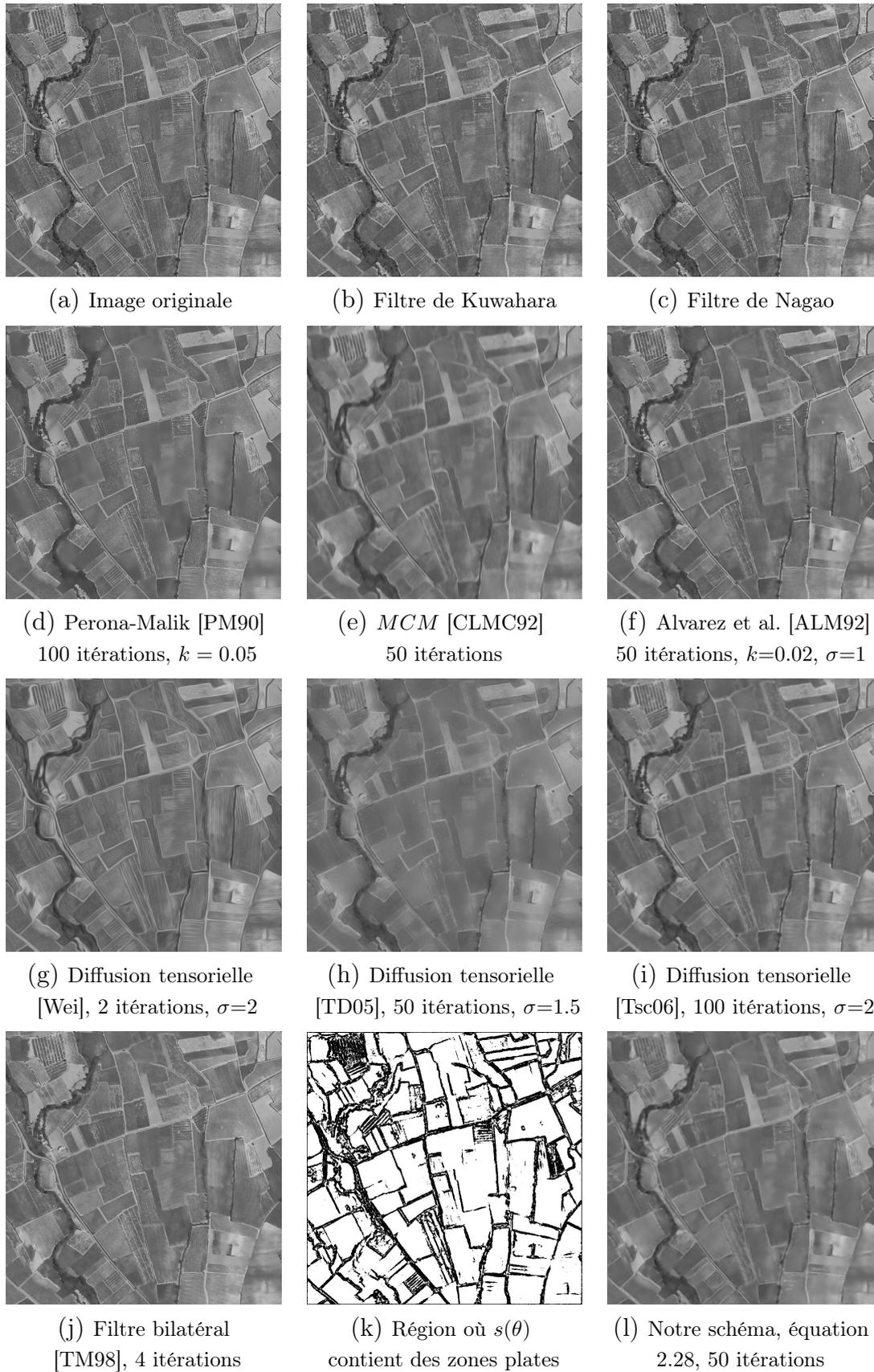


FIGURE 2.67 – Diffusion de texture de vignes dans une image aérienne.

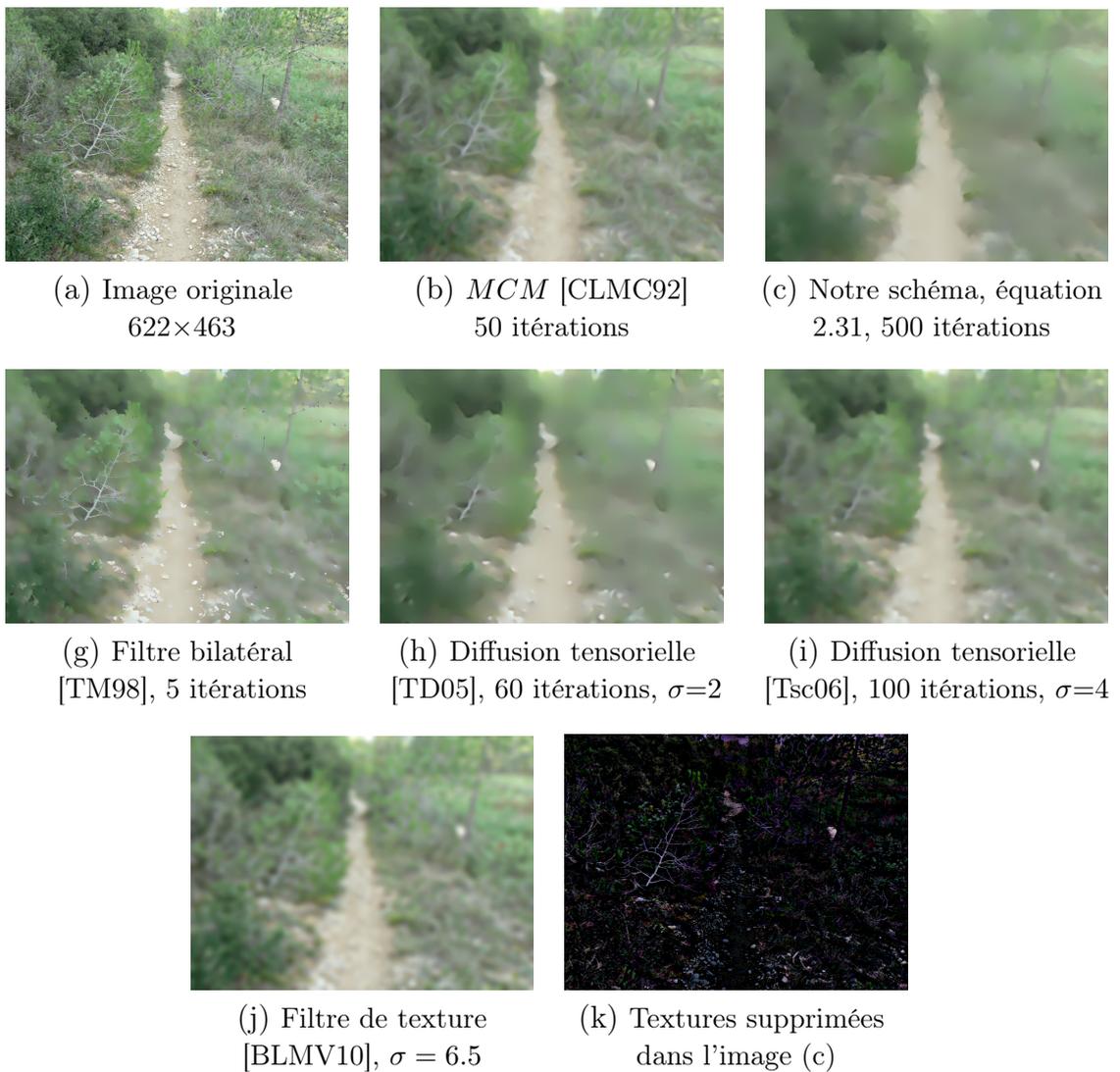


FIGURE 2.68 – Diffusion de texture dans une image réelle vectorielle.

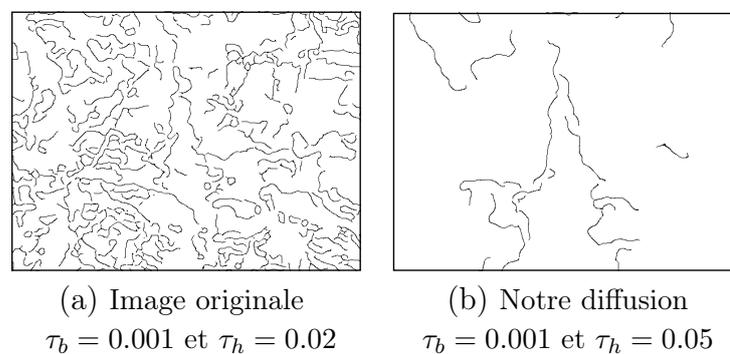


FIGURE 2.69 – Détection de contours [Der92] sur l'image diffusée,  $\sigma = 4$ .

paramètre écart type  $\sigma = 4$ .

## 4.4 Diffusion dans les deux directions des filtres de dérivation

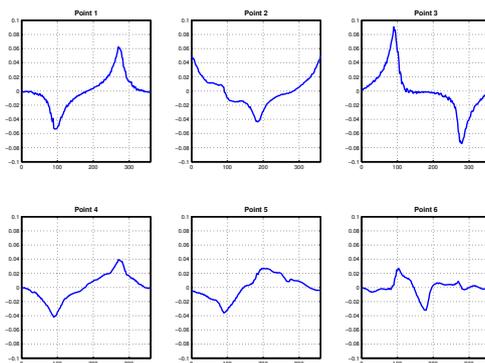
### Une fusion d'informations issues du filtre de lissage et du filtre de dérivation

Dans cette partie, nous présentons une méthode améliorée de la partie précédente pour supprimer la texture. En effet, la nouveauté de cette approche réside dans le mixage du calcul des zones plates du signal  $s(\theta)$  pour les pixels appartenant à des contours ou proche des bords et de la diffusion anisotropique suivant les deux directions du détecteur de contour anisotrope dans ces mêmes régions.

L'utilisation de la courbure du signal  $s(\theta)$  dans la partie précédente afin de déterminer deux directions de diffusion  $\theta_1$  et  $\theta_2$  n'était pas très précise et ne permet pas la conservation dans les images de petits objets. En effet, afin d'estimer ces deux orientations  $(\theta_1, \theta_2)$ , la dérivée seconde  $s_{\theta\theta}(\theta)$  est lissée plusieurs fois afin de supprimer un maximum de bruit et de fausses alertes. Le détecteur de contours développé dans la partie 2.2 de ce chapitre élimine ce problème. Une fois les régions détectées où  $s(\theta)$  contient des zones plates (figure 2.62), les directions de diffusions  $(\theta_1, \theta_2)$  sont estimées par les équations 2.7 et 2.8.



(a) Sélection de points (en vert)



(b) Fonctions  $\mathcal{Q}(x, y, \theta)$  associées

FIGURE 2.70 – Sélection de points et leurs fonctions  $\mathcal{Q}(x, y, \theta)$  associées,  $\mu = 10$ ,  $\lambda = 1$  et  $\Delta\theta = 2^\circ$ . Pour (b), l'axe des X représente l'orientation du filtre  $\mathcal{Q}(x, y, \theta)$  en fonction de  $\theta$  et l'axe des Y représente la valeur de  $\mathcal{Q}(x, y, \theta)$ .

Le schéma de diffusion devient alors :

$$\begin{cases} I_t &= F(I_0)\Delta I_t + (1 - F(I_0)) \cdot \frac{\partial^2 I_t}{\partial\theta_1\partial\theta_2} \\ I(0) &= I_0 \end{cases} \quad (2.32)$$

où :

- $t$  représente le temps de diffusion,
- $I(0)$  l'image originale,
- $I_t$  l'image diffusée au temps  $t$ ,
- $F(I_0)$  la fonction de contrôle  $F(I_0) = 1$  lorsqu'il s'agit d'une région homogène ou texturée,  $F(I_0) = 0$  dans une région de contours).

Résultats

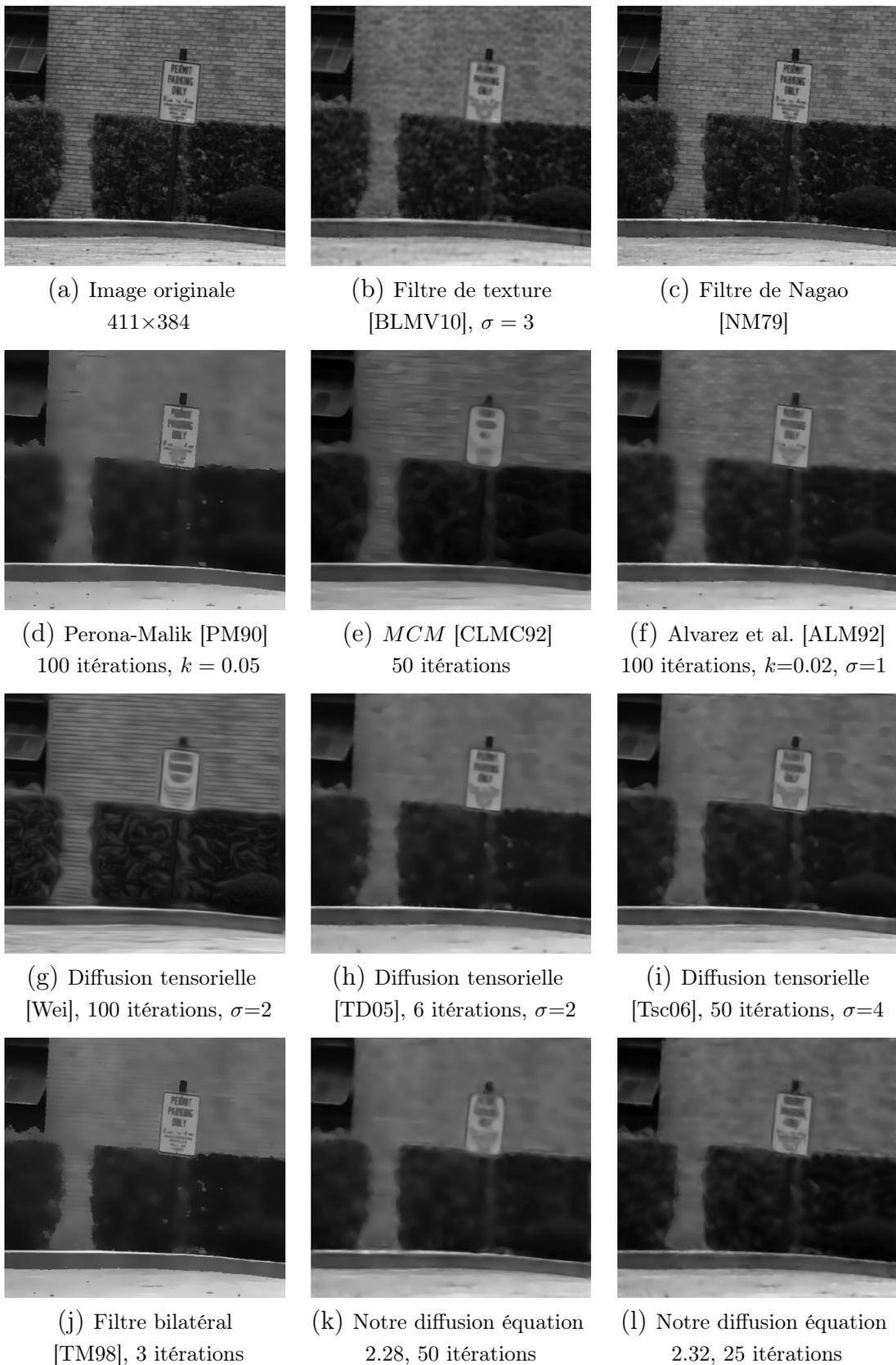


FIGURE 2.71 – Diffusion de textures dans une image naturelle.

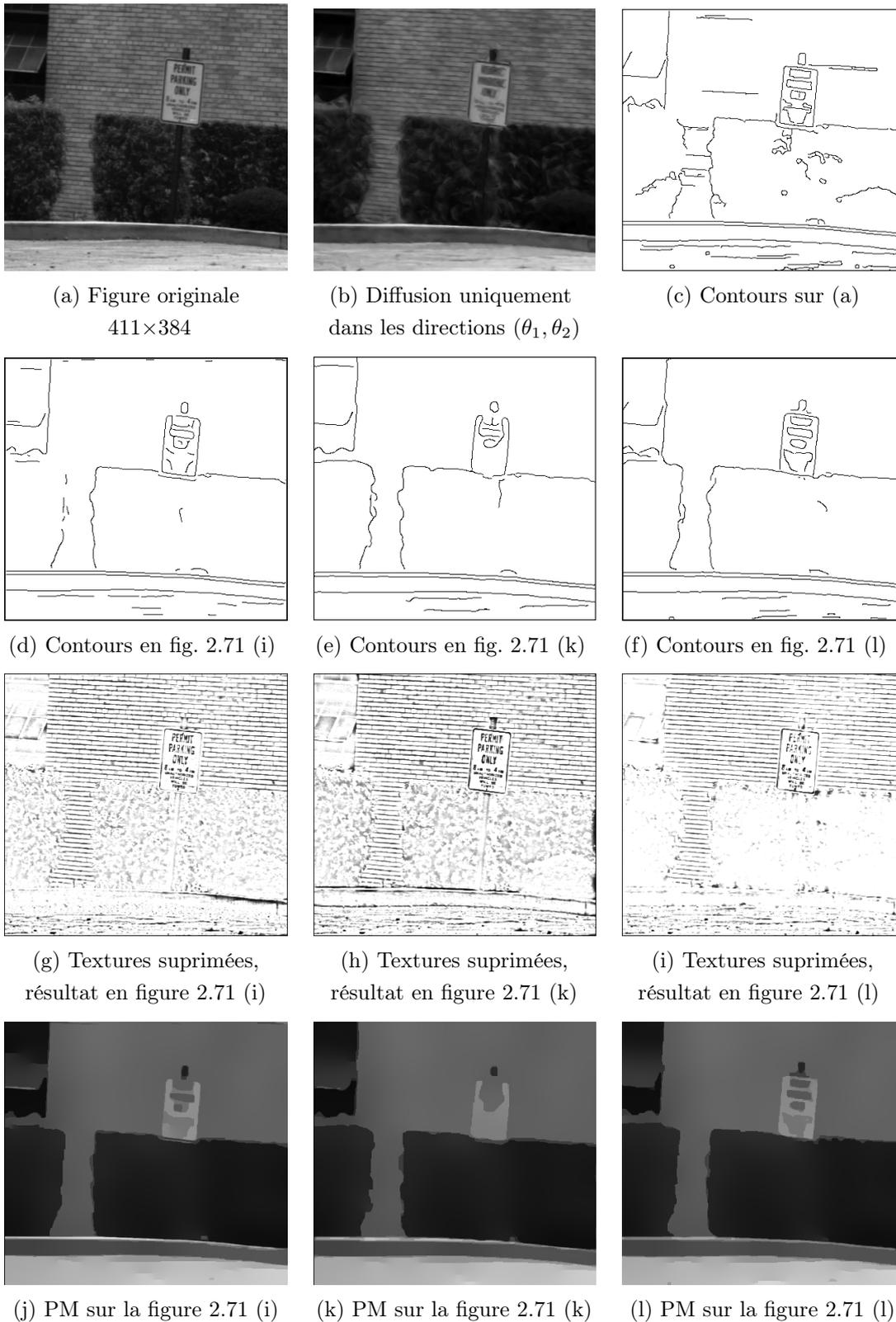


FIGURE 2.72 – Textures supprimées, détection de contours [Der92] sur nos résultats afin de montrer la stabilité de notre diffusion ( $\sigma = 2$ ) et application du modèle de Perona-Malik (PM) [PM90], 1000 itérations.

Dans l'image présentée en figure 2.71 (a), le but est de lisser les différentes

textures présentes dans l'image (les buissons, les briques à l'intérieur du mur) tout en préservant les objets (la fenêtre, le panneau, le trottoir, les contours des buissons). Nous utilisons notre détecteur avec  $\mu = 10$ ,  $\lambda = 1$  et  $\Delta\theta = 5^\circ$  pour la détection de zones plates (illustré en figure 2.62). Les paramètres utilisés pour notre détecteur anisotrope de contours afin d'estimer  $\theta_1$  et  $\theta_2$  sont  $\mu = 5$ ,  $\lambda = 1.5$  and  $\Delta\theta = 2^\circ$ . Notre résultat est présenté en figure 2.71 (l) après 25 itérations. Les différentes textures sont ainsi diffusées alors que les objets et leurs coins sont préservés.

Nous avons comparé notre résultat avec les méthodes de Nagao [NM79], Perona-Malik [PM90], Alvarez et al. [ALM92], Buades et al. [BLMV10], les diffusions tensorielles de Weickert [Wei], Tschumperlé et Deriche [TD05] et Tschumperlé [Tsc06] ainsi que le filtre bilatéral [TM98], le schéma *MCM* [CLMC92] et notre précédente approche de l'équation 2.28.

Toutes ces méthodes n'ont pas complètement diffusé la texture du mur (figures 2.71 (b), (d), (e), (f) et (j)) et les bords du buisson (figures 2.71 (d), (f), (h) et (i)) ou les coins du panneau ne sont pas correctement préservés (figure 2.71 (k)). Les approche de Merona-Malik [PM90] et d'Alvarez et al. [ALM92] ne diffusent pas totalement la texture alors qu'il existe des fuites au niveau du buisson. Le filtre de Nagao [NM79] préserve les bords des objets mais également ceux de la texture. La schéma *MCM* diffuse la texture du buisson mais les briques ne deviennent pas un région homogène, de plus les objets tels que le panneau sont arrondis. Le filtre bilatéral [TM98] n'est pas adapté pour supprimer les textures dans cette image puisque les bords des buissons sont perdus alors que les briques sont toujours visibles. En général, le filtre bilatéral n'est pas adapté pour supprimer les textures lorsqu'elles sont trop marquées. Le filtre de texture de Buades et al. [BLMV10] ne permet pas de supprimer ce type de texture. Les bords du panneau sont diffus avec le mur alors que les différentes textures du mur et des buissons ne sont pas supprimés. Une valeur du paramètre  $\sigma$  plus grande supprimera totalement le panneau. L'approche tensorielle de Weickert (figure 2.71 (g)) n'est pas adaptée pour la suppression de texture dans les images puisque les structures du mur et des feuilles du buisson sont prolongées. La méthode de Tschumperlé et Deriche [TD05] apporte un résultat intéressant même si la texture du buisson n'est pas entièrement diffusée. Cependant, ce schéma de diffusion perd les contours et fusionne le bord du buisson de gauche dans l'image avec le mur. Aussi, le bord gauche du panneau devient flou. L'approche tensorielle préservant la courbure de Tschumperlé [Tsc06] avec un écart type plus important diffuse entièrement les textures, comme illustré en figure 2.71 (i). Toutefois les bords du buisson à gauche de l'image commencent à être flous.

L'amélioration apportée par rapport à notre approche précédente est que les contours entre les différentes textures sont mieux préservés voire amplifiés. En effet, par rapport à la figure 2.71 (k), les contours des buissons et du panneau en figure 2.71 (l) sont beaucoup plus tranchants. Egalement au niveau du mur de briques, à gauche de l'image près de la fenêtre, le bord vertical reste plus droit et le coin formé avec la fenêtre est mieux préservé.

La figure 2.72 (b) illustre l'intérêt de détecter les zones plates de  $s(\theta)$ . Lorsque la diffusion s'effectue dans les deux directions  $(\theta_1, \theta_2)$  sans aucune condition, les textures ne sont donc pas supprimées. Même avec en considérant l'amplitude du gradient ne suffirait pas. En effet, ce schéma serait proche de celui de Alvarez et al. [ALM92], qui ne diffuse que dans le sens des contours lorsque le gradient est

fort et isotropiquement lorsqu'il est faible. Cette condition ne permet pas de diffuser la texture comme souhaité et de plus perd les bords de certains objets puisque le gradient estimé à l'intérieur de la texture peut être important. En revanche, cette condition sur l'intensité du gradient pourrait être utilisée en restauration des images (voir partie suivante).

Les figures 2.72 (g), (h) et (i) illustrent les textures supprimées avec respectivement notre précédente méthode, notre nouvelle approche et celle de Tschumperlé [Tsc06]. Les bords des buissons, de la fenêtre et du trottoir n'ont pas été supprimés. Le contour du panneau est mieux préservé que la précédente méthode et les textures sont entièrement supprimées.

Afin de montrer l'efficacité de notre approche pour la diffusion de texture, nous comparons dans la figure 2.72 les contours détectés dans l'image originale et l'image obtenue après diffusion résultante de trois méthodes. La détection de contours après la diffusion est moins bruitée que l'originale, ce qui vérifie notre objectif initial. En effet, les bords détectés des différents objets sur notre résultat en figure 2.72 (f) apparaissent clairement. L'approche de Tschumperlé [Tsc06] rend flou le bord du buisson de gauche, tandis que notre première approche ne détectait pas le haut du panneau. La surface image présentée en figure 2.73 met en évidence également la préservation des contours ainsi que l'efficacité du lissage des textures.

Enfin, nous avons souhaité montrer la stabilité de notre approche en appliquant le schéma de Perona-Malik [PM90] sur notre résultat ainsi que notre précédente approche et le résultat de l'approche de Tschumperlé [Tsc06]. La grosse différence se situe au niveau du haut du panneau. En effet, le schéma de Perona-Malik [PM90] permet de segmenter le résultat de notre méthode (figure 2.72 (h)) et le panneau reste rectangulaire après 1000 itérations, donc la diffusion est considérée comme stable. Notre précédente approche ne permet pas de conserver le haut du panneau, tout comme l'approche de Tschumperlé. Le nouveau schéma de diffusion en équation

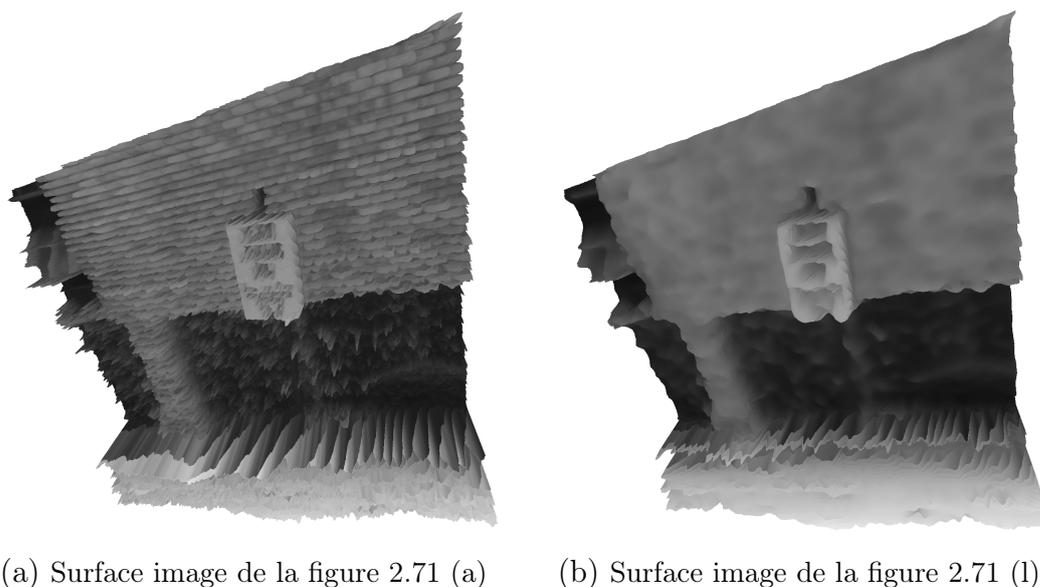


FIGURE 2.73 – Comparaison de la surface image originale avec celle de l'image où la texture a été diffusée.

2.32 est bien une amélioration de l'approche en équation 2.28.

La texture représente des zones irrégulières dans l'image, plus la texture est grossière, plus la zone est irrégulière. C'est pourquoi nous avons utilisé un filtre de lissage plus long que celui de contours ( $\mu = 10$  pour le lissage et  $\mu = 5$  pour les contours). Cette longueur associée au filtre de contours permet d'estimer des directions de diffusion même pour les petits objets dans les zones correspondantes aux contours détectés par le filtre de lissage.

Comme ce schéma de diffusion semble stable et robuste à la texture, nous souhaitons adapter notre approche pour restaurer des images fortement bruitées.

#### 4.5 Diffusion dans les deux directions des filtres de dérivation avec contrôle de la diffusion

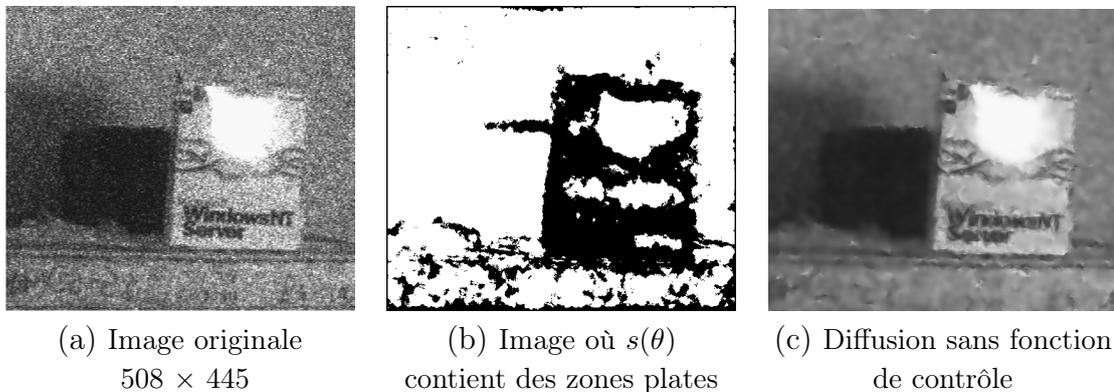


FIGURE 2.74 – Détection des zones plates de  $s(\theta)$  représentées en noir dans (b) et diffusion dans les directions  $(\theta_1, \theta_2)$  à l'intérieur des zones noires de (b).

Dans cette partie, nous présentons une nouvelle méthode de régularisation des images utilisant à la fois des demi-filtres orientés de lissage et des demi-filtres orientés de dérivation. La nouveauté de cette approche réside dans le fait d'utiliser à la fois une détection de contours larges et un processus de diffusion contrôlé aux niveaux des coins et des contours dans les zones de contours larges.

De nombreux schémas de régularisation d'images diffusent isotropiquement les zones de l'image dans les régions homogènes et anisotropiquement dans les régions proches des contours afin de préserver les structures importantes ou certains détails [PM90]. La détection de contours est souvent utilisée afin de contrôler le processus de diffusion. En effet, dans [ALM92], les auteurs fournissent des contraintes de diffusion dans les directions du gradient et tangentielles aux contours afin de mieux préserver les bords. L'intention est de restreindre le processus de diffusion seulement dans la direction tangentielle aux bords et de régler la *force* de diffusion par la magnitude du gradient. Cependant, au niveau d'un contour très marqué, la diffusion est inhibée et le contour peut toujours contenir du bruit. De plus, lorsque le bruit est trop fort, il n'est pas diffusé et le processus de diffusion n'apporte pas de zones homogènes. Donc, ces méthodes de diffusion sont peu adaptées pour la préservation d'objets fins dans des images fortement bruitées.

Nous allons présenter une nouvelle approche de restauration d'images capable de lisser les petites structures de l'image dans le sens des contours sans pour autant les diffuser. Contrairement aux méthodes de restauration d'images classiques, la diffusion s'effectue dans deux directions mais aussi, elle est contrôlée par l'angle formée par ces deux directions et la magnitude du gradient. Les approches des parties 4.3 et 4.4 de ce chapitre ne sont pas adaptées pour la restauration des images puisqu'elles ne permettent pas de contrôler le processus de diffusion ni aux niveaux des coins ni des contours (illustré en figure 2.74 (c)). En effet, les méthodes précédentes diffusent avec la même *force* lorsque le pixel appartient à un contour ou se situe à côté du contour.

### Une diffusion contrôlée au niveau des contours et des coins

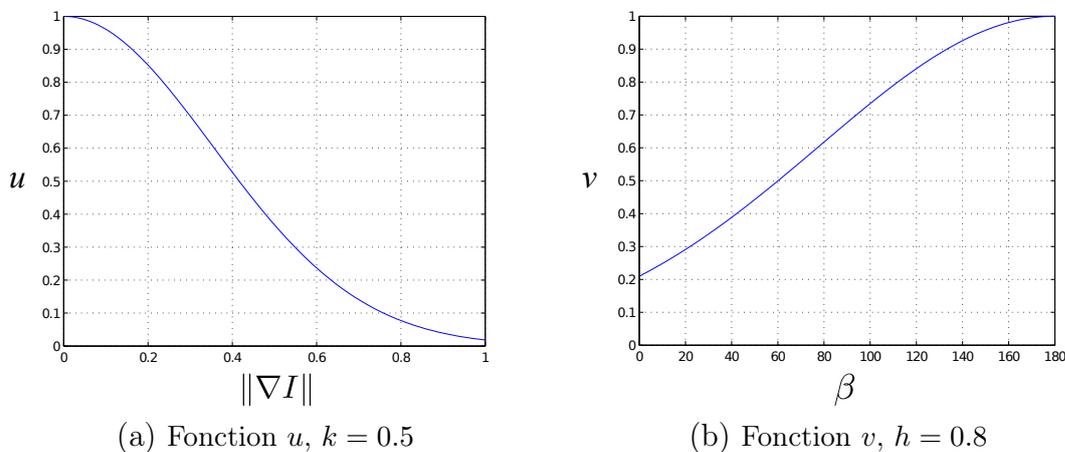


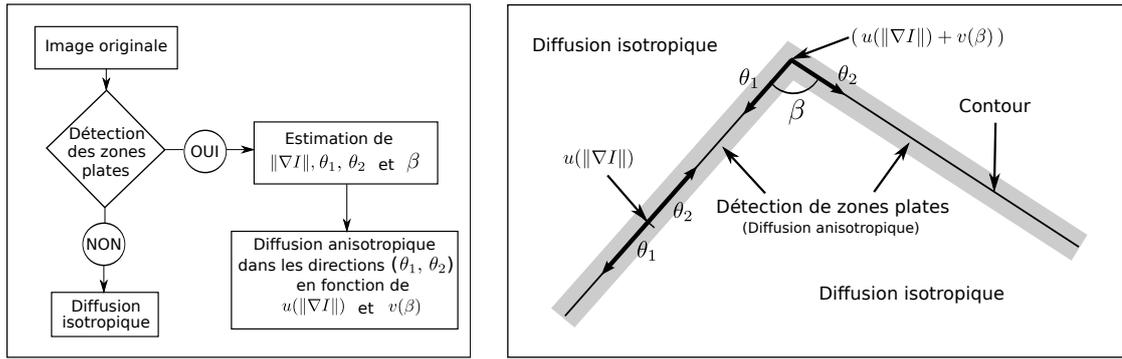
FIGURE 2.75 – Représentation des deux fonctions de contrôle.

Dans certains processus de diffusion, la diffusion s'effectue seulement dans la direction des contours, dirigée par la magnitude du gradient. Concernant notre approche, nous souhaitons diffuser seulement dans les directions  $\theta_1$  et  $\theta_2$  dans les régions classifiées comme pixels de contours seulement (un exemple est disponible en figure 2.74 (b)). Cependant, nous fixons deux contraintes de diffusion. Premièrement, nous contrôlons la diffusion en fonction de la magnitude du gradient estimé  $\|\nabla I\|$  et détaillé dans la partie 2.2 de ce chapitre, équation 2.9. Deuxièmement, la diffusion est également dirigée en fonction de l'angle entre les deux directions de diffusion  $\theta_1$  et  $\theta_2$ .

Afin de contrôler la diffusion en fonction de la magnitude du gradient, nous utilisons la fonction  $u$  suivante :

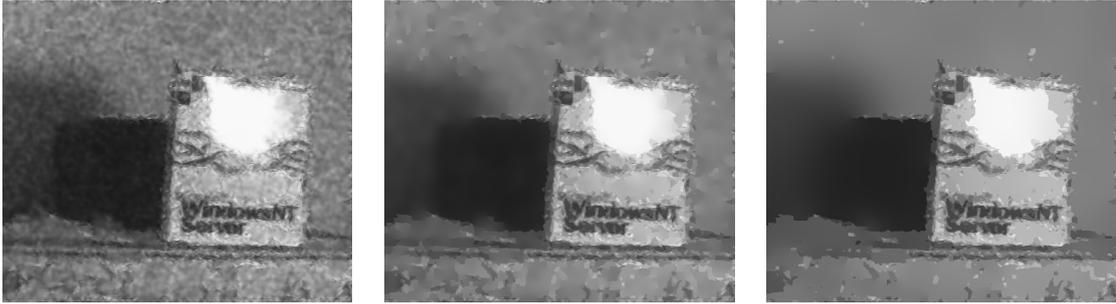
$$u(\|\nabla I\|) = e^{-\left(\frac{\|\nabla I\|}{k}\right)^2}, \quad k \in [0, 1]. \quad (2.33)$$

Utilisant également les deux directions  $\theta_1$  et  $\theta_2$  estimées d'après l'équation 2.9, la diffusion est dirigée par l'angle appelé  $\beta$  entre ces deux orientations (i.e.  $\beta$  est la bissectrice de l'angle formé par  $\theta_1$  et  $\theta_2$ , schématisé en figure 2.76 (b)). En effet, au



(a) Organigramme de notre méthode

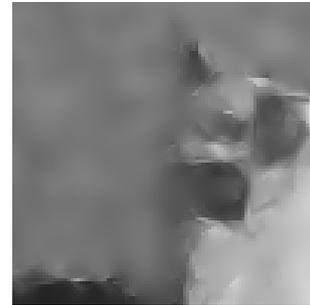
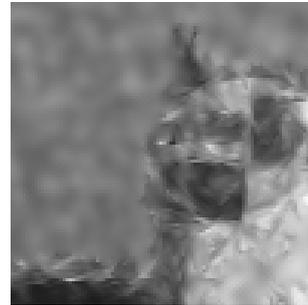
(b) Schéma de notre diffusion



(c) Diffusion anisotropique, équation 2.35

(d) PM sur notre résultat en (c),  $k = 0.02$ , 100 itérations

(e) PM sur notre résultat en (c),  $k = 0.02$ , 1000 itérations



(f) Zoom sur le coin gauche de la boîte avec fonctions de contrôle

(g) Zoom sur le coin gauche de la boîte sans fonction de contrôle

FIGURE 2.76 – Différentes étapes de notre schéma de diffusion.

niveau d'un pixel, plus l'angle  $\beta$  est proche de 0, moins la diffusion sera importante. Dans le cas contraire, plus l'angle  $\beta$  est proche de  $180^\circ$ , plus le lissage sera important. La fonction de contrôle angulaire  $v$  peut alors être définie comme suit :

$$v(\beta) = e^{-\left(\frac{180-\beta}{180-h}\right)^2}, \quad h \in [0, 1] \quad \text{et} \quad \beta = \theta_1 - \theta_2 \quad [180] \text{ (degrés)}. \quad (2.34)$$

figure 2.76, est décrit par l'équation de diffusion suivante :

$$\begin{cases} I_t &= F(I_0)\Delta I_t + (1 - F(I_0)) \cdot \left( \frac{u(\|\nabla I\|) + v(\beta)}{2} \right) \cdot \frac{\partial^2 I_t}{\partial \theta_1 \partial \theta_2} \\ I(0) &= I_0 \end{cases} \quad (2.35)$$

où :

- $t$  représente le temps de diffusion,
- $I(0)$  l'image originale,
- $I_t$  l'image diffusée au temps  $t$ ,
- $F(I_0)$  la fonction de contrôle  $F(I_0) = 1$  lorsqu'il s'agit d'une région homogène,  $F(I_0) = 0$  dans une région de contours).

La valeur  $\frac{1}{2}$  représente un coefficient de normalisation pour les deux fonctions de contrôle  $u$  et  $v$ . L'utilisation de la somme  $u(\|\nabla I\|) + v(\beta)$  permet de diffuser même lorsque le gradient est fort et que l'angle  $\beta$  est grand ; aussi, en présence d'un angle  $\beta$  aigu et d'un gradient faible, la diffusion sera également importante (ce qui ne serait pas valable avec l'utilisation de la multiplication). La diffusion en un pixel est faible lorsque l'angle  $\beta$  est aigu et que le gradient est fort, en fonction des deux paramètres  $(h, k)$ .

Contrairement à certaines approches de diffusion par EDP, nous ne souhaitons pas inhiber le lissage au niveau des contours car les deux directions  $\theta_1$  et  $\theta_2$  sont suffisamment précises pour préserver les contours. Les valeurs  $k = 0.5$  (équation 2.33) et  $h = 0.8$  (équation 2.34) permettent de diriger la diffusion avec précision au niveau des contours et des coins, un résultat est disponible en figure 2.76 (c).

Les paramètres utilisés pour le filtre de lissage sont  $\mu = 5$ ,  $\lambda = 1.5$ ,  $\Delta\theta = 5^\circ$  et pour le filtre dérivé  $\mu = 10$ ,  $\lambda = 1$ ,  $\Delta\theta = 5^\circ$ . Choisir un filtre de lissage avec une longueur plus courte que le filtre de dérivation permet de ne pas estimer

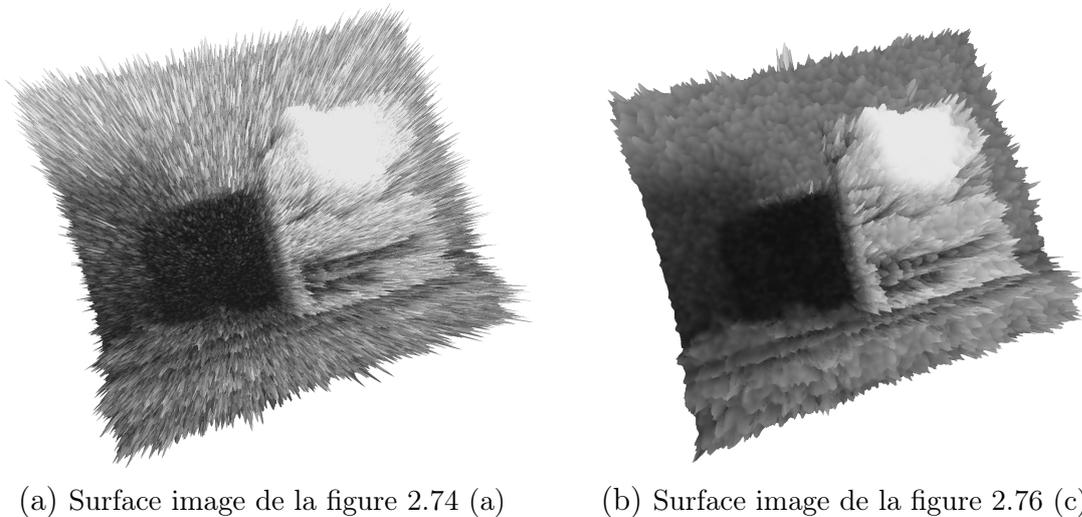


FIGURE 2.77 – Comparaison de la surface image originale avec la surface de l'image restaurée utilisant les fonctions de contrôle.

des bandes trop larges où la diffusion sera anisotropique. Cependant, comme le filtre de dérivation estimera les deux directions de diffusion aux endroits où celle-ci sera anisotropique, l'information est recherchée plus loin, ce qui permet d'être plus robuste au bruit.

L'algorithme de Perona-Malik [PM90] (noté PM) sur notre résultat en figure 2.76 illustre que notre diffusion orientée dans les directions des contours préserve les petits objets comme ceux de taille plus importante dans l'image. En effet, les lettres ainsi que le coté gauche de la boîte sont toujours visibles tout comme les deux traits horizontaux au bas de l'image.

Enfin, la figure 2.76 (f) illustre les contours préservés par rapport au schéma de diffusion sans fonction de contrôle en figure 2.76 (f) (équation 2.32) et la figure 2.77 représentant la surface image montre à la fois les contours tranchants ainsi que les régions régularisées.

### Adaptation du schéma de diffusion avec contrôle de la diffusion au niveau des contours et des coins pour les images couleurs

Une image couleur  $\mathbf{I}$  composée de  $d$  canaux  $I^i$  avec  $i^{\text{ème}}$  composante de  $\mathbf{I}$ , où  $1 \leq i \leq d$ . Afin de restaurer une image vectorielle, la méthode se décompose en deux étapes :

1. calculer les zones plates de  $s(\theta)$  pour chaque canal présentée en partie 4.3 de ce chapitre pour chaque  $I^i$ ,
2. régulariser chaque composante  $I^i$  de l'image  $\mathbf{I}$  avec le même schéma de diffusion présenté en équation 2.35

Nous appliquons notre détection de zones plates du signal  $s(\theta)$  présentée en partie 4.3 de ce chapitre pour chaque composante  $I^i$  de  $\mathbf{I}$ . Nous diffusons anisotropiquement l'image au niveau d'un pixel si nous détectons au moins une zone plate pour le signal  $s(\theta)$  ( $30^\circ < \alpha^i < 360^\circ$ , avec  $\alpha^i$  la zone plate la plus large pour un pixel du canal  $I^i$ ).

Puis, nous diffusons anisotropiquement chaque composante  $I^i$  de l'image  $\mathbf{I}$  où les zones plates ont été détectées dans les directions  $(\theta_1^{\text{couleur}}, \theta_2^{\text{couleur}})$  et isotropiquement dans les autres parties avant de resynthétiser l'image couleur résultat. Les deux directions  $(\theta_1^{\text{couleur}}, \theta_2^{\text{couleur}})$  sont estimées en chaque pixel de coordonnées  $(x, y)$  d'après l'équation 2.12 :

$$\mathcal{Q}(x, y, \theta_1^{\text{couleur}}) = \arg \max_{\mathcal{Q}(x, y, \theta_1^i), 1 \leq i \leq d} (\|\nabla \mathbf{I}(x, y)\|), \quad (2.36)$$

$$\mathcal{Q}(x, y, \theta_2^{\text{couleur}}) = \arg \max_{\mathcal{Q}(x, y, \theta_2^i), 1 \leq i \leq d} (\|\nabla \mathbf{I}(x, y)\|), \quad (2.37)$$

où  $(\theta_1^i, \theta_2^i)$  représentent les deux extrema de la fonction  $\mathcal{Q}$  pour chaque pixel de la composante  $I^i$  de l'image  $\mathbf{I}$ .

D'après l'équation de diffusion dirigée par les fonctions  $u$  et  $v$  (équation 2.35), l'EDP de diffusion couleur devient alors :

$$\begin{cases} I_t^i &= F_A(\mathbf{I}_0) \Delta I_t^i + (1 - F_A(\mathbf{I}_0)) \cdot \left( \frac{u(\|\nabla \mathbf{I}\|) + v(\beta^{\text{couleur}})}{2} \right) \cdot \frac{\partial^2 I_t^i}{\partial \theta_1^{\text{couleur}} \partial \theta_2^{\text{couleur}}} \\ \mathbf{I}(0) &= \mathbf{I}_0 \end{cases} \quad (2.38)$$

où :

- $t$  représente le temps de diffusion,
- $\mathbf{I}(0)$  l'image vectorielle originale,
- $I_t^i$  la composante  $I^i$  de l'image  $\mathbf{I}(0)$  diffusée au temps  $t$ ,
- $\beta^{couleur}$  la bissectrice de l'angle formé par les directions  $(\theta_1^{couleur}, \theta_2^{couleur})$ ,
- $F(\mathbf{I}_0)$  la fonction de contrôle  $F(\mathbf{I}_0) = 1$  lorsqu'il s'agit d'une région homogène,  $F(I_0) = 0$  dans une région de contours).

### Résultats et évaluation

Nous présentons des résultats à la fois sur images scalaires et vectorielles à différents niveaux de bruits. Dans un premier temps, nous apportons des résultats quantitatifs puis des résultats qualitatifs afin d'évaluer la préservation des contours par notre approche.

Dans cette partie, les résultats proposés sont issus d'images dégradées par un bruit blanc uniforme additif et ont été comparés avec deux autres méthodes [TM98] [Tsc06]. En notant  $d$  le nombre de canaux d'une image vectorielle  $\mathbf{I}$  et  $I^i$  sa  $i^{me}$  composante ( $1 \leq i \leq d$ ) le bruit additionné à l'image originale peut s'écrire par la formule suivante :

$$I_m^i = (1 - L)I_0^i + L \cdot I_N,$$

où  $I_0^i$  représente le  $i^{me}$  canal de l'image originale,  $I_N$  une image de bruit aléatoire et  $I_m^i$  la  $i^{me}$  composante de l'image bruitée résultante.

Dans notre test, nous avons dégradé les images avec  $L \in [0.1, 0.9]$  discrétisé par pas de 0.1. Nous obtenons ainsi des résultats évolutifs en fonction du niveau  $L$  de bruit dans les images. Afin d'apporter des résultats quantitatifs à notre approche, nous avons calculé la racine carrée de l'erreur quadratique moyenne RMSE (*root mean square error*) et le rapport signal sur bruit (en dB) appelé PSNR (*Peak Signal to Noise Ratio*) pixel à pixel donnés par les formules suivantes :

$$\text{RMSE} = \sqrt{\frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I_0(i, j) - I_m(i, j))^2}$$

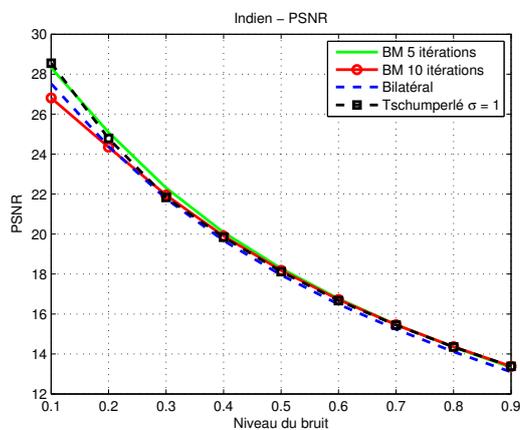
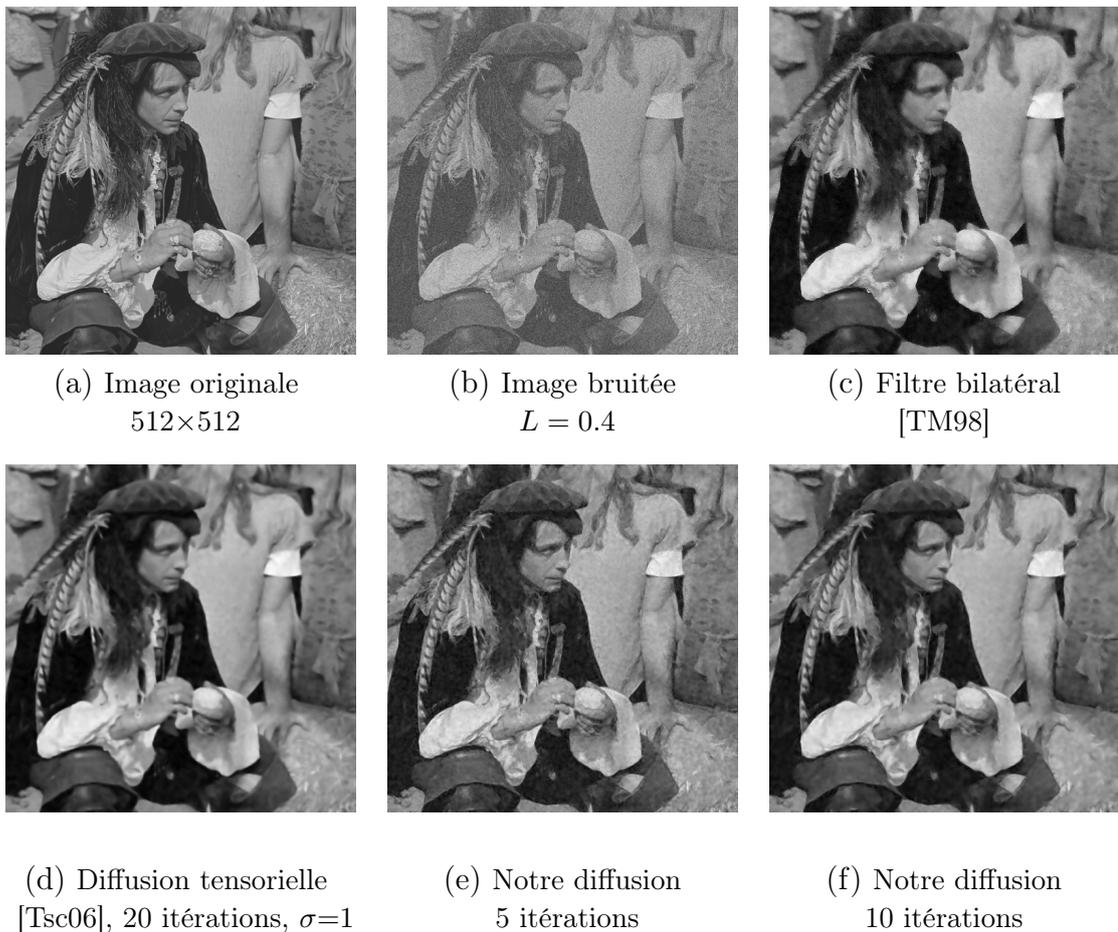
$$\text{PSNR} = 20 \cdot \log_{10} \left( \frac{\max_{(i,j) \in [0,m[ \times [0,n[} I_0(i, j)}{\text{RMSE}} \right)$$

où  $I_0(i, j)$  représente l'intensité lumineuse du pixel de coordonnées  $(i, j)$  de l'image  $I_0$  ayant pour taille  $n \times m$ .

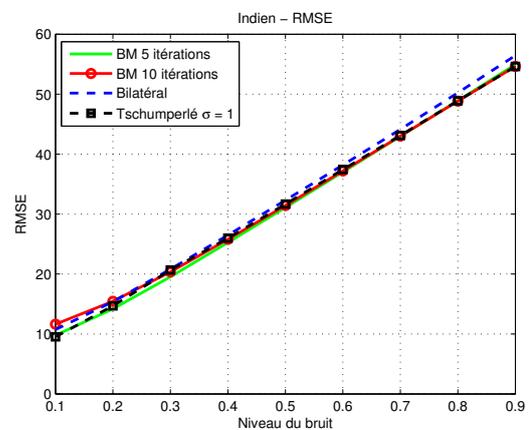
L'image sera considérée de meilleure qualité lorsque la valeur de l'erreur RMSE sera la moins élevée et celle du PSNR plus élevée.

**Résultats pour des images scalaires :** Les résultats obtenus sont comparés avec le filtre bilatéral [TM98] et la méthode de diffusion tensorielle de Tschumperlér [Tsc06] où le nombre d'itérations et les paramètres sont similaires pour chaque niveau de bruit. Nous appellerons par BM notre approche de restauration des images.

Dans la première image scalaire présentée en figure 2.78, de nombreux détails sont présents. Le filtre bilatéral permet de préserver ces détails mais ne supprime pas assez le bruit lorsque celui-ci devient trop fort. Même après plusieurs itérations, il faudrait alors utiliser un filtre plus large, ce qui perdrait de nombreux contours. L'approche de Tschumperlé diffuse très bien le bruit, cependant nous pouvons remarquer que l'image devient floue au bout de peu d'itérations (ici 20). Nos résultats

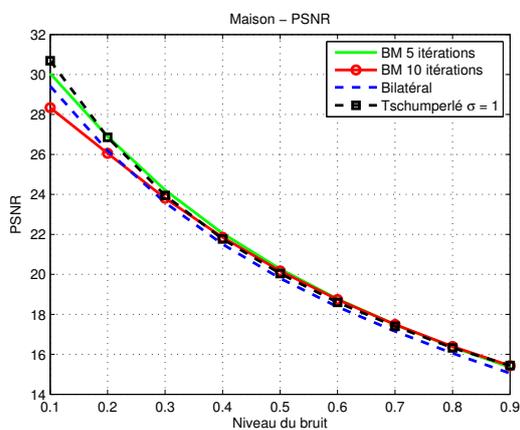
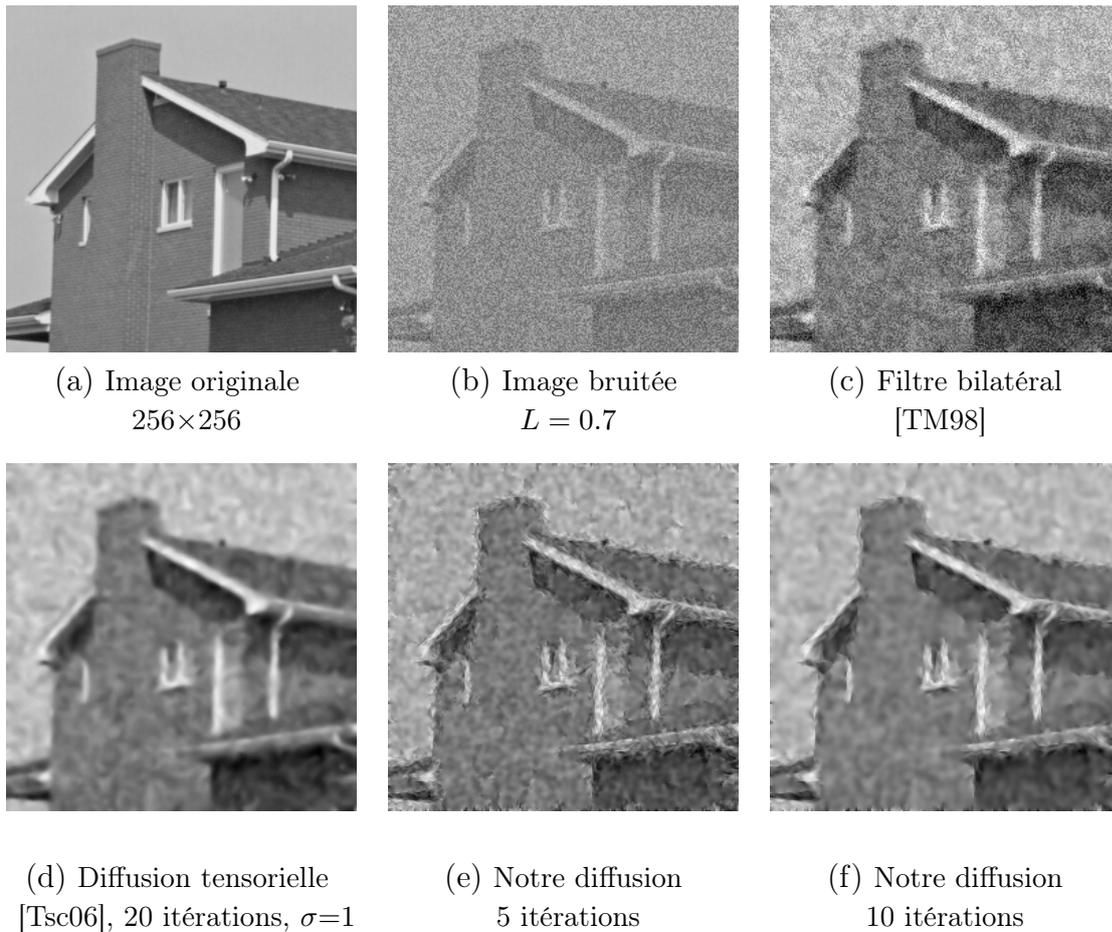


(g) Evolution du PSNR

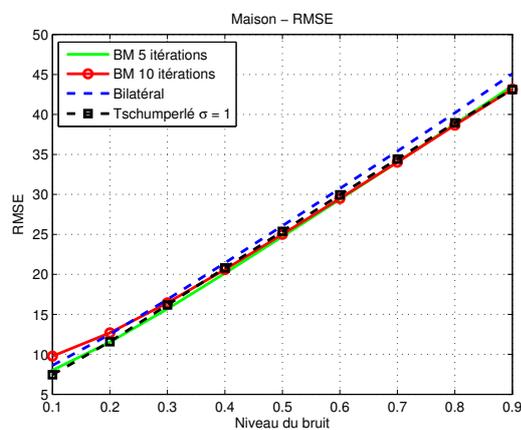


(h) Evolution du RMSE

FIGURE 2.78 – Restauration d'une image scalaire comportant de petits détails.



(g) Evolution du PSNR



(h) Evolution du RMSE

FIGURE 2.79 – Restauration d’une image scalaire comportant des structures linéaires.

présentés en figures 2.78 (e) et (f) montrent que les contours et les coins sont ainsi bien préservés. Afin de supprimer totalement de bruit pour une valeur  $L = 0.4$ , 10 itérations suffisent pour obtenir une image très propre. Les valeurs du PSNR et de la RMSE sont très proches entre les trois méthodes, cependant les résultats visuels sont loin d’être identiques puisqu’après 10 itérations avec notre approche,



(a) Image originale  
512×480

(b) Image bruitée  
 $L = 0.5$

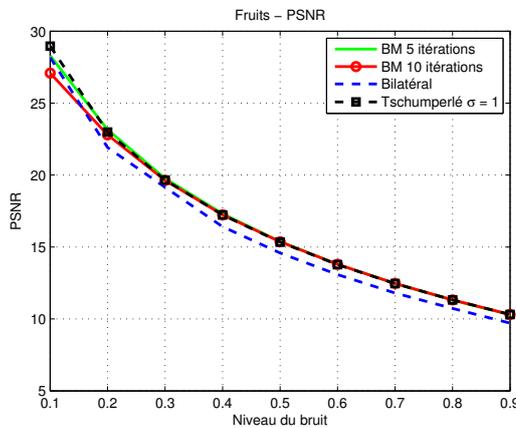
(c) Filtre bilatéral  
[TM98]



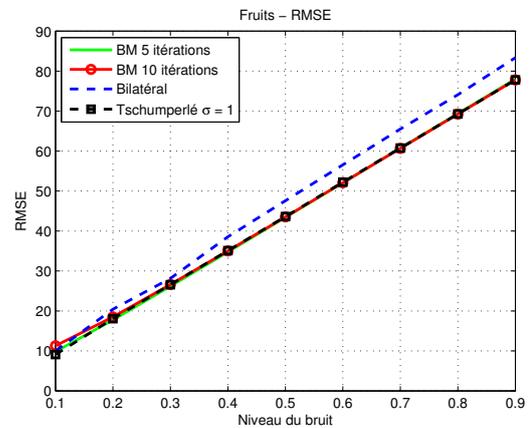
(d) Diffusion tensorielle  
[Tsc06], 20 itérations,  $\sigma=1$

(e) Notre diffusion  
5 itérations

(f) Notre diffusion  
10 itérations



(g) Evolution du PSNR

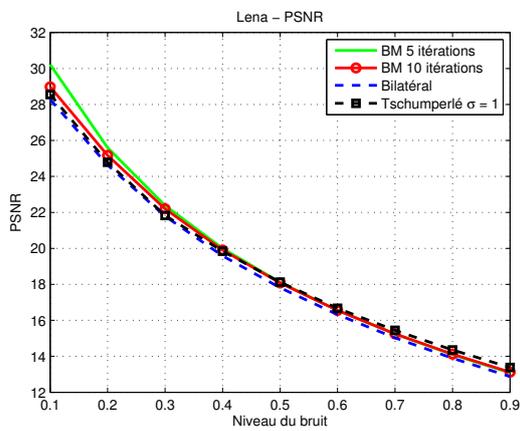


(h) Evolution du RMSE

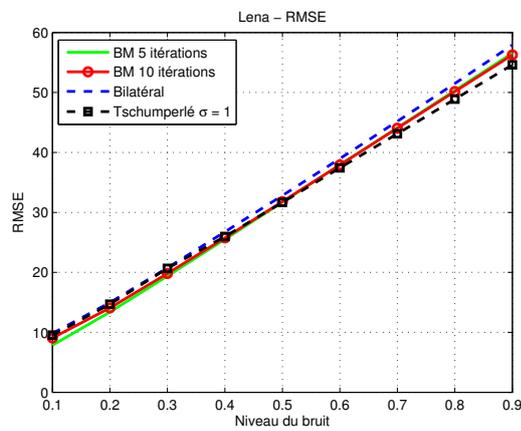
FIGURE 2.80 – Restauration d’une image vectorielle.

l’image semble plus régulière. La figure 2.82 (b) illustre également la préservation des contours qui deviennent tranchants ainsi que le résultat du lissage à l’intérieur des zones homogènes.

L’image de la maison en figure 2.79 (a) présente majoritairement des longues structures. En effet, les différentes toitures, gouttières et fenêtres représentent des contours linéaires dans l’image. Notre schéma de diffusion est très bien adapté à ce type d’images avec ses demi-noyaux de dérivation estimant les deux directions



(g) Evolution du PSNR



(h) Evolution du RMSE

FIGURE 2.81 – Restauration d’une image vectorielle comportant de petits détails.

de diffusion. Même si les valeurs du PSNR et de RMSE pour notre méthode sont très proches de la méthode de Tschumperlé, le résultat de notre approche présenté en figure 2.79 (f) montre que les principales structures de l’image sont préservées avec une valeur  $L = 0.7$ . Le bruit est trop important pour être éliminé par le filtre bilatéral, contrairement à l’approche de Tschumperlé mais qui rend flou les contours.

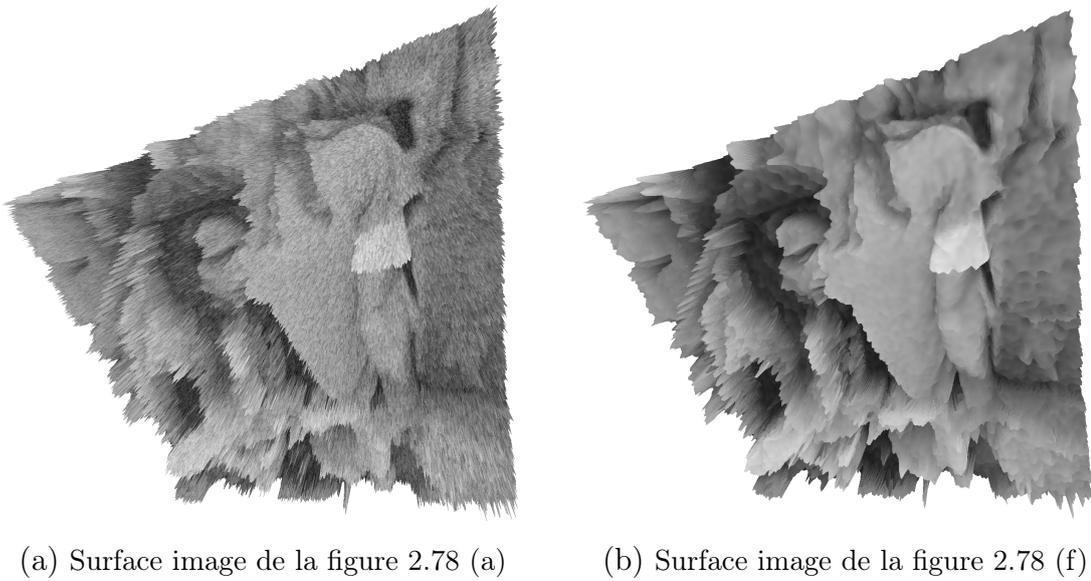


FIGURE 2.82 – Comparaison de la surface image originale bruitée avec la surface de l’image restaurée.

**Résultats pour des images vectorielles :** Nous présentons deux résultats obtenus sur deux images vectorielles à deux niveaux de bruits différents. Tout comme les images scalaires, la méthode de Tschumperlé [Tsc06] et le filtre bilatéral [TM98] sont efficaces lorsque le niveau de bruit n’est pas très important ( $L < 0.4$ ).

Les contours des fruits restent très nets avec notre méthode (figure 2.80 (e) et (f)) et le bruit a totalement disparu alors que l’image à restaurer était composée à 50 pour cent seulement de l’image originale. Le filtre bilatéral n’est pas efficace et les valeurs du PSNR et de la RMSE montrent également la supériorité de l’approche tensorielle [Tsc06] et de notre méthode. Même si les valeurs du PSNR et de la RMSE de notre méthode sont proches de celles de Tschumperlé, ces mesures ne sont pas pertinentes puisque les contours des différents objets sont mieux préservés avec notre approche.

Le dernier résultat présenté en figure 2.81 montre la robustesse de notre méthode pour restaurer à la fois les contours linéaires et les fines structures. En effet, l’image bien connue de Lena est composée de contours droits en arrière plan, le miroir, le bord de l’épaule et du visage mais aussi de structures fibreuses telles que les plumes et le chapeau. Cette image comporte aussi des objets fins comme les yeux, le nez, les lèvres ou encore les cheveux.

Le résultat de notre approche après 10 itérations (figure 2.81 (e)) préserve les fines structures, même les lèvres qui ne sont pas très marquées dans l’image bruitée. L’image en figure 2.81 (b) est composée majoritairement par le bruit plutôt que par l’image originale. Pourtant notre méthode apporte un résultat perceptuel acceptable préservant très bien les contours par rapport au niveau de bruit dans l’image à restaurer ( $L = 0.6$ ), c’est ce que nous allons évaluer dans le paragraphe suivant.

**Mesure de qualité visuelle :** Au lieu de calculer une différence locale comme le PSNR qui ne prend pas en compte la perception de l’image, il existe une mesure

qui estime la similarité entre deux images. Cette métrique appelée SSIM (*Structural SIMilarity*) mesure les structures similaires entre deux images [WBSS04]. Elle est basée sur plusieurs fenêtres de l'image. La SSIM se mesure entre les deux fenêtres  $x$  et  $y$  de taille  $N \times N$  (généralement  $8 \times 8$ ) et la formule mathématique est donnée par :

$$\text{SSIM}(x, y) = \frac{(2 \cdot \mu_x \cdot \mu_y + c_1)(2 \cdot \text{cov}(x, y) + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\text{var}(x) + \text{var}(y) + c_2)}$$

avec

- $\mu_x$  la moyenne de la fenêtre  $x$  ;
- $\mu_y$  la moyenne de la fenêtre  $y$  ;
- $\text{var}(x)$  la variance de la fenêtre  $x$  ;
- $\text{var}(y)$  la variance de la fenêtre  $y$  ;
- $\text{cov}(x, y)$  la covariance de  $x$  et  $y$  ;
- $c_1 = (0.01 \cdot L)^2$ ,  $c_2 = (0.03 \cdot L)^2$  deux variables destinées à stabiliser la division lorsque le dénominateur est très faible où  $L$  représente la dynamique des valeurs des pixels, soit 255 pour des images en niveaux de gris codées sur 8 bits.

La mesure rapporte des valeurs entre zéro et un, lorsque le résultat est proche de 1, les structures sont considérées comme très bien conservées. La mesure totale de la SSIM sur l'image entière est donnée par la moyenne des SSIM sur toutes les fenêtres  $x$  et  $y$ .

Ainsi, nous avons effectué les mesures de SSIM en fonction du niveau de bruit sur les quatre images présentées en figures 2.78, 2.79, 2.80 et 2.81. Cette mesure rend mieux compte de la préservation des contours.

De manière générale, concernant notre méthode, lorsque le bruit est inférieur à cinquante pour cent dans l'image, 5 itérations de diffusion permettent de conserver mieux les contours, mais lorsqu'il dépasse ce seuil, 10 itérations sont nécessaires. L'écart de la SSIM diminue entre 5 et 10 itérations pour un bruit inférieur à cinquante pour cent lorsque l'image est composée de textures pas très marquées comme sur l'image de la maison en figure 2.79. En effet, notre approche supprime les textures fines et les diffuse isotropiquement, d'où une perte d'information par rapport à l'image originale.

Lorsqu'une image est composée majoritairement de petits objets, le résultat de la SSIM est meilleur pour notre approche avec 5 itérations comme les images de Lena et de l'indien. Par contre, une image composée de gros objets, comme la maison ou les fruits les valeurs de SSIM se rejoignent à partir d'un niveau de bruit autour de quarante pour cent.

Le filtre bilatéral donne de bon résultats lorsque le bruit est faible mais les valeurs de la SSIM décroissent rapidement. Les valeurs de l'approche de Tschumperlé sont meilleures en général. Toutefois, notre méthode avec 10 itérations est supérieure en terme de SSIM lorsque le niveau du bruit est situé entre 30 et 70 pour cent (au delà, le bruit est trop fort pour comparer les résultats avec ce type de mesures).

**Choix des paramètres :** Dans les résultats proposés sur les images scalaires ou vectorielles, nous avons utilisé les mêmes paramètres :

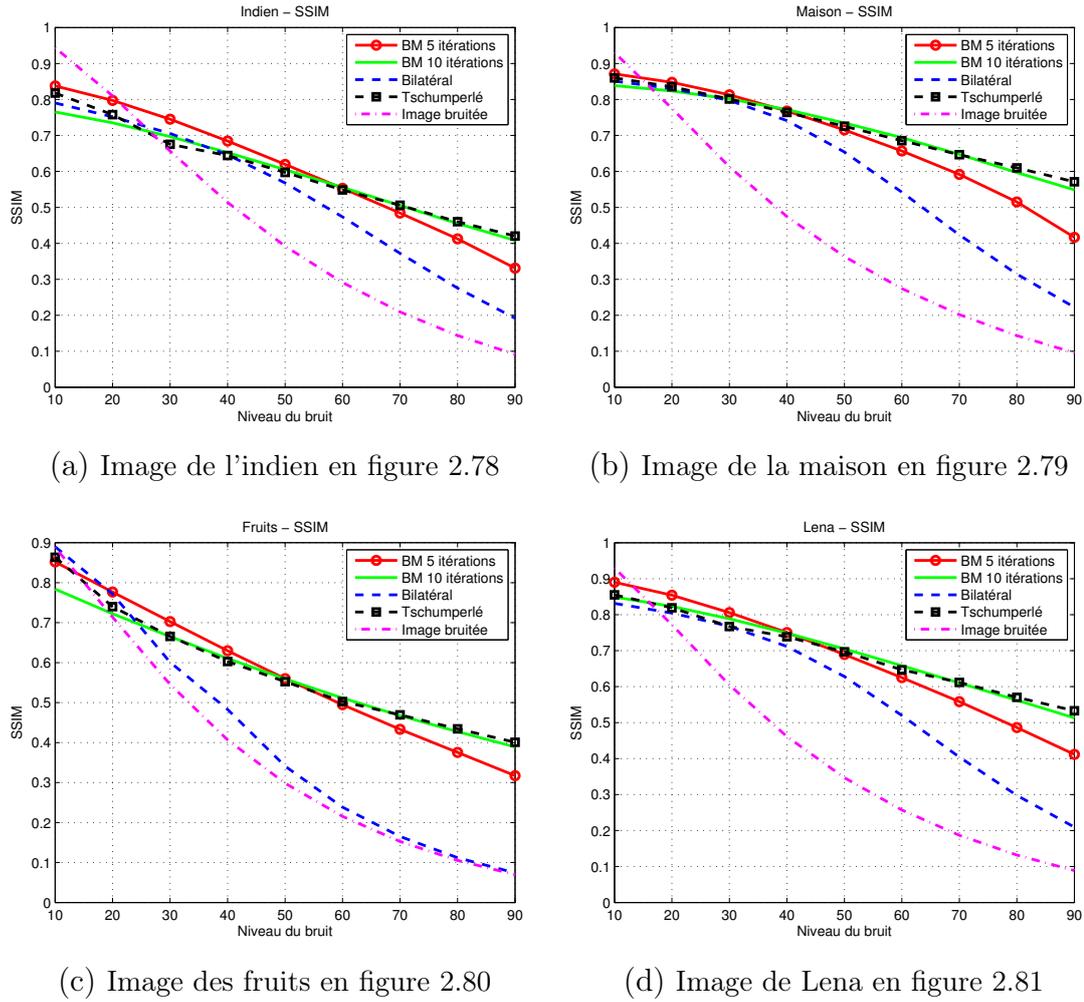


FIGURE 2.83 – Evolution du SSIM sur les différentes images.

- $\mu = 5$ ,  $\lambda = 1.5$ ,  $\Delta\theta = 5^\circ$  et  $s_{th} = 0.005$  pour le filtre de lissage afin de détecter les zones de l'image où  $s(\theta)$  contient des zones plates,
- $\mu = 10$ ,  $\lambda = 1$  et  $\Delta\theta = 2^\circ$  pour le filtre de dérivation  $Q(x, y, \theta)$  afin d'estimer les deux directions de lissage  $(\theta_1, \theta_2)$  et la valeur d'un gradient dans les zones de l'image où  $s(\theta)$  contient des zones plates.

Concernant le filtre de lissage, les paramètres choisis permettent de détecter des zones où la diffusion sera anisotropique. Cependant, il ne faut pas que ces zones soient trop éloignées des contours. C'est pour cette raison que nous avons considéré que pour  $\mu = 5$ , nous estimons grossièrement les contours sans obtenir une bande trop large autour des bords. Il est tout à fait possible de choisir  $\mu = 10$ , mais ce paramètre résultera de bandes plus larges et la diffusion anisotropique lissera peu l'image proche des bords si l'angle  $\beta$  est petit (pour la diffusion de texture grossière, le choix  $\mu = 10$  peut être efficace mais pas pour les images bruitées contenant des petits objets). Le choix  $\lambda = 1.5$  permet de lisser le bruit sans perdre l'information des contours. En effet, pour l'estimation des zones de l'image où  $s(\theta)$  contient des zones plates, il faut que le signal  $s(\theta)$  soit le plus régulier possible et d'après les tests,  $\lambda = 1$  n'est pas assez robuste. Aussi, une discrétisation de l'angle de rotation

$\Delta\theta = 5^\circ$  permet d'apporter l'information des intensités des pixels autour du pixel central. Il n'est pas nécessaire d'effectuer les calculs avec une discrétisation plus petite de l'angle puisque nous n'estimons pas dans cette méthode de directions de lissage avec le signal  $s(\theta)$  (dans la partie 4.3 de ce chapitre, nous avons estimé les courbures de  $s(\theta)$  et nous avons insisté sur le fait que les directions n'étaient pas assez précises et que cela rendait flous les contours). Enfin, le paramètre clef pour détecter les zones de l'image où  $s(\theta)$  contient des zones plates (i.e. contours ou zones homogènes) est le seuil en amplitude  $s_{th}$  de la dérivée première de  $s(\theta)$ . Avec le jeu de paramètre  $(\mu, \lambda, \Delta\theta)$  ainsi proposé,  $s_{th}$  a été fixé à 0.005 pour toutes les images restaurées, peu importe le niveau du bruit dans l'image. Un choix de seuil  $s_{th}$  plus petit détectera du bruit en plus des contours. Si  $s_{th}$  est plus grand, il ne restera que les structures très marquées mais plus les petits objets. Nous pouvons donc fixer les paramètres pour le filtre de lissage ainsi que le seuil en amplitude de la dérivée de  $s(\theta)$ , il reste donc à déterminer les paramètres du filtre de dérivation afin d'estimer deux directions de diffusion et l'amplitude du gradient.

Même si les paramètres du filtre de lissage peuvent être fixés, le choix des paramètres du filtre de dérivation dépend du contexte. En effet, afin de restaurer de longues structures, le paramètre  $\mu$  peut être choisi assez long (10, 15 ou 20) mais lorsque les objets à restaurer seront de petite taille, ce paramètre doit être plus court. Ayant choisi  $\mu = 10$  dans les tests effectués, nous pouvons remarquer que certains détails comme sur l'image de Lena au niveau de l'oeil droit ont été légèrement diffusés. Cependant, lorsque l'image contient un bruit fort ( $L > 0.5$ ), les directions de diffusion  $(\theta_1, \theta_2)$  sont moins précises avec un filtre plus court, d'où le choix  $\mu = 10$  qui est un bon compromis entre petites et grandes structures. Un paramètre  $\mu = 5$  est pourtant très efficace dans une image contenant un bruit faible. Nous avons effectué de nombreux tests sur différentes images bruitées à des niveaux différents. Ainsi, nous avons remarqué que pour une image très bruitée et un paramètre  $\mu = 5$ , nous arrivions à restaurer certaines petites structures (comme les yeux de l'image Lena) au prix d'une mauvaise diffusion des structures linéaires mais des valeurs de PSNR et RMSE équivalentes que pour un paramètre  $\mu = 10$ . Aussi, le choix  $\lambda = 1$  pour le filtre de dérivation permet de détecter des contours bien localisés au niveau des



FIGURE 2.84 – Restauration avec notre méthode d'une image vectorielle comportant de petits détails faiblement bruitée. Après 5 itérations, certains détails sont encore présents (texture du chapeau, cheveux) mais disparaissent après 10 itérations.

bords contrairement à un filtre plus large (discuté dans la partie 2 de ce chapitre). Enfin, le choix de l'angle de discrétisation  $\Delta\theta = 2^\circ$  permet d'obtenir des directions de diffusion très précises contrairement à un choix de  $\Delta\theta = 5^\circ$  ou  $\Delta\theta = 10^\circ$ . Toutefois, ces deux derniers choix suffisent en détection de contours mais lorsqu'il s'agit de diffusion, les angles doivent être les plus précis possibles pour éviter au maximum de diffuser dans de mauvaises directions.

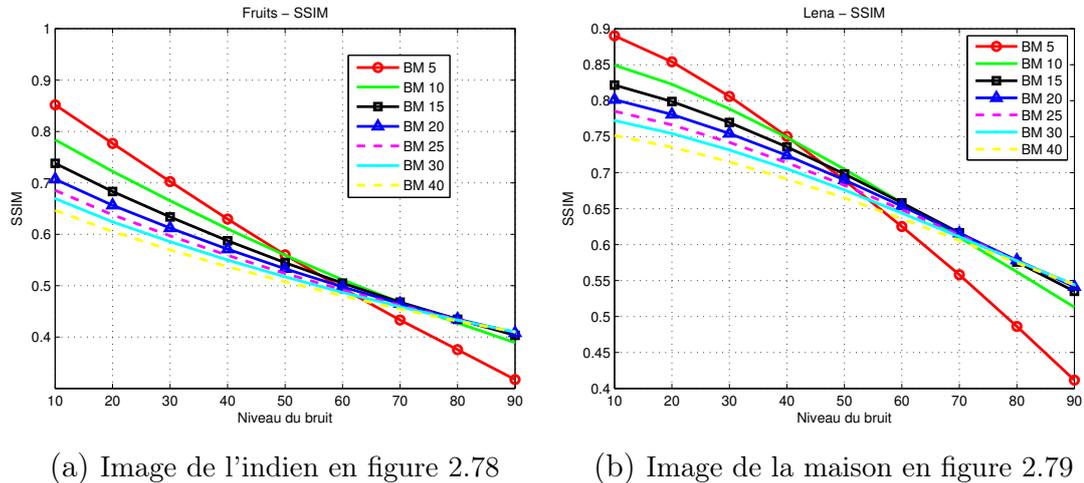


FIGURE 2.85 – Evolution du SSIM sur les différentes images en fonction du niveau de bruit et du nombre d'itérations.

Le dernier paramètre à régler est le nombre d'itérations pour la diffusion. Plus le nombre d'itérations sera important, plus l'image sera lissée et plus l'information sera perdue. Un trop grand nombre d'itérations perdra les petits détails par exemple dans les images faiblement bruitées (illustré en figure 2.84). Cependant, lorsque l'image est fortement bruitée, il est nécessaire d'augmenter le nombre d'itérations. Nous avons remarqué qu'un nombre de 10 itérations suffisaient pour restaurer nos images avec un bruit blanc uniforme additif, lorsque  $L > 0.3$ . Même si les valeurs du PSNR et de RMSE sont proches entre 5 et 10 itérations, perceptuellement et d'après la mesure SSIM, après 10 itérations, les résultats sont meilleurs (figures 2.78 (e) (f), 2.79 (e) (f), 2.80 (e) (f) et 2.81 (e) (f)). La figure 2.85 illustre les mesures de SSIM en fonction du nombre d'itérations de diffusion, 5 ou 10 itérations concernant notre approche permet d'obtenir les meilleurs résultats.

**Conclusion :** Les résultats présentés à la fois sur des images scalaires et vectorielles sont encourageants. En effet, même si les valeurs du PSNR et de RMSE sont similaires à celles du filtre bilatéral et de l'approche de Tschumperlé, perceptuellement, les résultats et les valeurs de la mesure SSIM sont meilleurs. Notre approche est capable de préserver les contours ainsi que les coins tout en restant précis au niveau des bords des objets puisque la diffusion est contrôlée dans deux directions par deux demi-noyaux longs et fins. L'étape de détection de contours larges permet de sélectionner les zones où la diffusion est isotropique, ce qui évite de lisser l'image anisotropiquement dans les zones homogènes et ainsi n'apporte pas un effet *fibreuse* causé par le bruit contrairement aux méthodes tensorielles. De plus, le choix des paramètres pour le filtre de lissage sont fixes, l'utilisateur doit régler les

paramètres pour le filtre de dérivation  $(\mu, \lambda)$  qui varient en fonction de la taille des objets recherchés.

## 4.6 Discussion

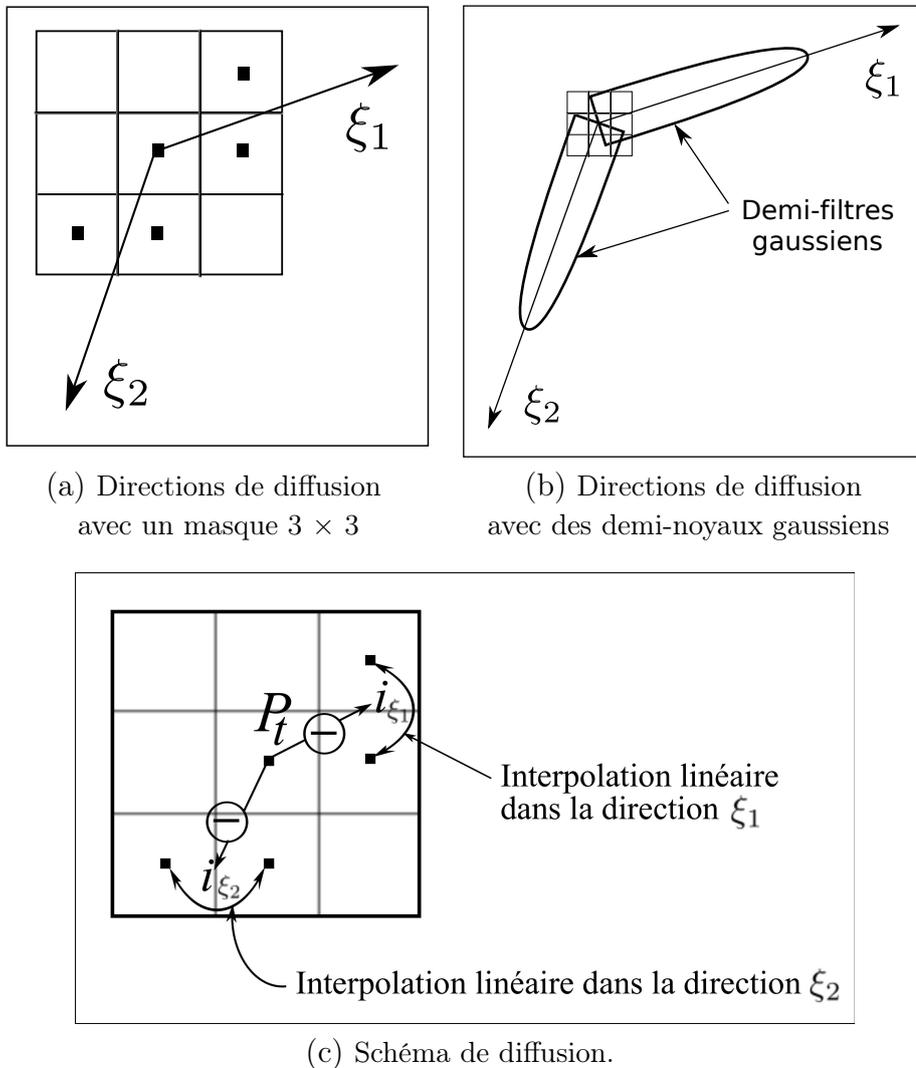


FIGURE 2.86 – Diffusion dans les directions  $\xi_1$  et  $\xi_2$ .

Théoriquement, le long d'un contour, si le pixel considéré de coordonnées  $(x, y)$  est localement positionné sur une isophote, avec notre méthode, la fonction  $F(I_0(x, y))$  est nulle et d'après la partie 4.3, le schéma de diffusion est le suivant :

$$\frac{\partial I_t}{\partial t} = \frac{\partial^2 I_t}{\partial \xi_1 \partial \xi_2}, \quad (2.39)$$

ou, si les directions de diffusion sont estimées à partir du détecteur anisotrope de contours (partie 4.4) :

$$\frac{\partial I_t}{\partial t} = \frac{\partial^2 I_t}{\partial \theta_1 \partial \theta_2}. \quad (2.40)$$

La discrétisation de ce schéma de diffusion revient à la différence entre une dérivée à gauche et une à droite au niveau du pixel comme schématisé en figure 2.86 (c). Pour deux directions de diffusions données  $(\xi_1, \xi_2)$ , le pixel diffusé au temps  $t$  appelé  $P_t$  résulte des interpolations à la fois dans la direction  $\xi_1$  et  $\xi_2$  notées respectivement  $i_{\xi_1}$  et  $i_{\xi_2}$ , soit l'opération suivante :

$$P_{t+1} = P_t + \epsilon \cdot (i_{\xi_1} + i_{\xi_2} - 2 \cdot P_t). \quad (2.41)$$

où  $\epsilon$  représente le coefficient de diffusion ( $\epsilon = 0.15$  dans nos résultats).

Supposons que les directions de diffusion soient  $\xi_1$  et  $\xi_2$ , le niveau de gris est constant le long des lignes isophotes dans ces deux directions. Donc la dérivée de l'image le long des isophotes est nulle :

$$\frac{\partial I_t}{\partial \xi_1} = \frac{\partial I_t}{\partial \xi_2} = 0 \quad (2.42)$$

et le schéma de diffusion résulte :

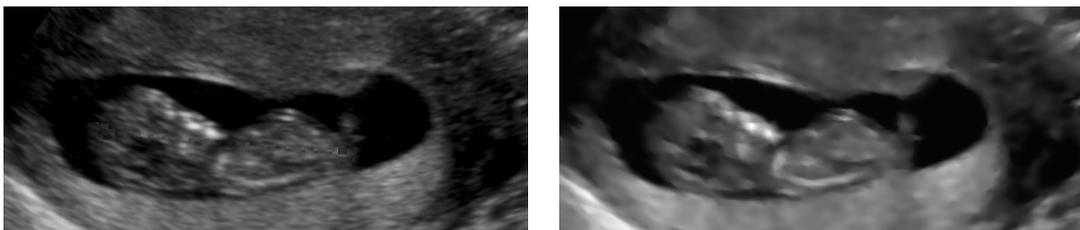
$$\frac{\partial I_t}{\partial t} = 0. \quad (2.43)$$

Si nous utilisons un schéma de diffusion avec un masque  $3 \times 3$ , la diffusion résulte en une interpolation linéaire dans les directions  $\xi_1$  et  $\xi_2$ . Dans l'exemple en figure 2.86 (a), nous souhaitons lisser les pixels dans ces deux directions (représentés par des carrés contenant un point) mais les équations de diffusion 2.39 ou 2.40 seront nulles. Toutefois, dans notre approche, les directions de diffusion ne sont pas calculées localement mais à partir de deux demi-filtre de lissages (détaillé dans la partie 4.2 de ce chapitre et illustré en figure 2.86 (b)) ou de dérivations (détaillé dans la partie 4.4 de ce chapitre). En effet, les noyaux gaussiens apportent une information supérieure à un masque  $3 \times 3$ , par exemple, si  $\mu = 5$ , la distance de lissage sera approximativement de 15 pixels, donc nous obtenons :

$$\frac{\partial I_t}{\partial \xi_1} \neq 0 \text{ et } \frac{\partial I_t}{\partial \xi_2} \neq 0. \quad (2.44)$$

Cependant, dans la pratique nous sommes rarement en présence de cas en équation 2.42. En effet, cela voudrait dire que le pixel considéré appartient à une région totalement sans bruit, par exemple une image synthétique.

## 4.7 Conclusion



(a) Image originale

(b) Image restaurée, 20 itérations

FIGURE 2.87 – Restauration d'une image d'échographie avec notre approche.

Dans cette partie, nous avons proposé une nouvelle méthode de régularisation des images scalaires ou vectorielles. Cette approche est composée d’une classification de pixels suivie d’une diffusion anisotropique de l’intensité de ces mêmes pixels. Notre classification semble très prometteuse puisque nous avons été capable de classifier correctement des régions homogènes et des régions de contours pour différents types d’images. Le processus de diffusion que nous avons développé combine précisément une diffusion isotropique et anisotropique.

Dans un premier temps nous avons développé un schéma permettant de diffuser la texture tout en préservant les principales structures de l’image puis dans une deuxième temps nous avons adapté cette approche à la restauration des images.

La combinaison d’un demi-filtre de lissage orienté et d’un détecteur anisotrope de contours basé sur deux demi-noyaux permettent de contrôler la diffusion au niveau des bords. Ainsi, les textures sont supprimées et les images sont régularisées en préservant leurs principales structures. Ces méthodes sont capables de garder les contours et les coins de différents objets, même dans des images fortement texturées ou bruitées. Les valeurs du PSNR et de RMSE sont similaires à l’approche tensorielle de Tschumperlé [Tsc06], cependant, perceptuellement, nos résultats sont meilleurs et les valeurs de la mesure SSIM montrent la supériorité de notre approche. Ainsi, comparer nos résultats avec les algorithmes existants et les résultats qualitatifs nous permettent de valider notre méthode.

Comme les paramètres pour le filtre de lissage peuvent rester fixés, adapter cet algorithme de restauration des images aux vidéo anciennes serait assez simple. En effet, il suffirait d’estimer les paramètres  $(\mu, \lambda)$  pour le filtre de dérivation et le nombre de diffusions en fonction du niveau du bruit pour obtenir des vidéos restaurées. Aussi, une application aux images médicales peut être développée. Les images médicales sont souvent très polluées par le bruit, il faut généralement un nombre conséquent d’itérations de diffusion afin d’obtenir des images bien lissées. Donc, pour adapter notre approche aux images médicales, le nombre d’itérations sera supérieur à 10 comme illustré en figure 2.87.

## 5 Conclusion

Dans ce chapitre ont été présentées nos contributions. Les trois avancées majeures décrites dans ce manuscrit sont :

- l’extraction anisotrope de bords permettant d’estimer un chemin entrant et un chemin sortant pour chaque pixel traversé par un contour,
- la détection de lignes de crêtes à partir de la différence de deux demi-filtres gaussiens orientés,
- la régularisation des images (suppression de la texture ou du bruit) par diffusion anisotropique dans les deux directions des contours.

Dans un premier temps, nous avons développé un détecteur de contours anisotrope performant. Il permet d’extraire des contours même dans le cas d’images fortement bruitées tout en restant précis au niveau des contours.

Le fait d’extraire les contours à l’aide de deux demi-noyaux gaussiens apporte au filtre de la précision au niveau des coins sans les délocaliser à l’intérieur même de l’objet, contrairement aux détecteurs de contours classiques ou anisotropes. De

plus, il permet d'estimer des contours droits sans être pour autant perturbé par la texture. Une première amélioration a été proposée en utilisant les courbes gamma, ce qui permet d'extraire les contours des objets dans les parties sombre ou sur-exposées des images. De potentielles applications seraient d'extraire des formes dans les images pour la reconnaissance d'objets et la détection de contours dans les images HDR.

Un premier travail applicatif a été effectué dans notre laboratoire à l'aide de ce détecteur pour reconnaître les personnes dans les images [MC09]. Ainsi, les bras, les jambes et le buste sont extraits comme des rectangles et les personnes sont reconnues par la combinaison et l'assemblage de tous ces rectangles. Cette approche pourrait être améliorée en considérant des images vectorielles et en utilisant cet algorithme en temps réel.

D'autres applications dans le domaine médical ou de l'industrie pourraient voir le jour. Une approche intéressante serait le suivi d'objets avec le gradient et sa direction ainsi estimée mais la contrainte majeure reste à paralléliser cet algorithme dans un premier temps.

La deuxième contribution détaillée dans ce rapport concerne la détection de crêtes ou de vallées dans les images. Basée sur la différence de deux demi-gaussiennes anisotropes de la même longueur mais pas la même largeur, notre approche permet d'extraire les contours dans dans images en niveaux de gris, même très bruitées ou floues.

Cette approche est efficace pour détecter les routes dans les images satellite mais aussi les réseaux linéiques dans les images médicales. Ainsi détecter les vaisseaux sanguins dans les images volumiques (par exemple les images IRM) pourrait être un thème intéressant. Cependant, cela nécessiterait de travailler avec un algorithme rapide, optimisé pour ce type d'applications.

Un autre type d'application serait d'extraire et mettre en correspondance des empreintes digitales. En effet, notre approche extrait les lignes de crêtes, même discontinues (qui pourraient être causées par une coupure ou une poussière). Ainsi, extraire soit les crêtes, soit les vallées pourrait mener à identifier l'empreinte relevée par rapport à une empreinte originale.

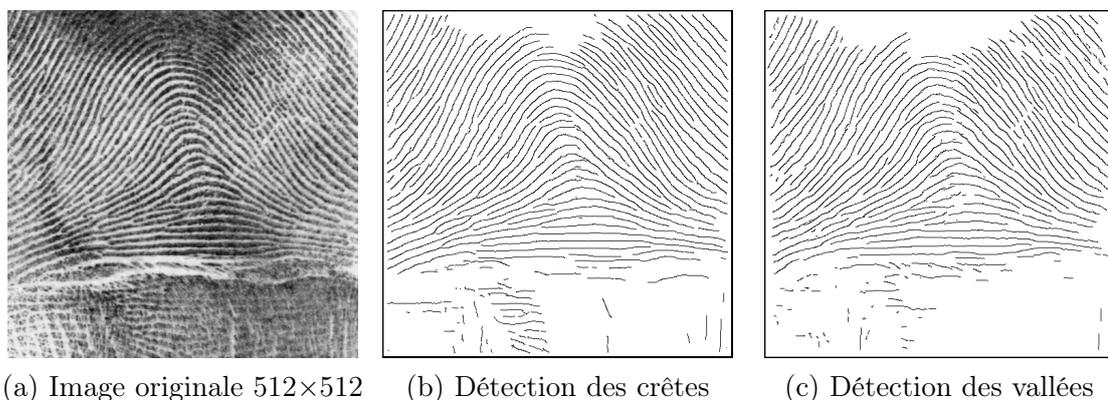


FIGURE 2.88 – Extraction des lignes de crêtes sur une image d'empreinte digitale.

La dernière partie est consacrée à une nouvelle approche de diffusion anisotrope dans les images. Nous avons développé une méthode capable de diffuser dans

les deux directions des contours. Ces deux directions sont issues soit :

- ▷ d'un signal obtenu à partir de demi-noyaux gaussiens orientés évaluant l'intensité lumineuse des pixels autour d'un même pixel,
- ▷ du détecteur anisotrope de contours précédemment détaillé.

Le processus de diffusion est précédé par une étape de détection de contours larges désignant les pixels qui seront diffusés anisotropiquement. Ces approches permettent de diffuser les textures en préservant les principales structures de l'image ou de restaurer les images. La méthode de débruitage permet de préserver les contours des images même en présence d'un bruit fort. Perceptuellement, les résultats obtenus sont meilleurs qu'avec les méthodes comparées et en terme de mesure SSIM, nous obtenons de bons résultats également.

L'approche de régularisation des images pourrait servir pour des applications médicales telles que les images d'échographies ou IRM. Ainsi, les utilisateurs souhaitant restaurer des images fortement bruitées pourraient utiliser une plateforme mise en ligne. L'utilisateur serait mené à faire varier les paramètres de la détection anisotrope de contours en fonction de la taille des objets recherchés dans l'image ainsi que le nombre d'itérations de diffusions. La mise en oeuvre de cette plateforme nécessite la condition d'optimiser au maximum notre algorithme afin qu'il tourne rapidement.

Une potentielle évolution de ce schéma de diffusion serait d'inclure la détection de crêtes ou de vallées à la place du détecteur de contours. En effet, cette approche aurait pour but de restaurer les réseaux fins comme les empreintes digitales ou les images médicales de vaisseaux sanguins par exemples.

Enfin, les demi-noyaux gaussiens anisotropes ont été récemment utilisés pour l'appariement de points dans les images [PMD12]. A partir de points d'intérêts [MGD98], la superposition de deux signaux identiques sur une plage angulaire de 360 degrés permet d'obtenir des résultats très intéressants. Cependant, les points d'intérêts ont été calculés sur toute l'image et peuvent donc être au milieu d'une zone de texture. L'idée serait de détecter les zones de contours larges dans un premier temps puis à l'intérieur de ces zones, détecter les points d'intérêts. Il serait ainsi plus aisé d'apparier les bords des objets pour la reconstruction 3D ou la calibration des caméras.

# Conclusion Générale

Ce manuscrit de thèse porte sur deux sujets majeurs en traitement des images. Le premier concerne la détection de contours, le second la régularisation des images par équations aux dérivées partielles.

La détection de contours est un problème fondamental en vision par ordinateur. Même si des filtres existent depuis plusieurs décennies, trouver un algorithme capable de détecter tous les contours des objets désirés en évitant les fausses détections n'est pas une tâche aisée. Ainsi, sont détaillées plusieurs approches de la littérature et notre algorithme capable de détecter à la fois les petits objets et les structures linéaires est présenté. Les résultats sur des images réelles ainsi que des résultats quantitatifs permettent de montrer la supériorité de notre approche et donc la valider.

Les lignes de crêtes sont aussi régulièrement recherchées dans les images. Ces contours de type toit représentent généralement les routes dans les images satellites ou les vaisseaux sanguins dans celles médicales. Les méthodes classiques recherchent la courbure de la surface de l'image. Ces méthodes détectent correctement les crêtes mais sont perturbées dès qu'un bruit est présent dans l'image. De plus, les lignes discontinues ne peuvent être extraites par ce type d'algorithme. Dans ce rapport de thèse, nous avons proposé une nouvelle approche capable d'extraire les lignes dans les images, même lorsqu'elles ne sont pas continues, même lorsqu'elles sont fortement coudées ou comportent des coins. La description de la méthode est suivie par des résultats ainsi qu'une étude quantitative montrant la résistance au bruit de notre algorithme. Nous avons également adapté cette approche à la détection de jonctions de lignes de crêtes. Des résultats quantitatifs montrent aussi la capacité d'extraire les jonctions en fonction du niveau du bruit dans les images.

La restauration des images est le troisième thème abordé dans ce manuscrit. Afin d'obtenir des images régulières pour un meilleur rendu visuel ou pour la segmentation, il est souvent nécessaire de supprimer le bruit ou la texture au préalable. Cette régularisation peut être résolue par des équation aux dérivées partielles (EDP). Ainsi, sont référencées les principales approches de régularisation des images par EDP puis nous avons également proposé nos propres schémas de diffusion. Nos algorithmes développés sont très performants pour supprimer les textures ou le bruit dans les images. De plus, ils permettent de garder les principales structures et les coins des objets présents dans l'image originale. La comparaison avec les méthodes existantes ainsi que le résultat de différentes mesures qualitatives permettent de valider nos schémas de diffusion.

Finalement, les algorithmes détaillés dans ce manuscrit et développés dans cette thèse ne sont qu'un début à de nombreux travaux qui peuvent en découler. En effet, en conclusion de chaque contribution sont proposé de nouvelles améliorations ou applications pour des travaux futurs.



# Chapitre 3

## ANNEXES

### 1 Implémentation récursive des filtres de Deriche

La transformée en  $Z$  des filtres  $\mathcal{L}$  et  $\mathcal{D}$  montre qu'ils peuvent être mis en oeuvre récursivement d'ordre 2. Considérant une image d'entrée  $x(m)$  et un image de sortie  $y(m)$  et le filtre de lissage de Deriche  $\mathcal{L}(x) = k \cdot (\alpha|m| + 1) \cdot e^{-\alpha|m|}$ , son implémentation récursive est la suivante :

$$y^+(m) = a_0 \cdot x(m) + a_1 \cdot x(m-1) - b_1 \cdot y^+(m-1) - b_2 y^+(m-2), \quad (3.1)$$

pour  $m = 1, \dots, N$ ,

$$y^-(m) = a_2 \cdot x(m+1) + a_3 \cdot x(m+2) - b_1 \cdot y^-(m+1) - b_2 \cdot y^-(m+2), \quad (3.2)$$

pour  $m = N, \dots, 1$ ,

$$y(m) = y^-(m) + y^+(m), \quad (3.3)$$

pour  $m = 1, \dots, N$ , avec :

$$\begin{aligned} a_0 &= k, \\ a_1 &= k \cdot (\alpha - 1) \cdot e^{-\alpha}, \\ a_2 &= k \cdot (\alpha + 1) \cdot e^{-\alpha}, \\ a_3 &= -k \cdot e^{-2\alpha}, \\ b_1 &= -2 \cdot e^{-\alpha}, \\ b_2 &= e^{-2\alpha}, \\ k &= \frac{(1 - e^{-\alpha})^2}{1 + 2 \cdot \alpha \cdot e^{-\alpha} - e^{-2\alpha}}. \end{aligned}$$

Le filtre de dérivation  $\mathcal{D}(x) = -C_1 \cdot x \cdot e^{-\alpha|x|}$  peut également s'implémenter récursivement :

$$y^+(m) = a \cdot x(m) - b_1 \cdot y^+(m-1) - b_2 \cdot y^+(m-2), \quad (3.4)$$

pour  $m = 1, \dots, N$ ,

$$y^-(m) = -a \cdot x(m+1) - b_1 \cdot y^-(m+1) - b_2 \cdot y^-(m+2), \quad (3.5)$$

pour  $m = N, \dots, 1$ ,

$$y(m) = y^-(m) + y^+(m), \quad (3.6)$$

pour  $m = 1, \dots, N$ , avec

$$\begin{aligned} a &= c \cdot e^{-\alpha}, \\ b_1 &= -2 \cdot e^{-\alpha}, \\ b_2 &= e^{-2\alpha}, \\ c &= \frac{(1 - e^{-\alpha})^2}{e^{-\alpha}} \end{aligned}$$

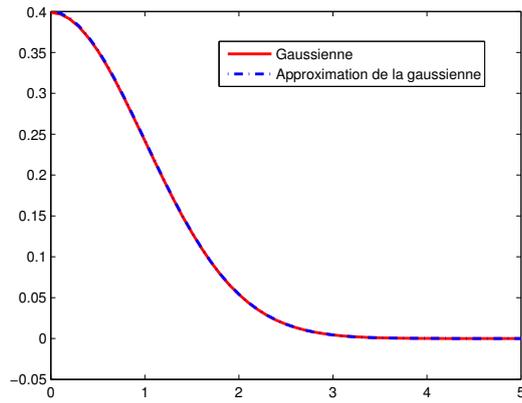
## 2 Approximation récursive du filtre gaussien et de ses dérivées

Le filtre gaussien est séparable mais d'après la transformée en  $Z$ , il ne peut pas s'implémenter récursivement. Il existe cependant de bonnes approximations récursives de ce filtre et de ses dérivées. En effet, Deriche a proposé une approximation dans [Der92] de la gaussienne et de ses dérivées jusqu'à l'ordre 2 par un filtre récursif d'ordre 4 de la forme :

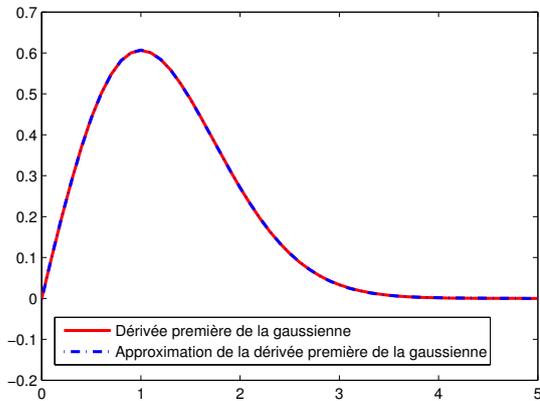
$$h_i(x) = \beta_i \left( \left( a_0 \cos\left(\frac{\omega_0}{\sigma}x\right) + a_1 \sin\left(\frac{\omega_0}{\sigma}x\right) \right) e^{-\frac{b_0}{\sigma}x} + \left( c_0 \cos\left(\frac{\omega_1}{\sigma}x\right) + c_1 \sin\left(\frac{\omega_1}{\sigma}x\right) \right) e^{-\frac{b_1}{\sigma}x} \right) \quad (3.7)$$

où  $i \in \{0, 1, 2\}$  dépend de l'ordre de la dérivée et  $\beta_0 = \frac{0.4}{\sigma}$  est un coefficient de normalisation pour  $h_0$ , 1 sinon. La gaussienne  $g_0$  et ses dérivées normalisées d'ordre 1 et 2 notées  $g_1$  et  $g_2$  sont approximées respectivement par  $h_1$  et  $h_2$ . Ces deux fonctions sont de la même forme que  $h_0$  mais avec différents coefficients donnés dans le tableau suivant :

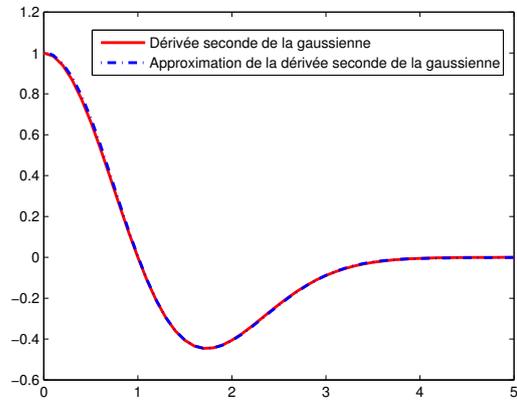
	$h_0(x)$	$h_1(x)$	$h_2(x)$
$a_0$	$1.68/\sigma$	$-0.6472/\sigma^2$	$-1.331/\sigma^3$
$a_1$	$3.735/\sigma$	$-4.531/\sigma^2$	$3.661/\sigma^3$
$c_0$	$-0.6803/\sigma$	$0.6494/\sigma^2$	$0.3225/\sigma^3$
$c_1$	$0.2598/\sigma$	$0.9557/\sigma^2$	$-1.738/\sigma^3$
$b_0$	$-1.783$	$-1.527$	$-1.24$
$b_1$	$-1.723$	$-1.516$	$-1.314$
$\omega_0$	$0.6318$	$0.6719$	$0.748$
$\omega_1$	$1.997$	$2.072$	$2.166$



(a) Gaussienne



(b) Dérivée première



(c) Dérivée seconde

 FIGURE 3.1 – Approximation de la gaussienne et de ses dérivées première et seconde avec  $\sigma = 1$ .

Pour une valeur  $\sigma = 1$ , l'erreur au sens des moindres carrés est inférieure à  $8.6 \cdot 10^{-8}$  pour  $h_0$ ,  $2.77 \cdot 10^{-6}$  pour  $h_1$ , et  $2.942 \cdot 10^{-3}$  pour  $h_2$ . La figure 3.1 illustre les allures de  $h_0$ ,  $h_1$  et  $h_2$  par rapport à  $g_0$ ,  $g_1$  et  $g_2$ .

Les filtres  $h_0$ ,  $h_1$  et  $h_2$  peuvent être implémentés de façon récursive avec  $X_k$  une image d'entrée et  $Y_k$  une image de sortie :

$$Y_k = Y_k^+ + Y_k^- . \quad (3.8)$$

pour  $k = 1 \dots N$ , où  $Y_k^+$  et  $Y_k^-$  représentent respectivement les filtres causal et anti-causal :

$$Y_k^+ = \begin{aligned} & n_0 X_k + n_1 X_{k-1} + n_2 X_{k-2} + n_3 X_{k-3} \\ & - d_1 X_{k-1} - d_2 X_{k-2} - d_3 X_{k-3} - d_4 X_{k-4}, \end{aligned} \quad (3.9)$$

pour  $k = 1 \dots N$ ,

$$\begin{aligned}
 Y_k^- = & n_1^- X_{k+1} + n_2^- X_{k+2} + n_3^- X_{k+3} \\
 & - d_1 X_{k+1} - d_2 X_{k+2} - d_3 X_{k+3} - d_4 X_{k+4}, \quad (3.10)
 \end{aligned}$$

pour  $k = N \dots 1$ , où  $N$  représente la taille du signal. Les coefficients  $d_i$ ,  $i \in \{1, 2, 3, 4\}$ , sont :

$$\begin{aligned}
 d_1 &= -2 e^{-\frac{b_1}{\sigma}} \cos\left(\frac{\omega_1}{\sigma}\right) - 2 e^{-\frac{b_0}{\sigma}} \cos\left(\frac{\omega_0}{\sigma}\right) \\
 d_2 &= 4 \cos\left(\frac{\omega_1}{\sigma}\right) \cos\left(\frac{\omega_0}{\sigma}\right) e^{-\frac{b_0}{\sigma} - \frac{b_1}{\sigma}} + e^{-2\frac{b_0}{\sigma}} + e^{-2\frac{b_1}{\sigma}} \\
 d_3 &= -2 \cos\left(\frac{\omega_1}{\sigma}\right) e^{-\frac{b_1}{\sigma} - 2\frac{b_0}{\sigma}} - 2 \cos\left(\frac{\omega_0}{\sigma}\right) e^{-2\frac{b_1}{\sigma} - \frac{b_0}{\sigma}} \\
 d_4 &= e^{-2\frac{b_1}{\sigma} - 2\frac{b_0}{\sigma}}
 \end{aligned}$$

puis les coefficients  $n_i$ ,  $i \in \{1, 2, 3, 4\}$ , sont :

$$\begin{aligned}
 n_0 &= a_0 + c_0 \\
 n_1 &= e^{-\frac{b_1}{\sigma}} \left( c_1 \sin\left(\frac{\omega_1}{\sigma}\right) - (c_0 + 2a_0) \cos\left(\frac{\omega_1}{\sigma}\right) \right) \\
 &+ e^{-\frac{b_0}{\sigma}} \left( a_1 \sin\left(\frac{\omega_0}{\sigma}\right) - (2c_0 + a_0) \cos\left(\frac{\omega_0}{\sigma}\right) \right) \\
 n_2 &= 2 e^{-\frac{b_1}{\sigma} - \frac{b_0}{\sigma}} \left( (a_0 + c_0) \cos\left(\frac{\omega_1}{\sigma}\right) \cos\left(\frac{\omega_0}{\sigma}\right) - a_1 \cos\left(\frac{\omega_1}{\sigma}\right) \sin\left(\frac{\omega_0}{\sigma}\right) \right) \\
 &+ 2 e^{-\frac{b_1}{\sigma} - \frac{b_0}{\sigma}} \left( -c_1 \cos\left(\frac{\omega_0}{\sigma}\right) \sin\left(\frac{\omega_1}{\sigma}\right) \right) + c_0 e^{-2\frac{b_0}{\sigma}} + a_0 e^{-2\frac{b_1}{\sigma}} \\
 n_3 &= e^{-\frac{b_1}{\sigma} - 2\frac{b_0}{\sigma}} \left( c_1 \sin\left(\frac{\omega_1}{\sigma}\right) - c_0 \cos\left(\frac{\omega_1}{\sigma}\right) \right) \\
 &+ e^{-\frac{b_1}{\sigma} - 2\frac{b_0}{\sigma}} \left( a_1 \sin\left(\frac{\omega_0}{\sigma}\right) - a_0 \cos\left(\frac{\omega_0}{\sigma}\right) \right) \\
 n_4 &= e^{-2\frac{b_1}{\sigma} - \frac{b_0}{\sigma}}
 \end{aligned}$$

La gaussienne et ses dérivées sont soit symétriques soit antisymétriques, dans la cas symétrique (la gaussienne et sa dérivée seconde), les coefficients  $n_i^-$ ,  $i \in \{1, 2, 3, 4\}$ , sont :

$$\begin{aligned}
 n_i^- &= n_i - d_i n_0 \quad \text{pour } i = 1, 2, 3 \\
 n_4^- &= -d_4 n_0,
 \end{aligned}$$

dans le cas de la dérivée première qui est antisymétrique :

$$\begin{aligned}
 n_i^- &= -n_i + d_i n_0 \quad \text{pour } i = 1, 2, 3 \\
 n_4^- &= d_4 n_0.
 \end{aligned}$$

### 3 Implémentation récursive des filtres de Shen-Castan

Les filtres de lissage et de dérivation de Shen et Castan s'implémentent récursivement à l'ordre 1. Considérant une image d'entrée  $x(m)$  et une image de sortie  $y(m)$ , le filtre de lissage :  $F(m) = C_0 \cdot e^{-\alpha|m|}$  est implémenté de la manière suivante :

$$y^+(m) = C_0 \cdot x(m) + e^{-\alpha} \cdot y^+(m-1) \quad (3.11)$$

pour  $m = 1, \dots, N$ .

$$y^-(m) = C_0 \cdot x(m+1) + e^{-\alpha} \cdot y^-(m+1) \quad (3.12)$$

pour  $m = N, \dots, 1$ .

$$y(m) = y^-(m) + y^+(m) \quad (3.13)$$

pour  $m = 1, \dots, N$ , avec  $C_0 = \frac{1-e^{-\alpha}}{1+e^{-\alpha}}$ .

Le filtre de dérivation  $\mathcal{S}$  de l'équation 1.95 s'implémente de la manière suivante :

$$y^+(m) = -C_1 \cdot e^{-\alpha} \cdot x(m) + e^{-\alpha} \cdot y^+(m-1) \quad (3.14)$$

pour  $m = 1, \dots, N$

$$y^-(m) = C_1 \cdot e^{-\alpha} \cdot x(m) + e^{-\alpha} \cdot y^-(m+1), \quad (3.15)$$

pour  $m = N, \dots, 1$ . Finalement,

$$y(m) = y^-(m) + y^+(m), \quad (3.16)$$

pour  $m = 1, \dots, N$ , avec  $C_1 = \frac{(1-e^{-\alpha})^2}{2 \cdot e^{-\alpha}}$ .

Le filtre de Shen-Castan est donc d'ordre 1, soit moins d'opération par pixel que celui de Deriche qui est d'ordre 2 et de l'approximation du filtre gaussien qui est d'ordre 4.

### 4 Axiomatisation de l'analyse multi échelle

**Axiome de récursivité :** Considérons une image  $I_0(x, y)$ , nous générons un semi-groupe de Lie (semi-groupe = pas d'inverse) par application de l'opérateur infinitésimal de l'équation de la chaleur (équation 1.109).

De la même manière, on peut souhaiter une propriété de semi groupe pour l'opérateur  $(T_t)_{t \geq 0}$ . Mathématiquement cet axiome peut s'exprimer de la manière suivante :

Si,  $s$  et  $t$  sont deux instants différents et  $s, t \geq 0$ , alors

$$\begin{aligned} T_0(I_0(x, y)) &= I_0(x, y) \\ T_s \circ T_t &= T_{s+t} \end{aligned}$$

Dans le cas de l'équation de la chaleur, cette propriété se traduit au niveau de la convolution par :

$$I(x, y, t + s) = I_0(x, y) * G_{\sqrt{2(s+t)}}(x, y) = I_0(x, y) * G_{\sqrt{2t}}(x, y) * G_{\sqrt{2s}}(x, y)$$

ce qui signifie que la convolution d'une gaussienne par une autre gaussienne résulte encore d'une gaussienne avec pour écart type la somme des deux précédentes.

Cet axiome imposant des restrictions trop fortes à l'opérateur cherché, on lui préférera une forme plus faible représentée par l'axiome de structure pyramidale.

**Axiome de structure pyramidale :** Cet axiome exprime d'une part que les hautes échelles ne doivent pas influencer les basses échelles et d'autre part que l'image à l'échelle  $t + h$  peut être obtenue directement à partir de l'image à l'échelle  $t$  sans passer par l'image initiale.

Soient  $s$  et  $t$  deux échelles ( $s, t \geq 0$ ),  $T_{s,t}$  un opérateur permettant de passer de l'échelle  $s$  à l'échelle  $t$  et considérant une image  $I_0(x, y)$ , alors :

$$\begin{aligned} T_{t,t}(I_0(x, y)) &= I_0(x, y) \\ T_0(I_0(x, y)) &= I_0(x, y) \\ \forall s \leq t, \exists T_{s,t} \mid T_t &= T_{s,t} \circ T_s \end{aligned}$$

**Axiome de comparaison :** L'analyse doit conserver l'ordre des intensités lumineuses globalement entre deux images. En effet, si une image  $J$  est plus "lumineuse" qu'une image  $I$ , alors cet ordre doit être respecté au cours de l'analyse, ce qui revient à l'axiome suivant :

Soit  $\mathbf{m}$  les coordonnées d'un point de l'image,

$$I(\mathbf{m}) \leq J(\mathbf{m}), \implies T(I(\mathbf{m})) \leq T(J(\mathbf{m})) \quad (3.17)$$

L'axiome de comparaison locale est une version locale de cet axiome.

**Axiome de comparaison locale :** Similairement à l'axiome précédent, cet axiome exprime la conservation de l'ordre des intensités lumineuses localement entre deux images.

Pour toutes les images  $I_t, J_t$ , si il existe une boule topologique de  $\mathbb{R}^2$  centrée en  $\mathbf{m}_0$  et de rayon  $r$  appelée  $B(\mathbf{m}_0, r)$  telle que  $I_t(\mathbf{m}) \leq J_t(\mathbf{m})$  sur  $\{B(\mathbf{m}_0, r) \setminus \{\mathbf{m}_0\}\}$ , alors pour un  $h \in \mathbb{R}$  suffisamment petit :

$$(T_{t+h,t}(I_t))(\mathbf{m}) \leq (T_{t+h,t}(J_t))(\mathbf{m}) \quad (3.18)$$

**Axiome de régularité :** Cet axiome impose des contraintes de régularité à l'évolution d'une image au cours du temps. En effet, l'évolution d'une image régulière (i.e. de forme quadratique) doit se faire d'une façon régulière.

Si l'image  $I$  est une fonction quadratique au voisinage d'un point  $\mathbf{m}_0$ ,  $I$  peut alors s'écrire :

$$I(\mathbf{m}) = a + \mathbf{p}^T (\mathbf{m} - \mathbf{m}_0) + \frac{1}{2} (\mathbf{m} - \mathbf{m}_0)^T \mathbf{A} (\mathbf{m} - \mathbf{m}_0) \quad (3.19)$$

où  $a$  représente un scalaire,  $\mathbf{p}$  un vecteur de dimension deux et  $\mathbf{A}$  une matrice  $2 \times 2$  symétrique définie positive.

La régularité s'exprime alors par l'existence d'une fonction  $F$  non décroissante :

$$F(t, \mathbf{m}, I, \mathbf{p}, \mathbf{A}) : [0, +\infty[ \times \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R}^2 \times \mathbf{S}(2) \longrightarrow \mathbb{R}$$

où  $\mathbf{S}(2)$  est l'ensemble des matrices  $2 \times 2$  symétriques telle que  $F$  soit continue par rapport à  $\mathbf{A}$  et :

$$\lim_{h \rightarrow 0} \frac{T_{t+h,t}(I_t(\mathbf{m})) - I_t(\mathbf{m})}{h} = F(t, \mathbf{m}, I, \mathbf{p}, \mathbf{A})$$

**Théorème fondamental :** Si une analyse multi-échelle  $T_t(I)$  vérifie les axiomes de **structure pyramidale** de **comparaison locale** et de **régularité**, alors :

$$I(x, y, t) = (T_t(I_0))(x, y)$$

est une solution de viscosité (voir [AGLM93]) de l'EDP suivante :

$$\frac{\partial I}{\partial t} = F(\mathcal{H}(I), \nabla I, I, x, y, t) \quad (3.20)$$

où :

$I(x, y, 0) = I_0(x, y)$  est l'image originale,  $F$  est une fonction non décroissante,  $\nabla I$  et  $\mathcal{H}(I)$  représentent respectivement le gradient et le Hessien de l'image  $I$ .

Il s'agit d'une notion faible de solution définie dans le cadre de la théorie des EDP et introduite pour montrer l'existence de solutions de certaines EDP.

Réciproquement, il a été montré que tout opérateur  $T_t$  vérifiant l'équation 3.20, vérifie aussi les différents axiomes architecturaux décrits plus haut.

### Classification des analyses multi-échelles

Nous allons maintenant passer en revue un certain nombre de propriétés d'invariances intéressantes qui vont nous permettre d'effectuer une classification de ces analyses.

**Linéarité :** La propriété de linéarité de l'opérateur  $(T_t)_{t \geq 0}$  est certainement la première propriété que l'on peut espérer d'une transformation. La linéarité énoncée sur l'opérateur infinitésimal peut s'exprimer de la manière suivante :

$$T_{t,t+h}(a I_t + b J_t) = a T_{t,t+h}(I_t) + b T_{t,t+h}(J_t) \quad (3.21)$$

où  $a$  et  $b$  représentent des scalaires,  $I$  et  $J$  deux images.

Autrement dit, l'analyse de la somme pondérée de deux images est la somme pondérée des analyses.

**Invariance euclidienne ou affine :** Soit  $F_{A,b}$  une transformation opérant sur les coordonnées des points :  $F_{A,b} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  définie par  $F_{A,b}(\mathbf{m}) = A\mathbf{m} + b$  où  $A$  représente une matrice non singulière de  $\mathbb{R}^2 \rightarrow \mathbb{R}^2$  et  $b$  un vecteur translation de  $\mathbb{R}^2$ .

Si  $A$  est une isométrie, alors la propriété d'invariance euclidienne s'écrit sous la forme suivante :

$$(T_{t,t+h}(I_t)) \circ F_{A,B} = T_{t,t+h}(I_t \circ F_{A,B}) \quad (3.22)$$

Dans le cas où  $A$  est une transformation affine :

$$AI(x, y) = I(a_{11}x + a_{12}y, a_{21}x + a_{22}y)$$

où  $a_{ij}, (i,j) \in \{1,2\}^2$  sont les coefficients de la matrice  $A$ .

On ne peut alors imposer dans ce cas qu'une propriété de commutation faible définie par :

$$A \cdot T_{t,t+h'} = T_{t,t+h} \cdot A \quad (3.23)$$

avec  $h' = \sqrt{|\det A|} \cdot h$ .

**Invariance morphologique :** Une analyse multi-échelle possédant cette propriété, doit commuter avec tout changement de contraste réalisé à l'aide d'une fonction strictement monotone (cette fois-ci ce sont les intensités lumineuses qui se transforment). Il résulte de cette propriété d'invariance morphologique que seule la notion d'isophote est importante. Dans ce cas, l'analyse affecte uniquement les isophotes.

Soit  $g(\cdot)$  une fonction strictement monotone :  $g : \mathbb{R} \rightarrow \mathbb{R}$ , alors :

$$T_{t,t+h}(g \circ I_t) = g \circ T_{t,t+h}(I_t) \quad (3.24)$$

## 5 Estimation de la pose d'une caméra omnidirectionnelle centrale

# Highly Specific Pose Estimation with a Catadioptric Omnidirectional Camera

Baptiste Magnier, Frédéric Comby, Olivier Strauss, Jean Triboulet  
LIRMM Université Montpellier II  
161 rue Ada, 34392 Montpellier cedex 5, France  
Email: [magnier, comby, strauss, triboule]@lirmm.fr

Cédric Demonceaux  
MIS, 7 Rue du Moulin Neuf,  
80000 Amiens, France  
Email: cedric.demonceaux@u-picardie.fr

**Abstract**—This article presents a new method for estimating the pose of para-catadioptric vision systems. It is based on the estimation of vanishing points associated with vertical edges of the environment. However, unlike classical approaches no feature (line, circle) extraction and/or identification is needed. A sampled domain of possible vanishing points is tested and histograms are built to characterize the soundness of these points. A specificity index allows to find the more relevant histogram and the pose of the sensor. This method has been tested on simulated and real images giving very promising results (maximum angular error of 0.18 degree).

## I. INTRODUCTION

Pose estimation is a classical issue in many robotics applications such as drone navigation and humanoid robotics. This topic is usually addressed using proprioceptive sensors such as inertial sensors: gyroscope and/or inclinometers, or exteroceptive sensors such as cameras. The main issue introduced by proprioceptive sensors is the bias drift of the measures it provides. This bias is due to the fact that pose estimation is achieved by integrating a noisy signal, which leads to an unbounded error growth.

Solutions have been proposed to partially cope with this bias drift. For example, in [5], the authors introduce a state estimation algorithm for rejecting noise and tracking bias of inertial measurement when the motion is intermittent. Another major drawback of inertial measurement is the precision of the pose measurement. Usually, classical performances of such devices are about one to five degree [7][8][9]. A more precise estimation may be useful, especially when the safety of the robot is engaged.

Using exteroceptive sensors, like perspective cameras, is a way to overcome this bias. In that case, the pose is estimated using 3D environment features at each frame and no integration process is needed. However, classical cameras often suffer from a limited field of view, which low their efficiency especially when features: lines, corners, have to be tracked all along a sequence. Solutions using non-conventional cameras (e.g. parabolic catadioptric cameras) have already been proposed to overcome this limited field of view matter [1]. These methods are usually based on tracking and identifying particular geometrical features such as lines in the omnidirectional image [12]. Such features may be difficult to track due to strong radial deformation brought by projections on the mirror

and occlusions. The track can also be affected by the irregular resolution due to anamorphosis on the omnidirectional image.

This paper presents a new approach based on the pose estimation proposed in [2]. It consists in estimating the position of the two vanishing points associated to a set of vertical lines. In opposition to traditional approach, this method does not need any feature extraction. It is based on associating a particular histogram to each possible pose of the sensor. The property used here is the following: the more specific is the histogram, the more likely is the pose estimation. Thus the method consists in finding, in a set of possible orientations of the sensor, the one that is associated to the most specific histogram. Experiments on synthetic and real images are presented to illustrate the good performances we obtain with this method, i.e. a precision about 0.1 degree.

This article is organized as follows: the first section deals with the catadioptric sensor model we use and the relation between pose and vanishing points. The second section details the way histograms are built and the specificity indices we use to select the most probable orientation. The third section presents a synthetic and a real experiment.

## II. OMNIDIRECTIONAL CATADIOPTRIC VISION

### A. Central Catadioptric Model

Geyer and Daniilidis [4] have shown that an omnidirectional image is equivalent to a spherical image if the sensor satisfies the single view point constraint. The construction of this spherical image is depicted in figure 1. The projection of any point  $P$  of the 3D environment of the sensor can be decomposed into two projections: first  $P$  projects on the mirror ( $P_m$ ) then this projection is projected on the image plane ( $P_i$ ) via the optical device. Such a projection is equivalent to first projecting  $P$  on a unitary sphere centered on the mirror focus  $F$  ( $P_s$  is this projection), then projecting  $P_s$  on the image plane in  $P_i$  from the point  $O_s$ . Such an equivalent projection needs the catadioptric camera to be fully calibrated (i.e. know the intrinsic parameters of the projection and the parameters  $\xi = |O_s F|$  and  $\psi = |O_i F|$ ). Then, the image can be fully represented by the projection of all the points of the image plane on the equivalence sphere. This is what we call *the spherical equivalent image*. This kind of model allows the use of the usual projective geometry defined for perspective cameras while using catadioptric cameras.

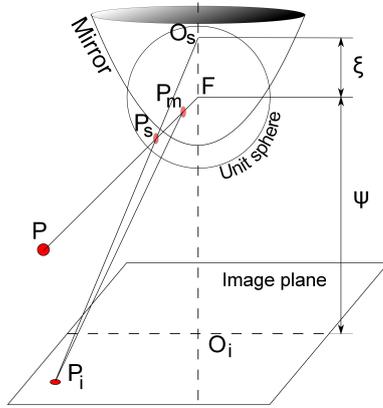


Fig. 1. Equivalence between catadioptric omnidirectional projection and two-step mapping via the sphere

*B. Relation between Pose and Omnidirectional Image*

As proved in [4], a set of lines having the same direction  $\vec{u}$  projects on the sphere into circles that intersect in two antipodal points  $VP_1$  and  $VP_2$  in the sphere, that corresponds to the vanishing directions. Demonceaux et al. [2] have proved that the direction of the vector  $\overrightarrow{VP_1VP_2}$  is parallel to  $\vec{u}$ , and used this property to recover the 3D direction  $\vec{u}$  of an unknown set of parallel lines (via a voting Hough-like process). Let us consider the set of all projections on the sphere of a set of points belonging to a 3D line. Then, every plane passing through each point of this set, the focus  $F$  and one of the antipodal points  $VP$  have exactly the same orientation (Fig. 2). They also contain the other antipodal point. Thus, if the environment is composed of a set of parallel lines, then a histogram of the orientations of the planes generated with every edge point of the image,  $F$  and  $VP_1$  (or  $VP_2$ ) is composed of peaks, each peak corresponding to a line direction. If the environment is not composed only of parallel lines but also of a minority of other points, then these points will generate noise on the histogram.

III. A POLLING PROCESS FOR POSE ESTIMATION

*A. Principle of the method*

The idea that underlies our method is the following: as in indoors environment vertical edges are the most frequent, we will use their high density compared to other edge points to estimate the pose of the device. In fact, if the direction of the catadioptric device is known, so is known the position of the two antipodal points  $VP_1$  and  $VP_2$ . Thus, the histogram of all the directions of the planes passing through every edge point of the spherical image,  $F$  and  $VP_1$  will be very specific. Indeed, this histogram will be composed of as many peaks as vertical edges of the environment and of a random noise due to other points not belonging to vertical edges (Fig. 3 (a)). A wrong orientation of the device will lead to mispositioning the

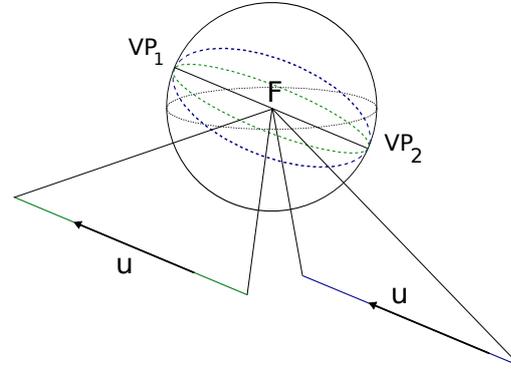


Fig. 2. Parallel lines projections on a spherical image.

two antipodal points  $VP_1$  and  $VP_2$ . This mispositioning will alter the histogram and lower its specificity (Fig. 3 (b)). Thus, if the direction of the catadioptric device is unknown, then one can seek for this direction by constructing such a histogram for each possible direction. What rises from the geometrical properties of the spherical projection is that the more specific the histogram is, the more likely is its associated direction to be the true one (see Figure 3).

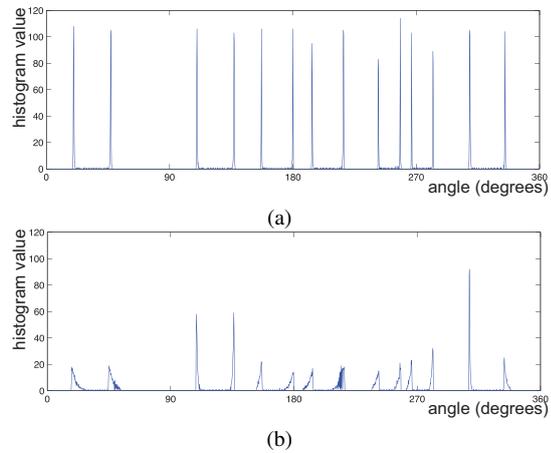


Fig. 3. Specific (a) and sparse (b) histogram.

*B. Construction of a histogram associated to each angle position*

The overall method can be set as follow. First, the pose of the omnidirectional vision device is characterized by two angles  $\alpha$  and  $\beta$  (e.g. pitch and yaw). We then define a sampled domain of all possible values of  $(\alpha, \beta)$  and associate with each sampled value a histogram. This histogram is built by computing the orientation of all the planes obtained by the triplet  $(\vec{u}_{(\alpha,\beta)}, P_E, F)$ , where  $\vec{u}_{(\alpha,\beta)}$  is the orientation vector

and  $P_E$  an edge point. After this construction, we chose for the pose, the value of  $(\alpha, \beta)$  associated to the most specific histogram.

### C. Specificity index of a histogram

In probability theory, it is instrumental to be able to order probability distributions from the point of view of their dispersion [3]. Numerous indices can be used to achieve this goal that are originally defined to make an information-based comparison between two distributions. The most popular of those indices are the Shannon entropy [10] and the Simpson index [11]. More recently, a specificity index has been rediscovered due to Birnbaum under the name of peakedness index [3] that is closely related to the Gini index. Due to the fact that a histogram can be seen as a density distribution estimator, such an ordering can also be used to compare two histograms from the point of view of their relative dispersion.

In this article, we consider the Shannon entropy and Simpson indices and the Birnbaum peakedness. Let  $H = \{h_k\}_{k=1\dots p}$  be the set of values of a histogram built on a partition made of  $p$  bins with  $n$  observations ( $n = \sum_{k=1}^p h_k$ ). The Shannon entropy  $\Phi(H)$ , Simpson index  $S(H)$  and Birnbaum peakedness index  $B(H)$  are defined by:

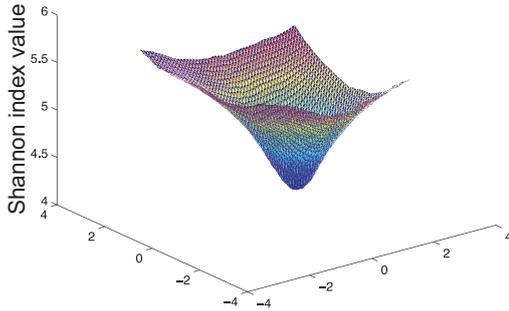


Fig. 4. Shannon index versus roll and pitch angles

$$\Phi(H) = \log(n) - \frac{1}{n} \sum_{k=1}^p h_k \cdot \log(h_k), \quad (1)$$

$$S(H) = \frac{\sum_{k=1}^p h_k \cdot (h_k - 1)}{n \cdot (n - 1)}, \quad (2)$$

$$B(H) = \frac{1}{n} \sum_{l=1}^p \sum_{k=l}^p h_{(k)}, \quad (3)$$

(.) being a permutation that sorts the  $\{h_k\}_{k=1\dots p}$  in ascending order, i.e.  $h_{(1)} \leq \dots \leq h_{(p)}$ . Note that  $S(H)$  and  $B(H)$  are specificity indices while  $\Phi(H)$  is a dispersion index. Therefore, finding the most peaked histogram will lead to maximize  $S(H)$  and  $B(H)$  and minimize  $\Phi(H)$ . Figure 4 illustrates this property by plotting the Shannon index versus the possible roll and pitch angles.

## IV. EXPERIMENTATIONS

### A. Experiment with Synthetic Images

The first experiment has been carried out with a synthetic image sequence. This sequence has been obtained by simulating, with the PovRay software, an environment made of seven black piles (Figure 5(a)), and rotating the omnidirectional device around an axis perpendicular to its symmetry axis with a step of  $\Delta\theta = 0.1^\circ$ . In this experiment, since the omnidirectional vision sensor has been simulated, the projection and mirror parameters are perfectly known and thus the experimental results will not be altered by an ill calibration of the sensor.

### B. Experiment with Real Images

The second experiment has been carried out with a sequence of real images obtained by rotating a para-catadioptric sensor with a step motor with an accuracy of  $0.001^\circ$ . As can be remarked on Fig. 5(b), the image is not made only of vertical lines. In this experiments, all edge points of the image has been used. Due to this fact, the obtained histograms are very noisy, so are the indices. Thus, it can be appropriate to smooth the indices before the sought for extrema: for example a 2D exponential smoothing filter (4) can be used.  $\gamma$  was set to 0.1 for the experiments, which is almost equivalent to a  $\pm 10$  pixels horizon filter.

$$H(x, y) = Ke^{\gamma(|x|+|y|)} \quad (4)$$

Table II presents the results obtained with smoothed and non smoothed indices. The angular step of the rotation is also  $\Delta\theta = 0.1^\circ$ .

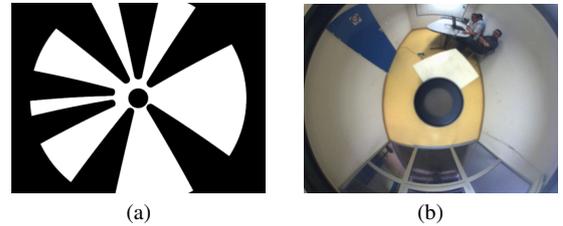


Fig. 5. (a) Synthesized catadioptric image. (b) Example of real image tested.

### C. Experimental results

For both real and simulated experiments, the angular histogram has been built on a high specific partition made of 1800 bins (i.e. the bin width equals  $0.2^\circ$ ). To characterize the accuracy of the method, three dispersion indices have been used: the standard deviation,  $\sigma = \sqrt{\frac{1}{N} \sum_{k=1}^N (\alpha_k^{measured} - \alpha_k^{true})^2}$ , the mean absolute error,  $|E| = \frac{1}{N} \sum_{k=1}^N |\alpha_k^{measured} - \alpha_k^{true}|$  and the maximal error  $E_{max} = \max_{k=1}^N |\alpha_k^{measured} - \alpha_k^{true}|$ . The simulated experiments shows that any specificity index can be used to achieve the sought for orientation since the obtained errors are comparable. Errors obtained on simulated sequences are mainly due to sampling noise introduced by

TABLE I

ESTIMATION ERROR: EXPERIMENTATION WITH THE SIMULATED IMAGES WITH A  $(\alpha, \beta)$  DOMAIN SAMPLING STEP OF  $0.05^\circ$

Sp. Index	smoothed	$ E $	$\sigma$	$E_{max}$
Shannon	no	0.056 <sup>o</sup>	0.075 <sup>o</sup>	0.20 <sup>o</sup>
Simpson	no	0.063 <sup>o</sup>	0.082 <sup>o</sup>	0.20 <sup>o</sup>
Birnbaum	no	0.063 <sup>o</sup>	0.080 <sup>o</sup>	0.20 <sup>o</sup>
Shannon	yes	0.050 <sup>o</sup>	0.068 <sup>o</sup>	0.20 <sup>o</sup>
Simpson	yes	0.057 <sup>o</sup>	0.075 <sup>o</sup>	0.20 <sup>o</sup>
Birnbaum	yes	0.060 <sup>o</sup>	0.076 <sup>o</sup>	0.20 <sup>o</sup>

TABLE II

ESTIMATION ERROR: EXPERIMENTATION WITH THE REAL IMAGES WITH A  $(\alpha, \beta)$  DOMAIN SAMPLING STEP OF  $0.05^\circ$

Sp. Index	smoothed	$ E $	$\sigma$	$E_{max}$
Shannon	no	0.082 <sup>o</sup>	0.106 <sup>o</sup>	0.25 <sup>o</sup>
Simpson	no	0.080 <sup>o</sup>	0.111 <sup>o</sup>	0.40 <sup>o</sup>
Birnbaum	no	0.134 <sup>o</sup>	0.182 <sup>o</sup>	0.60 <sup>o</sup>
Shannon	yes	0.059 <sup>o</sup>	0.081 <sup>o</sup>	0.20 <sup>o</sup>
Simpson	yes	0.053 <sup>o</sup>	0.073 <sup>o</sup>	0.20 <sup>o</sup>
Birnbaum	yes	0.075 <sup>o</sup>	0.120 <sup>o</sup>	0.45 <sup>o</sup>

image or histogram sampling. Within the real experiments, it appears that the Shannon and Simpson indices are less sensitive to the noise induced by minority edges than the Birnbaum index. When comparing the results obtained with and without a preliminary smoothing, it appears clearly that smoothing induces a gain in accuracy. This accuracy can go up to  $0.18^\circ$  when using the Simpson index. This precision is also strongly related to the sampling step of the  $(\alpha, \beta)$  space. The smaller the sampling step is, the better the precision is. For example the mean absolute error drops down from  $0.11^\circ$  to  $0.08^\circ$  with the Shannon index when the step on  $(\alpha, \beta)$  goes from  $0.1^\circ$  to  $0.05^\circ$ . Nevertheless, the computing time increases with the size of the discretized  $(\alpha, \beta)$  space. Under  $0.05^\circ$  the precision seems to stabilize.

### V. CONCLUSION

In this paper, we have proposed a new method for omnidirectional camera pose estimation. The main characteristic of this method is that it does not need any feature identification like lines or circles. It is based on estimating the orientation of the catadioptric device by exploring a sampled space of different possible orientations and associating a histogram to each orientation. The histogram construction is based on the particular geometrical properties of single viewpoint catadioptric cameras. Due to these properties, the specificity of the histogram reflects the soundness of the orientation it is associated with. Different specificity indices have been tested. From these tests, it appears that, in this context, the well known Shannon entropy provides the best results. Within both simulated and real image based experiment, this method can provide a pose estimation with a mean resolution of  $0.06$  degree. The resolution of the estimation is naturally highly linked to both sampling of the orientation space and noise in the images.

The test we achieved prove both feasibility and efficiency of this approach. However, a major drawback remains: the computation time can be quite long: about 19 seconds on a 2.13 G Hz processor for a  $640 \times 480$  image, 2600 sample of the orientation space and 1800 bins for each histogram. First of all optimizing the implementation can improve the overall performances. Second, we consider using histograms built upon fuzzy partition rather than binary ones [6] in order to reduce the number of histogram's bins without reducing the precision of the specificity index. Third, we propose to use the gradient orientation to remove, from the set of edge pixels, any point having an orientation that is too far away from a radial orientation (which should be the orientation of vertical edges). This third modification should drastically reduce the computation time and the noise in histograms. Finally, we propose to dynamically define the sampled orientation domain by using a Kalman filtering-based approach. It can allow a higher precision in the exploration domain and reduce also the computation time.

### ACKNOWLEDGMENT

The authors would like to thank the ANR SHERPA project for their fundings.

### REFERENCES

- [1] J-C. Bazin, C. Démonceaux, P. Vasseur, and I-S. Kweon. Motion estimation by decoupling rotation and translation in catadioptric vision. *Computer Vision and Image Understanding*, In Press, Corrected Proof, 2009.
- [2] C. Démonceaux, P. Vasseur, and C. Pégard. Uav attitude computation by omnidirectional vision in urban environment. In *ICRA*, pages 2017–2022, 2007.
- [3] D. Dubois and A. Hullermeier. Comparing probability measures using possibility theory: A notion of relative peakedness. *International Journal of Approximate Reasoning*, 45(2):64–385, 2007.
- [4] C. Geyer and K. Daniilidis. Catadioptric projective geometry. *International Journal of Computer Vision*, 45(3):223–243, december 2001.
- [5] E. A. Johnson, S. J. M. Bamberg, and M. A. Minor. A state estimator for rejecting noise and tracking bias in inertial sensors. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3256 – 3263, 19-23 May 2008.
- [6] K. Loquin and O. Strauss. Histogram density estimators based upon a fuzzy partition. *Statistics and Probability Letters*, 78(13):1863–1868, September 2008.
- [7] N. Metni, J-M. Pfimlin, T. Hamel, and P. Soueres. Attitude and gyro bias estimation for a vtol uav. *Control Engineering Practice*, 14(12):1511 – 1520, 2006.
- [8] G. Muscato and G. Spampinato. Kinematical model and control architecture for a human inspired five dof robotic leg. *Mechatronics*, 17(1):45 – 63, 2007.
- [9] I-W. Park, J-Y. Kim, B-K. Cho, and J-H. Oh. Control hardware integration of a biped humanoid robot with an android head. *Robotics and Autonomous Systems*, 56(1):95 – 103, 2008.
- [10] C. Shannon and W. Weaver. *The mathematical theory of communication*. University of Illinois Press, 1949.
- [11] E. Simpson. Measurement of diversity. *Nature*, 163:688, 1949.
- [12] B. Vandeportaele, M. Cattoen, and P. Marthon. Realtime localization of a central catadioptric camera using vertical lines. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 416–421, Setubal, Portugal, February 2006.

# Bibliographie

- [AGLM92] L. Alvarez, F. Guichard, P.L. Lions, and J.M. Morel. Axiomes et équations fondamentales du traitement d'images.(analyse multiéchelle et edp). *Comptes rendus de l'Académie des sciences. Série 1, Mathématique*, 315(2) :135–138, 1992.
- [AGLM93] L. Alvarez, F. Guichard, P.L. Lions, and J.M. Morel. Axioms and fundamental equations of image processing. *Archive for rational mechanics and analysis*, 123(3) :199–257, 1993.
- [AK06] G. Aubert and P. Kornprobst. *Mathematical problems in image processing : partial differential equations and the calculus of variations (second edition)*, volume 147. Springer-Verlag, 2006.
- [ALM92] L. Alvarez, P-L. Lions, and J-M. Morel. Image selective smoothing and edge detection by nonlinear diffusion, ii. *SIAM Journal of Numerical Analysis*, 29(3) :845–866, 1992.
- [Alq98] L. Alquier. Analyse et représentation de scènes complexes par groupement perceptuel. *PhD Thesis, Montpellier II University*, 1998.
- [AM94] L. Alvarez and L. Mazorra. Signal and image restoration using shock filters and anisotropic diffusion. *SIAM Journal on Numerical Analysis*, pages 590–605, 1994.
- [AM98] L. Alquier and P. Montesinos. Representation of linear structures using perceptual grouping. In *1st workshop on Perceptual Organization in Computer Vision*, pages 1–21, 1998.
- [AMM97] N. Armande, P. Montesinos, and O. Monga. Thin nets extraction using a multi-scale approach. *Scale-Space Theory in Computer Vision*, pages 361–364, 1997.
- [BC98] P. Blomgren and T.F. Chan. Total variation methods for restoration of vector-valued images. *Transactions on Image Processing*, 7(3) :304–309, 1998.
- [BCM<sup>+</sup>06] A. Buades, B. Coll, J.M. Morel, et al. A review of image denoising algorithms, with a new one. *Multiscale Modeling and Simulation*, 4(2) :490–530, 2006.
- [BGPT02] E. Bourennane, P. Gouton, M. Paindavoine, and F. Truchetet. Generalization of canny-deriche filter for detection of noisy exponential edge. *Signal processing*, 82(10) :1317–1328, 2002.
- [BL06] A. Berengolts and M. Lindenbaum. On the distribution of saliency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1973–1990, 2006.

- [BLMV10] A. Buades, T.M. Le, J.M. Morel, and L.A. Vese. Fast cartoon+texture image filters. *IEEE Transactions on Image Processing*, 19(8) :1978–1986, 2010.
- [Bre83] H. Brezis. *Analyse fonctionnelle : théorie et applications*. Masson, 1983.
- [BRLM90] P. Bolon, A. Raji, P. Lambert, and M. Mouhoub. Symmetrical recursive median filters - application to noise reduction and edge detection. In *5th European Signal Processing Conference*, pages 813–816. IEEE, 1990.
- [Can86] F.J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6) :679–698, 1986.
- [CBFAB94] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *IEEE Proceedings of International Conference on Image Processing*, volume 2, pages 168–172. IEEE, 1994.
- [CD85] J.P. Cocquerez and J. Devars. Détection de contours dans les images aériennes : nouveaux opérateurs. *Traitement du Signal*, 2(1) :45–65, 1985.
- [CLMC92] F. Catté, P.L. Lions, J.M. Morel, and T. Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical Analysis*, pages 182–193, 1992.
- [CM98] V. Caselles and J. Morel. Introduction to the special issue on partial differential equations and geometry-driven diffusion in image processing and analysis. *IEEE Transactions on Image Processing*, 7(3) :269–273, 1998.
- [CMS<sup>+</sup>10] Q. Chen, P. Montesinos, Q.S. Sun, P.A. Heng, and D.S. Xia. Adaptive total variation denoising based on difference curvature. *Image and Vision Computing*, 28(3) :298–306, 2010.
- [CMSSX10] Q. Chen, P. Montesinos, Q.S. Sun, and D. Shen Xia. Ramp preserving Perona-Malik model. *Signal Processing*, 90(6) :1963–1975, 2010.
- [CVDV08] L. Condat and D. Van De Ville. Fully reversible image rotation by 1-d filtering. In *IEEE International Conference on Image Processing*, pages 913–916. IEEE, 2008.
- [Dav75] L.S. Davis. A survey of edge detection techniques. *Computer Graphics and Image Processing*, 4(3) :248–270, 1975.
- [DC76] M.P. Do Carmo. *Differential geometry of curves and surfaces*, volume 3. Prentice-Hall Englewood Cliffs, NJ, 1976.
- [Dem02] D. Demigny. On optimal linear filtering for edge detection. *Image Processing, IEEE Transactions on*, 11(7) :728–737, 2002.
- [Der87] R. Deriche. Using Canny’s criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2) :167–187, 1987.

- [Der92] R. Deriche. Recursively implementing the gaussian and its derivatives. In *International Conference On Image Processing. A longer version is INRIA Research Report RR-1893*, pages 263–267, 1992.
- [DF95] R. Deriche and O. Faugeras. Les edp en traitement des images et vision par ordinateur. *Rapports de recherche-INRIA numéro 2697*, 1995.
- [DZ86] S. Di Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics and Image Processing*, 33(1) :116–125, 1986.
- [EL99] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1033–1038, 1999.
- [EMED08] S. El Mejdani, R. Egli, and F. Dubeau. Old and new straight-line detectors : Description and comparison. *Pattern Recognition*, 41(6) :1845–1866, 2008.
- [FA91] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13 :891–906, 1991.
- [FLW02] R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range compression. *Association for Computing Machinery's Transactions on Graphics*, 21(3) :249–256, 2002.
- [GM85] S. Geman and D.E. McClure. Bayesian image analysis : an application to single photon emission tomography. In *Proceedings of the Statistical Computing Section*, page 12. American Statistical Association., 1985.
- [GM96] G. Guy and G. Medioni. Inferring global perceptual contours from local features. *International Journal of Computer Vision*, 20(1) :113–133, 1996.
- [Gre90] P.J. Green. Bayesian reconstructions from emission tomography data using a modified em algorithm. *IEEE Transactions on Medical Imaging*, 9(1) :84–93, 1990.
- [GSvdW02] J.M. Geusebroek, A. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. *European Conference on Computer Vision*, pages 99–112, 2002.
- [Har05] R.M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5) :786–804, 2005.
- [HH93] L. Herault and R. Horaud. Figure-ground discrimination : A combinatorial optimization approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9) :899–914, 1993.
- [HM95] R. Horaud and Olivier Monga. *Vision par ordinateur : outils fondamentaux*. Traité des nouvelles technologies. Série informatique. Hermès, 1995.
- [Hou62] P.V.C. Hough. Method and means for recognizing complex patterns, 1962. US Patent 3,069,654.
- [HS88] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, page 50. Manchester, UK, 1988.

- [HSZ87] R.M. Haralick, S.R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (4) :532–550, 1987.
- [JU04] M. Jacob and M. Unser. Design of steerable filters for feature detection using canny-like criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8) :1007–1019, 2004.
- [Jul81] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802) :91–97, 1981.
- [KDA97] P. Kornprobst, R. Deriche, and G. Aubert. Nonlinear operators in image restoration. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 325–331, 1997.
- [KHEK76] M. Kuwahara, K. Hachimura, S. Eiho, and M. Kinoshita. Processing of ri-angiocardigraphic images. *Digital Processing of Biomedical Images*, pages 187–203, 1976.
- [Kir71] R. Kirsch. Computer determination of the constituent structure of biological images. *Computer Biomedical Research*, 4 :315–328, 1971.
- [KJB96] K. Karu, A.K. Jain, and R.M. Bolle. Is there any texture in the image? *Pattern Recognition*, 29(9) :1437–1446, 1996.
- [Koe84] J.J. Koenderink. The structure of images. *Biological cybernetics*, 50(5) :363–370, 1984.
- [KPP] N. Krüger, M. Pöttsch, and G. Peters. Principles of cortical processing applied to and motivated by artificial object recognition. *Information Theory and the Brain*, pages 223–228.
- [KQ04] C. Kirbas and F. Quek. A review of vessel extraction techniques and algorithms. *ACM Computing Surveys*, 36(2) :81–121, 2004.
- [KR82] L. Kitchen and A. Rosenfeld. Gray-level corner detection. *Pattern Recognition Letters*, 1(2) :95–102, 1982.
- [KvdWHR07] D. Knossow, J. van de Weijer, R. Horaud, and R. Ronfard. Articulated-Body Tracking Through Anisotropic Edge Detection. *Dynamical Vision*, pages 86–99, 2007.
- [Lac04] C. Lacoste. Extraction de réseaux linéiques à partir d’images satellitaires et aériennes par processus ponctuels marqués. *PhD Thesis, University of Nice Sophia Antipolis*, 2004.
- [LDA<sup>+</sup>97] L. Lucido, R. Deriche, L. Alvarez, V. Rigaud, et al. Sur quelques schémas numériques de résolution d’équations aux dérivées partielles pour le traitement d’images. 1997.
- [Lin93] T. Lindeberg. *Scale-space theory in computer vision*. Springer, 1993.
- [Lin98] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2) :117–154, 1998.
- [LM01] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1) :29–44, 2001.

- [LML<sup>+</sup>00] I. Laptev, H. Mayer, T. Lindeberg, W. Eckstein, C. Steger, and A. Baumgartner. Automatic extraction of roads from aerial images based on scale space and snakes. *Machine Vision and Applications*, 12(1) :23–31, 2000.
- [Low84] D.G. Lowe. Perceptual organization and visual recognition. *PhD Thesis, Stanford University*, 1984.
- [LTD05] O. Laligant, F. Truchetet, and A. Dupasquier. Edge enhancement by local deconvolution. *Pattern recognition*, 38(5) :661–672, 2005.
- [LTM07] O. Laligant, F. Truchetet, and F. Meriaudeau. Regularization preserving localization of close edges. *Signal Processing Letters, IEEE*, 14(3) :185–188, 2007.
- [MAM95] O. Monga, N. Armande, and P. Montesinos. Thin nets and crest lines : Application to satellite data and medical images. In *IEEE International Conference on Image Processing*, volume 2, pages 468–471, 1995.
- [Mat75] G. Matheron. *Random sets and integral geometry*, volume 9. Wiley New York, 1975.
- [MBP<sup>+</sup>09] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *IEEE International Conference on Computer Vision*, pages 2272–2279, 2009.
- [MC09] P. Montesinos and G. Claude. Finding people in internet images. In *Proceedings of the 2009 11th IEEE International Symposium on Multimedia*, pages 159–164. IEEE Computer Society, 2009.
- [MCS<sup>+</sup>10] B. Magnier, F. Comby, O. Strauss, J. Triboulet, and C. Demonceaux. Highly specific pose estimation with a catadioptric omnidirectional camera. In *IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 229–233. IEEE, 2010.
- [MD97a] P. Montesinos and S. Dattenny. Sub-pixel accuracy using recursive filtering. In *Scandinavian conference on image analysis*, pages 523–530, 1997.
- [MD97b] P. Montesinos and S. Dattenny. Sub-pixel accuracy using recursive filtering. In *Scandinavian conference on image analysis*, pages 523–530, 1997.
- [MDA07] M. Mühlich, T. Dahmen, and T. Aach. Design of multi-steerable filters and their application for the detection of corners and junctions. In *IEEE International Conference on Image Processing*, volume 4, pages IV–33. IEEE, 2007.
- [MDM] B. Magnier, D. Diep, and P. Montesinos. Perceptual curve extraction. In *IEEE 10th IVMS (Image, Video, and Multidimensional Signal Processing Technical Committee) on "Perception and Visual Signal Analysis" Workshop*, pages 93–98.
- [Men03] J. B. Mena. State of the art on automatic road extraction for gis update : a novel classification. *Pattern Recognition Letters*, 24(16) :3037–3058, 2003.

- [MFA11] M. Mühlich, D. Friedrich, and T. Aach. Design and Implementation of Multi-Steerable Matched Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.
- [MGD98] P. Montesinos, V. Gouet, and R. Deriche. Differential invariants for color images. In *Fourteenth IEEE International Conference on Pattern Recognition, 1998. Proceedings.*, volume 1, pages 838–840, 1998.
- [MH80] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167) :187, 1980.
- [MM10] P. Montesinos and B. Magnier. A New Perceptual Edge Detector in Color Images. In *Advances Concepts for Intelligent Vision Systems*, pages 209–220, 2010.
- [MMD11a] B. Magnier, P. Montesinos, and D. Diep. Fast Anisotropic Edge Detection Using Gamma Correction in Color Images. In *IEEE 7th International Symposium on Image and Signal Processing and Analysis*, 2011.
- [MMD11b] B. Magnier, P. Montesinos, and D. Diep. Ridge and valley junctions extraction. In *International Conference on Image Processing, Computer Vision and Pattern Recognition*, 2011.
- [MMD11c] B. Magnier, P. Montesinos, and D. Diep. Ridges and valleys detection in images using difference of rotating half smoothing filters. In *Advances Concepts for Intelligent Vision Systems*, pages 261–272. Springer, 2011.
- [MMD11d] B. Magnier, P. Montesinos, and D. Diep. Texture Removal by Pixel Classification using a Rotating Filter. In *IEEE 36th International Conference on Acoustics, Speech and Signal Processing*, pages 1097–1100, 2011.
- [MMD11e] B. Magnier, P. Montesinos, and D. Diep. Texture Removal in Color Images by Anisotropic Diffusion. In *International Conference on Computer Vision Theory and Applications*, pages 40–50, 2011.
- [MMD12a] B. Magnier, P. Montesinos, and D. Diep. A New region-based PDE for Perceptual image restoration. In *International Conference on Computer Vision Theory and Applications*, 2012.
- [MMD12b] B. Magnier, P. Montesinos, and D. Diep. Texture suppression preserving edges by anisotropic diffusion. In *IEEE 37th International Conference on Acoustics, Speech and Signal Processing*, 2012.
- [Nag83] H.H. Nagel. Constraints for the estimation of displacement vector fields from image sequences. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, volume 2, pages 945–951. Citeseer, 1983.
- [NB80] R. Nevatia and K.R. Babu. Slinear feature extraction and description. *Computer graphics and Image Processing*, 13 :257–269, 1980.
- [NM79] M. Nagao and T. Matsuyama. Edge preserving smoothing. *Computer Graphics and Image Processing*, 9 :394–407, 1979.

- [OR90] S. Osher and L.I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis*, pages 919–940, 1990.
- [PD02] N. Paragios and R. Deriche. Geodesic active contours for supervised texture segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, 2002.
- [Per92] P. Perona. Steerable-scalable kernels for edge detection and junction analysis. *Image and Vision Computing*, 10(10) :663–672, 1992.
- [Per95] P. Perona. Deformable kernels for early vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(5) :488–499, 1995.
- [PKT09] S. Paris, P. Kornprobst, and J. Tumblin. *Bilateral Filtering*. Now Publishers Inc, 2009.
- [PM90] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 12 :629–639, 1990.
- [PMD12] J. L. Palomares, P. Montesinos, and D. Diep. A New Affine Invariant Method for Image Matching. In *IEEE SPIE Conference on 3D Image Processing (3DIP) and Applications*, 2012.
- [Poy96] C.A. Poynton. *A technical introduction to digital video*. John Wiley & Sons, Inc. New York, NY, USA, 1996.
- [Pre70] J. M. S. Prewitt. *Object enhancement and extraction*. Picture Processing and Psychopictorics, B. Lipkin and A. Rosenfeld. Eds. Academic Press, New York, 1970.
- [Rob65] L. G. Roberts. *Machine perception of three-dimensional solids*. Optical and Electro-Optical Information Processing, J. T. Tippett et al. Eds. Cambridge, MA : MIT Press, 1965.
- [Rob77] S. Robinson. Edge detection by compass gradient masks. *Computer Graphics Image Processing*, 6 :492–501, 1977.
- [ROF92] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D : Nonlinear Phenomena*, 60(1-4) :259–268, 1992.
- [SB97] S.M. Smith and J.M. Brady. SUSAN A new approach to low level image processing. *International Journal of Computer Vision*, 23(1) :45–78, 1997.
- [SC92] J. Shen and S. Castan. An optimal linear operator for step edge detection. *CVGIP : Graphical Models and Image Processing*, 54(2) :112–133, 1992.
- [SF96] E.P. Simoncelli and H. Farid. Steerable wedge filters for local orientation analysis. *IEEE Transactions on Image Processing*, 5(9) :1377–1382, 1996.
- [SMB00] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of computer vision*, 37(2) :151–172, 2000.

- [Sob70] I.E. Sobel. Camera Models and Machine Perception. *PhD Thesis, Stanford University*, 1970.
- [SR96] G. Sapiro and DL Ringach. Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Transactions on Image Processing*, 5(11) :1582–1586, 1996.
- [Ste98] C. Steger. An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2) :113–125, 1998.
- [SU88] A. Sha’ashua and S. Ullman. Structural Saliency : The Detection of Globally Salient Structures Using Locally Connected Network. In *International Conference on Computer Vision*, pages 321–327, 1988.
- [TD02] D. Tschumperlé and R. Deriche. Diffusion pde’s on vector-valued images. *IEEE Signal Processing Magazine*, 19(5) :16–25, 2002.
- [TD05] D. Tschumperlé and R. Deriche. Vector-valued image regularization with pdes : A common framework for different applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 506–517, 2005.
- [Ter04] R. Terebes. Diffusion directionnelle. applications à la restauration et à l’amelioration d’images de documents anciens. *PhD Thesis, University of University Bordeaux I and Technical University of Cluj Napoca*, 2004.
- [Tik63] A.N. Tikhonov. Regularization of incorrectly posed problems. In *Soviet Math. Dokl*, volume 4, pages 1624–1627, 1963.
- [TM98] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *IEEE International Conference on Computer Vision*, pages 839–846, 1998.
- [TM08] T. Tuytelaars and K. Mikolajczyk. A survey on local invariant features. *Foundations and Trends in Computer Graphics and Vision*, (3), pages 177–280, 2008.
- [TP86] V. Torre and T.A. Poggio. On edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2) :147–163, 1986.
- [Tra06] T.T.H. Tran. Etude de lignes d’intérêt naturelles pour la représentation d’objets en vision par ordinateur. *PhD Thesis, Institut National Polytechnique de Grenoble*, 2006.
- [Tsc06] D. Tschumperlé. Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE’s. *International Journal of Computer Vision*, 68(1) :65–82, 2006.
- [UTY95] M. Unser, P. Thevenaz, and L. Yaroslavsky. Convolution-based interpolation for fast, high-quality rotation of images. *IEEE Transactions on Image Processing*, 4(10) :1371–1381, 1995.
- [WBSS04] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment : From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4) :600–612, 2004.

- 
- [Wei] J. Weickert. Anisotropic diffusion in image processing. 1998. *Teubner-Verlag, Stuttgart, Germany*.
- [Wei99] J. Weickert. Coherence-enhancing diffusion of colour images. *Image and Vision Computing*, 17(3-4) :201–212, 1999.
- [Wit83] A.P. Witkin. Scale-space filtering. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 1019–1021, 1983.
- [YDS01] W. Yu, K. Daniilidis, and G. Sommer. Approximate orientation steerability based on angular gaussians. *IEEE Transactions on Image Processing*, 10(2) :193–205, 2001.
- [ZBSA06] J. Zhou, W.F. Bischof, and A. Sanchez-Azofeifa. Extracting lines in noisy image using directional information. In *18th IEEE International Conference on Pattern Recognition*, volume 2, pages 215–218, 2006.
- [Zio91] D. Ziou. Line detection using an optimal IIR filter. In *Pattern Recognition*, pages 465–478, 1991.
- [ZT98] D. Ziou and S. Tabbone. Edge detection techniques : an overview. *International Journal on Pattern Recognition and Image Analysis*, 8(4) :537–559, 1998.