



HAL
open science

Constance de largeur et désocclusion dans les images digitales

Emmanuel Villéger

► **To cite this version:**

Emmanuel Villéger. Constance de largeur et désocclusion dans les images digitales. Mathématiques [math]. Université Nice Sophia Antipolis, 2005. Français. NNT : . tel-00011229

HAL Id: tel-00011229

<https://theses.hal.science/tel-00011229>

Submitted on 17 Dec 2005

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

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

UNIVERSITÉ DE NICE-SOPHIA ANTIPOLIS – UFR SCIENCES

École doctorale sciences fondamentales et appliquées

THÈSE

Présentée pour obtenir le titre de

Docteur en SCIENCES
de l'Université de Nice-Sophia Antipolis
Spécialité : *Mathématiques*

par

Emmanuel VILLÉGER

Constance de largeur et désocclusion dans les images digitales

Soutenue publiquement le 6 décembre 2005 devant le jury composé de :

Jacques	Blum	<i>Président</i>
Christine	Graffigne	<i>Rapporteur</i>
Vicent	Caselles	<i>Rapporteur</i>
Gilles	Aubert	<i>Directeur</i>
Laure	Blanc-Féraud	<i>Co-directrice</i>
Lionel	Moisan	<i>Examineur</i>

à 10h30 au laboratoire Jean-Alexandre Dieudonné

Thèse dirigée par G. Aubert et L. Blanc-Féraud et préparée à l'INRIA Sophia Antipolis, projet Ariana

Remerciements

Je tiens tout d'abord à remercier Jacques Blum qui m'a fait l'honneur de présider le jury de ma thèse. Son éternelle bonne humeur est agréable et appréciée. Je tiens également à remercier Christine Graffigne et Vicent Caselles qui ont accepté d'être rapporteurs et par conséquent de lire ce manuscrit en entier.

Je remercie évidemment Gilles Aubert et Laure Blanc-Féraud qui m'ont accueilli au laboratoire J.A. Dieudonné et à l'INRIA et qui ont lu plusieurs fois ce manuscrit.

Mais je tiens aussi à remercier Jean-Michel Morel et Lionel Moisan qui m'ont proposé le sujet de la constance de largeur qui est le point de départ de tout ce travail. Les longues et fréquentes discussions de l'époque ont permis de mettre au point l'algorithme de la première partie. Lionel m'a bien aidé pour mettre au point mes premiers programmes avec MegaWave2. C'est avec Megawave2 que les algorithmes de la première partie ont été programmés et j'espère que Megawave2 va continuer à se développer à l'avenir.

Comme j'évoque ma première année de recherche passée au CMLA, j'en profite pour remercier en plus de Jean-Michel et Lionel, toute l'équipe qui y était en même temps que moi notamment Lorenzo, Morgan, Julie, Benjamin, Agnès, André, Saïd, Pascal, Nicolas, Véronique, etc. J'en oublie sûrement et j'en suis désolé, je les remercie également.

Je veux aussi remercier les équipes au sein desquelles je suis arrivé ensuite. Celle du projet Ariana, dont beaucoup sont maintenant partis, Jean-François, Mathias, Caroline, Caroline, Oscar, Marie et Guillaume et les autres, sans oublier Josiane Zérubia, Xavier, Corinne et Ian. Je remercie aussi toute la partie administration de l'INRIA et du laboratoire I3S qui m'a permis de travailler dans de bonnes conditions pendant ces trois années. Je remercie aussi les amis et collègues de Dieudonné, encore une fois Jean-François et Caroline, et aussi Didier, Maëlle, Alexis, Guillaume, et les autres et aussi Janine, Isabelle, Heike, Jean-Marc et Michel, etc.

Je souhaite vraiment remercier tous ceux que je n'ai pas cités, après quatre ans pour le CMLA et après un an et demi d'enseignement à plein temps en lycée pour les autres, il m'est difficile de me rappeler de tout le monde ici, mais je les remercie chaleureusement aussi.

Je tiens également à remercier la fondation des Treilles. J'ai pu passer une semaine dans le Var pour travailler avec Jean-Michel, Lionel et Agnès dans un cadre magnifique. Nourrit superbement et logé dans une maison individuelle avec une grande salle pour travailler. Ce fut une semaine inoubliable. Je remercie également le GdR ISIS qui m'a permis d'aller travailler une semaine à Barcelone avec Vicent Caselles. Une deuxième semaine formidable durant laquelle j'ai pu beaucoup avancer sur la deuxième partie de mon travail.

Ces deux semaines sont très importantes pour moi. Pendant la première j'ai

réussi à mettre au point la première version du programme pour la constance de largeur. Et durant la deuxième semaine, j'ai enfin eu des résultats encourageant avec mon programme sur la désocclusion.

Et par-dessus tout je tiens à remercier ma femme qui m'a soutenu tout au long de ma thèse et surtout pendant le moment très délicat de la rédaction. Sa participation active me fut très précieuse, sans elle ce manuscrit n'existerait sûrement pas, merci.

Table des matières

Table des matières	v
Table des figures	vii
Liste des tableaux	xi
Introduction	1
I Constance de largeur	5
1 Théorie de la Gestalt et vision par ordinateur	7
1.1 L'école gestaltiste	7
1.1.1 Objets physiques et objets perceptifs	7
1.1.2 Les mécanismes de la perception	8
1.1.3 Et la constance de largeur ?	15
1.2 État de l'art sur la détection de lignes et de routes	19
1.2.1 Couplage de détecteurs de bord	19
1.2.2 Regroupement en lignes	19
1.2.3 Prise en compte du biais	20
1.2.4 Amincissement	20
1.2.5 Arbre et tests actifs	21
1.2.6 Chaîne de Markov	22
1.3 Détection <i>a contrario</i>	22
2 Modèles de probabilité sur les courbes d'une image digitale	25
2.1 Distance entre courbes	26
2.1.1 Distance de Hausdorff	27
2.1.2 Pseudo-distance	28
2.1.3 Probabilité	33
2.1.4 Probabilité de parallélisme de deux courbes	36
2.2 Modèle de courbes polygonales	37

2.2.1	Modèle	37
2.2.2	Probabilité de changement de direction	38
2.2.3	Extension du voisinage à $\mathbb{R}^2 \times \mathcal{S}^1$	42
2.3	Probabilités de présence et de sortie	44
2.3.1	Probabilité de présence	44
2.3.2	Propagation de la probabilité de présence	45
2.3.3	Probabilité de sortie	46
3	Calcul de la probabilité	49
3.1	Passage au discret	49
3.1.1	Discrétisation du domaine	49
3.1.2	Propagation discrète	51
3.1.3	Calibration des paramètres	59
3.2	Méthode de Monte-Carlo	67
3.2.1	Estimateur	67
3.2.2	Calibration des paramètres	69
3.3	Équation aux dérivées partielles	69
3.3.1	Obtention d'une EDP	69
3.3.2	Discussion par rapport à γ	78
3.3.3	Résolution numérique	82
3.3.4	Calibration des paramètres	84
4	Analyse et utilisation de la probabilité	91
4.1	Comparaison	91
4.1.1	Longueur	91
4.1.2	Épaisseur	94
4.1.3	Courbure	96
4.2	Vers la détection	99
II	Désocclusion	105
5	État de l'art	107
5.1	Introduction	107
5.2	Désocclusion basée sur les lignes de niveau	110
5.3	Méthode inspirée de la restauration de peinture	115
5.4	Interpolation conjointe d'un champ de vecteurs et des niveaux de gris	116
5.5	Diffusion dirigée par courbure	117
5.6	Méthode inspirée de la dynamique des fluides	118
5.7	Conclusion	120

6	Présentation de notre modèle	121
6.1	Principe de l'algorithme	121
6.1.1	Orientation des lignes de niveau	122
6.2	Incertitude sur l'orientation d'un vecteur	124
6.3	Champ de probabilité associé	126
6.4	Désocclusion du champ de probabilité	128
6.5	Reconstruction de l'image	131
6.6	Système de deux équations	132
7	Mise en œuvre numérique	135
7.1	Schéma de résolution	135
7.2	Équation de désocclusion	138
7.2.1	Le schéma pour la désocclusion et sa stabilité	138
7.2.2	Mise en œuvre de la désocclusion du champ de probabilités	144
7.3	Équation de reconstruction	145
7.3.1	Le schéma de la reconstruction et sa stabilité	146
7.3.2	Mise en œuvre de la reconstruction de l'image	151
7.4	Couplage des deux équations	158
7.4.1	Initialisation	158
7.4.2	Résultats	159
	Conclusion	163
	III Annexes	165
	A Diffusions	167
	B Détecteur optimal de bord de Deriche	175
	Bibliographie	179

Table des figures

1.1	Triangle de Kanizsa.	9
1.2	Camouflage d'un hexagone.	10
1.3	Illusion de Sander.	10
1.4	Illusion de Zöllner.	11
1.5	Un chandelier ou bien deux visages face à face ?	12
1.6	Vingt-cinq points noirs.	13
1.7	Vingt-cinq points regroupés par colonnes.	14
1.8	Un arc de cercle, un segment et une ligne brisée.	15
1.9	Des cercles ou des sortes de x ?	16
1.10	Aucun lien apparent entre courbes.	16
1.11	Un hexagone et un cube.	17
1.12	Bandes blanches et bande noire.	18
1.13	Profil idéal d'une ligne.	20
2.1	Distance entre deux courbes.	28
2.2	Appariement de points entre deux courbes.	29
2.3	Distance de Hausdorff et dilatations.	30
2.4	Défaut de la distance de Hausdorff.	31
2.5	Distance de Hausdorff quotient.	32
2.6	Tube autour d'une courbe.	33
2.7	Un voisinage $V(\mathcal{C}, \varepsilon)$	34
2.8	Construction d'un voisinage.	35
2.9	Plusieurs voisinages autour d'une même courbe.	36
2.10	Une courbe polygonale.	38
2.11	Construction d'une courbe polygonale.	39
2.12	Probabilités de changement de direction, lois de Laplace.	41
2.13	Probabilité de changement de direction, loi empirique.	43
2.14	Le voisinage $V(\mathcal{C}, \varepsilon)$ de largeur ε d'une courbe \mathcal{C}	44
2.15	Extension de \mathbb{R}^2 à $\mathbb{R}^2 \times \mathcal{S}^1$	45
2.16	Extrémités des voisinages.	46
2.17	Cercle d'intégration.	47
2.18	Sortie du voisinage.	48

3.1	Discrétisations.	50
3.2	Suivi d'une direction discrète.	52
3.3	Un voisinage à différentes échelles.	60
3.4	Influence de r sur le modèle $\tilde{\mathcal{M}}_r$	61
3.5	Comparaison loi de Laplace/loi empirique.	63
3.6	Influence de K sur le modèle $\tilde{\mathcal{M}}_r$	64
3.7	Influence de s sur le modèle $\tilde{\mathcal{M}}_r$	65
3.8	Influence de λ sur le modèle $\tilde{\mathcal{M}}_r$	66
3.9	L'étape initiale de l'estimation de Monte-Carlo.	68
3.10	Une seconde itération de l'estimation de Monte-Carlo.	70
3.11	Influence de N_0 sur le modèle \mathcal{M}_r	71
3.12	Influence de r sur le modèle \mathcal{M}_r	72
3.13	Influence de K sur le modèle \mathcal{M}_r	73
3.14	Influence de s sur le modèle \mathcal{M}_r	74
3.15	Influence de λ sur le modèle \mathcal{M}_r	75
3.16	Courbes générées par l'estimateur.	76
3.17	EDP de transport seul.	80
3.18	EDP de transport et de diffusion.	81
3.19	Influence de α sur le modèle \mathcal{M}_0	85
3.20	Influence de K sur le modèle \mathcal{M}_0	86
3.21	Influence de ε sur le modèle \mathcal{M}_0	87
3.22	Influence de η sur le modèle \mathcal{M}_0	88
3.23	Influence de s sur le modèle \mathcal{M}_0	89
4.1	Influence de la longueur.	93
4.2	Influence de l'épaisseur.	94
4.3	Angle d'observation de Γ_f depuis Γ_d	95
4.4	Influence de l'épaisseur en échelle linéaire.	95
4.5	Influence de la courbure.	97
4.6	Courbure des courbes polygonales.	98
4.7	Dix courbes parallèles.	101
5.1	Projection d'une scène	108
5.2	Image de phase en interférométrie radar RSO	108
5.3	Un trou dans une image	109
5.4	Jonctions en T	111
5.5	Désocclusion manuelle	112
5.6	Reconstruction rectiligne sur un disque	115
5.7	Reconstruction d'une barre	119
6.1	Lignes de niveau et changement de contraste	123
6.2	Orientation d'une ligne de niveau	124
6.3	Gradient d'une image	129
6.4	Distributions de probabilité	130

7.1	Les étapes de l'algorithme	137
7.2	Images synthétiques	146
7.3	Effet de tourbillon	147
7.4	Résultat de désocclusion	148
7.5	Images synthétiques	153
7.6	Champs d'orientations moyennes	154
7.7	Résultats des images synthétiques	155
7.8	Résultat en séparant les niveaux	156
7.9	Image de phare	157
7.10	Image de phare	157
7.11	Image de phare	158
7.12	Initialisations	159
7.13	Résultats	161
7.14	Étapes sur la courbe floue	162
A.1	Exemple de diffusions	171
A.2	Diffusions de type Perona-Malik	172
A.3	Histogramme de la norme du gradient	173
B.1	Filtre de Deriche	177

Liste des tableaux

3.1	Valeurs par défaut pour le modèle $\tilde{\mathcal{M}}_r$.	59
4.1	Paramètres par défaut lors des expériences.	92
4.2	Premier pas vers la détection.	102
A.1	Termes de diffusion	169

Introduction

Les algorithmes de traitement d'images ont souvent un nombre important de paramètres. Le choix de paramètres optimaux pose souvent problème. Heureusement dans certains cas l'influence des paramètres est connue et permet d'optimiser le résultat. Ainsi le filtre de Canny et Deriche [10] pour la détection de contours ne possède qu'un ou deux paramètres qui ont une signification claire. L'un des paramètres est un seuil à partir duquel la réponse est considérée être un contour. Le second paramètre concerne la précision de la localisation de ces contours. Comme l'influence de ces paramètres est connue, il est possible de les régler relativement facilement. Mais avec déjà trois paramètres, voire plus le choix de la valeur de chacun d'entre eux pose des problèmes importants. Certaines techniques existent cependant pour sélectionner des paramètres optimaux.

À l'inverse la voie proposée par Desolneux, Moisan et Morel dans [12] est de réduire le nombre de paramètres. En effet ils proposent un paradigme réduisant le nombre de paramètres à un. Dans [12], ce paradigme est appliqué à la détection d'alignements dans les images. Dans sa thèse [11], Desolneux applique également ce paradigme à la détection des modes dans un histogramme ainsi qu'à la détection des contours dans une image.

Le paradigme qu'ils utilisent est le principe de Helmholtz. Ce dernier stipule que ce qui est remarquable ce sont les grandes déviations par rapport à un fond uniforme. Par exemple pour les alignements, dans un cadre classique de maximisation *a posteriori*, il faut définir pour chaque emplacement possible d'un alignement une probabilité d'avoir un alignement. La réponse du détecteur serait donc d'autant plus grande si dans l'image ce qui se trouve à cet emplacement ressemble à un alignement. Au contraire avec le principe de Helmholtz, nous supposons que l'image est une sorte de bruit blanc. Pour les alignements les auteurs de [12] supposent qu'en chaque point l'orientation des lignes de niveau suit une loi uniforme et que celles-ci sont indépendantes. Ce qu'ils cherchent alors dans l'image ne sont pas les emplacements où il est plus probable d'avoir un alignement, mais les endroits où on observe un alignement des points qui est très peu probable par rapport à leur hypothèse de répartition uniforme et indépendante des orientations.

Ils appellent ce modèle un modèle *a contrario*. Ce qu'ils cherchent ce sont les phénomènes "rares," les apparitions d'au moins k points alignés avec la même

orientation. Il cherchent dans l'image les endroits où leur hypothèse d'uniformité est la plus mise en défaut. Ils utilisent la probabilité qu'au moins k points soient alignés avec la même orientation. Ce faisant un paramètre est introduit qui est relatif à l'image, la longueur minimale requise pour que ces points forment un segment. Cependant les auteurs ne s'intéressent pas à la probabilité d'avoir ce nombre de points alignés. Ce qui les intéresse est le nombre de fausses détections. En effet la probabilité d'un événement a beau être faible, en faisant suffisamment de tests aléatoires, tant que la probabilité n'est pas nulle il est toujours possible que cela se produise "par hasard." Ce nombre de fausses alarmes est lié à l'espérance d'observer dans une image avec des orientations uniformes au moins k points alignés avec la même orientation. Il semble naturel de vouloir moins d'une fausse alarme sur l'image. Pour cela les auteurs veulent que l'espérance (non pas la probabilité) du nombre d'observations d'au moins k points alignés avec la même orientation soit plus petite que 1. Comme les orientations sont indépendantes pour majorer l'espérance il suffit de multiplier cette probabilité par le nombre de tests effectués *a priori*, sans connaître la valeur de k .

Le sujet de [12], la détection des alignements n'a pas été choisi par hasard par les auteurs. Les alignements correspondent à une des Gestalts énumérées dans [22]. Dans [22] Kanizsa explique la théorie de la Gestalt. Dans cette théorie de la perception, lorsque nous voyons des images, certains points de l'image sont automatiquement regroupés par notre cerveau. Ce phénomène de regroupement et le résultat s'appelle une Gestalt. Ce phénomène de regroupement est très fort et important, c'est ce phénomène qui est à l'origine de plusieurs illusions d'optique où des regroupements se font naturellement alors qu'en fait ce qui est regroupé ne vient pas du même objet (ou n'a pas la même cause). Dans [22] sont décrites plusieurs Gestalts. Les points d'une image sont regroupés en fonction de différents critères : même orientation dans le cas des alignements, même couleur, même forme, etc. Ces regroupements constituent les Gestalts. Dans la théorie de la Gestalt le tout est plus fort que la somme des parties. En effet le fait que ces points se regroupent avec les points voisins qui ont la même orientation donne un sens plus fort à ces points que juste plusieurs points séparés. Le fait qu'ils forment une Gestalt les associe fortement. La réunion de ces points permet de comprendre plus facilement l'image. À leur tour, ces Gestalts peuvent présenter des caractères en commun pour former ensemble des sortes de super-Gestalts. Nous voyons ainsi se dessiner un algorithme pyramidal qui permettrait d'interpréter facilement une image : détecter les Gestalts présentes dans l'image, puis détecter à partir de ces Gestalts des super-Gestalts, et ainsi de suite avec pour but final une analyse de la scène à la manière de celle effectuée par le cerveau humain. C'est avec ce but final que les auteurs de [12] se sont attaqués à la détection des alignements. Ils ont ensuite utilisé le même paradigme utilisant le principe de Helmholtz pour d'autres Gestalts dans [13, 14], les modes des histogrammes, les contours.

Une des Gestalts citée par Kanizsa dans [22] est la constance de largeur. La constance de largeur apparaît lorsque dans une image est présente une structure que nous qualifierions de longiligne, par exemple un ruban qui flotte au vent. Un autre exemple peut être une ligne tracée au crayon sur du papier. Pour utiliser le principe de Helmholtz, nous devons pouvoir calculer la probabilité que deux courbes restent parallèles. Pour cela nous sommes amenés à proposer une modélisation de courbes. Cette modélisation consiste à fabriquer des courbes en mettant bout à bout des segments de longueur r . À partir de cette modélisation nous proposons un modèle limite en faisant tendre r vers 0. Nous présentons également un estimateur de la probabilité pour deux courbes de rester parallèles.

Le modèle limite obtenu aboutit à une équation aux dérivées partielles qui contient à la fois un terme de diffusion et un terme de transport. Cette équation est du même type que les équations utilisées pour résoudre le problème de la désocclusion. Ce problème apparaît lorsqu'il y a un trou dans une image. La désocclusion consiste à remplir le trou de l'image de manière indétectable à partir des informations présentes dans l'image autour du trou. En partant de l'équation aux dérivées partielles trouvée dans la première partie nous proposons un modèle pour résoudre le problème de la désocclusion. La différence de notre méthode par rapport aux modèles existants est le fait que nous prenons en compte l'incertitude de la mesure du gradient d'une image connue uniquement sur une grille discrète de points. En chaque point de la grille nous calculons le gradient de l'image. Nous avons donc une valeur approchée de l'orientation du gradient (qui correspond à l'orientation des lignes de niveau) et de sa norme. Cependant au lieu d'utiliser directement ces valeurs pour compléter l'image dans le trou en suivant les lignes de niveau, nous introduisons des probabilités d'orientation. En chaque point nous introduisons une probabilité, sa moyenne est l'orientation approchée trouvée et son écart-type dépend de la précision du calcul. Nous savons que l'orientation calculée est d'autant plus précise que la norme du gradient est élevée, et nous utilisons cela pour les écarts-types. Ceci nous conduit en fait à deux équations aux dérivées partielles : une première équation pour compléter ce champ de probabilités à l'intérieur du trou et une deuxième équation pour reconstruire l'image dans le trou à partir de ce champ de probabilité. Cette méthode de désocclusion n'est pas adaptée aux images de texture mais aux images contenant plutôt de l'information géométrique.

Ce manuscrit s'organise en deux parties. La première partie est consacrée à la modélisation de la Gestalt constance de largeur. Dans la deuxième partie nous nous intéressons au problème de la désocclusion.

La première partie commence par une introduction à la théorie de la Gestalt. Le chapitre 1 est consacré à la présentation de la théorie de la Gestalt et à la vision par ordinateur. Nous étudions aussi le lien qui existe entre ces deux théories à travers le modèle *a contrario* introduit par Desolneux, Moisan et Morel dans [12]. Nous étudions également le problème de la constance de

largeur. Nous rappelons plusieurs méthodes existantes qui se rapprochent le plus du problème qui nous intéresse de la Gestalt constance de largeur. Le chapitre 2 est consacré à la présentation de notre modèle de génération de courbes à partir de segments de longueur r . Ce modèle nous sert dans le chapitre 3 pour calculer la probabilité pour deux courbes de rester parallèles. Nous présentons une méthode de calcul découlant de notre modèle de génération de courbes. Nous présentons également un estimateur qui constitue une méthode bien plus rapide que le calcul direct. Enfin nous étudions le modèle limite obtenu en faisant tendre r vers 0 dans notre modélisation et nous obtenons alors un modèle d'équation aux dérivées partielles. Le dernier chapitre de cette première partie, le chapitre 4 contient l'étude des probabilités obtenues par les trois méthodes sur différentes courbes. L'ensemble de ce travail doit rendre possible la détection automatique des constances de largeur dans les images.

La deuxième partie traite du problème de la désocclusion. Dans le chapitre 5, nous avons rassemblé les descriptions des méthodes de désocclusion pour les images de type géométrique. Dans le chapitre 6 nous présentons notre propre modèle de désocclusion basé sur une probabilité d'orientation des lignes de niveau de l'image prenant en compte les incertitudes du calcul de l'orientation de ces lignes de niveau. Au chapitre 7 nous introduisons nos schémas numériques pour la résolution du système de deux EDPs que nous obtenons avec notre modèle. Nous montrons également quelques résultats numériques dans ce chapitre.

Pour finir nous présentons les conclusions et les perspectives de ce travail puis deux annexes. Une première annexe, l'annexe A est consacrée à quelques termes de diffusion fréquemment rencontrés pour la régularisation d'images par équations aux dérivées partielles. La deuxième annexe, l'annexe B présente un filtre pour un calcul utilisant le gradient d'une image, le filtre de Deriche présenté dans [10].

Première partie
Constance de largeur

Chapitre 1

Théorie de la Gestalt et vision par ordinateur

1.1 L'école gestaltiste

Dans la grammaire du voir [22], Kanizsa explique les règles qui régissent la perception. Kanizsa fait partie de l'école gestaltiste qui a commencé à Berlin à la fin du XIX^{ème} siècle et qui a continué en Italie par la suite, puis aux États-Unis d'Amérique. Le fondement du Gestaltisme est que lorsque nous voyons un carré, notre cerveau ne voit pas simplement les quatre côtés, ou bien l'ensemble des points lumineux qui forment ses quatre côtés mais qu'il perçoit quelque chose en plus. La perception est modifiée par l'objet perçu. Une phrase simplificatrice courante est de dire que le tout masque les parties. Ceci signifie que le tout n'est pas la simple somme des parties, il rajoute quelque chose en plus. Ce tout est appelé une *Gestalt*. Depuis le début du XX^{ème} siècle l'école gestaltiste s'est intéressée à comment se forment ces Gestalts. Le premier chapitre du livre de Kanizsa fait le point sur les règles de formation des Gestalts.

1.1.1 Objets physiques et objets perceptifs

Le processus de perception peut sembler simple de prime abord. Des photons se réfléchissent sur les objets. Ces photons passent par notre œil pour arriver sur les cônes et les bâtonnets de notre rétine. Ensuite cette information est transmise par les nerfs à notre cerveau. Celui-ci nous dit alors ce que nous voyons. Il n'y a qu'à analyser chacune de ces étapes pour savoir ce que nous percevons.

Cependant nous n'avons pas du tout accès aux étapes intermédiaires. Ce que nous avons à notre disposition n'est que le produit fini de cette longue chaîne. Ceci pose un premier problème : il faut bien distinguer les objets physiques—les objets réellement présents—et les objets perceptifs—les objets

que nous voyons. En effet jusqu'à la fin du XIX^{ème} siècle l'idée commune était de penser que les briques de base de la perception étaient les stimuli visuels arrivant sur la rétine. Les Gestaltistes ont remis en cause cette vision en montrant que nous percevons en fait autre chose que cette masse de stimuli et que des objets sont perçus au-delà de ces stimuli. Par exemple dans le triangle de Kanizsa, figure 1.1 nous voyons le bord du triangle blanc même là où il n'est physiquement pas présent. En fait le triangle blanc tout entier n'est pas physiquement présent dans cette image. Les objets physiquement présents dans cette image sont seulement trois angles et trois portions de disque. Et inversement nous pouvons être véritablement aveugle et ne pas parvenir à voir un objet physiquement présent. Par exemple dans les deux images de la figure 1.2 se trouve le même hexagone, cependant dans l'image 1.2(b), celui-ci n'est pas visible, nous ne le percevons pas ou alors difficilement et de manière fugitive. En effet il est caché par le reste de la figure qui forme un tout et l'hexagone n'y est plus perceptible.

De la même manière les illusions optico-géométriques mettent également en défaut une correspondance entre réalité et perception. Ainsi l'illusion de Sander (figure 1.3) consiste en deux parallélogrammes, les diagonales de ces parallélogrammes ont physiquement la même longueur alors que nous les percevons de longueurs différentes. Dans la figure 1.4 nous avons reproduit l'illusion de Zöllner, nous ne voyons pas parallèles les diagonales du carré. Pourtant, physiquement, elles le sont.

Ces illusions optico-géométriques sont souvent considérées comme des distractions. Vis-à-vis du processus de perception elles ne seraient que des exceptions, des aberrations, elles le mettraient en défaut et l'empêcheraient de bien faire son travail. Kanizsa rejette cette idée, le processus régissant la perception ne peut souffrir aucune exception. Par conséquent les illusions sont des perceptions comme les autres. Loin d'être des exceptions à une loi générale, elles permettent de mettre en avant ces rouages. **Les illusions optico-géométriques permettent d'isoler les rouages du processus de perception.**

1.1.2 Les mécanismes de la perception

Ce qui nous permet de percevoir les objets physiques sont les photons qu'ils réfléchissent. Tous ces photons étant indépendants les objets physiques sont complètement dématérialisés, en passant de la réalité physique à la réalité perceptive. Comme nous percevons néanmoins des objets, nous les voyons, il faut bien que nous reconstruisions des objets perceptifs à partir des images qui nous parviennent. Rubin a étudié la formation des objets perceptifs. Il regarde pour cela comment une certaine partie d'une image prend le rôle de figure au-dessus du fond. Il utilise la *méthode phénoménologique* de l'école gestaltiste, c'est-à-dire qu'il présente à l'observateur des expériences aux paramètres bien déterminés. Ensuite il demande à l'observateur une description la plus

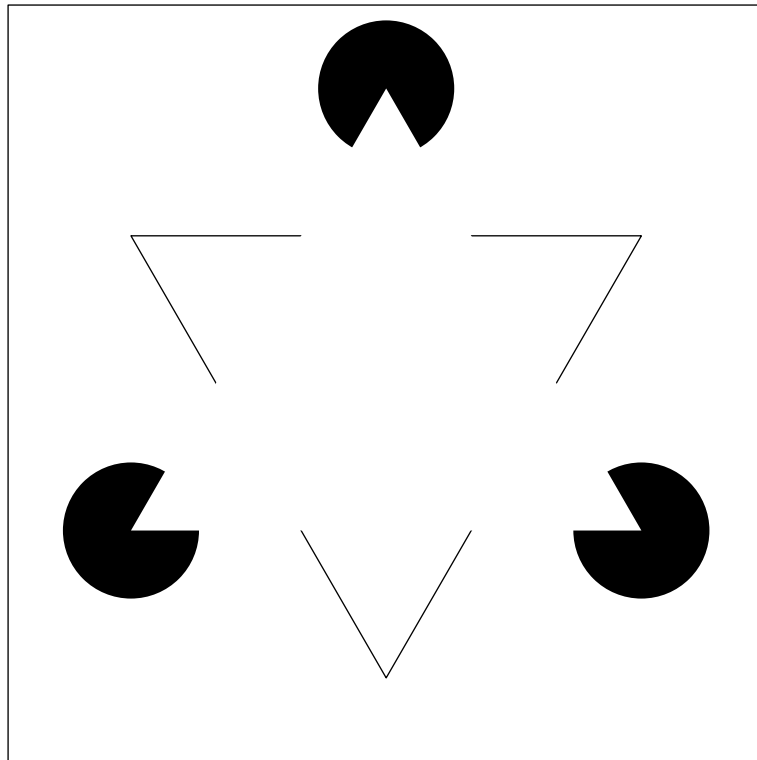


FIG. 1.1 – Le triangle de Kanizsa. Dans cette figure nous percevons un triangle blanc au-dessus d'un autre triangle et de trois disques. Pourtant ne sont physiquement présents que trois angles et trois portions de disque. De plus le triangle blanc est plus blanc que le fond qui est pourtant de la même couleur. Nous “voyons” les bords du triangle blanc aux endroits où pourtant il n'y a rien sur la figure. Il s'agit là de bord suggestifs suggérés par l'alignement des portions de disque et des angles.

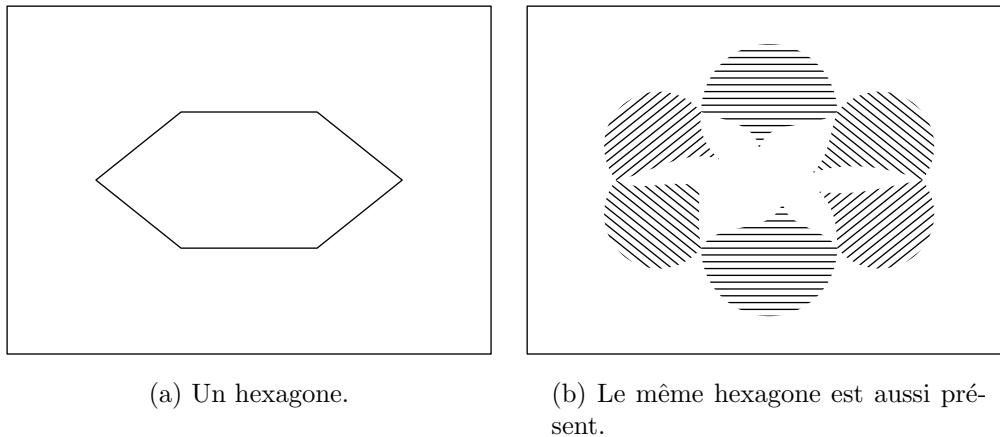


FIG. 1.2 – Camouflage d’un hexagone. L’hexagone de la figure de gauche est également présent dans la figure de droite. Cependant les segments parallèles à ses côtés le masquent totalement. Pour arriver à voir l’hexagone il faut suivre son contour et même dans ce cas il est difficile de le voir longtemps.

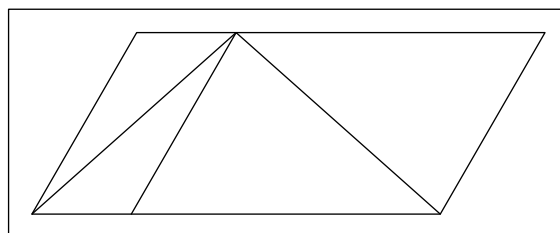


FIG. 1.3 – Illusion de Sander. Sur cette figure sont représentés deux parallélogrammes et pour chacun d’eux une de leurs diagonales. Les deux diagonales sont de même longueur et pourtant la diagonale du parallélogramme de gauche semble plus petite que celle du parallélogramme de droite.

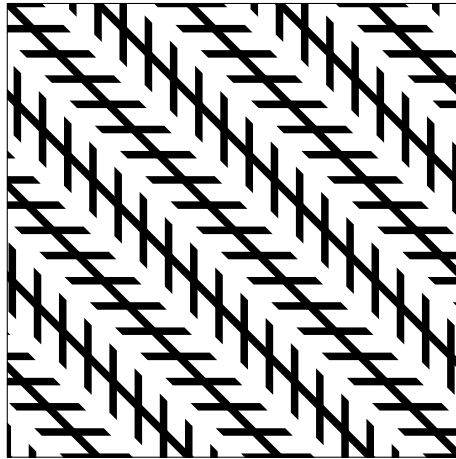


FIG. 1.4 – Illusion de Zöllner. Les lignes les plus longues sont toutes parallèles à la diagonale du carré. Cependant les petits segments qui leur sont superposés changent notre perception de leur direction de sorte que ces lignes parallèles nous semblent divergentes ou convergentes mais certainement pas parallèles.

immédiate et authentique possible, sans interprétation ou réflexion de sa part.

Dans l'image de la figure 1.5 par exemple, le plus souvent est perçu un espèce de chandelier noir avec des sortes de feuilles, c'est-à-dire que la partie noire de l'image prend le rôle de figure, tandis que la partie blanche joue le rôle de fond. Cependant il est également possible de voir deux visages en face à face sur fond noir. Il est intéressant de constater que nous ne pouvons pas percevoir simultanément le chandelier et les deux visages, alors même qu'ils sont tous constamment présents dans l'image. En effet la figure et le fond sont perceptivement très distincts. La figure est présente, elle occupe de l'espace, c'est un objet perceptif alors que le fond est vide et s'étend par derrière la figure. Les contours de la figure sont *unilatéraux*, ils ne délimitent que la figure et pas le fond qui lui n'a pas de forme propre.

Rubin a commencé à dégager des règles qui peuvent permettre de savoir comment se fait l'articulation figure/fond. Il a mis en avant la taille, la relation topologique (si l'un inclus l'autre), et les types de bords (convexité, concavité). Par la suite Wertheimer [40] a cherché à isoler plus systématiquement ces règles, Kanizsa les résume comme suit.

Les règles de formation des unités perceptives sont : la proximité, la ressemblance, la continuité de direction, la fermeture, la prégnance (*Prägnanz*) et l'expérience passée. Nous appelons ces règles des *Gestalts*.

La proximité. Dans la figure 1.6 se trouvent 25 points noirs. Il y a un très grand nombre de configurations possibles. Il est possible d'en voir certaines

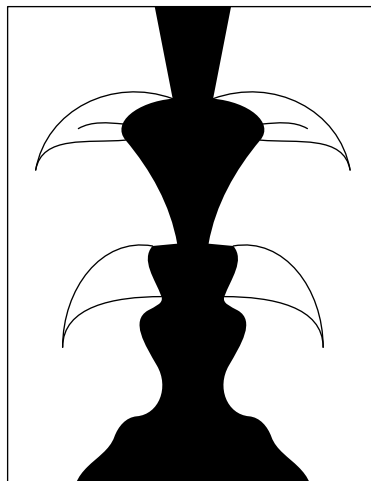


FIG. 1.5 – Un chandelier ou bien deux visages face à face ? En général lors de la première vision de ce dessin la partie noire forme une figure et nous voyons une sorte de chandelier avec des appendices en forme de feuille au dessus d'un fond blanc. Cependant il est également possible de faire prendre à la partie blanche le rôle de la figure au-dessus d'un fond noir. Dans ce cas là nous voyons deux visages face à face. Cependant ces deux interprétations ne peuvent pas cohabiter, le fond passe derrière la figure, de sorte qu'il n'a pas de forme propre. Ainsi lorsque la partie blanche prend le rôle de la figure, la partie noire n'a pas de forme et ne saurait être vue comme un chandelier, et vice-versa.

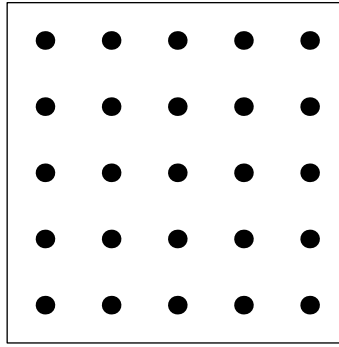


FIG. 1.6 – Vingt-cinq points régulièrement répartis. Aucun des vingt-cinq points ne se distingue des autres. Il existe un nombre impressionnant de façon de regrouper vingt-cinq points. Il est possible de mentalement les regrouper par ligne ou par colonne par exemple, cependant il faut faire un effort pour maintenir ce regroupement. En fait aucun regroupement n'est mis en avant dans ce dessin.

mais aucune n'est vraiment stable. Par contre dans la figure 1.7(a) des regroupements verticaux se forment spontanément et restent stables. Cette figure illustre la Gestalt de proximité. En effet dans la figure 1.7(a) les deuxième et quatrième colonnes ont simplement été décalées respectivement à gauche et à droite. La proximité entre deux colonnes de gauche et celle des deux colonnes de droite, nous les fait les assembler et nous regroupons alors les points verticalement.

La ressemblance. Dans la figure 1.7(b) les objets se regroupent une fois encore verticalement. Cette fois-ci c'est la Gestalt de ressemblance qui est illustrée par cette figure. En effet les deuxième et quatrième colonnes sont formées de triangles alors que les trois autres sont formées de points. Le regroupement des triangles se fait spontanément et les regroupements ici encore se font verticalement.

La continuité de direction. Dans la figure 1.8 c'est la règle de continuité de direction qui nous fait voir un segment, une ligne et un arc de cercle. En effet tous ces objets perceptifs ne sont pas physiquement présent, il n'y a que des points. De plus au croisement entre la ligne brisée et le segment, nous prolongeons en ligne droite ce que nous voyons.

La fermeture. Nous préférons également les regroupements formant des objets fermés plutôt que des objets ouverts. C'est le cas dans la figure 1.9. Cette règle de fermeture met en avant des cercles plutôt que des sortes de x . Pour-

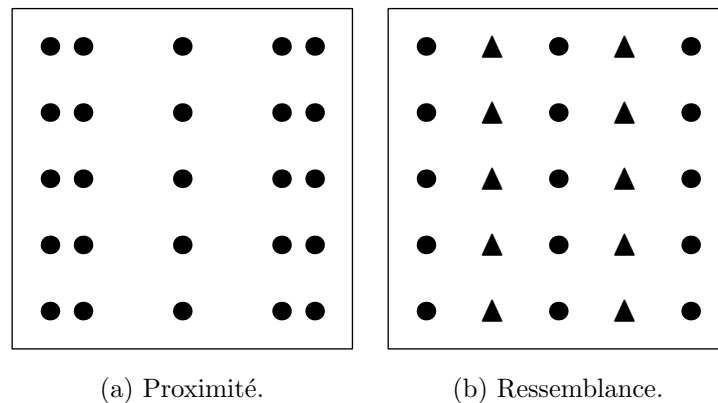


FIG. 1.7 – Toujours vingt-cinq points mais cette fois-ci ils se regroupent par colonnes. En effet dans le dessin de gauche la deuxième colonne a été rapprochée de la première colonne et la quatrième de la cinquième. Cette fois-ci les points se regroupent spontanément verticalement. Dans le dessin de droite dans la deuxième et la quatrième colonne les disques ont été remplacés par des triangles. Là encore ce changement provoque un regroupement vertical des disques ensemble et des triangles ensemble. Dans le premier cas la proximité des colonnes provoque un regroupement vertical, c'est le regroupement par **proximité** et dans le second cas la similarité des formes d'une même colonne provoque le même phénomène, il s'agit du regroupement par **ressemblance**.

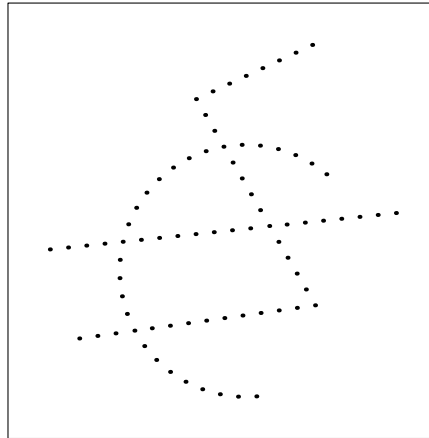


FIG. 1.8 – Un arc de cercle, un segment et une ligne brisée. Sur ce dessin sont dessinés des points noirs régulièrement espacés suivant des courbes. Ces points se regroupent spontanément pour former un arc de cercle, un segment et une ligne brisée. Ceci est dû au fait que lors de chaque croisement nous avons “choisi” une continuation parmi celles possibles de façon à continuer les courbes dans la même direction, c’est-à-dire que les regroupements ont eu lieu en utilisant la règle de **continuité de direction**.

tant l’espace entre deux branches de demi cercle est plus important que l’espace entre les demi-cercles.

L’expérience passée. L’expérience passée joue également un rôle. Son influence est moindre que les autres règles mais elle est réelle. Par exemple dans la figure 1.10 les courbes semblent indépendantes et sans rapports cependant après une rotation de $\pi/2$ radians le mot THE apparaît clairement, ceci n’est possible qu’en connaissant l’alphabet.

1.1.3 Et la constance de largeur ?

La Gestalt de prégnance regroupe pour Kanizsa le fait que la figure obtenue doit obéir à certains critères de simplicité, d’ordre, de symétrie, de régularité, de cohérence structurelle et de caractère unitaire. Ceci pour préciser le sens du principe de la prégnance aussi appelé bonne Gestalt et qui sert parfois d’explication hâtive. Comme exemple de ce principe de prégnance, dans la figure 1.11(a) nous percevons un hexagone et dans la figure 1.11(b) un cube alors même que les deux figures peuvent être considérées comme des vues en perspective d’un cube. Seulement la figure 1.11(a) est assez régulière en deux dimensions pour être interprétée comme un hexagone, sans relief, alors que la figure 1.11(b) en deux dimensions n’a pas de régularité. Cependant en étant

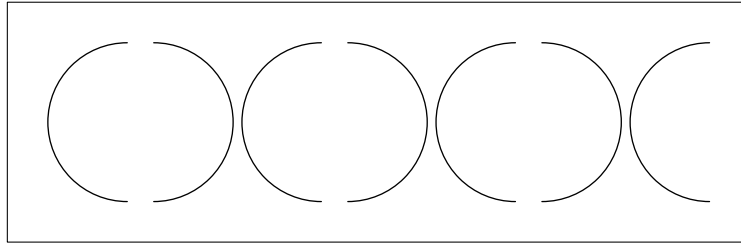


FIG. 1.9 – Des cercles ou des sortes de x ? Sept demi-cercles sont dessinés. Même si les demi-cercles qui se tournent le dos sont plus proches que les demi-cercles qui se font face, le regroupement spontané est de regrouper les demi-cercles en face à face de manière à former des cercles. Le regroupement spontané privilégie les cercles, des formes fermées à des sortes de x qui sont des formes ouvertes. C'est le regroupement par **fermeture**.

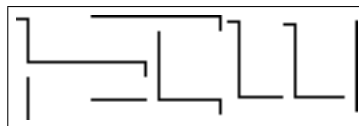


FIG. 1.10 – Il n'y a aucun lien apparent entre les courbes du dessin. Cependant si nous penchons le dessin sur la droite, le mot "THE" apparaît écrit verticalement. Ce faisant certaines des courbes sont regroupées comme faisant partie d'une même lettre. Les lettres ne sont pas innées, elles dépendent de la culture, elles sont apprises. Ainsi ce regroupement s'effectue en fonction de **l'expérience passée** de l'observateur.

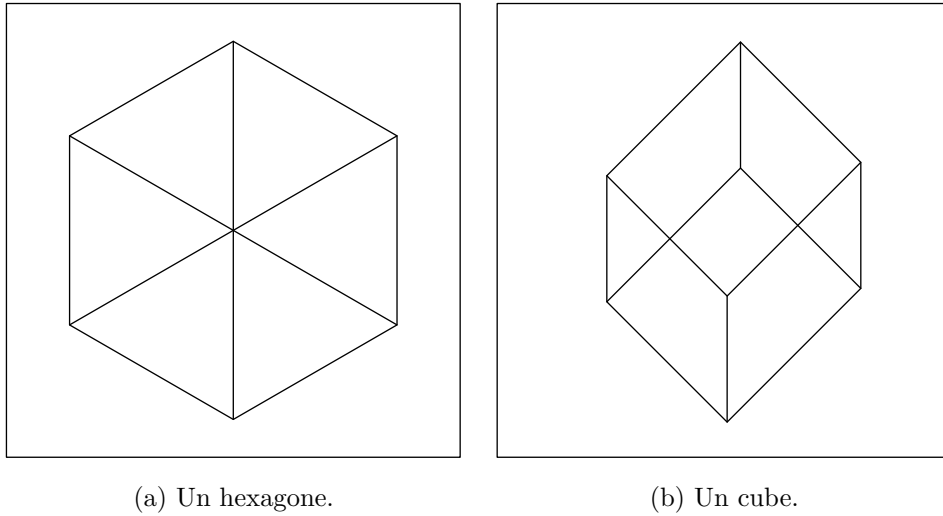
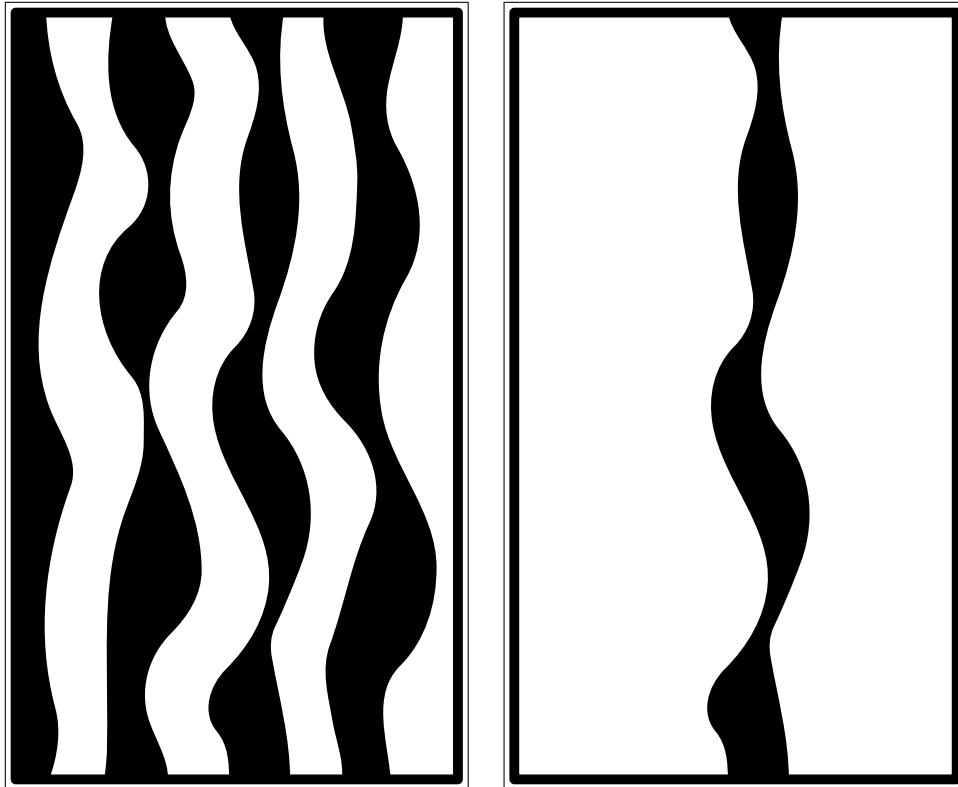


FIG. 1.11 – Un hexagone et un cube, deux projections possibles d'un même cube. En effet sur le dessin de gauche nous voyons un hexagone et sur le dessin de droite nous voyons une vue en perspective d'un cube. Pourtant l'hexagone du dessin de gauche pourrait être vu comme une vue en perspective particulière d'un cube. L'hexagone en deux dimensions a assez de régularité par lui-même ce qui n'est pas le cas de la figure de droite vue comme un dessin en deux dimensions. Pour le dessin de droite son interprétation comme la projection d'un cube lui apporte de la régularité qu'il n'a pas en deux dimensions. Cette interprétation obéit à la règle de **prégnance**, la recherche des figures les plus régulières.

considérée comme une vue en perspective d'un cube, la figure obtient toute la régularité du cube.

Les figures 1.12(a) et 1.12(b) mettent en avant un autre aspect du principe de prégnance. Dans la figure 1.12(a) les bandes blanches ont des bords à peu près parallèles et sont de largeur constante. Elles prennent préférentiellement le rôle de figure au-dessus d'un fond noir. Plutôt que les bandes noires aux bords irréguliers au-dessus d'un fond blanc. Par contre dans la figure 1.12(b) où il ne reste qu'une des bandes noires c'est elle qui est la figure sur un fond blanc. C'est cet aspect du principe de prégnance que nous appelons la Gestalt *constance de largeur*.



(a) Des bandes blanches.

(b) Une bande noire.

FIG. 1.12 – Bandes blanches et bande noire. Sur le dessin de gauche nous voyons plutôt des bandes blanches sur fond noir que l'inverse. En effet les bords des bandes blanches sont plus réguliers que ceux des bandes noires. Ceux-ci sont plus parallèles et donc la largeur des bandes blanches varie moins que celle des bandes noires. C'est un regroupement par **constance de largeur**. Ceci est relatif car dans le dessin de droite où ne subsiste qu'une des bandes noires du dessin de gauche, nous voyons une bande noire sur fond blanc bien que ses bords soient aussi peu réguliers que sur le dessin de gauche.

1.2 État de l'art sur la détection de lignes et de routes

C'est la Gestalt de constance de largeur que nous étudions. Dans l'histoire de la vision par ordinateur les objets de largeur constante qui ont été le plus étudiés sont d'une part les lignes et d'autre part les routes. Nous commençons donc par présenter ce qui a été fait dans ces domaines.

1.2.1 Couplage de détecteurs de bord

Koller et al. dans [24] partent du constat que le détecteur de ligne de Canny répond également aux bords. Pour résoudre ce problème ils proposent de découpler la détection de ligne en deux détecteurs de bord. Ils prennent alors le minimum des deux détecteurs de sorte que s'il ne s'agit que d'un bord, le résultat soit nul et celui-ci ne génère pas de réponse.

Pour détecter une ligne ils utilisent deux détecteurs de bord décentrés et symétriques l'un par rapport à l'autre. Soient B_g et B_d ces deux détecteurs. Ils utilisent en fait des dérivées de la gaussienne. L'utilisation de ces dérivées fait alors penser à une approximation par différence finie de la dérivée seconde de la gaussienne qui est l'opérateur proposé par Canny. Cette séparation en deux détecteurs de bord est aussi à la base de l'algorithme proposé par Subirana-Vilanova et Kung dans [36].

Le détecteur unidimensionnel proposé par Koller et al. est

$$\begin{cases} L_s = \min((B_g * f)^+, (B_d * f)^+), \\ B_g = -G'_s(x + s) \text{ et } B_d(x) = G'_s(x - s), \end{cases} \quad (1.1)$$

où $f^+ = \max(f, 0)$ est la partie positive de la fonction f . En bidimensionnel ils calculent une direction de ligne en chaque point à partir de la dérivée seconde lissée de l'image. Ensuite ils utilisent leur détecteur dans la direction orthogonale. Ils suppriment alors les non-maxima et ils font cela pour toutes les échelles.

Avec leur algorithme il est possible d'avoir une idée de la largeur des lignes en regardant l'échelle donnant la plus forte réponse. Cependant là encore le résultat n'est pas un ensemble de lignes.

1.2.2 Regroupement en lignes

Dans [19], Glazer commence classiquement par détecter les points de ligne avec la dérivée seconde de l'image. Les points de ligne sont les points tels que la deuxième valeur propre de la hessienne de l'image (la courbure secondaire) est au dessus d'un seuil et que dans cette direction l'image passe par un maximum au point considéré. Ensuite avec une interpolation quadratique l'auteur trouve

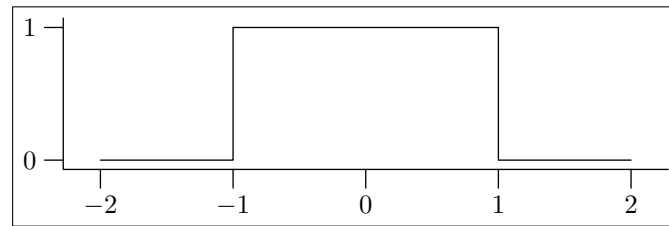


FIG. 1.13 – Profil idéal d'une ligne.

une approximation plus précise du point de ligne en coordonnées continues à l'intérieur du pixel. Par la suite les points de lignes sont regroupé en chaînes. Pour cela l'auteur utilise la direction des points de ligne et leur proximité. Enfin ces chaînes sont regroupées entre elles pour former des lignes avec des critères similaires. De découplage en deux temps du rassemblement des points en des lignes permet à l'auteur de traiter plus facilement les endroits où plusieurs lignes se croisent.

Cette fois-ci l'algorithme donne bien un résultat sous forme de lignes.

1.2.3 Prise en compte du biais

Steger dans [34] a une approche de géométrie différentielle et d'espace échelle. Il étudie précisément l'influence du facteur d'échelle sur l'épaisseur détectée. De plus il ne regarde pas seulement le cas d'une fonction créneau comme sur la figure 1.13 mais il utilise une fonction qui n'a pas la même valeur des deux côtés du créneau. Il étudie là aussi l'influence de cette dissymétrie sur la localisation de la ligne. Le même genre de technique que dans [19] est utilisé pour trouver les points de lignes.

Une fois les points de ligne précisément localisés, Steger cherche les bords de la ligne avec des algorithmes classiques dans la direction orthogonale à la ligne. Ceci lui permet d'avoir une largeur pour la ligne. Il utilise alors son étude sur les biais pour évaluer plus précisément la localisation du centre de la ligne et sa largeur.

1.2.4 Amincissement

La méthode décrite par Treash et Amaratunga dans [37] part d'un algorithme de détection de bord. Cette détection est faite avec des masques locaux, chacun correspondant à une direction de bord. Ensuite il font un seuillage du résultat et les points isolés sont éliminés. Chaque voisinage d'un point de bord est examiné. Pour chaque configuration sur un voisinage 3×3 possible, les auteurs proposent une modification qui revient à amincir les lignes formées de points de bord.

Sur les points de bord restants les auteurs les relient entre-eux selon des critères de distance et d'adéquation entre les directions. Lors de cette phase les directions sont lissées entre les points chaînés. Une fois les points de bord reliés entre eux, Treash et Amaratunga fabriquent des dipôles en reliant entre eux des bords de directions opposées qui se font face.

1.2.5 Arbre et tests actifs

L'algorithme présenté par Geman et Jedynak dans [18] est un algorithme de détection de route semi-automatique. L'objectif de cet algorithme est, étant donnée une image satellite et un point de départ de route, de prolonger la détection de cette route à travers l'image. Pour cela les auteurs font des hypothèses sur les routes, ils supposent qu'elles sont de niveau de gris homogène, qu'elle ne font que quelques pixels de large et que leur courbure est faible.

Leur idée de départ est de mettre un arbre ternaire sur l'image, la racine de cet arbre est le point de départ. Les trois descendants de cette racine sont le point atteint en continuant dans la direction donnée au départ, le point adjacent en allant dans une direction un peu plus à gauche et le point adjacent en allant un peu plus à droite. Le reste de l'arbre suit le même principe. Ils s'intéressent à l'ensemble des parcours possibles de l'arbre \mathcal{X} . S'il est de profondeur L , leur nombre est 3^L . Un parcours X de cet arbre suit la route qu'ils cherchent à détecter. *A priori* ils ne le connaissent pas et chacun des parcours a la même probabilité d'être ce parcours X . Au départ donc

$$P(x = X) = 3^{-L}.$$

Ils élaborent des tests à pratiquer sur les branches de l'arbre ternaire. Ils établissent alors sur des images tests la probabilité des réponses possibles à ces tests en fonction de la position de la branche sur une route ou non. En faisant des hypothèses d'invariance spatiale et d'indépendance ils cherchent à maximiser *a posteriori* la probabilité qu'un parcours suive la route en connaissant les résultats des tests sur ses branches.

Pour accélérer ce processus ils fabriquent en fait plusieurs arbres ternaires. Pour cela ils décident de temps en temps, lorsqu'une branche est presque sûrement le long de la route de redéfinir cette branche comme racine d'un nouvel arbre. Ils arrivent ainsi à parcourir de grandes images en peu de temps.

Le problème de cette approche pour la détection de ligne est la forte limitation sur la courbure et sur la largeur. En effet cet algorithme est prévu pour détecter des routes sur un type d'image, donc à courbure faible et de largeur quelques pixels. De plus c'est un algorithme supervisé, il faut préciser un point de départ et l'algorithme détecte alors une route partant de ce point et une seule.

1.2.6 Chaîne de Markov

Stoica et al. dans [35] utilisent un modèle de processus ponctuel. Ils cherchent un réseau de route dans une image comme réalisation d'un processus ponctuel. Les points du processus sont des points d'un espace différent de celui de l'image. Chaque point correspond en fait à un rectangle sur l'image. Pour simuler ce processus ponctuel ils utilisent une chaîne de Markov de Monte-Carlo, c'est-à-dire qu'ils font différentes simulations, les unes à la suite des autres, chaque simulation ne dépendant que de la précédente. Pour mener à bien le passage d'une simulation à l'autre, ils utilisent la méthode de Metropolis-Hastings et Green [20]. Ceci consiste à choisir parmi une liste la modification à apporter à la simulation pour qu'elle soit plus proche de la réalité. Cette modification est acceptée en fonction d'une certaine probabilité.

La probabilité d'acceptation dépend de différents facteurs. Ces facteurs prennent en compte le nombre de rectangles libres, c'est-à-dire sans voisins proches dans la simulation, le nombre de rectangles qui en touchent un autre en une extrémité, en deux ou qui se croisent. Il y a aussi un critère pour favoriser d'abord les alignements, ensuite les croisements francs et donc sont pénalisés les rectangles plus ou moins parallèles. Et enfin sont favorisés les rectangles dont les pixels ont des valeurs proches.

1.3 Détection *a contrario*

De façon plutôt surprenante il n'y a pas eu d'interaction forte entre la théorie de la Gestalt et la vision par ordinateur. Le livre de David Marr [26] de 1982, fondateur pour la vision par ordinateur ne prend pas en compte à sa juste mesure la théorie de la Gestalt. De même les travaux de Wertheimer [40], Metzger [28] et Kanizsa [22] ne prennent pas en compte la théorie de l'information développée par Shannon [33] sur l'échantillonnage des images.

Depuis quelques années Desolneux, Moisan et Morel ont entrepris d'utiliser la théorie de la Gestalt en vision par ordinateur [12, 13, 14]. Le but de leur entreprise est d'utiliser les Gestalts de base de la théorie de la Gestalt comme briques élémentaires de l'analyse d'images. Ils ont besoin pour cela de concevoir des algorithmes pour détecter ces Gestalts partielles dans les images. Ensuite il leur faudra trouver un autre algorithme pour rassembler les résultats de toutes ces détections et analyser réellement l'image en ayant géré les conflits entre les différentes Gestalts [15]. Pour l'instant Desolneux, Moisan et Morel ont conçus des algorithmes pour détecter

- les alignements de direction sur un segment [12],
- les alignements de points [14],
- les lignes de niveau de contraste minimal élevé [13],
- les similarités par rapport à une quantité scalaire uniforme (niveau de gris, orientation, etc.) [14],

- les similarités par rapport à une quantité scalaire suivant une loi décroissante (longueur, aire, etc.) [14],
- les groupes de points [14].

Chacune de ces détections repose sur un modèle de détection *a contrario* fondé sur le principe d’Helmholtz. Selon ce principe nous percevons les structures qui s’éloignent le plus d’une formation aléatoire. Nous donnons ici au mot *aléatoire* plus que le simple fait d’obéir à une loi de probabilité, nous voulons aussi dire que cette loi est uniforme, son “uniformité” dépendant du contexte.

Desolneux, Moisan et Morel utilisent ce principe pour faire de la détection *a contrario*. Ils cherchent dans une image les structures qui mettent en défaut une hypothèse de répartition uniforme, c’est-à-dire les structures les moins probables selon cette hypothèse. Ils définissent une structure ε -significative comme une structure qui apparaît en moyenne sous cette hypothèse moins de ε fois sur une image.

DÉFINITION 1 :

Soit N_{conf} . le nombre de réalisations possibles d’une structure S . Le nombre de fausses alarmes d’une occurrence s de cette structure est

$$NFA(s) = N_{\text{conf}}.P(s). \quad (1.2)$$

La structure s est dite ε -significative si

$$NFA(s) \leq \varepsilon.$$

La probabilité $P(s)$ est à définir avec une loi de probabilité à l’uniformité adaptée au problème traité.

Pour la détection d’alignement de directions sur un segment [12], Desolneux, Moisan et Morel traitent l’événement “Au moins k points sur le segment s de longueur l sont alignés avec s avec une précision inférieure à p .” Leur hypothèse d’uniformité est que les directions des points sont toutes indépendantes et suivent la loi uniforme sur \mathcal{S}^1 , le cercle unité. Dans ce cas le nombre de configurations possibles est le nombre de segments dans une image. Si la taille de l’image est $N \times N$, il y en a $N^2 \times N^2$ car il y a N^2 possibilités pour chaque extrémité du segment. La probabilité d’avoir k points alignés parmi l avec la précision p suit une loi binomiale de paramètre p . Pour l’événement considéré le nombre de fausses alarmes est

$$NFA(k, l) = N^4 \left(\sum_{i=k}^l C_l^i p^i (1-p)^{l-i} \right). \quad (1.3)$$

Par contre dans certains cas donner une loi uniforme *a priori* n’est pas seulement peu aisé mais peut également être source d’erreur. C’est le cas pour la détection des lignes de niveau significatives. Desolneux, Moisan et Morel

dans [13] considèrent l'événement "Le contraste minimal le long de la ligne de niveau L de longueur l est au moins égal à μ ." Pour cet événement ils supposent que le contraste en chaque point est indépendant et suit une certaine loi à définir. Cette hypothèse d'indépendance est grossièrement fautive et c'est justement ce défaut d'indépendance qu'ils veulent mettre en défaut. Une fois définie $H(\mu)$, la probabilité que le contraste soit supérieur ou égal à μ en un point, le nombre de fausses alarmes est

$$NFA(\mu, l) = N_{\text{ln}} [H(\mu)]^l, \quad (1.4)$$

où N_{ln} est le nombre de lignes de niveau de l'image. La loi $H(\mu)$ est à définir sur \mathbb{R}^+ . S'ils choisissent une loi par avance, il y a un risque pour que les lignes de niveau soient significatives en mettant en défaut cette loi particulière plutôt que l'hypothèse d'indépendance du contraste le long de la ligne de niveau. Pour circonvier à ce problème les auteurs utilisent la distribution du contraste de l'image elle-même pour calculer la loi empirique du contraste de l'image.

Nous allons à notre tour adapter ce modèle de détection *a contrario* pour la détection de la constance de largeur.

Chapitre 2

Modèles de probabilité sur les courbes d'une image digitale

La figure 1.12(a) montre un exemple de constance de largeur. Chacune des bandes blanches que nous percevons est constituée à partir de deux courbes : une pour chaque bord. Pour détecter les constances de largeur dans les images nous allons donc prendre en compte des paires de courbes. Les courbes considérées seront des morceaux de ligne de niveau. Lorsque nous observons un couple de morceaux de lignes de niveau nous pouvons faire deux hypothèses :

H0 : Les deux courbes proviennent de deux objets différents de l'image. Dans ce cas les deux courbes ne sont pas liées causalement et nous pouvons supposer les deux courbes indépendantes.

H1 : Les deux courbes proviennent du même objet. Dans ce cas là les deux courbes ont une cause commune. Nous ne pouvons pas faire l'hypothèse d'indépendance.

Dans une approche Bayésienne classique nous calculerions un maximum *a posteriori* de la probabilité pour deux courbes d'être parallèles. Grenander, Mumford, S. Geman utilisent une approche Bayésienne en analyse d'image. Avec cette approche il faut avoir une probabilité *a priori* de chaque structure à détecter [5, 31]. Zhu applique ce genre de méthode pour générer des formes, des contours d'objets en prenant en compte les règles de regroupement de la théorie de la Gestalt [41].

La technique du maximum *a posteriori* (MAP) part de l'égalité

$$P(m|o) = \frac{P(m \cap o)}{P(o)} = \frac{P(o|m)P(m)}{P(o)}, \quad (2.1)$$

où o est l'objet observé et m l'interprétation, le modèle, cherché. Le but est de maximiser la probabilité du modèle m connaissant l'observation o en utilisant (2.1). Pour ce faire ils maximisent en fait $P(o|m)P(m)$ car o étant donné, il ne rentre pas dans les variables pour le maximum. En fait la variable est le

modèle m . Il faut donner une valeur à chacun des deux facteurs du produit. Le premier terme $P(o|m)$ est un terme d'attache aux données, c'est la probabilité d'observer l'image à partir du modèle m . Cela traduit par exemple le bruit du capteur de la caméra lors de l'acquisition d'une image digitale. Le deuxième terme $P(m)$ est beaucoup plus délicat, c'est la probabilité d'un modèle m *a priori*. Cette probabilité peut être par exemple apprise à partir d'une base de données ou bien être imposée pour favoriser certains modèles auquel cas il y a souvent beaucoup de paramètres associés.

Pour la constance de largeur cela voudrait dire qu'il nous faudrait un modèle *a priori* pour des couples de courbes. Il faut alors modéliser la dépendance de deux courbes. Nous n'allons pas aller dans cette direction, nous préférons utiliser le modèle *a contrario* de Desolneux, Moisan et Morel. Dans l'approche *a contrario* nous n'avons pas besoin d'un modèle aussi complexe que dans l'approche Bayésienne. Au lieu d'avoir un seuil d'acceptation pour l'hypothèse **H1** nous utilisons un seuil de rejet pour **H0**. Nous allons détecter comme formant une constance de largeur les courbes qui auront une faible probabilité en tant que couple de courbes indépendantes. Si cette probabilité est assez faible, elle traduit le fait que les deux courbes ont une cause commune et que nous sommes alors dans l'hypothèse **H1** et non dans **H0**.

2.1 Distance entre courbes

Presque toutes les méthodes de détection de ligne partagent les trois problématiques suivantes

1. la détection de bord,
2. l'appariement de bords opposés,
3. le chaînage de points pour former des lignes.

Le premier point concerne la détection de points dans l'image où se trouvent des bords. Le deuxième point concerne la mise en relation de deux bords opposés qui sont alors supposés être les deux bords d'une seule ligne. Et le dernier point concerne le passage de plusieurs points en des courbes. C'est-à-dire que nous voulons relier les points au centre des bords appariés pour former les lignes. Ces trois points ne sont pas abordés toujours de la même manière ni dans le même ordre.

Certaines méthodes abordent les deux premiers points simultanément en détectant directement un profil de ligne plutôt que deux profils de bords à appairer par la suite. La détection d'un seul profil nous paraît plus adaptée pour la détection des lignes peu épaisses. D'autres méthodes forment d'abord des lignes avec les points où se trouvent les bords puis ensuite appariement les lignes de bords opposées. Même si l'ordre et les méthodes diffèrent, ces trois problématiques se retrouvent dans la plupart des méthodes.

Pour éviter le problème de former des courbes en reliant des points entre eux nous utilisons directement des courbes déjà formées. Nous utilisons en fait des morceaux de lignes de niveau de l'image. En effet les lignes de niveau font partie des bords de l'image. Cependant il n'y a pas de correspondance entre les lignes de niveau et les bords de l'image. Il se peut que pour certains bords, il n'y ait pas de ligne de niveau qui le suive tout du long, dans ce cas il faut utiliser plusieurs morceaux de lignes de niveau pour suivre ce bord. De même les lignes de niveau se prolongent en général au-delà des bords. C'est pour cette raison que nous utilisons des morceaux de lignes de niveau plutôt que les lignes de niveau en entier. Dans l'article [13], Desolneux et al. considèrent significatif un morceau de ligne de niveau lorsque le contraste de l'image tout le long du morceau de ligne est assez élevé. Ce sont ces morceaux de lignes de niveau significatif que nous allons utiliser pour la détection de la constance de largeur.

Ainsi nous n'avons pas le problème de la liaison des points de bord en courbes de bord. De plus nous limitons également les tests à effectuer. Il ne reste qu'à apparier des courbes de bords opposées. Pour faire cet appariement nous cherchons à qualifier le "degré" de parallélisme entre deux courbes.

Il nous faut donc définir ce que sont deux courbes parallèles et un moyen de mesurer si nous sommes proches où loin d'un parallélisme. Nous voulons formaliser la propriété que la distance entre les deux courbes reste à peu près constante et nous devons trouver comment mesurer cette fluctuation autour d'une distance constante.

2.1.1 Distance de Hausdorff

La distance classique ne suffit pas pour caractériser un parallélisme. En effet si \mathcal{C} est une courbe et si x est un point alors la distance de x à \mathcal{C} est

$$d(x, \mathcal{C}) = \inf\{\|x - y\|, y \in \mathcal{C}\}.$$

Et si \mathcal{C}_1 et \mathcal{C}_2 sont deux courbes, leur "distance" (ce n'est pas une distance sur l'ensemble des parties de \mathbb{R}^2) est

$$d(\mathcal{C}_1, \mathcal{C}_2) = \inf\{\|x - y\|, x \in \mathcal{C}_1, y \in \mathcal{C}_2\}.$$

Dans cette définition, seules interviennent les parties les plus proches de \mathcal{C}_1 et \mathcal{C}_2 . Par exemple dans la figure 2.1, les courbes \mathcal{C}_1 et \mathcal{C}_2 ne sont pas parallèles alors que les courbes \mathcal{C}_2 et \mathcal{C}_3 sont parallèles. Par contre la distance entre les courbes \mathcal{C}_1 et \mathcal{C}_2 est plus petite que celle entre les courbes \mathcal{C}_2 et \mathcal{C}_3 .

C'est pour cela que nous avons besoin de connaître non pas la distance entre deux courbes, mais la variation de la distance entre deux points se déplaçant simultanément de façon appropriée sur chaque courbe. Nous voudrions pouvoir définir une espèce de distance "glissante" entre les courbes et ainsi étudier ses variations. Cependant apparier des points entre deux courbes n'est pas

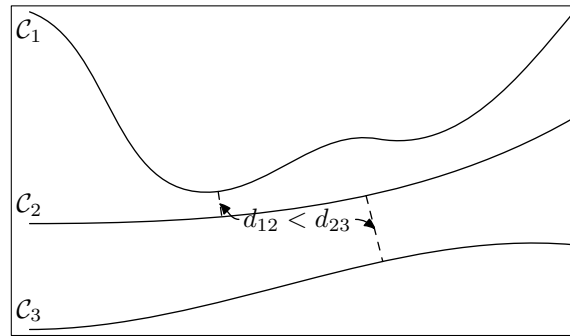


FIG. 2.1 – La distance entre deux courbes ne caractérise pas leur parallélisme. d_{12} est la distance entre les courbes \mathcal{C}_1 et \mathcal{C}_2 et d_{23} est la distance entre les courbes \mathcal{C}_2 et \mathcal{C}_3 . La courbe \mathcal{C}_1 est plus proche de \mathcal{C}_2 que ne l'est \mathcal{C}_3 mais les courbes \mathcal{C}_2 et \mathcal{C}_3 sont bien plus parallèles.

une tâche facile. Sur la figure 2.2 nous avons choisi des points régulièrement répartis sur les deux courbes \mathcal{C}_1 et \mathcal{C}_2 . Ces deux courbes sont approximativement parallèles mais l'appariement choisi ne met pas en avant cette propriété. Dans ce cas précis il est aisé de déterminer un appariement adapté, mais il est difficile de généraliser ce choix pour un couple arbitraire de courbes.

Plutôt que de continuer en utilisant la distance classique, nous allons travailler à partir de la distance de Hausdorff qui, elle, est une vraie distance sur l'ensemble des parties de \mathbb{R}^2 .

DÉFINITION 2 (Distance de Hausdorff) :

La distance de Hausdorff entre deux courbes \mathcal{C}_1 et \mathcal{C}_2 est

$$d_H(\mathcal{C}_1, \mathcal{C}_2) = \max\{\rho(\mathcal{C}_1, \mathcal{C}_2), \rho(\mathcal{C}_2, \mathcal{C}_1)\},$$

où si \mathcal{C} et \mathcal{C}' sont des courbes, $\rho(\mathcal{C}, \mathcal{C}') = \sup\{d(x, \mathcal{C}'), x \in \mathcal{C}\}$.

Comme cela est illustré sur la figure 2.3, $\rho(\mathcal{C}_1, \mathcal{C}_2)$ correspond à la dilatation nécessaire de \mathcal{C}_1 pour contenir \mathcal{C}_2 .

2.1.2 Pseudo-distance

À l'inverse de la distance classique, seules interviennent pour la distance de Hausdorff les parties les plus éloignées des courbes. Sur la figure 2.4(a) les deux courbes sont très parallèles alors que dans la figure 2.4(b), elles ne le sont pas. Cependant ces dernières sont plus proches avec la distance de Hausdorff que les premières. Pour prévenir ce problème nous allons définir une nouvelle distance en ne considérant les courbes qu'à une translation près.

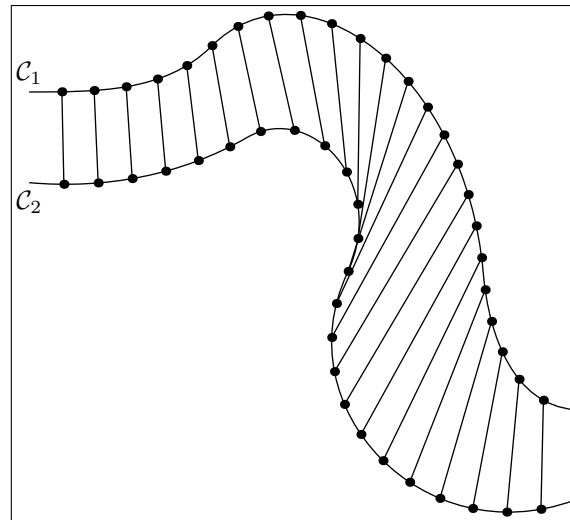
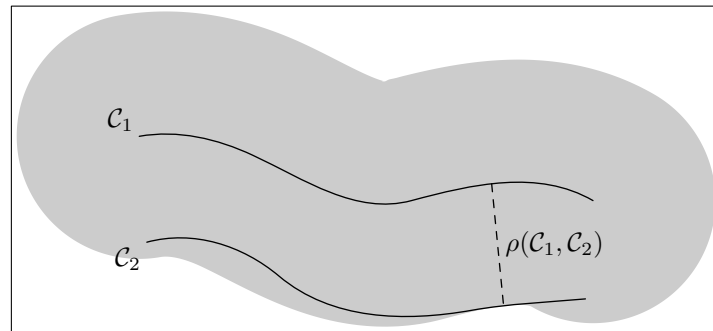
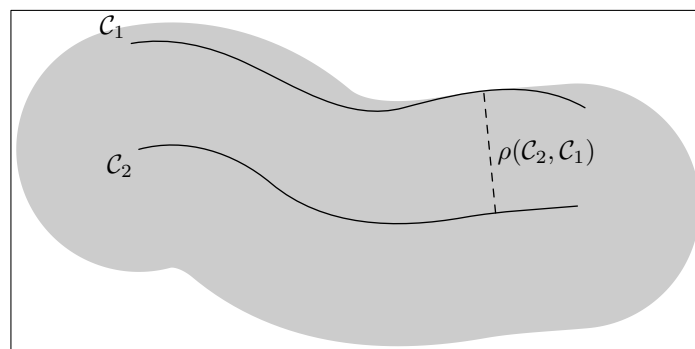


FIG. 2.2 – Appariement de points entre deux courbes. Pour définir une distance entre deux courbes qui suive leur parcours nous pouvons apparier deux à deux des points des deux courbes. Ainsi nous avons la distance entre chacun des deux points de ces paires comme repère pour connaître la variation de la distance entre les deux courbes. Cependant il n'est pas facile de trouver comment effectuer cet appariement. Par exemple sur ce dessin nous avons choisi des points régulièrement répartis sur les courbes. Ici les deux courbes sont presque parallèles et la variation de la distance entre ces points ne caractérise pas trop ce parallélisme. Dans ce cas précis nous pourrions définir un appariement qui conviendrait. Cependant l'enjeu est de trouver une manière d'apparier qui conviendrait dans tous les cas, même les plus exotiques.



(a) Dilatation de \mathcal{C}_1 .



(b) Dilatation de \mathcal{C}_2 .

FIG. 2.3 – Distance de Hausdorff et dilatations. La figure du haut montre les points réalisant la distance $\rho(\mathcal{C}_1, \mathcal{C}_2)$. Nous voyons aussi la dilatation de $\rho(\mathcal{C}_1, \mathcal{C}_2)$ de la courbe \mathcal{C}_1 qui contient effectivement la courbe \mathcal{C}_2 . Le dessin du bas montre l'inverse, la plus petite dilatation de la courbe \mathcal{C}_2 contenant \mathcal{C}_1 .

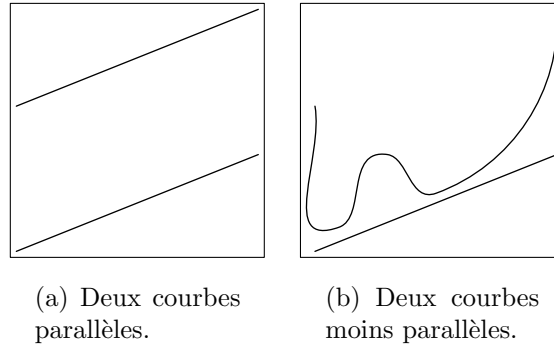


FIG. 2.4 – Défaut de la distance de Hausdorff. Les deux courbes du dessin de gauche sont parallèles, les deux courbes du dessin de droite sont bien moins parallèles. Pourtant chacune de ces paires de courbes sont à la même distance de Hausdorff.

Soit \mathcal{C} un ensemble de courbes. Soit \mathcal{T} l'ensemble des translations du plan. Nous considérons les courbes de $\mathcal{C}_{\mathcal{T}} = \mathcal{C}/\mathcal{T}$ l'ensemble des courbes quotienté par les classes d'équivalence définies par l'action de \mathcal{T} sur \mathcal{C} . La distance de Hausdorff devient par passage au quotient

DÉFINITION 3 :

Soient $\tilde{\mathcal{C}}_1$ et $\tilde{\mathcal{C}}_2$ deux courbes de $\mathcal{C}_{\mathcal{T}}$,

$$\tilde{d}_H(\tilde{\mathcal{C}}_1, \tilde{\mathcal{C}}_2) = \inf\{d_H(\mathcal{C}_1, \mathcal{C}_2), \mathcal{C}_1 \in \tilde{\mathcal{C}}_1, \mathcal{C}_2 \in \tilde{\mathcal{C}}_2\}.$$

Ainsi pour les courbes de la figure 2.4, des représentants proches de cet infimum sont représentés sur la figure 2.5. Avec cette définition, les courbes parallèles sont identiques, et les autres courbes sont bien différentes.

Cependant passer au quotient sur les translations pose un problème : deux segments alignés sont à distance nulle alors qu'ils ne forment pas forcément une constance de largeur. Pour ce faire il faudrait qu'ils soient face à face, formant presque un rectangle. Nous définissons alors une pseudo distance

DÉFINITION 4 :

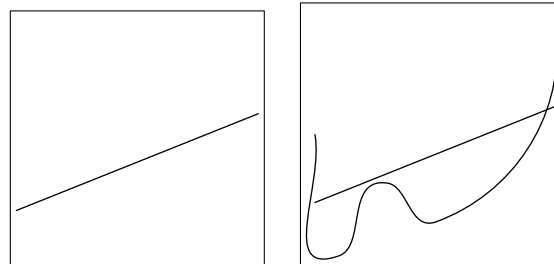
Soient \mathcal{C}_1 et \mathcal{C}_2 deux courbes de \mathcal{C} . Nous définissons

$$\rho'(\mathcal{C}_1, \mathcal{C}_2) = \rho(\mathcal{C}_1, \mathcal{C}_2) - d(\mathcal{C}_1, \mathcal{C}_2),$$

et

$$d'_H(\mathcal{C}_1, \mathcal{C}_2) = \max\{\rho'(\mathcal{C}_1, \mathcal{C}_2), \rho'(\mathcal{C}_2, \mathcal{C}_1)\}.$$

Regardons ce que cela signifie sur la figure 2.6. En fait $\rho'(\mathcal{C}_1, \mathcal{C}_2)$ correspond à l'épaisseur nécessaire pour mettre la courbe \mathcal{C}_2 entre deux dilatés de la



(a) Les deux courbes de la figure 2.4(a) sont identiques à une translation près.

(b) Les deux courbes de la figure 2.4(b) sont toujours différentes.

FIG. 2.5 – Distance de Hausdorff quotient. Les dessins de cette figure reprennent les courbes de la figure 2.4. Les courbes sur les dessins de la présente figure sont des représentants des courbes précédentes à une translation près qui minimise la distance de Hausdorff entre les classes d'équivalence des courbes. Les deux courbes du dessin de gauche étaient identiques à une translation près et leur distance de Hausdorff quotient est donc nulle. Ce n'est pas le cas pour les deux courbes du dessin de droite qui sont différentes et qui sont donc bien à une distance de Hausdorff quotient non nulle l'une de l'autre.

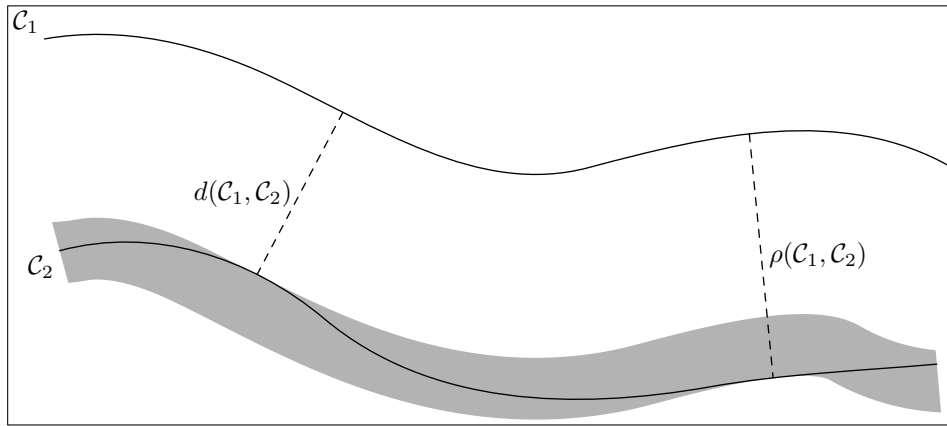


FIG. 2.6 – Tube autour d’une courbe. $d(\mathcal{C}_1, \mathcal{C}_2)$ est la distance minimale entre les points de \mathcal{C}_2 et la courbe \mathcal{C}_1 , par contre $\rho(\mathcal{C}_1, \mathcal{C}_2)$ est la distance maximale entre les points de \mathcal{C}_2 et la courbe \mathcal{C}_1 . Donc tous les points de \mathcal{C}_2 sont situés à une distance de \mathcal{C}_1 comprise entre ces deux extrêmes, $d(\mathcal{C}_1, \mathcal{C}_2)$ et $\rho(\mathcal{C}_1, \mathcal{C}_2)$. C’est-à-dire que \mathcal{C}_2 est coincée dans un tube de largeur $\rho'(\mathcal{C}_1, \mathcal{C}_2) = \rho(\mathcal{C}_1, \mathcal{C}_2) - d(\mathcal{C}_1, \mathcal{C}_2)$ formé à partir des dilatés de \mathcal{C}_1 .

courbe \mathcal{C}_1 . C’est la largeur minimale d’un tube formé à partir de \mathcal{C}_1 pouvant contenir \mathcal{C}_2 . Cependant d'_H ne définit pas une distance. En effet par exemple deux segments parfaitement parallèles, c’est-à-dire tels qu’en joignant les extrémités ils forment un rectangle, sont à distance nulle.

Cette pseudo-distance nous intéresse néanmoins car contrairement à ce qui se passe avec la distance de Hausdorff quotient, deux segments alignés de même longueur ne sont pas à distance nulle. Pour fabriquer une distance à partir de d'_H il suffit de quotienter \mathcal{C} par l’ensemble de courbes à distance nulle.

2.1.3 Probabilité

Nous voulons traduire le fait que deux courbes sinueuses parallèles sont plus remarquables que deux segments de droite parallèles. En effet des courbes sinueuses changent régulièrement d’orientation, il est donc plus remarquable dans le cas de deux courbes sinueuses parallèles que les changements de directions soient comparables ou proches. Nous voulons donc également prendre en compte toute la géométrie des courbes, pas seulement leur longueur mais aussi leur courbure etc.

Pour prendre en compte la géométrie des courbes, nous allons définir une probabilité sur l’espace des courbes. Celle-ci dépendra des courbes et de la tolérance de non parallélisme. La distance d'_H nous permet de définir une topologie sur \mathcal{C} . Et cette topologie à son tour induit une tribu sur \mathcal{C} que l’on appelle la tribu des boréliens \mathcal{B} .

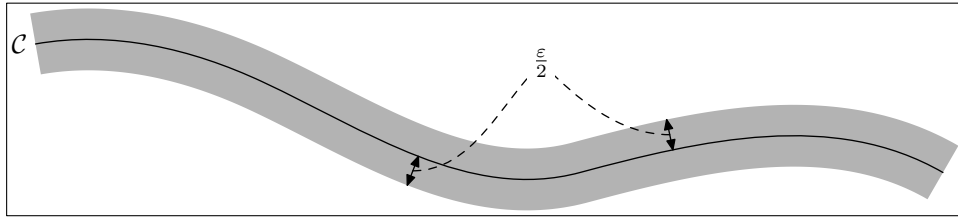


FIG. 2.7 – Un voisinage $V(\mathcal{C}, \varepsilon)$. Le dessin montre un voisinage $V(\mathcal{C}, \varepsilon)$ de largeur ε construit à partir d'une courbe \mathcal{C} .

Nous introduisons un système de voisinage sur \mathcal{C} . Pour commencer nous introduisons les “voisinages” de courbes dans \mathbb{R}^2 .

DÉFINITION 5 :

Soit \mathcal{C} une courbe \mathcal{C}^1 par morceaux. $V(\mathcal{C}, \varepsilon)$ est l'ensemble des points situés à une distance inférieure à $\varepsilon/2$ de \mathcal{C} et situés entre les deux orthogonales aux extrémités de \mathcal{C} .

Sur la figure 2.7 est représenté un tel voisinage. Sa construction est détaillée sur les dessins de la figure 2.8. La figure 2.9 montre $V(\mathcal{C}, \varepsilon)$ pour plusieurs ε .

Ces parties de \mathbb{R}^2 nous permettent maintenant d'introduire les voisinages de courbes :

DÉFINITION 6 :

Soit \mathcal{C} une courbe \mathcal{C}^1 par morceaux. $\mathcal{V}(\mathcal{C}, \varepsilon)$ est l'ensemble des courbes de \mathcal{C} qui sont incluses dans $V(\mathcal{C}, \varepsilon)$.

Ces voisinages définissent bien un ensemble de voisinages :

PROPOSITION 7 :

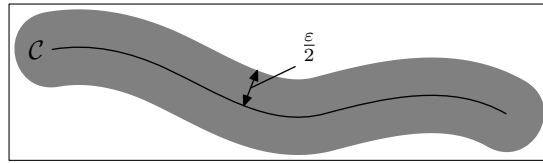
Les ensembles $\{\mathcal{V}(\mathcal{C}, \varepsilon), \mathcal{C} \in \mathcal{C}, \mathcal{C} \text{ est } \mathcal{C}^1 \text{ par morceaux}, \varepsilon > 0\}$ forment un ensemble de voisinages pour \mathcal{C} muni de la topologie induite par d'_H .

Dans les sections suivantes, nous allons définir une probabilité $P(\mathcal{C}, \varepsilon)$ sur ces voisinages. Cette probabilité est fonction de la géométrie des courbes. A partir d'une telle probabilité nous pouvons définir la probabilité de constance de largeur.

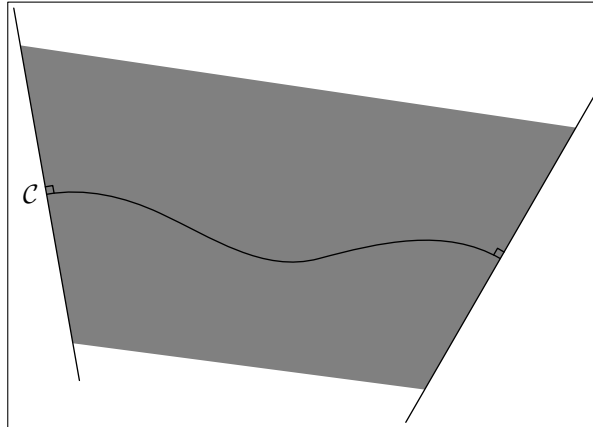
DÉFINITION 8 (Probabilité de constance de largeur) :

Soit P une probabilité sur $(\mathcal{C}, \mathcal{B})$. Pour toute courbe \mathcal{C} dans \mathcal{C} et \mathcal{C}^1 par morceaux on note $P(\mathcal{C}, \varepsilon)$ la probabilité du voisinage $\mathcal{V}(\mathcal{C}, \varepsilon)$,

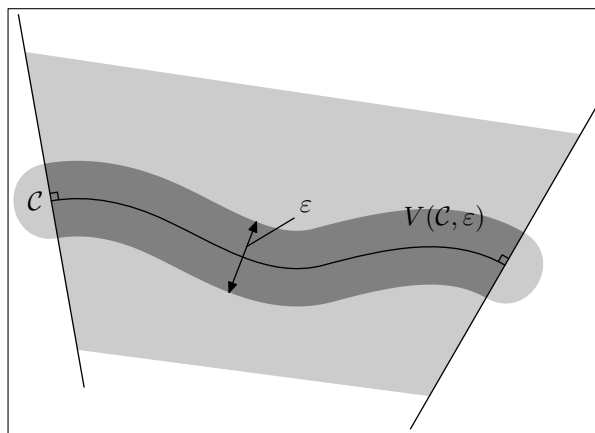
$$P(\mathcal{C}, \varepsilon) = P(\mathcal{V}(\mathcal{C}, \varepsilon)).$$



(a) Dilaté d'une courbe.



(b) Orthogonales aux extrémités.



(c) Intersection des deux.

FIG. 2.8 – Construction d'un voisinage. Pour construire un voisinage nous construisons deux ensembles. Premièrement nous considérons le dilaté $\varepsilon/2$ de la courbe comme sur le premier dessin. Ensuite nous considérons les points situés entre les deux droites orthogonales aux extrémités de la courbe comme sur le deuxième dessin. Enfin le voisinage $V(\mathcal{C}, \varepsilon)$ est l'intersection de ces deux ensembles comme sur le troisième dessin.

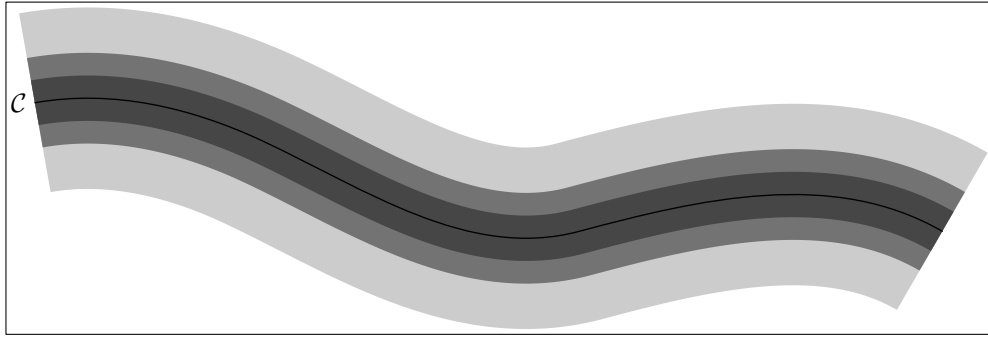


FIG. 2.9 – Plusieurs voisinages autour d’une même courbe. Plusieurs voisinages $V(\mathcal{C}, \varepsilon)$ autour d’une même courbe \mathcal{C} . La courbe \mathcal{C} ne change pas mais nous avons pris différentes valeurs pour ε .

Soient \mathcal{C}_1 et \mathcal{C}_2 dans \mathcal{C} et \mathcal{C}^1 par morceaux,

$$\mathbb{P}(\mathcal{C}_1, \mathcal{C}_2) = \max\{P(\mathcal{C}_1, \rho'(\mathcal{C}_1, \mathcal{C}_2)), P(\mathcal{C}_2, \rho'(\mathcal{C}_2, \mathcal{C}_1))\}$$

est la probabilité de constance de largeur des deux courbes \mathcal{C}_1 et \mathcal{C}_2 .

2.1.4 Probabilité de parallélisme de deux courbes

Avant de définir les probabilités sur les voisinages, nous précisons quel est l’intérêt de cette probabilité \mathbb{P} sur deux courbes. Dans le contexte de l’analyse d’image, nous ne nous intéressons pas à des courbes quelconques mais à des courbes extraites de lignes de niveau d’une image digitale. Les modèles de génération de courbes que nous choisissons doivent générer des courbes partageant les mêmes propriétés que les lignes de niveau :

- elles sont plutôt régulières,
- elles ne s’auto-intersectent pas.

Il n’est pas difficile de concevoir un modèle générant des courbes régulières. Par contre le problème de la non auto-intersection est beaucoup plus complexe. Les travaux de Werner et al. [25] dans ce domaine montrent bien la complexité d’un tel modèle.

De plus les voisinages $\mathcal{V}(\mathcal{C}, \varepsilon)$ sont fabriqués avec une courbe \mathcal{C} ayant une régularité supérieure à \mathcal{C}^1 par morceaux, car \mathcal{C}^1 par morceaux est le cas le moins favorable, et lorsque ε est petit, comme sur la figure 2.7. Pour qu’une courbe s’auto-intersecte dans de tels voisinages, il faudrait qu’elle ait une courbure très élevée par endroits et donc elle ne serait pas régulière. Nous allons donc nous attacher à produire des courbes respectant la première propriété, la régularité.

En fait nous n’avons qu’à définir un générateur de courbe aléatoire (aléatoire au sens probabilité d’une courbe sans *a priori*) et à faire une hypothèse d’indépendance sur les courbes entre elles. Avec ce modèle lorsque deux courbes

sont parallèles, elles ne seront pas indépendantes et l'hypothèse **H0** ne sera pas vérifiée. Nous devons calculer un nombre de fausses alarmes NFA qui est le nombre de mauvaises détections possibles dans l'image. Si N_{test} est le nombre de tests effectués dans l'image alors

DÉFINITION 9 :

Soient \mathcal{C}_1 et \mathcal{C}_2 deux courbes,

$$NFA(\mathcal{C}_1, \mathcal{C}_2) = N_{\text{tests}} \mathbb{P}(\mathcal{C}_1, \mathcal{C}_2).$$

Une paire de courbe est significative si son NFA est faible.

DÉFINITION 10 :

$(\mathcal{C}_1, \mathcal{C}_2)$ est une paire de courbe ε -significative si

$$NFA(\mathcal{C}_1, \mathcal{C}_2) \leq \varepsilon.$$

Avec ce modèle nous avons juste besoin d'un modèle de génération de courbes régulières indépendantes. Ainsi nous devons donc générer une seule courbe régulière, les tirages ensuite étant indépendants.

2.2 Modèle de courbes polygonales

2.2.1 Modèle

Pour un modèle de courbes régulières nous définissons un modèle de courbes polygonales.

DÉFINITION 11 :

Soit $\mathcal{M}_r = (\mathcal{C}_r, \mathcal{B}, P_r)$ le modèle de courbes polygonales,

$$\mathcal{C}_r = \{\text{courbes formées de segments de longueur } r\},$$

P_r est la probabilité associée au générateur de courbes consistant à tirer l'angle de changement de direction des segments selon une probabilité μ_r sur les directions.

Les directions sont les éléments du cercle unité $\mathcal{S}^1 = \mathbb{R}/2\pi\mathbb{Z}$. La courbe de la figure 2.10 est un exemple de courbe de \mathcal{C}_r .

La génération de courbe se passe alors comme décrit ci-après. Le vecteur $\vec{u}(\theta) = (\cos(\theta), \sin(\theta))$ est le vecteur unitaire associé à la direction θ .

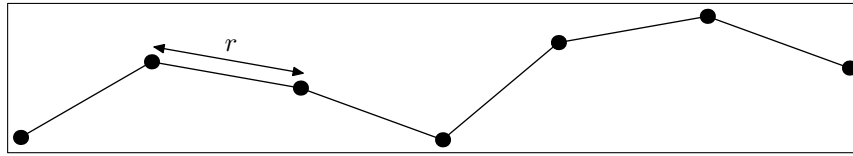


FIG. 2.10 – Une courbe polygonale. La courbe du dessin est constituée de plusieurs segments, tous de la même longueur r , le rayon de la courbe.

ALGORITHME 12 (générateur de courbes du modèle \mathcal{M}_r) :

Fournir (x_0, θ_0) ,
Pour n de 0 à $N - 1$ **faire**
 $x_{n+1} \leftarrow x_n + r\vec{u}(\theta_n)$
 tirer $\Delta\theta_n$ selon μ_r
 $\theta_{n+1} \leftarrow \theta_n + \Delta\theta_n$

DÉFINITION 13 (courbe aléatoire pour le modèle \mathcal{M}_r) :

Une courbe aléatoire pour le modèle \mathcal{M}_r est une suite finie $\{(x_0, \theta_0), \dots, (x_N, \theta_N)\}$ de points de $\mathbb{R}^2 \times \mathcal{S}^1$ tels que

$$\begin{cases} (x_0, \theta_0) \in \mathbb{R}^2 \times \mathcal{S}^1, \\ \forall n \in \llbracket 0, N - 1 \rrbracket, x_{n+1} = x_n + r\vec{u}(\theta_n), \\ \forall n \in \llbracket 0, N - 1 \rrbracket, \theta_{n+1} - \theta_n \sim \mu_r. \end{cases}$$

Ainsi x_0 est la position initiale du début de la courbe et θ_0 sa direction initiale. Pour passer de (x_n, θ_n) à (x_{n+1}, θ_{n+1}) au niveau de la position il suffit de suivre la direction θ_n à partir de x_n sur une distance r ; concernant la direction, le changement de direction est une variable aléatoire qui suit la loi de probabilité μ_r . Pour résumer

$$\begin{cases} x_{n+1} = x_n + r\vec{u}(\theta_n), \\ \theta_{n+1} = \theta_n + \Delta\theta_n, \end{cases} \quad (2.2)$$

$$\Delta\theta \sim \mu_r, \quad (2.3)$$

La figure 2.11 montre quatre premières étapes de notre modèle pour générer une courbe.

2.2.2 Probabilité de changement de direction

C'est avec la probabilité de changement de direction μ_r que nous pouvons choisir la régularité des courbes. En effet plus les grands changements de direction sont probables et plus les courbes générées auront tendance à changer

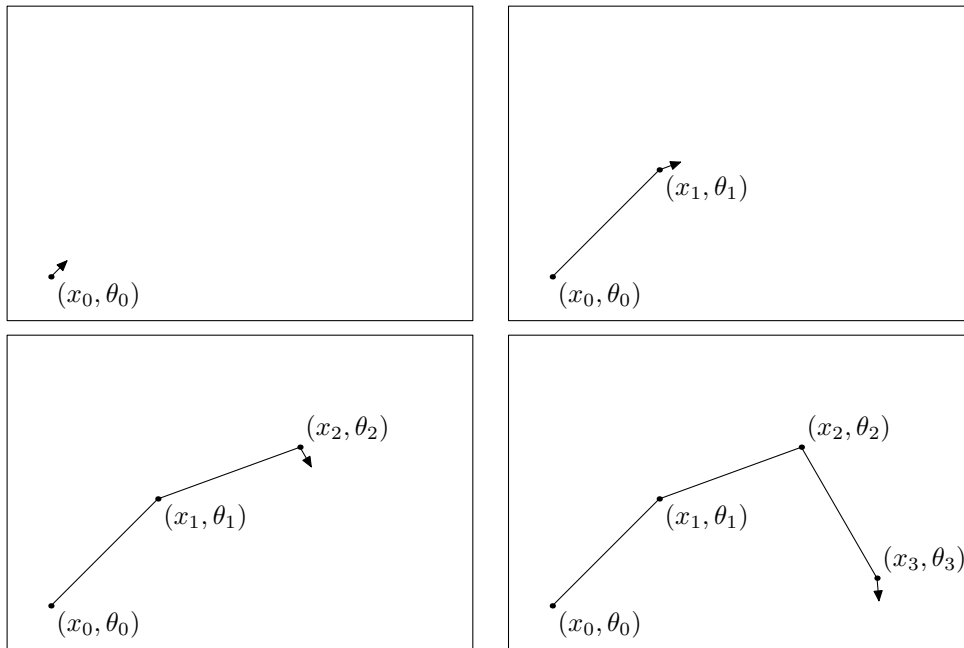


FIG. 2.11 – Construction d’une courbe polygonale. Le dessin en haut à gauche est le point de départ de notre construction à partir du modèle \mathcal{M}_r . La génération est décrite par le générateur de l’algorithme 12. Le point de départ est une extrémité de la courbe. Nous commençons la courbe à partir d’un point x_0 et d’une direction θ_0 . Ensuite nous suivons cette direction sur une distance r pour arriver en x_1 puis nous changeons de direction passant ainsi de θ_0 à θ_1 . À la fin de cette première étape nous arrivons au dessin en haut à droite. À la fin de l’étape suivante nous arrivons au dessin en bas à gauche et avec une étape supplémentaire à celui en bas à droite.

brutalement de direction fréquemment. À l'autre extrême si tous les changements de direction sont improbables, c'est-à-dire si le support de μ_r se réduit à 0 alors dans ce cas les courbes générées sont des segments de droite. Sur la figure 2.12 plusieurs courbes ont été générées avec différentes probabilités de changement de direction. Il est bien visible que plus les changements de direction non nuls sont probables, moins les courbes sont régulières. Pour ces exemples, nous avons pris des lois de Laplace.

DÉFINITION 14 :

μ_r suit une loi de Laplace de paramètre λ si

$$d\mu_r(\theta) = \frac{e^{-\lambda|\theta|}}{\mathcal{Z}} d\theta, \quad (2.4)$$

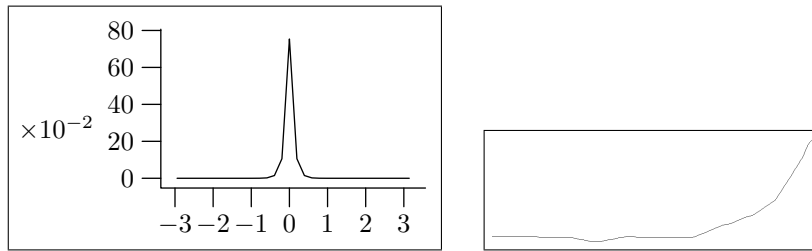
$$\mathcal{Z} = \int_{S^1} e^{-\lambda|\theta|} d\theta. \quad (2.5)$$

Dans ce cas nous notons $\mu_r \sim \Lambda(\lambda)$.

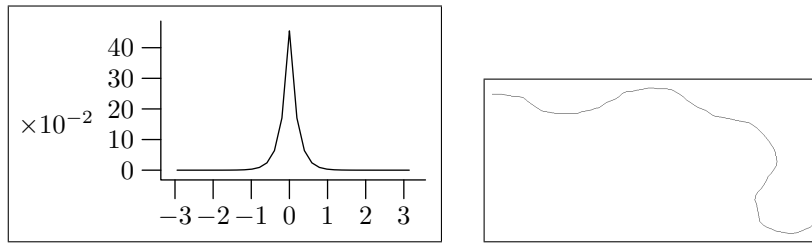
Nous choisissons une loi de Laplace car elle est symétrique par rapport à zéro. Pour les courbes cela signifie qu'elles tournent soit vers la droite soit vers la gauche mais avec la même probabilité, il n'y a pas de biais sur la direction. De plus la loi de Laplace est maximum en zéro et ensuite elle décroît en s'éloignant de zéro. Pour la génération de courbe cela signifie que les petits changements de direction sont plus probables que les grands changements de direction. Et en effet sur la figure 2.12 ces propriétés se traduisent bien au niveau des courbes. Lorsque λ tend vers l'infini, la probabilité de changement de direction tend vers une masse de Dirac en zéro. Effectivement lorsque λ augmente les courbes simulées sont de plus en plus droites. Inversement pour λ égal à zéro, la loi de Laplace correspondante est en fait la loi uniforme, tous les changements de direction sont aussi probables les uns que les autres.

Le mouvement Brownien correspond à $d\mu_r(\theta) = \frac{e^{-\lambda\theta^2}}{\mathcal{Z}} d\theta$, où \mathcal{Z} joue le même rôle. Le mouvement Brownien reproduit mal l'allure des courbes naturelles. Les bords dans les images contiennent la plupart des lignes de niveau et les bords sont souvent dus à un processus d'occlusion, un objet qui masque en partie un autre objet. Les lignes de niveau qui passent par ces bords ont alors tendance à présenter à la fois de brusques changements de direction et des paliers réguliers. Ce sera le cas pour les courbes que nous voulons étudier. La figure 2.13 montre le résultat du calcul d'une loi empirique calculée à partir des lignes de niveau d'une image naturelle. Cette loi ressemble effectivement bien plus à une loi de Laplace qu'à un mouvement Brownien.

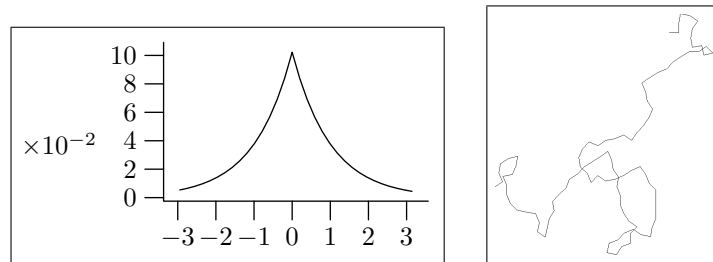
Afin de pouvoir s'adapter au type d'images étudiées, nous pouvons également calculer une loi empirique sur un ensemble $\tilde{\mathcal{C}}$ de courbes.



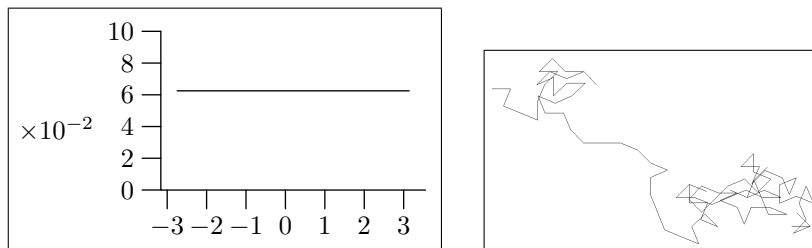
(a) Loi de Laplace de paramètre 10.



(b) Loi de Laplace de paramètre 5.



(c) Loi de Laplace de paramètre 1.



(d) Loi de Laplace de paramètre 0.

FIG. 2.12 – Probabilités de changement de direction. Dans chacune des lignes ci-dessus, à gauche se trouve une probabilité de changement de direction et à droite une courbe générée avec cette probabilité de changement de direction. Les probabilités de changement de direction utilisées sont des lois de Laplace définies dans la définition 14, pour un paramètre prenant les valeurs 10, 5, 1 et enfin 0. La génération de courbe à partir de ces probabilités de changement de direction se fait suivant le générateur de l'algorithme 12.

ALGORITHME 15 (calcul empirique) :

Soit $N_{\text{angles}} = 0$ le nombre d'angles enregistrés
Pour toute courbe \mathcal{C} de $\tilde{\mathcal{C}}$ **faire**
 Soit $N > 2$ le nombre de points de \mathcal{C}
 Soient x_1, \dots, x_N les points de \mathcal{C}
 $n_1 \leftarrow 1$
 Tant que VRAI **faire** ▷ Une boucle infinie
 Trouver $n_2 > n_1$ tel que

$$d_{\mathcal{C}}(x_{n_1}, x_{n_2-1}) < r \text{ et } d_{\mathcal{C}}(x_{n_1}, x_{n_2}) \geq r$$

 Sinon sortir
 Trouver $n_3 > n_2$ tel que

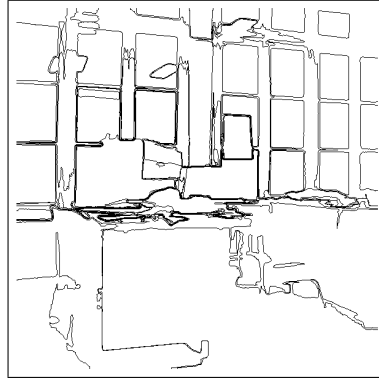
$$d_{\mathcal{C}}(x_{n_2}, x_{n_3-1}) < r \text{ et } d_{\mathcal{C}}(x_{n_2}, x_{n_3}) \geq r$$

 Sinon sortir
 $A \leftarrow x_{n_1}$
 $B \leftarrow x_{n_2}$
 $C \leftarrow x_{n_3}$
 Soit θ l'angle entre les demi-droites $[BC)$ et $[AB)$
 Soit $\tilde{\theta}$ l'angle discret correspondant à θ .
 $\mu_r(\tilde{\theta}) \leftarrow \mu_r(\tilde{\theta}) + 1$
 $N_{\text{angles}} \leftarrow N_{\text{angles}} + 1$
 Pour toutes les valeurs de $\tilde{\theta}$ **faire**
 $\mu_r(\tilde{\theta}) \leftarrow \mu_r(\tilde{\theta}) / N_{\text{angles}}$

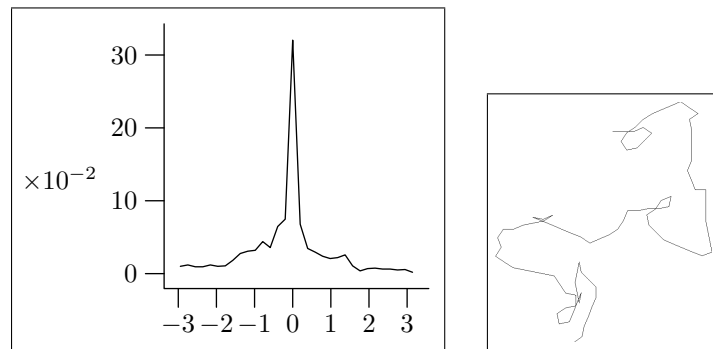
La distance $d_{\mathcal{C}}$ est la distance le long de la courbe. Ainsi nous regardons les triplets (A, B, C) de points sur les courbes à peu près à distance r les uns des autres et nous mesurons le changement de direction représenté par ce triplet. La figure 2.13(a) montre un ensemble de courbes $\tilde{\mathcal{C}}$, ce sont les lignes de niveau significatives d'une image digitale. Sur la figure 2.13(b) nous avons mis la loi empirique calculée à partir de cet échantillon et une courbe simulée suivant cette loi. Nous constatons que l'allure de la loi empirique est comparable à l'allure d'une loi de Laplace.

2.2.3 Extension du voisinage à $\mathbb{R}^2 \times \mathcal{S}^1$

Nous voulons utiliser notre modèle sur les courbes pour générer des courbes aléatoires dans un voisinage $V(\mathcal{C}, \varepsilon)$ que nous avons fabriqué à partir d'une courbe \mathcal{C} . Ce voisinage est l'ensemble des points situés entre les deux orthogonales aux extrémités de \mathcal{C} et à une distance inférieure ou égale à $\varepsilon/2$ comme cela est rappelé par la figure 2.14.



(a) Courbes utilisées pour le calcul empirique.



(b) Loi calculée empiriquement.

FIG. 2.13 – Comme sur la figure 2.12 nous avons mis en bas une probabilité de changement de direction à gauche et à droite une courbe générée à partir de cette probabilité en suivant le générateur de courbe de l’algorithme 12. Cette fois-ci la probabilité de changement de direction est obtenue empiriquement à partir de l’ensemble de courbes du dessin du haut selon le calcul de l’algorithme 15 avec $r = 8$. La loi obtenue et la courbe simulée peuvent tout à fait être comparées à celles sur la figure 2.12 fabriquées avec des lois de Laplace.

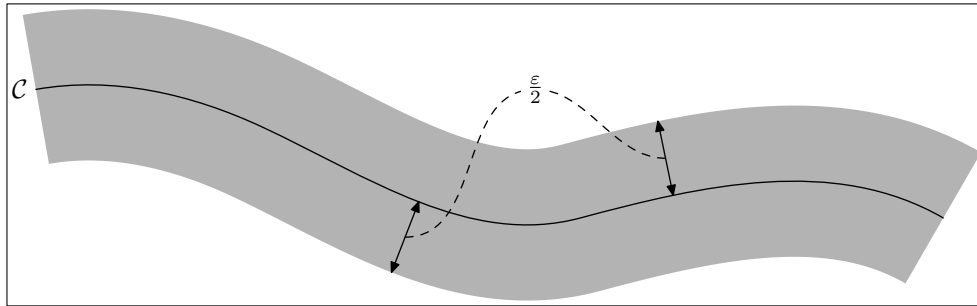


FIG. 2.14 – Le voisinage $V(\mathcal{C}, \varepsilon)$ de largeur ε d'une courbe \mathcal{C} . Il est constitué des points à distance inférieure ou égale à $\varepsilon/2$ de la courbe et entre les deux orthogonales aux extrémités, cf. figure 2.8. Ce voisinage est inclus dans \mathbb{R}^2

Les courbes de notre modèle sont constituées de points (x, θ) de $\mathbb{R}^2 \times \mathcal{S}^1$ et $V(\mathcal{C}, \varepsilon)$ est une partie de \mathbb{R}^2 . Pour placer des extrémités de courbe sur $V(\mathcal{C}, \varepsilon)$ nous construisons un domaine $\mathcal{D}_r(\mathcal{C}, \varepsilon)$ de $\mathbb{R}^2 \times \mathcal{S}^1$ à partir de $V(\mathcal{C}, \varepsilon)$. Pour chaque position x dans le voisinage nous considérons l'ensemble $\mathcal{S}_r(x) \subset \mathcal{S}^1$ des directions que l'on peut suivre à partir de cette position tout en restant dans le voisinage, c'est-à-dire

$$\mathcal{S}_r(x) = \{\theta \in \mathcal{S}^1 \mid x + r\vec{u}(\theta) \in V(\mathcal{C}, \varepsilon)\}.$$

Ceci est illustré par la figure 2.15. Sur cette figure, $\mathcal{S}_r(x_1)$ contient un peu plus d'un quart des directions de \mathcal{S}^1 , par contre $\mathcal{S}_r(x_2)$ est égal à \mathcal{S}^1 tout entier et $\mathcal{S}_r(x_3)$ est constitué d'un peu moins de la moitié des directions de \mathcal{S}^1 .

DÉFINITION 16 :

Nous définissons le domaine $\mathcal{D}_r(\mathcal{C}, \varepsilon)$ comme l'ensemble des positions dans le voisinage $V(\mathcal{C}, \varepsilon)$ avec les directions pointant vers $V(\mathcal{C}, \varepsilon)$,

$$\mathcal{D}_r(\mathcal{C}, \varepsilon) = \cup_{x \in V(\mathcal{C}, \varepsilon)} \{x\} \times \mathcal{S}_r(x).$$

2.3 Probabilités de présence et de sortie

2.3.1 Probabilité de présence

Nous faisons donc évoluer des courbes suivant le modèle \mathcal{M}_r dans le domaine $\mathcal{D}_r(\mathcal{C}, \varepsilon)$ de $\mathbb{R}^2 \times \mathcal{S}^1$.

DÉFINITION 17 :

Soit $Q_n : \mathcal{D}_r(\mathcal{C}, \varepsilon) \rightarrow \mathbb{R}$ la probabilité de présence des extrémités de courbe à l'étape n .

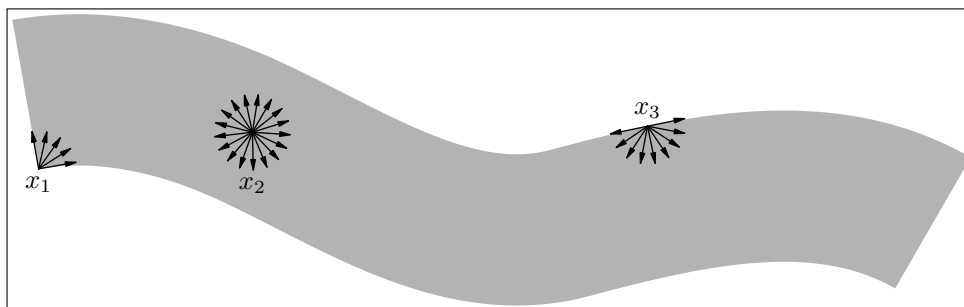


FIG. 2.15 – Extension de \mathbb{R}^2 à $\mathbb{R}^2 \times \mathcal{S}^1$. Pour générer des courbes avec le générateur 12, nous devons prendre des extrémités de courbe. Une extrémité de courbe est constituée d'une position et d'une direction. Pour générer des courbes dans le voisinage $V(\mathcal{C}, \varepsilon)$ nous devons donc étendre ce voisinage de \mathbb{R}^2 à $\mathbb{R}^2 \times \mathcal{S}^1$. Pour cela nous regardons en chaque point x les directions $\mathcal{S}_r(s)$ venant du voisinage et d'une distance r . Ainsi pour le point x_1 il s'agit d'un peu plus du quart de \mathcal{S}^1 pour x_2 de \mathcal{S}^1 en entier et d'un peu moins de la moitié de \mathcal{S}^1 pour x_3 .

C'est-à-dire que nous plaçons des extrémités initiales de courbe (x_0, θ_0) à travers une probabilité de présence initiale Q_0 . Ainsi $Q_0(x, \theta)$ est la probabilité de présence d'une extrémité de courbe en (x, θ) . Les courbes que nous prenons en compte sont seulement les courbes qui commencent d'un côté du voisinage $V(\mathcal{C}, \varepsilon)$ et qui sortent de l'autre côté. Nous définissons pour cela une extrémité de début et une extrémité de fin.

Le voisinage $V(\mathcal{C}, \varepsilon)$ est formé à partir de la courbe \mathcal{C} en utilisant les orthogonales aux extrémités de \mathcal{C} , cf. figure 2.8(c). L'intersection des orthogonales avec $\bar{V}(\mathcal{C}, \varepsilon)$ forme en général deux segments. Soient Γ_d et Γ_f ces deux segments, sur la figure 2.16 ces segments apparaissent avec des traits forts. Le début des courbes est situé sur Γ_d et leur fin est sur Γ_f .

DÉFINITION 18 :

L'ensemble de départ des extrémités de courbe est

$$O_d = \cup_{\Gamma_d} \{x\} \times (-\mathcal{S}_r(x)).$$

Pour que les courbes commencent sur O_d nous prenons Q_0 suivant une loi uniforme sur O_d .

2.3.2 Propagation de la probabilité de présence

Le modèle \mathcal{M}_r nous permet de connaître Q_{n+1} en connaissant Q_n .

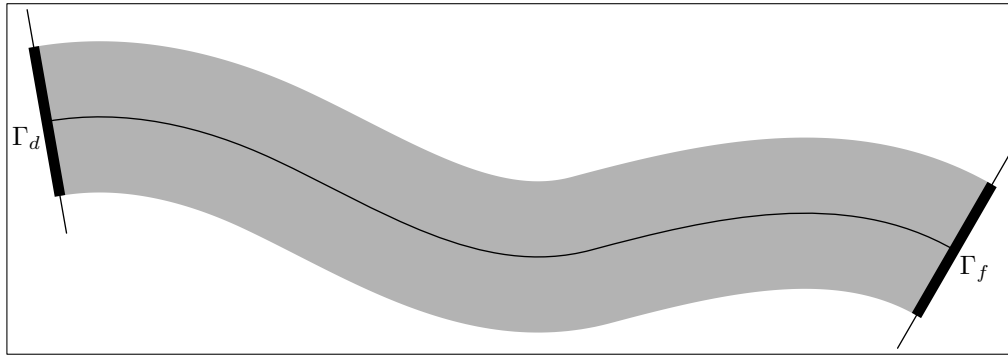


FIG. 2.16 – Extrémités des voisinages. Nous définissons pour chaque voisinage $V(\mathcal{C}, \varepsilon)$ deux extrémités. Il s'agit des bords du voisinage construits à partir des droites orthogonales aux extrémités de \mathcal{C} . Une des extrémités est notée Γ_d , c'est celle où débiteront les courbes, et l'autre est notée Γ_f c'est par là que les courbes doivent sortir du voisinage.

PROPOSITION 19 :

Avec le modèle de courbe \mathcal{M}_r la probabilité de présence Q vérifie

$$Q_{n+1}(x, \theta) = \int_{S_x} Q_n(x - r\vec{u}(\varphi), \varphi) d\mu_r(\theta - \varphi).$$

Preuve : En effet avec le modèle \mathcal{M}_r si une courbe arrive en (x, θ) à partir de (y, φ) alors

$$\begin{cases} x &= y + r\vec{u}(\varphi), \\ \theta &= \varphi + \Delta\varphi. \end{cases}$$

Ce qui signifie

$$\begin{cases} y &= x - r\vec{u}(\varphi), \\ \Delta\varphi &= \theta - \varphi. \end{cases}$$

Ainsi pour qu'une courbe arrive en (x, θ) , il faut que sa position précédente soit située sur le cercle de centre x et de rayon r avec une direction pointant vers x et que le changement de direction aboutisse à θ . Ceci est illustré par la figure 2.17. Cela nous permet d'écrire la formule de la proposition 19. \square

2.3.3 Probabilité de sortie

Nous avons appliqué le modèle \mathcal{M}_r pour simuler des courbes à l'intérieur du voisinage $V(\mathcal{C}, \varepsilon)$ qui commencent sur le côté Γ_d . Mais ce qui nous intéresse réellement ce sont les courbes qui sortent de l'autre côté. Pour cela

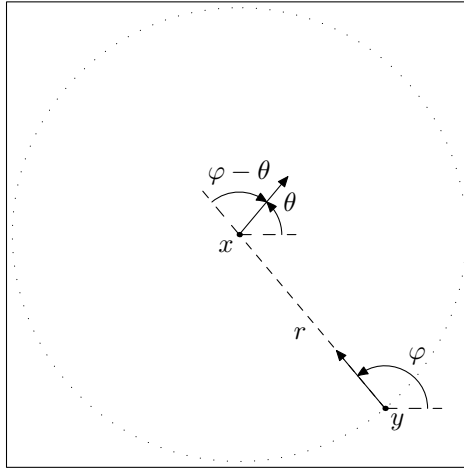


FIG. 2.17 – Cercle d'intégration. Pour passer de (y, φ) à (x, θ) en suivant le modèle de génération 12, le point y doit être sur le cercle de centre x et de rayon r . De plus la direction φ doit correspondre à la direction du vecteur $y\vec{x}$. Le changement de direction pour passer de (y, φ) à (x, θ) est $\theta - \varphi$.

nous appelons O_f l'ensemble des points du domaine $\mathcal{D}_r(\mathcal{C}, \varepsilon)$ tels qu'il existe un $r_0 \in [0, r[$ avec $[x, x + r_0\vec{u}(\theta)] \subset V(\mathcal{C}, \varepsilon)$ et tel qu'il existe $\eta > 0$ avec $]x + r_0\vec{u}(\theta), x + (r_0 + \eta)\vec{u}(\theta)] \subset \mathbb{R}^2 \setminus V(\mathcal{C}, \varepsilon)$ avec $x + r_0\vec{u}(\theta)$ sur Γ_f . Sur la figure 2.18 les deux points (x, θ) et (x', θ') sont des extrémités qui font sortir les courbes du voisinage, par contre seul le point (x, θ) appartient à O_f car avec cette extrémité une courbe quitte le voisinage $V(\mathcal{C}, \varepsilon)$ à travers le côté Γ_f .

Soit $P_n = \iiint_{O_f} Q_n(x, \theta) dx d\theta$, P_n est la probabilité qu'une courbe reste dans le tube durant $(n - 1)$ itérations et sorte du tube à la n -ème itération.

Ainsi $\sum_{k=0}^n P_k$ est la probabilité qu'une courbe de longueur inférieure ou égale à nr sorte du tube par la sortie S_f . La probabilité de sortie que nous cherchons est donc

DÉFINITION 20 :

Avec le modèle de courbe \mathcal{M}_r , nous avons une probabilité sur un voisinage $V(\mathcal{C}, \varepsilon)$ définie par

$$P_r(\mathcal{C}, \varepsilon) = \sum_{k=0}^{\infty} P_k = \sum_{k=0}^{\infty} \iiint_{O_f} Q_k(x, \theta) dx d\theta.$$

Nous voulons calculer cette probabilité P_r .

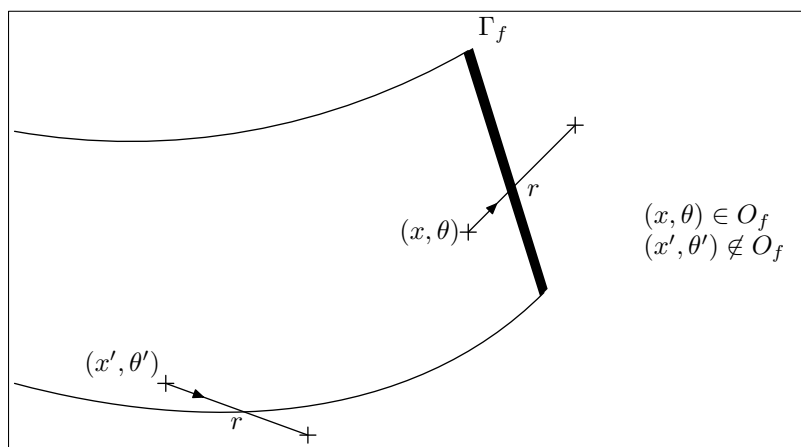


FIG. 2.18 – Sortie du voisinage. Les courbes qui nous intéressent sont les courbes qui sortent du voisinage en traversant l'extrémité Γ_f . Sur le dessin les courbes passant par (x, θ) et celles passant par (x', θ') sortent toutes du voisinage à l'étape suivante. Par contre seules celles passant par (x, θ) le font en traversant Γ_f .

Chapitre 3

Calcul de la probabilité

La complexité du modèle \mathcal{M}_r ne permet pas d'envisager un calcul analytique de P_r . Nous devons donc définir d'autres modèles pour en avoir un calcul approché. Nous avons plusieurs options. La première est de discrétiser le voisinage $V(\mathcal{C}, \varepsilon)$. Nous définissons alors la fonction \tilde{Q} , adaptation de Q à ces voisinages discrétisés. La deuxième option est d'estimer P_r en effectuant des simulations de courbe. Enfin la troisième méthode que nous abordons est de trouver un modèle \mathcal{M}_0 de courbes dérivé du modèle \mathcal{M}_r lorsque nous faisons tendre r vers 0. À ce modèle infinitésimal \mathcal{M}_0 est associée une équation aux dérivées partielles que nous utilisons pour calculer $P_0(\mathcal{C}, \varepsilon)$.

3.1 Passage au discret

3.1.1 Discrétisation du domaine

La première solution pour calculer P_r est de définir un modèle de courbe $\tilde{\mathcal{M}}_r$ sur des domaines discrétisés. La génération des courbes se fait sur une grille $(\delta i, \delta j)$, où δ est le pas de discrétisation. À un point (i, j) de \mathbb{Z}^2 nous associons le pixel

$$X_{i,j} = [\delta i - \delta/2; \delta i + \delta/2[\times [\delta j - \delta/2; \delta j + \delta/2[,$$

de centre $x_{i,j} = (\delta i, \delta j)$, comme cela est illustré sur la figure 3.1(a).

Pour la discrétisation du cercle unité, \mathcal{S}^1 est divisé en K intervalles. À un entier k de $\mathbb{Z}/K\mathbb{Z}$ nous associons le secteur angulaire

$$I_k = [\theta_k - \pi/K; \theta_k + \pi/K[$$

de centre $\theta_k = 2k\pi/K$ comme cela est illustré par la figure 3.1(b).

La génération de courbe se fait alors comme suit

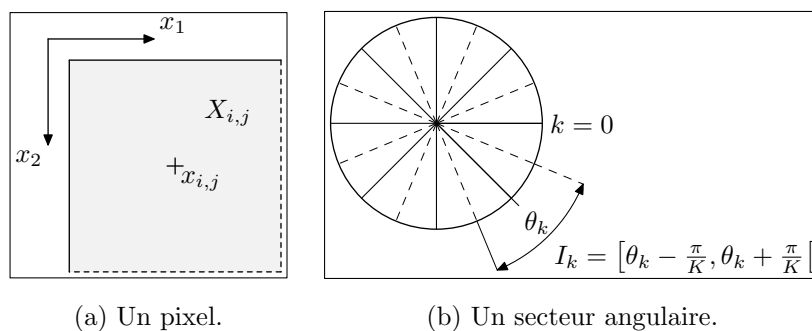


FIG. 3.1 – Discrétisations. Nous avons deux discrétisations à effectuer. La première est la discrétisation du plan \mathbb{R}^2 et la deuxième est celle du cercle unité \mathcal{S}^1 . Pour la discrétisation de \mathbb{R}^2 nous considérons la grille constituée des points $x_{i,j}$ de coordonnées $(\delta i, \delta j)$. Le pixel $X_{i,j}$ est le carré de centre $x_{i,j}$ et de côté δ . Un tel pixel est dessiné dans la figure de gauche. Pour la discrétisation de \mathcal{S}^1 nous considérons les directions θ_k valant $2k\pi/K$. Le secteur angulaire I_k est l'intervalle de direction de centre θ_k et d'amplitude $2\pi/K$. Ces secteurs angulaires sont dessinés dans la figure de droite.

ALGORITHME 21 (générateur de courbe du modèle $\tilde{\mathcal{M}}_r$) :

Fournir (i_0, j_0, k_0) le point de départ correspondant à $(x_{i_0, j_0}, \theta_{k_0})$
Pour n de 0 à $N - 1$ **faire**
 Tirer η_n uniformément sur $[-\pi/K; \pi/K[$
 $x \leftarrow x_{i_n, j_n} + r\vec{u}(\theta_{k_n} + \eta_n)$
 Soit $x_{i_{n+1}, j_{n+1}}$ tel que $x \in X_{i_{n+1}, j_{n+1}}$
 Tirer Δk_n selon la loi $\tilde{\mu}_r$,
 $k_{n+1} \leftarrow k_n + \Delta k_n$.

En partant du point $(x_{i_n, j_n}, \theta_{k_n})$ nous suivons la direction $\theta_{k_n} + \eta_n$ où η_n est tirée uniformément de sorte que $\theta_{k_n} + \eta_n$ soit choisi uniformément sur I_{k_n} . Le point $x_{i_{n+1}, j_{n+1}}$ est alors le centre du pixel d'arrivée et la nouvelle direction est $\theta_{k_{n+1}} = \theta_{k_n} + \Delta k_n$ où Δk_n suit une loi $\tilde{\mu}_r$ définie sur $\mathbb{Z}/K\mathbb{Z}$.

DÉFINITION 22 :

Le modèle polygonal discret $\mathcal{M}_r = (\tilde{\mathcal{C}}_r, \mathcal{B}, \tilde{P}_r)$ est tel que

- $\tilde{\mathcal{C}}_r$ est l'ensemble des courbes qui peuvent être simulées par le générateur 21,
- la tribu \mathcal{B} est en fait la trace sur $\tilde{\mathcal{C}}_r$ des boréliens construits à partir de l'ensemble \mathcal{C} des courbes de longueur finie de \mathbb{R}^2 ,

- \tilde{P}_r est la probabilité de sortie que nous allons définir sur les voisinages discrétisés avec le générateur de courbe 21.

3.1.2 Propagation discrète

Pour calculer \tilde{P}_r nous avons besoin de \tilde{Q} ,

DÉFINITION 23 :

La probabilité de présence discrétisée est

$$\tilde{Q}_n(i, j, k) = \int_{I_k} \iint_{X_{i,j}} Q_n(x, \theta) dx d\theta.$$

De la formule de la proposition 19 nous allons déduire la formule suivante pour la version discrète \tilde{Q}

PROPOSITION 24 :

$$\tilde{Q}_{n+1}(i, j, k) = \sum_{l=0}^{K-1} \sum_{(a,b) \in \mathbb{Z}^2} \omega(i, j, a, b, l) \tilde{\mu}_r(k-l) \tilde{Q}_n(a, b, l) + R(\delta, K),$$

avec

$$|R(\delta, K)| \leq \frac{5\pi\delta^2}{K} + \frac{4r\pi^2\delta}{K^2}.$$

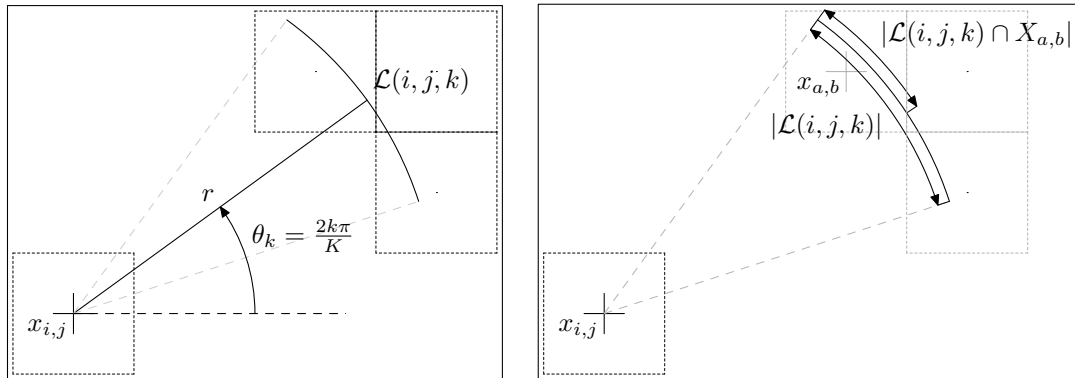
Dans la formule de la proposition (24) $\tilde{\mu}_r(k) = \mu_r(I_k)$ et $\omega(i, j, a, b, k)$ est un poids. Soit $\mathcal{L}(i, j, k)$ l'arc de cercle de centre $x_{i,j}$ correspondant au secteur angulaire I_k . La figure 3.2(a) montre les pixels traversés par cet arc de cercle. $\omega(i, j, a, b, k)$ est le poids correspondant à la longueur relative du morceau d'arc de cercle qui traverse le pixel $X_{a,b}$ comme le montre la figure 3.2(b).

$$\omega(i, j, a, b, k) = \begin{cases} 0 & \text{si } \mathcal{L}_{i,j,k} \cap X_{a,b} = \emptyset, \\ \frac{|\mathcal{L}_{i,j,k} \cap X_{a,b}|}{|\mathcal{L}_{i,j,k}|} & \text{sinon.} \end{cases} \quad (3.1)$$

Pour prouver cette proposition nous avons besoin de deux lemmes,

LEMME 25 :

$$\begin{aligned} & \sum_{l=0}^{K-1} \int_{I_{k-l}} \int_{I_{k-\psi}} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta), \theta) dx d\theta d\mu_r(\psi) \\ &= \sum_{l=0}^{K-1} \int_{I_{k-l}} \int_{I_l} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta), \theta) dx d\theta d\mu_r(\psi) + R_1(\delta, K), \end{aligned}$$



(a) Pixels traversés par un secteur angulaire.

(b) Longueur relative.

FIG. 3.2 – Suivi d’une direction discrète. Chaque direction θ_k représente en fait tout un intervalle de direction I_k . Sur le dessin de gauche, nous avons dessiné le point de départ $x_{i,j}$ et la direction θ_k . L’arc de cercle constitué de tous les points d’arrivée possibles en partant de $x_{i,j}$ et en suivant les directions de I_k forment l’arc $\mathcal{L}(i, j, k)$. Si nous tirons selon une loi uniforme une direction de I_k alors la probabilité d’arriver dans le pixel $X_{a,b}$, traversé par $\mathcal{L}(i, j, k)$ est égale à la longueur relative de l’intersection. Sur le dessin de droite nous avons montré la longueur relative de l’arc de cercle traversant un pixel.

avec

$$|R_1(\delta, K)| \leq \frac{\pi\delta^2}{K}.$$

Ce premier lemme sert à remplacer une intégrale sur $I_k - \psi$ par une intégrale sur I_l . Ceci revient à remplacer ψ par θ_{k-l} avec ψ appartenant à I_{k-l} qui est centré autour de θ_{k-l} .

LEMME 26 :

$$\begin{aligned} & \sum_{l=0}^{K-1} \int_{I_{k-l}} \int_{I_l} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta), \theta) dx d\theta d\mu_r(\psi) \\ &= \sum_{l=0}^{K-1} \int_{I_{k-l}} \sum_{(a,b) \in \mathbb{Z}^2} \omega(i, j, a, b, l) \tilde{Q}_n(a, b, l) d\mu_r(\psi) + R_2(\delta, K), \end{aligned}$$

avec

$$|R_2(\delta, K)| \leq \frac{4\pi\delta^2}{K} + \frac{4r\pi^2\delta}{K^2}.$$

La deuxième lemme consiste à préciser ce qui se passe lorsque nous passons des intégrales

$$\int_{I_l} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta), \theta) dx d\theta$$

à la variable discrétisée \tilde{Q}_n . En effet cette expression ressemble à la définition 23 de \tilde{Q} . Ce lemme nous permet de contrôler l'erreur que nous faisons en passant de Q à \tilde{Q} .

Commençons par montrer le lemme 25.

Preuve : Les majorations suivantes sont évidentes,

$$\begin{aligned} & \left| \sum_{l=0}^{K-1} \int_{I_{k-l}} \int_{I_{k-\psi}} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta), \theta) dx d\theta d\mu_r(\psi) \right. \\ & \quad \left. - \sum_{l=0}^{K-1} \int_{I_{k-l}} \int_{I_l} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta), \theta) dx d\theta d\mu_r(\psi) \right| \\ & \leq \sum_{l=0}^{K-1} \int_{I_{k-l}} \iint_{X_{i,j}} \left| \int_{I_{k-\psi}} Q_n(x - r\vec{u}(\theta), \theta) d\theta \right. \\ & \quad \left. - \int_{I_l} Q_n(x - r\vec{u}(\theta), \theta) d\theta \right| dx d\mu_r(\psi) \end{aligned}$$

Ensuite nous remarquons que la différence des intégrales sur les ensembles communs est nulle. Ensuite nous utilisons le fait que Q_n est positif, ceci découle facilement de l'équation intégrale de la proposition 19 vérifiée par Q (si cette propriété est vérifiée par Q_0).

$$\begin{aligned}
 & \sum_{l=0}^{K-1} \int_{I_{k-l}} \iint_{X_{i,j}} \left| \int_{I_k-\psi} Q_n(x - r\vec{u}(\theta), \theta) d\theta \right. \\
 & \qquad \qquad \qquad \left. - \int_{I_l} Q_n(x - r\vec{u}(\theta), \theta) d\theta \right| dx d\mu_r(\psi) \\
 & \leq \sum_{l=0}^{K-1} \int_{I_{k-l}} \iint_{X_{i,j}} \left| \int_{I_k-\psi \setminus I_l} Q_n(x - r\vec{u}(\theta), \theta) d\theta \right. \\
 & \qquad \qquad \qquad \left. - \int_{I_l \setminus (I_k-\psi)} Q_n(x - r\vec{u}(\theta), \theta) d\theta \right| dx d\mu_r(\psi) \\
 & \leq \sum_{l=0}^{K-1} \int_{I_{k-l}} \iint_{X_{i,j}} \int_{(I_k-\psi \setminus I_l) \cup (I_l \setminus (I_k-\psi))} Q_n(x - r\vec{u}(\theta), \theta) d\theta dx d\mu_r(\psi)
 \end{aligned}$$

De la même manière que Q_n est positif si Q_0 l'est, Q_n est majoré par 1 si Q_0 l'est.

$$\begin{aligned}
 & \sum_{l=0}^{K-1} \int_{I_{k-l}} \iint_{X_{i,j}} \int_{(I_k-\psi \setminus I_l) \cup (I_l \setminus (I_k-\psi))} Q_n(x - r\vec{u}(\theta), \theta) d\theta dx d\mu_r(\psi) \\
 & \leq \sum_{l=0}^{K-1} \int_{I_{k-l}} \iint_{X_{i,j}} |(I_k - \psi) \Delta I_l| dx d\mu_r(\psi)
 \end{aligned}$$

Dans cette formule, $|(I_k - \psi) \Delta I_l|$ désigne la mesure de la différence symétrique des deux intervalles $I_k - \psi$ et I_l . Maintenant nous utilisons le fait que ψ est dans I_{k-l} pour majorer la différence symétrique des deux intervalles par la longueur d'un demi-intervalle, $\frac{\pi}{K}$.

$$\begin{aligned}
 & \sum_{l=0}^{K-1} \int_{I_{k-l}} \iint_{X_{i,j}} |(I_k - \psi) \Delta I_l| dx d\mu_r(\psi) \\
 & \leq \sum_{l=0}^{K-1} \int_{I_{k-l}} \iint_{X_{i,j}} \frac{\pi}{K} dx d\mu_r(\psi) \\
 & \leq \sum_{l=0}^{K-1} \int_{I_{k-l}} \frac{\pi \delta^2}{K} d\mu_r(\psi) \\
 & \leq \frac{\pi \delta^2}{K}
 \end{aligned}$$

où nous avons utilisé le fait que μ_r est une probabilité sur \mathcal{S}^1 . □

Démontrons maintenant le lemme 26

Preuve : Nous avons

$$\begin{aligned} & \left| \sum_{l=0}^{K-1} \int_{I_{k-l}} \int_{I_l} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta), \theta) dx d\theta d\mu_r(\psi) \right. \\ & \quad \left. - \sum_{l=0}^{K-1} \int_{I_{k-l}} \sum_{(a,b) \in \mathbb{Z}^2} \omega(i, j, a, b, l) \tilde{Q}_n(a, b, l) d\mu_r(\psi) \right| \\ & \leq \sum_{l=0}^{K-1} \int_{I_{k-l}} \left| \int_{I_l} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta), \theta) dx d\theta \right. \\ & \quad \left. - \sum_{(a,b) \in \mathbb{Z}^2} \omega(i, j, a, b, l) \tilde{Q}_n(a, b, l) \right| d\mu_r(\psi) \end{aligned}$$

Ensuite nous utilisons la définition de \tilde{Q}_n .

$$\begin{aligned} & \sum_{l=0}^{K-1} \int_{I_{k-l}} \left| \int_{I_l} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta), \theta) dx d\theta \right. \\ & \quad \left. - \sum_{(a,b) \in \mathbb{Z}^2} \omega(i, j, a, b, l) \tilde{Q}_n(a, b, l) \right| d\mu_r(\psi) \\ & \leq \sum_{l=0}^{K-1} \int_{I_{k-l}} \left| \int_{I_l} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta), \theta) dx d\theta \right. \\ & \quad \left. - \sum_{(a,b) \in \mathbb{Z}^2} \omega(i, j, a, b, l) \int_{I_l} \iint_{X_{a,b}} Q_n(x, \theta) dx d\theta \right| d\mu_r(\psi) \end{aligned}$$

Nous effectuons un changement de variable en la variable x

$$\begin{aligned} & \sum_{l=0}^{K-1} \int_{I_{k-l}} \left| \int_{I_l} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta), \theta) dx d\theta \right. \\ & \quad \left. - \sum_{(a,b) \in \mathbb{Z}^2} \omega(i, j, a, b, l) \int_{I_l} \iint_{X_{a,b}} Q_n(x, \theta) dx d\theta \right| d\mu_r(\psi) \\ & \leq \sum_{l=0}^{K-1} \int_{I_{k-l}} \int_{I_l} \left| \iint_{X_{i,j-r\vec{u}(\theta)}} Q_n(x, \theta) dx \right. \\ & \quad \left. - \sum_{(a,b) \in \mathbb{Z}^2} \omega(i, j, a, b, l) \iint_{X_{a,b}} Q_n(x, \theta) dx \right| d\theta d\mu_r(\psi) \end{aligned}$$

Puis nous utilisons le fait que l'ensemble des pixels forme une partition de \mathbb{R}^2 .

$$\begin{aligned}
 & \sum_{l=0}^{K-1} \int_{I_{k-l}} \int_{I_l} \left| \iint_{X_{i,j-r\vec{u}(\theta)}} Q_n(x, \theta) dx \right. \\
 & \qquad \qquad \qquad \left. - \sum_{(a,b) \in \mathbb{Z}^2} \omega(i, j, a, b, l) \iint_{X_{a,b}} Q_n(x, \theta) dx \right| d\theta d\mu_r(\psi) \\
 & \leq \sum_{l=0}^{K-1} \int_{I_{k-l}} \int_{I_l} \sum_{(a,b) \in \mathbb{Z}^2} \left| \iint_{(X_{i,j-r\vec{u}(\theta)}) \cap X_{a,b}} Q_n(x, \theta) dx \right. \\
 & \qquad \qquad \qquad \left. - \omega(i, j, a, b, l) \iint_{X_{a,b}} Q_n(x, \theta) dx \right| d\theta d\mu_r(\psi) \\
 & \leq \sum_{l=0}^{K-1} \int_{I_{k-l}} \int_{I_l} \sum_{(a,b) \in \mathbb{Z}^2} \left| (1 - \omega(i, j, a, b, l)) \iint_{(X_{i,j-r\vec{u}(\theta)}) \cap X_{a,b}} Q_n(x, \theta) dx \right. \\
 & \qquad \qquad \qquad \left. - \omega(i, j, a, b, l) \iint_{X_{a,b} \setminus (X_{i,j-r\vec{u}(\theta)})} Q_n(x, \theta) dx \right| d\theta d\mu_r(\psi)
 \end{aligned}$$

La différence dans la valeur absolue est majorée par

$$[(1 - \omega) + \omega] \delta^2 = \delta^2.$$

Donc pour majorer la somme sur les couples (a, b) il nous faut connaître le nombre de termes non nuls. La première intégrale est nulle dès que $(X_{i,j-r\vec{u}(\theta)}) \cap X_{a,b} = \emptyset$, or ceci est vrai pour tous les pixels sauf au plus quatre. Enfin le deuxième terme est en facteur de $\omega(i, j, a, b, l)$ et est donc nul dès que $\mathcal{L}_{i,j,l} \cap X_{a,b} = \emptyset$. Il faut donc trouver une majoration du nombre de pixels traversés par un arc de cercle de rayon r et d'amplitude $2\pi/K$. Pour cela nous regardons combien de pixels sont traversés au plus verticalement et horizontalement et nous ferons la somme des deux. Comme les deux situations sont symétriques, nous ne regardons qu'un seul cas. La longueur de l'arc de cercle est $2r\pi/K$ il traverse donc horizontalement au plus $(2r\pi)/(K\delta)$ pixels de largeur δ . Même si l'arc de cercle est de courte amplitude il peut traverser deux pixels (horizontalement dans le pire des cas, en étant à cheval sur deux pixels). Pour majorer le nombre de pixels traversés horizontalement nous utilisons donc

$$2 + \frac{2r\pi}{K\delta}.$$

Nous obtenons donc en utilisant la même majoration pour le nombre de

pixels traversés verticalement

$$\begin{aligned}
 \sum_{l=0}^{K-1} \int_{I_{k-l}} \int_{I_l} \sum_{(a,b) \in \mathbb{Z}^2} & \left| (1 - \omega(i, j, a, b, l)) \iint_{(X_{i,j} - r\vec{u}(\theta)) \cap X_{a,b}} Q_n(x, \theta) dx \right. \\
 & \left. - \omega(i, j, a, b, l) \iint_{X_{a,b} \setminus (X_{i,j} - r\vec{u}(\theta))} Q_n(x, \theta) dx \right| d\theta d\mu_r(\psi) \\
 & \leq \sum_{l=0}^{K-1} \int_{I_{k-l}} \int_{I_l} \left(4 + \frac{4r\pi}{K\delta} \right) \delta^2 d\theta d\mu_r(\psi)
 \end{aligned}$$

Et donc au final nous obtenons bien

$$\begin{aligned}
 & \sum_{l=0}^{K-1} \int_{I_{k-l}} \int_{I_l} \left(4 + \frac{4r\pi}{K\delta} \right) \delta^2 d\theta d\mu_r(\psi) \\
 & \leq \sum_{l=0}^{K-1} \int_{I_{k-l}} \left(4 + \frac{4r\pi}{K\delta} \right) \frac{\pi\delta^2}{K} d\mu_r(\psi) \\
 & \leq \left(4 + \frac{4r\pi}{K\delta} \right) \frac{\pi\delta^2}{K} \\
 & \leq \frac{4\pi\delta^2}{K} + \frac{4r\pi^2\delta}{K^2}
 \end{aligned}$$

qui est le résultat annoncé. \square

Avec ces deux lemmes la démonstration de la proposition 24 est linéaire.

Preuve : Nous partons de la définition de \tilde{Q}_n et de la formule de la proposition 19 sur Q .

$$\begin{aligned}
 \tilde{Q}_{n+1}(i, j, k) &= \int_{I_k} \iint_{X_{i,j}} Q_{n+1}(x, \theta) dx d\theta, \\
 &= \int_{I_k} \iint_{X_{i,j}} \int_{S^1} Q_n(x - r\vec{u}(\varphi), \varphi) d\mu_r(\theta - \varphi) dx d\theta,
 \end{aligned}$$

Les termes sous le signe intégral étant positifs, nous pouvons inverser les intégrales indépendantes,

$$\tilde{Q}_{n+1}(i, j, k) = \int_{I_k} \int_{S^1} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\varphi), \varphi) dx d\mu_r(\theta - \varphi) d\theta.$$

Nous effectuons le changement de variable $\psi = \theta - \varphi$ de manière à pouvoir

effectuer une autre inversion d'intégrales.

$$\begin{aligned}
 \tilde{Q}_{n+1}(i, j, k) &= \int_{I_k} \int_{\mathcal{S}^{1+\theta}} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta - \psi), \theta - \psi) dx d\mu_r(\psi) d\theta, \\
 &= \int_{I_k} \int_{\mathcal{S}^1} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta - \psi), \theta - \psi) dx d\mu_r(\psi) d\theta, \\
 &= \int_{\mathcal{S}^1} \int_{I_k} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta - \psi), \theta - \psi) dx d\theta d\mu_r(\psi), \\
 &= \int_{\mathcal{S}^1} \int_{I_k} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta - \psi), \theta - \psi) dx d\theta d\mu_r(\psi).
 \end{aligned}$$

Nous effectuons maintenant un changement de variable sur θ , la variable de l'intégrale sur I_k .

$$\begin{aligned}
 &= \int_{\mathcal{S}^1} \int_{I_k} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta - \psi), \theta - \psi) dx d\theta d\mu_r(\psi), \\
 &= \int_{\mathcal{S}^1} \int_{I_{k-\psi}} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta), \theta) dx d\theta d\mu_r(\psi).
 \end{aligned}$$

Or $\mathcal{S}^1 = \cup_{k \in \mathbb{Z}/K\mathbb{Z}} I_k$ est une partition du cercle unité. Donc

$$\tilde{Q}_{n+1}(i, j, k) = \sum_{l=0}^{K-1} \int_{I_{k-l}} \int_{I_{k-\psi}} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta), \theta) dx d\theta d\mu_r(\psi).$$

À ce stade nous utilisons le lemme 25,

$$\tilde{Q}_{n+1}(i, j, k) = \sum_{l=0}^{K-1} \int_{I_{k-l}} \int_{I_l} \iint_{X_{i,j}} Q_n(x - r\vec{u}(\theta), \theta) dx d\theta d\mu_r(\psi) + R_1(\delta, K),$$

puis le lemme 26,

$$\begin{aligned}
 \tilde{Q}_{n+1}(i, j, k) &= \sum_{l=0}^{K-1} \int_{I_{k-l}} \sum_{(a,b) \in \mathbb{Z}^2} \omega(i, j, a, b, l) \tilde{Q}_n(a, b, l) d\mu_r(\psi) + R_1(\delta, K) + R_2(\delta, K), \\
 &= \sum_{l=0}^{K-1} \sum_{(a,b) \in \mathbb{Z}^2} \omega(i, j, a, b, l) \tilde{Q}_n(a, b, l) \mu_r(I_{k-l}) + R_1(\delta, K) + R_2(\delta, K), \\
 &= \sum_{l=0}^{K-1} \sum_{(a,b) \in \mathbb{Z}^2} \omega(i, j, a, b, l) \tilde{Q}_n(a, b, l) \tilde{\mu}_r(k-l) + R_1(\delta, K) + R_2(\delta, K).
 \end{aligned}$$

Ce qui montre bien le résultat attendu en posant $R = R_1 + R_2$. \square

Valeurs par défaut des paramètres	
r	16
K	24
s	3
λ	5

TAB. 3.1 – Valeurs par défaut pour le modèle $\tilde{\mathcal{M}}_r$. Ce tableau contient les valeurs utilisées par défaut pour les paramètres du modèle $\tilde{\mathcal{M}}_r$ dans les expériences des figures 3.4 à 3.8.

Nous avons deux possibilités pour mettre en pratique cette formule numériquement. La première consiste à parcourir les points (i, j, k) et à calculer les intégrales correspondantes en parcourant les points (a, b, l) . La deuxième possibilité est de parcourir les points (a, b, l) et de rajouter la quantité correspondante aux points (i, j, k) . Cette deuxième approche est un modèle de propagation de courbe, plus efficace qui consiste à prendre toutes les extrémités possibles et à les prolonger avec une pondération correspondant à leur probabilité en utilisant le générateur de courbe 21.

Ensuite la probabilité de sortie \tilde{P}_r est calculée par rapport à \tilde{Q} de la même manière que P_r est calculée par rapport à Q .

3.1.3 Calibration des paramètres

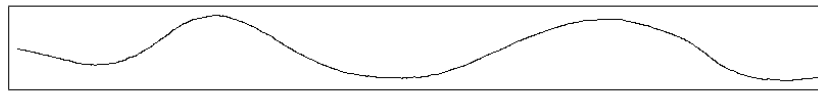
Nous avons différents paramètres à régler. Les deux paramètres de l'algorithme sont le rayon des courbes r , et le nombre d'angles discrets K . Cependant pour accélérer les calculs nous avons ajouté un paramètre, un facteur d'échelle pour diminuer la taille du voisinage $V(\mathcal{C}, \varepsilon)$. Mais il y a un autre paramètre, le paramètre λ de la loi de Laplace.

Il y a donc deux paramètres pour décrire le modèle, le rayon r et le paramètre λ de la loi de Laplace, et deux paramètres numériques, le nombre d'angles discrets K et le facteur d'échelle s .

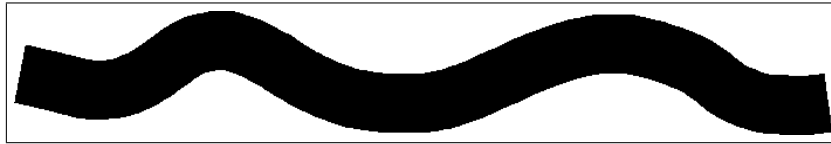
Pour calibrer les paramètres, nous effectuons des expériences à partir d'une courbe représentée dans la figure 3.3(a). Les figures 3.3(b) à 3.3(f) montrent le voisinage construit à partir de cette courbe pour des échelles de 1 : 1 à 1 : 5.

La table 3.1 montre les valeurs par défaut utilisées pour les paramètres lors des expériences.

Le rayon. Le rayon r est un vrai paramètre, il correspond à l'échelle à laquelle nous nous plaçons pour analyser les lignes de niveau. Il ne faut pas prendre r trop petit car dans ce cas cela signifie que nous observons sur les



(a) Courbe \mathcal{C} .



(b) Voisinage à l'échelle 1 : 1



(c) Voisinage à l'échelle 1 : 2



(d) Voisinage à l'échelle 1 : 3



(e) Voisinage à l'échelle 1 : 4



(f) Voisinage à l'échelle 1 : 5

FIG. 3.3 – Un voisinage à différentes échelles. Le voisinage $V(\mathcal{C}, \varepsilon)$ de la courbe \mathcal{C} peut être utilisé à différentes échelles. La courbe \mathcal{C} est la courbe du dessin du haut. Au-dessous sont dessinés les voisinages $V(\mathcal{C}, \varepsilon)$ de cette courbe pour un ε fixé avec des échelles allant de 1 : 1 à 1 : 5.

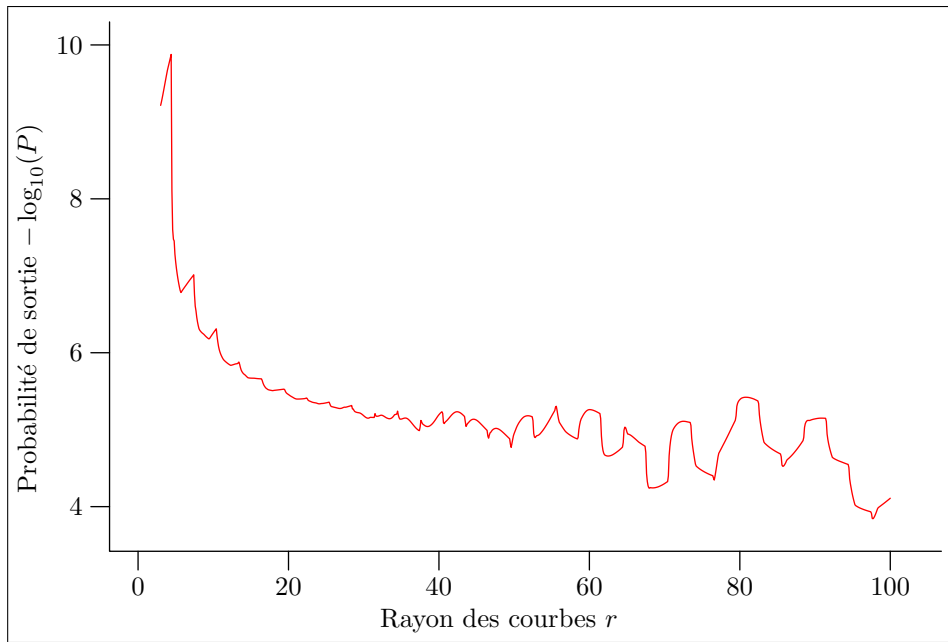


FIG. 3.4 – Influence de r sur le modèle $\tilde{\mathcal{M}}_r$. La courbe représentée ci-dessus montre les différentes valeurs prises par la probabilité de sortie calculée avec le modèle discret $\tilde{\mathcal{M}}_r$ lorsque le rayon r des courbes varie. Les valeurs obtenues sont fiables pour des valeurs de r comprises entre 10 et 50. Au-delà les valeurs de la probabilité de sortie ne sont pas fiables. Les valeurs prises par les autres paramètres sont indiquées sur le tableau 3.1.

lignes de niveau des points très rapprochés pour observer le changement de direction. Et si les points sont trop près l'hypothèse tacite d'indépendance des points n'est pas du tout respectée. Dans le cas extrême de courbes de niveau sur une image sans interpolation (image constante sur chaque pixel) avec un rayon r égal à un pixel il n'y a que quatre orientations possibles. La figure 3.4 montre la courbe des valeurs prises par la probabilité de sortie P lorsque r varie de 2,2 jusqu'à 100. En ordonnée nous figurons $-\log_{10}(P)$ de sorte que 6 corresponde donc à la valeur 10^{-6} pour P . Sur ce graphique il apparaît clairement que pour les petites valeurs de r la probabilité de sortie n'est pas du tout fiable. De même lorsque r est supérieur à 50 des artefacts numériques importants apparaissent et s'amplifient. Entre 10 et 50 il y a une plage raisonnable pour choisir r . Par la suite nous prenons $r = 16$.

Pour cette expérience nous avons toujours pris la même probabilité de changement de direction. En principe la probabilité de changement de direction dépend du rayon r . En effet si nous calculons empiriquement cette probabilité, l'échantillonnage dépend de ce paramètre r . La figure 3.5(a) montre deux courbes, la courbe continue correspond à l'expérience de la figure 3.4

et la courbe en pointillés correspond aux calculs effectués avec des probabilités de changement de direction calculées empiriquement sur les courbes de la figure 3.5(b).

Le nombre d'angles. La courbe de la figure 3.6 montre l'influence du nombre d'angles discrets K sur le calcul. Comme prévu plus K est grand, plus précis est le calcul et ceci apparaît bien sur la courbe qui devient vite presque constante. À partir de cette valeur n'importe quelle valeur de K convient. Pour rendre les calculs plus rapides, il est possible de prendre K parmi les premières valeurs stabilisées. En fait la valeur de K pour laquelle la probabilité se stabilise correspond à peu près à la valeur de K égale au nombre de pixels sous le cercle de rayon r .

L'échelle. La courbe pour le facteur d'échelle s est donné par la figure 3.7. Pour le facteur d'échelle, lorsque le domaine diminue, le nombre de pixels dans le voisinage $V(\mathcal{C}, \varepsilon)$ par rapport au nombre de pixels sur le bord diminue. En effet, pour faire une approximation supposons que le voisinage est un rectangle de longueur L et de largeur ε . Posons alors

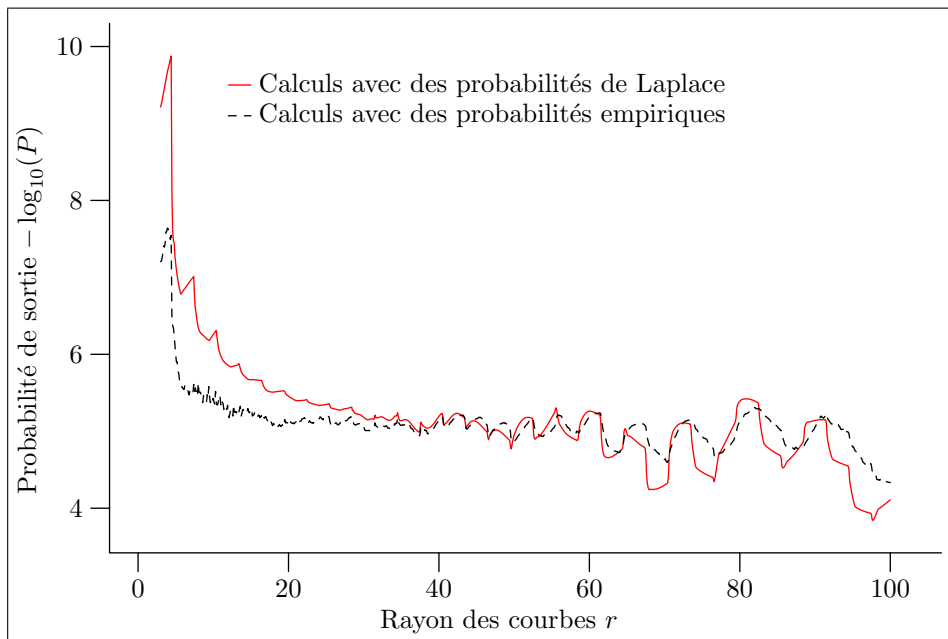
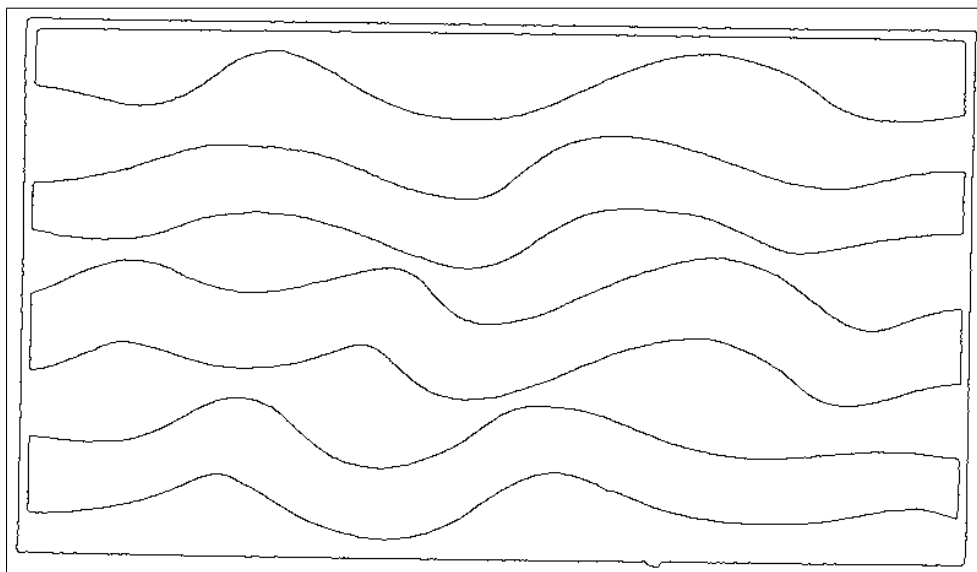
$$N = L\varepsilon, \quad N_b = 2(L + \varepsilon), \quad N_i = N - N_b.$$

Ainsi N est la surface du voisinage et correspond donc au nombre de pixels du voisinage, N_b est le périmètre du voisinage et correspond alors au nombre de pixels à la frontière du voisinage et finalement N_i est le nombre de pixels à l'intérieur du voisinage. Posons maintenant $\tilde{L} = L/s$, et $\tilde{\varepsilon} = \varepsilon/s$ les dimensions du voisinage après changement d'échelle. En définissant de la même manière \tilde{N} , \tilde{N}_b et \tilde{N}_i nous avons

$$\frac{\tilde{N}_i}{\tilde{N}_b} = 1 - s \frac{N_b}{N},$$

et ce rapport diminue donc bien lorsque s augmente (au moins tant que $s \leq N/N_b$, ensuite le voisinage ne contient plus de pixels). Cette augmentation du rapport entre les pixels du bord et ceux à l'intérieur augmente la probabilité pour les courbes de sortir du voisinage avant d'atteindre Γ_f . Ceci explique donc la diminution de la probabilité de sortie lorsque le facteur d'échelle augmente. Pour s vraiment grand les résultats ne sont vraiment plus fiables. Ici l'intervalle $[1; 5]$ semble correct. Nous prenons $s = 3$.

La loi de Laplace. Un dernier paramètre que nous pouvons faire varier est le paramètre de la probabilité de Laplace utilisée. L'influence de ce paramètre λ est montrée sur la courbe de la figure 3.8. Sur ce graphe figure une ligne en pointillés, c'est la valeur calculée pour la probabilité de sortie avec la probabilité calculée empiriquement. Nous prenons $\lambda = 5$.

(a) Influence de r pour chacun des lois.

(b) Courbes utilisées pour le calcul empirique.

FIG. 3.5 – Comparaison loi de Laplace/loi empirique. La courbe représentée en haut en trait plein est la même que la courbe de la figure 3.4. La courbe en pointillés est la courbe obtenue de la même manière que la précédente sauf qu'au lieu d'utiliser une loi de Laplace de paramètre 5 pour la probabilité de changement de direction nous avons cette fois calculé empiriquement une loi de changement de direction suivant le calcul 15 sur les courbes de la figure du bas.

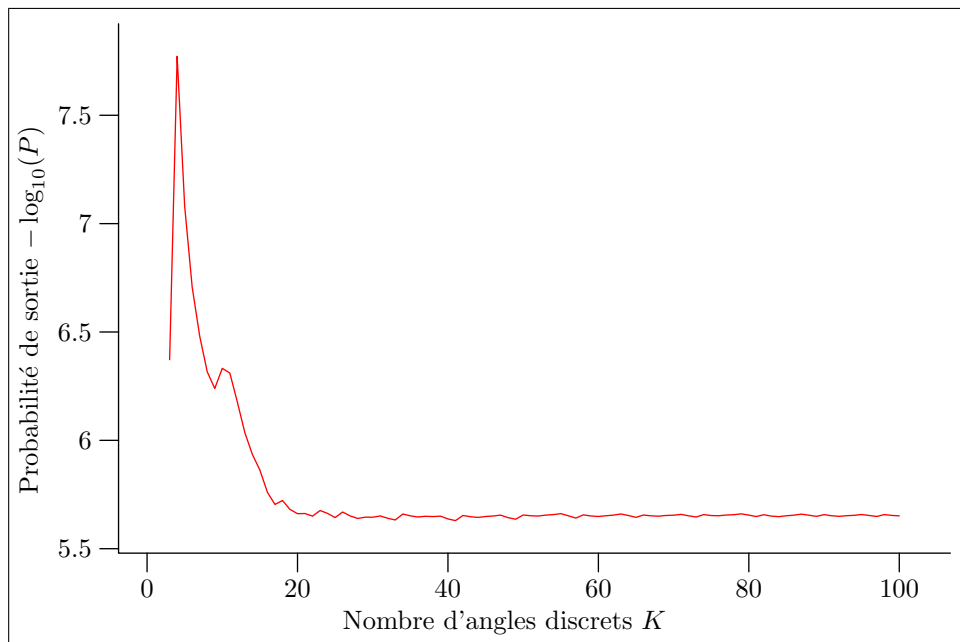


FIG. 3.6 – Influence de K sur le modèle $\tilde{\mathcal{M}}_r$. La courbe ci-dessus montre la probabilité de sortie obtenue avec le modèle $\tilde{\mathcal{M}}_r$ en fonction du nombre d'angles discrets K . Sur cette courbe nous observons une stabilisation autour d'une valeur fixe dès que K dépasse 20. Les valeurs des autres paramètres sont celles indiquées dans le tableau 3.1.

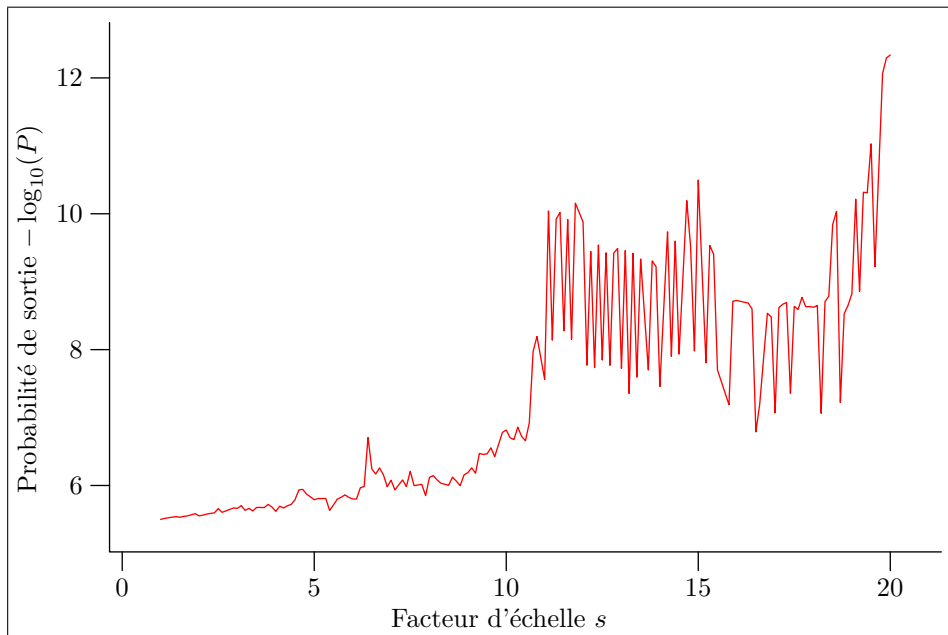


FIG. 3.7 – Influence de s sur le modèle $\tilde{\mathcal{M}}_r$. La courbe montre la probabilité de sortie avec ce modèle en fonction du facteur d'échelle s . Le comportement de la probabilité de sortie est bon lorsque s ne dépasse pas 5 ; au delà les valeurs prises sont très instables. Pour les autres paramètres du modèle $\tilde{\mathcal{M}}_r$ leurs valeurs sont celles du tableau 3.1.

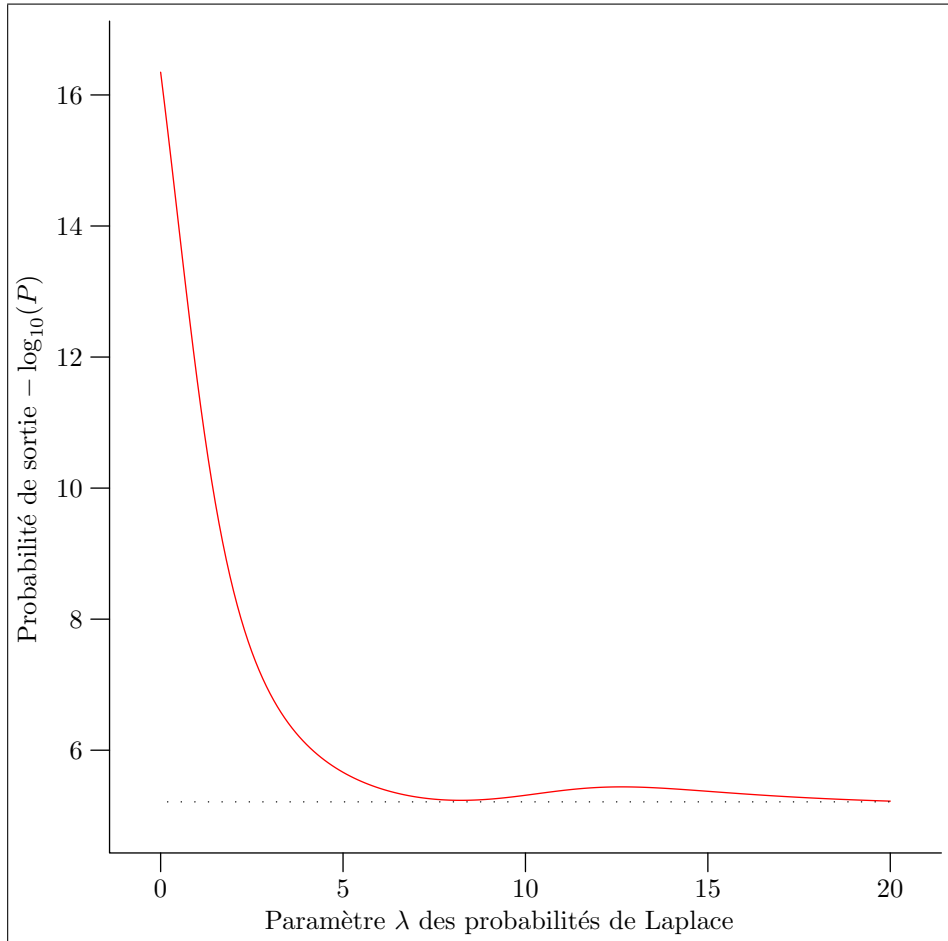


FIG. 3.8 – Influence de λ sur le modèle $\tilde{\mathcal{M}}_r$. La courbe de la probabilité de sortie en fonction du paramètre λ de la loi de Laplace est représentée ci-dessus en trait plein. La loi de Laplace est la loi choisie pour la probabilité de changement de direction. Lorsque λ tend vers 0 le changement de direction est de plus en plus uniforme et les courbes générées dans le voisinage sont de plus en plus des réalisations de mouvement Brownien. Dans de telles conditions il est de plus en plus improbable que les courbes suivent les méandres du tube formé par le voisinage. Il n'est donc pas étonnant que la probabilité tende vers 0 lorsque λ tend vers 0. Les autres paramètres utilisés pour le modèle $\tilde{\mathcal{M}}_r$ sont indiqués sur le tableau 3.1.

3.2 Méthode de Monte-Carlo

Au lieu de calculer empiriquement la probabilité de sortie il est possible de l'estimer. Cette méthode est plus rapide et il est alors possible de garder le caractère continu des courbes simulées même si le domaine est discrétisé. Comme la probabilité à estimer est très faible, il faut *a priori* tester un très grand nombre de courbes avant d'en trouver une qui va ressortir du tube à travers Γ_f et avoir une probabilité estimée non nulle. Cependant, nous pouvons contourner ce problème par un procédé de duplication.

3.2.1 Estimateur

Soit N_0 le nombre d'extrémités de courbes prises au hasard dans S_d . Par exemple sur la figure 3.9(a) nous avons pris au hasard quatre extrémités de courbes sur Γ_d . Nous laissons évoluer ces N_0 courbes pour une étape. Nous comptons alors les P_0 courbes qui sortent du domaine par Γ_f , les D_0 autres courbes qui sortent autrement du tube et il reste donc dans le domaine $N'_0 = N_0 - P_0 - D_0$ courbes. La figure 3.9(b) montre le résultat de cette évolution pour les 4 courbes tirées.

Pour compenser les courbes perdues et maintenir un nombre constant de courbes dans le tube, nous "dupliquons" artificiellement certaines des courbes restantes. Nous choisissons donc aléatoirement (tirage avec remise) $N''_0 = N_0 - N'_0$ extrémités de courbes parmi les N'_0 qui sont restées dans le domaine. Nous avons donc à nouveau $N'_0 + N''_0 (= N_0)$ courbes au début de l'étape suivante. Sur la figure 3.10(a) nous avons tiré une extrémité en double pour compenser la perte d'une courbe, c'est celle marquée avec $\times 2$.

Le procédé est le même à l'étape n . Supposons être à la n -ème étape avec N_n extrémités de courbes à suivre. Nous laissons évoluer ces courbes pour une étape. Soient P_n le nombre de courbes sortant du tube par Γ_f , D_n les autres courbes sortantes et $N'_n = N_n - P_n - D_n$ les courbes restant dans le domaine. Nous tirons parmi ces N'_n courbes qui sont encore dans le domaine $N''_n = N_0 - N'_n$ courbes. Nous avons donc $N_{n+1} = N'_n + N''_n (= N_0)$.

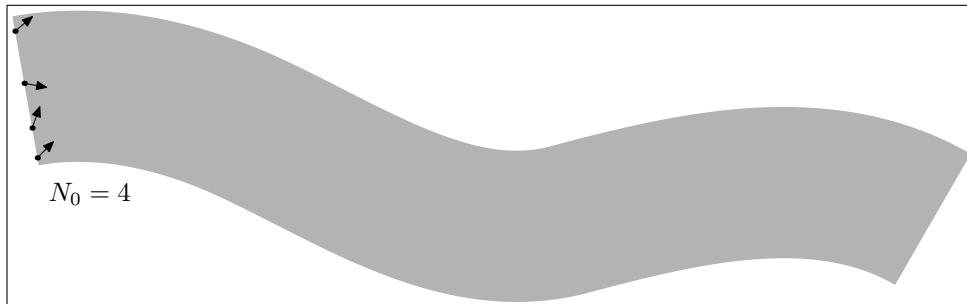
Avec cette méthode de simulation de courbes, nous avons une estimation de la probabilité P_r ,

PROPOSITION 27 :

L'estimateur de la probabilité P_r est

$$P_{mc} = \sum_{n=0}^{\infty} \left(\frac{P_n}{N_n} \prod_{k=0}^{n-1} \frac{N'_k}{N_k} \right)$$

Preuve : En effet la probabilité estimée de rester dans $V(\mathcal{C}, \varepsilon)$ lors de la k -ème



(a) Tirage des extrémités avec $N_0 = 4$.



(b) Résultat d'une première étape.

FIG. 3.9 – L'étape initiale de l'estimation de Monte-Carlo. Au début de l'estimation, nous tirons des extrémités de courbe au hasard sur l'extrémité Γ_d du voisinage (tirage selon la probabilité Q_0 en fait). Sur le dessin du dessus nous avons choisi quatre extrémités de départ. Nous allons donc suivre quatre courbes à travers le voisinage. Nous prolongeons les extrémités selon le modèle \mathcal{M}_r en utilisant le générateur 12. Le résultat est illustré sur le dessin du bas.

étape est N'_k/N_k et ainsi celle d'y rester durant les $(n-1)$ premières étapes est

$$\prod_{k=0}^{n-1} \frac{N'_k}{N_k}.$$

La probabilité de sortir à la n -ème étape est donc

$$\left(\prod_{k=0}^{n-1} \frac{N'_k}{N_k} \right) \frac{P_n}{N_n}.$$

L'estimateur que nous cherchons est donc bien

$$P_{mc} = \sum_{n=0}^{\infty} \left(\prod_{k=0}^{n-1} \frac{N'_k}{N_k} \right) \frac{P_n}{N_n}, \quad (3.2)$$

comme attendu. □

3.2.2 Calibration des paramètres

En prenant $N_n = N_0$ le seul paramètre de cet estimateur est alors le nombre N_0 de courbes suivies dans le domaine. La probabilité de sortie doit être indépendante de ce paramètre. Nous avons effectué plusieurs simulations avec le voisinage de la figure 3.3(a). Les résultats sont sur la figure 3.11. La courbe montre clairement la très faible dépendance du résultat par rapport au nombre de courbes N_0 dès que celui-ci est supérieur à quelques centaines.

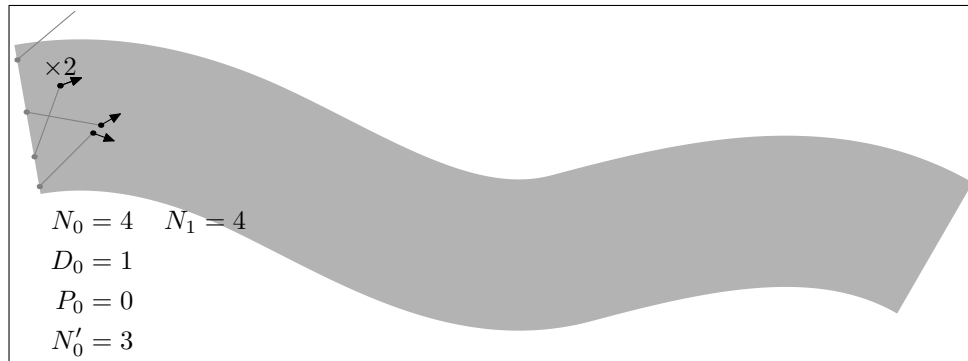
Nous avons ensuite effectué les mêmes tests que pour la méthode directe pour pouvoir comparer les résultats. Pour ces tests nous avons pris $N_0 = 300$. Les résultats sont dans les figures 3.12 à 3.15. Ces courbes montrent que les courbes sont très proches, en tout cas pour des paramètres restants dans un domaine raisonnable.

La figure 3.16 montre des courbes simulées pour une estimation dans le voisinage de la figure 3.3(a) avec $N_0 = 10$.

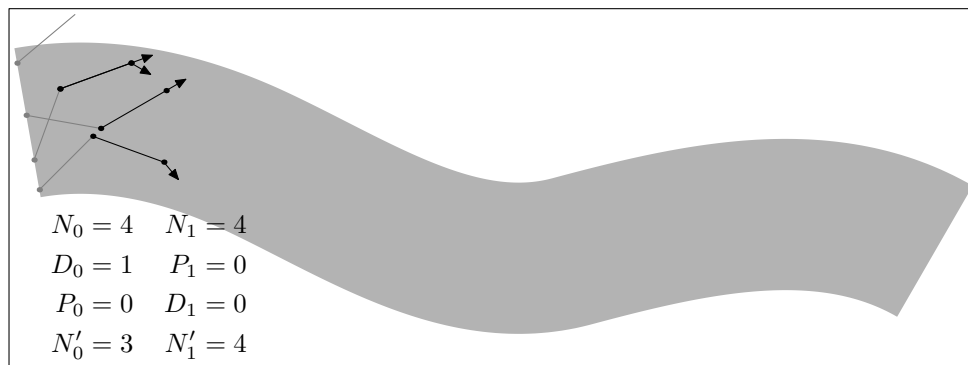
3.3 Équation aux dérivées partielles

3.3.1 Obtention d'une EDP

Nous allons maintenant chercher un modèle limite \mathcal{M}_0 en faisant tendre r vers 0 dans les modèles \mathcal{M}_r . La limite des courbes polygonales de \mathcal{C}_r est l'ensemble des courbes continues \mathcal{C} . Pour calculer la probabilité sur les voisinages $\mathcal{V}(\mathcal{C}, \varepsilon)$, nous partons encore une fois de la probabilité de présence.



(a) Tirage d'une nouvelle extrémité.



(b) Résultat d'une deuxième étape.

FIG. 3.10 – Une seconde itération de l'estimation de Monte-Carlo. À la fin de l'étape précédente présentée sur la figure 3.9, il ne restait que trois extrémités de courbe dans le voisinage. Pour repartir avec à nouveau quatre extrémités dans cette deuxième étape, nous choisissons au hasard une des extrémités dans le voisinage et nous la dupliquons. Ce choix est présenté dans le dessin du haut, l'extrémité dupliquée est marquée d'un $\times 2$. Le résultat d'une deuxième étape de suivi des courbes est sur le dessin du bas. L'extrémité dupliquée a donné lieu à deux courbes qui suivent le même chemin lors de cette étape. Cependant à la fin deux tirages différents d'un changement de direction sont effectués pour chacune d'entre elles. C'est pour cette raison qu'au bout du chemin nous avons deux directions différentes, une par courbe.

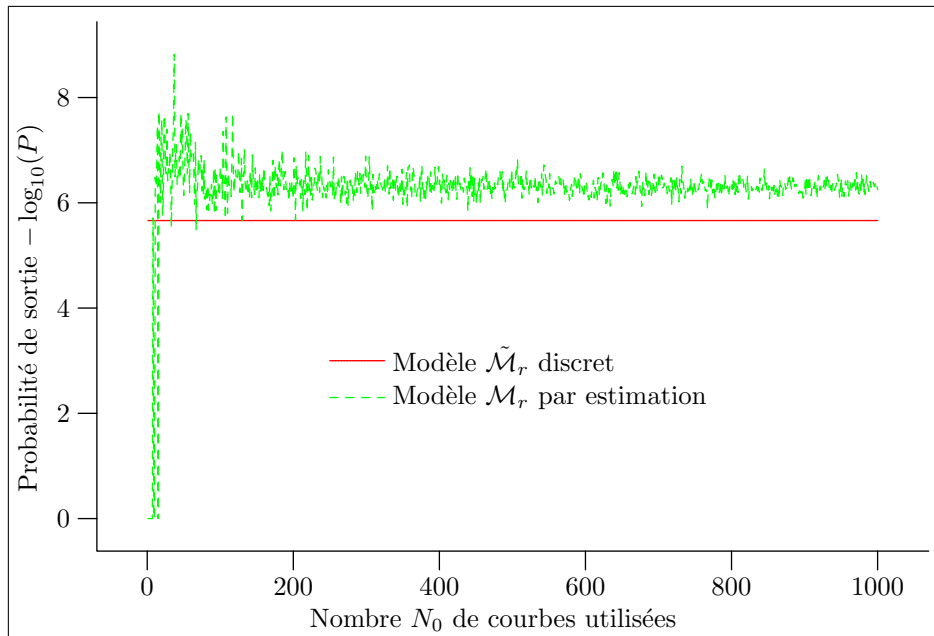


FIG. 3.11 – Influence de N_0 sur le modèle \mathcal{M}_r . Ci-dessus est représentée la courbe de la probabilité de sortie estimée pour le modèle \mathcal{M}_r en fonction du nombre de courbes utilisées pour l'estimation de Monte-Carlo. À chaque N_0 correspond une réalisation de l'algorithme. Nous devinons sur cette courbe que l'écart-type du résultat va bien en diminuant lorsque le nombre de courbes suivies augmente. Nous voyons donc que quelques centaines de courbes suffisent pour avoir une bonne estimation de la probabilité de sortie. Par la suite nous prenons $N_0 = 300$. Les autres valeurs des paramètres sont pris comme pour le modèle $\tilde{\mathcal{M}}_r$ sur le tableau 3.1.

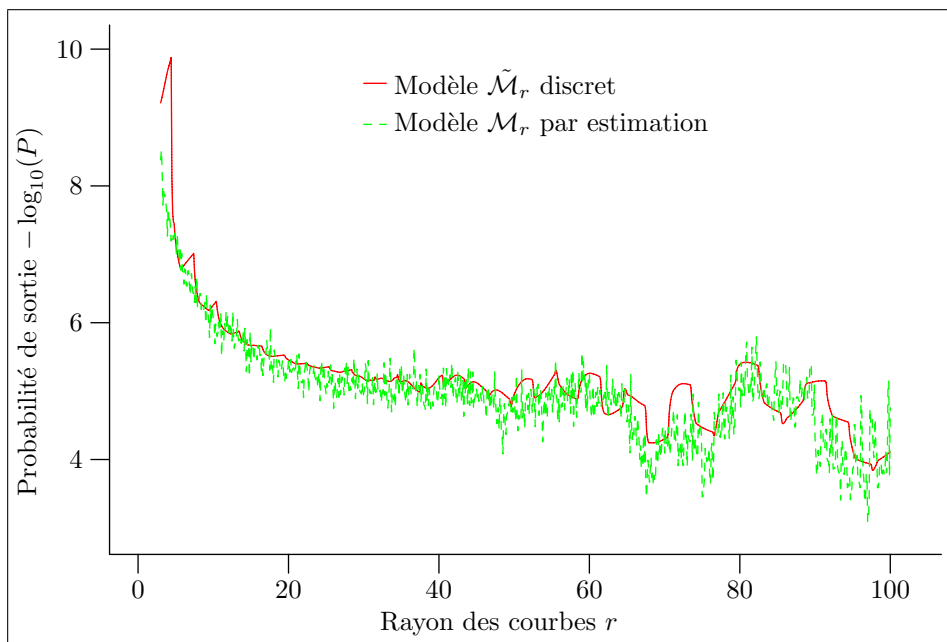


FIG. 3.12 – Influence de r sur le modèle \mathcal{M}_r . Cette fois ci nous étudions l’influence du rayon r des courbes sur la probabilité de sortie. Nous pouvons faire les mêmes remarques que pour le calcul avec le modèle discret $\tilde{\mathcal{M}}_r$ présenté sur la figure 3.4 tant les résultats sont proches.

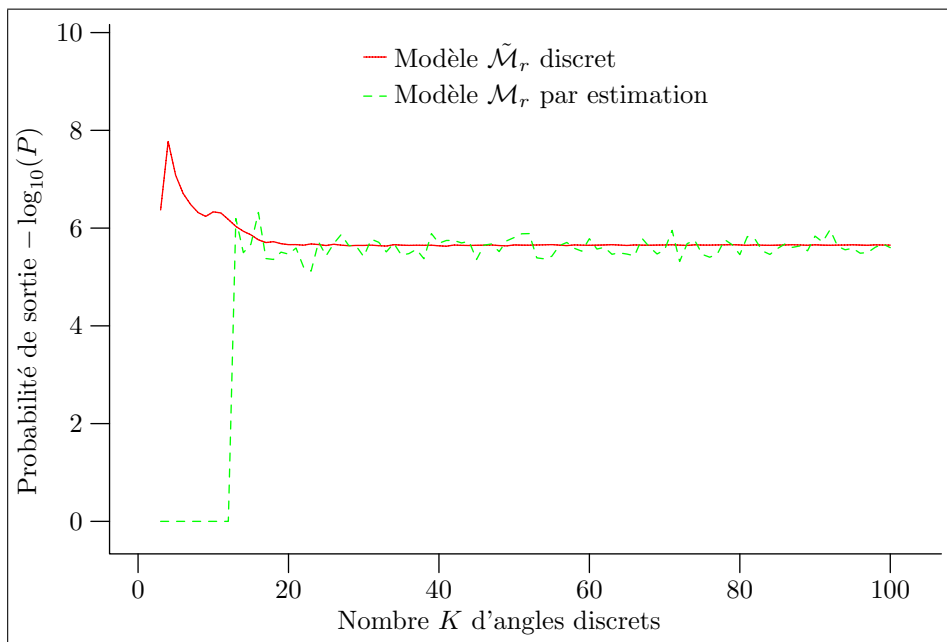


FIG. 3.13 – Influence de K sur le modèle \mathcal{M}_r . Là encore le résultat est comparable avec ceux obtenus avec le modèle discret $\tilde{\mathcal{M}}_r$. Toutes les valeurs au-dessus de 20 conviennent. Par contre pour les faibles valeurs de K aucune courbe n'arrive à sortir du voisinage et l'estimateur donne donc une probabilité nulle. Ceci pourrait être amélioré en augmentant le nombre N_0 de courbes simulées. Sur cette échelle logarithmique nous avons mis la valeur 0 dans ce cas.

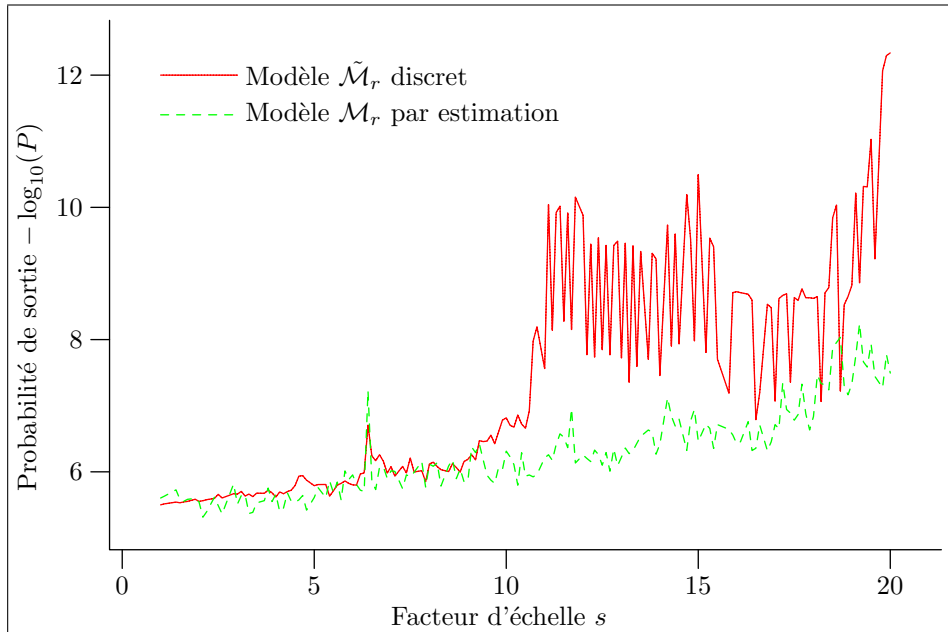


FIG. 3.14 – Influence de s sur le modèle \mathcal{M}_r . Une fois de plus l'influence du facteur d'échelle est comparable à l'influence sur le modèle discret $\tilde{\mathcal{M}}_r$. Nous pouvons cependant noter que le comportement de l'estimateur du modèle \mathcal{M}_r est plus robuste que le modèle discret $\tilde{\mathcal{M}}_r$. En effet l'estimateur est moins sujet aux artefacts de discrétisation et se comporte mieux pour les grandes valeurs du facteur d'échelle. Malgré cela il faut prendre un facteur d'échelle modéré, inférieur à 5. Ce meilleur comportement pour les échelles très grossières du voisinage $V(\mathcal{C}, \varepsilon)$ montre bien que c'est la pixellisation du bord qui pose problème avec le modèle $\tilde{\mathcal{M}}_r$, et non pas la discrétisation des courbes.

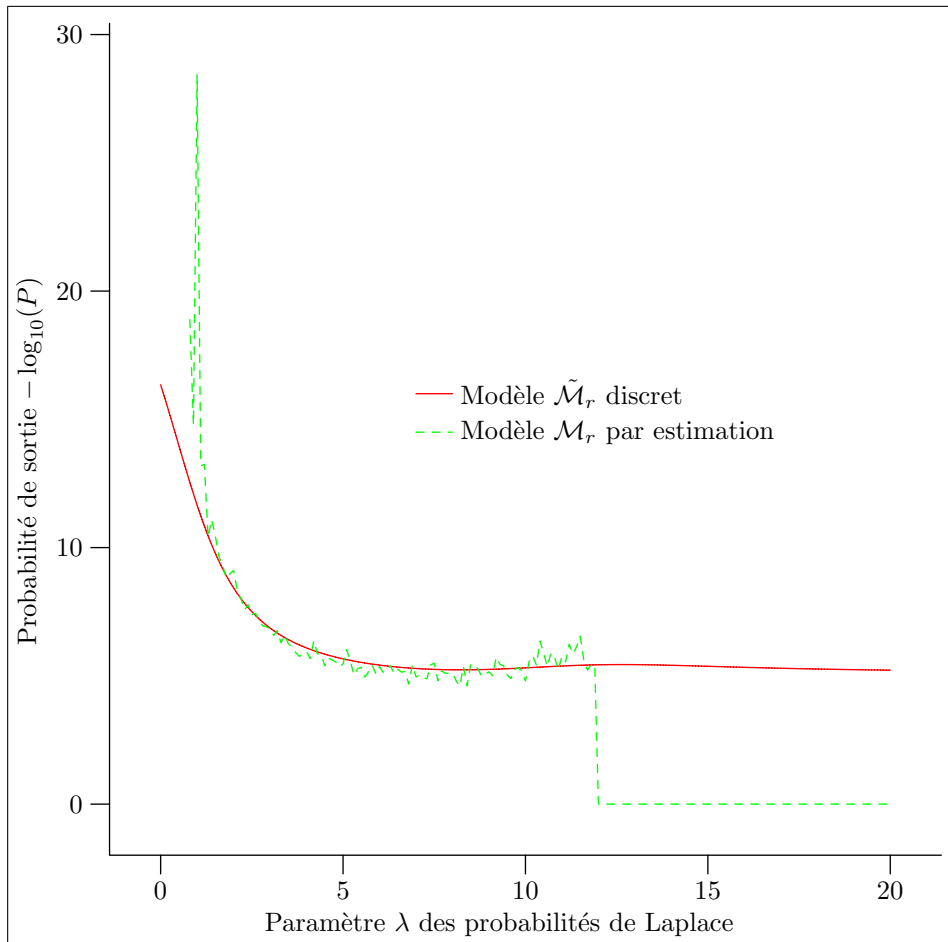


FIG. 3.15 – Influence de λ sur le modèle \mathcal{M}_r . L'influence du paramètre de la loi de Laplace utilisée sur ce modèle est identique à son influence sur le modèle discret $\tilde{\mathcal{M}}_r$ pour des valeurs de λ prises entre 2 et 12. En dehors de cette plage il y a trop peu de courbes qui parviennent à suivre le voisinage et l'estimateur n'est plus fiable.

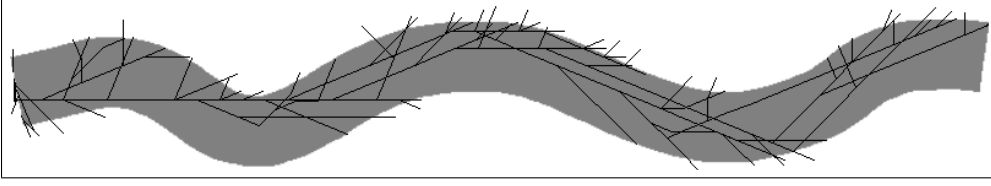


FIG. 3.16 – Courbes générées par l'estimateur. Le dessin ci-dessus montre les courbes générées par l'estimateur pour estimer la probabilité de sortie selon le modèle \mathcal{M}_r . Pour des raisons de lisibilité, nous avons pris pour cette estimation $N_0 = 10$ courbes au départ, en pratique nous prenons $N_0 = 300$ ce qui conduit à une meilleure exploration du tube. Au début de chaque étape nous tirons des extrémités de courbes en dupliquant certaines extrémités qui sont encore dans le voisinage de façon à toujours avoir dix extrémités au début de chaque étape.

Nous partons de la formule de la proposition 19 sur la probabilité de présence Q du modèle \mathcal{M}_r ,

$$Q_{n+1}(x, \theta) = \int_{S^1} Q_n(x - r\vec{u}(\varphi), \varphi) d\mu_r(\theta - \varphi).$$

Nous introduisons maintenant la somme des probabilités de présence pour tous les moments n

$$q_r(x, \theta) = \sum_{n=0}^{+\infty} Q_n(x, \theta),$$

comme cette quantité (ainsi que les q_n d'ailleurs) dépend de r , nous mettons r en indice. En sommant la formule précédente pour tout n on trouve alors

$$q_r(x, \theta) - Q_0(x, \theta) = \int_{S^1} q_r(x - r\vec{u}(\varphi), \varphi) d\mu_r(\theta - \varphi). \quad (3.3)$$

Or Q_0 est nulle partout sauf sur une partie du bord de $V(\mathcal{C}, \varepsilon)$, donc à l'intérieur de $V(\mathcal{C}, \varepsilon)$ nous trouvons l'équation

$$q_r(x, \theta) = \int_{S^1} q_r(x - r\vec{u}(\varphi), \varphi) d\mu_r(\theta - \varphi). \quad (3.4)$$

Nous cherchons maintenant une EDP vérifiée par une solution de cette équation intégrale. Ceci nous conduit à faire des hypothèses sur les probabilités de changement de direction μ_r .

PROPOSITION 28 :

Si $(\mu_r)_{r>0}$ est une famille de mesures sur \mathcal{S}^1 telle qu'il existe $\gamma > 0$ et une mesure μ vérifiant,

$$\forall r > 0, \quad r^\gamma d\mu_r(r^\gamma \varphi) = d\mu(\varphi), \quad (3.5)$$

– la mesure μ est une probabilité,

$$\int_{S^1} d\mu(\varphi) = 1, \quad (3.6)$$

– de moyenne nulle

$$\int_{S^1} \varphi d\mu(\varphi) = 0, \quad (3.7)$$

– et de variance $\alpha > 0$

$$\int_{S^1} \varphi^2 d\mu(\varphi) = \alpha. \quad (3.8)$$

et si $(x, \theta, r) \mapsto q_r(x, \theta)$ est une fonction admettant localement des dérivées partielles par rapport à x et θ d'ordre 3 continues au voisinage du point $(x, \theta, 0)$ et vérifiant l'équation intégrale 3.4 en (x, θ) pour r assez proche de 0, alors

$$-r \nabla_x q_r \cdot \vec{u}(\theta) + \frac{r^{2\gamma} \alpha}{2} \frac{\partial^2 q_r}{\partial \theta^2} = o_{r \rightarrow 0}(r^{\min(1, 2\gamma)}), \quad (3.9)$$

au point (x, θ) .

Preuve : Afin d'expliciter la dépendance en r de μ_r , nous utilisons la règle d'échelle sur la famille $(\mu_r)_r$ en posant le changement de variable $\theta - \varphi = r^\gamma \psi$,

$$\begin{aligned} q_r(x, \theta) &= \int_{S^1} q_r(x - r\vec{u}(\varphi), \varphi) d\mu_r(\theta - \varphi), \\ &= \int_{S^1} q_r(x - r\vec{u}(\theta - r^\gamma \psi), \theta - r^\gamma \psi) r^\gamma d\mu_r(r^\gamma \psi), \\ &= \int_{S^1} q_r(x - r\vec{u}(\theta - r^\gamma \psi), \theta - r^\gamma \psi) d\mu(\psi). \end{aligned} \quad (3.10)$$

Nous voulons maintenant utiliser un développement limité de q_r dans l'intégrale. Pour cela nous regardons le développement limité de q_s au point $(x - r\vec{u}(\theta - r^\gamma \psi), \theta - r^\gamma \psi)$

$$\begin{aligned} q_s(x - r\vec{u}(\theta - r^\gamma \psi), \theta - r^\gamma \psi) &= q_s(x, \theta) - r \nabla_x q_s(x, \theta) \cdot \vec{u}(\theta) + r \varepsilon_s(r) \\ &\quad - r^\gamma \psi \frac{\partial q_s}{\partial \theta} + \frac{r^{2\gamma} \psi^2}{2} \frac{\partial^2 q_s}{\partial \theta^2}(x, \theta) + r^{2\gamma} \eta_s(r^{2\gamma}), \end{aligned} \quad (3.11)$$

avec ε_s et η_s telles que

$$|\varepsilon_s(r)| \leq r M_1 \quad \text{et} \quad |\eta_s(r^{2\gamma})| \leq r^{2\gamma} M_2,$$

Les constantes M_1 et M_2 sont en fait des bornes supérieures des dérivées de q_s d'ordre 2 et 3 par rapport à x et θ . En prenant encore une fois la borne supérieure de ces quantités sur un voisinage de 0 par rapport à s , nous pouvons supposer que M_1 et M_2 ne dépendent pas de s . Ceci signifie que le reste dans le

développement limité (3.11) tend uniformément vers 0 par rapport à s lorsque r tend vers 0. Nous pouvons donc dire maintenant que nous avons

$$q_r(x - r\vec{u}(\theta - r^\gamma\psi), \theta - r^\gamma\psi) = q_r(x, \theta) - r\nabla_x q_r(x, \theta) \cdot \vec{u}(\theta) + o_{r \rightarrow 0}(r) - r^\gamma\psi \frac{\partial q_r}{\partial \theta} + \frac{r^{2\gamma}\psi^2}{2} \frac{\partial^2 q_r}{\partial \theta^2}(x, \theta) + o_{r \rightarrow 0}(r^{2\gamma}), \quad (3.12)$$

En introduisant ce développement limité (3.12) dans l'équation (3.10) nous obtenons

$$q_r(x, \theta) = \int_{S^1} q_r(x, \theta) d\mu(\psi) - r\nabla_x q_r(x, \theta) \cdot \vec{u}(\theta) \int_{S^1} d\mu(\psi) + o_{r \rightarrow 0}(r) - r^\gamma \frac{\partial q_r}{\partial \theta}(x, \theta) \int_{S^1} \psi d\mu(\psi) + \frac{r^{2\gamma}}{2} \frac{\partial^2 q_r}{\partial \theta^2}(x, \theta) \int_{S^1} \psi^2 d\mu(\psi) + o_{r \rightarrow 0}(r^{2\gamma}). \quad (3.13)$$

En prenant en compte les égalités (3.6), (3.7) et (3.8), l'équation finale obtenue est

$$-r\nabla_x q_r \cdot \vec{u}(\theta) + \frac{r^{2\gamma}\alpha}{2} \frac{\partial^2 q_r}{\partial \theta^2} = o_{r \rightarrow 0}(r^{\min(1, 2\gamma)}),$$

d'où la proposition. □

Il nous reste maintenant à discuter selon les valeurs de γ .

3.3.2 Discussion par rapport à γ

On se pose maintenant la question de savoir s'il existe une équation limite quand r tend vers 0, c'est-à-dire un modèle continu limite des modèles discrets. On suppose que les fonctions q_r tendent vers une fonction q lorsque r tend vers 0. Trois différents cas se présentent en fonction de la position de γ par rapport à $1/2$.

– si $\gamma > 1/2$

Dans ce cas l'équation 3.9 devient

$$\nabla_x q \cdot \vec{u}(\theta) = 0$$

en faisant tendre r vers 0. Si q vérifie cette équation sur tout le voisinage $V(\mathcal{C}, \varepsilon)$, avec une valeur sur le bord fixe $q(x, \theta) = q_0(x, \theta)$ alors dans ce cas

$$q(x, \theta) = \begin{cases} q_0(x_0, \theta) & \text{s'il existe } \lambda \geq 0 \text{ tel que} \\ & x = x_0 + \lambda\vec{u}(\theta) \text{ et} \\ & \text{le segment } [x_0; x] \text{ est inclus} \\ & \text{dans } V(\mathcal{C}, \varepsilon), \\ 0 & \text{sinon.} \end{cases}$$

La figure 3.17 montre le résultat de la simulation numérique par l'EDP d'évolution

$$\frac{\partial q}{\partial t} = \nabla_x q \cdot \vec{u}(\theta). \quad (3.14)$$

En chaque point x , l'intensité de l'image correspond à la valeur de l'intégrale de $q(x, \theta)$ par rapport à θ . Dans le voisinage, les couleurs les plus claires correspondent aux probabilités de présence les plus faibles. La probabilité de présence n'arrive pas à se propager en suivant les méandres du voisinage et elle finit par être nulle sur tout le voisinage.

En fait le cas $\gamma > 1/2$ peut être comparé au modèle discret en prenant pour la probabilité de changement de direction une loi de Laplace de paramètre $\lambda = +\infty$.

– si $\gamma < 1/2$

Cette fois-ci l'équation est

$$\frac{\partial^2 q}{\partial \theta^2} = 0.$$

Donc q ne dépend pas de θ , il n'y a aucune contrainte sur x . Il existe une infinité de solutions à cette EDP. Bref, ce cas n'a pour nous aucun intérêt.

– si $\gamma = 1/2$

L'équation possède alors deux termes

$$\frac{\alpha}{2} \frac{\partial^2 q}{\partial \theta^2} - \nabla_x q \cdot \vec{u}(\theta) = 0, \quad (3.15)$$

un terme de transport et un terme de diffusion. C'est le seul cas intéressant. Les lignes de niveau sont prolongées et en même temps elles sont diffusées entre les angles. Les lignes de niveau ne sont donc pas simplement prolongées en ligne droite. Le résultat de l'équation d'évolution

$$\begin{cases} q(x, \theta, 0) = q_0(x, \theta) \\ \frac{\partial q}{\partial t} = \frac{\alpha}{2} \frac{\partial^2 q}{\partial \theta^2} - \nabla_x q \cdot \vec{u}(\theta) \end{cases} \quad (3.16)$$

est illustré par la figure 3.18. Cette fois-ci nous voyons bien la probabilité de présence suivre la courbe. Nous pouvons remarquer que la probabilité des courbes qui sont le plus loin dans le tube est plus faible, les courbes qui épousent parfaitement la géométrie du tube sont rares. Par contre la partie contenant les courbes les plus probables se situe juste en arrière du front. Les courbes les plus probables n'épousent pas exactement la géométrie du tube.

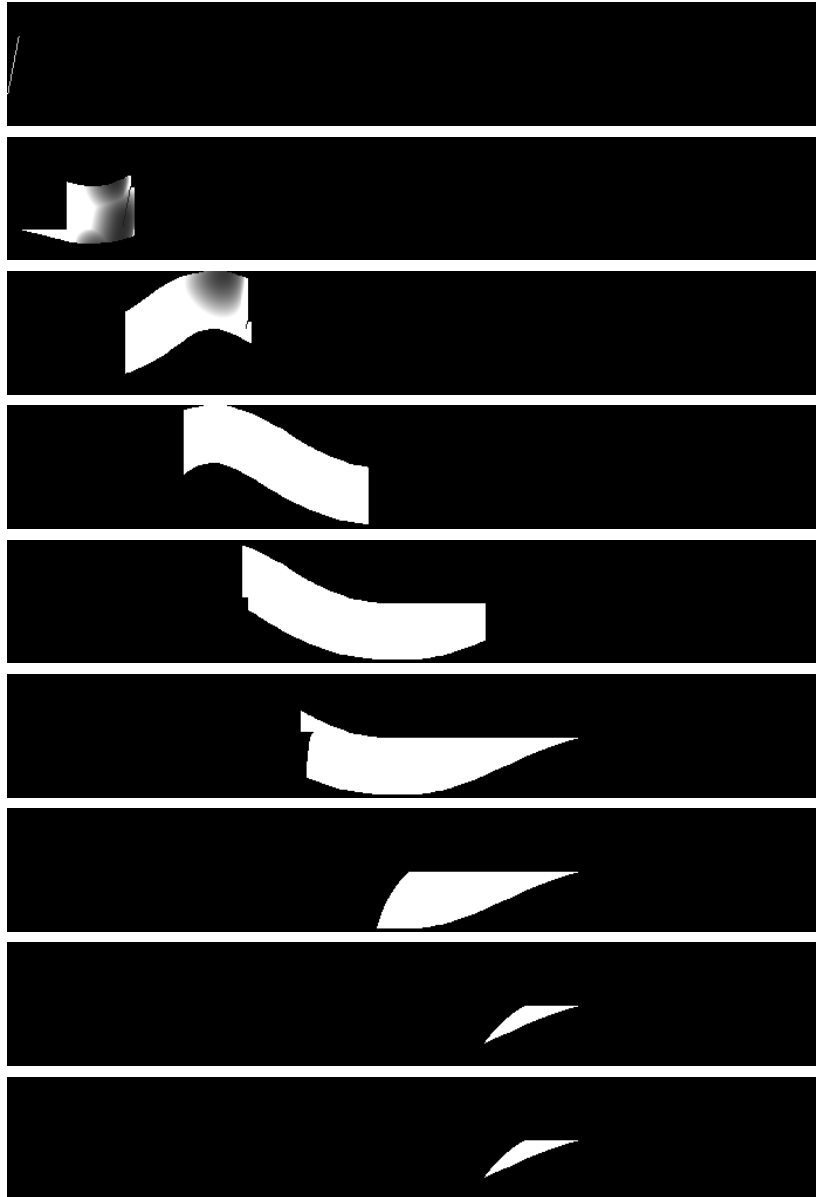


FIG. 3.17 – EDP de transport seul. Les dessins ci-dessus montrent la valeur de la probabilité de présence q correspondant à l'équation 3.14 pour différentes valeurs de temps. La probabilité de présence initiale est transportée cependant elle ne parvient pas à suivre les méandres du voisinage et finit par disparaître totalement.

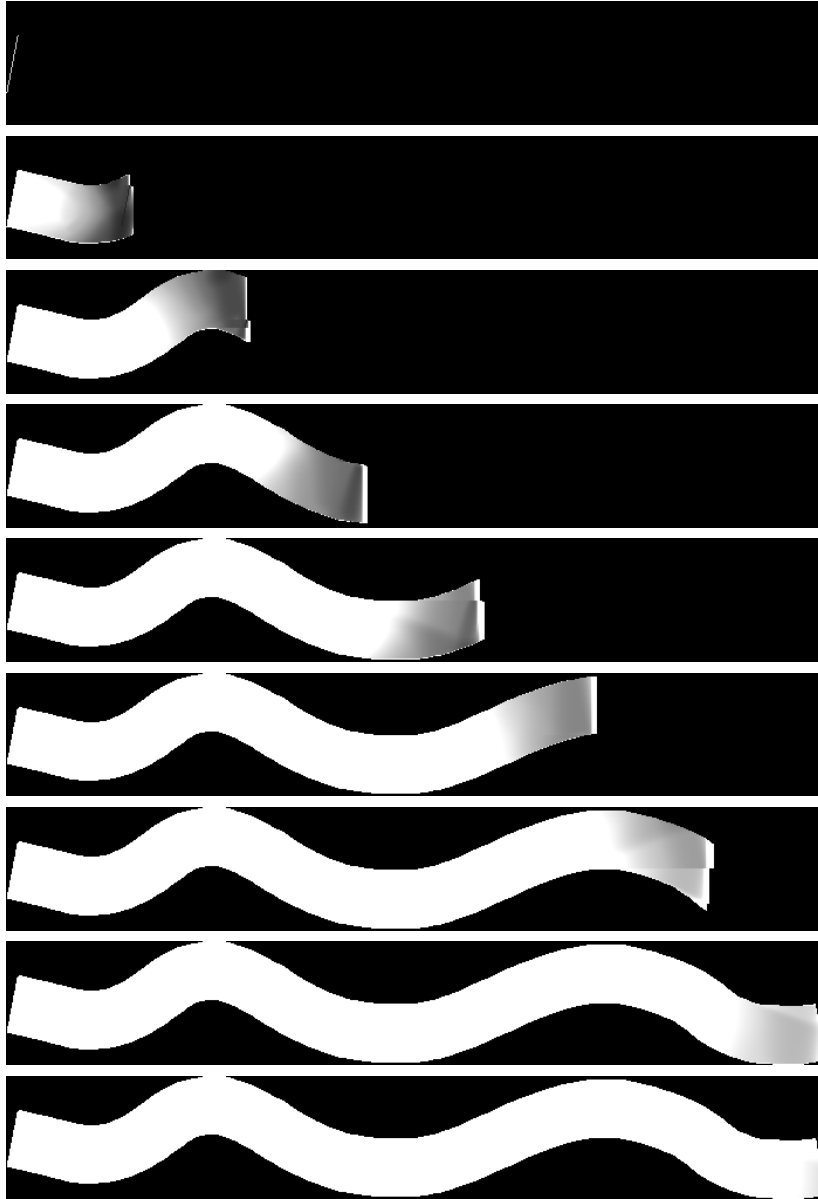


FIG. 3.18 – EDP de transport et de diffusion. Nous avons représenté la solution de l'équation 3.16 qui est présentée pour différents temps. Le terme de transport permet le transport effectif de la probabilité de présence depuis la probabilité de présence initiale. Le terme de diffusion permet à la probabilité de présence de suivre les méandres du voisinage.

3.3.3 Résolution numérique

Pour résoudre l'équation (3.16) nous utilisons un schéma itératif de sorte que nous cherchons une solution de l'équation d'évolution

$$\begin{cases} \frac{\partial q}{\partial t} = \frac{\alpha}{2} \frac{\partial^2 q}{\partial \theta^2} - \nabla_x q \cdot \vec{u}(\theta) & \text{dans } \overset{\circ}{\mathcal{D}}_0(\mathcal{C}, \varepsilon) \times [0, \infty[, \\ q(x, \theta, t) = 0 & \text{sur } \partial \mathcal{D}_0(\mathcal{C}, \varepsilon) \times [0, \infty[, \\ q(x, \theta, 0) = q_0(x, \theta) & \text{sur } \mathcal{D}_0(\mathcal{C}, \varepsilon). \end{cases} \quad (3.17)$$

Dans cette équation $\mathcal{D}_0(\mathcal{C}, \varepsilon)$ est la limite de $\mathcal{D}_r(\mathcal{C}, \varepsilon)$ lorsque r tend vers 0. C'est-à-dire que si $\mathcal{S}_0(x)$ est la limite de $\mathcal{S}_r(x)$ alors

$$\mathcal{D}_0(\mathcal{C}, \varepsilon) = \cup_{x \in \vec{V}(\mathcal{C}, \varepsilon)} \{x\} \times \mathcal{S}_0(x).$$

Maintenant si $V(\mathcal{C}, \varepsilon)$ est régulier alors $\mathcal{S}_r(x)$ tend vers \mathcal{S}^1 lorsque x est à l'intérieur de $V(\mathcal{C}, \varepsilon)$, vers un demi cercle lorsque x est sur le bord et vers un quart de cercle lorsque x est dans un coin, cf. figure 2.15.

Pour résoudre l'équation (3.17) le schéma que nous utilisons est

$$q_{i,j,k}^{n+1} = \eta_1 q_{i,j,k}^n + \eta_2 (q_{i,j,k+1}^n + q_{i,j,k-1}^n) + \eta_3 \tilde{q}_{i_*,j_*,k}^n, \quad (3.18)$$

avec $q_{i,j,k}^n = q(\delta i, \delta j, 2k\pi/K, n\tau)$ où τ est le pas de temps utilisé. Dans le schéma (3.18) le terme $\tilde{q}_{i_*,j_*,k}^n$ est une interpolation bilinéaire entre les valeurs de q en $(\delta i_0, \delta j_0)$, $(\delta i_0 + \delta, \delta j_0)$, $(\delta i_0, \delta j_0 + \delta)$ et $(\delta i_0 + \delta, \delta j_0 + \delta)$ où i_0 et j_0 sont les parties entières de i_* et j_* , et $(\delta i_*, \delta j_*)$ correspond au transport de $(\delta i, \delta j)$ dans la direction θ_k . Nous avons donc

$$\begin{cases} i_* = i - \frac{\varepsilon}{\delta} \cos(\theta_k), \\ j_* = j - \frac{\varepsilon}{\delta} \sin(\theta_k), \end{cases} \quad \begin{cases} i_0 = [i_*], \\ j_0 = [j_*], \end{cases} \quad (3.19)$$

et

$$\begin{aligned} \tilde{q}_{i_*,j_*,k}^n &= (1 - (i_* - i_0))(1 - (j_* - j_0))q_{i_0,j_0,k}^n \\ &\quad + (1 - (i_* - i_0))(j_* - j_0)q_{i_0,j_0+1,k}^n \\ &\quad + (i_* - i_0)(1 - (j_* - j_0))q_{i_0+1,j_0,k}^n \\ &\quad + (i_* - i_0)(j_* - j_0)q_{i_0+1,j_0+1,k}^n. \end{aligned} \quad (3.20)$$

Pour que le schéma (3.18) soit monotone il suffit de prendre η_1 , η_2 et η_3 tels que

$$\eta_1, \eta_2, \eta_3 \in [0, 1] \quad \text{et} \quad \eta_1 + 2\eta_2 + \eta_3 = 1. \quad (3.21)$$

Pour connaître les conditions pour que le schéma soit consistant nous considérons une fonction \mathcal{C}^2 en un point $(\delta i, \delta j, 2k\pi/K, n\tau)$.

PROPOSITION 29 :

Soit q une fonction \mathcal{C}^2 sur un voisinage du point $(x, \theta, t) = (\delta i, \delta j, 2k\pi/K, n\tau)$ contenant la boule de rayon $\varepsilon + \sqrt{2}\delta$ autour de x . Alors nous avons

$$\begin{aligned} q_{i,j,k}^{n+1} &- \eta_1 q_{i,j,k}^n - \eta_2 (q_{i,j,k+1}^n + q_{i,j,k-1}^n) - \eta_3 \tilde{q}_{i_*,j_*,k}^n \\ &= \tau \frac{\partial q}{\partial t} - \eta_2 \frac{4\pi^2}{K^2} \frac{\partial^2 q}{\partial \theta^2} + \eta_3 \varepsilon \nabla_x q \cdot \vec{u}(\theta_k) \\ &+ o_{\tau \rightarrow 0}(\tau) + o_{K \rightarrow +\infty}(1/K^2) + o_{\delta \rightarrow 0}(\delta) + o_{\varepsilon \rightarrow 0}(\varepsilon). \end{aligned} \quad (3.22)$$

Preuve : En (x, θ, t) nous avons les développements limités

$$q_{i,j,k}^{n+1} = q_{i,j,k}^n + \tau \frac{\partial q}{\partial t} + o_{\tau \rightarrow 0}(\tau), \quad (3.23)$$

$$q_{i,j,k+1}^n = q_{i,j,k}^n + \frac{2\pi}{K} \frac{\partial q}{\partial \theta} + \frac{2\pi^2}{K^2} \frac{\partial^2 q}{\partial \theta^2} + o_{K \rightarrow +\infty}(1/K^2), \quad (3.24)$$

$$q_{i,j,k-1}^n = q_{i,j,k}^n - \frac{2\pi}{K} \frac{\partial q}{\partial \theta} + \frac{2\pi^2}{K^2} \frac{\partial^2 q}{\partial \theta^2} + o_{K \rightarrow +\infty}(1/K^2), \quad (3.25)$$

$$q_{i_*,j_*,k}^n = q_{i,j,k}^n - \varepsilon \vec{u}(\theta_k) \cdot \nabla_x q + o_{\varepsilon \rightarrow 0}(\varepsilon). \quad (3.26)$$

Et ailleurs mais dans le voisinage de x

$$q_{i_*,j_*,k}^n = q_{i_0,j_0,k}^n + \delta \begin{pmatrix} i_* - i_0 \\ j_* - j_0 \end{pmatrix} \cdot \nabla_x q + o_{\delta \rightarrow 0}(\delta), \quad (3.27)$$

$$q_{i_*,j_*,k}^n = q_{i_0+1,j_0,k}^n + \delta \begin{pmatrix} i_* - i_0 - 1 \\ j_* - j_0 \end{pmatrix} \cdot \nabla_x q + o_{\delta \rightarrow 0}(\delta), \quad (3.28)$$

$$q_{i_*,j_*,k}^n = q_{i_0,j_0+1,k}^n + \delta \begin{pmatrix} i_* - i_0 \\ j_* - j_0 - 1 \end{pmatrix} \cdot \nabla_x q + o_{\delta \rightarrow 0}(\delta), \quad (3.29)$$

$$q_{i_*,j_*,k}^n = q_{i_0+1,j_0+1,k}^n + \delta \begin{pmatrix} i_* - i_0 - 1 \\ j_* - j_0 - 1 \end{pmatrix} \cdot \nabla_x q + o_{\delta \rightarrow 0}(\delta). \quad (3.30)$$

Les gradients sont pris en différents points. Cependant comme q est supposée \mathcal{C}^2 et que la distance entre deux points est un multiple de δ et enfin toutes ces égalités contiennent le terme $o_{\delta \rightarrow 0}(\delta)$. Nous pouvons donc écrire tous ces développements limités en utilisant le même point pour tous les gradients.

La somme des coefficients utilisés pour définir $\tilde{q}_{i_*,j_*,k}^n$ vaut 1. Avec les développements limités juste au-dessus nous obtenons alors

$$q_{i_*,j_*,k}^n = \tilde{q}_{i_*,j_*,k}^n + o_{\delta \rightarrow 0}(\delta), \quad (3.31)$$

car tous les termes contenant un gradient s'annulent.

En ajoutant les développements limités (3.24) et (3.25) nous obtenons alors

$$q_{i,j,k+1}^n + q_{i,j,k-1}^n = 2q_{i,j,k}^n + \frac{4\pi^2}{K^2} \frac{\partial^2 q}{\partial \theta^2} + o_{K \rightarrow +\infty}(1/K^2). \quad (3.32)$$

Maintenant nous utilisons la relation (3.21) pour décomposer 1

$$\begin{aligned}
 q_{i,j,k}^{n+1} &- \eta_1 q_{i,j,k}^n - \eta_2 (q_{i,j,k+1}^n + q_{i,j,k-1}^n) - \eta_3 \tilde{q}_{i^*,j^*,k}^n \\
 &= q_{i,j,k}^{n+1} - q_{i,j,k}^n + q_{i,j,k}^n - \eta_1 q_{i,j,k}^n - \eta_2 (q_{i,j,k+1}^n + q_{i,j,k-1}^n) \\
 &\quad - \eta_3 \tilde{q}_{i^*,j^*,k}^n, \\
 &= q_{i,j,k}^{n+1} - q_{i,j,k}^n - \eta_2 (q_{i,j,k+1}^n + q_{i,j,k-1}^n - 2q_{i,j,k}^n) \\
 &\quad - \eta_3 (\tilde{q}_{i^*,j^*,k}^n - q_{i,j,k}^n), \\
 &= \tau \frac{\partial q}{\partial t} + o_{\tau \rightarrow 0}(\tau) - \eta_2 \frac{4\pi^2}{K^2} \frac{\partial^2 q}{\partial \theta^2} + o_{K \rightarrow +\infty}(1/K^2) \\
 &\quad - \eta_3 (q_{i^*,j^*,k}^n - q_{i,j,k}^n) + o_{\delta \rightarrow 0}(\delta), \\
 &= \tau \frac{\partial q}{\partial t} + o_{\tau \rightarrow 0}(\tau) - \eta_2 \frac{4\pi^2}{K^2} \frac{\partial^2 q}{\partial \theta^2} + o_{K \rightarrow +\infty}(1/K^2) \\
 &\quad + \eta_3 \varepsilon \vec{u}(\theta_k) \cdot \nabla_x q + o_{\varepsilon \rightarrow 0}(\varepsilon) + o_{\delta \rightarrow 0}(\delta).
 \end{aligned} \tag{3.33}$$

Cette dernière écriture est bien celle que nous cherchions. \square

Pour que le schéma soit consistant il suffit donc que les relations suivantes soient satisfaites

$$\tau \frac{\alpha}{2} - \eta_2 \frac{4\pi^2}{K^2} = 0 \quad \text{et} \quad -\tau + \eta_3 \varepsilon = 0. \tag{3.34}$$

Pour l'algorithme nous avons donc plusieurs types de paramètres.

- Paramètre de l'EDP : α est le paramètre de l'équation (3.15).
- Paramètres de discrétisation : K , δ et ε sont les différents pas de discrétisation.
- Paramètre du schéma : η_1 est le paramètre qui est lié au schéma numérique que nous utilisons.
- Paramètres calculés : τ , η_2 et η_3 sont calculés à partir des précédents grâce aux relations que nous voulons vérifier pour que le schéma soit consistant.

Ces derniers paramètres, τ , η_2 et η_3 sont calculés comme suit

$$\begin{cases} \tau = & (1 - \eta_1) / \left(\frac{\alpha K^2}{4\pi^2} + \frac{1}{\varepsilon} \right), \\ \eta_2 = & \alpha \tau K^2 / 8\pi^2, \\ \eta_3 = & \tau / \varepsilon, \end{cases} \tag{3.35}$$

Par la suite nous appelons η le paramètre η_1 du schéma car c'est le seul coefficient parmi les trois η_1 , η_2 et η_3 que nous utilisons pour calculer les autres.

3.3.4 Calibration des paramètres

Paramètre de l'EDP. Sur la figure 3.19 apparaît la courbe de la probabilité de sortie en fonction de α . Le paramètre α décrit les probabilités de

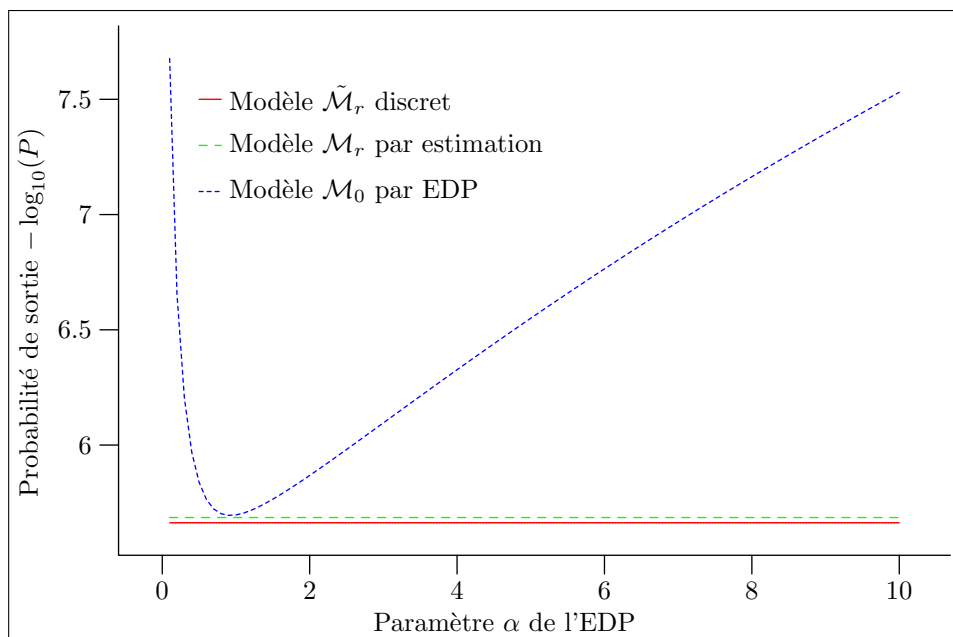


FIG. 3.19 – Influence d' α sur le modèle \mathcal{M}_0 . La courbe de la probabilité de sortie calculée par EDP en fonction du paramètre α de l'EDP montre un minimum proche des valeurs obtenues avec les deux autres modèles. Par la suite nous prenons $\alpha = 0,5$ pour les expériences. Pour les paramètres du schéma numérique de l'EDP nous prenons $\eta = 0$ et $\varepsilon = 2$.

changement de direction. Cependant les hypothèses que nous avons faites pour obtenir l'EDP sont fortes et le modèle de probabilité de changements de directions ainsi obtenu ne correspond pas nécessairement au type de distribution souhaitée. Le terme de diffusion de l'EDP suggère plus une loi sous-jacente Gaussienne qu'une loi de type Laplace. Le paramètre α permet de compenser ces approximations pour obtenir un résultat proche de celui du modèle $\tilde{\mathcal{M}}_r$. Nous prenons pour la suite $\alpha = 0,5$. Avec cette valeur le résultat obtenu est proche de ceux obtenus avec les autres méthodes.

Paramètres de discrétisation. Sur la figure 3.20 la courbe de la méthode par EDP n'a pas le même comportement que celle des deux autres méthodes. Il faut donc prendre K autour de 20 pour obtenir les mêmes résultats avec les trois méthodes. Le pas de discrétisation spatial δ est fixe. Il reste le paramètre ε . Il correspond à la distance utilisée pour l'équation de transport. Nous voulons le prendre le plus petit possible. Sur la courbe de la figure 3.21 nous observons qu'en deçà de 2, les valeurs obtenues ne sont plus bonnes du tout. Nous choisissons donc $\varepsilon = 2$.

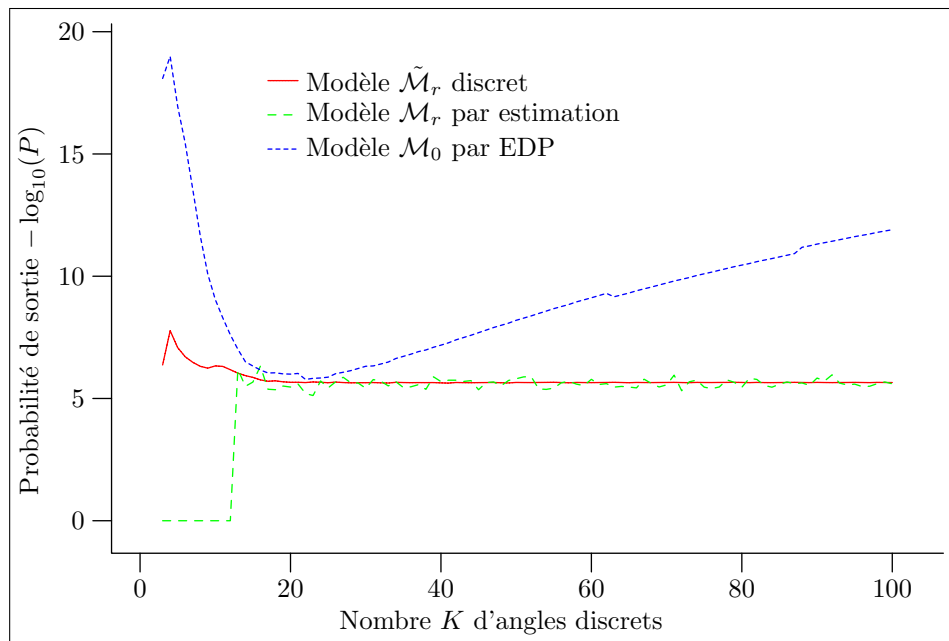


FIG. 3.20 – Influence de K sur le modèle \mathcal{M}_0 . Cette fois-ci avec le modèle \mathcal{M}_0 le résultat de la probabilité de sortie ne tend pas vers une valeur fixe lorsque le nombre d’angles discrets augmente. Cependant autour de $K = 20$ la courbe passe par un minimum qui correspond aux valeurs trouvées avec les deux autres modèles.

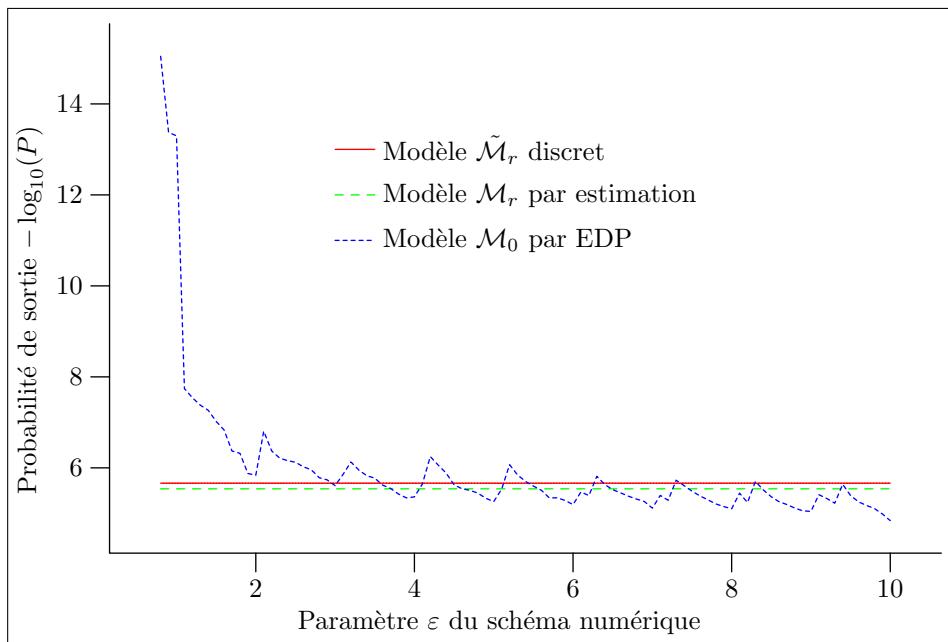


FIG. 3.21 – Influence de ε sur le modèle \mathcal{M}_0 . Lorsque le paramètre ε du schéma numérique de l'EDP du modèle \mathcal{M}_0 varie nous observons quelques artefacts numériques. Cependant à partir de $\varepsilon = 2$ la valeur de la probabilité de sortie est comparable à celle obtenue par les deux autres modèles. En deçà de cette valeur, le résultat n'est pas fiable. Nous prenons donc $\varepsilon = 2$.

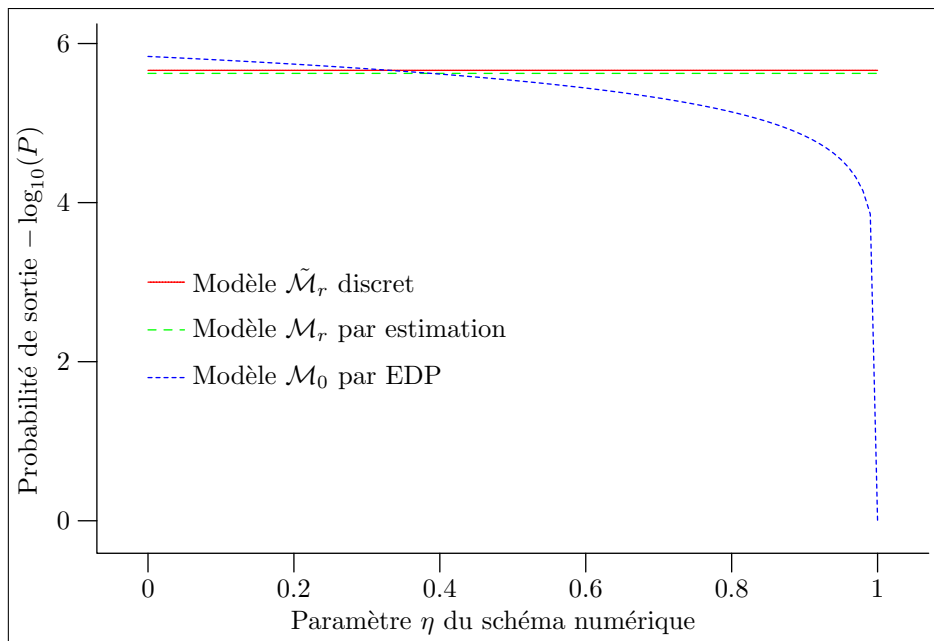


FIG. 3.22 – Influence de η sur le modèle \mathcal{M}_0 . Pour les valeurs de η inférieures à 0,8 la probabilité de sortie calculée par EDP suivant le modèle \mathcal{M}_0 est proche des valeurs calculées avec les deux autres modèles. Par la suite nous prenons $\eta = 0$.

Paramètre du schéma. Le schéma numérique possède un paramètre : η . La figure 3.22 montre l'influence de η sur la probabilité de sortie. Le coefficient η doit être pris le plus petit possible pour accélérer la résolution numérique de l'équation. En effet η correspond à la partie de la variable au temps précédent que nous gardons pour fabriquer la variable au temps suivant. Pour aller plus vite, il ne faut rien garder de fixe. Sur la courbe nous voyons que nous pouvons prendre $\eta = 0$. C'est ce que nous ferons par la suite.

Autres paramètres Les autres méthodes possèdent d'autres paramètres, le rayon des courbes, la probabilité de changement de direction et le facteur d'échelle. Le rayon des courbes n'est plus un paramètre car nous l'avons fait tendre vers 0 pour obtenir l'EDP. De même la probabilité de changement de direction n'apparaît dans l'EDP qu'avec le scalaire α . Le seul paramètre qui reste est donc le facteur d'échelle. La figure 3.23 montre le comportement de cette méthode par rapport au facteur d'échelle. Cette figure nous indique que nous devons prendre le facteur d'échelle inférieur à 4 pour obtenir les mêmes résultats qu'avec les autres méthodes.

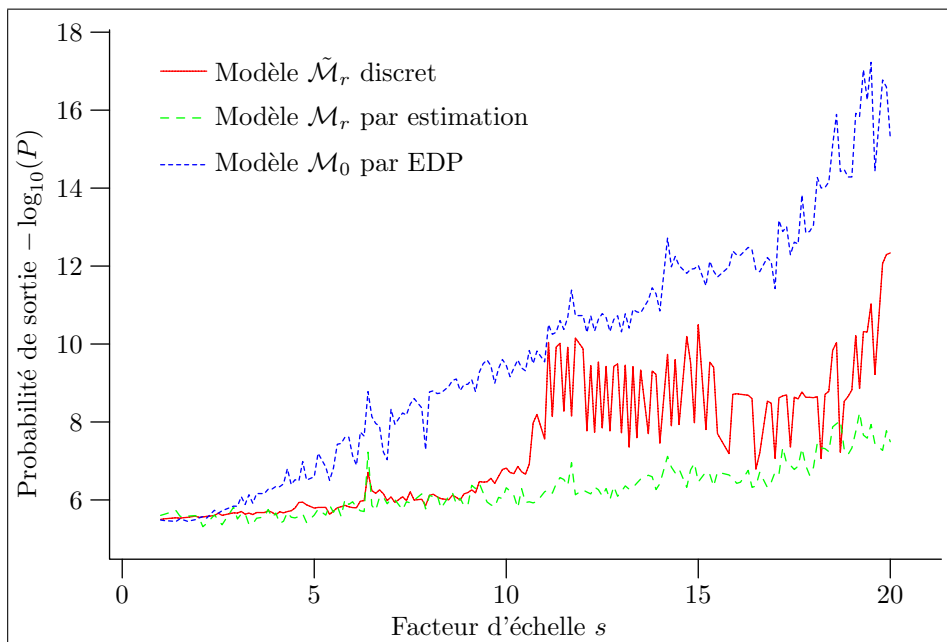


FIG. 3.23 – Influence de s sur le modèle \mathcal{M}_0 . Le comportement de la probabilité de sortie calculée par EDP suivant le modèle \mathcal{M}_0 diffère de celui de la probabilité calculée avec les deux autres modèles. Avec le modèle \mathcal{M}_0 la probabilité diminue plus vite lorsque s augmente qu'avec les deux autres modèles. Cependant pour les petites valeurs de s les résultats sont très proches, quel que soit le modèle.

Chapitre 4

Analyse et utilisation de la probabilité

4.1 Comparaison

Nous comparons les trois modèles présentés,

- le modèle discret $\tilde{\mathcal{M}}_r$,
- l'estimation par méthode de Monte-Carlo pour le modèle \mathcal{M}_r ,
- le modèle \mathcal{M}_0 par EDP.

Pour cela nous allons regarder l'influence de trois paramètres des voisinages $V(\mathcal{C}, \varepsilon)$ sur la probabilité de sortie. Nous allons considérer des courbes \mathcal{C} de courbure constante et les paramètres que nous étudions sont

1. la longueur L de la courbe \mathcal{C} ,
2. l'épaisseur ε du tube,
3. la courbure κ de la courbe \mathcal{C} .

Pour ces comparaisons, nous avons pris $\alpha = 2,5$, cette valeur permet de faire correspondre les courbes obtenues par le modèle $\tilde{\mathcal{M}}_r$ et par le modèle \mathcal{M}_0 .

Pour les deux premiers paramètres étudiés, la longueur et l'épaisseur, la courbure de la courbe est nulle, c'est-à-dire que nous avons effectué nos tests sur des droites pour ces paramètres.

4.1.1 Longueur

Nous commençons par faire varier la longueur de la courbe. La figure 4.1 montre les courbes obtenues par les trois méthodes. Les courbes obtenues semblent être des fonctions affines. Nous cherchons à modéliser la probabilité de sortie P en ne prenant en compte que les lignes droites. Soit q_0 la probabilité de présence initiale dans la direction du voisinage. Soit μ_0 la probabilité de ne pas changer de direction. Soit L la longueur du voisinage, il faut alors L/r

Valeur par défaut	
r	16
K	24
s	3
λ	5
N	300
α	2,5
η	0
ε_{EDP}	2
L	500
ε	50
κ	0

TAB. 4.1 – Paramètres par défaut lors des expériences. L’entier N est le nombre de courbes suivies lors de l’estimation de la probabilité de sortie suivant le modèle \mathcal{M}_r et ε_{EDP} est le paramètre ε du schéma numérique de l’EDP associée au modèle \mathcal{M}_0 .

itérations pour le modèle \mathcal{M}_r pour aller d’un bout à l’autre du voisinage. Nous avons donc

$$P \geq q_0 \mu_0^{L/r}, \quad (4.1)$$

et donc

$$-\log_{10}(P) \leq -\frac{L}{r} \log_{10}(\mu_0) - \log_{10}(q_0).$$

Ce modèle donne bien $-\log_{10}(P)$ comme une fonction affine de la longueur L . Nous prenons pour μ_0 la probabilité de changement de direction discrétisée en 0 et donc $\mu_0 = \tilde{\mu}_r(0)$. Pour la probabilité initiale q_0 nous prenons la longueur d’un secteur angulaire sur $\pi/2$, la longueur totale des départs possibles, et donc $q_0 = 4/K$. Si nous mesurons les coefficients des courbes affines de la figure 4.1 nous pouvons calculer des valeurs $\tilde{\mu}_0$ et \tilde{q}_0 . Nous avons pris les coefficients de la fonction affine de la figure 4.1. Ceci nous donne

$$\begin{aligned} \mu_0 &\approx 0,57, & \tilde{\mu}_0 &\approx 0,82, \\ q_0 &\approx 0,17, & \tilde{q}_0 &\approx 0,11. \end{aligned}$$

Les valeurs ne sont pas tout à fait proches, cependant si nous ajoutons à $\tilde{\mu}_r(0)$, les valeurs pour les angles contigus, nous obtenons 0,88, valeur beaucoup plus proche. Ceci est logique car le voisinage est assez large pour que les courbes de longueur L puissent aller dans une direction puis aller dans la direction opposée, et ainsi de suite et aller tout de même au bout du voisinage.

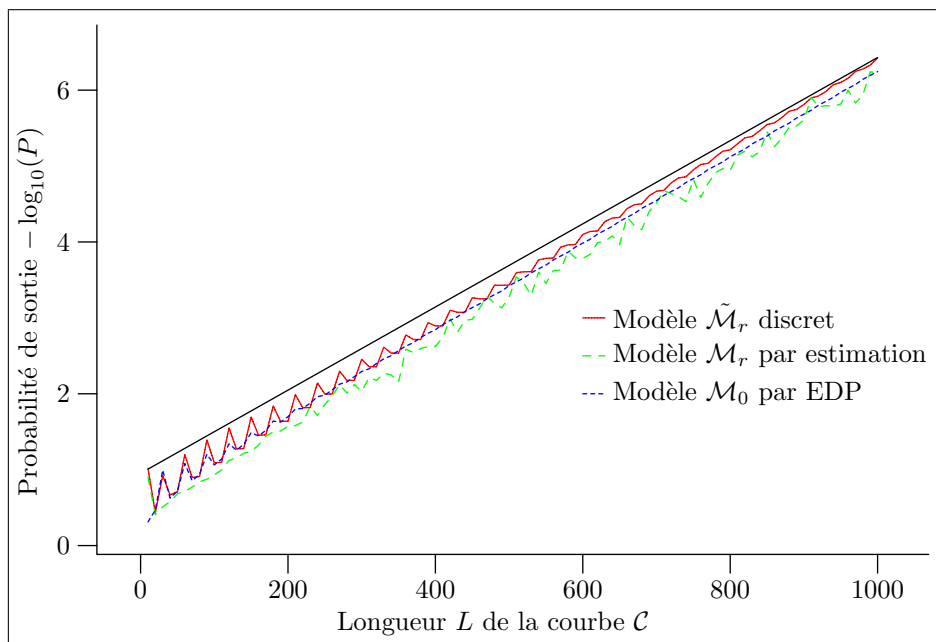


FIG. 4.1 – Influence de la longueur. La probabilité calculée par les trois modèles donne pour tous une droite affine en coordonnées semi-logarithmiques. Le modèle 4.1 donne une explication à ce phénomène. Il est toutefois logique que la probabilité diminue avec la longueur du voisinage.

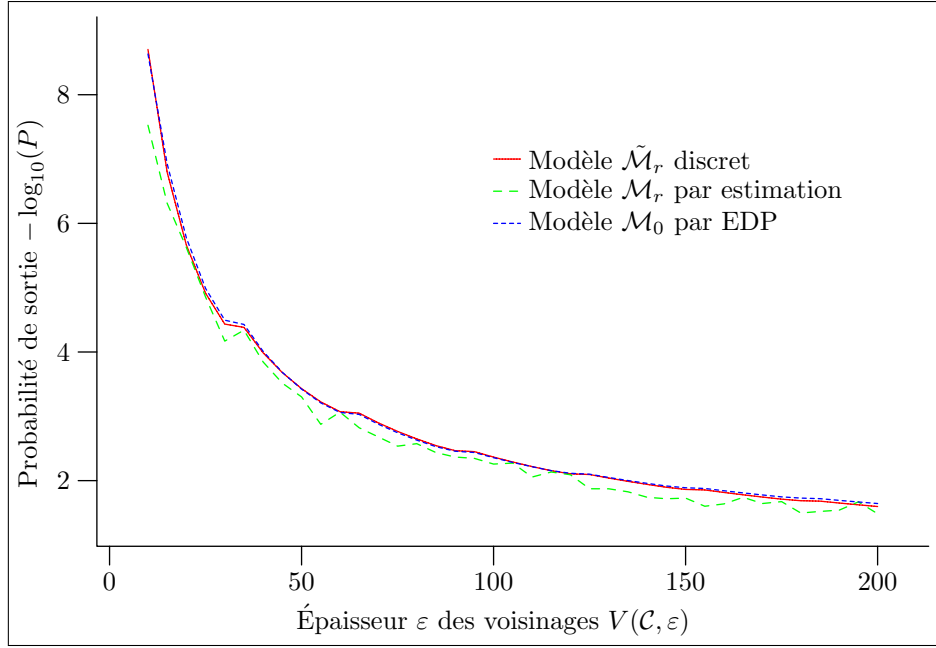


FIG. 4.2 – Influence de l'épaisseur. Cette fois-ci la probabilité de sortie augmente lorsque l'épaisseur du voisinage augmente. Ce qui est logique : plus le voisinage est épais, plus il est aisé de rester dedans.

4.1.2 Épaisseur

La figure 4.2 montre le résultat des calculs effectués pour un voisinage droit de longueur 500 pour des épaisseurs de 10 à 200 en allant de 5 en 5. Nous observons que plus le voisinage est épais, plus la probabilité est élevée, ce qui est bien conforme aux attentes. Nous allons prendre le modèle 4.1 pour tenter de modéliser l'influence de l'épaisseur utilisée. L'épaisseur du voisinage $V(\mathcal{C}, \varepsilon)$ est ε . Nous supposons ε très petit devant L . Nous considérons alors θ_0 l'angle d'observation de Γ_f à partir de Γ_d , que nous considérons constant sur Γ_d . Cet angle d'observation est illustré sur la figure 4.3.

Nous prenons $q_0 = \theta_0/(\pi/2)$. Sur la figure 4.3 nous constatons que $\tan(\theta_0) = \varepsilon/L$. Or avec nos hypothèses, nous avons $\tan(\theta_0) \approx \theta_0$ et donc $q_0 \approx 2\varepsilon/\pi L$. Ceci nous donne donc comme modèle

$$P \geq \frac{2\varepsilon}{\pi L} \mu_0^{L/r}. \quad (4.2)$$

Nous avons mis sur la figure 4.4 la courbe de P directement en fonction de l'épaisseur. En utilisant la fonction affine représentée sur cette figure nous avons calculé $\tilde{\mu}_0 \approx 0,87$, à peu près la même valeur que celle pour la comparaison sur les longueurs. Ce modèle ne permet pas d'expliquer toute la courbe car nous avons supposé ε très petit par rapport à L .

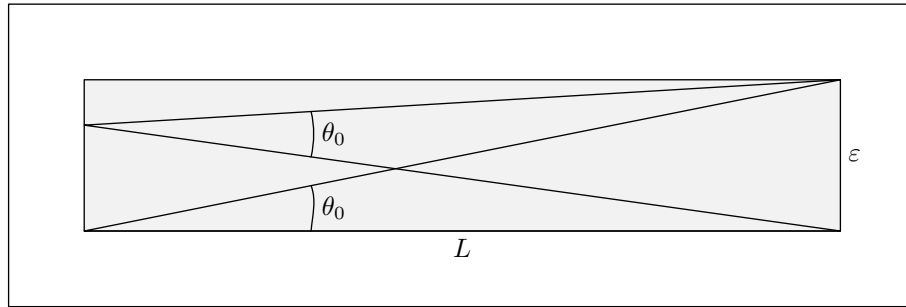


FIG. 4.3 – Angle d’observation de Γ_f depuis Γ_d . Nous supposons que la longueur de la courbe L est bien plus grande que l’épaisseur ε du voisinage rectangulaire. Dans cette situation l’angle d’observation d’une extrémité du voisinage est presque identique à partir de chacun des points de l’autre extrémité. Cette valeur commune est notée θ_0 .

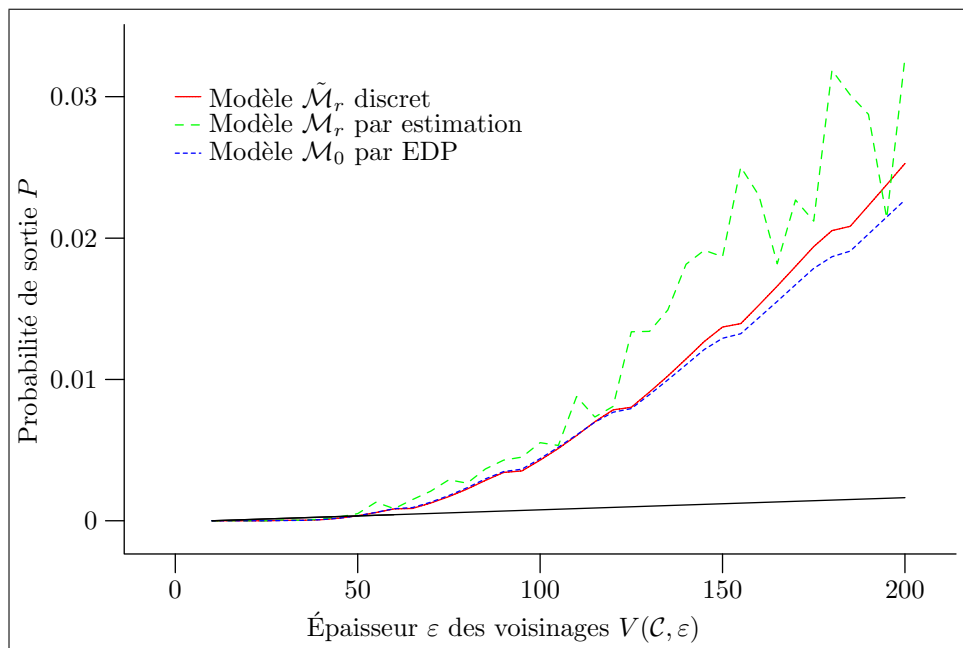


FIG. 4.4 – Influence de l’épaisseur en échelle linéaire. Le modèle 4.2 que nous avons trouvé pour la probabilité de sortie en fonction de l’épaisseur ε est une fonction affine en échelle linéaire. Nous représentons donc ci-dessus les mêmes courbes que sur la figure 4.2 mais avec une échelle linéaire.

4.1.3 Courbure

La figure 4.5 montre le résultat des calculs effectués pour un voisinage de longueur 500 et d'épaisseur 50 de courbure allant de 0 à 0,012 par pas de 0,0001. Contrairement aux résultats précédents le résultat est moins évident pour les modèles \mathcal{M}_r par estimation et \mathcal{M}_0 par EDP. La tendance globale est tout de même que plus le voisinage est courbe, plus la probabilité diminue.

Nous reprenons une fois encore le modèle 4.1 pour tenter de voir l'influence de la courbure sur la probabilité. Seulement cette fois nous remplaçons μ_0 par μ_κ , la probabilité d'avoir un angle qui produit des courbes de courbure κ . Nous commençons par regarder quel angle donne quelle courbure.

LEMME 30 :

Si θ est le changement de direction pour passer d'un segment à l'autre pour une courbe de \mathcal{C}_r alors la courbure κ en ce point vaut

$$\kappa = \frac{2}{r} \sin\left(\frac{\theta}{2}\right).$$

Ce qui nous permet ensuite de faire la minoration suivante,

PROPOSITION 31 :

Pour κ assez petit et K assez grand nous avons

$$-\log_{10}(P) \leq \frac{L\lambda}{\text{Ln}(10)}\kappa - \log_{10}(q_0) - \frac{L}{r} \log_{10}\left(\frac{2\pi}{KZ}\right).$$

Commençons par démontrer le lemme 30.

Preuve : Sur la figure 4.6 les segments $[A, B]$ et $[B, C]$ sont deux segments successifs d'une ligne polygonale. Ainsi nous avons $\overline{AB} = \overline{BC} = r$. Par les points A , B et C ne passe qu'un seul cercle, son centre est le point O , point d'intersection des médiatrices de $[A, B]$ et de $[B, C]$. Celles-ci coupent $[A, B]$ et $[B, C]$ respectivement en I_{AB} et I_{BC} les milieux des deux segments. Le rayon R du cercle est égal à \overline{OB} . Le changement de direction de la courbe au point B est représenté par l'angle θ . La courbure au point B est $\kappa = 1/R$. Les deux angles \widehat{CBO} et \widehat{OBA} sont égaux par symétrie, soit ψ cette valeur commune. Nous constatons que ces deux angles avec θ forment un angle plat,

$$2\psi + \theta = \pi.$$

De plus ψ et $\varphi/2$ sont deux angles d'un triangle rectangle et donc

$$\frac{\varphi}{2} + \psi = \frac{\pi}{2}.$$

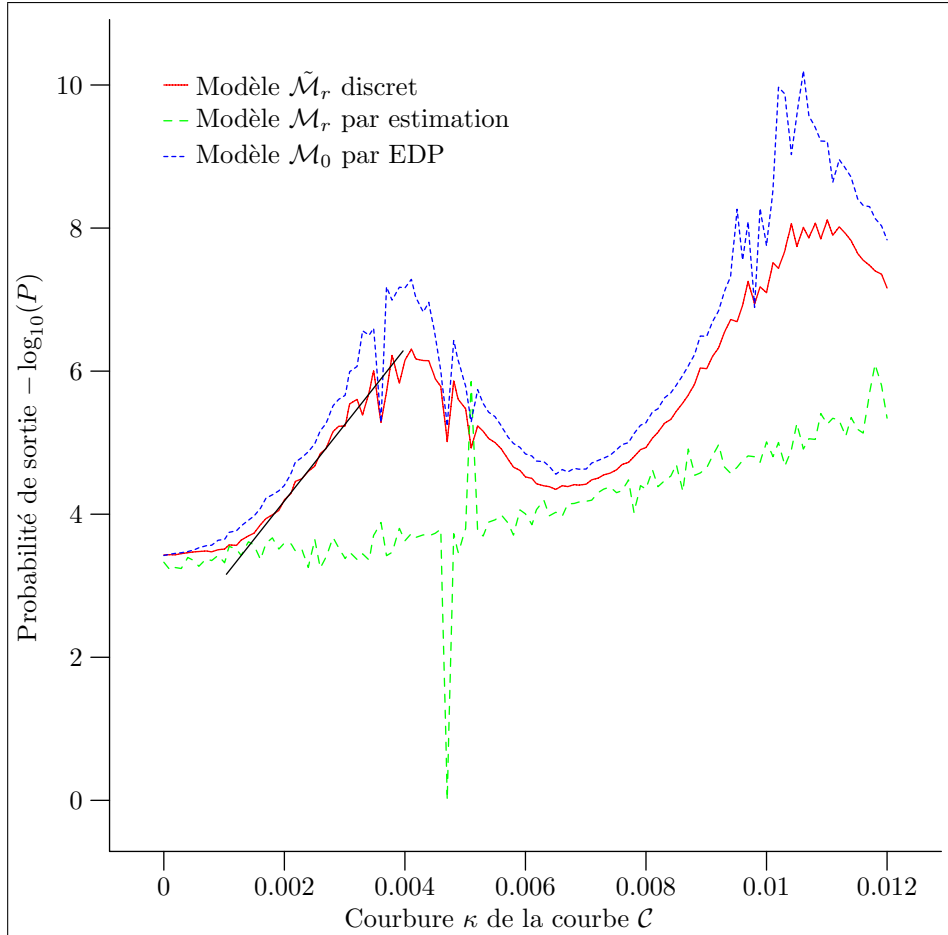


FIG. 4.5 – Influence de la courbure. L'influence de la courbure est différente selon les trois modèles utilisés. Seule l'estimation utilisée avec le modèle \mathcal{M}_r est monotone, la probabilité diminue lorsque la courbure augmente. Par contre avec les deux autres modèles nous avons un phénomène d'oscillation qui apparaît. Celui-ci est peut-être dû à l'influence des deux discrétisations. La première est la discrétisation du voisinage $V(\mathcal{C}, \varepsilon)$ dans \mathbb{R}^2 et la seconde est la discrétisation des directions de \mathcal{S}^1 .

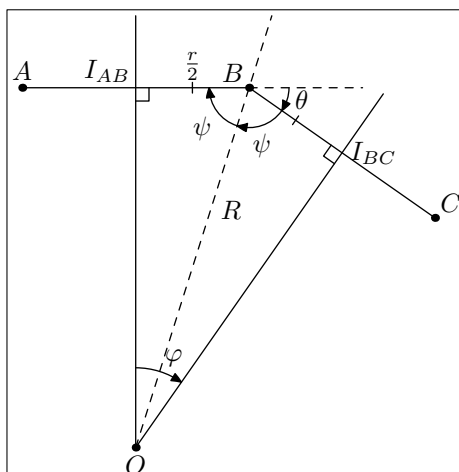


FIG. 4.6 – Courbure des courbes polygonales. Pour calculer la courbure d'une courbe polygonale nous regardons le rayon R du cercle passant par A , B et C , où A , B et C sont trois points successifs le long d'une courbe polygonale. Nous cherchons alors l'expression de ce rayon en fonction du changement de direction θ pour passer d'un segment à l'autre. La courbure de la courbe au point B est l'inverse du rayon R .

Au final nous obtenons

$$\theta = \varphi.$$

Or nous connaissons le sinus de $\varphi/2$,

$$\sin\left(\frac{\varphi}{2}\right) = \frac{r}{2R}.$$

Nous en déduisons l'expression de θ en fonction de κ ,

$$\theta = 2 \arcsin\left(\frac{r\kappa}{2}\right).$$

Ce qui prouve le lemme 30. □

Ensuite nous utilisons ce résultat pour montrer la proposition 31.

Preuve : Nous avons pris pour la loi de probabilité de μ_r une loi de Laplace de paramètre λ , de sorte que nous allons faire un calcul approché d'intégrale par la valeur médiane :

$$\mu_\kappa = \int_{\theta-\pi/K}^{\theta+\pi/K} \exp(-\lambda d(\varphi, 0)) / Z \, d\varphi = \frac{2\pi}{KZ} \exp(-\lambda|\theta|) + o_{K \rightarrow +\infty}(1/K).$$

Avec le lemme 30 nous avons

$$\mu_\kappa = \frac{2\pi}{KZ} \exp\left(-2\lambda \arcsin\left(\frac{r\kappa}{2}\right)\right) + o_{K \rightarrow +\infty}(1/K).$$

Nous faisons l'hypothèse supplémentaire que κ est très petit et ensuite des approximations successives

$$\mu_\kappa = \frac{2\pi}{KZ} \exp\left(-2\lambda \arcsin\left(\frac{r\kappa}{2}\right)\right) + o_{K \rightarrow +\infty}(1/K), \quad (4.3)$$

$$= \frac{2\pi}{KZ} \exp(-\lambda r\kappa) + o_{K \rightarrow +\infty}(1/K) + o_{\kappa \rightarrow 0}(\kappa), \quad (4.4)$$

$$= \frac{2\pi}{KZ} (1 - \lambda r\kappa) + o_{K \rightarrow +\infty}(1/K) + o_{\kappa \rightarrow 0}(\kappa), \quad (4.5)$$

Notre modélisation de la probabilité de sortie en fonction de la courbure est alors pour κ assez petit et K assez grand

$$P \geq q_0 \left(\frac{2\pi}{KZ} (1 - \lambda r\kappa) \right)^{L/r}$$

Et en passant au logarithme,

$$-\log_{10}(P) \leq -\frac{L}{r} \log_{10}(1 - \lambda r\kappa) - \log_{10}(q_0) - \frac{L}{r} \log_{10}\left(\frac{2\pi}{KZ}\right).$$

Avec notre hypothèse que κ est très petit nous pouvons simplifier cette expression en

$$-\log_{10}(P) \leq \frac{L\lambda}{\text{Ln}(10)} \kappa - \log_{10}(q_0) - \frac{L}{r} \log_{10}\left(\frac{2\pi}{KZ}\right). \quad (4.6)$$

Qui est celle annoncée dans la proposition 31. □

À partir de la pente segment de droite de la figure 4.5 nous en avons déduit une valeur de $\tilde{\lambda}$ de 4,91 au lieu de la véritable valeur 5.

4.2 Vers la détection

Pour passer du calcul de la probabilité $\mathbb{P}(\mathcal{C}_1, \mathcal{C}_2)$ à la détection de constance de largeur il faut que nous calculions le nombre de fausses alarmes $NFA(\mathcal{C}_1, \mathcal{C}_2)$. Ainsi il nous faudrait prendre chaque paire de courbes $(\mathcal{C}_1, \mathcal{C}_2)$, puis calculer $\rho'(\mathcal{C}_1, \mathcal{C}_2)$ et $\rho'(\mathcal{C}_2, \mathcal{C}_1)$ et enfin calculer $P(\mathcal{C}_1, \rho'(\mathcal{C}_1, \mathcal{C}_2))$ et $P(\mathcal{C}_2, \rho'(\mathcal{C}_2, \mathcal{C}_1))$. De cette manière nous avons à calculer $P(\mathcal{C}_1, \varepsilon)$ pour autant de ε que de courbes à tester. Pour éviter de faire autant de calculs nous calculons $P(\mathcal{C}_1, \varepsilon)$ à l'avance pour plusieurs ε . Nous avons alors pour chaque courbe \mathcal{C} des valeurs de $P(\mathcal{C}, \varepsilon)$ pour ε valant ε_1 à ε_N . Ensuite lorsque nous considérons un couple $(\mathcal{C}_1, \mathcal{C}_2)$, nous prenons ε_n tel que

$$\varepsilon_n = \min\{\varepsilon_k > \max(\rho'(\mathcal{C}_1, \mathcal{C}_2), \rho'(\mathcal{C}_2, \mathcal{C}_1))\}.$$

Et comme *a priori* $P(\mathcal{C}, \varepsilon)$ est croissant avec ε nous avons les majorations

$$P(\mathcal{C}_1, \rho'(\mathcal{C}_1, \mathcal{C}_2)) \leq P(\mathcal{C}_1, \varepsilon_n), \quad (4.7)$$

$$P(\mathcal{C}_1, \rho'(\mathcal{C}_2, \mathcal{C}_1)) \leq P(\mathcal{C}_2, \varepsilon_n). \quad (4.8)$$

Ces majorations nous donnent pour \mathbb{P} calculé avec ε_n au lieu du maximum des ρ' une surévaluation du *NFA* et donc les détections ε -significatives avec cette évaluation sont aussi ε -significatives pour le “vrai” *NFA*.

En utilisant un tableau de N_ε probabilités pour chaque courbe calculées à partir de N_ε épaisseurs, le nombre de tests effectués pour une détection est alors

$$N_{\text{tests}} = N_\varepsilon \frac{N_{\text{mln}}(N_{\text{mln}} - 1)}{2},$$

où N_{mln} est le nombre de morceaux de lignes de niveau que nous utilisons. Par exemple pour une image dont nous ne gardons qu’une centaine de lignes de niveau significatives, si nous coupons chacune de ces lignes en dix et que nous utilisons dix épaisseurs, nous avons à peu près $N_{\text{tests}} \approx 10^7$ tests.

Pour tester les résultats quantitativement entre différentes courbes, nous avons pris en compte les dix courbes de la figure 4.7, numérotées de 0 à 9. Nous avons choisi dix épaisseurs possibles, ε_1 à ε_{10} . Pour chacune des courbes nous avons calculé les probabilités de sortie pour ces dix épaisseurs. Enfin nous avons pris chaque paire de courbes et nous avons calculé $\max(\rho'(\mathcal{C}_1, \mathcal{C}_2), \rho'(\mathcal{C}_2, \mathcal{C}_1))$ entre les deux courbes de la paire, puis nous avons pris parmi les dix épaisseurs ε_n la plus petite majorant ce maximum. Dans le tableau 4.2 les deux premières colonnes correspondent à la paire de courbes considérée, la troisième colonne est le maximum des ρ' pour cette paire et dans la quatrième c’est l’épaisseur ε_n retenue. Les deux colonnes suivantes contiennent les probabilités de sortie pour cette épaisseur respectivement pour les courbes de la première et de la deuxième colonne. Enfin la dernière colonne présente $\mathbb{P}(\mathcal{C}_1, \mathcal{C}_2)$ calculée avec l’épaisseur ε_n sélectionnée de la quatrième colonne plutôt qu’avec le maximum des ρ' de la troisième colonne.

Les six valeurs en gras correspondent aux six plus faibles probabilités. Celles-ci correspondent aux paires

$$(5, 9), (3, 4), (1, 6), (0, 6), (0, 1) \text{ et } (7, 8)$$

respectivement. C’est-à-dire que si l’on met à part les lignes droites, les trois bandes blanches les plus régulières de la figure 1.12(a) sont bien détectées comme les moins probables, donc les plus remarquables. De plus en considérant que N_{tests} est proche de 10^7 comme ce serait approximativement le cas pour une détection dans une image digitale, les courbes avec des probabilités \mathbb{P} les plus faibles ont une probabilité inférieure ou égale à 10^{-7} et donc ont un *NFA* inférieur à 1. Cela signifie que les 6 paires que nous avons “détectées” sont les courbes 1-significatives de l’“image”.

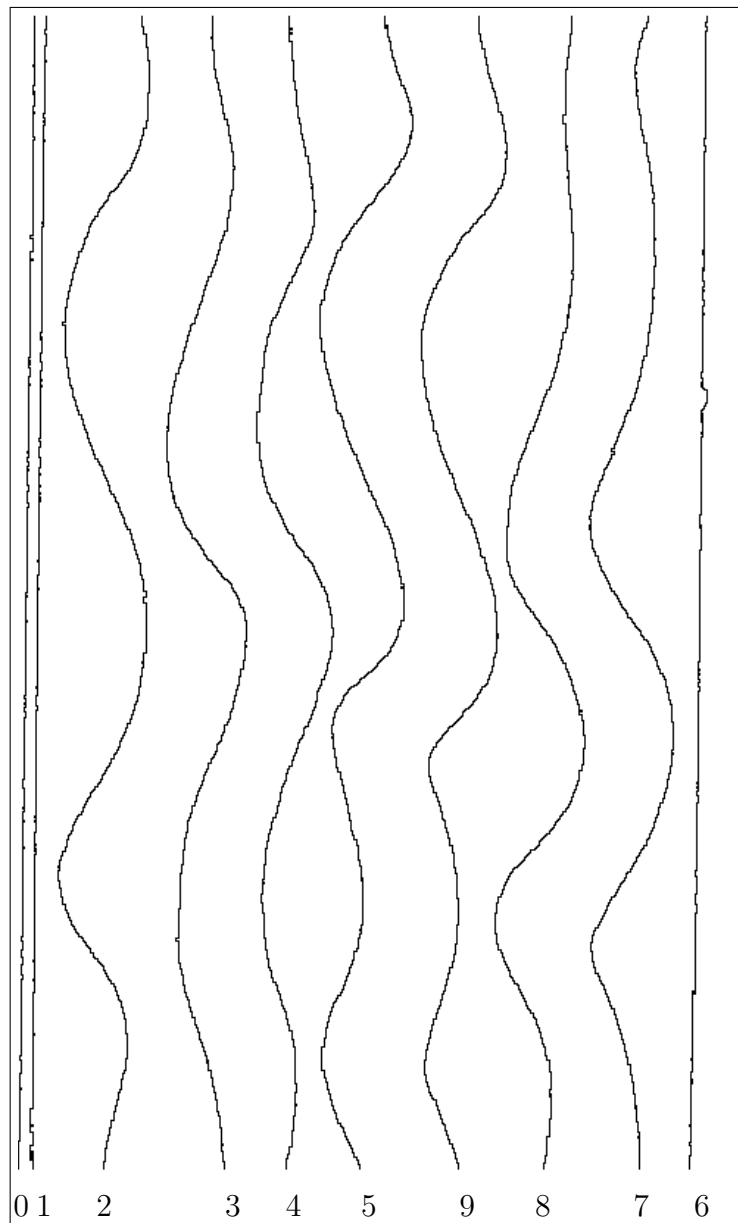


FIG. 4.7 – Dix courbes parallèles. Nous avons utilisé ces dix courbes pour faire un pas vers la détection. Nous avons calculé la probabilité de constance de largeur \mathbb{P} pour chaque paire de courbes choisie parmi celles-ci, le résultat est dans le tableau 4.2. Le modèle utilisé pour faire ces calculs est le modèle \mathcal{M}_r , par estimation.

Courbe \mathcal{C}_1	Courbe \mathcal{C}_2	Max. des ρ'	Épaisseur	$-\log_{10}(P(\mathcal{C}_1))$	$-\log_{10}(P(\mathcal{C}_2))$	$-\log_{10}(\mathbb{P}(\mathcal{C}_1, \mathcal{C}_2))$
0	1	3.15798	21.3333	7.59059	7.6183	7.59059
0	2	54.0562	69.5363	3.69107	4.3995	3.69107
0	3	43.8987	53.4686	4.34663	5.06777	4.34663
0	4	37.1318	37.401	5.77424	6.43212	5.77424
0	5	40.3717	53.4686	4.34663	5.60437	4.34663
0	6	2.39507	21.3333	7.59059	7.87542	7.59059
0	7	36.6363	37.401	5.77424	6.64284	5.77424
0	8	40.1641	53.4686	4.34663	5.30385	4.34663
0	9	36.9442	37.401	5.77424	7.03998	5.77424
1	2	53.9923	69.5363	3.70601	4.3995	3.70601
1	3	46.0075	53.4686	4.2691	5.06777	4.2691
1	4	38.238	53.4686	4.2691	5.06398	4.2691
1	5	39.9786	53.4686	4.2691	5.60437	4.2691
1	6	3.01498	21.3333	7.6183	7.87542	7.6183
1	7	37.5608	53.4686	4.2691	5.19984	4.2691
1	8	41.2422	53.4686	4.2691	5.30385	4.2691
1	9	36.346	37.401	5.55673	7.03998	5.55673
2	3	49.6882	53.4686	5.21797	5.06777	5.06777
2	4	41.6684	53.4686	5.21797	5.06398	5.06398
2	5	63.7195	69.5363	4.3995	4.27229	4.27229
2	6	53.9384	69.5363	4.3995	3.87124	3.87124
2	7	73.5804	85.6039	3.82141	3.43119	3.43119
2	8	76.4211	85.6039	3.82141	3.54746	3.54746
2	9	57.3712	69.5363	4.3995	4.41662	4.3995
3	4	19.0164	21.3333	9.43339	9.39912	9.39912
3	5	51.7356	53.4686	5.06777	5.60437	5.06777
3	6	50.869	53.4686	5.06777	4.53611	4.53611
3	7	52.1319	53.4686	5.06777	5.19984	5.06777
3	8	54.6561	69.5363	4.15312	4.30229	4.15312
3	9	46.7997	53.4686	5.06777	5.30751	5.06777
4	5	54.4745	69.5363	4.1641	4.27229	4.1641
4	6	48.4729	53.4686	5.06398	4.53611	4.53611
4	7	58.3294	69.5363	4.1641	4.16564	4.1641
4	8	60.6493	69.5363	4.1641	4.30229	4.1641
4	9	45.0305	53.4686	5.06398	5.30751	5.06398
5	6	55.9485	69.5363	4.27229	3.87124	3.87124
5	7	82.3449	85.6039	3.72212	3.43119	3.43119
5	8	84.2217	85.6039	3.72212	3.54746	3.54746
5	9	15.8669	21.3333	11.0339	11.0629	11.0339
6	7	52.3879	53.4686	4.53611	5.19984	4.53611
6	8	48.5657	53.4686	4.53611	5.30385	4.53611
6	9	45.502	53.4686	4.53611	5.30751	4.53611
7	8	22.4513	37.401	6.64284	6.98918	6.64284
7	9	70.2084	85.6039	3.43119	3.59791	3.43119
8	9	74.939	85.6039	3.54746	3.59791	3.54746

TAB. 4.2 – Premier pas vers la détection. Dans ce tableau nous avons considéré toutes les paires possibles de courbes en prenant les courbes de la figure 4.7. Pour chaque paire, nous avons mis le maximum des ρ' , l'épaisseur choisie, celle juste au-dessus parmi les dix épaisseurs choisies par avance, les probabilités sur chacun des voisinages $V(\mathcal{C}_1, \varepsilon)$ et $V(\mathcal{C}_2, \varepsilon)$ où ε est l'épaisseur choisie. Le calcul a été fait par estimation selon le modèle \mathcal{M}_r . Enfin la dernière colonne contient la probabilité de constance de largeur, le maximum des deux probabilités précédentes. Nous avons mis en gras les six plus faibles probabilités. Celles-ci correspondent bien aux bandes blanches de la figure 1.12(a).

Ainsi notre approche pour la détection de constance de largeur convient bien qualitativement. En effet les courbes qui sont relativement aux autres les plus détectables sont bien celles attendues. De plus quantitativement notre résultat nous porte à penser que cette approche permettra effectivement de détecter automatiquement les constances de largeur en prenant les constances de largeur 1-significatives.

Deuxième partie

Désocclusion

Chapitre 5

État de l'art

5.1 Introduction

Lorsqu'un objet est devant un autre objet, lors de la projection d'une scène dans une image, une partie d'un objet au premier plan peut masquer un objet qui est au second plan. Sur la figure 5.1(a) est représentée une scène en trois dimensions contenant une sphère et un cube. Sur cette figure nous avons mis également un œil. Le point de vue de l'œil est représenté sur la figure 5.1(b). Avec ce point de vue, le cube se retrouve devant la sphère. Si nous voulons retirer le cube de cette scène en deux dimensions, il va nous falloir reconstruire la partie de la sphère cachée par le cube.

Un autre exemple d'application est le retrait d'une personne sur une photographie, ou bien l'effacement d'objets dans une scène pour faire un trucage. Le problème de la restauration de vieilles photographies est similaire. Les mêmes techniques sont employées pour enlever des rayures sur des photographies. Le champ d'application pour la désocclusion d'image est vaste. Le problème central de la désocclusion est de remplir un trou dans une image, que ce trou résulte du retrait d'un objet ou d'une éraflure.

Dans le projet Ariana (<http://www-sop.inria.fr/ariana>) nous nous intéressons plus particulièrement aux images radar RSO (radar à synthèse d'ouverture) et à la vision radar interférométrique. Les images de phase obtenues en interférométrie radar contiennent de grandes zones sans aucune information exploitable. La figure 5.2 montre une telle image. Nous nous sommes donc intéressés au problème de la désocclusion pour pouvoir éventuellement l'appliquer à ces images.

Pour mettre en œuvre un algorithme de désocclusion nous commençons par poser mathématiquement le problème de la désocclusion.

Soit \mathcal{D} le domaine de l'image I , \mathcal{D} est un compact de \mathbb{R}^2 . Nous ne connaissons pas l'image sur un trou Ω inclus dans le domaine \mathcal{D} (cf. la figure 5.3). Ainsi, nous voulons déduire une fonction $I: \mathcal{D} \rightarrow \mathbb{R}$ à partir d'une fonction $I_0: \mathcal{D} \setminus \Omega \rightarrow \mathbb{R}$. Notre but est de trouver I telle que $I = I_0$ sur $\mathcal{D} \setminus \Omega$. La vali-

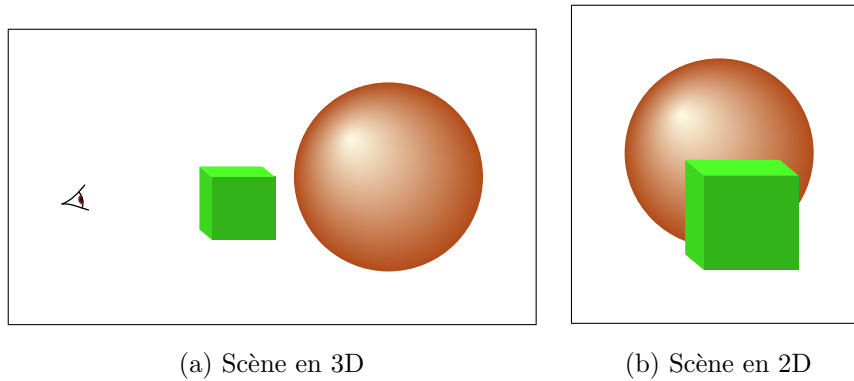


FIG. 5.1 – À gauche nous avons une scène en trois dimensions, à droite nous avons la même scène en deux dimensions vue par l'œil de la figure (a). Sur cette projection en deux dimensions de la scène le cube est devant la sphère et un morceau de la sphère est caché par le cube. Si nous voulons retirer le cube de cette image, il faut pouvoir reconstruire la partie cachée de la sphère.

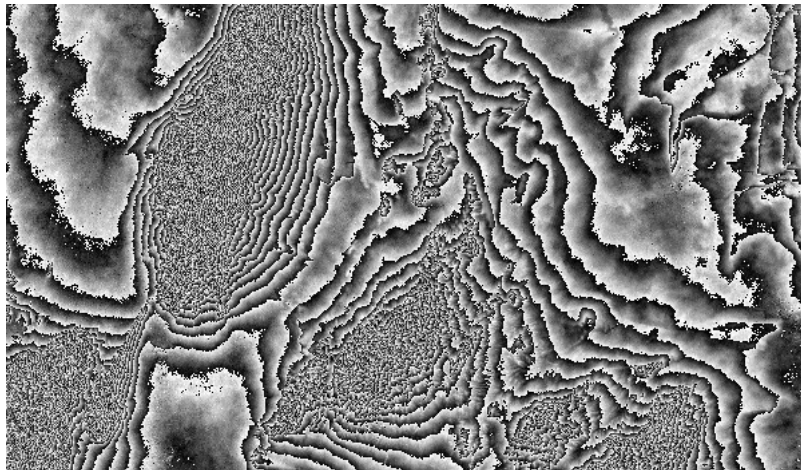


FIG. 5.2 – Cette image représente la différence entre les phases de deux images RSO d'une même scène prise sous deux angles différents. Les informations portées par cette image correspondent au relief de la scène. C'est effectivement le relief qui introduit des différences entre les deux images RSO. Cette différence de phase est connue modulo 2π c'est pourquoi l'image contient des successions de bandes allant du noir au blanc. En effet le modulo introduit des discontinuités là où il n'y en a en fait pas. Ces images de phases contiennent souvent de grandes zones sans information exploitable, en particulier lorsque la pente du terrain est élevée. Dans ce cas la précision du signal radar est insuffisante pour suivre et différencier les franges interférométriques.

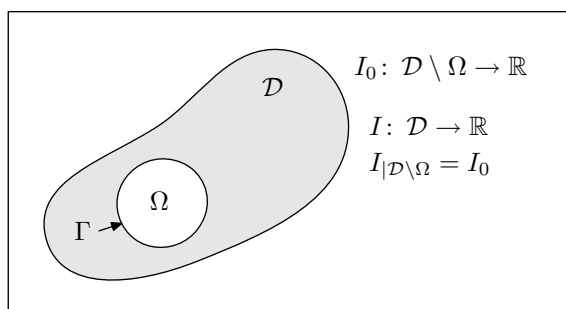


FIG. 5.3 – Un trou dans une image. L'image I est censée être définie sur tout le domaine \mathcal{D} . Le problème de la désocclusion est qu'en fait nous ne connaissons pas l'image sur tout \mathcal{D} . Il y a une zone Ω dans \mathcal{D} où l'image n'est pas connue, il y a un trou dans l'image. Ce que nous connaissons est donc une fonction I_0 définie sur le domaine sans le trou $\mathcal{D} \setminus \Omega$ et notre objectif est d'obtenir une fonction I définie sur \mathcal{D} en entier et égale à I_0 en dehors de Ω . Pour cela nous allons prolonger I_0 au delà de la frontière Γ du trou Ω .

dation d'un résultat n'est pas aisée. Visuellement il ne doit pas être possible en voyant le résultat I de percevoir où se trouvait le trou Ω . Plusieurs approches sont possibles.

Le cas des films est particulier et mieux posé. En effet un film est une séquence d'images. En plus de l'image où se trouve le trou, nous avons accès aux informations contenues dans les images qui précèdent ou qui suivent. Il est donc plus facile de reconstituer l'information manquante. Des techniques de désocclusion de film sont présentées dans l'article de Kokaram, Morris et Fitzgerald [23].

Pour le remplissage d'images fixes, il faut distinguer les zones à remplir par de la texture des zones à remplir sans texture. Pour les zones avec de la texture il faut savoir faire de la synthèse de texture dans le trou pour le combler. En effet le problème de la synthèse de texture est : étant donné un échantillon de texture comment créer une image du même type de texture. Efros et Leung dans [16] exposent un algorithme pour faire de la synthèse de texture. Dans le cas de la désocclusion ce genre d'approche produit des algorithmes de type supervisé. Il faut pour chaque trou fournir un échantillon de la texture à utiliser. Il est toutefois possible de donner comme échantillon l'image elle-même mais ceci a des inconvénients notamment pour les images de grande taille avec peu d'échantillons correspondants à la partie à reconstruire.

Pour le cas des zones sans textures, nous considérons une image comme une application d'un domaine \mathcal{D} de \mathbb{R}^2 , le plus souvent un rectangle, à valeurs dans \mathbb{R} avec une certaine régularité. C'est le cas que nous traitons. Par exemple nous pouvons considérer que les images sont des fonctions à variations bornées. Dans ce cas le but est proche de l'interpolation de fonction. Il s'agit de

prolonger l'application à l'intérieur du trou en préservant la régularité de l'application. Plusieurs approches existent pour les images sans textures. Certaines s'appuient sur les lignes de niveau de l'image comme par exemple [27, 4, 2]. Une autre approche est de considérer sur les équations de Navier-Stokes. C'est le cas dans [3]. Et il est aussi possible de partir d'algorithmes de débruitage, c'est le cas de Chan et Shen dans [9].

Nous pouvons imaginer séparer la texture de la structure de l'image. Des algorithmes de décomposition existent pour cela comme dans [1] ou [38]. Ces décompositions sont des implémentations numériques de la décomposition proposée par Meyer dans [29]. Ensuite il est possible d'utiliser une méthode de désocclusion appropriée pour chaque type d'image (texture et structure). Il ne reste plus qu'à les recomposer pour obtenir l'image complète. C'est la méthode employée dans l'article de Bertalmio, Vese, Sapiro et Osher [7].

Nous présentons dans ce chapitre des méthodes existantes pour la désocclusion des images sans texture.

5.2 Désocclusion basée sur les lignes de niveau

La contribution de Masnou et Morel en 1998 dans [27] a pour point de départ le livre de Nitzberg, Mumford et Shiota [30] paru en 1993 lié au problème de la segmentation d'image. Leur approche est liée à la désocclusion. En effet ils reconstruisent les contours des objets présents dans l'image. Cependant comme certains objets en cachent d'autres, les auteurs font de la désocclusion sur ces contours. La segmentation proposée par Nitzberg, Mumford et Shiota se déroule en plusieurs étapes. Elle repose sur les jonctions en T . Ils utilisent les jonctions en T pour détecter les endroits où un objet est caché par un autre objet. En effet lorsque cela arrive, comme sur la figure 5.4(a) par exemple, le contour de l'objet partiellement caché arrive orthogonalement sur celui de l'objet qui le cache. Ainsi leurs contours forment une sorte de T . Les jonctions en T des contours de la figure 5.4(a) sont encerclés dans la figure 5.4(b). Les étapes de l'algorithme de Nitzberg, Mumford et Shiota sont

1. lissage de l'image par de la diffusion non-linéaire (Perona-Malik),
2. détection des bords visibles des objets,
3. fabrication des lignes de contours en reliant les points de bord,
4. détection des coins et des jonctions en T ,
5. lissage des lignes de contour,
6. fermeture des contours en reliant les jonctions en T .

Quand ils ont obtenu les jonctions en T ils relient entre elles les lignes de contour des objets cachés qui s'arrêtent sur ces jonctions en T . Ils obtiennent alors des contours fermés pour chaque objet qu'ils souhaitent segmenter. Pour

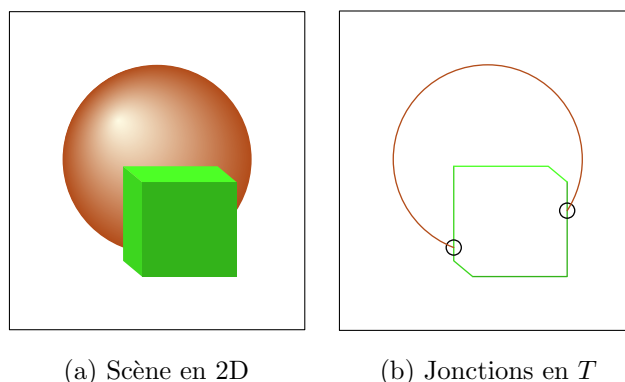


FIG. 5.4 – Jonctions en T . Dans la scène en deux dimensions de la figure de gauche un cube cache en partie une sphère. Le contour de la sphère s’arrête abruptement sur le contour du cube. Sur la figure de droite ne sont représentés que les contours des objets. Les endroits où le contour de la sphère s’arrête sur celui du cube sont encerclés. À ces endroits le contour de la sphère arrive presque orthogonalement sur celui du cube. Ce type de configuration s’appelle une jonction en T : à ces endroits les deux contours forment ensemble une sorte de T .

relier ces jonctions en T ils utilisent des élastiques d’Euler. Les élastiques d’Euler sont les courbes reliant deux points en minimisant la longueur et la courbure avec des tangentes fixées à chaque extrémité. Toutes les possibilités de liaisons sont explorées et celle retenue est celle minimisant la longueur et la courbure de l’ensemble des courbes.

C’est cette idée de reconstruction de contour que Masnou et Morel adaptent dans [27] pour la désocclusion d’image. Dans [30] les auteurs referment les contours des objets et cette idée est reprise par les auteurs de [27] reprenne. Ils utilisent les lignes de niveau de l’image car celles-ci contiennent en fait la description complète de l’image (voir l’équation (6.1)). La figure 5.5 montre comment procéder pour reconstruire une image avec ses lignes de niveau.

La figure 5.5(a) est l’image originale, un extrait d’une photographie de paysage prise au Pouldu dans le Finistère. Cette image est représentée avec ses lignes de niveau dans la figure 5.5(b). Pour retirer la maison présente au centre de l’image nous plaçons un trou au-dessus de celle-ci. Ce trou correspond au carré blanc de la figure 5.5(c). les lignes de niveau de l’image forment des jonctions en T avec le bord du trou. Dans la figure 5.5(d) nous avons manuellement reconnecté les lignes de niveau qui arrivaient sur la frontière du trou. Sur la figure 5.5(e) nous avons rempli l’espace entre les lignes de niveau avec le niveau de gris correspondant à ces lignes. Enfin la figure 5.5(f) montre le résultat de la désocclusion.

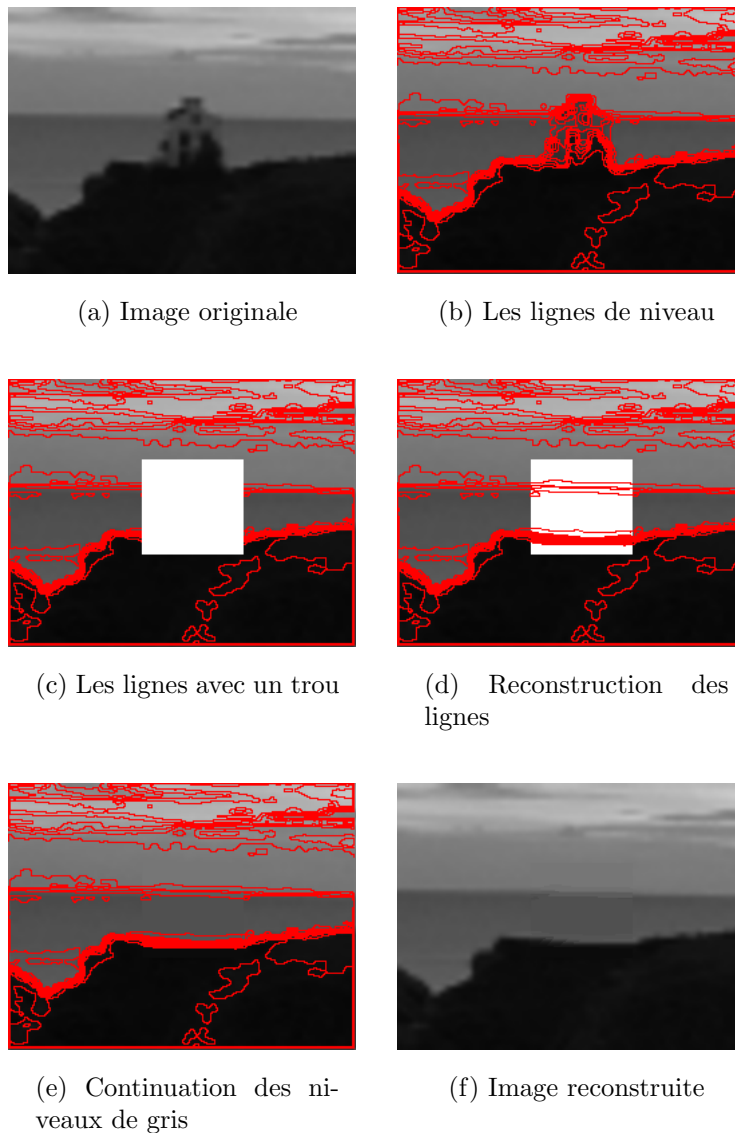


FIG. 5.5 – Désocclusion manuelle. En haut à gauche nous avons une image originale, c'est un extrait d'une photographie prise au Pouldu dans le Finistère. Nous voulons enlever la maison de la photo. En haut à droite nous avons superposé à l'image ses lignes de niveau. Au centre à gauche nous avons placé un trou, le carré blanc, au-dessus de la maison. Nous voulons boucher ce trou et donc faire de la désocclusion. Pour cela nous avons continué à la main les lignes de niveau de l'image. Le résultat est au centre à droite. Une fois les lignes de niveau complètes, nous avons rempli les espaces inter-lignes de niveau avec les niveaux de gris correspondant aux lignes de niveau. En bas à gauche nous avons placé l'image reconstruite avec ses lignes de niveau et en bas à droite l'image seule. La maison a bien été enlevée.

La plupart des lignes de niveau de l'image arrivent comme pour les lignes de contour sur la frontière du trou en formant des jonctions en T . Masnou et Morel utilisent les mêmes courbes pour refermer les lignes de niveau que Nitzberg, Mumford et Shiotani utilisent pour refermer les contours. La motivation pour l'utilisation des lignes de niveau est double. Tout d'abord il s'agit aussi de reconnecter des jonctions en T et une fois ces connections établies il est facile de reconstruire l'image. Il suffit pour cela de remplir le trou avec le niveau de gris adéquat entre deux lignes de niveau comme nous l'avons fait sur la figure 5.5. De plus les expériences de l'école Gestaltiste de psychophysique (voir le livre de Kanizsa [22]) montrent que l'œil est peu sensible au changement de contraste d'une image. Ce qui justifie l'utilisation des lignes de niveau d'une image qui sont invariantes par changement de contraste (cf. figure 6.1). En se basant uniquement sur les lignes de niveau de l'image Masnou et Morel peuvent reconstruire un bord indépendamment de son intensité. Voyons plus en détails comment ils procèdent.

Si nous considérons les images non texturées, la plupart sont à variations bornées. L'espace des fonctions à variations bornées (noté BV) est donc souvent l'espace utilisé en traitement d'image. L'image I à reconstruire est cherchée dans cet espace des fonctions à variations bornées, $BV(\mathcal{D})$, dont nous rappelons la définition : soit

$$\int_{\mathcal{D}} \|\nabla I\| = \sup\left\{ \int_{\mathcal{D}} I \operatorname{div}(\varphi), \text{ avec } \varphi \in \mathcal{C}^\infty(\mathcal{D}), \|\varphi\|_\infty \leq 1 \right\}, \quad (5.1)$$

alors $BV(\mathcal{D})$ est défini par

$$BV(\mathcal{D}) = \left\{ I \in L^1(\mathcal{D}); \int_{\mathcal{D}} \|\nabla I\| < \infty \right\}, \quad (5.2)$$

$\int_{\mathcal{D}} \|\nabla I\|$ est appelé la variation totale de I . Il s'agit en fait de la mesure totale de \mathcal{D} avec comme mesure le gradient de I , $\|\nabla I\|(\mathcal{D})$. Si une fonction indicatrice d'un ensemble E est à variation bornée alors dans ce cas on dit que l'ensemble est de périmètre fini et on note $P(E) = \int_{\mathcal{D}} \|\nabla \mathbb{1}_E\|$, le périmètre de E . Celui-ci correspond au périmètre usuel lorsqu'il peut être défini. Les fonctions à variations bornées vérifient la formule de la co-aire

$$\int_{\mathcal{D}} \|\nabla I\| = \int_{\mathbb{R}} P(\chi_\lambda I) d\lambda$$

où $\chi_\lambda(I) = \{x \in \mathcal{D} \text{ tels que } I(x) \geq \lambda\}$ sont les ensembles de niveau supérieur de l'image I . Comme les lignes de niveau d'une image sont les frontières des ensembles de niveau, minimiser $\int_{\mathcal{D}} \|\nabla I\|$ équivaut à minimiser la longueur des lignes de niveau de l'image. C'est pour cette raison que Masnou et Morel proposent de minimiser

$$\int_{\Omega} \|\nabla I\| (1 + \operatorname{curv}(I)) dx,$$

où

$$\text{curv}(I) = \text{div}(\nabla I / \|\nabla I\|)$$

désigne la courbure des lignes de niveau de I . Il minimisent ainsi la longueur et la courbure des lignes de niveau de l'image I .

Les lignes de niveau ont une propriété remarquable, elles ne s'intersectent pas. Cette propriété permet aux auteurs d'utiliser la technique de programmation dynamique et ainsi de diminuer le temps de calcul. Il s'agit de minimiser des coûts sur des chemins. Si on note $(L_i)_i$ les lignes de niveau arrivant à la frontière du trou alors on note $L_{i,j}$ la ligne de niveau joignant la ligne L_i à la ligne L_j . Toutes les connections ne sont pas permises. La fonction de coût pour des connections admissibles est

$$\mathcal{C} = \sum \int_{L_{i,j}} (1 + |\sigma(s)|) ds$$

où $\int_{L_{i,j}} |\sigma(s)| ds$ est la variation totale de l'orientation de la tangente le long de $L_{i,j}$, ce qui correspond à la mesure de la courbure. Il faut donc calculer les chemins $L_{i,j}$ et leurs coûts. Pour cela les auteurs utilisent l'algorithme à barrière de Heshberger et Snoeyink [21].

L'algorithme de Masnou et Morel est scindé en deux parties. Premièrement ils complètent les lignes de niveau. Pour cela ils utilisent la programmation dynamique et l'algorithme de Heshberger et Snoeyink pour calculer les géodésiques reconnectant les lignes de niveau en minimisant leurs longueurs. Ensuite ils utilisent les lignes de niveau comme frontière des ensembles de niveau pour remplir l'espace entre les lignes de niveau avec le bon niveau de gris, celui du bord du trou.

Cette méthode permet d'avoir une reconstruction qui est indépendante des changements de contraste. De plus les auteurs respectent la règle de bonne continuation énoncée par l'école Gestaltiste de psychophysique. Par contre les lignes de niveau reconstruites sont presque rectilignes. En effet la principale minimisation est faite sur la longueur des lignes de niveau et non sur leur courbure.

$$\int_{\Omega} \|\nabla I\| (1 + \text{curv}(I)) dx = \int_{\Omega} \|\nabla I\| dx + \int_{\Omega} \|\nabla I\| \text{curv}(I) dx.$$

En diminuant la longueur, les deux termes diminuent, par contre diminuer la courbure ne diminue que le second terme. De même les angles des lignes de niveau à la frontière du trou ne sont généralement pas respectés. Ceci peut conduire à une reconstruction d'un objet qui ne sera pas conforme à ce que notre cerveau fait en pratique. En effet dans le cas où une partie d'un disque est cachée par un objet, notre œil aura plutôt tendance à *voir* le disque, ceci est illustré dans la figure 5.6. Une autre limitation est que cet algorithme ne fonctionne que pour des trous dont la topologie est assez simple. Ainsi il ne doit pas y avoir de trou dans la partie à reconstruire (il ne doit pas y avoir de trou dans le trou!).

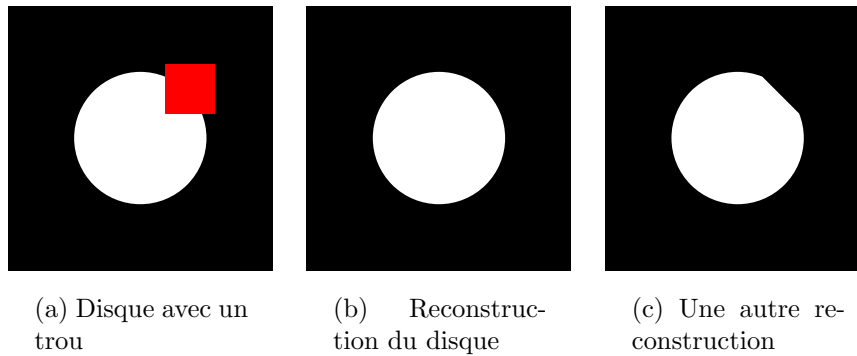


FIG. 5.6 – Un disque et deux reconstructions possibles.

5.3 Méthode inspirée de la restauration de peinture

La connaissance des lignes de niveau permet de reconstruire l'image et évidemment reconstruire l'image permet de connaître les lignes de niveau. Masnou et Morel dans [27] séparent les deux aspects, ils reconstruisent les lignes de niveau puis ils reconstruisent l'image à partir des lignes de niveau. Dans l'article [4] présenté par Bertalmio, Sapiro, Caselles et Ballester en 2000 les auteurs proposent un algorithme itératif qui effectue les deux aspects en parallèle. Ils s'appuient sur les lignes de niveau pour modifier l'image. Ceci modifie les lignes de niveau, et ils recommencent à modifier l'image en s'appuyant sur les nouvelles lignes de niveau et ainsi de suite.

Pour trouver un moyen de simultanément changer l'image et ses lignes de niveau Bertalmio et al. ont étudié le travail des conservateurs de l'institut des Arts de Minneapolis. Depuis très longtemps les artistes font de la désocclusion en restaurant les œuvres d'art. Les auteurs de [4] ont donc cherché à connaître la façon de faire de ces restaurateurs. Chaque restauration est unique mais il s'en dégage un schéma général. Tout d'abord les restaurateurs utilisent la région autour de la partie à restaurer. Ils complètent la structure dans la région à partir de cette information. Ils remplissent ensuite les régions avec les bonnes couleurs par rapport à la frontière de la région. Enfin ils rajoutent les détails, ce qui peut être assimilé à la texture.

Cette technique a inspiré les auteurs de [4] pour construire un algorithme de désocclusion. L'idée est de prolonger les lignes qui arrivent au bord du trou. Pour prolonger les lignes qui arrivent à la frontière Γ du trou les auteurs propagent une information dans une direction. La direction de propagation choisie est celle des lignes de niveau de l'image. L'information qu'ils choisissent de propager est le laplacien de l'image. Le laplacien contient de l'information

concernant la régularité de l'image. Les auteurs modélisent la propagation du laplacien dans la direction des lignes de niveau en résolvant l'équation aux dérivées partielles

$$I_t = \nabla(\Delta I) \cdot \nabla I^\perp \quad (5.3)$$

La notation ∇I^\perp correspond au vecteur ∇I qui a subi une rotation de $\pi/2$ radians. Pour améliorer la stabilité de leur algorithme numérique ils ne normalisent pas le gradient orthogonal. À l'état initial l'image dans le trou a une valeur constante.

La discrétisation en temps est décentrée avant

$$I_{i,j}^{n+1} = I_{i,j}^n + \Delta t I'_{i,j} \quad (5.4)$$

où $I'_{i,j}$ est la discrétisation de $\nabla(\Delta I) \cdot \nabla I^\perp$. Pour la discrétisation du laplacien et du gradient le schéma utilisé est le schéma centré classique. Les auteurs calculent en fait $\left(\nabla(\Delta I) \cdot \frac{\nabla I^\perp}{\|\nabla I^\perp\|} \right) \|\nabla I\|$. Ce qui leur permet d'utiliser des schémas décentrés pour le calcul de $\|\nabla I\|$ en fonction du signe de $\nabla(\Delta I) \cdot \frac{\nabla I^\perp}{\|\nabla I^\perp\|}$. Pour ne pas prolonger les lignes de niveau qu'en ligne droite et leur permettre de s'incurver ils alternent leur schéma avec une diffusion anisotrope qui dépend de la courbure. L'équation qu'ils utilisent est

$$I_t = \text{curv}(I) \|\nabla I\|.$$

Ils utilisent la première équation pour faire évoluer I sur quelques itérations, puis ils font quelques itérations de la deuxième et ainsi de suite.

Il est possible d'appliquer ce schéma aux images multi-canaux, par exemple les images couleurs en l'appliquant canal par canal. Dans le cas des images couleurs, pour éviter l'apparition de couleurs parasites il est préférable d'utiliser le modèle HSV plutôt que les canaux rouge, vert et bleu.

5.4 Interpolation conjointe d'un champ de vecteurs et des niveaux de gris

Les auteurs de [4] résolvent le problème de la désocclusion grâce à l'équation aux dérivées partielles (5.3) qui est d'ordre 3. Ils suggèrent alors pour diminuer l'ordre de découpler le problème de la désocclusion en résolvant deux équations d'ordre deux : une équation pour les directions des lignes de niveau et l'autre pour les niveaux de gris de l'image. Cette idée est exploitée par les auteurs de [2].

Dans [2] Ballester, Bertalmio, Caselles, Sapiro et Verdera étudient donc de façon découplée la direction des lignes de niveau et les niveaux de gris de l'image. Dans ce but, à la fonction I , l'image en niveau de gris, est associée une fonction $\theta : \mathcal{D} \rightarrow \mathbb{R}^2$ qui correspond à la direction du gradient de

l'image. Le champ d'orientation θ est relié à l'image par les contraintes suivantes, $\theta \cdot \nabla I = \|\nabla I\|$ et $\|\theta\| \leq 1$. Ainsi, lorsque $\|\nabla I\| \neq 0$, $\theta = \nabla I / \|\nabla I\|$. Les auteurs proposent de minimiser une énergie, composée d'un terme de régularisation des lignes de niveau et d'un terme de pénalisation pour la contrainte $\theta \cdot \nabla I = \|\nabla I\|$

$$\int_{\mathcal{D}} \|\operatorname{div}(\theta)\|^p (a + b\|\nabla k * I\|) dx + \alpha \int_{\mathcal{D}} (\|\nabla I\| - \theta \cdot \nabla I) dx,$$

où k est un noyau régularisant.

Comme θ correspond à $\nabla I / \|\nabla I\|$ le terme $\operatorname{div}(\theta)$ correspond à

$$\operatorname{div}(\nabla I / \|\nabla I\|) = \operatorname{curv}(I),$$

où $\operatorname{curv}(I)$ est la courbure des lignes de niveau. Comme k est un noyau régularisant, le terme $\|\nabla k * I\| = \|\nabla(k * I)\|$ est donc la norme du gradient régularisé de l'image I . Le premier terme peut être formellement décrit comme $\int_{\mathcal{D}} \|\operatorname{curv}(I)\|^p (a + b\|\nabla I\|) dx$ et peut être comparé à l'énergie minimisée par Masnou et Morel $\int_{\mathcal{D}} \|\nabla I\| (1 + \|\operatorname{curv}(I)\|)$. Dans les deux cas interviennent la longueur et la courbure des lignes de niveau. La différence est que les auteurs de [2] minimisent prioritairement la courbure des lignes de niveau et ensuite la longueur alors que Masnou et Morel dans [27] minimisent prioritairement la longueur des lignes de niveau comme nous l'avons déjà remarqué.

La minimisation de l'énergie est effectuée grâce à un algorithme de descente de gradient de plus grande pente. L'introduction du noyau régularisant est nécessaire pour que le problème de minimisation ait une solution mais les auteurs utilisent un noyau gaussien avec une faible variance d'un ou deux pixels. L'image I est découpée en ensembles de niveau. Une application directe de leur algorithme sur l'image I aurait pour effet que la reconstruction d'un bord dépendrait de son intensité. L'algorithme ne serait alors pas invariant par changement de contraste. L'algorithme est appliqué à chacun des ensembles de niveau avant de reformer l'image finale.

5.5 Diffusion dirigée par courbure

La méthode exposée par Chan et Shen dans [8] a pour origine les méthodes de restauration d'image et en particulier le modèle de Rudin, Osher et Fatemi basé sur la variation totale d'une fonction. Chan et Shen appellent leur méthode le *TV-inpainting* (TV : *total variation*, variation totale en anglais). La variation totale d'une fonction correspond dans le cas d'une fonction \mathcal{C}^1 à l'intégrale

$$\int_{\mathcal{D}} \|\nabla I\| dx$$

L'équation d'Euler-Lagrange associée à la minimisation de ce terme est formellement

$$\operatorname{div} \left(\frac{\nabla I}{\|\nabla I\|} \right) (= \operatorname{curv}(I)) = 0.$$

La désocclusion par *TV inpainting* est effectuée par la résolution d'une équation aux dérivées partielles d'ordre deux.

$$I_t = \begin{cases} \operatorname{curv}(I) + \lambda(I - I_0) & \text{dans } \mathcal{D} \setminus \Omega \\ \operatorname{curv}(I) & \text{dans } \Omega \end{cases}$$

Cet algorithme permet de faire la désocclusion dans Ω tout en restaurant l'image sur $\mathcal{D} \setminus \Omega$. L'inconvénient de cette méthode est qu'elle ne permet pas de reconnecter certains objets que l'œil humain reconnecte. Dans la figure 5.7 si le trou est plus haut que large l'algorithme de *TV inpainting* ne reconstruit pas la barre, le résultat est l'image à droite. Tandis que l'œil humain reconstitue la barre et imagine plutôt la reconstruction de l'image au centre. Cette remarque conduit Chan et Shen à changer le terme de diffusion. L'équation qu'ils proposent est alors

$$I_t = \begin{cases} \operatorname{curv}(I) + \lambda(I - I_0) & \text{dans } \mathcal{D} \setminus \Omega \\ \operatorname{div} \left(g(\|\operatorname{curv}(I)\|) \frac{\nabla I}{\|\nabla I\|} \right) & \text{dans } \Omega \end{cases}$$

où g est telle que $g(0) = 0$, $\lim_{s \rightarrow \infty} g(s) = \infty$ et pour $s \in]0, \infty[$, $g(s) \in]0, \infty[$, par exemple $g(s) = s^p$ avec $p \geq 1$. Ils appellent cet algorithme diffusion dirigée par courbure (en anglais *curvature driven diffusions* : CDD).

Le schéma décentré avant est utilisé pour la discrétisation en temps,

$$I_{i,j}^{n+1} = I_{i,j}^n + \Delta t I'_{i,j}^n.$$

Pour le terme de divergence, un schéma centré classique au demi-pixel est mis en œuvre. Pour le calcul de $\operatorname{div}(J)$ avec $J = (J^1, J^2)$ cela donne

$$\frac{J_{i+\frac{1}{2},j}^1 - J_{i-\frac{1}{2},j}^1}{\Delta x} + \frac{J_{i,j+\frac{1}{2}}^2 - J_{i,j-\frac{1}{2}}^2}{\Delta y}.$$

L'initialisation choisie est de remplir le trou avec des valeurs aléatoires. Ensuite ils appliquent la méthode de *TV inpainting*. Celle-ci est plus rapide que leur algorithme CDD. Ils utilisent le résultat du *TV inpainting* comme point de départ pour leur algorithme CDD.

5.6 Méthode inspirée de la dynamique des fluides

L'article [3] a pour point de départ l'équation de [4]

$$I_t = \nabla(\Delta I) \cdot \nabla I^\perp$$

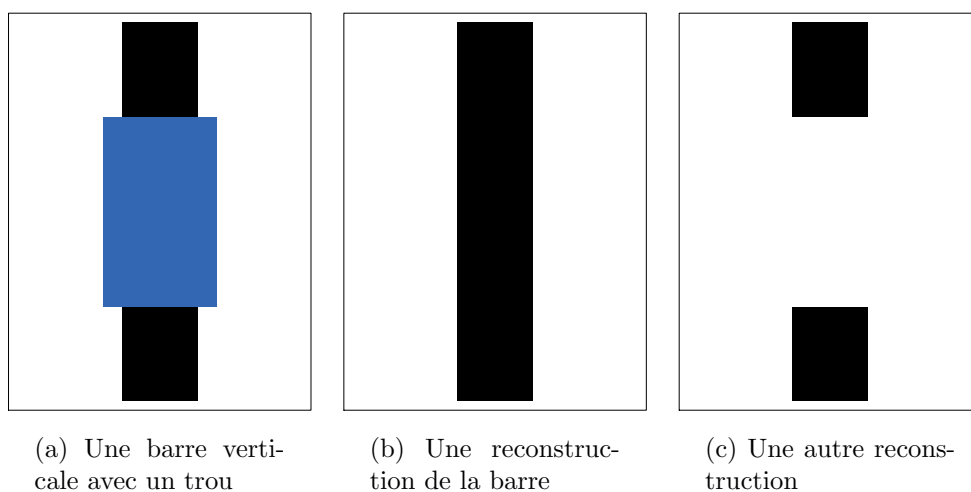


FIG. 5.7 – À gauche, une image avec un trou. Au milieu et à droite, deux reconstructions possibles.

Comme dans [4] cette équation est couplée avec de la diffusion anisotrope. Le modèle peut être schématisé par l'équation

$$I_t = \nabla(\Delta I) \cdot \nabla I^\perp + \nu \operatorname{div}(g(\|\nabla I\|)\nabla I).$$

Le travail effectué dans [3] est basé sur la remarque d'une similarité avec une équation de Navier-Stokes

$$\omega_t + v \cdot \nabla \omega = \nu \Delta \omega$$

où ω est la vorticit  de une fonction de flux ψ ($\omega = \Delta \psi$) et $v = \nabla \psi^\perp$ est la vitesse du fluide.

En mettant en correspondance ψ et I , v et ∇I^\perp et enfin ω et $w = -\Delta I$ ils proposent les  quations

$$w_t + \nabla I^\perp \cdot \nabla w = \nu \operatorname{div}(g(\|\nabla w\|)\nabla w), \quad (5.5)$$

$$\Delta I = -w, \quad I_\Gamma = I_0. \quad (5.6)$$

Pour la r solution de ce syst me la contrainte $\Delta I = -w$ pose probl me, la r solution directe est trop lente. Pour r soudre cette  quation en pratique, elle est remplac e par l' quation suivante

$$I_t = \alpha (\Delta I + w).$$

Mais cette op ration introduit du flou au niveau des bords de l'image. Pour compenser ce d faut cette  quation est altern e avec de la diffusion anisotrope. Le sch ma en temps est une diff rence finie d centr e avant. Les termes de divergence sont discr tiss s par des sch mas centr s classiques. Enfin pour les termes de transport des sch mas min-mod sont utilis s.

5.7 Conclusion

Plusieurs méthodes utilisent les lignes de niveau de l'image. La méthode de Masnou et Morel n'utilise que les lignes de niveau. Ce sont les lignes de niveau qui sont complétées avant de reconstruire l'image. Dans la méthode exposée par Bertalmio, Sapiro, Ballester et Caselles dans [4] seule l'orientation des lignes de niveau est utilisée. Cette idée est reprise dans [2] où les auteurs traitent séparément l'orientation des lignes de niveau et les niveaux de l'image.

Dans l'algorithme que nous allons présenter dans le chapitre suivant nous séparons aussi l'orientation des lignes de niveau de l'image et les niveaux de l'image. Cependant nous voulons prendre en compte l'incertitude sur l'orientation des lignes de niveau. En effet nous pouvons connaître cette incertitude pour ensuite prendre en compte ou non cette information lors de la reconstruction.

Chapitre 6

Présentation de notre modèle

Le modèle que nous présentons pour la désocclusion reprend l'idée générale des autres algorithmes de désocclusion qui est de continuer les lignes de niveau de l'image en partant du bord de l'image. Notre modèle reprend aussi l'idée de séparer la continuation des lignes de niveau de l'image et la continuation du niveau de gris de l'image. Cependant nous prenons en compte dans notre modèle l'imprécision de l'orientation que nous reconstruisons pour les lignes de niveau. Ceci nous conduit, au lieu de considérer une deuxième fonction correspondant à l'orientation des lignes de niveau, à considérer une deuxième fonction correspondant aux différentes orientations possibles avec leurs probabilités.

6.1 Principe de l'algorithme

Nous notons I l'image que nous voulons reconstruire. Comme dans les approches de Masnou et Morel [27] et de Bertalmio, Sapiro, Ballester et Caselles [4] notre modèle est basé sur l'utilisation des lignes de niveau. Les lignes de niveau d'une image sont définies à partir des ensembles de niveau de cette image. Les ensembles de niveau sont de deux types, les ensembles de niveau supérieurs

$$\chi_\lambda I = \{x | I(x) \geq \lambda\},$$

et les ensembles de niveau inférieurs

$$\chi_\lambda^- I = \{x | I(x) \leq \lambda\}.$$

Nous pouvons utiliser indifféremment les ensembles de niveau supérieurs ou inférieurs. Nous choisissons de travailler à partir des ensembles de niveau supérieurs. La famille des ensembles de niveau d'une image possède plusieurs caractéristiques. Une propriété fondamentale est qu'il est possible de reconstruire l'image à partir de cette famille grâce à la formule

$$I(x) = \sup\{\lambda | x \in \chi_\lambda I\}. \quad (6.1)$$

Les lignes de niveau de l'image sont les frontières des ensembles de niveau. Les images du bas de la figure 6.1 correspondent aux lignes de niveau des deux images du haut. Toutes les lignes de niveau ne sont pas représentées. Seul 1 niveau sur 50 est représenté.

Une autre caractéristique des ensembles de niveau est l'invariance par changement de contraste. Changer le contraste d'une image revient à la composer avec une fonction strictement croissante. Si g est une telle fonction, alors $J = g \circ I$ possède les mêmes ensembles de niveau que I . En normalisant les valeurs prises par une image entre 0 et 1, il faut aussi imposer que $g(0) = 0$ et $g(1) = 1$.

L'image en haut à droite de la figure 6.1 correspond à l'image en haut à gauche avec comme changement de contraste la fonction $g(x) = x^3$. À chaque ensemble de niveau de l'image de gauche correspond un ensemble de niveau de l'image de droite et vice-versa. Les images du bas montrent une ligne de niveau sur 50 de chacune des images. Même si ce ne sont pas des lignes d'ensembles qui se correspondent, les similitudes sont grandes.

Dans [27] les auteurs reconstruisent les lignes de niveau pour ensuite remplir les niveaux correspondants. La figure 5.5 illustre ce type de méthode en le faisant *à la main*. L'idée sous-jacente dans [4] est la même. Cependant au lieu de reconstruire les lignes, puis de remplir le trou, les deux se font simultanément. Le problème est que les lignes de niveau dans le trou sont inconnues. Il s'agit donc de reconstruire ces lignes de niveau et en même temps de propager l'information voulue. Notre algorithme est lui aussi de ce type.

Dans notre méthode nous reconstruisons les lignes de niveau en même temps que nous reconstruisons l'image. Les deux reconstructions (lignes et niveaux) sont liées de la manière suivante. D'une part l'image définit les lignes de niveau et d'autre part les lignes (ensembles) de niveau permettent de reconstruire l'image.

6.1.1 Orientation des lignes de niveau

Soit \mathcal{C} une ligne de niveau pour la valeur λ de l'image I . Nous savons que le long de la courbe \mathcal{C} l'image I prend toujours la valeur λ .

$$\forall x \in \mathcal{C} \quad I(x) = \lambda.$$

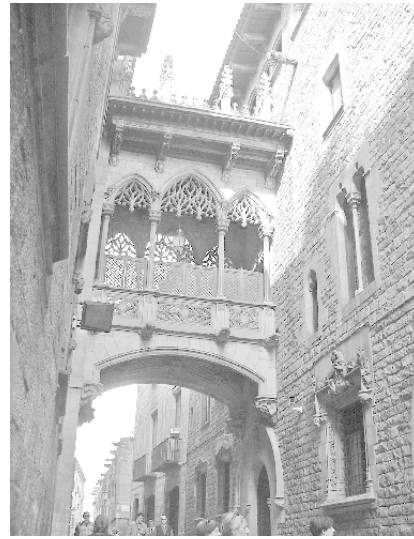
Soit $t \mapsto x(t)$ une paramétrisation normale de \mathcal{C} au voisinage d'un point x_0 . Alors pour tout t , $I(x(t)) = \lambda$ et en dérivant cette égalité en x_0 nous obtenons

$$\vec{\tau} \cdot \nabla I = 0$$

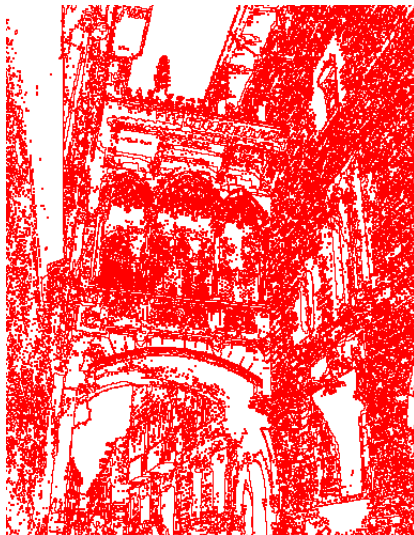
où $\vec{\tau}$ est le vecteur tangent à \mathcal{C} en x_0 . Le vecteur $\vec{\tau}$ représente l'orientation de la courbe de niveau \mathcal{C} et le gradient de l'image est orthogonal à la ligne de niveau. Cela nous donne l'orientation des lignes de niveau modulo π .



(a) Image originale



(b) Image avec changement de contraste



(c) Lignes de niveau avant changement de contraste



(d) Lignes de niveau après changement de contraste

FIG. 6.1 – Lignes de niveau et changement de contraste. En haut se trouvent deux images. À gauche l'image originale, une photographie prise à Barcelone et à droite la même image après un changement de contraste. Le changement de contraste utilisé est $x \mapsto x^3$. En dessous de chacune de ces images sont affichées leurs lignes de niveau (une sur cinquante en fait). À chaque ligne de niveau d'une des images correspond une ligne de niveau de l'autre image. En effet même si les lignes de niveau représentées sur les figures du bas ne sont pas forcément des lignes de niveau qui se correspondent, la similitude entre les lignes de niveau est grande.

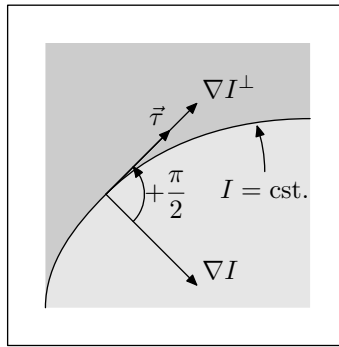


FIG. 6.2 – Orientation d’une ligne de niveau. Sur cette image la ligne entre les deux niveaux de gris représente une ligne de niveau de l’image. Le gradient de l’image le long de cette ligne de niveau est orthogonal à la ligne. Nous prenons comme vecteur tangent à la courbe le gradient de l’image après une rotation de $\pi/2$ radians.

Par convention nous prenons pour orienter les lignes de niveau l’orientation du gradient de l’image plus $\pi/2$ radians. L’orientation des lignes de niveau est donc égale à celle du gradient plus $\pi/2$ radians (cf. figure 6.2).

6.2 Incertitude sur l’orientation d’un vecteur

Pour connaître l’orientation des lignes de niveau nous devons calculer l’orientation du gradient de l’image. Dans le cas où le gradient n’est pas nul cette orientation est fournie par le vecteur unitaire $\nabla I / \|\nabla I\|$. Dans le cas contraire, si le gradient est nul, il n’y a pas d’orientation car il n’y a pas de lignes de niveau en ce point.

Nous supposons désormais que le gradient n’est pas nul. L’orientation des lignes de niveau est donc donnée par $\nabla I / \|\nabla I\|$ ($+\pi/2$). Ce calcul n’est subordonné qu’au calcul du gradient de l’image ∇I . Cependant il est bien connu dans le cadre de l’analyse d’image qu’il n’est pas possible de connaître exactement ∇I . En effet nous ne connaissons déjà pas I . Nous connaissons seulement les valeurs de I sur un réseau du domaine \mathcal{D} . Nous devons donc nous contenter d’une approximation. Nous avons choisi d’utiliser un schéma aux différences finies. Tout autre schéma ne donnera lui aussi qu’une approximation. Ainsi il y aura toujours une erreur par rapport à la valeur exacte du gradient de l’image. Formulons plus précisément cette incertitude.

PROPOSITION 32 :

Soient \vec{U} et \vec{V} deux vecteurs non nuls tels que

$$\vec{V} = \vec{U} + \vec{\varepsilon},$$

alors

$$\left\| \frac{\vec{V}}{\|\vec{V}\|} - \frac{\vec{U}}{\|\vec{U}\|} \right\| \leq \frac{e}{\|\vec{U}\|} + o(e),$$

où e est la norme de $\vec{\varepsilon}$.

Le vecteur \vec{V} correspond au résultat de calcul de ∇I par un schéma aux différences finies ou autre. C'est une approximation du vecteur inconnu \vec{U} . Et $\vec{\varepsilon}$ est l'erreur commise lors du calcul de \vec{U} .

Preuve : Nous commençons par calculer la norme de \vec{V}

$$\|\vec{V}\| = \|\vec{U} + \vec{\varepsilon}\| \quad (6.2)$$

$$= \sqrt{\langle \vec{U} + \vec{\varepsilon}, \vec{U} + \vec{\varepsilon} \rangle} \quad (6.3)$$

$$= \left(\|\vec{U}\|^2 + 2 \langle \vec{U}, \vec{\varepsilon} \rangle + o(e) \right)^{1/2} \quad (6.4)$$

$$= \|\vec{U}\| \left(1 + \frac{2}{\|\vec{U}\|} \left\langle \frac{\vec{U}}{\|\vec{U}\|}, \vec{\varepsilon} \right\rangle + o(e) \right)^{1/2} \quad (6.5)$$

$$= \|\vec{U}\| \left(1 + \frac{1}{\|\vec{U}\|} \left\langle \frac{\vec{U}}{\|\vec{U}\|}, \vec{\varepsilon} \right\rangle + o(e) \right) \quad (6.6)$$

Et donc

$$\frac{\vec{V}}{\|\vec{V}\|} = \frac{(\vec{U} + \vec{\varepsilon})}{\|\vec{U}\|} \left(1 + \frac{1}{\|\vec{U}\|} \left\langle \frac{\vec{U}}{\|\vec{U}\|}, \vec{\varepsilon} \right\rangle + o(e) \right)^{-1} \quad (6.7)$$

$$= \frac{(\vec{U} + \vec{\varepsilon})}{\|\vec{U}\|} \left(1 - \frac{1}{\|\vec{U}\|} \left\langle \frac{\vec{U}}{\|\vec{U}\|}, \vec{\varepsilon} \right\rangle + o(e) \right) \quad (6.8)$$

$$= \frac{\vec{U}}{\|\vec{U}\|} + \frac{1}{\|\vec{U}\|} \left(\vec{\varepsilon} - \left\langle \frac{\vec{U}}{\|\vec{U}\|}, \vec{\varepsilon} \right\rangle \frac{\vec{U}}{\|\vec{U}\|} \right) + o(e) \quad (6.9)$$

Or le couple $\left(\vec{U}/\|\vec{U}\|, \vec{U}^\perp/\|\vec{U}\| \right)$ forme une base orthonormée de \mathbb{R}^2 et la formule que nous venons d'obtenir peut donc s'écrire

$$\frac{\vec{V}}{\|\vec{V}\|} = \frac{\vec{U}}{\|\vec{U}\|} + \frac{1}{\|\vec{U}\|} \pi_{\vec{U}^\perp}(\vec{\varepsilon}) + o(\|\vec{\varepsilon}\|). \quad (6.10)$$

Ici,

$$\pi_{\vec{U}^\perp}(\vec{\varepsilon}) = \vec{\varepsilon} - \left\langle \frac{\vec{U}}{\|\vec{U}\|}, \vec{\varepsilon} \right\rangle \frac{\vec{U}}{\|\vec{U}\|},$$

est la projection orthogonale du vecteur $\vec{\varepsilon}$ sur la droite orthogonale à \vec{U} . C'est un opérateur linéaire de norme 1. Ce qui nous donne le résultat de la proposition 32. \square

Un corollaire immédiat de la proposition 32 est

$$\left\| \frac{\vec{V}}{\|\vec{V}\|} - \frac{\vec{U}}{\|\vec{U}\|} \right\| \leq \frac{e}{\|\vec{V}\|} + o(e). \quad (6.11)$$

Ceci nous permet de majorer l'erreur en ne connaissant que $\|\vec{V}\|$. En effet nous ne connaissons pas \vec{U} . Pour résumer, la précision obtenue sur l'orientation d'un vecteur est inversement proportionnelle à la norme du vecteur.

6.3 Champ de probabilité associé

Nous nous plaçons en un point de l'image. Par ce point passe une ligne de niveau. Nous avons une orientation θ_c en ce point. L'orientation θ_c appartient au cercle unité $\mathcal{S}^1 = \mathbb{R}/2\pi\mathbb{Z}$ et est calculée avec une précision e_θ (cf. (6.11)). L'orientation réelle de la ligne de niveau n'est pas forcément θ_c mais une autre orientation θ_r proche ($\|\theta_r - \theta_c\| \leq e_\theta$).

Nous devons modéliser mathématiquement cette situation afin de l'utiliser dans un algorithme numérique. Pour cela nous disons que l'orientation θ_r a une certaine probabilité d'être répartie autour de θ_c . Nous supposons que la moyenne de cette probabilité est l'orientation calculée et son écart-type la précision e_θ du calcul.

Au lieu de considérer une orientation nous considérons une probabilité d'orientation en tout point de l'image. Ainsi au lieu d'avoir un champ d'orientation nous avons un champ de probabilités.

DÉFINITION 33 :

Un champ de probabilité P d'orientation sur le domaine \mathcal{D} de \mathbb{R}^2 est une fonction

$$\begin{aligned} P : \mathcal{D} &\rightarrow \mathcal{F}(\mathcal{S}^1, \mathbb{R}) \\ x &\mapsto (\theta \mapsto P(x, \theta)) \end{aligned} \quad (6.12)$$

qui associe à tout point de \mathcal{D} une probabilité sur \mathcal{S}^1 .

L'ensemble $\mathcal{F}(\mathcal{S}^1, \mathbb{R})$ est l'ensemble des fonctions de \mathcal{S}^1 dans \mathbb{R} . Nous devons imposer quelques contraintes à P .

- Comme $\theta \mapsto P(x, \theta)$ est une probabilité, nous voulons que pour tout $x \in \mathcal{D}$

$$\int_{\mathcal{S}^1} P(x, \theta) d\theta = 1. \quad (6.13)$$

- Nous voulons également imposer une valeur moyenne pour P .
Soit $\vec{u}(\theta) = (\cos(\theta), \sin(\theta))$ le vecteur unitaire d'orientation θ . Si nous posons $\vec{u}_P(x) = \int_{\mathcal{S}^1} \vec{u}(\theta) P(x, \theta) d\theta$ alors \vec{u}_P n'est pas nécessairement un vecteur unitaire et si $\vec{u}_P = 0$ alors la probabilité n'est pas intégrable. Si $\vec{u}_P \neq 0$ nous définissons la moyenne de P en x par l'orientation de $\vec{u}_P(x)$. Si θ_P est cette moyenne alors nous avons $\vec{u}_P(x) = \|\vec{u}_P(x)\| \vec{u}(\theta_P(x))$. Nous imposons comme valeur moyenne l'orientation calculée avec le gradient (θ_r) i.e.

$$\vec{u}(\theta_P) = \frac{\nabla I^\perp}{\|\nabla I\|} \quad (6.14)$$

- Ensuite il faut aussi imposer un écart-type pour ces probabilités.
Nous calculons l'écart-type de la manière suivante

$$\sigma_P^2 = \int_{\mathcal{S}^1} \text{dist}^2(\theta, \theta_P(x)) P(x, \theta) d\theta.$$

La distance dist est la distance sur le cercle unité \mathcal{S}^1 , elle est donc majorée par π radians. Si l'écart-type de l'erreur sur le gradient est σ alors nous imposons σ_P d'après (6.11)

$$\sigma_P = \frac{\sigma}{\|\nabla I\|}. \quad (6.15)$$

Sous ces hypothèses le champ de probabilités est compatible avec l'image I .

DÉFINITION 34 :

Un champ de probabilités sur le domaine \mathcal{D} est I -compatible où I est une fonction \mathcal{C}^1 sur \mathcal{D} si

1. *la moyenne θ_P de P est définie si et seulement si ∇I est non nul et dans ce cas*

$$\vec{u}(\theta_P) = \frac{\nabla I^\perp}{\|\nabla I\|},$$

2. *l'écart-type σ_P vérifie en ces points*

$$\sigma_P = \frac{\sigma}{\|\nabla I\|}.$$

Pour être un champ de probabilité I -compatible une fonction P est donc soumise à trois contraintes. La première signifie que $\{\theta \mapsto P(x, \theta)\}$ est une famille de probabilités. La deuxième contrainte porte sur leurs moyennes, celles-ci doivent être reliées à l'orientation du gradient de l'image. Et enfin la troisième contrainte imposée aux probabilités est leurs écarts-types qui doivent être reliés à la norme du gradient de l'image.

Le moyen le plus simple d'avoir une probabilité avec une moyenne et un écart-type fixés est de prendre une loi normale. Nous allons donc prendre initialement si $\nabla I \neq 0$

$$P(x, \theta) = \frac{1}{\mathcal{Z}} \exp(-\text{dist}^2(\theta, \theta_I + \pi/2) \|\nabla I\|^2 / \sigma^2) \quad (6.16)$$

où σ est l'écart-type de l'erreur sur le gradient, θ_I est l'orientation du gradient et \mathcal{Z} une constante de normalisation. Nous notons dans ce cas

$$P(x, \cdot) \sim \mathcal{N}(\theta_I + \pi/2, \sigma / \|\nabla I\|).$$

Si $\nabla I = 0$ alors nous choisissons

$$P(x, \theta) = \frac{1}{2\pi}, \quad (6.17)$$

que nous notons

$$P(x, \cdot) \sim \mathcal{U}.$$

Ce qui correspond à la limite de la loi précédente lorsque le gradient tend vers 0. Par contre nous n'imposons pas par la suite que les distributions de probabilité restent des lois normales ou uniformes.

Sur la figure 6.3, en haut à gauche se trouve un extrait d'une image naturelle. Pour faire de la désocclusion nous utilisons le gradient de l'image. Celui-ci est superposé à l'image sur l'image en haut à gauche. À partir du gradient de l'image nous calculons son orientation et sa norme. L'image du bas de la figure montre l'image de la norme du gradient et superposés les vecteurs unitaires représentant l'orientation du gradient. La figure 6.4 montre deux distributions de probabilité calculées à partir de l'image naturelle pour l'initialisation.

6.4 Désocclusion du champ de probabilité

Nous disposons maintenant d'un champ de probabilité d'orientation pour les lignes de niveau de l'image. Nous souhaitons propager de l'information à l'intérieur du trou Ω depuis l'extérieur $\mathcal{D} \setminus \Omega$ en suivant les lignes de niveau de l'image. Désormais nous n'avons plus d'orientation pour les lignes de niveau mais une probabilité d'orientation.

Transport. L'idée est de ne pas propager l'information dans une direction mais dans toutes les directions selon leurs probabilités. L'équation de transport classique pour transporter une information $L : \mathcal{D} \rightarrow \mathbb{R}$ est

$$\vec{\tau} \cdot \nabla L = 0$$

où $\vec{\tau}$ est l'orientation le long de laquelle nous voulons propager l'information L .

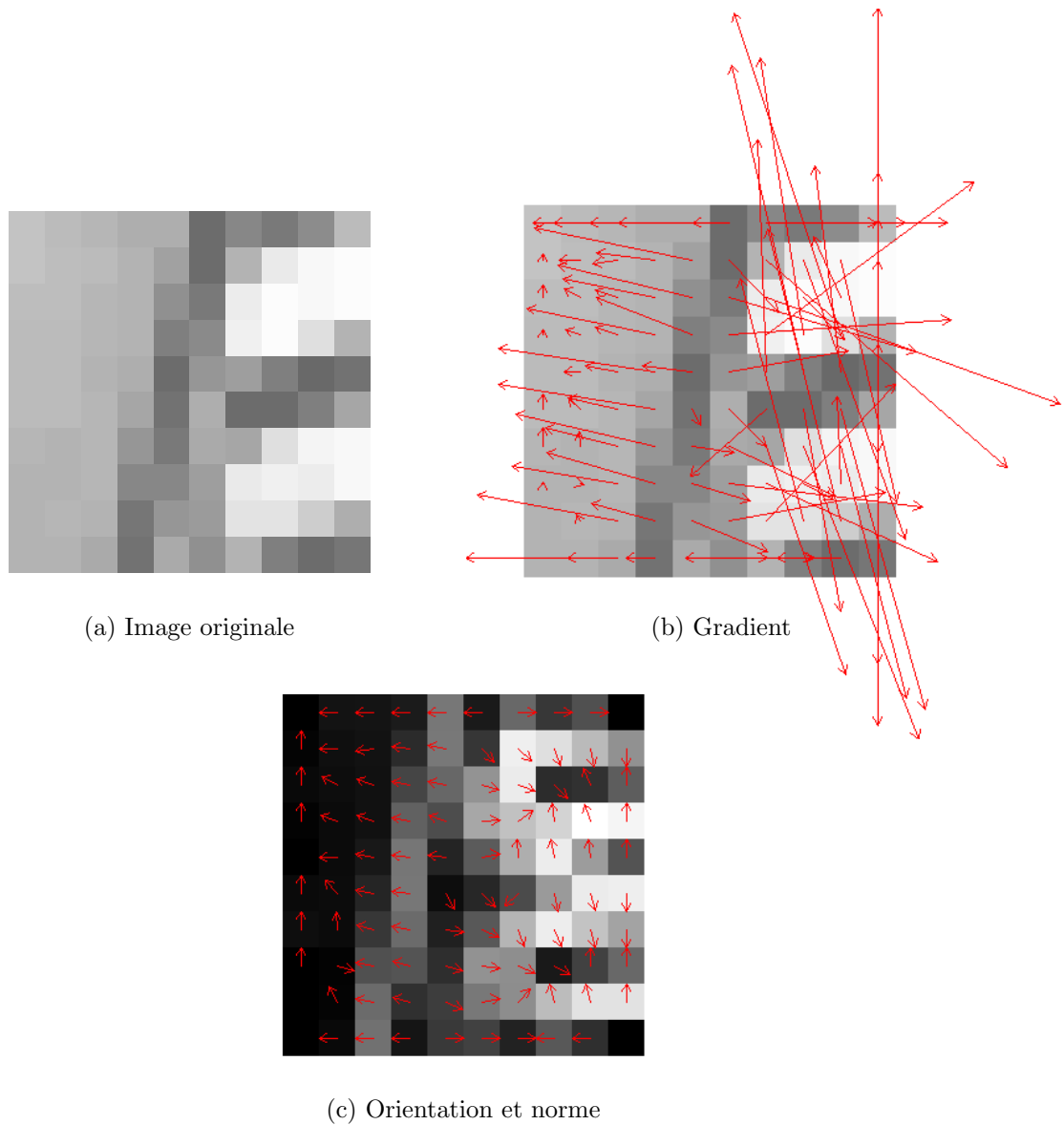


FIG. 6.3 – Gradient d’une image. En haut à gauche un extrait d’une image en niveau de gris, à sa droite le gradient de l’image. À chaque pixel de l’image est associé un vecteur gradient dans cette image ces vecteurs sont superposés à l’image originale. En bas l’image de fond est l’image de la norme du gradient et au dessus sont représentés des vecteurs unitaires dont l’orientation est celle du gradient. Le même type d’information est fourni par cette image que par celle du champ de vecteurs du gradient de l’image en haut à droite. Avec des vecteurs unitaire le champ d’orientation est plus visible que lorsque les vecteurs se croisent et se chevauchent.

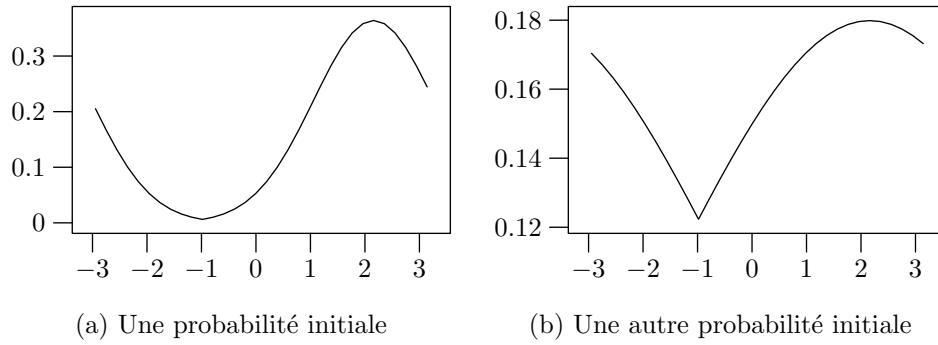


FIG. 6.4 – Deux exemples de distributions de probabilité calculées sur l’image naturelle de la figure 6.3.

Nous choisissons $\vec{\tau} = P(x, \theta)\vec{u}(\theta)$ car nous propageons l’information dans toutes les directions selon leurs probabilités. L’information que nous souhaitons propager est la loi de probabilité $P(x, \theta)$. Ce choix est naturel car c’est le champ de probabilité lui-même que nous souhaitons compléter. Nous avons donc à résoudre

$$P(x, \theta)\vec{u}(\theta) \cdot \nabla_x P = 0,$$

Le symbole ∇_x ne concerne que les dérivées spatiales,

$$\nabla_x P = (\partial_{x_1} P, \partial_{x_2} P).$$

Cette équation de transport n’est pas stable. Pour la rendre stable nous ajoutons un terme de diffusion.

Diffusion. Pour obtenir une équation stable nous introduisons, comme il est classique, de la diffusion. C’est ce que font également les auteurs de [4, 3]. L’équation devient désormais

$$P(x, \theta)\vec{u}(\theta) \cdot \nabla_x P - \alpha \frac{\partial^2 P}{\partial \theta^2} - \beta \Delta_x P = 0. \quad (6.18)$$

Une fois encore le terme Δ_x ne concerne que les dérivées spatiales,

$$\Delta_x P = \frac{\partial^2 P}{\partial x_1^2} + \frac{\partial^2 P}{\partial x_2^2}.$$

On retrouve dans le premier terme l’équation de transport de la distribution de probabilité dans toutes les directions en fonction de leurs probabilités. Les deux derniers termes correspondent à de la diffusion, entre les angles pour le premier et spatialement pour le second. Les coefficients de diffusion α et β

permettent de pondérer la diffusion par rapport au transport mais aussi entre elles.

Le terme de diffusion par rapport aux angles permet aux lignes de niveau de se courber. Plaçons-nous par exemple dans l'hypothèse où il n'y a aucun terme de diffusion. Imaginons également que les distributions de probabilité soient concentrées en une seule orientation. Dans ce cas le terme de transport ne ferait que transporter cette distribution de probabilité en ligne droite, selon son orientation. Et donc une fois l'image reconstruite les lignes de niveau seront rectilignes. Par contre avec le terme de diffusion entre les différentes orientations, la distribution de probabilité n'est plus simplement propagée en ligne droite, elle s'étend aussi aux orientations voisines. Les lignes de niveau ne seront alors plus totalement rectilignes.

Temps. Nous allons résoudre cette équation en utilisant un schéma itératif. Ceci correspond à ajouter artificiellement une variable de temps dans l'équation. Pour être exhaustif l'équation finale est

$$\begin{cases} \frac{\partial P}{\partial t} - \alpha \frac{\partial^2 P}{\partial \theta^2} - \beta \Delta_x P + P \vec{u}(\theta) \cdot \nabla_x P = 0 & \text{sur } \Omega \times \mathcal{S}^1 \times [0, \infty[, \\ P(x, \theta, t) = f_P(x, \theta, t) & \text{sur } \partial \Omega \times \mathcal{S}^1 \times [0, \infty[, \\ P(x, \theta, 0) = P_0(x, \theta) & \text{sur } \Omega \times \mathcal{S}^1. \end{cases} \quad (6.19)$$

Les fonctions P_0 et f_P sont les valeurs du champ de probabilité respectivement au temps $t = 0$ et sur le bord de Ω . Celles-ci sont définies dans le chapitre suivant.

6.5 Reconstruction de l'image

Notre algorithme est opérationnel si nous pouvons reconstruire l'image à partir du champ de probabilité complété. Nous devons donc d'abord déduire une orientation de chaque probabilité. Ensuite nous reconstruisons l'image à partir du champ d'orientation. Or nous savons que parmi les contraintes sur P , sa moyenne doit être égale à l'orientation du gradient de I augmenté de $\pi/2$. Il est donc logique de prendre comme orientation pour le gradient de l'image I la moyenne de P moins $\pi/2$.

Pour propager l'image le long de ces orientations nous utilisons l'équation de transport

$$\nabla I \cdot \vec{u}(\theta_P) = 0.$$

Puisqu'une équation de transport n'est pas stable numériquement, nous ajoutons de la diffusion comme nous l'avons fait pour l'équation sur P dans la section précédente. Ce qui donne

$$\nabla I \cdot \vec{u}(\theta_P) - \nu \Delta I = 0.$$

Le premier terme correspond à un transport de l'image I selon l'orientation moyenne de chacune des probabilités du champ. Le second terme est un terme classique de diffusion. Dans notre cas la connaissance de la probabilité P nous permet d'avoir une mesure de la pertinence de la direction moyenne θ_P . En effet nous pouvons calculer l'écart-type σ_P de la probabilité P . Si l'écart-type est élevé cela signifie que la précision de la direction moyenne θ_P n'est pas très fiable. Dans ce cas nous pouvons privilégier le terme de diffusion par rapport au terme de transport. Inversement si l'écart-type est faible, voire nul, cela signifie que la probabilité est très resserrée autour de la moyenne θ_P , voire que P est une probabilité triviale, dans ce cas nous pouvons minimiser l'effet du terme de diffusion. Pour faire cela, le plus simple est de multiplier le terme de diffusion par σ_P , ce qui donne désormais

$$\nabla I \cdot \vec{u}(\theta_P) - \sigma_P \nu \Delta I = 0.$$

Nous allons résoudre cette équation en utilisant un schéma itératif. Ceci correspond à ajouter artificiellement une variable de temps dans l'équation. Pour être exhaustif l'équation finale est

$$\begin{cases} \frac{\partial I}{\partial t} - \sigma_P \nu \Delta I + \vec{u}(\theta_P) \cdot \nabla I = 0 & \text{sur } \Omega \times [0, \infty[, \\ I(x, t) = f_I(x, t) & \text{sur } \partial \Omega \times [0, \infty[, \\ I(x, 0) = I_0(x) & \text{sur } \Omega. \end{cases} \quad (6.20)$$

La valeur initiale de I est la fonction I_0 prolongée dans Ω par une valeur constante ou par des valeurs aléatoires et la fonction f_I est la valeur au bord de I .

6.6 Système de deux équations

Le système complet que nous avons à résoudre comprend les deux équations (6.19) sur P et (6.20) sur I ainsi que les trois contraintes sur P (6.13), (6.14) et (6.15).

$$\left\{ \begin{array}{ll}
 \frac{\partial P}{\partial t} - \alpha \frac{\partial^2 P}{\partial \theta^2} - \beta \Delta_x P + P \vec{u}(\theta) \cdot \nabla_x P = 0 & \text{sur } \Omega \times \mathcal{S}^1 \times [0, \infty[, \\
 P(x, \theta, t) = f_P(x, \theta, t) & \text{sur } \partial \Omega \times \mathcal{S}^1 \times [0, \infty[, \\
 P(x, \theta, 0) = P_0(x, \theta) & \text{sur } \Omega \times \mathcal{S}^1. \\
 \frac{\partial I}{\partial t} - \sigma_P \nu \Delta I + \vec{u}(\theta_P) \cdot \nabla I = 0 & \text{sur } \Omega \times [0, \infty[, \\
 I(x, t) = f_I(x, t) & \text{sur } \partial \Omega \times [0, \infty[, \\
 I(x, 0) = I_0(x) & \text{sur } \Omega. \\
 \int_{\mathcal{S}^1} P(\cdot, \theta, \cdot) d\theta = 1 & \text{sur } \Omega \times [0, \infty[, \\
 \vec{u}(\theta_P) = \frac{\nabla I^\perp}{\|\nabla I\|} & \text{sur } \Omega \times [0, \infty[, \\
 \sigma_P = \frac{\sigma}{\|\nabla I\|} & \text{sur } \Omega \times [0, \infty[.
 \end{array} \right. \quad (6.21)$$

Nous avons donc un système de deux EDPs d'évolution à deux inconnues I et P . Les contraintes sur P sont dues au fait que P est un champ de probabilité I -compatible et I dépend de P dans le terme de transport dans l'EDP sur I . La valeur initiale de I est n'importe quelle valeur utilisée pour prolonger I_0 dans Ω et la valeur initiale de P est soit (6.16) soit (6.17) en fonction de la non nullité du gradient de I . Les valeurs sont fixées et ne dépendent pas du temps.

Chapitre 7

Mise en œuvre numérique

Dans ce chapitre, nous présentons la résolution numérique du système d'équations (6.21) que nous venons de proposer au chapitre 6. Nous commencerons par présenter le schéma global de l'algorithme qui s'articule autour de deux schémas numériques. Ensuite nous détaillerons successivement deux schémas, en premier lieu le schéma concernant l'équation de désocclusion puis celui concernant la reconstruction. À la fin du chapitre nous présentons les résultats obtenus avec l'algorithme numérique complet.

7.1 Schéma de résolution

Le cœur du problème est la résolution d'un système composé de deux équations d'évolution sur P et sur I , l'équation (6.19)

$$\frac{\partial P}{\partial t} = \alpha \frac{\partial^2 P}{\partial \theta^2} + \beta \Delta_x P - P \vec{u}(\theta) \cdot \nabla_x P,$$

et l'équation (6.20)

$$\frac{\partial I}{\partial t} = \sigma_P \nu \Delta I - \vec{u}(\theta_P) \cdot \nabla I.$$

Nous avons deux inconnues, les fonctions I et P . La première équation, l'équation sur P ne contient aucune référence à I tandis que la deuxième équation, celle sur I fait intervenir P à travers deux coefficients qui dépendent de P , θ_P et σ_P . Pour résoudre numériquement ce système de deux EDPs à deux inconnues, nous procédons en alternant la résolution de ces deux équations. La première étape est la résolution de l'équation (6.19) en fonction de la variable P . La seconde étape est la résolution de l'équation (6.20) en fonction de I en laissant P constant.

L'enchaînement de ces étapes est présenté dans la figure 35. Le schéma général que nous utilisons pour résoudre le système (6.21) est le suivant :

ALGORITHME 35 (Schéma général) :

- 1 : Initialisation de I dans Ω
 - 2 : Initialisation de P
 - 3 : Évolution de l'équation (6.19) sur P
 - 4 : Calcul des coefficients θ_P et σ_P dépendant de P dans (6.20)
 - 5 : Évolution de l'équation (6.20) sur I
 - 6 : Renormalisation de P pour satisfaire les contraintes (6.13), (6.14) et (6.15) d' I -compatibilité
 - 7 : Retour à l'étape 3
-

Étapes 1 et 2. L'initialisation consiste à donner une valeur aux fonctions I et P dans le trou Ω . Pour initialiser dans Ω l'image I , nous utilisons deux possibilités, soit donner à I une valeur constante dans Ω , soit donner une valeur aléatoire pour chaque pixel de Ω . Ensuite nous initialisons P . Pour initialiser P dans Ω nous utilisons les résultats de la section 6.3. Le champ de probabilité P initial doit être I -compatible (cf. définition 34). Pour initialiser P nous utilisons donc la méthode simple décrite à la fin de la section 6.3 pour obtenir un champ I -compatible :

- une loi uniforme (6.17) si ∇I est nul,
- une loi normale (6.16) de moyenne l'orientation de ∇I^\perp et d'écart-type $\sigma/\|\nabla I\|$ si ∇I n'est pas nul.

Étape 3. Nous décrirons dans la section 7.2 un schéma numérique pour résoudre l'équation d'évolution sur P .

Étape 4. Avant de résoudre l'équation sur I nous devons connaître les moyennes θ_P et des écarts-types σ_P du champ de probabilités P . Nous les calculons à cette étape de la manière décrite dans le chapitre précédent, section 6.3.

Étape 5. Nous décrirons dans la section 7.3 un schéma numérique pour résoudre l'équation d'évolution sur I .

Étape 6. Enfin comme l'équation sur P ne dépend pas explicitement de I elle ne préserve pas l' I -compatibilité de P . Avant de retourner de l'étape 3 à l'étape 5, nous devons rendre à nouveau le champ de probabilités P I -compatible. C'est l'étape 6 qui commence par une renormalisation des valeurs de P pour que P soit un champ de probabilités. Ensuite nous effectuons

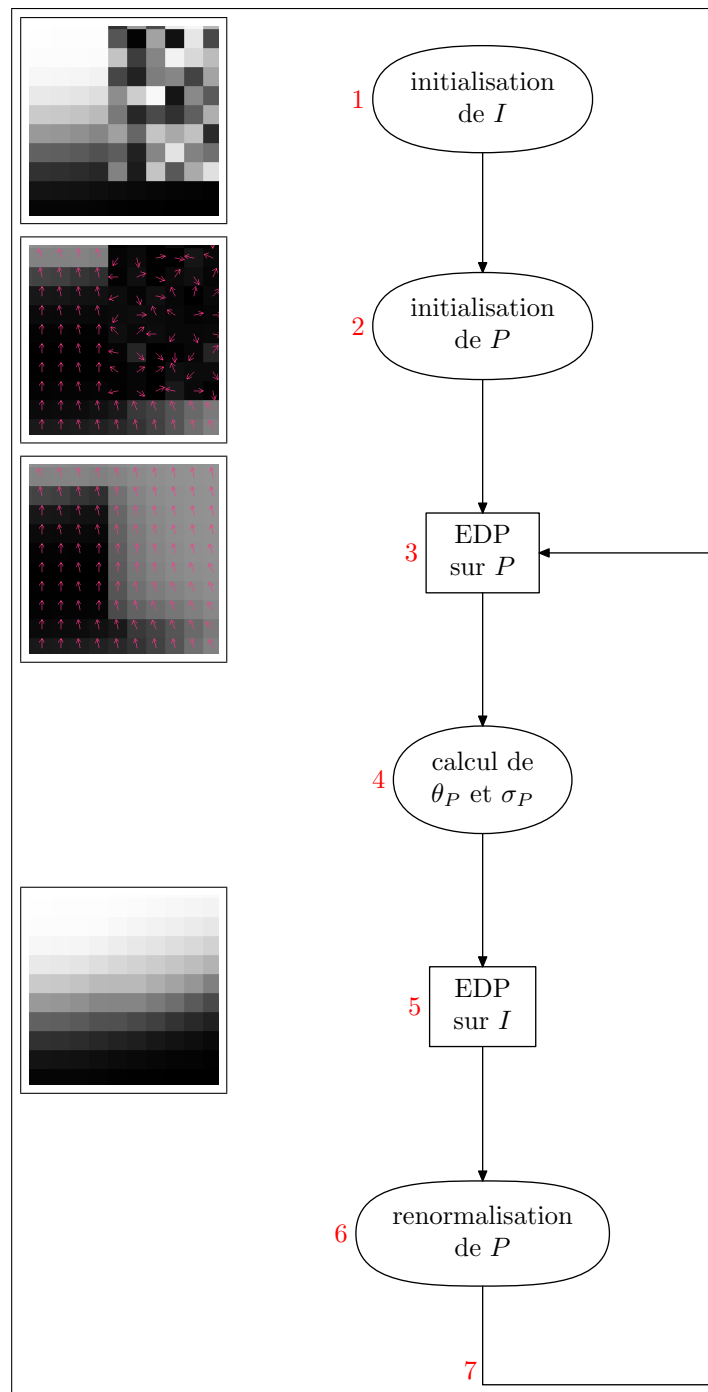


FIG. 7.1 – Les étapes de l’algorithme 35 qui résout un système de deux EDPs, l’une sur P l’autre sur I . Tout d’abord nous initialisons I puis P . À gauche de l’étape figure un détail sur une image, l’initialisation de I dans Ω est aléatoire, le trou est donc la partie supérieure droite de l’image avec des valeurs aléatoires. Ensuite nous appliquons le schéma sur P , nous calculons les coefficients θ_P et σ_P afin d’appliquer le schéma sur I . Ensuite nous renormalisons P pour rendre P I -compatible. Enfin nous alternons les schémas sur P et sur I .

en chaque point x de Ω une rotation des valeurs de $P(x, \cdot)$ sur \mathcal{S}^1 pour que l'orientation moyenne θ_P de P corresponde à l'orientation de ∇I^\perp . Ensuite nous effectuons une dilatation ou une contraction des valeurs de P pour que l'écart-type σ_P soit égal à $\sigma/\|\nabla I\|$.

Étape 7. La dernière étape consiste tout simplement à revenir à l'étape 3 de sorte que les deux schémas soient alternés.

7.2 Équation de désocclusion

Nous présentons dans cette section le schéma que nous utilisons pour résoudre l'équation (6.19)

$$\begin{cases} \frac{\partial P}{\partial t} - \alpha \frac{\partial^2 P}{\partial \theta^2} - \beta \Delta_x P + P \vec{u}(\theta) \cdot \nabla_x P = 0 & \text{sur } \Omega \times \mathcal{S}^1 \times [0, \infty[, \\ P(x, \theta, t) = f_P(x, \theta, t) & \text{sur } \partial \Omega \times \mathcal{S}^1 \times [0, \infty[, \\ P(x, \theta, 0) = P_0(x, \theta) & \text{sur } \Omega \times \mathcal{S}^1, \end{cases}$$

qui a pour but de compléter le champ de probabilités d'orientation P à l'intérieur du trou Ω . Après avoir décrit le schéma, nous étudierons le schéma proposé pour montrer qu'il est stable et consistant. Nous étudierons ensuite quelques résultats pour discuter de l'influence des différents termes de l'équation et du choix des paramètres.

7.2.1 Le schéma pour la désocclusion et sa stabilité

Schéma numérique. L'équation sur P est

$$\frac{\partial P}{\partial t} - \alpha \frac{\partial^2 P}{\partial \theta^2} - \beta \Delta_x P + P \vec{u}(\theta) \cdot \nabla_x P = 0.$$

Cette équation est équivalente à l'équation

$$\frac{\partial P}{\partial t} - \underbrace{\alpha \frac{\partial^2 P}{\partial \theta^2} - \beta \Delta_x P}_{\text{DIFFUSION}} + \underbrace{P \vec{u}(\theta) \cdot \nabla_x \frac{P^2}{2}}_{\text{TRANSPORT}} = 0. \quad (7.1)$$

Plutôt que de chercher un schéma numérique pour l'équation (6.19), nous allons chercher un schéma numérique pour l'équation (7.1). En effet dans cette dernière équation nous avons introduit le carré de P dans le gradient du terme de transport de sorte que P n'apparaît plus comme un produit dans ce terme.

Nous travaillons sur une grille régulière, les points qui nous intéressent sont les points de coordonnées $(x_{i,j}, \theta_k, t_n) = (\delta i, \delta j, 2k\pi/K, n\tau)$. Nous utilisons

donc les valeurs de P en ces points pour notre schéma. Nous posons $P_{i,j,k}^n = P(x_{i,j}, \theta_k, t_n)$. Nous posons également

$$\mathbf{k} = \frac{2\pi}{K},$$

la valeur du pas de discrétisation sur les directions.

Pour discrétiser l'équation (7.1) sur cette grille nous devons discrétiser chacun des termes, la dérivée par rapport au temps, les deux termes de diffusion et le terme de transport. Nous reprenons en partie le schéma présenté dans [17] pour l'équation de Burgers et son étude. En effet l'équation de Burgers en u s'écrit

$$u_t + uu_x = 0,$$

elle ressemble à l'équation que nous avons à résoudre au sens où le terme non linéaire est semblable au notre, il implique la dérivée de u^2 .

Comme nous utilisons un schéma itératif, la discrétisation de la dérivée par rapport au temps est donc un schéma aux différences finies décentrées avant.

$$\frac{\partial P}{\partial t} = \frac{P_{i,j,k}^{n+1} - P_{i,j,k}^n}{\tau} + o_{\tau \rightarrow 0}(1).$$

Ensuite nous utilisons la même discrétisation pour les autres termes, à savoir des schémas classiques aux différences finies centrées. Pour les termes de diffusion cela donne

$$\frac{\partial^2 P}{\partial \theta^2} = \frac{P_{i,j,k+1}^n + P_{i,j,k-1}^n - 2P_{i,j,k}^n}{\mathbf{k}^2} + o_{\mathbf{k} \rightarrow 0}(1) \quad (7.2)$$

et

$$\Delta_x P = \frac{P_{i+1,j,k}^n + P_{i-1,j,k}^n - 2P_{i,j,k}^n}{\delta^2} + \frac{P_{i,j+1,k}^n + P_{i,j-1,k}^n - 2P_{i,j,k}^n}{\delta^2} + o_{\delta \rightarrow 0}(1). \quad (7.3)$$

Enfin pour le terme non linéaire de transport, nous posons $\vec{u}_k = \vec{u}(\theta_k)$ et nous notons (u_k, v_k) les coordonnées du vecteur \vec{u}_k . La discrétisation de l'opérateur par un schéma aux différences finies centrées donne dans ce cas

$$\begin{aligned} \vec{u}(\theta) \cdot \nabla_x \frac{P^2}{2} &= u_k \frac{\frac{(P_{i+1,j,k}^n)^2}{2} - \frac{(P_{i-1,j,k}^n)^2}{2}}{2\delta} \\ &+ v_k \frac{\frac{(P_{i,j+1,k}^n)^2}{2} - \frac{(P_{i,j-1,k}^n)^2}{2}}{2\delta} + o_{\delta \rightarrow 0}(1). \end{aligned}$$

Ce qui s'écrit plus simplement

$$\begin{aligned} \vec{u}(\theta) \cdot \nabla_x \frac{P^2}{2} &= \frac{u_k}{4\delta} \left((P_{i+1,j,k}^n)^2 - (P_{i-1,j,k}^n)^2 \right) \\ &+ \frac{v_k}{4\delta} \left((P_{i,j+1,k}^n)^2 - (P_{i,j-1,k}^n)^2 \right) + o_{\delta \rightarrow 0}(1). \end{aligned} \quad (7.4)$$

Le schéma itératif complet est donc

$$\begin{aligned}
 P_{i,j,k}^{n+1} = & P_{i,j,k}^n + \frac{\alpha\tau}{\mathbf{k}^2} (P_{i,j,k+1}^n + P_{i,j,k-1}^n - 2P_{i,j,k}^n) \\
 & + \frac{\beta\tau}{\delta^2} (P_{i+1,j,k}^n + P_{i-1,j,k}^n + P_{i,j+1,k}^n + P_{i,j-1,k}^n - 4P_{i,j,k}^n) \\
 & - \frac{u_k\tau}{4\delta} \left((P_{i+1,j,k}^n)^2 - (P_{i-1,j,k}^n)^2 \right) \\
 & - \frac{v_k\tau}{4\delta} \left((P_{i,j+1,k}^n)^2 - (P_{i,j-1,k}^n)^2 \right)
 \end{aligned} \tag{7.5}$$

Pour résoudre l'équation (6.19) nous appliquons donc le schéma suivant.

ALGORITHME 36 (Schéma numérique pour la désocclusion du champ de probabilités P) :

Initialisation de P à partir de I

Pour n de 0 à $N - 1$ **faire**

Pour tout (i, j) dans le trou Ω **faire**

Pour k de 0 à K **faire**

$$\begin{aligned}
 P_{i,j,k}^{n+1} \leftarrow & P_{i,j,k}^n \\
 & + (\alpha\tau/\mathbf{k}^2) \times (P_{i,j,k+1}^n + P_{i,j,k-1}^n - 2P_{i,j,k}^n) \\
 & + (\beta\tau/\delta^2) \times (P_{i+1,j,k}^n + P_{i-1,j,k}^n \\
 & \quad + P_{i,j+1,k}^n + P_{i,j-1,k}^n - 4P_{i,j,k}^n) \\
 & - (u_k\tau/(4\delta)) \left((P_{i+1,j,k}^n)^2 - (P_{i-1,j,k}^n)^2 \right) \\
 & - (v_k\tau/(4\delta)) \left((P_{i,j+1,k}^n)^2 - (P_{i,j-1,k}^n)^2 \right)
 \end{aligned}$$

Stabilité. Le schéma de l'algorithme 36 est non linéaire à cause du terme de transport. En effet ce dernier est de la forme ∇P^2 . Nous ne pouvons donc pas étudier directement le schéma de l'algorithme 36. Nous devons d'abord le linéariser, c'est-à-dire étudier le schéma linéaire le plus proche du schéma non linéaire. Pour ce faire nous utilisons une identité remarquable pour transformer les différences de carrés en produits. Par exemple sur un des termes cela donne

$$(P_{i+1,j,k}^n)^2 - (P_{i-1,j,k}^n)^2 = 2(P_{i+1,j,k}^n - P_{i-1,j,k}^n) \times \frac{P_{i+1,j,k}^n + P_{i-1,j,k}^n}{2}.$$

Or $(P_{i+1,j,k}^n + P_{i-1,j,k}^n)/2$ est la moyenne entre les deux valeurs $P_{i+1,j,k}^n$ et $P_{i-1,j,k}^n$. Nous supposons que les valeurs de P fluctuent autour d'une valeur moyenne \bar{P} . Nous écrivons alors $P = \bar{P} + u$ où \bar{P} est une valeur constante et u une fonction

prenant des valeurs proches de 0. Nous pouvons alors écrire

$$\begin{aligned}
 & 2(P_{i+1,j,k}^n - P_{i-1,j,k}^n) \times (P_{i+1,j,k}^n + P_{i-1,j,k}^n)/2 \\
 &= 2(\bar{P} + u_{i+1,j,k}^n - \bar{P} - u_{i-1,j,k}^n) \times \frac{\bar{P} + u_{i+1,j,k}^n + \bar{P} + u_{i-1,j,k}^n}{2} \\
 &= 2(u_{i+1,j,k}^n - u_{i-1,j,k}^n) \times \frac{2\bar{P} + u_{i+1,j,k}^n + u_{i-1,j,k}^n}{2} \\
 &= 2\bar{P}(u_{i+1,j,k}^n - u_{i-1,j,k}^n) + (u_{i+1,j,k}^n)^2 - (u_{i-1,j,k}^n)^2. \\
 &= 2\bar{P}(u_{i+1,j,k}^n - u_{i-1,j,k}^n) + o_{u \rightarrow 0}(u) \\
 &= 2\bar{P}(P_{i+1,j,k}^n - P_{i-1,j,k}^n) + o_{P \rightarrow \bar{P}}(P - \bar{P}).
 \end{aligned}$$

Le schéma linéarisé est donc

$$\begin{aligned}
 P_{i,j,k}^{n+1} &= P_{i,j,k}^n + \frac{\alpha\tau}{\mathbf{k}^2} (P_{i,j,k+1}^n + P_{i,j,k-1}^n - 2P_{i,j,k}^n) \\
 &+ \frac{\beta\tau}{\delta^2} (P_{i+1,j,k}^n + P_{i-1,j,k}^n + P_{i,j+1,k}^n + P_{i,j-1,k}^n - 4P_{i,j,k}^n) \\
 &- \frac{u_k\tau\bar{P}}{2\delta} (P_{i+1,j,k}^n - P_{i-1,j,k}^n) - \frac{v_k\tau\bar{P}}{2\delta} (P_{i,j+1,k}^n - P_{i,j-1,k}^n).
 \end{aligned} \tag{7.6}$$

C'est la stabilité de ce schéma que nous étudions. Ce schéma est un schéma linéaire. Un schéma linéaire est stable si tous les coefficients qui sont devant les valeurs qui dépendent de P sont compris entre 0 et 1. En effet, il est tout d'abord aisé de vérifier que la somme de tous ces coefficients est 1. Donc si tous les coefficients sont compris entre 0 et 1 la nouvelle valeur $P_{i,j,k}^{n+1}$ de P est une moyenne pondérée de différentes valeurs de P à l'instant précédent. Dans ce cas les valeurs de P restent comprises entre la valeur minimale et la valeur maximale de P à l'instant initial.

PROPOSITION 37 :

Les coefficients du schéma (7.6) sont tous compris dans l'intervalle $[0; 1]$ si

$$\beta \geq \frac{\bar{P}\delta}{2} \tag{7.7}$$

et

$$\tau \leq \frac{\delta}{\frac{2\alpha\delta}{\mathbf{k}^2} + \frac{4\beta}{\delta} + \frac{\bar{P}}{2}}. \tag{7.8}$$

Montrons que les conditions de la proposition sont suffisantes pour contrôler les coefficients du schéma.

Preuve : Nous commençons par appeler $c_{i,j,k}$ le coefficient devant le terme $P_{i,j,k}^n$ dans le schéma (7.6) et de même nous appelons $c_{i\pm 1,j,k}$, $c_{i,j\pm 1,k}$ et $c_{i,j,k\pm 1}$ les

autres coefficients, de sorte que

$$\begin{aligned}
 c_{i,j,k} &= 1 - \frac{2\alpha\tau}{\mathbf{k}^2} - \frac{4\beta\tau}{\delta^2}, \\
 c_{i+1,j,k} &= \frac{\beta\tau}{\delta^2} - \frac{u_k\tau\bar{P}}{2\delta}, \\
 c_{i-1,j,k} &= \frac{\beta\tau}{\delta^2} + \frac{u_k\tau\bar{P}}{2\delta}, \\
 c_{i,j+1,k} &= \frac{\beta\tau_P}{\delta^2} - \frac{v_k\tau\bar{P}}{2\delta}, \\
 c_{i,j-1,k} &= \frac{\beta\tau_P}{\delta^2} + \frac{v_k\tau\bar{P}}{2\delta}, \\
 c_{i,j,k+1} &= \frac{\alpha\tau}{\mathbf{k}^2}, \\
 c_{i,j,k-1} &= \frac{\alpha\tau}{\mathbf{k}^2}.
 \end{aligned}$$

Il est immédiat que $c_{i,j,k}$ est majoré par 1. De même les deux coefficients $c_{i,j,k\pm 1}$ sont positifs.

Maintenant si

$$\tau \leq \frac{\delta}{\frac{2\alpha\delta}{\mathbf{k}^2} + \frac{4\beta}{\delta} + \frac{\bar{P}}{2}},$$

alors

$$\frac{2\alpha\tau}{\mathbf{k}^2} + \frac{4\beta\tau}{\delta^2} + \frac{\bar{P}\tau}{2\delta} \leq 1.$$

Avec cette inégalité il est évident que $c_{i,j,k}$ est positif et que les coefficients $c_{i\pm 1,j,k}$, $c_{i,j\pm 1,k}$ et $c_{i,j,k\pm 1}$ sont plus petits que un. En effet $(u_k, v_k) = \vec{u}(\theta_k)$ est un vecteur unitaire donc u_k et v_k sont plus petits que 1.

Et si

$$\beta \geq \frac{\bar{P}\delta}{2},$$

alors

$$\frac{\beta\tau}{\delta^2} - \frac{\bar{P}\tau}{2\delta} \geq 0,$$

ce qui entraîne la positivité des coefficients $c_{i\pm 1,j,k}$ et $c_{i,j\pm 1,k}$.

Au final si les deux inégalités (7.7) et (7.8) sont vérifiées, tous les coefficients sont positifs et majorés par 1. \square

Ceci nous permet de montrer que le schéma (7.6) est stable.

PROPOSITION 38 :

Si les inégalités

$$\beta \geq \frac{\bar{P}\delta}{2} \tag{7.9}$$

et

$$\tau \leq \frac{\delta}{\frac{2\alpha\delta}{\mathbf{k}^2} + \frac{4\beta}{\delta} + \frac{\bar{P}}{2}}. \quad (7.10)$$

sont vérifiées alors le schéma (7.6) est stable.

Preuve : Dans le schéma que nous avons choisi la somme des coefficients vaut 1 donc si en plus les coefficients sont compris entre 0 et 1 le schéma est stable. En effet dans ce cas le schéma consiste simplement à faire une moyenne pondérée entre différentes valeurs de P . \square

Consistance. L'algorithme 36 est bien consistant avec l'équation 6.19.

PROPOSITION 39 :

Soit P une fonction \mathcal{C}^2 sur $\mathbb{R}^2 \times \mathcal{S}^1 \times \mathbb{R}$ telle que P est solution de l'équation (6.19) en un point (x, θ, t) de $\mathbb{R}^2 \times \mathcal{S}^1 \times \mathbb{R}$. Si l'inégalité (7.8) est vérifiée alors sous ces hypothèses nous avons

$$\begin{aligned} & P_{i,j,k}^{n+1} - P_{i,j,k}^n - \frac{\alpha\tau}{\mathbf{k}^2} (P_{i,j,k+1}^n + P_{i,j,k-1}^n - 2P_{i,j,k}^n) \\ & \quad - \frac{\beta\tau}{\delta^2} (P_{i+1,j,k}^n + P_{i-1,j,k}^n + P_{i,j+1,k}^n + P_{i,j-1,k}^n - 4P_{i,j,k}^n) \\ & \quad + \frac{u_k\tau}{4\delta} \left((P_{i+1,j,k}^n)^2 - (P_{i-1,j,k}^n)^2 \right) + \frac{v_k\tau}{4\delta} \left((P_{i,j+1,k}^n)^2 - (P_{i,j-1,k}^n)^2 \right) \\ & \quad = o_{\tau \rightarrow 0}(\tau) + o_{\delta \rightarrow 0}(\delta^2) + o_{\mathbf{k} \rightarrow 0}(\mathbf{k}^2). \end{aligned}$$

Preuve : Il suffit d'utiliser les formules de Taylor pour P et $P^2/2$ et nous obtenons

$$\begin{aligned} & P_{i,j,k}^{n+1} - P_{i,j,k}^n - \frac{\alpha\tau}{\mathbf{k}^2} (P_{i,j,k+1}^n + P_{i,j,k-1}^n - 2P_{i,j,k}^n) \\ & \quad - \frac{\beta\tau}{\delta^2} (P_{i+1,j,k}^n + P_{i-1,j,k}^n + P_{i,j+1,k}^n + P_{i,j-1,k}^n - 4P_{i,j,k}^n) \\ & \quad + \frac{u_k\tau}{4\delta} \left((P_{i+1,j,k}^n)^2 - (P_{i-1,j,k}^n)^2 \right) + \frac{v_k\tau}{4\delta} \left((P_{i,j+1,k}^n)^2 - (P_{i,j-1,k}^n)^2 \right) \\ & = \tau \frac{\partial P}{\partial t} - \alpha\tau \frac{\partial^2 P}{\partial \theta^2} - \beta\tau \Delta_x P + \vec{u}(\theta) \cdot \nabla_x \frac{P^2}{2} \\ & \quad + o_{\tau \rightarrow 0}(\tau) + \tau o_{\delta \rightarrow 0}(1) + \tau o_{\mathbf{k} \rightarrow 0}(1). \end{aligned}$$

Comme P est solution de (6.19) nous obtenons donc

$$\begin{aligned} & P_{i,j,k}^{n+1} - P_{i,j,k}^n - \frac{\alpha\tau}{\mathbf{k}^2} (P_{i,j,k+1}^n + P_{i,j,k-1}^n - 2P_{i,j,k}^n) \\ & \quad - \frac{\beta\tau}{\delta^2} (P_{i+1,j,k}^n + P_{i-1,j,k}^n + P_{i,j+1,k}^n + P_{i,j-1,k}^n - 4P_{i,j,k}^n) \\ & \quad + \frac{u_k\tau}{4\delta} \left((P_{i+1,j,k}^n)^2 - (P_{i-1,j,k}^n)^2 \right) + \frac{v_k\tau}{4\delta} \left((P_{i,j+1,k}^n)^2 - (P_{i,j-1,k}^n)^2 \right) \\ & \quad = o_{\tau \rightarrow 0}(\tau) + \tau o_{\delta \rightarrow 0}(1) + \tau o_{\mathbf{k} \rightarrow 0}(1). \end{aligned}$$

Pour conclure nous utilisons le fait que l'inégalité (7.8) est vérifiée. En effet à partir de cette inégalité nous obtenons les deux inégalités

$$\tau \leq \frac{\delta^2}{4\beta} \text{ et } \tau \leq \frac{\mathbf{k}^2}{2\alpha}.$$

Ces deux inégalités permettent d'avoir

$$\tau o_{\delta \rightarrow 0}(1) = o_{\delta \rightarrow 0}(\delta^2) \text{ et } \tau o_{\mathbf{k} \rightarrow 0}(1) = o_{\mathbf{k} \rightarrow 0}(\mathbf{k}^2).$$

Ces deux égalités permettent de conclure facilement. \square

L'algorithme 36 est donc stable et consistant si l'inégalité (7.8) est vérifiée.

7.2.2 Mise en œuvre de la désocclusion du champ de probabilités

Pour mettre en œuvre l'algorithme 36 nous avons juste à choisir les paramètres. L'initialisation se fait automatiquement à partir des valeurs de I , il reste donc tout de même les paramètres. L'algorithme 36 possède 5 paramètres.

- Les deux paramètres de l'équation : α le coefficient de diffusion angulaire et β le coefficient de diffusion spatial.
- Les paramètres de discrétisation : δ le pas de discrétisation spatial, \mathbf{k} le pas de discrétisation angulaire et τ le pas de temps.

Les deux paramètres de l'équation α et β sont des coefficients de diffusion. Ils permettent de régler l'influence des termes de diffusion à la fois entre les deux termes de diffusion mais aussi entre ces termes et le terme de transport.

Le pas spatial δ est en fait égal à 1, c'est la distance entre deux pixels. Le pas angulaire \mathbf{k} dépend en fait de K , le nombre d'angles discrets sur \mathcal{S}^1 . En pratique nous prenons $K = 32$. Enfin le dernier pas de discrétisation, le pas de temps τ influe sur la rapidité de l'algorithme. Plus le pas de temps est grand, plus la convergence est rapide, ceci évidemment tout en respectant l'inégalité (7.8). Comme nous voulons évidemment que l'algorithme soit rapide, nous prenons le cas extrême, le plus grand pas de temps possible

$$\tau = \frac{\delta}{2\alpha\delta/\mathbf{k}^2 + 4\beta/\delta + \bar{P}/2}.$$

Pour en déduire τ nous devons donc connaître \bar{P} . La valeur \bar{P} est censée être une valeur autour de laquelle P oscille. Nous avons donc pour tout n

$$\min\{P_{i,j,k}^n | i, j, k\} \leq \bar{P} \leq \max\{P_{i,j,k}^n | i, j, k\}$$

Ce qui nous donne aussi

$$\bar{P} \leq \|P\|_{\infty},$$

et donc

$$\frac{\delta}{2\alpha\delta/\mathbf{k}^2 + 4\beta/\delta + \|P\|_\infty/2} \leq \frac{\delta}{2\alpha\delta/\mathbf{k}^2 + 4\beta/\delta + \bar{P}/2}$$

Pour que l'inégalité (7.8) soit satisfaite il suffit donc de prendre

$$\tau = \frac{\delta}{2\alpha\delta/\mathbf{k}^2 + 4\beta/\delta + \|P\|_\infty/2}.$$

Les seuls paramètres que nous avons encore à choisir sont les paramètres α et β . Ces paramètres contrôlent le réglage de la diffusion par rapport au transport. Mais en fait pour β , comme l'inégalité (7.7) doit aussi être satisfaite, nous pourrions prendre le cas d'égalité et ne plus avoir à nous occuper du paramètre β . Cependant même si le schéma est stable pour certaines valeurs de β trop petites certaines aberrations apparaissent. La figure 7.2 présente deux images synthétiques pour tester l'algorithme ainsi que le masque utilisé. Le résultat de l'algorithme sur ces deux images avec $\alpha = 1$ et $\beta = 1$ est présenté dans la figure 7.3, ce sont les images 7.3(a) et 7.3(b). Les vecteurs représentent l'orientation moyenne du champ de probabilité P obtenu en chaque point $\vec{u}(\theta_P)$. Nous observons que des *tourbillons* apparaissent dans le champ de vecteurs ce qui n'est pas désirable pour notre objectif.

L'origine de ce phénomène est le sens des lignes de niveau. En effet les lignes de niveau sont orientées et ceci a une incidence sur le terme de transport. Pour pallier à ce problème nous prenons une valeur de β un peu plus élevée. La figure 7.3 montre les résultats obtenus avec $\beta = 2$, ce sont les images 7.3(c) et 7.3(d). Avec cette valeur de β il n'y a plus de *tourbillons*. Nous constatons alors que le champ de vecteurs est visuellement dans la direction attendue mais qu'il y a de la diffusion, le champ de vecteurs est non nul sur une zone un peu plus large que la zone souhaitée.

La figure suivante, la figure 7.4 montre le résultat obtenu en partant de l'image 7.4(a) avec $\alpha = 2$ et $\beta = 4$. Les vecteurs représentent là aussi l'orientation moyenne du champ de probabilités. Cette fois l'intensité de l'image sous les vecteurs indique les valeurs des écarts-types des probabilités. Plus le fond est sombre, plus l'écart-type est proche de 0.

7.3 Équation de reconstruction

Nous présentons maintenant le schéma que nous utilisons pour résoudre l'équation (6.20) sur I .

$$\begin{cases} \frac{\partial I}{\partial t} - \sigma_P \nu \Delta I + \vec{u}(\theta_P) \cdot \nabla I = 0 & \text{sur } \Omega \times [0, \infty[, \\ I(x, t) = f_I(x, t) & \text{sur } \partial\Omega \times [0, \infty[, \\ I(x, 0) = I_0(x) & \text{sur } \Omega. \end{cases}$$

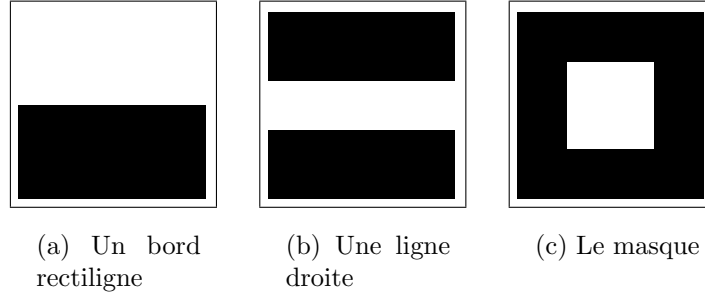


FIG. 7.2 – Nous utilisons les deux premières images pour faire tester l'équation de désocclusion (6.19) avec l'algorithme 36. La première image est l'image d'un bord droit. La deuxième image est l'image d'une ligne. La dernière image définit le trou Ω à l'intérieur duquel l'image doit être reconstruite.

L'objectif de cette équation est de reconstruire l'image dans le trou Ω en utilisant un champ de probabilité d'orientation des lignes de niveau P . Nous étudions le schéma proposé pour montrer qu'il est stable et consistant. Ensuite nous effectuons des tests sur quelques images simples pour discuter de la pertinence du schéma et pour contrôler l'influence des paramètres.

7.3.1 Le schéma de la reconstruction et sa stabilité

Schéma numérique. L'équation sur I est

$$\frac{\partial I}{\partial t} - \underbrace{\sigma_P \nu \Delta I}_{\text{DIFFUSION}} + \underbrace{\vec{u}(\theta_P) \cdot \nabla I}_{\text{TRANSPORT}} = 0.$$

Nous prenons les mêmes schémas de discrétisation que pour celui pour P (pour la dérivée en temps et pour le terme de diffusion) à savoir

$$\frac{\partial I}{\partial t} = \frac{I_{i,j}^{n+1} - I_{i,j}^n}{\tau} + o_{\tau \rightarrow 0}(1), \quad (7.11)$$

$$\Delta I = \frac{I_{i+1,j}^n + I_{i-1,j}^n - 2I_{i,j}^n}{\delta^2} + \frac{I_{i,j+1}^n + I_{i,j-1}^n - 2I_{i,j}^n}{\delta^2} + o_{\delta \rightarrow 0}(1). \quad (7.12)$$

Il ne reste que le terme de transport à discrétiser. Posons $\vec{u}(\theta_P) = (u_P, v_P)$ pour les coordonnées du vecteur du terme de transport. Pour ce terme nous allons utiliser un schéma aux différences finies décentrées avant ou arrière en fonction des signes de u_P et v_P . Le décentrage est choisi de sorte que le coefficient devant $I_{i,j}^n$ soit négatif. Cela donne donc

$$\vec{u}(\theta_P) \cdot \nabla I = |u_P| \frac{I_{i+\varepsilon_1,j}^n - I_{i,j}^n}{\delta} + |v_P| \frac{I_{i,j+\varepsilon_2}^n - I_{i,j}^n}{\delta} + o_{\delta \rightarrow 0}(1)$$

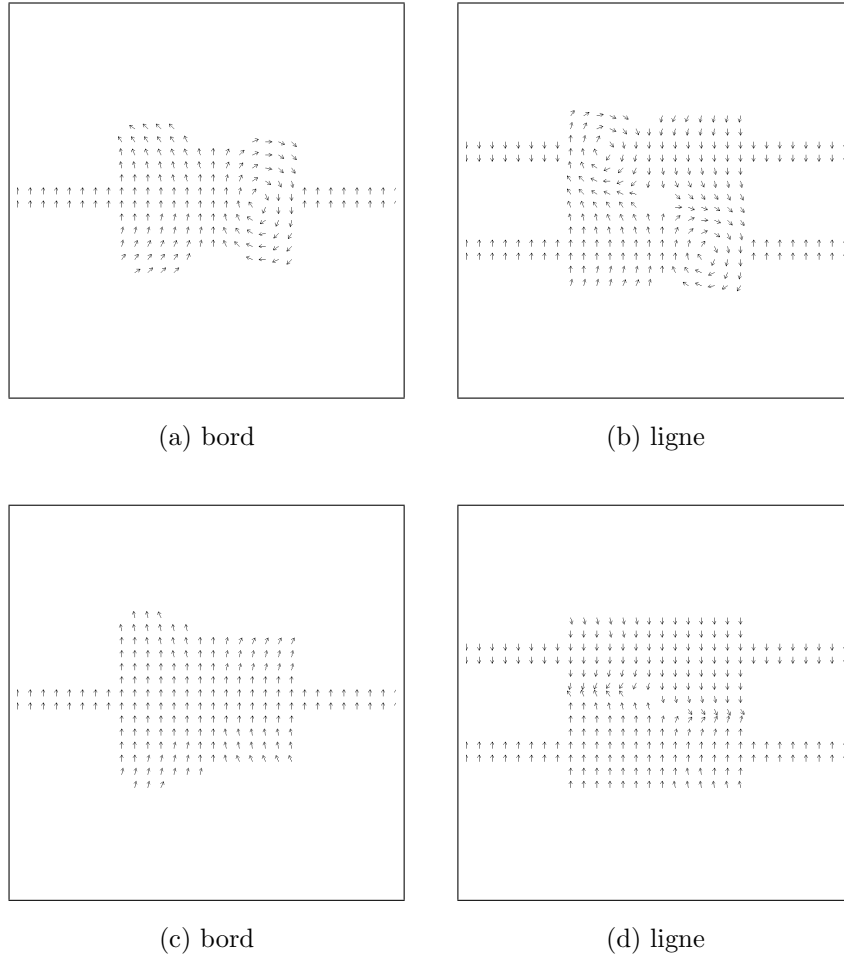
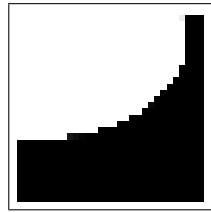
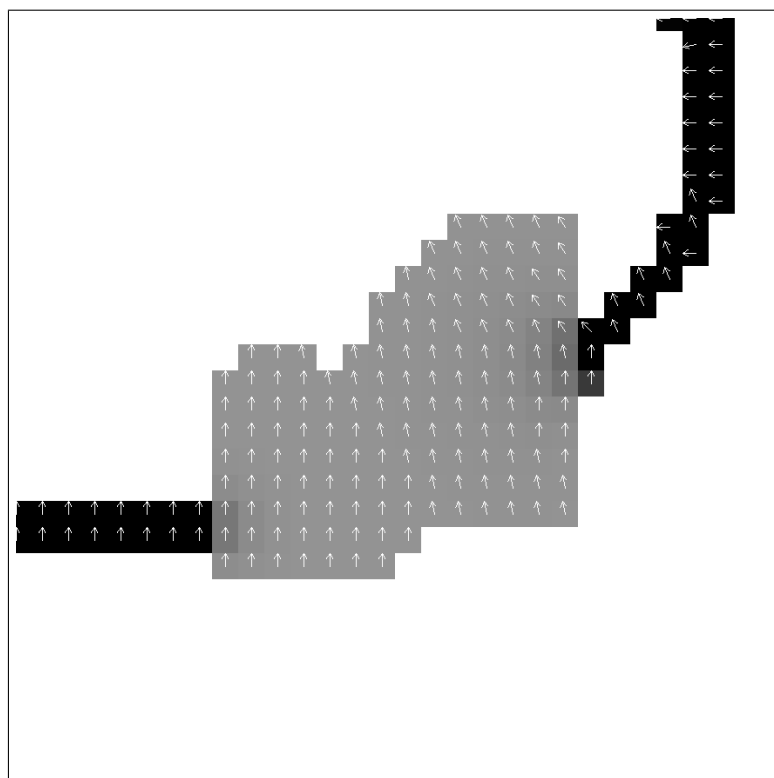


FIG. 7.3 – Résultats de l’algorithme 36 sur les images de la figure 7.2. Les résultats montrés sont des vecteurs. Chaque vecteur indique l’orientation moyenne du champ de probabilités P en ce point, c’est le vecteur $\vec{u}(\theta_P)$. Pour les images 7.3(a) et 7.3(b) les paramètres utilisés sont $\alpha = 1$ et $\beta = 1$ alors que pour 7.3(c) et 7.3(d) nous avons pris $\beta = 2$. Sur les deux premières images, le schéma est stable mais le fait que les lignes de niveau ont une orientation, un sens, génère les tourbillons visibles sur les images 7.3(a) et 7.3(b). En prenant β plus grand, ce phénomène disparaît, images 7.3(c) et 7.3(d).



(a) Un bord recourbé



(b) Résultat, moyenne et écart-type

FIG. 7.4 – L'image 7.4(a) est une autre image de test pour l'algorithme 36. L'image 7.4(b) quant à elle est le résultat obtenu sur ce bord recourbé avec $\alpha = 2$ et $\beta = 4$. Les vecteurs représentent les orientations moyennes. Par contre l'intensité de l'image sous les vecteurs correspond à l'écart-type des probabilités, plus l'image est sombre, plus l'écart-type est proche de 0.

avec

$$\varepsilon_1 = \begin{cases} +1 & \text{si } u_P \geq 0 \\ -1 & \text{si } u_P < 0 \end{cases} \quad \text{et} \quad \varepsilon_2 = \begin{cases} +1 & \text{si } v_P \geq 0 \\ -1 & \text{si } v_P < 0 \end{cases}$$

Le schéma que nous utilisons est donc

$$I_{i,j}^{n+1} = I_{i,j}^n + \frac{\sigma_P \nu \tau}{\delta^2} (I_{i+1,j}^n + I_{i-1,j}^n + I_{i,j+1}^n + I_{i,j-1}^n - 4I_{i,j}^n) \\ + \frac{|u_P| \tau}{\delta} (I_{i+\varepsilon_1,j}^n - I_{i,j}^n) + \frac{|v_P| \tau}{\delta} (I_{i,j+\varepsilon_2}^n - I_{i,j}^n) \quad (7.13)$$

Pour résoudre l'équation (6.20) nous appliquons donc le schéma suivant.

ALGORITHME 40 (Schéma numérique de reconstruction) :

Initialisation de I à l'intérieur du trou Ω par une valeur constante ou des valeurs aléatoires

Pour n de 0 à $N - 1$ **faire**

Pour tout (i, j) dans le trou Ω **faire**

$$I_{i,j}^{n+1} \leftarrow I_{i,j}^n + (\sigma_P \nu \tau / \delta^2) \times (I_{i+1,j}^n + I_{i-1,j}^n \\ + I_{i,j+1}^n + I_{i,j-1}^n - 4I_{i,j}^n) \\ + (|u_P| \tau / \delta) \times (I_{i+\varepsilon_1,j}^n - I_{i,j}^n) \\ + (|v_P| \tau / \delta) \times (I_{i,j+\varepsilon_2}^n - I_{i,j}^n)$$

Stabilité. Pour que le schéma soit stable il faut que ses coefficients soient tous compris entre 0 et 1. Et comme la somme de tous les coefficient vaut 1 cette condition est suffisante.

PROPOSITION 41 :

Si l'inégalité

$$\tau \leq \frac{\delta}{4\pi\nu/\delta + \sqrt{2}}. \quad (7.14)$$

est vérifiée alors les coefficients dans le schéma de l'algorithme (40) sont tous compris dans $[0; 1]$.

Preuve : Comme précédemment nous notons $c_{i,j}$, $c_{i\pm 1,j}$ et $c_{i,j\pm 1}$ les coefficients

devant $I_{i,j}^n$, $I_{i\pm 1,j}^n$ et $I_{i,j\pm 1}^n$. Dans le cas où u_P et v_P sont positifs cela donne

$$\begin{aligned} c_{i,j} &= 1 - \frac{4\sigma_P\nu\tau}{\delta^2} - \frac{|u_P|\tau}{\delta} - \frac{|v_P|\tau}{\delta} \\ c_{i+1,j} &= \frac{\sigma_P\nu\tau}{\delta^2} + \frac{|u_P|\tau}{\delta} \\ c_{i-1,j} &= \frac{\sigma_P\nu\tau}{\delta^2} \\ c_{i,j+1} &= \frac{\sigma_P\nu\tau}{\delta^2} + \frac{|v_P|\tau}{\delta} \\ c_{i,j-1} &= \frac{\sigma_P\nu\tau}{\delta^2} \end{aligned}$$

Si u_P est négatif les coefficients $c_{i+1,j}$ et $c_{i-1,j}$ sont intervertis et de même pour les coefficients $c_{i,j+1}$ et $c_{i,j-1}$ dans le cas où v_P est négatif.

Le coefficient $c_{i,j}$ est toujours plus petit que 1. De même les quatre autres coefficients sont toujours positifs. Supposons désormais que l'inégalité (7.14) est vérifiée

$$\tau \leq \frac{\delta}{4\pi\nu/\delta + \sqrt{2}}.$$

Dans ce cas

$$\frac{4\pi\nu\tau}{\delta^2} + \frac{\sqrt{2}\tau}{\delta} \leq 1.$$

Avec cette inégalité il est évident que les quatre coefficients $c_{i\pm 1,j}$ et $c_{i,j\pm 1}$ sont plus petits que 1. En effet d'une part la probabilité P est une probabilité sur le cercle \mathcal{S}^1 et l'écart-type σ_P est donc plus petit qu'un demi-cercle, $\sigma_P \leq \pi$, lorsque P est une probabilité uniforme, nous remplaçons alors σ_P par π . D'autre part les réels u_P et v_P sont les coordonnées d'un vecteur unitaire leur valeur absolue est plus petite que 1 et donc a fortiori plus petite que $\sqrt{2}$. De plus nous avons aussi l'inégalité

$$|u_P| + |v_P| \leq \sqrt{2}.$$

Avec cette dernière inégalité il apparaît facilement que $c_{i,j}$ est positif. Tous les coefficients sont donc bien compris entre 0 et 1 si l'inégalité (7.14) est vérifiée. \square

PROPOSITION 42 :

Pour que l'algorithme 40 soit stable il suffit que l'inégalité

$$\tau \leq \frac{\delta}{4\pi\nu/\delta + \sqrt{2}}. \quad (7.15)$$

soit vérifiée.

Preuve : Nous venons de montrer que si l'inégalité (7.14) est vérifiée alors tous les coefficients sont compris entre 0 et 1. Ceci suffit pour que le schéma soit alors stable. \square

Consistance. L'algorithme 40 permet effectivement de résoudre l'équation (6.20), nous l'avons choisi pour cela.

PROPOSITION 43 :

Soit I une fonction \mathcal{C}^2 sur $\mathbb{R}^2 \times \mathbb{R}$ telle que I soit solution de l'équation (6.20) en un point (x, t) de $\Omega \times \mathbb{R}$. Si l'inégalité (7.14) est vérifiée, alors dans ce cas I vérifie

$$\begin{aligned} I_{i,j}^{n+1} - I_{i,j}^n - \frac{\sigma_P \nu \tau}{\delta^2} (I_{i+1,j}^n + I_{i-1,j}^n + I_{i,j+1}^n + I_{i,j-1}^n - 4I_{i,j}^n) \\ - \frac{|u_P| \tau}{\delta} (I_{i+\varepsilon_1,j}^n - I_{i,j}^n) - \frac{|v_P| \tau}{\delta} (I_{i,j+\varepsilon_2}^n - I_{i,j}^n) = o_{\tau \rightarrow 0}(\tau) + o_{\delta \rightarrow 0}(\delta), \end{aligned}$$

au point (x, t) .

Preuve : Comme pour l'équation sur P il suffit d'utiliser le développement de Taylor de I pour obtenir

$$\begin{aligned} I_{i,j}^{n+1} - I_{i,j}^n - \frac{\sigma_P \nu \tau}{\delta^2} (I_{i+1,j}^n + I_{i-1,j}^n + I_{i,j+1}^n + I_{i,j-1}^n - 4I_{i,j}^n) \\ - \frac{|u_P| \tau}{\delta} (I_{i+\varepsilon_1,j}^n - I_{i,j}^n) - \frac{|v_P| \tau}{\delta} (I_{i,j+\varepsilon_2}^n - I_{i,j}^n) \\ = \tau \frac{\partial I}{\partial t} - \sigma_P \nu \tau \Delta I - \tau \vec{u}_P \cdot \nabla I + o_{\tau \rightarrow 0}(\tau) + \tau o_{\delta \rightarrow 0}(1), \end{aligned}$$

et comme I est solution de l'équation (6.20) nous obtenons

$$\begin{aligned} I_{i,j}^{n+1} - I_{i,j}^n - \frac{\sigma_P \nu \tau}{\delta^2} (I_{i+1,j}^n + I_{i-1,j}^n + I_{i,j+1}^n + I_{i,j-1}^n - 4I_{i,j}^n) \\ - \frac{|u_P| \tau}{\delta} (I_{i+\varepsilon_1,j}^n - I_{i,j}^n) - \frac{|v_P| \tau}{\delta} (I_{i,j+\varepsilon_2}^n - I_{i,j}^n) = o_{\tau \rightarrow 0}(\tau) + \tau o_{\delta \rightarrow 0}(1), \end{aligned}$$

qui est presque ce que nous voulons. Il reste à utiliser l'inégalité (7.14) pour dire qu'alors τ est majoré par $\delta/\sqrt{2}$ et donc que la fonction $\tau o_{\delta \rightarrow 0}(1)$ peut s'écrire $o_{\delta \rightarrow 0}(\delta)$. Ceci nous permet de conclure. \square

L'algorithme 40 que nous proposons est donc consistant et stable si l'inégalité (7.14) est respectée.

7.3.2 Mise en œuvre de la reconstruction de l'image

Pour mettre en œuvre l'algorithme 40 plusieurs choix se présentent. Tout d'abord, il y a le choix de l'initialisation et ensuite le choix des paramètres.

Pour l'initialisation nous prenons l'initialisation aléatoire, avec ce type d'initialisation, nous n'avons pas à faire le choix d'une valeur particulière. En effet si nous imposons une valeur constante dans le trou Ω à l'initialisation, il faut alors trouver un critère pour choisir cette valeur. Nous avons alors des risques d'avoir un biais et alors d'avoir des minima locaux. Nous prenons donc plutôt une initialisation aléatoire. De plus dans ce cas le gradient n'est pas nul dans le trou Ω ce qui fait que les valeurs bougent plus rapidement au moins au début.

Reste le choix des paramètres. L'algorithme 40 que nous utilisons sur I possède trois paramètres.

- Le paramètre de l'équation (6.20) : ν .
- Les paramètres de discrétisation : δ , le pas de discrétisation spatial, et τ , le pas de temps.

Le paramètre ν de l'équation permet d'augmenter ou de diminuer dans l'équation (6.20) l'influence du terme de diffusion. Plus ν est élevé, plus la diffusion prend le pas sur le transport.

Le pas spatial δ correspond à la distance entre deux pixels de l'image. Il est donc fixé par la nature de l'image, nous le prenons égal à 1. Le pas de temps influence la convergence de l'algorithme, plus le pas de temps τ est grand, plus la convergence est rapide. Nous avons donc intérêt à prendre τ le plus grand possible. Comme nous venons de voir que ce dernier doit vérifier l'inégalité (7.14), nous prenons le plus grand pas de temps possible :

$$\tau = \frac{\delta}{4\pi\nu/\delta + \sqrt{2}}.$$

Au final il ne reste donc plus que le paramètre du schéma, ν . Pour tester notre algorithme de reconstruction et voir l'influence du paramètre ν nous allons faire des tests sur des images. Dans un premier temps nous effectuons des tests sur des images synthétiques, ensuite nous regardons des résultats obtenus sur un exemple d'image naturelle.

Résultats sur des images synthétiques. Pour tester notre équation de reconstruction (6.20) nous commençons par des tests sur des images synthétiques simples. Nous choisissons trois images. Chacune de ces images représente un bord. L'image passe du blanc au noir. Comme sur des images naturelles le passage du noir au blanc ne se fait pas si brutalement, chaque image est légèrement lissée par un filtre gaussien. La première image correspond à un bord rectiligne. La deuxième image est en fait constituée de deux bords rectilignes, c'est l'image d'une ligne. Enfin la troisième image contient un bord recourbé. Ces images sont représentées figure 7.5 avec le masque qui sert à définir le trou Ω .

Nous souhaitons tester uniquement l'équation de reconstruction (6.20) à ce niveau. En conséquence nous calculons un champ de probabilités associé à l'image selon la méthode décrite à la fin de la section 6.3. Le calcul de ce champ

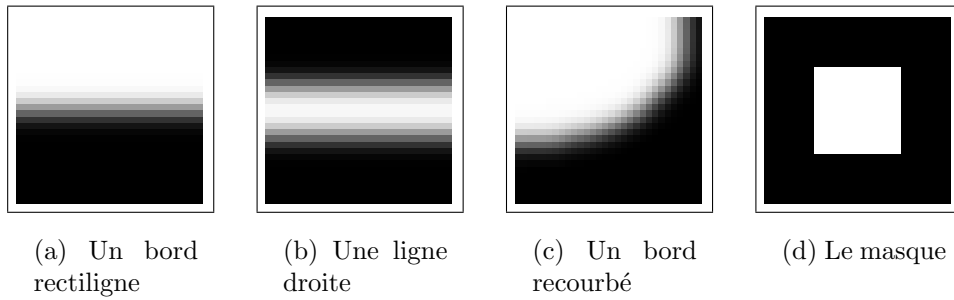


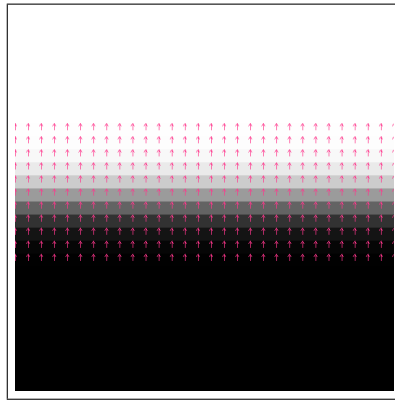
FIG. 7.5 – Nous utilisons les trois premières images pour faire tester l'équation de reconstruction (6.20) avec l'algorithme 40. La première image est l'image d'un bord droit. La deuxième image est l'image d'une ligne et la troisième est l'image d'un bord recourbé. Sur chacune de ces images est appliqué un petit filtre gaussien pour lisser les images. La quatrième image définit le trou Ω à l'intérieur duquel l'image doit être reconstruite.

de probabilités I -compatible est fait à partir de l'image complète, sans trou. Sur la figure 7.6 sont représentées les orientations moyennes de ces champs de probabilité.

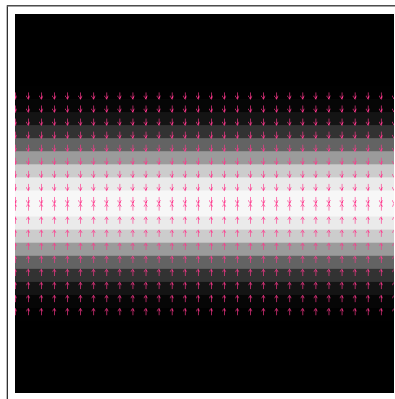
Ensuite nous initialisons l'image dans le masque avec des valeurs aléatoires. Enfin, nous appliquons l'algorithme 40 de reconstruction sur les images synthétiques de la figure 7.5 pour différentes valeurs du paramètre ν . Pour ces tests, nous connaissons l'image de départ, les bonnes valeurs que doit avoir la fonction dans le trou. Dans ce contexte, nous pouvons comparer les images obtenues pour les différentes valeurs de ν avec l'image originale. Nous cherchons un résultat le plus proche possible de l'image de départ. Le critère que nous utilisons pour sélectionner la valeur de ν qui donne le meilleur résultat est la norme L^2 de la différence entre le résultat obtenu et l'image originale.

Les résultats sont présentés dans la figure 7.7. Pour chacune des courbes à gauche l'ordonnée correspond à la norme L^2 de la différence entre le résultat obtenu et l'image originale en fonction de ν qui correspond à l'abscisse. Nous cherchons un minimum sur chacune de ces courbes. Pour les images du bord rectiligne et de la ligne droite, le minimum trouvé est très proche de $\nu = 0$, $\nu = 0,01$ pour le bord droit et c'est même sans diffusion que la reconstruction de la ligne droite est la meilleure. Par contre pour le bord recourbé le minimum a lieu pour $\nu = 10$. Chacun des résultats qui minimisent notre critère est présenté à côté de la courbe correspondante sur la figure 7.7.

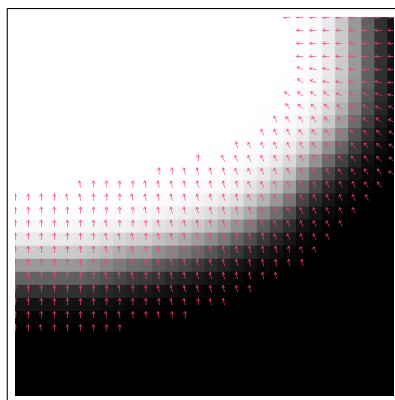
Les deux premières images sont parfaitement bien reconstruites. Pour la dernière, les lignes de niveau à reconstruire sont courbes. Cependant le transport de l'information ne se recolle pas forcément très bien vers le bord opposé du trou. L'information est transportée à partir d'un des bords de l'image. Vers



(a) Bord rectiligne

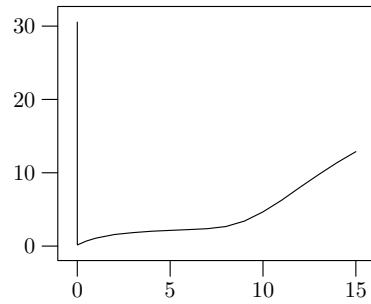


(b) Ligne

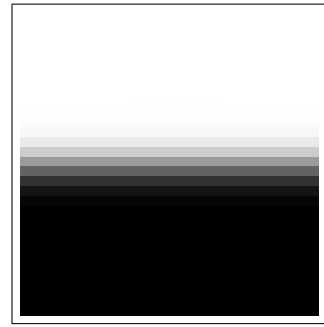


(c) Bord recourbé

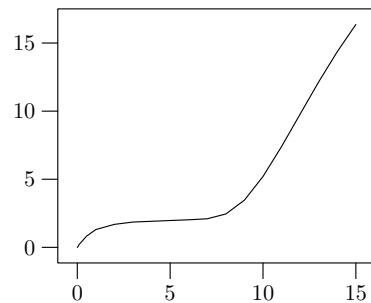
FIG. 7.6 – Au dessus de chacune des images synthétiques de la figure 7.5 sont superposés les champs d’orientations moyennes. Ces champs sont calculés à partir de ces images synthétiques.



(a) Erreur en fonction de ν pour le bord rectiligne



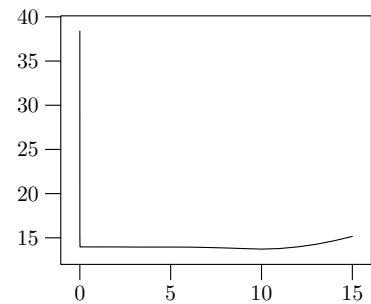
(b) Résultat avec $\nu = 0,01$ pour le bord rectiligne



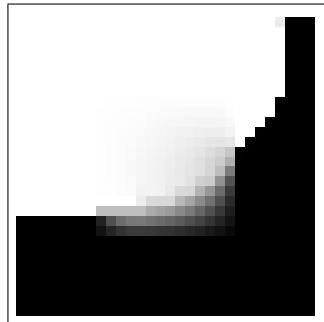
(c) Erreur en fonction de ν pour la ligne droite



(d) Résultat avec $\nu = 0$ pour la ligne droite



(e) Erreur en fonction de ν pour le bord recourbé



(f) Résultat avec $\nu = 10$ pour le bord recourbé

FIG. 7.7 – Les courbes de gauche représente la norme L^2 de la différence du résultat de la désocclusion avec l'image originale pour différentes valeurs de ν . Ces courbes représentent donc l'erreur. À droite de chaque courbe est représenté le résultat pour la valeur de ν qui minimise la courbe. De haut en bas nous avons $\nu = 0,01$, puis 0 et enfin 10.

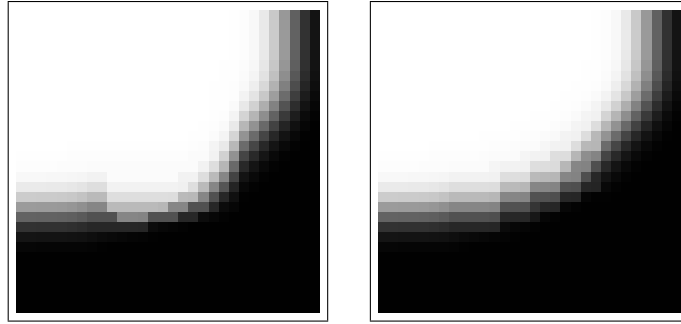


FIG. 7.8 – Le résultat obtenu avec l’algorithme 40 appliqué à chacun des niveaux de l’image séparément. À gauche sans filtre de Deriche et à droite avec un filtre de Deriche (cf. annexe B) pour lisser le gradient.

le bord opposé nous avons beaucoup de petites erreurs qui se sont ajoutées au fur et à mesure de l’algorithme ce qui explique ces défauts. Il est possible pour faire la reconstruction de séparer l’image en ensembles de niveau, puis de traiter chacun des ensembles de niveau séparément avant de rassembler tous les résultats pour reformer une image. La reconstruction sur des images binaires se fait plus facilement et évite ces erreurs. Le résultat de cette technique est l’image de la figure 7.8.

ALGORITHME 44 (Séparation par ligne de niveau) :

Pour n de 0 à 255 faire

Création de l’image contenant l’ensemble de niveau n de l’image initiale

Application de l’algorithme 40 à cette image

Reconstruction d’une image à partir de chacun des résultats en utilisant l’équation (6.1)

Résultats sur des images naturelles. Ensuite nous effectuons des tests sur un exemple d’image réelle. La figure 7.9 montre l’image de phare utilisée ainsi que le masque choisi.

La figure 7.10 montre le résultat obtenu avec $\nu = 1$. Pour améliorer le résultat nous utilisons le filtre de Deriche décrit dans [10] et dans l’annexe B pour calculer le gradient de l’image. En effet toutes les images naturelles contiennent du bruit. Le calcul du gradient de l’image est très sensible au bruit, le filtre de Deriche permet un calcul moins sensible au bruit du gradient. Le résultat est dans la figure 7.10. Nous voyons nettement sur l’image que la reconstruction est bien meilleure.



(a) Un phare

(b) Le masque

FIG. 7.9 – Une image naturelle et le masque utilisé pour tester l’algorithme 40.



(a) Le champ de vecteurs utilisé

(b) Le résultat de l’algorithme

FIG. 7.10 – Le résultat de l’algorithme 40 sur l’image.



(a) Le champ de vecteurs par le filtre de Deriche

(b) Le résultat avec ce champ de vecteurs

FIG. 7.11 – Le résultat sur l'image du phare en utilisant le filtre de Deriche.

7.4 Couplage des deux équations

7.4.1 Initialisation

Pour commencer l'algorithme il faut initialiser les deux fonctions P et I sur toute l'image, y compris dans le trou Ω . Pour les valeurs de P , nous utilisons la section 6.3 pour les calculer à partir de celles de I . Il ne reste donc qu'à initialiser les valeurs de I .

Initialisation de I . Pour initialiser I nous avons le choix entre deux possibilités. La figure 7.12 illustre ces deux possibilités. Pour les trois premières images nous avons rempli le trou Ω par une valeur constante, respectivement 0, 128 et 255. La première possibilité est effectivement de remplir le trou avec une valeur constante. Il y a aussi la possibilité utilisée pour la quatrième image de la figure, celle de remplir le trou avec des valeurs aléatoires.

Initialisation de P . Une fois que nous avons initialisé I , nous voulons initialiser P de manière à ce que le champ de probabilité initial soit I -compatible. Pour cela, nous utilisons le procédé décrit dans la section 6.3. Aux points de l'image où le gradient de l'image est non nul, nous initialisons P avec la formule (6.16) par une gaussienne. Aux points où le gradient de l'image est nul,

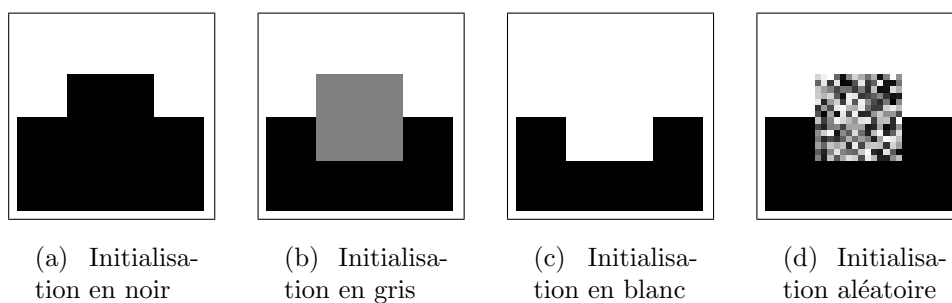


FIG. 7.12 – Différentes initialisations sont possibles pour I . Soit nous mettons une valeur constante dans le trou Ω , soit nous mettons des valeurs aléatoires comme sur la dernière image.

nous initialisons P par une distribution uniforme, c'est la formule (6.17). Ainsi lorsque le gradient de I n'est pas nul, l'orientation moyenne de P correspond à l'orthogonal direct de l'orientation du gradient et son écart-type est bien inversement proportionnel à la norme du gradient.

7.4.2 Résultats

Les paramètres du schéma complet sont listés ci-après.

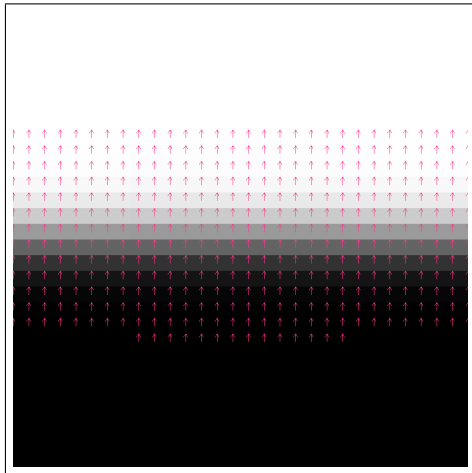
- Les paramètres de diffusion :
 - α la diffusion angulaire de P ,
 - β la diffusion spatiale de P ,
 - et ν la diffusion de I .
- les pas de discrétisation :
 - δ le pas de discrétisation spatial,
 - K le nombre d'angles discrets,
 - et τ le pas de temps.

Parmi tous ces paramètres, seuls ceux de diffusion sont réellement libres. Par contre pour résoudre simultanément les deux équations nous devons alterner les deux schémas. Ceci implique de nouveaux paramètres que nous appelons N_P et N_I qui sont respectivement le nombre d'itérations pour la boucle sur P et pour celle sur I lorsque nous alternons les deux schémas.

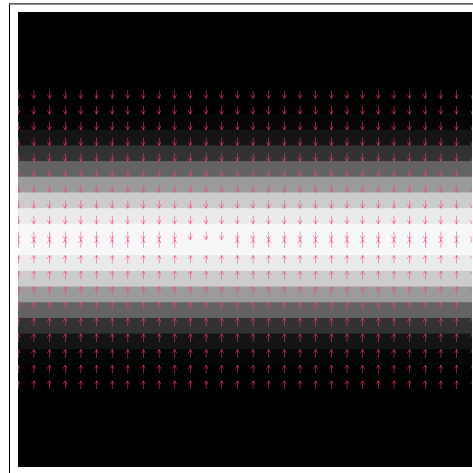
Lorsque ν est trop petit un nouveau problème apparaît, la différence entre deux itérations entre l'image avant et après ne tend pas forcément vers 0. En augmentant un peu ν , ce problème disparaît.

Pour que l'algorithme soit stable il faut introduire tous les termes de diffusion. Pour limiter la diffusion que cela entraîne nous avons *a priori* deux possibilités, soit nous changeons le terme ou les termes de diffusion linéaire par des termes de diffusion non linéaire, par exemple en remplaçant le laplacien par

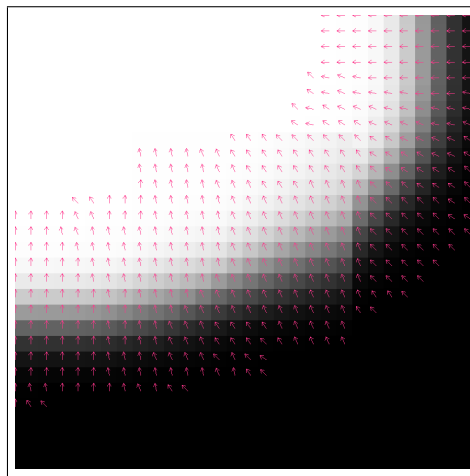
la courbure, soit nous séparons les lignes de niveau de l'image pour appliquer l'algorithme séparément à chacun des ensembles de niveau de l'image.



(a) bord flou



(b) ligne floue



(c) courbe floue

FIG. 7.13 – Voici les résultats obtenus sur les trois images de la figure 7.5 en prenant $N_P = 1000$ et $N_I = 20$.

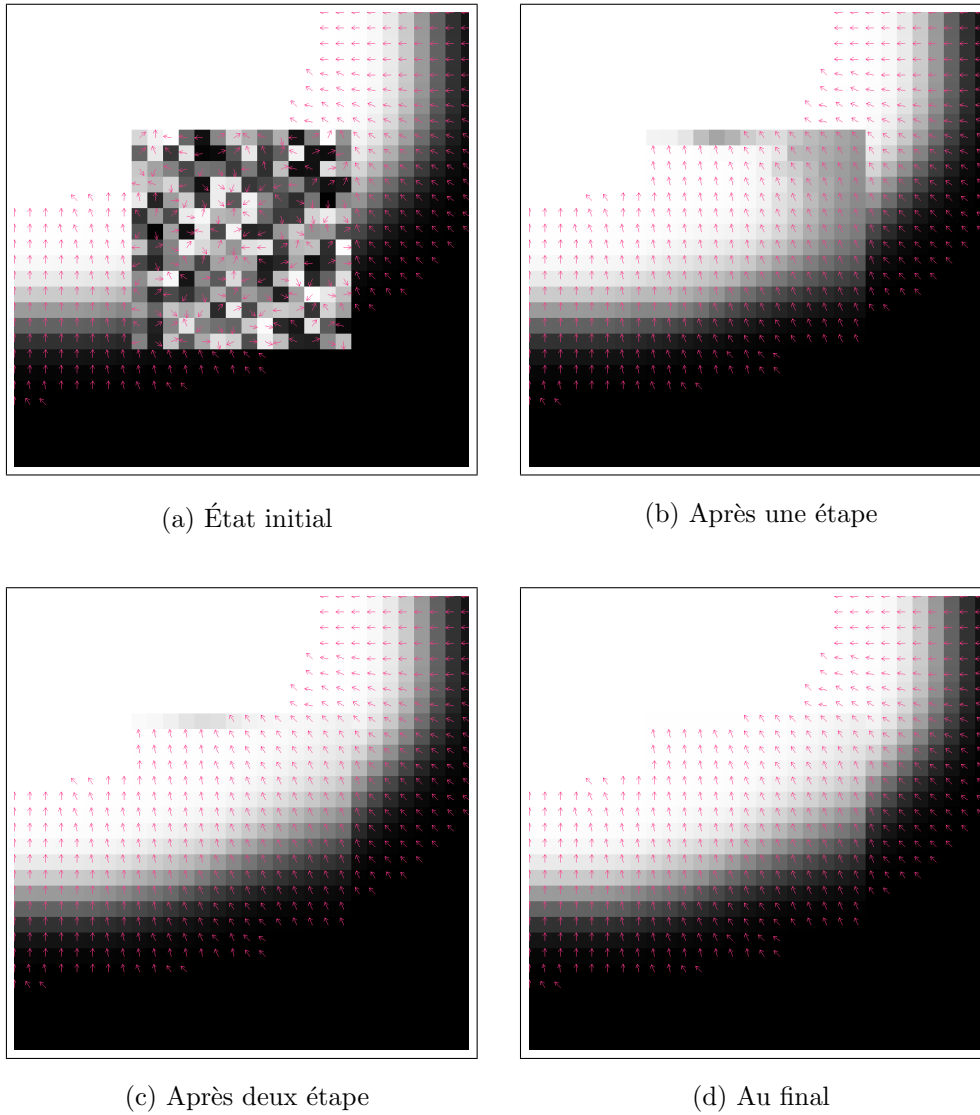


FIG. 7.14 – Voici les premières étapes pour le résultat 7.13(c) obtenu sur la courbe floue ainsi que le résultat final.

Conclusion

Dans la première partie nous avons commencé l'étude de la détection de constances de largeur dans les images. Pour cela nous avons utilisé l'approche de détection sans paramètre introduite par Morel, Moisan et Desolneux dans [12, 13, 14]. Cette approche est basée sur l'école Gestaltiste expliquée dans [22] et le principe de Helmholtz. Pour l'appliquer nous avons passé en revue différentes modélisations possibles du problème de la constance de largeur.

Pour détecter la constance de largeur nous devons détecter le fait que deux courbes restent parallèles. Dans cette partie nous introduisons un moyen d'évaluer quantitativement le parallélisme de deux courbes. Cette mesure permet de continuer l'approche de Morel et al. pour la constance de largeur. Il s'agit désormais de faire des tests systématiques sur les images et de garder automatiquement les meilleures mesures. Le concept qui restera alors à dégager est le concept de maximalité.

Au cours de l'élaboration de cette mesure de parallélisme, nous sommes partis d'un modèle discret pour obtenir finalement un estimateur de cette mesure, bien plus rapide. Nous avons également obtenu à partir du modèle discret une EDP correspondant au modèle limite obtenu en faisant tendre le paramètre r du modèle discret vers 0. L'EDP que nous avons obtenue est proche des équations utilisées pour le problème de la désocclusion. L'objet de la deuxième partie est d'appliquer cette équation au problème de la désocclusion.

Dans la deuxième partie nous proposons une nouvelle modélisation pour un problème qui devient classique, celui de la désocclusion. La méthode que nous proposons utilise une modélisation probabiliste de l'orientation du gradient dans l'image. Les résultats que nous obtenons ne sont pas encore optimaux. Cependant ils sont prometteurs.

Toute la modélisation que nous avons menée dans la première partie autour du parallélisme de deux courbes est nouvelle. Cette modélisation est nécessaire pour parvenir à mettre au point un détecteur de constance de largeur dans les images. Cette modélisation nous a mené à fournir trois outils pour effectuer une mesure quantitative du parallélisme de deux courbes. C'est cette mesure qui va nous permettre par la suite de faire de la détection. Pour détecter une constance de largeur dans une image il ne reste plus qu'à faire des tests systématiques sur les courbes significatives trouvées dans une image. Parmi tous ces tests ceux qui seront positifs seront ceux dépassant un certain seuil à

déterminer en fonction du nombre de tests qui auront été effectués.

Pour la désocclusion notre contribution est de modéliser l'incertitude sur la mesure du gradient de l'image en introduisant un champ de probabilités. Le champ de probabilités introduit prend en compte à la fois le résultat de la mesure du gradient mais également la précision de la mesure. Nous avons ensuite introduit deux équations couplées à résoudre, une équation que doit vérifier le champ de probabilités dont le but est de le reconstruire dans le trou et une seconde équation qui permet de reconstruire l'image étant donné le champ de probabilités. La désocclusion est un problème complexe et il n'y a pas de solution unique pour chaque image. Nous avons introduit des schémas stables pour résoudre le système obtenu contenant deux EDPs. Enfin nous avons examiné l'influence de différents paramètres sur les résultats obtenus. Une fois le problème du choix des paramètres résolu, il est relativement facile d'adapter l'algorithme à un nombre plus élevé de dimensions. En effet il n'y a à cela aucun obstacle théorique, il ne reste qu'à adapter les schémas numériques.

Notre méthode de désocclusion ne convient que pour les images ne contenant pas de texture. Pour les images de textures d'autres méthodes existent qui leur sont spécifiques comme par exemple [16]. N'importe quelle image naturelle comporte à la fois des textures et des informations de nature géométrique. Une idée est de décomposer l'image en deux composantes suivant le livre de Meyer [29], une composante de texture et une composante géométrique. Pour cela Vese et Osher ont présenté dans [38] une méthode numérique pour appliquer la décomposition. Aujol et al. [1] ont également proposée une autre méthode pour cette décomposition. Une fois l'image décomposée, nous avons alors deux images pour la désocclusion. Une image contenant les textures de l'image originale et une deuxième image contenant les informations géométriques. Pour la partie contenant les textures nous pouvons employer la méthode d'Efros et Leung pour faire la désocclusion. Sur l'autre image, celle contenant la géométrie nous appliquons notre algorithme. Enfin il faut recomposer les deux images obtenues pour avoir le résultat final, la désocclusion de l'image originale.

Troisième partie

Annexes

Annexe A

Diffusions

Pour améliorer la netteté des images, nous pouvons remplacer la diffusion isotrope engendrée par le terme ΔI par un terme de diffusion anisotrope. Deux des choix possibles sont par exemple la courbure $\text{curv}(I) = \text{div}(\nabla I / \|\nabla I\|)$, ou bien l'opérateur de Malik et Perona $\text{div}(\nabla I / (1 + \|\nabla I\|^2 / \lambda^2))$.

Intéressons-nous au terme de diffusion. C'est un terme de diffusion classique et il peut être relié à une minimisation d'énergie. En effet une équation de diffusion peut être une équation d'Euler-Lagrange associée à une minimisation. Prenons donc le cas du laplacien et examinons l'énergie suivante

$$J_{\Delta}(u) = \int_{\Omega} \|\nabla u\|^2.$$

Un théorème fondamental en analyse est celui qui stipule qu'en un minimum local d'une application dérivable sa dérivée s'annule. Ce théorème nous permet de passer du problème de la minimisation d'énergie à la résolution d'une équation. Ici l'équation d'Euler-Lagrange est

$$\nabla J_{\Delta}(u) = 0$$

où ∇J_{Δ} désigne la dérivée au sens de Gâteaux de la fonctionnelle. Pour la calculer nous regardons

$$J_{\Delta}(u + tv) - J_{\Delta}(u).$$

Si u minimise J_{Δ} alors cette quantité est positive quelle que soit la fonction v pour t proche de 0. Regardons donc $J_{\Delta}(u + tv)$.

$$\begin{aligned} J_{\Delta}(u + tv) &= \int_{\Omega} \|\nabla(u + tv)\|^2 \\ &= \int_{\Omega} \|\nabla u + t\nabla v\|^2 \\ &= \int_{\Omega} \|\nabla u\|^2 + 2t\nabla u \cdot \nabla v + o_{t \rightarrow 0}(t) \\ &= J_{\Delta}(u) - 2t \int_{\Omega} \text{div}(\nabla u)v + o_{t \rightarrow 0}(t) \end{aligned}$$

Et donc nous avons $\nabla J_{\Delta}(u) = -2\operatorname{div}(\nabla u) = -2\Delta u$

Ainsi le terme de diffusion ΔI correspond à la minimisation de la norme L^2 du gradient. Cette diffusion est connue pour rendre l'image floue très rapidement. De plus la position des bords de l'image n'est pas conservée. Ceci est rappelé par exemple dans l'article de Weickert [39].

Ceci nous incite à regarder d'autres fonctionnelles qui permettent de contourner ces inconvénients. Par exemple nous avons vu que la fonctionnelle de la variation totale

$$J_{VT}(u) = \int_{\Omega} \|\nabla u\|$$

minimise la longueur des lignes de niveau. Examinons l'équation d'Euler-Lagrange associée.

$$\begin{aligned} J_{VT}(u + tv) &= \int_{\Omega} \|\nabla(u + tv)\| \\ &= \int_{\Omega} \sqrt{\|\nabla(u + tv)\|^2} \\ &= \int_{\Omega} \sqrt{\|\nabla u\|^2 + 2t\nabla u \cdot \nabla v + o_{t \rightarrow 0}(t)} \\ &= \int_{\Omega} \|\nabla u\| \sqrt{1 + 2t \frac{\nabla u}{\|\nabla u\|^2} \cdot \nabla v + o_{t \rightarrow 0}(t)} \\ &= \int_{\Omega} \|\nabla u\| \left(1 + t \frac{\nabla u}{\|\nabla u\|^2} \cdot \nabla v + o_{t \rightarrow 0}(t) \right) \\ &= J_{VT}(u) + t \int_{\Omega} \frac{\nabla u}{\|\nabla u\|} \cdot \nabla v + o_{t \rightarrow 0}(t) \\ &= J_{VT}(u) - t \int_{\Omega} \operatorname{div} \left(\frac{\nabla u}{\|\nabla u\|} \right) \nabla v + o_{t \rightarrow 0}(t) \end{aligned}$$

Nous trouvons donc que $\nabla J_{VT}(u) = -\operatorname{div}(\nabla u / \|\nabla u\|) = -\operatorname{curv}(u)$. Ainsi si nous utilisons $\operatorname{curv}(I)$ comme terme de diffusion, nous minimisons la longueur des lignes de niveau de I .

Avec le terme de courbure la diffusion devient anisotrope et non-linéaire. Perona et Malik dans [32] introduisent un autre type de diffusion non-linéaire. Regardons donc une troisième fonctionnelle,

$$J_{PM}(u) = \int_{\Omega} \frac{\|\nabla u\|^2}{1 + \frac{\|\nabla u\|^2}{\lambda^2}}$$

Équation d'Euler-Lagrange	Énergie minimisée
$\Delta u = 0$	$\int_{\Omega} \ \nabla u\ ^2$
$\text{curv}(u) = 0$	$\int_{\Omega} \ \nabla u\ $
$\text{div} \left(\frac{\ \nabla u\ ^2}{\left(1 + \frac{\ \nabla u\ ^2}{\lambda^2}\right)^2} \right) = 0$	$\int_{\Omega} \frac{\ \nabla u\ ^2}{1 + \frac{\ \nabla u\ ^2}{\lambda^2}}$

TAB. A.1 – Des équations de diffusion et l'énergie correspondante.

et sa dérivée,

$$\begin{aligned}
J_{PM}(u + tv) &= \int_{\Omega} (\|\nabla u\|^2 + 2t \nabla u \cdot \nabla v + o_{t \rightarrow 0}(t)) \\
&\quad \times \left(1 + \frac{\|\nabla u\|^2}{\lambda^2} + 2t \frac{\nabla u \cdot \nabla v}{\lambda^2} + o_{t \rightarrow 0}(t) \right)^{-1} \\
&= \int_{\Omega} \frac{\|\nabla u\|^2 + 2t \nabla u \cdot \nabla v + o_{t \rightarrow 0}(t)}{1 + \frac{\|\nabla u\|^2}{\lambda^2}} \\
&\quad \times \left(1 + 2t \frac{\nabla u}{\lambda^2 \left(1 + \frac{\|\nabla u\|^2}{\lambda^2}\right)} \cdot \nabla v + o_{t \rightarrow 0}(t) \right)^{-1} \\
&= \int_{\Omega} \frac{\|\nabla u\|^2 + 2t \nabla u \cdot \nabla v + o_{t \rightarrow 0}(t)}{1 + \frac{\|\nabla u\|^2}{\lambda^2}} \\
&\quad \times \left(1 - 2t \frac{\nabla u}{\lambda^2 \left(1 + \frac{\|\nabla u\|^2}{\lambda^2}\right)} \cdot \nabla v + o_{t \rightarrow 0}(t) \right) \\
&= J_{PM}(u) + 2t \int_{\Omega} \text{div} \left(\frac{\nabla u}{\left(1 + \frac{\|\nabla u\|^2}{\lambda^2}\right)^2} \right) v + o_{t \rightarrow 0}(t)
\end{aligned}$$

Ainsi un terme de diffusion du type $\text{div}(\nabla I / (1 + (\nabla I^2 / \lambda^2))^2)$ correspond à la minimisation de

$$\int_{\Omega} \frac{\|\nabla I\|^2}{1 + \frac{\|\nabla I\|^2}{\lambda^2}}.$$

Le tableau A.1 récapitule les différents termes de diffusion ainsi que les énergies qu'ils minimisent.

La figure A.1 montre l'effet de ces équations de diffusion sur la photographie d'un lampadaire à Durham. Les coefficients utilisés sont 0,3 pour le pas de temps, 0,8 pour le coefficient de diffusion et pour 100 itérations. Dans le cas

du terme de diffusion du type Perona-Malik, dans cette figure nous avons utilisé $\lambda = 20$. La figure A.2 montre l'effet de cette diffusion pour $\lambda = 1, 5, 10, 15$ et 20 . Pour choisir la valeur de ce paramètre λ il est utile de regarder la distribution de la norme du gradient. La figure A.3 montre l'histogramme de la norme du gradient de la photo de Durham. Sur cet histogramme on constate l'effet "ciel bleu", un nombre important de petits gradients dûs aux zones où l'image varie peu. On déduit de l'histogramme qu'il faut prendre λ autour de 10 .



(a) Image originale



(b) Laplacien



(c) Courbure



(d) Perona-Malik,
 $\lambda = 20$

FIG. A.1 – Effets des différents types de diffusion. Pas de temps de 0,3 et coefficient de diffusion de 0,8 sur 100 itérations. Pour la diffusion de type Perona-Malik, nous avons pris $\lambda = 20$.



(a) $\lambda = 1$



(b) $\lambda = 5$



(c) $\lambda = 10$



(d) $\lambda = 15$

FIG. A.2 – Plusieurs diffusions de type Perona-Malik, pour différentes valeurs de λ . En haut à gauche $\lambda = 1$, en haut à droite, $\lambda = 5$, en bas à gauche, $\lambda = 10$ et en bas à droite $\lambda = 15$

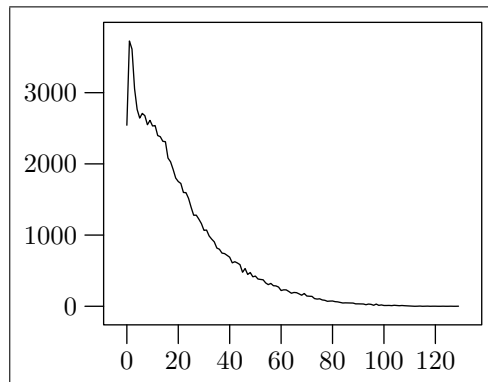


FIG. A.3 – Histogramme de la norme du gradient de l'image de Durham

Annexe B

Détecteur optimal de bord de Deriche

Dans [10] Deriche analyse le travail de Canny dans [6]. Il en conclut que l'opérateur optimal pour détecter les bords (dans un signal unidimensionnel) est la convolution par la fonction

$$g(x) = -cxe^{-\alpha|x|}, \quad (\text{B.1})$$

où c est une constante et α est un paramètre. En fait $1/\alpha$ joue le même rôle que l'écart-type σ d'une gaussienne, plus α est petit moins bonne est la localisation du bord par le détecteur, cependant le rapport signal sur bruit est meilleur et vice-versa.

Après avoir présenté l'opérateur optimal (B.1) Deriche propose une implémentation de cet opérateur par un filtrage récursif. Ceci permet de simplifier l'opérateur par rapport à un calcul direct d'un produit de convolution. Pour utiliser ce filtrage récursif plusieurs coefficients sont nécessaires qui sont tous définis à partir du paramètre α . Le premier coefficient est la constante c de l'opérateur (B.1) qui est en fait un coefficient de normalisation.

$$c = \frac{(1 - e^{-\alpha})^2}{e^{-\alpha}} \text{ et } k = \frac{(1 - e^{-\alpha})^2 \alpha^2}{1 + 2\alpha e^{-\alpha} - e^{-2\alpha}}, \quad (\text{B.2})$$

$$a = -ce^{-\alpha}, b_1 = -2e^{-\alpha} \text{ et } b_2 = e^{-2\alpha}, \quad (\text{B.3})$$

$$c_1 = \frac{k}{\alpha} \text{ et } c_2 = \frac{k}{\alpha^2}, \quad (\text{B.4})$$

$$a_0 = c_2, a_1 = (c_1 - c_2)e^{-\alpha}, a_2 = a_1 - c_2b_1 \text{ et } a_3 = -c_2b_2 \quad (\text{B.5})$$

Le filtrage récursif s'effectue à partir de l'image I connue sur les points (i, j)

de la grille. Le premier filtrage sert à calculer $r(i, j)$ qui correspond à $\partial I/\partial y$.

$$y^+(i, j) = I(i, j - 1) - b_1 y^+(i, j - 1) - b_2 y^+(i, j - 2), \quad (\text{B.6})$$

$$y^-(i, j) = I(i, j + 1) - b_1 y^-(i, j + 1) - b_2 y^-(i, j + 2), \quad (\text{B.7})$$

$$r^0(i, j) = a(y^+(i, j) - y^-(i, j)), \quad (\text{B.8})$$

$$r^+(i, j) = a_0 r^0(i, j) + a_1 r^0(i - 1, j) - b_1 r^+(i - 1, j) - b_2 r^+(i - 2, j), \quad (\text{B.9})$$

$$r^-(i, j) = a_2 r^0(i + 1, j) + a_3 r^0(i + 2, j) - b_1 r^-(i + 1, j) - b_2 r^-(i + 2, j), \quad (\text{B.10})$$

$$r(i, j) = r^-(i, j) + r^+(i, j). \quad (\text{B.11})$$

Enfin le filtre symétrique permet de calculer $s(i, j)$ qui correspond cette fois à $\partial I/\partial x$.

$$x^+(i, j) = I(i - 1, j) - b_1 x^+(i - 1, j) - b_2 x^+(i - 2, j), \quad (\text{B.12})$$

$$x^-(i, j) = I(i + 1, j) - b_1 x^-(i + 1, j) - b_2 x^-(i + 2, j), \quad (\text{B.13})$$

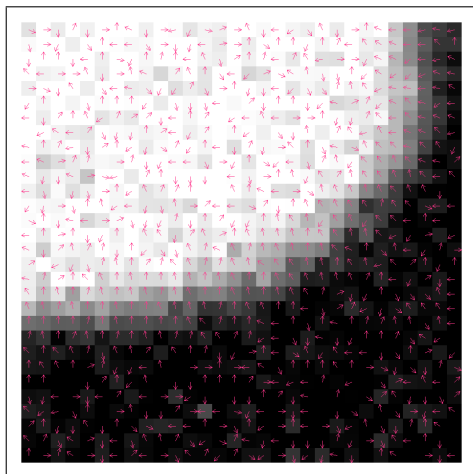
$$s^0(i, j) = a(x^+(i, j) - x^-(i, j)), \quad (\text{B.14})$$

$$s^+(i, j) = a_0 s^0(i, j) + a_1 s^0(i, j - 1) - b_1 s^+(i, j - 1) - b_2 s^+(i, j - 2), \quad (\text{B.15})$$

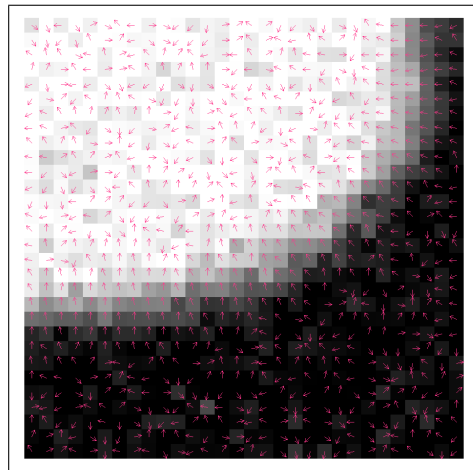
$$s^-(i, j) = a_2 s^0(i, j + 1) + a_3 s^0(i, j + 2) - b_1 s^-(i, j + 1) - b_2 s^-(i, j + 2), \quad (\text{B.16})$$

$$s(i, j) = s^-(i, j) + s^+(i, j). \quad (\text{B.17})$$

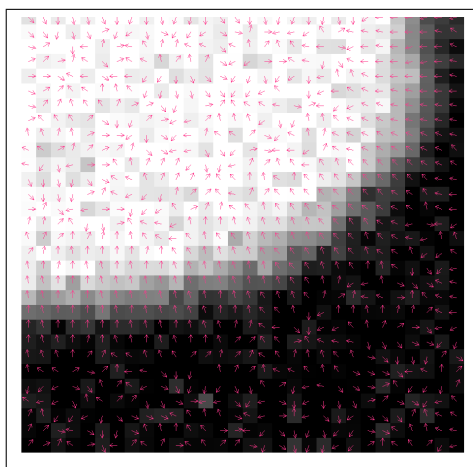
Les résultats de ce filtre sont présentés sur la figure **B.1**. L'image (a) montre l'image bruitée qui est utilisée et en surimpression, les vecteurs représentent l'orientation obtenue par différence finie, sans aucun filtre. Les images suivantes, (b), (c) et (d) montrent l'effet sur le champ d'orientation d'un filtrage avec $1/\alpha$ valant respectivement 0,1 puis 0,5 et enfin 1. La régularisation du champ d'orientation, au moins aux points où le gradient a une norme importante, c'est-à-dire le long du bord, est très visible dès l'image (b).



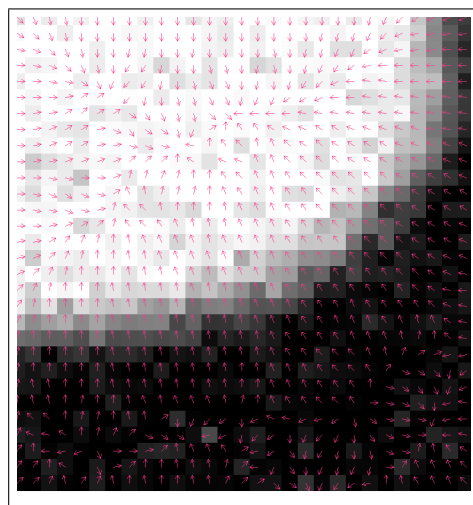
(a) Sans filtre



(b) Filtre avec $1/\alpha = 0,1$



(c) Filtre avec $1/\alpha = 0,5$



(d) Filtre avec $1/\alpha = 1$

FIG. B.1 – Résultat du filtre de Deriche sur l'orientation du gradient.

Bibliographie

- [1] J.-F. AUJOL, G. AUBERT, L. BLANC-FÉRAUD et A. CHAMBOLLE : Image Decomposition. Application to SAR Images. Dans *Scale Space'03*, volume 1682 de *Lecture Notes in Computer Science (LNCS)*, 2003.
- [2] C. BALLESTER, M. BERTALMIO, V. CASELLES, G. SAPIRO et J. VERDERA : Filling-In Interpolation by Variational Method. Dans *IEEE International Conference on Computer Vision (ICCV'01)*, 2001.
- [3] M. BERTALMIO, A. BERTOZZI et G. SAPIRO : Navier-Stokes, Fluid-Dynamics and Image and Video Inpainting. Dans *IEEE Computer Vision and Pattern Recognition (CVPR'01)*, 2001.
- [4] M. BERTALMIO, G. SAPIRO, C. BALLESTER et V. CASELLES : Image Inpainting. Dans Kurt AKELEY, éditeur : *SIGGRAPH'00*, Computers Graphics proceedings, pages 417–424, 2000.
- [5] E. BIENENSTOCK, S. GEMAN et D. POTTER : Compositionality, MDL Priors, and Object Recognition. Dans M.I. Jordan et T. Petsche M.C. MOZER, éditeur : *Advances in Neural Information Processing Systems 9*. MIT Press, 1998.
- [6] J.F. CANNY : Finding Edges and Lines in Images. Rapport technique AI-TR-720, Artificial Intelligence Laboratory, juin 1983.
- [7] T. CHAN, S. H. KANG et J. SHEN : Simultaneous Structure and Texture Image Inpainting. *IEEE Transaction on Image Processing*, 12(8):882–889, 2003.
- [8] T. CHAN et J. SHEN : Mathematical Models for Local Non-Texture Inpaintings. *SIAM Journal on Applied Mathematics*, 62(3):1019–1043, 2001.
- [9] T. CHAN et J. SHEN : Non-Texture Inpainting by Curvature-Driven Diffusion (CDD). *Journal of Visual Communication and Image Representation*, 12(4):436–449, 2001.
- [10] R. DERICHE : Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector. *International Journal of Computer Vision*, 1(2):167–187, 1987.
- [11] A. DESOLNEUX : *Événements significatifs et applications à l'analyse d'images*. Mathématiques, École Normale Supérieure de Cachan, décembre 2000.

- [12] A. DESOLNEUX, L. MOISAN et J.-M. MOREL : Meaningful Alignments. *International Journal of Computer Vision*, 40(1):7–23, 2000.
- [13] A. DESOLNEUX, L. MOISAN et J.-M. MOREL : Edge Detection by Helmholtz Principle. *Journal of Mathematical Imaging and Vision*, 14(3):271–284, 2001.
- [14] A. DESOLNEUX, L. MOISAN et J.-M. MOREL : A Grouping Principle and Four Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):508–513, avril 2003.
- [15] A. DESOLNEUX, L. MOISAN et J.-M. MOREL : *Gestalt Theory and Computer Vision*, chapitre de “Seeing, Thinking and Knowing”, pages 71–101. Kluwer Academic Publishers, 2004.
- [16] A. EFROS et T. LEUNG : Texture Synthesis by Non-parametric Sampling. Dans *IEEE International Conference on Computer Vision (ICCV’99)*, pages 1033–1038, 1999.
- [17] D. EUVRARD : *Résolution numérique des équations aux dérivées partielles, de la physique, de la mécanique et des sciences de l’ingénieur : différences finies, éléments finis, problèmes en domaine non borné*. Masson, 1994.
- [18] D. GEMAN et B. JEDYNAK : An Active Testing Model for Tracking Roads in Satellite Images. Rapport de recherche 2757, INRIA Sophia Antipolis, décembre 1995.
- [19] F. GLAZER : Curve Finding by Ridge Detection and Grouping. Dans W. KROPATSCH et H. BISCHOF, éditeurs : *Mustererkennung*, volume 5 de *Deutsche Arbeitsgemeinschaft für Mustererkennung*, pages 109–116. Informatik Xpress, 1994.
- [20] P. GREEN : Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination. *Biometrika*, 82:711–732, 1995.
- [21] J. HERSHBERGER et J. SNOEYINK : Computing Minimum Length Paths of a Given Homotopy Class. *Computational Geometry Theory and Applications*, 4(D):63–98, 1994.
- [22] G. KANIZSA : *La grammaire du voir*. Editions Diderot, 1981.
- [23] A.C. KOKARAM, R.D. MORRIS et W.J. FITZGERALD : Interpolation of Missing Data in Image Sequences. *IEEE Transaction on Image Processing*, 11(4):1496–1508, 1995.
- [24] T. KOLLER, G. GERIG, G. SZÉKELY et D. DETTWILER : Multiscale Detection of Curvilinear Structures in 2D and 3D Image Data. Dans *IEEE International Conference on Computer Vision (ICCV’95)*, pages 864–869, 1995.
- [25] G.F. LAWLER, O. SCHRAM et W. WERNER : On the Scaling Limit of Planar Self-avoiding Walk. math.PR/0204277, 2002. à paraître dans

- Fractal Geometry and Applications : A jubilee of Benoit Mandelbrot, AMS Proc. Symp. Pure Math.
- [26] D. MARR : *Vision*. Freeman and Co., 1982.
- [27] S. MASNOU et J.-M. MOREL : Level-Lines Based Disocclusion. Dans *IEEE International Conference on Image Processing (ICIP'98)*, 1998.
- [28] W. METZGER : *Gesetze des Sehens*. Waldemar Kramer, 1975.
- [29] Y. MEYER : *Oscillating Patterns in Image Processing and in Some Non-linear Evolution Equations*. The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures, 12 mai 2001.
- [30] M. NITZBERG, D. MUMFORD et T. SHIOTA : *Filtering, Segmentation and Depth*, volume 662 de *Lecture Notes in Computer Science (LNCS)*. Springer-Verlag, Berlin, 1993.
- [31] K.S. PEDERSEN et A.B. LEE : Toward a Full Probability Model of Edges in Natural Images. Dans *European Conference on Computer Vision (ECCV'02)*, numéro 2350 dans *Lecture Notes in Computer Science (LNCS)*, page 328 et suivantes. Springer-Verlag, 2002.
- [32] P. PERONA et J. MALIK : Scale-Space and Edge Detection Using Anisotropic Diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.
- [33] C.E. SHANNON : A Mathematical Theory of Communication. *Bell System Technical Journal*, XXVII :3:379–423, 1948.
- [34] C. STEGER : An Unbiased Detector of Curvilinear Structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):113–125, février 1998.
- [35] R. STOICA, X. DESCOMBES et J. ZERUBIA : A Markov Point Process for Road Extraction in Remote Sensed Images. Rapport de recherche 3923, INRIA Sophia Antipolis, avril 2000.
- [36] J. SUBIRANA-VILANOVA et K. SUNG : Multi-Scale Vector-Ridge Detection for Perceptual Organization Without Edges. Rapport technique 1318, A.I. Memo, décembre 1992.
- [37] K. TREASH et K. AMARATUNGA : Automatic Road Detection in Grayscale Aerial Images. *Journal of Computing in Civil Engineering*, 14(1):60–69, janvier 2000.
- [38] L. VESE et S. OSHER : Modelling Textures with Total Variation Minimization and Oscillating Patterns in Image Processing. *Journal of Scientific Computing*, 19(1-3):553–572, mai 2003.
- [39] J. WEICKERT : A Review of Nonlinear Diffusion Filtering. *Scale-Space Theory in Computer Vision*, 1252:3–28, 1997.
- [40] M. WERTHEIMER : Untersuchungen zur Lehre der Gestalt. *Psychologische Forschung*, 4(301–350), 1923.

BIBLIOGRAPHIE

- [41] S.-C. ZHU : Embedding Gestalt Laws in Markov Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1170–1187, novembre 1999.

RÉSUMÉ :

L'école Gestaltiste s'intéresse à la vision, leur point de vue est que nous regroupons des points lumineux et/ou des objets selon certaines règles pour former des objets plus gros, des Gestalts.

La première partie de cette thèse est consacrée à la constance de largeur. La Gestalt constance de largeur regroupe des points situés entre deux bords qui restent parallèles. Nous cherchons donc dans les images des courbes "parallèles." Nous voulons faire une détection *a contrario*, nous proposons donc une quantification du "non parallélisme" de deux courbes par trois méthodes. La première méthode utilise un modèle de génération de courbes régulières et nous calculons une probabilité. La deuxième méthode est une méthode de simulation de type Monte-Carlo pour estimer cette probabilité. Enfin la troisième méthode correspond à un développement limité de la première en faisant tendre un paramètre vers 0 sous certaines contraintes. Ceci conduit à une équation aux dérivées partielles (EDP). Parmi ces trois méthodes la méthode de type Monte-Carlo est plus robuste et plus rapide.

L'EDP obtenue est très similaire à celles utilisées pour la désocclusion d'images. C'est pourquoi dans la deuxième partie de cette thèse nous nous intéressons au problème de la désocclusion. Nous présentons les méthodes existantes puis une nouvelle méthode basée sur un système de deux EDPs dont l'une est inspirée de celle de la première partie. Nous introduisons la probabilité de l'orientation du gradient de l'image. Nous prenons ainsi en compte l'incertitude sur l'orientation calculée du gradient de l'image. Cette incertitude est quantifiée en relation avec la norme du gradient.

Avec la quantification du non parallélisme de deux courbes, l'étape suivante est la détection de la constance de largeur dans les images. Il faut alors définir un seuil pour sélectionner les bonnes réponses du détecteur et surtout parmi les réponses définir des réponses "maximales." Le système d'EDPs pour la désocclusion dépend de beaucoup de paramètres, il faut trouver une méthode de calibration des paramètres pour obtenir de bons résultats adaptés à chaque image.

MOTS-CLÉS : analyse d'images, modèle *a contrario*, EDP, détection de structure, génération de courbes, probabilité géométrique, méthode de Monte-Carlo, Gestalts, désocclusion d'images, équation de transport-diffusion, stabilité et cohérence de schéma.

ABSTRACT :

Gestaltists study vision, they think that we put objects together using rules to make bigger objects, Gestalts.

The first part of this thesis deals with the constant width Gestalt, it puts together points between parallel borders. We seek in images “parallel” curves. We use an a contrario model : we therefore introduce a quantification of the “no parallelism” of two curves in three ways. First we compute a probability using a model to generate regular curves. Then we estimate this probability by a Monte-Carlo method. Finally a Taylor expansion in the first computation leads to a partial differential equation (PDE). The Monte-Carlo method is fastest and most robust of the three.

Our PDE is very similar to the PDE used in image disocclusion, thus the second part is about image disocclusion. We talk of the existing methods and then present a new one based on two PDEs. We introduce the probability of the image gradient orientation. We take into account the uncertainty upon the orientation due to its computation. This uncertainty is quantified with respect to the gradient norm.

The futur developments of this work are for the first part to use the computed probability to detect constant width, and for the second one to tune the parameters in order to have good results on natural images.

KEYWORDS : image analysis, *a contrario* model, PDE, feature detection, curve generator, geometric probability, Monte-Carlo method, Gestalts, image disocclusion, transport-diffusion equation, scheme stability and consistency.