

# Techniques de débruitage d'images

Humbert Florent

23 janvier 2008

# Table des matières

<b>1</b>	<b>Avant-propos</b>	<b>3</b>
1.1	Introduction . . . . .	3
1.2	Remerciements . . . . .	3
<b>2</b>	<b>Théorie</b>	<b>4</b>
2.1	Introduction . . . . .	4
2.2	Simplification . . . . .	5
2.2.1	Théorème fondamental (Équation d'Euler-Lagrange) . . .	5
2.2.2	Preuve . . . . .	5
2.3	Réécriture . . . . .	5
2.3.1	Théorème . . . . .	6
2.3.2	Preuve . . . . .	6
2.4	Résolution de l'équation . . . . .	6
2.5	Résolution dans un espace discret . . . . .	6
2.5.1	Discrétisation selon $t$ . . . . .	6
2.5.2	Discrétisation selon $(x, y)$ . . . . .	7
2.6	Conclusion . . . . .	10
<b>3</b>	<b>Application</b>	<b>11</b>
3.1	Type abstrait image . . . . .	11
3.1.1	Opérations sur le type . . . . .	11
3.1.2	Extension de l'accessor <code>imageLire</code> . . . . .	12
3.2	Opérateurs différentiel . . . . .	12
3.2.1	Dérivée première . . . . .	12
3.2.2	Dérivée seconde . . . . .	12
3.2.3	Dérivée directionnelle . . . . .	13
3.3	Présentation de l'image . . . . .	13
3.4	Méthode de Tikhonov . . . . .	13
3.4.1	Théorie . . . . .	13
3.4.2	Algorithme . . . . .	14
3.4.3	Application . . . . .	15
3.5	Méthode de la variation totale . . . . .	15
3.5.1	Théorie . . . . .	15
3.5.2	Algorithme . . . . .	16
3.5.3	Application . . . . .	16
3.6	Méthode des hypersurfaces . . . . .	16
3.6.1	Théorie . . . . .	17
3.6.2	Algorithme . . . . .	17

---

3.6.3	Application . . . . .	18
<b>4</b>	<b>Conclusion</b>	<b>19</b>
<b>5</b>	<b>Annexe</b>	<b>20</b>
5.1	Notations . . . . .	20
5.2	Ensembles . . . . .	20
5.2.1	Fonctions . . . . .	20
5.2.2	Norme . . . . .	21
5.3	ISNR . . . . .	21
5.4	Théorème . . . . .	22
5.5	Preuve . . . . .	22
5.5.1	Calcul de $D_{nn}(u)$ et $D_{n_{\perp}n_{\perp}}(u)$ . . . . .	22
5.5.2	Preuve du théorème fondamental . . . . .	23
5.5.3	Preuve du passage en dérivée directionnelle . . . . .	25

# Chapitre 1

## Avant-propos

### 1.1 Introduction

Un problème intéressant en traitement numérique d'image est la restauration d'images dégradées. Il arrive souvent lors de l'acquisition d'une image (par photographie notamment) que l'image obtenue soit différente de l'image espérée. En effet, certains phénomènes peuvent apparaître. Notamment de bruit (par exemple lors d'une mauvaise réception de données), de flou (dus à une mauvaise mise au point), ou encore des pertes de qualité (dus à une mauvaise luminosité).

Les dégradations que subissent les images sont en général dans les catégories que l'on vient d'énoncer.

Notre problème consistera donc à récupérer une image proche de l'image originale à partir d'une image de mauvaise qualité. Dans ce cours, nous ne nous focaliserons que sur le problème du bruit.

Au début, les premières méthodes de débruitage consistaient à utiliser un filtre passe-bas (suppression des hautes fréquences) ce qui avait pour inconvénient d'atténuer les contours de l'image. Pour parer à ce problème, des techniques nouvelles plus performantes ont vu le jour, notamment des méthodes utilisant des diffusions anisotropes (qui dépendent de la direction) utilisant des équations aux dérivées partielles.

Nous allons partir d'une « simple » équation. Puis nous baserons notre théorie autour de celle-ci. Elle nous permettra d'en tirer plusieurs méthodes de restaurations que nous appliquerons.

### 1.2 Remerciements

Je tiens à remercier Miles pour les corrections qu'il a su apporter.

# Chapitre 2

## Théorie

Dans cette partie, nous allons mettre en place les outils nécessaires à un certain type de restauration par équations aux dérivées partielles.

Le type d'équations aux dérivées partielles que nous allons voir a été introduit en traitement numérique d'image il y a une quinzaine d'année, les premières étaient basées sur les équations physiques de la diffusion de la chaleur (méthode de Malik et Perona), puis d'autres, plus générales, sont venues.

Je présenterai une classe « simple » et performante de ces équations basées sur une méthode variationnelle.

L'ensemble des notations est disponible en annexe section 5.1 page 20

### 2.1 Introduction

Dans l'ensemble du chapitre, nous utiliserons toujours certaines notations.

- $\Omega$  désignera un ouvert de  $\mathbb{R}^2$ , tel que  $\partial\Omega$  soit une courbe fermée de classe  $C^1$  par morceaux.
- $\alpha$  désignera un réel strictement positif.
- $f_r, b, f$  et  $f_o$  désigneront des applications de l'ouvert  $\Omega$  dans  $\mathbb{R}$  de classe  $C^2$  bornées et à variations bornées dans  $\Omega$ , c'est à dire  $(f_r, b, f, f_o) \in BV(\Omega)_2^4$ . De plus,  $\nabla(f)$  aura une limite sur  $\partial\Omega$  qui vaudra 0. On pourra donc éventuellement prolonger par continuité  $\nabla(f)$  sur la frontière.
- $H$  désignera une application de  $\mathbb{R}$  dans  $\mathbb{R}$  de classe  $C^2$ .

Soit  $f_o$  l'image observée et  $f_r$  l'image réelle non dégradée. Afin de chercher l'image réelle  $f_r$  à partir de  $f_o$ , nous allons chercher à minimiser l'expression suivante :

$$\Psi(f) = \iint_{\Omega} (f(x, y) - f_o(x, y))^2 dx dy + \alpha \iint_{\Omega} H(\|\nabla(f)(x, y)\|^2) dx dy \quad (2.1)$$

Que l'on peut noter :

$$\begin{aligned}\Psi(f) &= \iint_{\Omega} (f - f_o)^2 + \alpha \iint_{\Omega} H(\|\nabla(f)\|^2) \\ &= \|f - f_o\|_2 + \alpha \iint_{\Omega} H(\|\nabla(f)\|^2)\end{aligned}$$

Il faut bien noter que la norme  $\|\cdot\|$  n'est pas une norme de fonction mais la norme euclidienne.

La solution qui minimise cette expression s'appelle en général la solution régularisée. La première partie de l'expression est bien définie car  $f$  et  $f_o$  sont bornées, la deuxième partie est quant à elle bien définie car  $H$  est continue et  $f$  est à variation bornée.

**Interprétations** Dans cette expression, le premier terme sert à rapprocher la solution de l'image observée et le deuxième terme est un terme de régularisation qui est différent selon la valeur donnée à  $H$ . Lorsque l'image est bruitée, la valeur  $\|\nabla(f)\|$  aura tendance à avoir des pics au point où il y a du bruit, ce terme sert ainsi à réduire ces pics. De plus, l'utilisation de ce terme permet de réaliser un processus anisotrope (qui dépend de la direction et de la valeur du gradient), ce qui est important pour permettre de garder les contours mais d'éliminer le bruit.

Dans la littérature classique dans ce domaine, on ne fait en général pas intervenir le carré de la norme du gradient, mais simplement sa norme. Mais ceci peut poser de très gros problèmes de division par 0 que beaucoup ne semblent pas remarquer.

## 2.2 Simplification

### 2.2.1 Théorème fondamental (Équation d'Euler-Lagrange)

Afin d'utiliser la propriété précédente pour reconstituer l'image, nous allons utiliser un théorème très important permettant d'« éliminer » l'intégrale double.

**Théorème 1** Si  $\forall(x, y) \in \partial\Omega, \frac{\partial f}{\partial x}(x, y) = \frac{\partial f}{\partial y}(x, y) = 0$ . Alors,  $f$  minimise l'équation 2.1 si et seulement si  $f$  vérifie l'équation :

$$\alpha \operatorname{div} \left( H'(\|\nabla(f)\|^2) \cdot \nabla(f) \right) - 2 \cdot (f - f_o) = 0 \quad (2.2)$$

### 2.2.2 Preuve

Afin d'alléger la partie théorie, la preuve du théorème est accessible en annexe page 23 sous section 5.5.2.

## 2.3 Réécriture

Afin de programmer une méthode de restauration, on réécrit souvent d'une autre manière l'équation 5.1.

### 2.3.1 Théorème

**Théorème 2** *On suppose que  $\nabla(f)$  ne s'annule en aucun point (sauf sur la frontière  $\partial f$ , mais la fonction n'est a priori pas définie dans cet ensemble).*

*On peut donc poser  $n = \frac{\nabla(f)}{\|\nabla(f)\|}$  et  $n_\perp$  son vecteur orthogonal.*

*En posant  $\forall x \in \mathbb{R}, \phi(x) = H(x^2)$ , on dispose du théorème suivant :*

$$\begin{aligned} 2 \cdot \operatorname{div} \left( H'(\|\nabla(f)\|^2) \cdot \nabla(f) \right) &= \operatorname{div} \left( \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} \cdot \nabla(f) \right) \\ &= \phi''(\|\nabla(f)\|) \cdot \frac{\partial^2 f}{\partial n^2} + \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} \cdot \frac{\partial^2 f}{\partial n_\perp^2} \end{aligned}$$

On pose souvent :

$$\frac{1}{2} \phi''(\|\nabla(f)\|) = c_n$$

Et

$$\frac{1}{2} \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} = c_{n_\perp}$$

Ce qui permet d'écrire :

$$\alpha \left( c_n \frac{\partial^2 f}{\partial n^2} + c_{n_\perp} \frac{\partial^2 f}{\partial n_\perp^2} \right) - 2(f - f_o) = 0$$

### 2.3.2 Preuve

La preuve est accessible en annexe page 25 sous section 5.5.3.

## 2.4 Résolution de l'équation

Nous allons proposer une méthode permettant de résoudre l'équation :

$$2(f - f_o) - \alpha \operatorname{div} \left( H'(\|\nabla(f)\|^2) \cdot \nabla(f) \right) = 0$$

La méthode classique de résolution de cette équation et de poser l'équation aux dérivées partielles suivantes pour une application  $u$  définie de  $\Omega \times \mathbb{R}^+$  dans  $\mathbb{R}$ .

$$\begin{aligned} \forall (x, y) \in \Omega, \forall t > 0 \quad , \quad \frac{\partial u}{\partial t}(x, y, t) &= \Psi(u) = \alpha \operatorname{div} \left( H'(\|\nabla(u)\|^2) \cdot \nabla(u) \right) - 2(u - f_o) \\ \forall (x, y) \in \Omega \quad , \quad u(x, y, 0) &= f_o(x, y) \end{aligned}$$

Le but étant de chercher un état stationnaire à cette équation.

## 2.5 Résolution dans un espace discret

### 2.5.1 Discrétisation selon $t$

Nous n'allons pas résoudre directement cette équation mais nous allons chercher une approximation de la solution. Nous allons proposer une méthode basique d'approximation de solution d'équation différentielle qui consiste à discrétiser le domaine de  $t$  par un pas  $\Delta_t$  et à approximer  $\forall i, u(x, t, i \cdot \Delta_t)$ .

On détermine  $u(x, y, (i + 1) \cdot \Delta_t)$  en fonction de  $u(x, y, i \cdot \Delta_t)$  en utilisant la relation :

$$u(x, y, (i + 1) \cdot \Delta_t) = u(x, y, i \cdot \Delta_t) + \Delta_t \cdot \Psi(u)$$

On ne connaît  $u$  que sur certains points selon  $t$ , mais comme on le connaît sur tout  $\Omega$ , *a priori*,  $\Psi(u)$  est bien définie.

On peut faire une remarque importante sur cette relation. On peut reconnaître une suite récurrente définie comme suit :

$v$  est une suite de  $\mathbb{N}$  dans  $BV(\Omega)$ .

$F$  est une fonction définie de l'espace  $BV(\Omega)$  dans ce même espace par :

$$\forall g \in BV(\Omega), F(g) = g + \Delta_t \Psi(g)$$

L'équation de récurrence est donc :

$$\begin{aligned} v_{n+1} &= F(v_n) \\ v_0 &= f_0 \end{aligned}$$

Ainsi, si  $v$  converge, alors  $F$  admet un point fixe qui sera solution de l'équation. En effet, l'ensemble  $BV(\Omega)$  est un espace topologique normé complet car ses éléments sont bornés et donc le théorème du point fixe peut s'y appliquer.

On peut montrer que si  $H$  admet certaines propriétés, alors  $v$  convergera.

### 2.5.2 Discrétisation selon $(x, y)$

En pratique, on ne connaîtra  $f$  que sur certains points (à cause de la numérisation de l'image). C'est pourquoi il faut pouvoir définir une approximation de  $\Psi(u)$ .

Dans le cas où  $\nabla(f)$  ne s'annule pas. On peut écrire  $\Psi(u)$  sous la forme :

$$\Psi(u) = \alpha \left( c_n \frac{\partial^2 u}{\partial n^2} + c_{n_\perp} \frac{\partial^2 u}{\partial n_\perp^2} \right) - 2(u - f_0) = 0$$

#### Première méthode

Nous cherchons à approcher les fonctions  $\frac{\partial^2 u}{\partial n^2}$  et de  $\frac{\partial^2 u}{\partial n_\perp^2}$ .

Pour cela, la première méthode consiste à discrétiser  $\frac{\partial^2 u}{\partial x^2}$ ,  $\frac{\partial u}{\partial y}$ ,  $\frac{\partial^2 u}{\partial y^2}$  et  $\frac{\partial^2 u}{\partial x \partial y}$  selon un pas  $h$  et à appliquer la relation entre ces dérivées partielles.

**Notation 1** On discrétise  $\frac{\partial u}{\partial x}$  par

$$D_x(u)(x, y) = \frac{1}{2h} (u(x + h, y) - u(x - h, y))$$

La précision est en  $o(h^2)$ .

**Notation 2** On discrétise  $\frac{\partial u}{\partial y}$  par

$$D_y(u)(x, y) = \frac{1}{2h} (u(x, y + h) - u(x, y - h))$$



**Notation 3** On discrétise  $\frac{\partial^2 f}{\partial x^2}$  par

$$D_{xx}(f)(x, y) = \frac{1}{h^2}(f(x+h, y) - 2f(x, y) + f(x-h, y))$$

Qui est en  $o(h^2)$ .

**Notation 4** On discrétise  $\frac{\partial^2 f}{\partial y^2}$  par

$$D_{yy}(f)(x, y) = \frac{1}{h^2}(f(x, y+h) - 2f(x, y) + f(x, y-h))$$

**Notation 5** Et finalement  $\frac{\partial^2 f}{\partial x \partial y}$  par :

$$D_{xy}(f)(x, y) = \frac{1}{h^2}(f(x+h, y+h) - f(x+h, y) - f(x, y+h) + f(x, y))$$

Qui est en  $o(h)$  où bien par :

$$D_{xy}(f)(x, y) = \frac{1}{4h^2}(f(x+h, y+h) - f(x+h, y-h) - f(x-h, y+h) + f(x-h, y-h))$$

Qui est en  $o(h^2)$ .

Ainsi, on peut écrire  $\Psi(u)$  :

$$\Psi(u)(x, y, t) = c_n(x, y) \cdot D_{nn}(u)(x, y, t) + c_{n_\perp}(x, y) \cdot D_{n_\perp n_\perp}(u)(x, y, t)$$

Où :

$$D_{nn}(u) = \frac{1}{\|\nabla(u)\|^2} (D_x(u)^2 \cdot D_{xx}(u) + 2D_x(u)D_y(u)D_{xy}(u) + D_y(u)^2 D_{yy}(u))$$

$$D_{n_\perp n_\perp}(u) = \frac{1}{\|\nabla(u)\|^2} (D_x(u)^2 \cdot D_{xx}(u) - 2D_x(u)D_y(u)D_{xy}(u) + D_y(u)^2 D_{yy}(u))$$

Et

$$\|\nabla(u)\| = \sqrt{D_x(u)^2 + D_y(u)^2}$$

Une preuve de cette propriété est disponible en annexe 22 section 5.5.1.

### Deuxième méthode

Cette méthode a été utilisée dans le code source. Elle repose sur une approximation utilisée par Sapiro et Tannenbaum.

**Détermination de  $D_{nn}(u)$**  Nous approchons  $D_{nn}(u)$  par un pas  $h$  en utilisant la formule suivante :

$$\begin{aligned} D_{nn}(u)(x, y) &= \lambda_4 \cdot u(x-h, y-h) + \lambda_2 \cdot u(x, y-h) \\ &+ \lambda_3 \cdot u(x+h, y-h) + \lambda_1 \cdot u(x-h, y) \\ &- 4\lambda_0 \cdot u(x, y) + \lambda_1 \cdot u(x+h, y) \\ &+ \lambda_3 \cdot u(x-h, y+h) + \lambda_2 \cdot u(x, y+h) \\ &+ \lambda_4 \cdot u(x+h, y+h) \end{aligned}$$

Avec au point  $(x, y)$  :

$$\begin{aligned}\lambda_0 &= 1/2 \\ \lambda_1 &= 2\lambda_0 - \left( \frac{D_y(u)}{\|\nabla(u)\|} \right)^2 \\ \lambda_2 &= 2\lambda_0 - \left( \frac{D_x(u)}{\|\nabla(u)\|} \right)^2 \\ \lambda_3 &= \frac{1}{2} \left( \frac{D_x(u)}{\|\nabla(u)\|} \frac{D_y(u)}{\|\nabla(u)\|} + 1 - 2\lambda_0 \right) \\ \lambda_4 &= \frac{1}{2} \left( 1 - 2\lambda_0 - \frac{D_x(u)}{\|\nabla(u)\|} \frac{D_y(u)}{\|\nabla(u)\|} \right)\end{aligned}$$

Il y a en fait plusieurs possibilités de choix de  $\lambda_0$ , j'ai choisi la valeur proposée par Sapiro et Tannenbaum,  $\lambda_0 = 1/2$ .

**Détermination de  $D_{n_\perp n_\perp}(u)$**  Nous approchons cette dérivée directionnelle par la même méthode que précédemment.

$$\begin{aligned}D_{n_\perp n_\perp}(u)(x, y) &= \lambda_4 \cdot u(x-h, y-h) + \lambda_2 \dot{u}(x, y-h) \\ &\quad + \lambda_3 \cdot u(x+h, y-h) + \lambda_1 \dot{u}(x-h, y) \\ &\quad - 4\lambda_0 \cdot u(x, y) + \lambda_1 \dot{u}(x+h, y) \\ &\quad + \lambda_3 \cdot u(x-h, y+h) + \lambda_2 \dot{u}(x, y+h) \\ &\quad + \lambda_4 \cdot u(x+h, y+h)\end{aligned}$$

Avec au point  $(x, y)$  :

$$\begin{aligned}\lambda_0 &= 1/2 \\ \lambda_1 &= 2\lambda_0 - \left( \frac{D_x(u)}{\|\nabla(u)\|} \right)^2 \\ \lambda_2 &= 2\lambda_0 - \left( \frac{D_y(u)}{\|\nabla(u)\|} \right)^2 \\ \lambda_3 &= \frac{1}{2} \left( -\frac{D_x(u)}{\|\nabla(u)\|} \frac{D_y(u)}{\|\nabla(u)\|} + 1 - 2\lambda_0 \right) \\ \lambda_4 &= \frac{1}{2} \left( 1 - 2\lambda_0 + \frac{D_x(u)}{\|\nabla(u)\|} \frac{D_y(u)}{\|\nabla(u)\|} \right)\end{aligned}$$

Nous avons choisi le même  $\lambda_0$  que précédemment.

**Écriture de  $\Psi$**  On connaît à présent  $D_{nn}$  et  $D_{n_\perp n_\perp}$ , on peut donc écrire  $\Psi(u)$  sous la forme :

$$\Psi(u)(x, y, t) = c_n(x, y) \cdot D_{nn}(u)(x, y, t) + c_{n_\perp}(x, y) \cdot D_{n_\perp n_\perp}(u)(x, y, t)$$

## 2.6 Conclusion

On dispose à présent d'une manière de résoudre de manière plus ou moins parfaite l'équation initiale. Il faudra de nombreuses itérations dans le calcul de  $u$  pour approcher  $u$  à l'infini selon  $t$ . Mais en règle générale,  $u$  converge assez rapidement. Il faudra rarement plus de 500 itérations.

Pour récapituler, on choisit une fonction  $\phi$ , puis on pose :

$$\Psi(u)(x, y, t) = c_n(x, y) \cdot D_{nn}(u)(x, y, t) + c_{n_\perp}(x, y) \cdot D_{n_\perp n_\perp}(u)(x, y, t)$$

Avec :

$$c_n = \frac{1}{2} \phi''(\|\nabla(f)\|)$$

Et

$$c_{n_\perp} = \frac{1}{2} \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|}$$

On calcule de manière itérative :

$$\begin{aligned} v_{n+1} &= v_n + \Delta_t \Psi(v_n) \\ v_0 &= f_0 \end{aligned}$$

Où  $f_0$  est l'image originale.

## Chapitre 3

# Application

Dans cette partie, nous allons utiliser les théorèmes démontrés dans la partie théorique pour permettre la restauration d'image bruitée. Chaque section définira une méthode de restauration qui sera découpée en partie théorie, partie algorithmique et partie exemples d'applications et critiques de la méthode.

### 3.1 Type abstrait image

Afin de comprendre les algorithmes dans les sections suivantes. Nous allons définir un type abstrait de données `ImageMono`, qui correspondra à une image mono couleur (avec un seul canal).

#### 3.1.1 Opérations sur le type

Nous listons l'ensemble des opérations sur le type.

##### Constructeur

`imageCreer(Entier largeur, Entier hauteur) -> ImageMono` créera une image d'une certaine largeur et d'une certaine hauteur.

Parfois, on utilisera la notation `image1 = image2` pour copier toute l'image 2 dans l'image 1.

##### Accesseurs

`imageLire(Image i, Entier x, Entier y) -> Réel` permettra de lire une couleur de type `réel` au position  $(x, y)$ . Cette opération est définie *a priori* uniquement pour  $x$  entier entre 0 et `largeur - 1`, et pour  $y$  entier entre 0 et `longueur - 1`.

Mais en pratique, on étend souvent cette fonction pour permettre de calculer plus simplement les dérivées partielles.

`imageEcrire(Image i, Entier x, Entier y, Réel couleur)` permettra d'écrire une couleur au position  $(x, y)$ .

`imageLargeur(Image i)` -> Entier permettra de déterminer la largeur d'une image.

`imageHauteur(Image i)` -> Entier permettra de déterminer la hauteur d'une image.

Si l'image est mono couleur, lors du chargement de l'image, les couleurs seront *a priori* bornées (par exemple par 0 et 255). Mais pour les calculs, il vaut mieux ne pas borner les résultats. C'est pourquoi on travaille sur un type réel.

### 3.1.2 Extension de l'accessor `imageLire`

Pour permettre d'écrire plus facilement certains opérateurs (par exemple pour les différentielles), on étendra l'accessor `imageLire`.

Si  $m$  est la largeur de l'image et  $n$  sa longueur. On définit :

$$\forall (p, q) \notin [[0, m-1]] \times [[0, n-1]], \text{imageLire}(\text{image}, p, q) = \text{imageLire}(\text{image}, p', q')$$

Où le couple  $(p', q')$  est un élément de  $[[0, m-1, \infty]] \times [[0, n-1]]$  et minimise la distance  $\|(p - p', q - q')\|$ .

On peut aussi simplement l'étendre à 0, mais cela peut avoir de très mauvais effets si il y a beaucoup d'étapes dans les algorithmes.

## 3.2 Opérateurs différentiel

On définit les opérateurs différentiels suivant :

### 3.2.1 Dérivée première

Soit  $im$  une image.

On définit les dérivées premières au point  $(x, y)$  comme suit :

$$Dx(im, x, y) = \frac{1}{2}(\text{imageLire}(im, x+1, y) - \text{imageLire}(im, x-1, y))$$

$$Dy(im, x, y) = \frac{1}{2}(\text{imageLire}(im, x, y+1) - \text{imageLire}(im, x, y-1))$$

On définit également :

$$\text{NormeGrad}(im, x, y) = \sqrt{Dx(im, x, y)^2 + Dy(im, x, y)^2}$$

### 3.2.2 Dérivée seconde

On définit les dérivées secondes comme suit :

$$Dxx(im, x, y) = \text{imageLire}(im, x+1, y) - 2\text{imageLire}(im, x, y) + \text{imageLire}(im, x-1, y)$$

$$Dyy(im, x, y) = imageLire(im, x, y+1) - 2imageLire(im, x, y) + imageLire(im, x, y-1)$$

$$Dxy(im, x, y) = \frac{1}{4} \left( imageLire(im, x+1, y+1) - imageLire(im, x+1, y-1) \right. \\ \left. - imageLire(im, x-1, y+1) + imageLire(im, x-1, y-1) \right)$$

### 3.2.3 Dérivée directionnelle

Comme dans la partie théorie, il y a deux méthodes pour déterminer les dérivées directionnelles.

#### Première méthode

Si  $NormeGrad(im, x, y)$  vaut 0. Alors ces dérivées sont non définies. Sinon, en désignant  $im, x, y$  par  $u$  :

$$Dnn(im, x, y) = \frac{1}{NormeGrad(u)^2} (Dx(u)^2 \cdot Dxx(u) + 2Dx(u)Dy(u)Dxy(u) + Dy(u)^2Dyy(u))$$

$$Dntnt(im, x, y) = \frac{1}{NormeGrad(u)^2} (Dx(u)^2 \cdot Dxx(u) - 2Dx(u)Dy(u)Dxy(u) + Dy(u)^2Dyy(u))$$

#### Deuxième méthode

Vous pouvez voir directement la partie théorie page 8 concernant le calcul de ces dérivées directionnelles. Il suffira de remplacer  $h$  par 1.

## 3.3 Présentation de l'image

Dans ce chapitre, nous allons tenter de restaurer l'image bruitée 3.1

## 3.4 Méthode de Tikhonov

Nous allons présenter une première méthode de restauration proposée par Tikhonov en 1977.

### 3.4.1 Théorie

Tikhonov propose de prendre  $\forall x \in \mathbb{R}^2, H(x) = x$ . Dans la littérature classique, c'est :  $\phi(x) = x^2$  avec  $\phi(x) = H(x^2)$ .

On a donc :  $\forall x \in \mathbb{R}, H'(x) = 1$ .

Cette fonction est  $C^1$ , on peut donc appliquer le théorème fondamental.

Minimiser :

$$\Psi(f) = \iint_{\Omega} (f - f_o)^2 + \alpha \iint_{\Omega} \|\nabla(f)\|^2$$



FIG. 3.1 – Lenna bruitée

Revient à résoudre :

$$\alpha \cdot \operatorname{div}(\nabla(f)) - 2(f - f_o) = 0$$

Soit :

$$\alpha \Delta(f) - 2(f - f_o) = 0$$

Pour cela, on peut directement utiliser la méthode de résolution sans les dérivées directionnelles, c'est à dire sans faire l'hypothèse que  $\nabla(f)$  ne s'annule pas.

### 3.4.2 Algorithme

L'algorithme prend en entrée l'image bruitée, le nombre d'itération à effectuer, le paramètre  $\alpha$  et le taux de discrétisation  $dt$ .

```
tikhonov(Image im, Réel pos dt, Réel pos alpha, Entier pos iteration)
    -> Image
sortie = im
tempo = im
Pour n = 0 à iteration
    | Pour y = 0 à imageLongueur(im)-1
    | Pour x = 0 à imageLargeur(im)-1
    |   dxx = Dxx(sortie, x, y)
    |   dyy = Dyy(sortie, x, y)
    |   imageEcrire(tempo, x, y, dt * (alpha * (dxx + dyy))
    |           - 2(ImageLire(sortie, x, y) - imageLire(im, x, y))
    |           + imageLire(sortie, x, y))
    | sortie = tempo
Retourner sortie
```

À noter que dans cet algorithme, on prendra en général  $dt$  assez petit et  $alpha$  assez grand. Ainsi, le terme  $ImageLire(sortie, x, y) - imageLire(im, x, y)$  va souvent être négligeable. C'est pourquoi on peut écrire l'algorithme sous la forme suivante :

```

tikhonov(Image im, Réel pos dt, Réel pos alpha, Entier pos iteration) -> Image
  sortie = im
  tempo = im
  Pour n = 0 à iteration
    |Pour y = 0 à imageLongueur(im)-1
    | Pour x = 0 à imageLargeur(im)-1
    |   dxx = Dxx(sortie, x, y)
    |   dyy = Dyy(sortie, x, y)
    |   imageEcrire(tempo, x, y, dt * (alpha * (dxx + dyy)
    |     /* on retire le terme : - 2(ImageLire(sortie, x, y) - imageLire(im, x, y))*/
    |       + imageLire(sortie, x, y))
    |sortie = tempo
  Retourner sortie

```

### 3.4.3 Application

Avec le première algorithme, en utilisant :

- n=100
- epsilon = 0.01
- alpha = 5

On obtient la figure 3.2.



FIG. 3.2 – Correction par Tikhonov, ISNR = -0.97

On peut noter que cette méthode a tendance à rendre flou l'image. Les détails disparaissent.

Nous avons utilisé un indice pour quantifier la restauration, l'ISNR. Son calcul est détaillé en annexe page 21 section 5.3.

## 3.5 Méthode de la variation totale

Nous allons voir une méthode de meilleure qualité que la précédente.

### 3.5.1 Théorie

Cette méthode s'appuie sur une théorie physique. Nous considérons que le gradient de l'image ne s'annule pas. On choisit :  $H(x) = \sqrt{x}$ , ce qui revient au même que de prendre  $\phi(x) = x$ .



On a ainsi les coefficients suivants :

$$c_{nn} = 0$$

Et

$$c_{n_{\perp}b_{\perp}} = \frac{1}{2\|\nabla(f)\|}$$

Le problème est qu'en général, le gradient d'une image s'annule très souvent en certains points.

Mais, on peut remarquer que si le gradient s'annule, alors c'est que l'image est localement constante dans toutes les directions. Cela signifie qu'il n'y a localement pas de bruit. Il n'y a donc pas besoin de retoucher à l'image. Avec cette affirmation, on peut étendre la théorie à toute image en ne considérant que les points ayant un gradient supérieur à un  $\epsilon$  donné.

### 3.5.2 Algorithme

Nous pouvons à présent définir l'algorithme corrigeant l'image.

```

variation(Image im, Réel pos dt, Réel pos alpha, Entier pos iteration) -> Image
  sortie = im
  tempo = im
  Pour n = 0 à iteration
    | Pour y = 0 à imageLongueur(im)-1
    | Pour x = 0 à imageLargeur(im)-1
    |   grad = NormeGrad(tempo, x, y)
    |   Si grad > epsilon alors
    |     cnt = 1/(2 * grad)
    |     dnt = Dntnt(sortie, x, y)
    |     imageEcrire(tempo, x, y, dt * (alpha * (dnt * cnt)
    |                   - 2(ImageLire(sortie, x, y) - imageLire(im, x, y))
    |                   + imageLire(sortie, x, y))
    | sortie = tempo
  Retourner sortie

```

### 3.5.3 Application

En utilisant :

- n=400
- dt = 0.001
- alpha = 250
- epsilon = 0.1

On obtient la figure 3.3.

On peut remarquer que le résultat est bien meilleur qu'avec la méthode de Tikhonov. Malheureusement, il nécessite un grand nombre d'itérations, ce qui peut-être parfois lent.

## 3.6 Méthode des hypersurfaces

La méthode des hypersurfaces donne des résultats de très bonne qualité.



FIG. 3.3 – Correction par variation totale, ISNR = 2.02

### 3.6.1 Théorie

Nous choisissons une fonction  $H$  définie par :

$$H(x) = \sqrt{1+x}$$

C'est à dire une fonction  $\phi$  définie par :

$$\phi(x) = \sqrt{1+x^2}$$

On a ainsi les coefficients définies par :

$$c_{nn} = \frac{1}{2(1 + \|\nabla(f)\|^2)^{3/2}}$$

Et

$$c_{n_{\perp}b_{\perp}} = \frac{1}{2\sqrt{1 + \|\nabla(f)\|^2}}$$

De la même manière que précédemment, on peut en déduire une extension de la théorie à des images ayant un gradient qui s'annule en réalisant un test sur  $\|\nabla(f)\|$ .

### 3.6.2 Algorithme

On en déduit l'algorithme suivant :

```

hypersurfaces(Image im, Réel pos dt, Réel pos alpha, Entier pos iteration) -> Image
sortie = im
tempo = im
Pour n = 0 à iteration
| Pour y = 0 à imageLongueur(im)-1
| Pour x = 0 à imageLargeur(im)-1
|   grad = NormeGrad(im, x, y)
|   Si grad > epsilon alors
|     cnn = 1/(2 * sqrt(1+grad^2))
|     cnt = 1/(2 * (1+grad^2 * sqrt(1+grad^2)))
|     dnt = Dntnt(sortie, x, y)

```

```
|     dnn = Dnn(sortie, x, y)
|     imageEcrire(tempo, x, y, dt * (alpha * (dnt * cnt + cnn * dnn))
|         - 2(ImageLire(sortie, x, y) - imageLire(im, x, y))
|         + imageLire(sortie, x, y))
| sortie = tempo
Retourner sortie
```

### 3.6.3 Application

En utilisant les valeurs suivantes :

- n=200
- dt = 0.001
- alpha = 250
- epsilon = 0.1

On obtient la figure 3.4.



FIG. 3.4 – Correction par la méthode des hypersurfaces, ISNR = 2.005

Cette méthode donne des résultats surprenants et de bonne qualité. L'image semble meilleure qu'avec la méthode de la variation totale. Cependant, sur ce jeu de test, l'indice ISNR montre que la méthode de la variation totale est légèrement plus performante (2.02 au lieu de 2.005).

## Chapitre 4

# Conclusion

Nous avons présenté une série d'opération permettant le débruitage d'images. La dernière méthode proposée, celle des hypersurfaces, semble être celle qui donne les meilleurs résultats.

Il faut noter que l'on peut étendre la théorie à de nombreuses autres fonctions  $H$  que je n'ai pas détaillées dans ce cours.

J'espère ainsi que vous avez pu voir en quoi les équations différentielles peuvent être utiles en traitement numérique de l'image.

# Chapitre 5

## Annexe

### 5.1 Notations

Voici une petite liste de notations (assez classiques) utilisées dans ce cours.

### 5.2 Ensembles

**Notation 6 (Ensemble de fonctions)** On notera  $\mathcal{F}(E, F)$  l'ensemble des fonctions de l'ensemble  $E$  dans l'ensemble  $F$ .

**Notation 7 (Ensembles classiques)** Les ensembles  $\mathbb{C}$ ,  $\mathbb{R}$ ,  $\mathbb{N}$  et  $\mathbb{Z}$  désignent respectivement l'ensemble des complexes, des réels, des entiers naturels et des entiers relatifs.

L'ensemble  $\mathbb{K}$  désignera  $\mathbb{C}$  ou  $\mathbb{R}$ .

**Notation 8 (Ensemble à variation bornée)** Soit  $\Omega$  un ouvert de  $\mathbb{R}^2$  On désignera par  $BV(\Omega)_n$  l'ensemble des applications  $C^n$  ( $n > 0$ ) à variation bornée. L'ensemble  $BV(\Omega)$  désignera simplement  $BV(\Omega)_1$ .

**Notation 9 (Ensemble d'entier)** On note  $[[n, m]]$  l'ensemble des entiers compris entre  $n$  et  $m$ . (Donc  $[[n, m]] = [n, m] \cap \mathbb{Z}$ )

**Notation 10 (Frontière)** Soit  $F$  un ensemble fermé. On note  $\partial F$  la frontière de  $F$ .

On peut éventuellement étendre cette définition à des ouverts bornés.

#### 5.2.1 Fonctions

**Notation 11 (Gradient)** Soit  $f \in \mathcal{F}(\Omega, \mathbb{R})$  une application de classe  $C^1$  et où  $\Omega$  est un ouvert de  $\mathbb{R}^2$ , on note :  $\nabla(f)$  le gradient de  $f$  défini par :

$$\forall (x, y) \in \Omega, \nabla(f)(x, y) = \left( \frac{\partial f}{\partial x}(x, y), \frac{\partial f}{\partial y}(x, y) \right)$$

**Notation 12 (Divergence)** Soit  $f \in \mathcal{F}(\Omega, \mathbb{R}^2)$  une application de classe  $C^1$  et où  $\Omega$  est un ouvert de  $\mathbb{R}^2$ .  $f$  peut s'écrire  $f = (f_x, f_y)$  où  $f_x \in \mathcal{F}(\Omega, \mathbb{R})$  et  $f_y \in \mathcal{F}(\Omega, \mathbb{R})$ . On note :  $\operatorname{div}()$  l'opérateur divergence défini par :

$$\forall (x, y) \in \Omega, \operatorname{div}(f)(x, y) = \frac{\partial f_x}{\partial x}(x, y) + \frac{\partial f_y}{\partial y}(x, y)$$

**Notation 13 (Produit scalaire)** On notera  $\langle, \rangle$  le produit scalaire usuel entre deux vecteurs de  $\mathbb{R}^n$  défini par :

$$\forall u \in \mathbb{R}^n, \forall v \in \mathbb{R}^n, \langle u, v \rangle = \sum_{i=1}^n u_i \cdot v_i$$

**Notation 14 (Dérivée partielle)** Soit  $u$  un vecteur de  $\mathbb{R}^2$  non nul. Soit  $f$  une application de classe  $C^1$  de  $\mathbb{R}^2$  dans  $\mathbb{R}$ .

On note :  $\frac{\partial f}{\partial u}$  l'application définie par

$$\frac{\partial f}{\partial u} = \langle \nabla(f), u \rangle$$

**Notation 15 (Opérateur Laplacien)** Soit  $f$  une application  $C^2$  définie sur un ouvert  $\Omega$  de  $\mathbb{R}^2$  à valeurs dans  $\mathbb{R}^2$ , on note  $\Delta$  l'opérateur laplacien défini par :

$$\forall (x, y) \in \Omega, \Delta(f)(x, y) = \frac{\partial^2 f}{\partial x^2}(x, y) + \frac{\partial^2 f}{\partial y^2}(x, y)$$

### 5.2.2 Norme

**Notation 16 (Norme de vecteur)** Soit  $v$  un vecteur de  $\mathbb{R}^n$ .

On note :

$$\|v\| = \sqrt{\sum_{i=1}^n v_i^2}$$

**Notation 17 (Norme de fonction)** Dans le cas d'existence de cette expression, on définit la norme suivante pour une application  $f$  définie sur une partie  $U$  de  $\mathbb{R}^2$  à valeurs dans  $\mathbb{R}$ .

$$\|f\|_2 = \sqrt{\iint_U |f(x, y)|^2 dx dy}$$

## 5.3 ISNR

Afin de quantifier la qualité de la restauration pour une image bruitée, on utilise parfois une mesure.

Soit  $f$  l'image originale,  $g$  l'image modifiée et  $f'$  l'image restaurée. Alors l'ISNR<sup>1</sup> est définie par :

$$ISNR(f, f', g) = 10 \log_{10} \frac{\sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} (f(i, j) - g(i, j))^2}{\sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} (f(i, j) - f'(i, j))^2}$$

Si l'image corrigée est proche de l'image originale, alors le dénominateur va être proche de 0, et donc l'ISNR sera proche de l'infini (il faut noter qu'avec le logarithme en base 10, cette fonction augmente lentement).

<sup>1</sup>Improvement Signal to Noise Ratio

Si l'image corrigée est exactement identique à l'image bruitée, alors la fraction sera égale à 1, donc l'ISNR sera égale à 0.

Enfin, si l'image restaurée est vraiment différente de l'image originale, le dénominateur sera assez important, l'ISNR tendra vers moins l'infini.

On peut résumer :

- Si l'ISNR est négatif, la restauration est mauvaise;
- Si l'ISNR est nulle, alors il n'y a pas eu de restauration (on reste proche de l'image bruitée);
- Si l'ISNR est positif, alors la restauration est de bonne qualité. À noter que l'ISNR évolue lentement à cause du logarithme.

## 5.4 Théorème

**Théorème 3 (Green-Riemann)** Soient :

- $D$  une partie de  $\mathbb{R}^2$  limitée par une courbe paramétrée fermée ( $C$ ) de classe  $C^1$  par morceaux parcourue dans le sens direct.
- $\Omega$  un ouvert de  $\mathbb{R}^2$  tel que  $D \subset \Omega$
- $(P, Q) \in \mathcal{F}(\Omega, D)$  de classe  $C^1$ .

Alors :

$$\int_{(C)} P(x, y)dx + Q(x, y)dy = \iint_D \left( \frac{\partial Q}{\partial x}(x, y) - \frac{\partial P}{\partial y}(x, y) \right) dx dy$$

## 5.5 Preuve

### 5.5.1 Calcul de $D_{nn}(u)$ et $D_{n_{\perp}n_{\perp}}(u)$

On montre que :

$$D_{nn}(u)(x, y) = \frac{1}{\|\nabla(u)\|^2} (D_x(u)^2 \cdot D_{xx}(u) + 2D_x(u)D_y(u)D_{xy}(u) + D_y(u)^2 D_{yy}(u))$$

Rappelons déjà que :  $D_v(u) = \langle \nabla(u), v \rangle$  où  $v$  est un vecteur de  $\mathbb{R}^2$ .

Comme  $n = \nabla(u) / \|\nabla(u)\|$  et  $D_n(u) = \|\nabla(u)\|$

On en déduit que (afin de simplifier les notations, nous avons omis de noter  $(x, y)$ ) :

$$D_{nn}(u) = \langle \nabla(\|\nabla(u)\|), \frac{\nabla(u)}{\|\nabla(u)\|} \rangle$$

Or :

$$\begin{aligned} \nabla(\|\nabla(u)\|) &= \nabla \left( \left\| \sqrt{D_x(u)^2 + D_y(u)^2} \right\| \right) \\ &= \frac{1}{2\|\nabla(u)\|} (2D_{xx}(u)D_x(u) + D_x(D_y(u)^2), 2D_{yy}(u)D_y(u) + D_y(D_x(u)^2)) \\ &= \frac{1}{2\|\nabla(u)\|} (2D_x(u)^2 + 2D_{xy}(u)D_y(u), D_y(u)^2 + 2D_{xy}(u)D_x(u)) \end{aligned}$$

Ici, nous avons directement appliqué la propriété  $(\sqrt{(u)})' = \frac{u'}{2\sqrt{u}}$   
 En effectuant le produit scalaire avec  $\nabla(u) / \|\nabla(u)\| = (D_x(u), D_y(u)) / \|\nabla(u)\|$ .  
 On en déduit directement l'expression :

$$D_{nn}(u)(x, y) = \frac{1}{\|\nabla(u)\|^2} (D_x(u)^2 \cdot D_{xx}(u) + 2D_x(u)D_y(u)D_{xy}(u) + D_y(u)^2 D_{yy}(u))$$

Le calcul de  $D_{n_\perp n_\perp}(u)$  peut se faire de la même manière, ou bien se faire en utilisant la propriété :

$$\Delta(f) = D_{nn}(f) + D_{n_\perp n_\perp}(f)$$

(le laplacien d'une fonction est indépendant de la base dans lequel on l'exprime).

### 5.5.2 Preuve du théorème fondamental

On rappelle le théorème :

Si  $\forall(x, y) \in \partial\Omega, \frac{\partial f}{\partial x}(x, y) = \frac{\partial f}{\partial y}(x, y) = 0$ . Alors,  $f$  minimise l'équation 2.1 si et seulement si  $f$  vérifie l'équation :

$$\alpha \operatorname{div} \left( H'(\|\nabla(f)\|^2) \cdot \nabla(f) \right) - 2 \cdot (f - f_o) = 0 \quad (5.1)$$

Si  $f$  minimise l'équation 2.1, alors pour tout  $g \in BV(\Omega)_2$ , les applications  $\Psi_g$  de  $\mathbb{R}$  dans  $\mathbb{R}$  définies par  $\Psi_g(\lambda) = \Psi(f + \lambda g)$  vérifient :

$$\Psi'_g(0) = 0$$

Ainsi, comme toutes les fonctions  $f$ ,  $f_o$  et  $g$  sont dans  $BV(\Omega)$ , on peut permuter l'intégrale et la dérivée et écrire :

$$\begin{aligned} \Psi'_g(\lambda) &= \frac{\partial}{\partial \lambda} \iint_{\Omega} (f + \lambda g - f_o)^2 + \alpha \cdot \frac{\partial}{\partial \lambda} \iint_{\Omega} H(\|\nabla(f + \lambda g)\|^2) \\ &= \iint_{\Omega} \frac{\partial(f + \lambda g - f_o)^2}{\partial \lambda} + \alpha \iint_{\Omega} \frac{\partial H(\|\nabla(f + \lambda g)\|^2)}{\partial \lambda} \end{aligned}$$

Posons :

$$\psi_1(\lambda) = \iint_{\Omega} (f + \lambda g - f_o)^2$$

Et

$$\psi_2(\lambda) = \iint_{\Omega} H(\|\nabla(f + \lambda g)\|^2)$$

On a donc dans ce cas :

$$\Psi_g = \psi_1 + \alpha \psi_2$$

#### Calcul de la dérivée de $\psi_1$

Nous allons déterminer la valeur de la dérivée de  $\psi_1$  en 0.

$$\begin{aligned} \forall \lambda \in \mathbb{R}, \psi'_1(\lambda) &= \iint_{\Omega} \frac{\partial(f + \lambda g - f_o)^2}{\partial \lambda} \\ &= \iint_{\Omega} 2g(f + \lambda g - f_o) \end{aligned}$$



D'où :

$$\psi_1'(0) = \iint_{\Omega} 2g(f - f_0)$$

### Calcul de la dérivée de $\psi_2$

Le calcul et la simplification de la dérivée de  $\psi_2$  est plus complexe que pour  $\psi_1$ . Nous allons en effet utiliser certains théorèmes complexes, tel que le théorème de Green-Riemann.

$$\begin{aligned} \forall \lambda \in \mathbb{R}, \psi_2'(\lambda) &= \iint_{\Omega} \frac{\partial H(\|\nabla(f + \lambda g)\|^2)}{\partial \lambda}(\lambda) \\ &= \iint_{\Omega} \frac{\partial \|\nabla(f + \lambda g)\|^2}{\partial \lambda}(\lambda) \cdot H'(\|\nabla(f + \lambda g)\|^2) \end{aligned}$$

Or,

$$\|\nabla(f + \lambda g)\|^2 = \left(\frac{\partial f}{\partial x} + \lambda \frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y} + \lambda \frac{\partial g}{\partial y}\right)^2$$

D'où :

$$\frac{\partial \|\nabla(f + \lambda g)\|^2}{\partial \lambda}(\lambda) = \frac{\partial g}{\partial x} \left(\frac{\partial f}{\partial x} + \lambda \frac{\partial g}{\partial x}\right) + \frac{\partial g}{\partial y} \left(\frac{\partial f}{\partial y} + \lambda \frac{\partial g}{\partial y}\right)$$

Ainsi, en 0, on trouve :

$$\psi_2'(0) = \iint_{\Omega} \langle \nabla(g), \nabla(f) \rangle \cdot H'(\|\nabla(f)\|^2)$$

Nous allons réaliser une intégration par partie en utilisant les deux égalités suivantes :

$$\frac{\partial}{\partial x} \left( g \frac{\partial f}{\partial x} H'(\|\nabla(f)\|^2) \right) = \frac{\partial g}{\partial x} \frac{\partial f}{\partial x} H'(\|\nabla(f)\|^2) + g \frac{\partial^2 f}{\partial x^2} H'(\|\nabla(f)\|^2) + g \frac{\partial f}{\partial x} \frac{\partial}{\partial x} \left( H'(\|\nabla(f)\|^2) \right)$$

Et

$$\frac{\partial}{\partial y} \left( g \frac{\partial f}{\partial y} H'(\|\nabla(f)\|^2) \right) = \frac{\partial g}{\partial y} \frac{\partial f}{\partial y} H'(\|\nabla(f)\|^2) + g \frac{\partial^2 f}{\partial y^2} H'(\|\nabla(f)\|^2) + g \frac{\partial f}{\partial y} \frac{\partial}{\partial y} \left( H'(\|\nabla(f)\|^2) \right)$$

De plus, en utilisant la décomposition de la divergence :

$$\operatorname{div} \left( H'(\|\nabla(f)\|^2) \cdot \nabla(f) \right) = \frac{\partial}{\partial x} \left( H'(\|\nabla(f)\|^2) \cdot \frac{\partial f}{\partial x} \right) + \frac{\partial}{\partial y} \left( H'(\|\nabla(f)\|^2) \cdot \frac{\partial f}{\partial y} \right)$$

On en déduit l'égalité :

$$\begin{aligned} &\frac{\partial}{\partial x} \left( g \frac{\partial f}{\partial x} H'(\|\nabla(f)\|^2) \right) + \frac{\partial}{\partial y} \left( g \frac{\partial f}{\partial y} H'(\|\nabla(f)\|^2) \right) = \\ &\langle \nabla(f), \nabla(g) \rangle \cdot H'(\|\nabla(f)\|^2) + g \cdot \operatorname{div} \left( H'(\|\nabla(f)\|^2) \cdot \nabla(f) \right) \end{aligned}$$

D'où :

$$\psi_2'(0) = \iint_{\Omega} \frac{\partial}{\partial x} \left( g \frac{\partial f}{\partial x} H'(\|\nabla(f)\|^2) \right) + \frac{\partial}{\partial y} \left( g \frac{\partial f}{\partial y} H'(\|\nabla(f)\|^2) \right) - g \cdot \operatorname{div} \left( H'(\|\nabla(f)\|^2) \cdot \nabla(f) \right)$$

**Utilisation du théorème de Green-Riemann** Comme :

- $\Omega$  est un ouvert
- $\partial\Omega$  est une courbe fermée de classe  $C^1$ .
- $g$  est  $C^1$ ,  $\frac{\partial f}{\partial x}$  et  $\frac{\partial f}{\partial y}$  sont  $C^1$ ,  $\|\nabla(f)\|^2$  est également de classe  $C^1$
- $H$  est  $C^2$ .

On en déduit que l'on peut appliquer le théorème de Green-Riemann.

Par application de ce théorème, on obtient :

$$\iint_{\Omega} \frac{\partial}{\partial x} \left( g \frac{\partial f}{\partial x} H'(\|\nabla(f)\|^2) \right) + \frac{\partial}{\partial y} \left( g \frac{\partial f}{\partial y} H'(\|\nabla(f)\|^2) \right) dx dy =$$

$$\int_{\partial C} -g(x, y) \frac{\partial f}{\partial y}(x, y) H'(\|\nabla(f)(x, y)\|^2) dx + g(x, y) \frac{\partial f}{\partial x}(x, y) H'(\|\nabla(f)(x, y)\|^2) dy = 0$$

Ce terme est nul à cause des conditions des dérivées partielles de  $f$  sur les bords.

On en déduit donc ainsi :

$$\psi'_2(0) = \iint_{\Omega} -g \cdot \operatorname{div} \left( H'(\|\nabla(f)\|^2) \cdot \nabla(f) \right)$$

**Presque la fin**

On en déduit donc que :

$$0 = \Psi_g(0) = \iint_{\Omega} 2g(f - f_o) - \alpha \cdot g \cdot \operatorname{div} \left( H'(\|\nabla(f)\|^2) \cdot \nabla(f) \right)$$

Comme cette propriété est vraie pour tout  $g$  bornée et à variation bornée, on en déduit :

$$2(f - f_o) - \alpha \operatorname{div} \left( H'(\|\nabla(f)\|^2) \cdot \nabla(f) \right) = 0$$

La réciproque ne sera pas démontrée ici. On ne peut pas remonter directement les étapes car on pourra seulement montrer que  $f$  minimise  $\Psi$  localement.

### 5.5.3 Preuve du passage en dérivée directionnelle

Nous rappelons que nous cherchons à démontrer la propriété :

$$2 \cdot \operatorname{div} \left( H'(\|\nabla(f)\|^2) \cdot \nabla(f) \right) = \operatorname{div} \left( \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} \cdot \nabla(f) \right)$$

$$= \phi''(\|\nabla(f)\|) \cdot \frac{\partial^2 f}{\partial n^2} + \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} \cdot \frac{\partial^2 f}{\partial n_{\perp}^2}$$

Afin d'alléger les notations, nous allons noter :  $f_x = \frac{\partial f}{\partial x}$ ,  $f_y = \frac{\partial f}{\partial y}$ ,  $f_{xx} = \frac{\partial^2 f}{\partial x^2}$ ,  $f_{yy} = \frac{\partial^2 f}{\partial y^2}$  et  $f_{xy} = \frac{\partial^2 f}{\partial x \partial y}$ .

**Première égalité**

Nous allons montrer dans un premier temps la première égalité.  
Comme  $\forall x \in \mathbb{R}, \phi(x) = H(x^2)$ , on en déduit par dérivation que :

$$\forall x \in \mathbb{R}, \phi'(x) = 2xH'(x^2)$$

On peut remplacer  $x$  par  $\|\nabla(f)\|$ , ce qui donne :

$$\phi'(\|\nabla(f)\|) = 2\|\nabla(f)\|H'(\|\nabla(f)\|^2)$$

Comme  $\|\nabla(f)\|$  ne s'annule pas, on en déduit :

$$2H'(\|\nabla(f)\|^2) = \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|}$$

D'où :

$$2 \cdot \operatorname{div} \left( H'(\|\nabla(f)\|^2) \cdot \nabla(f) \right) = \operatorname{div} \left( \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} \cdot \nabla(f) \right)$$

**Deuxième égalité**

On décompose l'opération de divergence :

$$\begin{aligned} \operatorname{div} \left( \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} \cdot \nabla(f) \right) &= \frac{\partial}{\partial x} \left( \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} \right) \cdot f_x + \frac{\partial}{\partial y} \left( \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} \right) \cdot f_y \\ &\quad + \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} \cdot \Delta(f) \end{aligned}$$

Hors, on sait que :

$$\frac{\partial \phi'(\|\nabla(f)\|)}{\partial x} = \phi''(\|\nabla(f)\|) \cdot (f_x f_{xx} + f_y f_{xy})$$

On en déduit donc que :

$$\begin{aligned} \frac{\partial}{\partial x} \left( \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} \right) &= \frac{1}{\|\nabla(f)\|^2} \left( \frac{\partial \phi'(\|\nabla(f)\|)}{\partial x} \|\nabla(f)\| - \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} (f_x f_{xx} + f_y f_{xy}) \right) \\ &= \frac{1}{\|\nabla(f)\|^2} (\phi''(\|\nabla(f)\|) - \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|}) (f_x f_{xx} + f_y f_{xy}) \end{aligned}$$

Ainsi,

$$\begin{aligned} \operatorname{div} \left( \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} \cdot \nabla(f) \right) &= \frac{1}{\|\nabla(f)\|^2} (\phi''(\|\nabla(f)\|) - \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|}) \cdot \\ &\quad \left( (f_x)^2 f_{xx} + 2f_x f_y f_{xy} + (f_y)^2 f_{yy} \right) \\ &\quad + \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} \Delta(f) \end{aligned}$$

**Rappel 1 (Dérivée seconde directionnelle)** Nous rappelons que pour une application  $f$  de classe  $C^2$  :

$$\frac{\partial^2 f}{\partial n^2} = \frac{1}{\|\nabla(f)\|^2} \left( (f_x)^2 f_{xx} + 2f_x f_y f_{xy} + (f_y)^2 f_{yy} \right)$$

**Rappel 2** Nous rappelons également que pour une application  $f$  de classe  $C^2$  tel que  $\nabla(f)$  ne s'annule pas, l'égalité suivante :

$$\Delta(f) = \frac{\partial^2 f}{\partial n^2} + \frac{\partial^2 f}{\partial n_\perp^2}$$

En utilisant ces deux rappels, on peut en déduire ce que l'on cherchait.

$$\operatorname{div} \left( \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} \cdot \nabla(f) \right) = \phi''(\|\nabla(f)\|) \cdot \frac{\partial^2 f}{\partial n^2} + \frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} \cdot \frac{\partial^2 f}{\partial n_\perp^2}$$

# Table des figures

3.1	Lenna bruitée . . . . .	14
3.2	Correction par Tikhonov, ISNR = -0.97 . . . . .	15
3.3	Correction par variation totale, ISNR = 2.02 . . . . .	17
3.4	Correction par la méthode des hypersurfaces, ISNR = 2.005 . . . . .	18