



UNIVERSITE MOHAMED BOUDIAFDE M'SILA

Faculté des Mathématiques et de l'Informatique

Département de Mathématiques



MEMOIRE DE FIN D'ETUDE

Présenté pour l'obtention du Diplôme de **MASTER**

Domaine : Mathématiques et Informatique

Filière: Mathématiques

**Option : Equation aux dérivées partielles
et Application**

Par

KHACHAB Sarra

Sujet

Classes de solution pour le modèle de Perona-Malik unidimensionnel

Devant le jury :

Mr. A. MERZOUGUI	MCA. Univ de M'sila	Président
Mr. N. BENHAMIDOUCHE	Prof. Univ de M'sila	Encadreur
Mr. N. BOUNAB	MCB. Univ de M'sila	Examineur

Promotion : 2017 / 2018

Remerciements

Je tiens tout d'abord à remercier Dieu «**ALLAH**» le tout puissant et miséricordieux, qui j'a donné la force et la patience d'accomplir ce modeste travail.

En second lieu, je tiens à remercier mon encadreur **Prof : BENHAMIDOUCHE.N** pour ses efforts précieux, son aide, son encouragement continu et ses conseils durant toute la période du travail.

Mon vifs remerciements vont également aux membre du jury pour l'intérêt qu'ils ont porté à ma recherche en acceptant d'examiner mon travail et de l'enrichir par leurs propositions.

J'adresse aussi des remerciements spéciaux à **Mr : CHOUDER.R** et **Mr : BOUGHE-RARA.B**

J'adresse également mes sincères remerciements à tous les enseignants de la faculté des mathématiques et d' informatique.

Enfin, mes gratitudes les plus profonds vont à mes chers parents pour leurs soutien.précieux.

Merci à tous mes camarades, ainsi que tous ceux qui ont contribué, de près ou de loin à l'aboutissement de ce travail.

Table des matières

Introduction	1
1 EDP et traitement de l'image	3
1.1 Définition d'une image numérique	3
1.2 Types d'images numériques	4
1.3 Définition d'une image mathématique	6
1.4 Filtrage linéaire	6
1.4.1 Différents filtres linéaires	7
1.4.2 Equation de la chaleur et filtrage linéaire	10
1.5 Filtrage de diffusion non linéaire	11
1.5.1 Modèle de Perona-Malik 2-D	11
2 Le modèle de Perona-Malik unidimensionnel	15
2.1 L'analyse mathématique du modèle	15
2.1.1 Formulation de problème	15
2.1.2 Propriétés de la solution	16
2.1.3 Les paramètres du problème	21
2.1.4 Le problème non homogène	21
2.2 Les filtres souhaitables et leurs Propriétés	22
2.2.1 Propriété de préservation de bords	22
2.2.2 Propriété d'amélioration des bords	23
2.2.3 Construction des filtres	25
2.2.4 Particularités de la diffusivité :	25

3	Résolution numérique du modèle	32
3.1	Méthode des differences finies	32
3.2	Approximations par différences finies	33
3.2.1	Notations	33
3.2.2	Approximations des dérivées partielles	35
3.3	Résolution numérique du modèle de filtrage linéaire - équation de la chaleur 2-D	36
3.3.1	Schémas numériques	36
3.3.2	Résultat numérique	37
3.4	Résolution numérique du modèle de filtrage non linéaire - équation de Perona- Malik 1-D	38
3.5	Résolution numérique du modèle de filtrage non linéaire - équation de Perona- Malik 2-D	40
3.5.1	Discrétisation de l'opérateur DIV	41
3.5.2	Equation de diffusion non linéaire discrétisée	42
3.5.3	Forme matricielle	43
3.5.4	Résultat numérique	46
3.6	Comparaison entre l'EDP de Perona-Malik et l'équation de chaleur	47
	conclusion	48
	Bibliographie	49
	Annexe	51
	Annexe	51

Introduction

En modélisation par des EDPs du traitement d'images, on peut distinguer en particulier deux domaines d'application. Le premier est la restauration des images, qui consiste à améliorer l'image afin de la rendre plus significative à l'oeil en renforçant les contrastes de l'image originale et en corrigeant les distorsions (bruit) introduites lors de son acquisition. L'autre domaine est celui de la segmentation ou d'une manière plus générale celui de l'interprétation des images qui consiste à détecter les contours des objets significatifs de l'image. D'une manière générale, le pré-traitement représente une étape très importante en traitement d'image et en vision par ordinateur, puis qu'il permet un débruitage et une augmentation des contrastes qui facilite la segmentation par la suite. Le travail présenté dans ce mémoire se situe dans le premier domaine.

Mathématiquement, une image plane, généralement un rectangle, consiste en un ensemble discret de point «pixel» porteur une fonction u , le niveau de gris. Par ailleurs en traitement d'image et en vision par ordinateur, il est difficile d'analyser l'information contenue dans une image à partir de l'intensité lumineuse de chaque pixel. Par contre on peut saisir les changements d'intensité et la taille des zones présentant des propriétés d'homogénéité (ou regroupant des pixels faisant partie d'un "même" objet). D'où, la nécessité d'utiliser des équations aux dérivées partielles **EDPs** qui respectent la taille et les propriétés de ces zones. Ces approches ont de plus l'avantage de donner des résultats d'existence et d'unicité de la solution (image restaurée) et de permettre de mettre en oeuvre des schémas numériques stables.

Parmi les premiers modèles mathématiques utilisés, c'est le modèle de la chaleur qui constitue un modèle de filtrage linéaire. Cependant, ce modèle ne permet pas de détecter les contours d'image, Perona et Malik [10] ont proposé ensuite un modèle de diffusion non linéaire afin d'améliorer la détection des contours en modifiant l'équation de la chaleur.

L'objectif de ce mémoire est d'étudier en détails ce modèle de point de vue théorique et numérique, et établir une comparaison entre ces deux modèles.

Ce mémoire est organisé de la manière suivante. Le premier chapitre présente quelques notions de base sur l'objet image comme : la définition d'une image numérique et ses types, les différents filtres utilisés pour la traiter. Il présente encore le filtrage linéaire et non linéaire par l'EDP de la chaleur.

Le deuxième chapitre représente et analyse le modèle de Perona-Malik unidimensionnel, les filtres souhaitables et leurs propriétés pour préserver et améliorer les bords. On représente aussi les particularités de la fonction de diffusivité .

Le troisième chapitre présente la résolution numérique basée sur les approximations par différences finies. Nous allons voir dans un premier temps le cadre général de ces approches, ensuite, les principes théoriques des schémas utilisés pour les EDPs présentées précédemment. Enfin nous présentons quelques résultats numériques illustrés par des images pour l'équation de la chaleur et l'équation de Perona et Malik, et on termine cette présentation avec une comparaison entre les deux modèles.

Chapitre 1

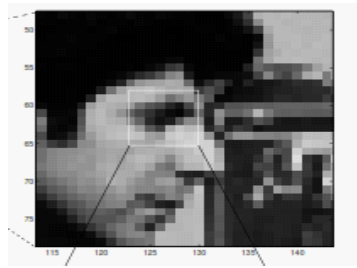
EDP et traitement de l'image

Le traitement d'image linéaire est basé sur les deux mêmes techniques classique : convolution et analyse de Fourier. La convolution est la plus importante de ces deux, puisque les images ont leurs informations codées dans le domaine spatial plutôt que dans le domaine fréquentiel. Le filtrage linéaire peut améliorer les images de plusieurs façons : aiguïser les bords des objets, réduire le bruit aléatoire, corriger l'illumination inégale, déconvolution pour corriger le flou et le mouvement, etc. Ces procédures sont effectuées en convoluant l'image originale avec un noyau de filtre approprié, produire l'image filtrée. Ce chapitre présente des définitions générales sur l'images numériques et ces types et présente des stratégies pour concevoir des noyaux de filtre pour diverses tâches de traitement d'image et présente encore le filtrage linéaire et non linéaire par L'EDP de la chaleur.

1.1 Définition d'une image numérique

Les images numériques sont constituées d'éléments d'image appelés pixels. Généralement, les pixels sont organisés dans un tableau rectangulaire ordonné. La taille d'une image est déterminée par les dimensions de ce tableau de pixels. La largeur de l'image est le nombre de colonnes et la hauteur de l'image est le nombre de lignes dans le tableau, matrice de pixels est une matrice de M colonnes \times N lignes. Pour désigner un pixel spécifique dans la matrice d'image, nous définissons sa coordonnée en x et y . Le système de coordonnées des matrices d'images définit x comme croissant de gauche à droite et y comme croissant du haut vers le bas.

Exemple pour une image numérique :



43	109	153	162	165	175	171	110	22
166	184	187	193	180	170	171	166	65
185	185	189	181	158	115	135	154	123
174	155	118	90	77	44	28	77	138
150	102	78	56	35	19	14	43	102
175	139	104	47	25	36	90	140	141
189	188	158	95	68	172	198	186	188
192	198	193	164	154	201	209	204	210

1.2 Types d'images numériques

On a Trois types d'image

1. Les images binaires(noir ou blanc) :

Une image binaire est une image numérique qui n'a que deux valeurs d'intensité possibles pour chaque pixel. Généralement, les deux couleurs utilisées pour une image binaire sont le noir et le blanc. L'image binaire est également appelée bi-niveau ou à deux niveaux. Cela signifie que chaque pixel est stocké sous la forme d'un seul bit, c'est-à-dire, un 0 pour le noir et 1 ou 255 pour le blanc. Les images montrées ci-dessous sont des exemples typiques des images binaires ici le pixel peut prendre seulement deux valeurs, 0 ou 255 [haut ou bas].

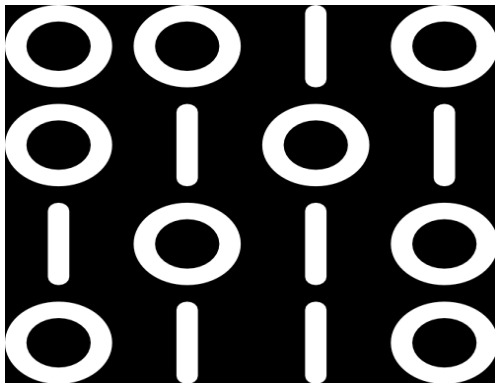


Image binaire

2. Les images en teintes de gris :

Niveaux de gris est la gamme de nuances monochromatiques du noir au blanc. L'image en niveaux de gris est une image dans laquelle la valeur de chaque pixel est un échantillon unique représentant seulement une quantité de lumière, c'est-à-dire qu'il ne transporte que des informations d'intensité. Les images de ce genre sont composées exclusivement de nuances de gris, allant du noir à l'intensité la plus faible au blanc au plus fort.



Image en niveaux de gris

3. Les images en couleurs :

Une image couleur est une image numérique qui inclut des informations de couleur pour chaque pixel. Pour des résultats visuellement acceptables, il est nécessaire (et presque suffisant) de fournir trois échantillons pour chaque pixel, qui sont interprétés comme des coordonnées dans un espace de couleur. L'espace couleur Rouge, Vert, Bleu (**RVB**) est le plus utilisé pour le maniement des images numériques.



49	55	56	57	52	53	64	76	82	79	78	78	66	80	77	80	87	77
58	60	60	58	55	57	93	93	91	91	86	86	81	93	96	99	86	85
58	58	54	53	55	56	88	82	88	90	88	89	83	83	91	94	92	88
83	78	72	69	68	69	125	119	113	108	111	110	135	128	126	112	107	106
88	91	91	84	83	82	137	136	132	128	126	120	141	129	129	117	115	101
69	76	83	78	76	75	105	108	114	114	118	113	95	99	109	108	112	109
61	69	73	78	76	76	96	103	112	108	111	107	84	93	107	101	105	102
Red						Green						Blue					

1.3 Définition d'une image mathématique

Soit Ω un ouvert borné de \mathbb{R}^2 (ex rectangle) [7].

Une image est définie comme une fonction

$$f : \Omega \text{ dans } \mathbb{R} \text{ (pour une image niveau de gris)}$$

et

$$f : \Omega \text{ dans } \mathbb{R}^3 \text{ (pour une image couleur)}$$

1.4 Filtrage linéaire

Le filtrage est une technique pour modifier ou améliorer une image. Par exemple, vous pouvez filtrer une image pour mettre en valeur certaines fonctionnalités ou supprimer d'autres fonctionnalités. Le filtrage est une opération de voisinage, dans laquelle la valeur de n'importe quel pixel donné dans l'image de sortie est déterminée en appliquant un certain algorithme aux valeurs des pixels dans le voisinage du pixel d'entrée correspondant. Le voisinage d'un pixel est un ensemble de pixels, définis par leurs emplacements par rapport à ce pixel. Le filtrage linéaire est un filtrage dans lequel la valeur d'un pixel de sortie est une combinaison linéaire des valeurs des pixels dans le voisinage du pixel d'entrée.

Soit I une image numérique.

Soit h une fonction de $[x_1, x_2] \times [y_1, y_2]$ à valeur réelles. La convolution de I par h est définie par :

$$(I * h) = \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} h(i, j). I(x - i, y - j)$$

h est dit le noyau de convolution.

Les nouvelles valeurs du pixel sont calculées par produit scalaire entre le noyau de convolution et le voisinage correspondant du pixel.

100	100	100	100	100
100	100	100	100	100
100	100	150	100	100
100	100	100	100	100
100	100	100	100	100

 \ast

0	-1	0
-1	5	-1
0	-1	0

 $=$

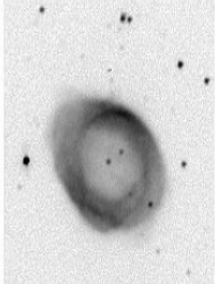
100	100	100	100	100
100	100	50	100	100
100	50	350	50	100
100	100	50	100	100
100	100	100	100	100

Exemple de convolution numérique

1.4.1 Différents filtres linéaires

– Le Filtre de moyenne [7]

Le filtrage moyen est une méthode simple, intuitive et facile à mettre en œuvre pour lisser des images, c'est-à-dire réduire la quantité de variation d'intensité entre un pixel et le suivant. Il est souvent utilisé pour réduire le bruit dans les images. Le filtre moyenne est un cas particulier de filtre de convolution, qui remplace chaque pixel par la moyenne des valeurs adjacents et du pixel central.


 \ast

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

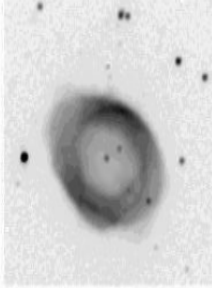
 $=$




Image en niveaux de gris filtrée par le filtre
moyenne

– **Le Filtre médian** [7]

le filtre médian n'est pas exactement un produit de convolution, est une technique de filtrage numérique, souvent utilisée pour atténuer des pixels isolés, d'une valeur très différente de leur entourage. Le filtrage médian est très largement utilisé dans le traitement d'image numérique car, dans certaines conditions, il préserve les bords tout en supprimant le bruit, ayant également des applications dans le traitement du signal. La valeur médiane d'une série de valeurs de pixel est par définition celle qui sépare l'échantillon en deux parties de population égale.

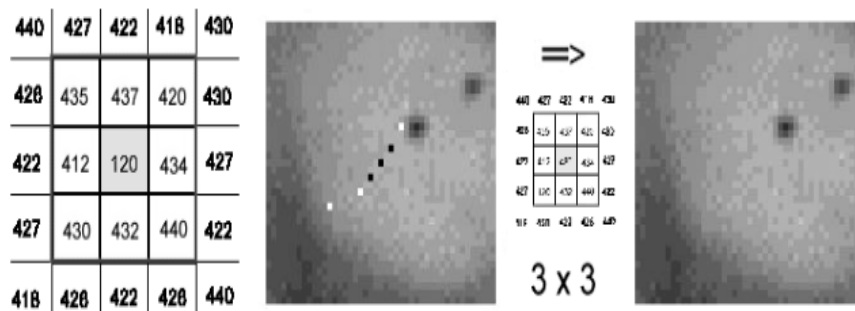


Image en niveau dr gris filtrée par le médian

– **Le Filtre Gaussien** [7]

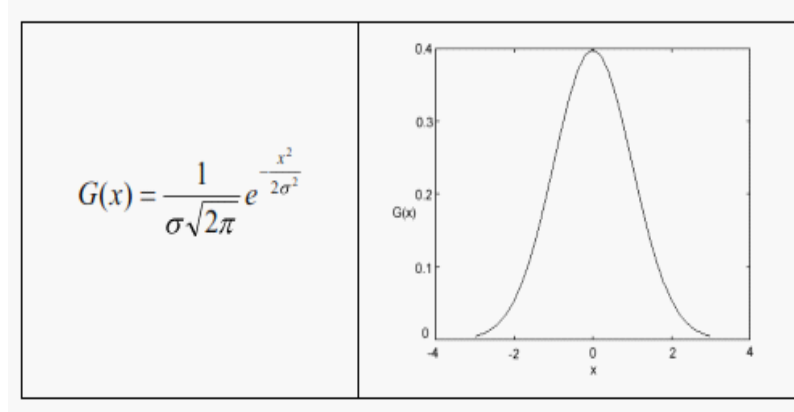
L'opérateur de lissage gaussien est un opérateur de convolution à 2-D qui est utilisé pour "brouiller" les images et supprimer les détails et le bruit. En ce sens, il est similaire au filtre moyen, mais il utilise un noyau différent qui représente la forme d'une bosse gaussienne (en forme de cloche). Ce noyau a des propriétés spéciales qui sont détaillées ci-dessous :

La distribution gaussienne en 1-D a la forme :

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

avec σ représente l'échelle de filtrage

où est l'écart-type de la distribution. Nous avons également supposé que la distribution a une moyenne de zéro (c'est-à-dire qu'elle est centrée sur la ligne $x = 0$). La distribution est illustrée à la figure ci-dessous.

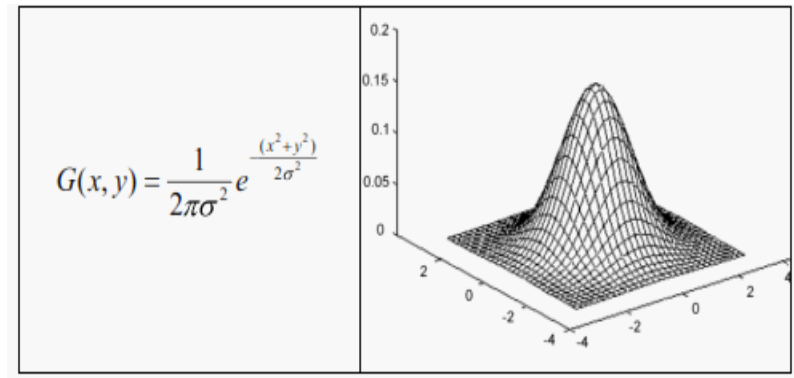


Distribution gaussienne 1-D avec moyenne 0 et $\sigma = 1$.

En 2-D, un gaussien isotrope [7] (c'est-à-dire circulairement symétrique) a la forme :

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

Cette distribution est illustrée à la figure ci-dessous.

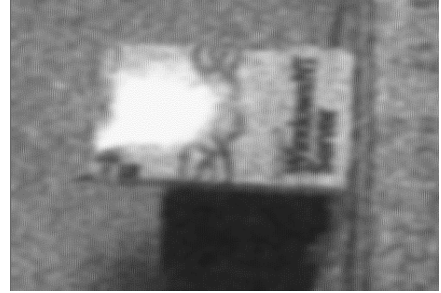


Distribution gaussienne 2-D avec moyenne $(0, 0)$ et $\sigma = 1$.

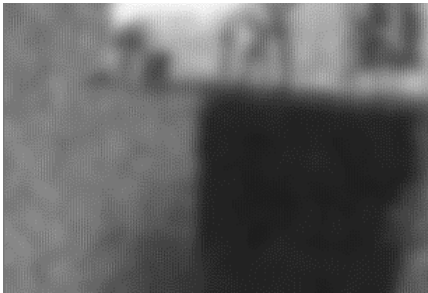
Exemple de filtrage Gaussien



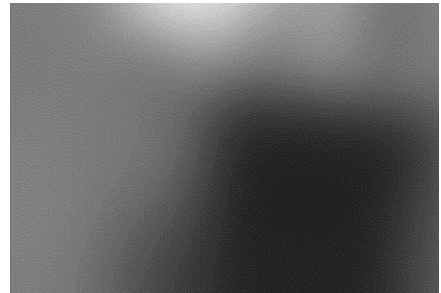
sigma = 10



sigma=20



sigma=500



sigma=1000

1.4.2 Equation de la chaleur et filtrage linéaire

– Equation de la chaleur sur \mathbb{R}

Soit le problème de la chaleur suivant :

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, & x \in \mathbb{R}, t > 0 \\ u(x, 0) = f(x) \end{cases}$$

On sait que pour $f(x)$ continue, bornée, la solution du problème s'écrit comme :

$$u(x, t) = \frac{1}{\sqrt{4\pi Kt}} \int_{\mathbb{R}} f(y) e^{-\frac{(x-y)^2}{4Kt}} dy, \quad K > 0$$

Et on peut écrire la solution comme produit de convolution :

$$u(x, t) = f * g_{\sqrt{2t}},$$

avec le filtre Gaussien :

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

– **Equation de chaleur sur $\mathbb{R} \times \mathbb{R}$**

Soit l'image définie par :

$f : \mathbb{R}^2 \rightarrow \mathbb{R}$, et soit l'équation de la chaleur donnée par

$$\begin{cases} \frac{\partial u}{\partial t} = \Delta u(x, y, t); & t > 0, \\ u(x, y, 0) = f(x, y). \end{cases}$$

Pour f continue et bornée, la solution est donnée par :

$$u(x, y, t) = (f * G_{\sqrt{2t}})(x, y),$$

avec :

$$\mathbf{G}_\sigma(\mathbf{x}, \mathbf{y}) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

On peut remarquer que la solution de l'équation de la chaleur [7] représente un filtrage linéaire d'une image initial f par le filtre Gaussien à l'échelle $\sqrt{2t}$

σ : indique la position du contour de l'image en fonction de σ .

Ce filtrage ne permet pas de détecter des contours d'images on a recoure aux filtrage non linéaire

1.5 Filtrage de diffusion non linéaire

1.5.1 Modèle de Perona-Malik 2-D

Les méthodes de lissage adaptif sont basées sur l'idée d'appliquer un processus qui depend lui-même des propriétés locales de l'image. Bien que ce concept soit bien connu dans la communauté du traitement d'image. Le modèle de diffusion non linéaire, d'abord introduit par Perona et Malik [10] en 1990. Dans ce qui suit nous allons discuter ce modèle en détail cela donne à étudier l'analyse de l'amélioration des bords.

On considère le problème linéaire de l'équation de la chaleur suivant : [2]

$$\frac{\partial u}{\partial t} = \Delta u \quad x \in \Omega, \quad t > 0 \tag{1.1}$$

avec une condition initial :

$$u(x, 0) = u_0(x) \quad x \in \Omega \quad (1.2)$$

et condition de Neumann

$$\frac{\partial u}{\partial n} \Big|_{\partial \Omega} = 0 \quad t > 0. \quad (1.3)$$

Le défaut du filtre de la chaleur image flou pas de détection de contours. Et pour améliorer les résultats obtenue par L'EDP de ce filtre Perona et Malik, ils ont proposé de remplacer l'équation de la chaleur par l'équation de diffusion non linéaire suivante :

$$\frac{\partial u}{\partial t} = \operatorname{div}(g(|\nabla u|)\nabla u) \quad x \in \Omega, \quad t > 0. \quad (1.4)$$

Où g est une fonction décroissante avec $g(0) = 1$ et $g(s)$ tendant à zéro quand s tend vers l'infini (Voir Fig 1.1). Notez qu'ici la diffusivité $g(|\nabla u|)$ depend explicitement de $|\nabla u|$ elle-même.

La principale raison de Perona et Malik [10] pour choisir ce type de diffusivités est de rendre le processus obtenu par l'équation (1.4) "adaptive". Cela signifie diffuser $u(x, t)$ moins aux endroits où $|\nabla u(x, t)|$ est le plus grand, puisque $g(s)$ est petit pour un grand s et la plupart aux endroits où $|\nabla u(x, t)|$ est le plus petit, puisque $g(0) = 1$. Par conséquent, au fur et à mesure que le processus évolue, $u(x, t)$ est lissé des deux côtés d'un "bord" avec un lissage minimal du bord lui-même (nous entendons par le bord un point où $|\nabla u(x, t)|$ est grand). Cela signifie que les détails fins qui correspondent aux "petits" gradients sont éliminés par le processus de diffusion tandis que les détails importants tels que les bords qui correspondent aux "grands" gradients sont conservés. Le mot "petit" (resp. grand) est utilisé pour exprimer que les gradients sont en dessous (resp. au-dessus) d'un certain seuil λ .

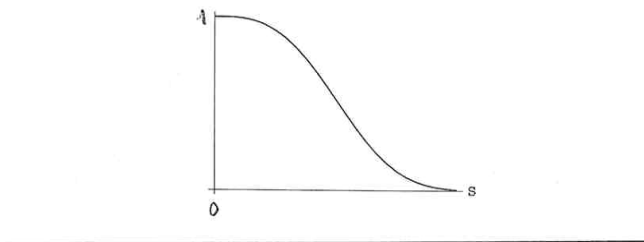


Figure 1.1 : Fonction de diffusivité $g(s)$

Perona et Malik [10] ont fourni quelques exemples de fonctions g qui permettent à l'équation (1.4) d'afficher également le comportement «amélioration des bords». ils ont proposé des diffusivités du type :

$$g(s) = \frac{1}{1 + (\frac{s}{\lambda})^2} \quad (1.5)$$

et,

$$g(s) = e^{-s^2/\lambda^2} \quad (1.6)$$

et ils ont utilisé une flux fonction définie par :

$$\phi(s) = sg(|s|) \quad (1.7)$$

cette fonction est une fonction non monotone. Plus précisément $\phi(s)$ croissante pour $s < \lambda$ mais décroissante pour $s > \lambda$, et de plus $\phi(s)$ tend vers zéro quand s tend vers l'infini.

La forme de cette fonction est montrée dans Fig.1.2.

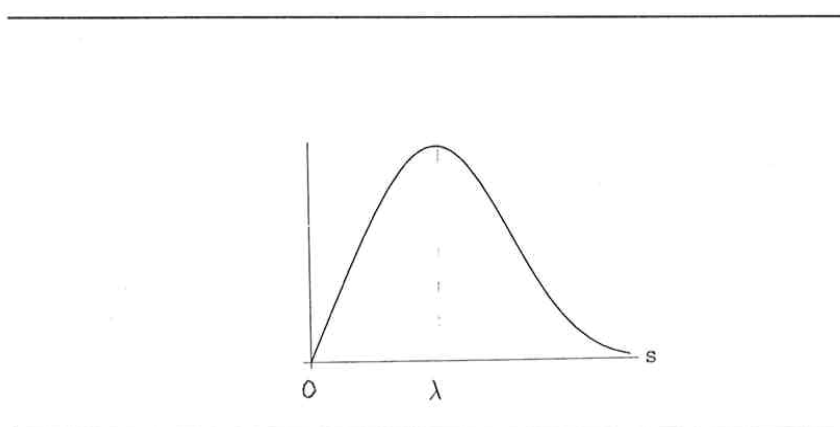


Figure 1.2 : Fonction de flux $\phi(s)$ qui conduit à l'amélioration des bords

Exemples pour le modèle de Perona Malik :



Image original



filtre de PM



Image bruité



Image filtré par PM

Chapitre 2

Le modèle de Perona-Malik unidimensionnel

Dans ce chapitre, nous allons étudier et analyser le modèle de Perona-Malik unidimensionnel

2.1 L'analyse mathématique du modèle

Perona et Malik proposent une méthode de diffusion non linéaire pour éviter le flou, préserver les bords (tous en restituant les image bruitées) ,améliorer et localiser les problèmes de diffusion linéaires. Nous allons détailler des travaux de BENHAMOUDA [2], dans le cas unidimensionnel.

2.1.1 Formulation de problème

Nous allons étudier le problème de la valeur limite et initial suivant :

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(g \left(\left| \frac{\partial u}{\partial x} \right| \right) \frac{\partial u}{\partial x} \right) \quad x \in \Omega =]0, 1[, \quad t > 0, \quad (2.1)$$

avec u est la solution de ce problème et définie sur

$$u : \Omega \times \mathbb{R}^+ \rightarrow \mathbb{R}$$

avec une donnée initiale lisse

$$u(x, 0) = u_0(x) \quad x \in \Omega, \quad (2.2)$$

et avec les conditions aux limites de Neuman homogènes

$$\frac{\partial u}{\partial x}(0, t) = \frac{\partial u}{\partial x}(1, t) = 0 \quad \text{pour tout } t > 0. \quad (2.3)$$

La fonction $g : [0, +\infty[\rightarrow]0, 1]$, est la diffusivité et elle dépend du modèle avec : $g(0) = 1$ et $g(s)$ tends vers zéro quand s tend à l'infini, ainsi l'équation devient non linéaire si la fonction de diffusivité dépend de la solution, la fonction g est supposée être une fonction de C^∞ monotone et décroissante.

Pour une constante $g = 1$, nous obtenons un modèle linéaire simple de l'équation de la chaleur, ce qui conduit au problème de cauchy suivant :

$$\begin{cases} u_t = \frac{\partial^2 u}{\partial x^2} & \text{pour } x \in \mathbb{R}, t > 0, \\ u(x, 0) = f(x) & \text{pour } x \in \mathbb{R}. \end{cases}$$

Les conditions aux limites sont des conditions de flux nulle, i.e,

$$\phi\left(\frac{\partial u}{\partial x}\right)_{x=0} = \phi\left(\frac{\partial u}{\partial x}\right)_{x=1} = 0.$$

avec $\phi(s) = sg(s)$

Il correspondent a une prolongation par réflexion de l'image à travers la frontière. Nous supposons que la donnée initiale est une fonction croissante monotone, i.e ; $\frac{\partial u}{\partial x}(x, 0) \geq 0$ pour tout $x \in \Omega$ (voir Fig.2.1).

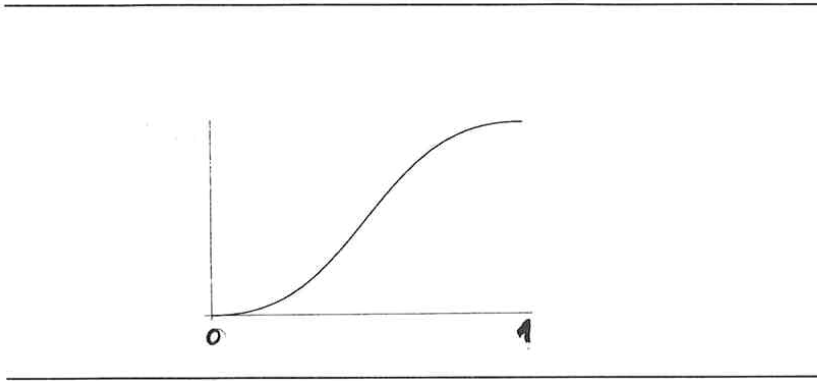


Figure2.1 : Le bord initial $u_0(x)$

2.1.2 Propriétés de la solution

nous supposons que le problème initial de valeur limite(2.1),(2.2) et (2.3) a une solution unique $u(x, t)$, dans cette hypothèse, la solution a les propriétés suivantes :

(i) Principe de maximum-minimum :

$$\begin{aligned}\max_{0 \leq x \leq 1} u(x, t_1) &\geq \max_{0 \leq x \leq 1} u(x, t_2) \text{ et} \\ \min_{0 \leq x \leq 1} u(x, t_1) &\leq \min_{0 \leq x \leq 1} u(x, t_2) \text{ pour } t_1 \leq t_2\end{aligned}$$

(ii) Conservation moyenne du niveau de gris :

$$\int_0^1 u(x, t) dx = \int_0^1 u_0(x) = cste. \quad \text{pour tous } t > 0 \quad (2.4)$$

(iii) Préservation de positivité :

$$u_0(x) \geq 0 \implies u(x, t) \geq 0 \quad x \in \Omega, \quad t > 0 \quad (2.5)$$

(iv) Préservation de la monotonie :

$$\frac{\partial u_0(x)}{\partial x} \geq 0 \implies \frac{\partial u(x, t)}{\partial x} \geq 0 \quad x \in \Omega, \quad t > 0 \quad (2.6)$$

(v) Energie décroissante :

$$\|u(t)\|_2 =: \left(\int_0^1 u^2(x, t) dx \right)^{1/2} \quad (2.7)$$

est une fonction décroissante pour $t > 0$

(vi) Estimation de gradient :

Soit λ la valeur de s où ϕ atteint son maximum alors :

$$\frac{\partial u}{\partial x}(x_0, t_0) \leq \lambda \implies \frac{\partial u}{\partial x}(x_0, t) \leq \lambda \text{ pour tous } t \geq t_0 \quad (2.8)$$

L'interprétation de la partie (vi) est qu'une fois $\frac{\partial u}{\partial x} \leq \lambda$ tient à un certain point x_0 , pour $t = t_0$, alors $\frac{\partial u}{\partial x}$ reste borné par λ pour tous $t > t_0$.

Preuve de propriétés

(i) Principe de maximum-minimum :

La preuve du principe de maximum-minimum peut trouvée dans [11]

(ii) Propriété de conservation moyenne du niveau de gris : [2]

On a :

$$\frac{\partial u}{\partial t} - \frac{\partial}{\partial x} \left(g \left(\left| \frac{\partial u}{\partial x} \right| \right) \frac{\partial u}{\partial x} \right) = 0$$

on intégrant sur $[0, 1]$ par rapport à x :

$$\frac{\partial}{\partial t} \int_0^1 u(x, t) dx - \int_0^1 \frac{\partial}{\partial x} \left(g \left(\left| \frac{\partial u}{\partial x} \right| \right) \frac{\partial u}{\partial x} \right) dx = 0$$

donc,

$$\frac{d}{dt} \int_0^1 u(x, t) dx - \left[g \left(\left| \frac{\partial u}{\partial x} \right| \right) \frac{\partial u}{\partial x} \right]_0^1 = 0$$

et on a :

$$\left[g \left(\left| \frac{\partial u}{\partial x} \right| \right) \frac{\partial u}{\partial x} \right]_0^1 = 0$$

à partir des conditions aux limites de Neumann :

$$\frac{\partial u}{\partial x}(0, t) = \frac{\partial u}{\partial x}(1, t) = 0$$

d'où

$$\frac{d}{dt} \int_0^1 u(x, t) dx = 0$$

$$\implies \int_0^1 u(x, t) dx = cte = \int_0^1 u(x, 0) dx, \forall t \geq 0$$

(iii) Propriété de préservation de positivité : [2]

Nous pouvons prouver cette propriété en la déduisant de le principe de maximum-minimum.

nous avons l'inégalité suivante :

$$\min_{0 \leq x \leq 1} u(x, 0) \leq \min_{0 \leq x \leq 1} u(x, t) \leq u(x, t) \quad \text{pour tous } t \geq 0$$

donc si $u(x, 0) \geq 0$ pour tous $0 \leq x \leq 1$ alors $u(x, t) \geq 0$ pour tous $0 \leq x \leq 1$ et $t \geq 0$.

(iv) Propriété de préservation de la monotonie : [8]

Notons que si la fonction initial $\frac{\partial u}{\partial x}(x, 0)$ est monotone par rapport à x , alors la solution de l'ensemble des équations sera monotone par rapport à x pour tout $t > 0$. Car, soit $u_0(x)$ est monotone (i.e, $u_0(x_{i+1}) > u_0(x_i)$ pour tout i). Supposons que pour certain $t > 0$ $u(x_i, t)$ n'est pas monotone par rapport à x_i , notons par t_0 le plus petit t (qui existe évidemment) puis avec $t = t_0$, pour $u(x_i, t_0)$ nous avons $u(x_{i+1}, t_0) \geq u(x_i, t_0)$ pour $i = 0, 1, \dots, N-1$; il y a un i tel que $u(x_i, t_0) = u(x_{i+1}, t_0)$. Ensuite, il y a un i_0 tel que $u(x_{i_0-1}, t_0) < u(x_{i_0}, t_0) = u(x_{i_0+1}, t_0) \leq u(x_{i_0+2}, t_0)$ ou un i_0 tel que $u(x_{i_0-1}, t_0) \leq u(x_{i_0}, t_0) = u(x_{i_0+1}, t_0) < u(x_{i_0+2}, t_0)$. Montrons

que cela est impossible pour la breVité, nous ne considérerons que la première hypothèse de $u(x_{i_0-1}, t_0) < u(x_{i_0}, t_0) = u(x_{i_0+1}, t_0) \leq u(x_{i_0+2}, t_0)$ il s'ensuit que

$$\frac{\partial u(x_{i_0}, t_0)}{\partial t} < 0, \quad \frac{\partial u(x_{i_0+1}, t_0)}{\partial t} \geq 0.$$

Par conséquent pour cela t_0 nous avons

$$\frac{\partial u}{\partial t}(u(x_{i_0+1}, t_0) - u(x_{i_0}, t_0)) > 0$$

et donc, pour t suffisamment proche de et moins que t_0 nous avons $u(x_{i+1}, t) - u(x_i, t) < 0$. Ce qui est impossible dans la mesure où c'est la moindre pour laquelle la monotonie de $u(x_i, t)$ par rapport à x_i est détruite.

(v) Propriété d' énergie décroissante : [2]

Pour démontrer la propriété d'energie décroissante nous dérivons $\|u(t)\|_2^2$ par rapport au temps on a :

$$\|u(t)\|_2^2 = \int_0^1 u^2(x, t) dx$$

d'où

$$\frac{d}{dt} \|u(t)\|_2^2 = 2 \int_0^1 \frac{d}{dt} u(x, t) u(x, t) dx$$

donc,

$$\frac{1}{2} \frac{d}{dt} \|u(t)\|_2^2 = \frac{d}{dt} \int_0^1 u^2(x, t) dx$$

on multiplie l'équation (2.1) par $\int_0^1 u(x, t) dx$ on obtient :

$$\frac{d}{dt} \int_0^1 u^2(x, t) dx = \int_0^1 u(x, t) \frac{\partial}{\partial x} \left(g \left(\left| \frac{\partial u}{\partial x} \right| \right) \frac{\partial u}{\partial x} \right) dx$$

et on a :

$$\frac{d}{dt} \int_0^1 u^2(x, t) dx = \frac{1}{2} \frac{d}{dt} \|u(t)\|_2^2$$

donc l'équation (2.1) devient ,

$$\frac{1}{2} \frac{d}{dt} \|u(t)\|_2^2 = \int_0^1 u(x, t) \frac{\partial}{\partial x} \left(\phi \left(\frac{\partial u}{\partial x} \right) \right) dx \leq 0 \quad \text{pour tout } t > 0$$

avec :

$$\phi \left(\frac{\partial u}{\partial x} \right) = \frac{\partial u}{\partial x} g \left(\left| \frac{\partial u}{\partial x} \right| \right)$$

après on intègre par partie et en utilisant les conditions aux limites (2.3)

$$\left[u(x, t) \phi \left(\frac{\partial u}{\partial x} \right) \right]_0^1 - \int_0^1 \frac{\partial u}{\partial x}(x, t) \phi \left(\frac{\partial u}{\partial x} \right) \leq 0 \quad \text{pour tout } t > 0$$

on a

$$\left[u(x, t) \phi \left(\frac{\partial u}{\partial x} \right) \right]_0^1 = 0$$

finalement on obtient

$$\frac{1}{2} \frac{d}{dt} \|u(t)\|_2^2 = - \int_0^1 \frac{\partial u}{\partial x}(x, t) \phi \left(\frac{\partial u}{\partial x} \right) \leq 0 \quad \text{pour tout } t > 0.$$

Puisque $\frac{\partial u}{\partial x}(x, t) \geq 0$ de la monotonie alors, l'énergie $\|u(t)\|_2$ est une fonction décroissante pour $t > 0$.

(vi) Propriété d'estimation de gradient : [2]

Pour prouver la propriété (vi), supposons que $u \in C^3$, puis dérivons l'équation (2.1) une fois par rapport à la variable x :

$$\frac{\partial}{\partial x} \left(\frac{\partial u}{\partial t} \right) = \frac{\partial}{\partial x} \left(\frac{\partial}{\partial x} \phi \left(\frac{\partial u}{\partial x} \right) \right)$$

donc,

$$\frac{\partial}{\partial t} \left(\frac{\partial u}{\partial x} \right) = \frac{\partial}{\partial x} \left(\frac{\partial}{\partial x} \left(\frac{\partial u}{\partial x} \right) \phi' \left(\frac{\partial u}{\partial x} \right) \right)$$

on pose :

$$v = \frac{\partial u}{\partial x}, \quad k(v) = \phi'(v)$$

alors on obtient

$$\frac{\partial v}{\partial t} = \frac{\partial}{\partial x} \left(k(v) \frac{\partial v}{\partial x} \right). \quad (2.9)$$

Cette équation est de type parabolique avec $k(v) > 0$ pour tout $v < \lambda$. Par conséquent, en utilisant le théorème 12 à la page 187 indiqué dans [11].

Soit $E = \Omega \times \mathbb{R}^+$, on pose :

$$L(v) = \frac{\partial}{\partial x} \left(k(v) \frac{\partial v}{\partial x} \right) - \frac{\partial v}{\partial t}$$

on suppose que v la solution de $L(v) = f(x, t)$ dans E , et v satisfait la condition initiale

$$\begin{aligned} v(x, 0) &= g_1(x) \quad \text{dans } E \\ v(x, t) &= g_2(x, t) \quad \text{dans } \partial\Omega \times \mathbb{R}^+ \end{aligned}$$

on suppose que z et Z satisfaisent l'inégalité suivante

$$L(z) \leq f(x, t) \leq L(Z)$$

Si

$$\begin{aligned} z(x, 0) &\leq g_1(x) \leq Z(x, 0) && \text{dans } \Omega \\ z(x, t) &\leq g_1(x) \leq Z(x, t) && \text{dans } \partial\Omega \times \mathbb{R}^+ \end{aligned}$$

alors

$$z(x, t) \leq v(x, t) \leq Z(x, t) \quad \text{dans } E$$

alors nous obtenons un principe de maximum pour l'équation (2.9). D'où nous avons l'inégalité suivante

$$\max_{0 \leq x \leq 1} v(x, t) \leq \max_{0 \leq x \leq 1} v(x, t_0) \quad \text{pour tout } t \geq t_0.$$

Donc si $v(x, t_0) < \lambda$ alors $v(x, t) < \lambda$ pour tout $t \geq t_0$.

2.1.3 Les paramètres du problème

Le modèle de Perona et Malik dépend de la diffusivité g et du paramètre de temps t qui mesure l'étendue à laquelle l'image est traitée. Trop d'itérations dans le temps diffuseront complètement l'image et toutes les informations seront perdues puisque $u(x, t)$ tend vers une constante quand t tend vers l'infini. Pour notre famille de filtres, la diffusivité g dépend de deux paramètres λ et α . Le paramètre λ joue le rôle d'un seuil pour les gradients. Les gradients inférieurs à λ sont lissés tandis que les gradients supérieurs à λ sont aiguisés. Le paramètre α contrôle la décroissance de la diffusivité g . Si $\alpha > 1$ alors la diffusivité g est une fonction rapidement décroissante pour de grandes valeurs de α . En revanche, si $\alpha < 1$ alors g diminue rapidement pour de faibles valeurs de α .

2.1.4 Le problème non homogène

Lors du traitement des images, une question qui se pose immédiatement est comment savoir le bon moment t pour arrêter le processus. Avoir de telles informations est important puisque la solution $u(x, t)$ de (1.4) tend vers un état stationnaire constant quand $t \rightarrow \infty$.

Cela signifie que toutes les informations importantes concernant les détails importants sont perdues. Afin d'éviter que $u(x, t)$ ne présente un état stationnaire constant, *NORDSTRÖM* [14] a introduit dans le membre de droite de l'équation (1.4) un terme de forçage du type :

$$f(x, u) = \nu.(u_0(x) - u(x, t)) \quad \text{avec} \quad \nu = 1. \quad (2.10)$$

Cette idée, cependant, ne réduit pas le nombre de paramètres du problème. En fait, le paramètre T n'est remplacé que par le paramètre "caché" $\nu = 1$ dans la partie droite. De plus, les calculs numériques montrent que ce terme de forçage a un effet sur l'amélioration des bords. Un autre point important à noter est que ce terme forçant tend également à déstabiliser le processus

2.2 Les filtres souhaitables et leurs Propriétés

L'obtention du comportement souhaité à partir de l'équation (2.1) dépend d'un choix approprié de la diffusivité g . L'exigence de base qui doit être satisfaite par un filtre est la propriété de préservation de contour. Nous entendons par là que le filtre garde les emplacements des bords initiaux de l'image originale inchangés. Une autre propriété importante qu'un filtre doit avoir est la propriété d'amélioration des bords. Nous entendons par là que le filtre affine les bords. Ces deux propriétés sont discutées ci-dessous

2.2.1 Propriété de préservation de bords

Pour que l'équation (2.1) préserve les bords, il faut limiter la diffusion aux bords. Ainsi, la diffusivité g devrait être une fonction décroissante monotone non négative tendant vers zéro quand s tend vers l'infini avec $g(0) = 1$. Nous reprenons ces deux propriétés souhaitables dans les hypothèses suivantes :

$$\begin{aligned} \textbf{(P1)} \quad & 0 < g(s) < 1 \quad \text{pour tout} \quad s > 0 \quad \text{et} \quad g(0) = 1 \\ \textbf{(P2)} \quad & g'(s) < 0 \quad \text{pour tout} \quad s > 0 \end{aligned}$$

Une forme typique de telles fonctions g est montrée dans (Fig.1.1)

2.2.2 Propriété d'amélioration des bords

pour permettre à l'équation (2.1) à améliorer les bords, la diffusivité g doit avoir une propriété supplémentaire : g doit décroître rapidement lorsque s tend vers l'infini de sorte que la fonction de flux $\phi(s) = sg(|s|)$ tend vers zéro lorsque s tend vers l'infini. Ceci implique qu'il existe une certaine valeur de seuil λ où $\phi'(\lambda) = 0$ et telle que $\phi(s)$ croissante pour $s < \lambda$ et diminue pour $s > \lambda$. Nous reprenons ces deux propriétés dans les hypothèses suivantes :

$$(E1) \quad \phi(s) \rightarrow 0 \quad \text{quand} \quad s \rightarrow \infty.$$

$$(E2) \quad \phi'(\lambda) = 0, \quad \phi'(s) > 0 \quad \text{pour} \quad s < \lambda \quad \text{et}, \quad \phi'(s) < 0 \quad \text{pour} \quad s > \lambda$$

Une forme typique de la fonction de flux $\phi(s)$ qui conduit à l'amélioration des bords est illustrée à Fig.1.2. Nous sommes préoccupés par la variation dans le temps des gradients dans la région du bord. nous écrivons l'équation (2.9) comme suit :

$$\frac{\partial}{\partial t} \left(\frac{\partial u}{\partial x} \right) = \frac{\partial}{\partial x} \left(\phi' \left(\frac{\partial u}{\partial x} \right) \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial x} \right) \right)$$

après avoir fait la dérivation on obtient

$$\frac{\partial u_x}{\partial t} = u_{xx} \phi''(u_x) (u_x)_x + (u_x)_{xx} \phi'(u_x)$$

donc l'équation devient

$$\frac{\partial u_x}{\partial t} = \phi''(u_x) u_{xx}^2 + \phi'(u_x) u_{xxx} \quad (2.11)$$

au bord $u_{xx}(x, t) = 0$, et $u_{xxx}(x, t) \leq 0$ puisque à ce point le gradient est maximum (voir Fig.2.2). Ainsi, pour un choix approprié de la diffusivité g , le gradient au bord augmente dans le temps si $\phi'(u_x(x, t)) < 0$ alors que les gradients voisins diminuent. Ce processus conduit à des bords vives. Si en revanche, $\phi'(u_x(x, t)) > 0$ alors le bord est lissé puisque $\frac{\partial u_x}{\partial t} \leq 0$. Par conséquent, les fonctions g pour lesquelles ϕ satisfait les propriétés (E1) et (E2) donnent les propriétés souhaitables d'amélioration des bords et de flou des petits détails.

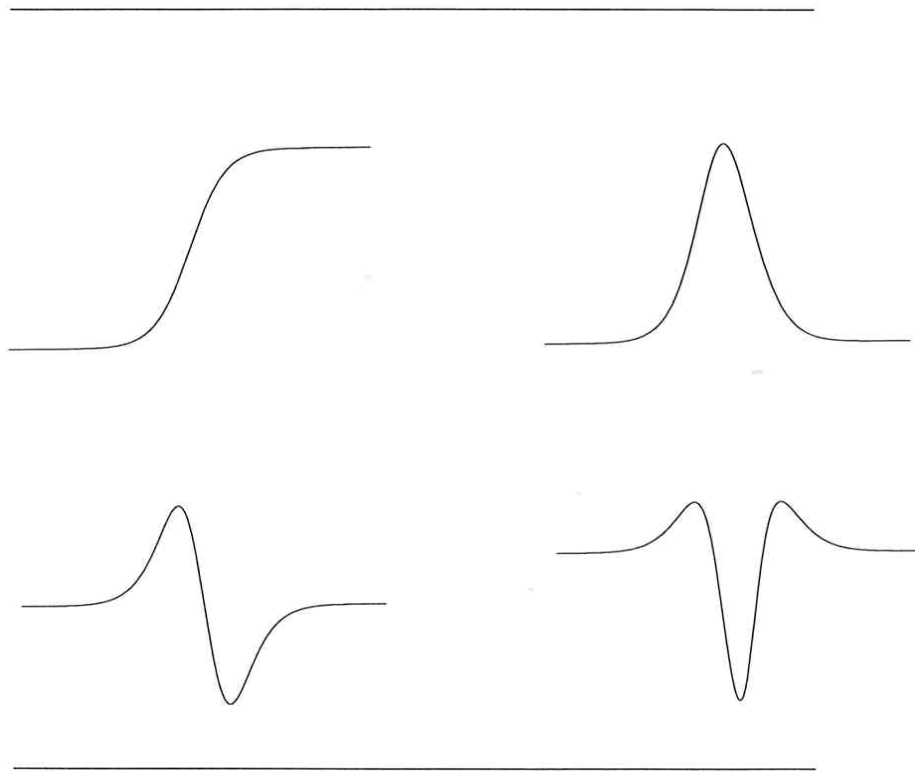


Figure 2.2 : (De gauche à droite) Un bord de pas mollifié et ses dérivées 1^{ère}, 2^{ème} et 3^{ème}

Le filtrage par diffusion non linéaire peut lisser le bruit tout en respectant la région limites et petites structures dans l'image, aussi longtemps que certains de ses paramètres cruciaux sont déterminés ou estimés correctement. Selon **Perona et Malik**, le choix de fonctions g pour lesquelles ϕ satisfait certaines propriétés conduit au résultat souhaitable des bords renforcement. Les fonctions de diffusivité les plus couramment utilisées dans la littérature sont listées dans le tableau suivant :

Function	Référence
$g(s) = 1$	Diffusion linéaire
$g(s) = \frac{1}{\sqrt{1+\frac{s^2}{\lambda^2}}}$	[5]
$g(s) = \frac{1}{1+\frac{s^2}{\lambda^2}}$	[10]
$g(s) = \exp\left(-\frac{s^2}{\lambda^2}\right)$	[10]
$g(s) = 0.5 [(\tanh [0.2 (\lambda - s)]) + 1]$	[9]
$g(s) = \begin{cases} 1 - \exp\left[-\frac{3.15}{(\frac{s}{\lambda})^4}\right], & s > 0 \\ 1 & s \leq 0 \end{cases}$	[15]
$g(s) = \begin{cases} \frac{1}{2} \left[1 - \left(\frac{s}{\lambda}\right)^2\right]^2, & s > \lambda \\ 0 & \text{autrement} \end{cases}$	[4]
$g(s) = \begin{cases} \frac{P(T+\varepsilon)^{P-1}}{T}, & s < T \\ \frac{P(s+\varepsilon)^{P-1}}{s}, & s \geq T \end{cases}$	[17]
$g(s) = \begin{cases} \frac{P(T+\varepsilon)^{P-1}}{T} + \frac{1}{T}, & s < T \\ \frac{P(s+\varepsilon)^{P-1}}{s} + \frac{1}{s}, & s \geq T \end{cases}$	[17]

Remarque : λ est le seuil de bruit.

2.2.3 Construction des filtres

Pour faire évoluer le modèle de Perona et Malik en une étape et pour mieux comprendre la formation des pas artificiels, nous laissons la diffusivité g dépendre de deux paramètres α et λ . La famille de filtres que nous proposons ici est $g(s) = \exp(-\frac{1}{\alpha} (\frac{s}{\lambda})^\alpha)$ où $\alpha > 0$. Les formes de ces filtres sont illustrées à la (Fig.2.3). Cette famille de filtres a la propriété de préserver les bords et même de les rehausser.

2.2.4 Particularités de la diffusivité :

Dans l'analyse de l'amélioration des bords, nous aurons besoin de quelques propriétés de ces diffusivités.

(A) Propriétés de la diffusivité g

La famille des diffusivités que nous étudions est la famille des fonctions suivantes :

$$g(s) = \exp \left(-\frac{1}{\alpha} \left(\frac{s}{\lambda} \right)^\alpha \right), \text{ d'où } \alpha > 0 \text{ et } \lambda > 0$$

(A1) Limitations

pour $s \geq 0$ la fonction g est bornée tel que

$$0 \leq g(s) \leq 1 \quad \text{pour tout } s \geq 0 \text{ et } g(0) = 1$$

et elle contient une limite nulle à l'infini :

$$\lim_{s \rightarrow \infty} g(s) = 0$$

(A2) Décroissante monotone :

La fonction g est décroissante de façon monotone pour $s \geq 0$. Cette propriété est nécessaire pour réduire la diffusion aux bords on a :

$$g(s) = \exp \left(-\frac{1}{\alpha} \left(\frac{s}{\lambda} \right)^\alpha \right)$$

on dérive g par rapport à s on obtient

$$g'(s) = -\frac{1}{\alpha} \frac{\alpha}{\lambda} \left(\frac{s}{\lambda} \right)^{\alpha-1} \exp \left(-\frac{1}{\alpha} \left(\frac{s}{\lambda} \right)^\alpha \right)$$

alors,

$$g'(s) = -\frac{1}{\lambda} \left(\frac{s}{\lambda} \right)^{\alpha-1} g(s)$$

pour $s > 0$ $g'(s) < 0$

pour $\alpha > 1$ on a $g'(s) = 0$

pour $\alpha = 1$ $g'(s) = -\frac{1}{\lambda}$ et $\lim_{s \rightarrow \infty} g'(s) = -\infty$

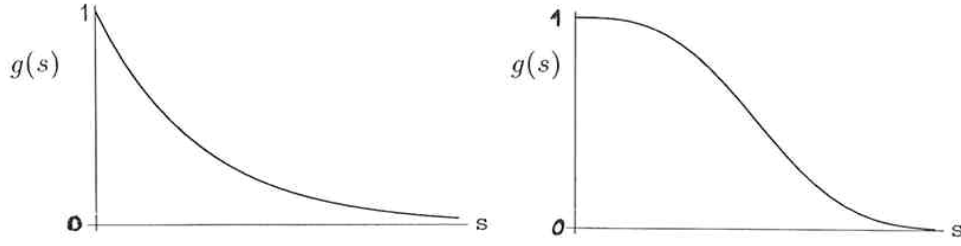


Figure 2.3 : (De gauche à droite) Fonctions de diffusivités $g(s)$ pour $\alpha = 1$ et $\alpha > 1$

(B) Propriétés de la fonction de flux

La fonction de flux est définie par $\phi(s) = sg(|s|)$ et possède les propriétés suivantes :

(B1) Positivité

On a :

$$0 < \phi(s) \leq \phi(\lambda) \quad \text{pour tout } s > 0$$

(B2) Propriété de non-flux

La disparition de $\phi(s)$ en $s = 0$ et en l'infini i.e

$$\lim_{s \rightarrow \infty} \phi(s) = \phi(0) = 0,$$

indiquent qu'il n'y pas de flux aux bords aussi bien qu'aux grands gradients. Cette propriété est également nécessaire pour assurer la non-monotonie de ϕ . En effet cela implique qu'il existe un certain seuil $\lambda > 0$ pour lequel $\phi'(\lambda) = 0$ et tel que $\phi'(s) > 0$ pour $s < \lambda$ et $\phi'(s) < 0$ pour $s > \lambda$.

(B3) Non-monotonie

La propriété de non-monotonie de la fonction de flux ϕ est nécessaire pour améliorer le plus grand gradient. La dérivée de la fonction de flux ϕ a les propriétés suivantes :

on a

$$\begin{aligned} \phi(s) &= sg(s) \\ &= s \exp\left(-\frac{1}{\alpha} \left(\frac{s}{\lambda}\right)^\alpha\right) \end{aligned}$$

$$\begin{aligned}
\phi'(s) &= \exp\left(-\frac{1}{\alpha}\left(\frac{s}{\lambda}\right)^\alpha\right) + \left(-\frac{1}{\alpha}\frac{\alpha}{\lambda}\left(\frac{s}{\lambda}\right)^{\alpha-1}\exp\left(-\frac{1}{\alpha}\left(\frac{s}{\lambda}\right)^\alpha\right)\right) \times s \\
&= \exp\left(-\frac{1}{\alpha}\left(\frac{s}{\lambda}\right)^\alpha\right) - \frac{s}{\lambda}\left(\frac{s}{\lambda}\right)^{\alpha-1}\exp\left(-\frac{1}{\alpha}\left(\frac{s}{\lambda}\right)^\alpha\right) \\
&= \left(1 - \left(\frac{s}{\lambda}\right)^\alpha\right)g(s) \quad \text{avec } \phi'(\lambda) = 0
\end{aligned}$$

Nous avons

$$\phi'(s) > 0 \iff \left(1 - \left(\frac{s}{\lambda}\right)^\alpha\right) \exp\left(-\frac{1}{\alpha}\left(\frac{s}{\lambda}\right)^\alpha\right)$$

ona

$$\exp\left(-\frac{1}{\alpha}\left(\frac{s}{\lambda}\right)^\alpha\right) > 0$$

donc

$$1 - \left(\frac{s}{\lambda}\right)^\alpha > 0 \iff \left(\frac{s}{\lambda}\right)^\alpha < 1 \iff \frac{s}{\lambda} < 1$$

$$\implies s < \lambda$$

alors

$$\phi'(s) > 0 \quad \text{pour } s < \lambda.$$

Cette propriété permet au filtre non linéaire de lisser les gradients qui sont plus petits que λ .

on a

$$\phi'(s) < 0 \iff \left(1 - \left(\frac{s}{\lambda}\right)^\alpha\right) \exp\left(-\frac{1}{\alpha}\left(\frac{s}{\lambda}\right)^\alpha\right) < 0$$

nous avons toujours

$$\exp\left(-\frac{1}{\alpha}\left(\frac{s}{\lambda}\right)^\alpha\right) > 0$$

donc

$$1 - \left(\frac{s}{\lambda}\right)^\alpha < 0$$

alors

$$1 < \left(\frac{s}{\lambda}\right)^\alpha \implies \lambda < s \text{ ou } s > \lambda.$$

Cette propriété permet au filtre non linéaire d'améliorer les gradients qui sont plus grands que λ

on a

$$\begin{aligned}\phi'(0) &= \left(1 - \left(\frac{0}{\lambda}\right)^\alpha\right) g(0) = g(0) = 1 \text{ et} \\ \lim_{s \rightarrow \infty} \phi'(s) &= 0.\end{aligned}$$

La première dérivée a le représenté sur Fig.2.4.

Par conséquent, notre famille de filtres a les propriétés souhaitées de préserver les bords et même les améliorer

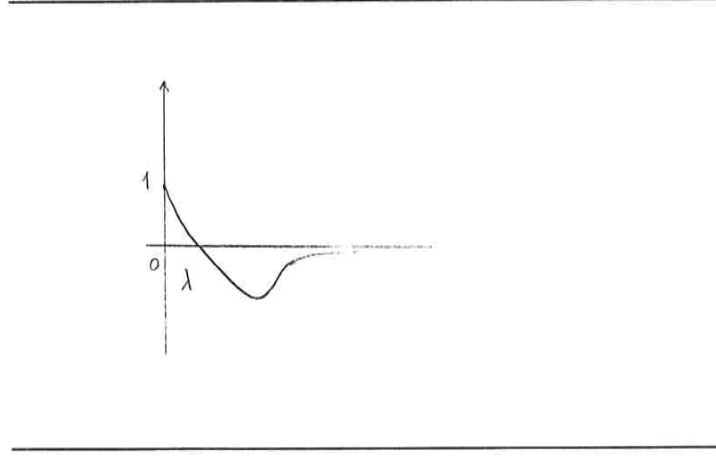


Figure 2.4 : La dérivée première de $\phi(s)$

(B4) La dérivée première de ϕ est :

$$\phi'(s) = \left(1 - \left(\frac{s}{\lambda}\right)^\alpha\right) \exp\left(-\frac{1}{\alpha} \left(\frac{s}{\lambda}\right)^\alpha\right)$$

et la deuxième dérivée de ϕ est :

$$\begin{aligned}\phi''(s) &= -\frac{\alpha}{\lambda} \left(\frac{s}{\lambda}\right)^{\alpha-1} \exp\left(-\frac{1}{\alpha} \left(\frac{s}{\lambda}\right)^\alpha\right) - \frac{1}{\lambda} \left(\frac{s}{\lambda}\right)^{\alpha-1} \exp\left(-\frac{1}{\alpha} \left(\frac{s}{\lambda}\right)^\alpha\right) \times \left(1 - \left(\frac{s}{\lambda}\right)^\alpha\right) \\ &= -\frac{\alpha}{\lambda} \left(\frac{s}{\lambda}\right)^{\alpha-1} g(s) - \frac{1}{\lambda} \left(\frac{s}{\lambda}\right)^{\alpha-1} \left(1 - \left(\frac{s}{\lambda}\right)^\alpha\right) g(s) \\ &= \frac{1}{\lambda} \left(\frac{s}{\lambda}\right)^{\alpha-1} \left[-\alpha - 1 + \left(\frac{s}{\lambda}\right)^\alpha\right] g(s) \\ \phi''(s) &= \frac{1}{\lambda} \left(\frac{s}{\lambda}\right)^{\alpha-1} \left[\left(\frac{s}{\lambda}\right)^\alpha - (1 + \alpha)\right] g(s).\end{aligned}$$

On a

$$\phi''(s) = 0 \iff \frac{1}{\lambda} \left(\frac{s}{\lambda}\right)^{\alpha-1} \left[\left(\frac{s}{\lambda}\right)^\alpha - (1 + \alpha)\right] g(s) = 0$$

nous avons

$$g(s) = 0 \quad \text{ou} \quad \frac{1}{\lambda} \left(\frac{s}{\lambda}\right)^{\alpha-1} \left[\left(\frac{s}{\lambda}\right)^\alpha - (1 + \alpha)\right] = 0$$

on a $g(s) \neq 0$ alors

$$\left(\frac{s}{\lambda}\right)^{\alpha-1} = 0 \quad \text{ou} \quad \left(\frac{s}{\lambda}\right)^\alpha = (1 + \alpha)$$

si $\left(\frac{s}{\lambda}\right)^{\alpha-1} = 0$ alors $s = 0$ ou si $\left(\frac{s}{\lambda}\right)^\alpha = (1 + \alpha)$ alors $s = \lambda(1 + \alpha)^{\frac{1}{\alpha}}$

$\phi''(s)$ a toujours la racine $\mu = \lambda(1 + \alpha)^{\frac{1}{\alpha}}$.

La dérivée second de ϕ a la propriété de tendre vers zéro quand s tend vers l'infini. Pour de petites valeurs de α , la racine augmente et nous avons $\mu \rightarrow \lambda e$ quand $\alpha \rightarrow 0$. Ainsi, le décalage du point d'inflexion vers la droite. Cela nous permet d'obtenir des arêtes(bords) vives en une seul étape.

La valeur de la dérivée second au seuil λ est $\phi''(\lambda) = -\frac{\alpha}{\lambda} e^{-\frac{1}{\alpha}}$.

Concavité de ϕ dans $[0, \mu]$

Cette propriété est très importante car, elle assure le comportement en une étape. La fonction ϕ est une fonction concave si $\phi''(s) < 0$ et nous avons

$$\begin{aligned} \phi''(s) &< 0 \iff \frac{1}{\lambda} \left(\frac{s}{\lambda}\right)^{\alpha-1} \left[\left(\frac{s}{\lambda}\right)^\alpha - (1 + \alpha)\right] g(s) < 0 \\ &\iff \frac{1}{\lambda} \left(\frac{s}{\lambda}\right)^{\alpha-1} \left[\left(\frac{s}{\lambda}\right)^\alpha - (1 + \alpha)\right] < 0 \end{aligned}$$

donc

$$\left(\frac{s}{\lambda}\right)^{\alpha-1} \left(\frac{s}{\lambda}\right)^\alpha - \left(\frac{s}{\lambda}\right)^{\alpha-1} (1 + \alpha) < 0$$

alors

$$\left(\frac{s}{\lambda}\right)^{2\alpha-1} < \left(\frac{s}{\lambda}\right)^{\alpha-1} (1 + \alpha)$$

alors

$$\left(\frac{s}{\lambda}\right) < (1 + \alpha)^{\frac{1}{\alpha}} \implies s < \lambda(1 + \alpha)^{\frac{1}{\alpha}} = \mu \quad \text{donc} \quad s < \mu$$

Convexité de ϕ dans $[\mu, \infty]$

La fonction ϕ est une fonction convexe si la deuxième dérivée est positive $\phi''(s) > 0$ et nous avons

$$\begin{aligned} \phi''(s) &> 0 \iff \frac{1}{\lambda} \left(\frac{s}{\lambda}\right)^{\alpha-1} \left[\left(\frac{s}{\lambda}\right)^\alpha - (1 + \alpha)\right] g(s) > 0 \\ &\iff \frac{1}{\lambda} \left(\frac{s}{\lambda}\right)^{\alpha-1} \left[\left(\frac{s}{\lambda}\right)^\alpha - (1 + \alpha)\right] > 0 \end{aligned}$$

donc

$$\left(\frac{s}{\lambda}\right)^{\alpha-1} \left(\frac{s}{\lambda}\right)^{\alpha} - \left(\frac{s}{\lambda}\right)^{\alpha-1} (1+\alpha) > 0$$

alors

$$\left(\frac{s}{\lambda}\right)^{2\alpha-1} > \left(\frac{s}{\lambda}\right)^{\alpha-1} (1+\alpha)$$

alors

$$\left(\frac{s}{\lambda}\right) > (1+\alpha)^{\frac{1}{\alpha}} \implies s > \lambda(1+\alpha)^{\frac{1}{\alpha}} = \mu \text{ donc } s > \mu.$$

On distingue trois cas :

(1) Cas de $\alpha < 1$:

$$\lim_{s \rightarrow 0} \phi''(s) = -\infty$$

(2) Cas de $\alpha = 1$:

$$\phi''(0) = -\frac{2}{\lambda}$$

(3) Cas de $\alpha > 1$:

$$\phi''(0) = 0.$$

Le graphe de la dérivée seconde de ϕ sont représentés dans la Fig.2.5.

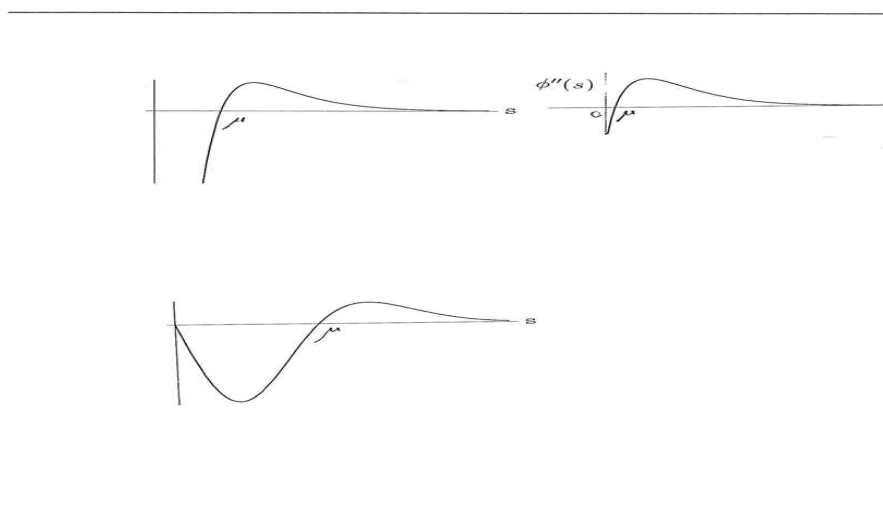


Figure 2.5 : (De gauche à droite) La dérivée seconde de ϕ pour $\alpha < 1$, $\alpha = 1$ et $\alpha > 1$

Chapitre 3

Résolution numérique du modèle

Dans ce chapitre, on va aborder la résolution numérique du modèle de la chaleur et de Perona-Malik. Les approximations par différences finies sont utilisées de manière prépondérante. En réalité, les images étant définies sur des grilles fixes, l'analyse numérique des modèles à travers des méthodes de différences finies, est la mieux adaptée. Nous allons voir dans un premier temps le cadre général de ces approches et ensuite, exposer les principes théoriques des schémas utilisés pour les EDPs présentées auparavant. Finalement, nous présentons quelques résultats numériques pour faire une comparaison entre le filtrage linéaire et le filtrage non linéaire.

3.1 Méthode des différences finies

La méthode consiste à remplacer les dérivées partielles par des différences divisées ou combinaisons de valeurs ponctuelles de la fonction en un nombre fini de points discrets ou noeuds du maillage.

Avantages : grande simplicité d'écriture et faible coût de calcul.

Inconvénients : limitation à des géométries simples, difficultés de prise en compte des conditions aux limites de type Neumann.

3.2 Approximations par différences finies

3.2.1 Notations

Toutes les EDPs analysées nécessitent la connaissance d'un domaine $\Omega = [0, m] \times [0, n]$ et des conditions initiales et aux limites.

La discrétisation du domaine est implicite. Dans la plupart des situations, une image est représentée sur une grille de points. Nous considérons dans ce chapitre des grilles régulières avec un espacement identique h selon les directions horizontale x et verticale y . Les valeurs de la fonction initiale sont données pour des échantillons (pixels) dans les deux directions. La dimension de l'image (en pixels) est $M \times N$; avec

$$M = \frac{m}{h} \text{ et } N = \frac{n}{h}$$

Mathématiquement, la fonction continue $u : \Omega \times [0, +\infty[\longrightarrow \mathbb{R}$ est remplacée par :

$$u(x, y, t) \longrightarrow u(ih, jh, t), \quad i = \{0, 1, \dots, N\}, \quad j = \{0, 1, \dots, M\} \quad (3.1)$$

Toutes les équations présentées sont des processus récursifs nécessitant une discrétisation en temps. Nous considérons des temps discrets uniformément distribués, multiples d'un pas de discrétisation en temps, noté Δt : La représentation à l'instant t de l'image numérique qui évolue est donnée par :

$$u(i, j, t) \longrightarrow u(i, j, n\Delta t) = u_{i,j}^n \quad (3.2)$$

qui est utilisée pour pointer sur les pixels.

En (3.2) nous notons par n le nombre d'itérations nécessaires pour atteindre l'échelle t : Le même raisonnement s'applique sur les conditions initiales :

$$u(x, y, 0) \longrightarrow u_{i,j}^0 \quad (3.3)$$

D'un point de vue pratique, dans le traitement des images par ordinateurs, les conditions aux limites ne sont pas très importantes. Elles ne sont imposées que pour affranchir des problèmes liés aux bords. Les positions sur une grille sont données en utilisant les deux directions cardinale et intercardinale, comme défini dans la figure **(3.1)**. L'idée est de simplifier la

notation des versions discrétisées des équations qui peuvent être plutôt désordonnées

NO	N	NE
O	C	E
SO	S	SE

Figure 3.1 : Directions sur une grille. Pour simplifier la notation, les deux directions cardinale et inter-cardinale sont utilisées. Ici O, N, E, S et C correspondent respectivement à l'Ouest, au Nord, à l'Est, au Sud, et au Centre..

avec :

$$C \equiv i, j, E \equiv i + 1, O \equiv i - 1, N \equiv j + 1, S \equiv j - 1$$

Systèmes de numérotation des pixels

Comme il a déjà été mentionné, nous pouvons nous référer à n'importe quel pixel de l'image en utilisant $u_{i,j}$, où $0 \leq i \leq m$; $0 \leq j \leq n$. Ici, le paramètre de discrétisation $h = (h_x, h_y)$ a été choisi pour que le domaine discrétisé soit $\Omega : [0, m] \times [0, n]$. Une autre façon particulièrement utile est de considérer l'image discrétisée comme un vecteur $u_I \in \mathbb{R}^N$. Maintenant, les composantes du vecteur sont u_I où $I \in \{1, \dots, N\}$ et N est le nombre de pixels dans l'image. Ce second schéma de numérotation est utile pour les descriptions algorithmiques et en notation matricielle, comme nous le verrons plus tard. La figure 2 représente à la fois l'ordre de déplacement des pixels dans les colonnes et les rangées. Selon le solveur, une traversée en colonnes ou en lignes, ou une combinaison des deux (par exemple, première colonne puis rangée), peut être utilisée.

1	4	7
2	5	8
3	6	9

Par colonne

1	2	3
4	5	6
7	8	9

par ligne

Figure.3.2 : Ordonnances de pixels dans les colonnes et les rangées. Ici, $I \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

voisinages de pixel

Afin de simplifier la notation, par exemple dans les descriptions algorithmiques, nous définissons les différents types de voisinage de pixels. Avec le voisinage de pixels nous voulons

dire pixels d'image u_J qui sont voisins d'un pixel d'intérêt u_I . Les voisinages sont légèrement différents selon que l'on parle d'un élément ou d'un solveur par blocs. Par solveur élémentaire, nous entendons un solveur itératif de type Jacobi ou Gauss-Seidel, qui cherchent la solution pour un seul élément à la fois. D'un autre côté, les solveurs de type bloc recherchent une solution pour un groupe d'éléments (ou un bloc). Cependant, comme les voisinages ont la même fonction dans les deux cas mentionnés ci-dessus, nous utilisons le même opérateur de voisinage pour désigner les voisins. Il devrait être clair pour la structure du solveur quel genre de voisinage est en question. $J \in N(I)$ désigne l'ensemble des voisins J de I .

3.2.2 Approximations des dérivées partielles

Les dérivées spatiales et temporelles sont déduites en partant des développements en série de Taylor en des points $(x + k, y + l, t + m)$. Afin de ne pas modifier le caractère local des EDPs, les dérivées premières sont remplacées par des approximations d'ordre 2 ou 1. La méthode des différences centrales utilise des approximations symétriques :

$$\frac{\partial}{\partial x}(u(x, y, t)) = D_x^0[u(x, y, t)] = \frac{u(x + k, y, t) - u(x - k, y, t)}{2|k|}$$

et des formules similaires pour y et t .

En considérant $x = ih, y = jh, t = n\Delta t$ et de déplacement de type $k = \pm h, l = \pm h, m = \pm \Delta t$ Ces formules s'expriment-pour les dérivées spatiales sous la forme simplifiée :

$$D_x^0(u_{i,j}^n) = \frac{u_{i+1,j}^n - u_{i-1,j}^n}{2h}, \quad D_y^0(u_{i,j}^n) = \frac{u_{i,j+1}^n - u_{i,j-1}^n}{2h}$$

Des approximations d'ordre 1 progressives et régressives peuvent également être utilisées :

$$\begin{aligned} D_x^+(u_{i,j}^n) &= \frac{u_{i+1,j}^n - u_{i,j}^n}{h}, & D_x^-(u_{i,j}^n) &= \frac{u_{i,j}^n - u_{i-1,j}^n}{h} \\ D_y^+(u_{i,j}^n) &= \frac{u_{i,j+1}^n - u_{i,j}^n}{h}, & D_y^-(u_{i,j}^n) &= \frac{u_{i,j}^n - u_{i,j-1}^n}{h} \end{aligned}$$

Dans toutes les méthodes ,la dérivée par rapport au temps est remplacée par une approximation d'ordre 1 progressive :

$$D_t^+(u_{i,j}^n) = \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t}$$

Les dérivées secondes se déduisent de manière similaire en utilisant des approximations d'ordre 2 :

$$\left\{ \begin{array}{l} \frac{\partial^2 u_{i,j}^n}{\partial x^2} = \frac{u_{i+1,j}^n + u_{i-1,j}^n - 2u_{i,j}^n}{h^2} = D_x^- [D_x^+(u_{i,j}^n)] \\ \frac{\partial^2 u_{i,j}^n}{\partial y^2} = \frac{u_{i,j+1}^n + u_{i,j-1}^n - 2u_{i,j}^n}{h^2} = D_y^- [D_y^+(u_{i,j}^n)] \\ \frac{\partial^2 u_{i,j}^n}{\partial x \partial y} = \frac{u_{i+1,j+1}^n + u_{i-1,j-1}^n - u_{i+1,j-1}^n - u_{i-1,j+1}^n}{4h^2} \end{array} \right.$$

La forme numérique du laplacien la plus utilisée correspond à :

$$\Delta u(i, j, t) = \Delta u_{i,j}^n = \frac{\partial^2 u_{i,j}^n}{\partial x^2} + \frac{\partial^2 u_{i,j}^n}{\partial y^2} = D_x^- [D_x^+(u_{i,j}^n)] + D_y^- [D_y^+(u_{i,j}^n)]$$

3.3 Résolution numérique du modèle de filtrage linéaire - équation de la chaleur 2-D

Soit le problème de l'équation de la chaleur :

$$\frac{\partial u}{\partial t} = \Delta u$$

Notation

On note Δt le pas de discrétisation en temps, et $h = 1$, le pas de discrétisation en espace (on prend le même pas selon les deux axes). $u_{i,j}^n$ désigne la valeur de $u(n\Delta t, ih, jh)$

3.3.1 Schémas numériques

On peut utiliser deux approches pour discrétiser cette équation de la chaleur. La première **explicite** et la deuxième **implicite**

Schéma explicite

On a

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} = \frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{h^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{h^2}$$

$$u_{i,j}^{n+1} - u_{i,j}^n = \Delta t (u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - 4u_{i,j}^n)$$

alors

$$u_{i,j}^{n+1} = \Delta t (u_{i+1,j}^n + u_{i-1,j}^n) + \Delta t (u_{i,j+1}^n + u_{i,j-1}^n) + (1 - 4\Delta t) u_{i,j}^n$$

avec i, j variants de 1 à $N - 1$

Schéma implicite

On utilise un schéma implicite, comme le suivant :

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} = \frac{u_{i+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i-1,j}^{n+1}}{h^2} + \frac{u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}}{h^2}$$

$$u_{i,j}^{n+1} - u_{i,j}^n = \Delta t(u_{i+1,j}^{n+1} + u_{i-1,j}^{n+1} + u_{i,j+1}^{n+1} + u_{i,j-1}^{n+1} - 4u_{i,j}^{n+1})$$

donc

$$u_{i,j}^{n+1} - u_{i,j}^n = \Delta t(u_{i+1,j}^{n+1} + u_{i-1,j}^{n+1}) + \Delta t(u_{i,j+1}^{n+1} + u_{i,j-1}^{n+1}) - 4\Delta t u_{i,j}^{n+1}$$

Pour ce schéma implicite, chaque pas de temps nécessite la résolution d'un système linéaire

$$(1 + 4\Delta t)u_{i,j}^{n+1} - \Delta t(u_{i+1,j}^{n+1} + u_{i-1,j}^{n+1}) - \Delta t(u_{i,j+1}^{n+1} + u_{i,j-1}^{n+1}) = u_{i,j}^n$$

3.3.2 Résultat numérique

Nous utiliserons image suivante pour tester nos algorithmes.

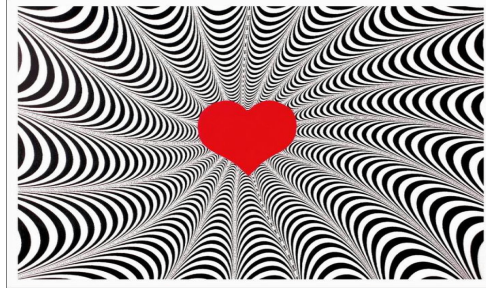


Figure 3.3 : Image originale

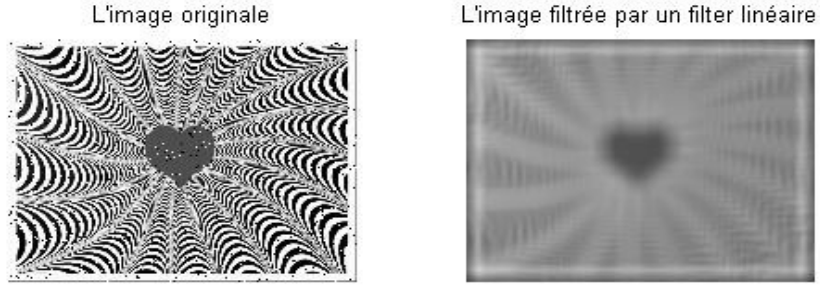


Figure 3.4 : Image bruitée(à gauche) filtrée par un filtre linéaire(à droite)

3.4 Résolution numérique du modèle de filtrage non linéaire - équation de Perona-Malik 1-D

Soit le modèle de Perona-Malik unidimensionnel

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(g \left(\left| \frac{\partial u}{\partial x} \right| \right) \frac{\partial u}{\partial x} \right)$$

En utilisant une discrétisation progressive en temps l'équation s'écrit :

$$\frac{\partial u}{\partial t} \approx \frac{u^{n+1} - u^n}{\Delta t} = \frac{\partial}{\partial x} \left[g \left(\left| \frac{\partial u}{\partial x} \right| \right) \cdot \frac{\partial u}{\partial x} \right]$$

Si les dérivées spatiales sont calculées à l'échelle $n\Delta t$ le schéma s'appelle explicite en temps (ou Euler) :

$$\frac{\partial u}{\partial t} \approx \frac{u^{n+1} - u^n}{\Delta t} = \frac{\partial}{\partial x} \left[g \left(\left| \frac{\partial u^n}{\partial x} \right| \right) \cdot \frac{\partial u^n}{\partial x} \right]$$

Dans ce cas la solution à l'échelle $(n+1)\Delta t$ est complètement déterminée par les valeurs calculées à l'instant $n\Delta t$:

$$u^{n+1} = u^n + \Delta t \cdot \frac{\partial}{\partial x} \left[g \left(\left| \frac{\partial u^n}{\partial x} \right| \right) \cdot \frac{\partial u^n}{\partial x} \right]$$

Ce type de schéma est utilisé de manière prépondérante pour la résolution des EDPs. Son principal avantage est la simplicité, même si, en général, il nécessite un grand nombre d'ité-

rations pour converger. Le schéma est dit semi-implicite si la solution u^{n+1} est calculée en résolvant un système d'équations avec des coefficients qui dépendent seulement de la solution à l'instant $n\Delta t$

$$\frac{\partial u}{\partial t} \approx \frac{u^{n+1} - u^n}{\Delta t} = \frac{\partial}{\partial x} \left[g \left(\left| \frac{\partial u^{n+1}}{\partial x} \right| \right) \cdot \frac{\partial u^n}{\partial x} \right]$$

En discrétisant les dérivées spatiales on obtient :

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{1}{h} [g [|D_x^+(u_i^n)|] D_x^+(u_i^{n+1}) - g [|D_x^-(u_i^n)|] D_x^-(u_i^{n+1})]$$

En considérant $h = 1$ et en regroupant les termes :

$$u_i^{n+1}(1 + \Delta t(g(|D_x^+(u_i^n)|) + g(|D_x^-(u_i^n)|))) - \Delta t g(|D_x^+(u_i^n)|) u_{i+1}^{n+1} - \Delta t g(|D_x^-(u_i^n)|) u_{i-1}^{n+1} = u_i^n$$

Pour une longueur fixe de la suite numérique (N) ; en introduisant les notations

$$u^k = [u_0^k \ u_1^k \dots u_N^k]^T$$

$$A[u^n]_{N \times N} = [a_{i,k}(u^n)] = \begin{cases} -g(|D_x^-(u_i^n)|) & \text{pour } k = i - 1 \\ -g(|D_x^+(u_i^n)|) & \text{pour } k = i + 1 \\ g(|D_x^-(u_i^n)|) + g(|D_x^+(u_i^n)|) & \text{pour } k = 0 \\ 0 & \text{en reste} \end{cases}$$

se réduit au système linéaire :

$$u^{n+1} (I_N + \Delta t A[u^n]) = u^n$$

La résolution du système est généralement réalisée en utilisant des méthodes de relaxation telles que les méthodes Jacobi ou Gauss-Seidel. l'avantage de ces méthodes est relatif : même si elles sont stables pour n'importe quel pas Δt ; en partant d'une solution initiale, plusieurs itérations sont nécessaires pour converger vers la solution du système et, de plus, le nombre de calculs nécessaires pour chaque itération est plus grand que celui d'une méthode explicite. Une comparaison intéressante entre les schémas explicite et semi-implicite est réalisée par Weickert [16]. En comparant les erreurs introduites par les approximations, les résultats montrent qu'en terme d'erreur, le schéma explicite sous des contraintes de stabilité est meilleur.

Shéma numérique :

En utilisant le schéma de discrétisation explicite du modèle de Perona et Malik en 1-D en tous points (i, j) et à l'instant $n + 1$ est le suivant :

$$u_{i,j}^{n+1} = u_{i,j}^n + \Delta t \left[g(|u_{i+1,j}^n - u_{i,j}^n|) \cdot (u_{i+1,j}^n - u_{i,j}^n) - g(|u_{i,j}^n - u_{i-1,j}^n|) \cdot (u_{i,j}^n - u_{i-1,j}^n) + g(|u_{i,j+1}^n - u_{i,j}^n|) \right]$$

Les résultats que nous analyserons obtenus en considérant la fonction de convergence g définie par :

$$g : \mapsto \frac{1}{1 + \frac{x^2}{\lambda^2}}$$

où λ est un paramètre de conservation des détails.

3.5 Résolution numérique du modèle de filtrage non linéaire - équation de Perona-Malik 2-D

La formule de base décrivant une diffusion de valeur scalaire est :

$$\frac{\partial}{\partial t}(u(x, y, t)) = \text{DIV}(g(x, y, t) \nabla u(x, y, t)) \quad (3.4)$$

avec, $g(x, y, t)$ qui définit les poids de diffusion (scalaire), et $\nabla := \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right]$ est le gradient spatial. Puisque $g(x, y, t)$ est une fonction de t , la diffusion est non-linéaire. Cela signifie que les poids de diffusion définis par la fonction $g(x, y, t)$ changent par rapport au temps t . Un possible $g(x, y, t)$ est, par exemple

$$g(x, y, t) = \frac{1}{1 + \left(\frac{\|\nabla u(x, y, t)\|}{\lambda} \right)^2} \quad (3.5)$$

où $\|\nabla u(x, y, t)\|$ se réfère à la magnitude des dérivées de l'image (Elle obtient une plus grande valeur quand il y a un changement dans l'information d'image). λ est un coefficient qui est utilisé pour contrôler les grandeurs de poids de diffusion par rapport à l'image dérivée.

– **Schéma explicite**

$$\frac{(u)^{n+1} - u^n}{\Delta t} = \text{DIV}(g^n \nabla(u)^n) \quad (3.6)$$

– **Schéma implicite**

$$\frac{(u)^{n+1} - u^n}{\Delta t} = \text{DIV}(g^n \nabla(u)^{n+1}) \quad (3.7)$$

d'où $g^n = g(x, y, t)$. En d'autres termes, nous utilisons les valeurs de g disponibles à l'instant n , lors de la résolution d'une nouvelle valeur à l'instant $n + 1$.

3.5.1 Discrétisation de l'opérateur DIV

Maintenant que nous savons comment approcher les dérivées de premier et de deuxième ordre, nous pouvons discrétiser l'opérateur divergence DIV . Conceptuellement, nous avons deux cas différents :

$$DIV(\nabla u) \tag{3.8}$$

$$DIV(g(x, y, t)\nabla u)$$

Ici, l'interprétation physique de la divergence a un sens, celle de la diffusion. Dans le cas de $DIV(\nabla u)$, la diffusivité est la même dans chaque direction, alors que dans le cas de $DIV(g(x, y, t)\nabla u)$, la diffusivité est définie (ou contrôlée) par la fonction g et elle n'est pas nécessairement la même dans toutes les directions. Mathématiquement, pour une fonction vectorielle différentiable $F = \vec{U}_i + \vec{V}_j$, l'opérateur de divergence est défini comme :

$$DIV(F) = \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} \tag{3.9}$$

En d'autres termes, la divergence est une somme de dérivées partielles d'une fonction vectorielle différentiable. Par conséquent, dans notre cas, nous avons ce qui suit.

$$DIV(\nabla u) = \frac{\partial}{\partial x}(u_x) + \frac{\partial}{\partial y}(u_y) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \Delta u \tag{3.10}$$

$$DIV(g(x, y, t)\nabla u) = \frac{\partial}{\partial x}(g(x, y, t)u_x) + \frac{\partial}{\partial y}(g(x, y, t)u_y) = \nabla g \cdot \nabla u + g\Delta u \tag{3.11}$$

Maintenant, en utilisant simplement les différences finies présentées ci-dessus, une façon de discrétiser les termes de divergence dans (3.10) utilise la différence centrale. Premièrement, nous appliquons la différence centrale, puis les différences avant et arrière pour approximer les dérivées correspondantes. L'objectif ici est de réaliser que $(u_x)(x + 0.5, y, t)$ est en fait la différence directe $D_x^+ u(x)$, tandis que $(u_x)(x - 0.5, y, t)$ est la différence arrière $D_x^- u(x)$. Les équations (3.12) et (3.14) montrent respectivement les discrétisations pour $DIV(\nabla u)$ et $DIV(g(x, y, t)\nabla u)$.

$$\begin{aligned}
\frac{\partial}{\partial x}(u_x)(x, y, t) + \frac{\partial}{\partial y}(u_y)(x, y, t) &= (u_x)(x + 0.5, y, t) - (u_x)(x - 0.5, y, t) \\
&\quad + (u_y)(x, y + 0.5, t) - (u_y)(x, y - 0.5, t) \\
&= u(x + 1, y, t) - u(x, y, t) + u(x - 1, y, t) - u(x, y, t) \\
&\quad + u(x, y + 1, t) - u(x, y, t) + u(x, y - 1, t) - u(x, y, t) \\
&= \nabla_E u + \nabla_O u + \nabla_S u + \nabla_N u
\end{aligned} \tag{3.12}$$

d'où $\nabla_{\{O, N, E, S\}} u$ indique la différence dans les directions données par O, N, E, S . Comme il a déjà été mentionné, d'abord nous appliquons la différence centrale de premier ordre sur $u_x(x, y, t)$, et obtenons aussi $D_0^x u_x(x, y, t) = (u_x)(x + 0.5, y, t) - (u_x)(x - 0.5, y, t)$.

$$\begin{aligned}
\frac{\partial}{\partial x}(gu_x)(x, y, t) + \frac{\partial}{\partial y}(gu_y)(x, y, t) &= (gu_x)(x + 0.5, y, t) - (gu_x)(x - 0.5, y, t) \\
&\quad + (gu_y)(x, y + 0.5, t) - (gu_y)(x, y - 0.5, t) \\
&= g(x + 0.5, y, t)(u(x + 1, y, t) - u(x, y, t)) \\
&\quad + g(x - 0.5, y, t)(u(x - 1, y, t) - u(x, y, t)) \\
&\quad + g(x, y + 0.5, t)(u(x, y + 1, t) - u(x, y, t)) \\
&\quad + g(x, y - 0.5, t)(u(x, y - 1, t) - u(x, y, t)) \\
&= g_E \nabla_E u + g_O \nabla_O u + g_S \nabla_S u + g_N \nabla_N u
\end{aligned} \tag{3.14}$$

où $g_{\{W; N; E; S\}}$ désigne la diffusivité dans les directions données par $W; N; E; S$. Comme on peut l'observer à partir de l'équation (3.14), nous devons approximer la diffusivité entre les pixels. Une approximation "2 points" simple serait la moyenne entre pixels voisins. Par exemple $g(x + 0.5, y, t) = [g(x + 1, y, t) + g(x, y, t)]/2$. Une approximation plus précise, conduisant à des meilleurs résultats, est une approximation "6 points" de Brox [3]

3.5.2 Equation de diffusion non linéaire discrétisée

Pour résoudre l'équation (3.7), nous devons discrétiser l'opérateur de divergence, nous commençons par marquer les positions des pixels d'intérêt avec (i, j) :

$$\frac{(u)_{i,j}^{n+1} - (u)_{i,j}^n}{\Delta t} = DIV(g^n \nabla u^{n+1}) \tag{3.15}$$

Après cela, nous discrétisons l'opérateur DIV , comme indiqué par (3.14), et obtenons ce qui suit :

$$\begin{aligned}
(u)_{i,j}^{n+1} - (u)_{i,j}^n &= \Delta t g_N^n ((u)_{i-1,j}^{n+1} - (u)_{i,j}^{n+1}) \\
&+ \Delta t g_S^n ((u)_{i+1,j}^{n+1} - (u)_{i,j}^{n+1}) \\
&+ \Delta t g_O^n ((u)_{i,j-1}^{n+1} - (u)_{i,j}^{n+1}) \\
&+ \Delta t g_E^n ((u)_{i,j+1}^{n+1} - (u)_{i,j}^{n+1})
\end{aligned} \tag{3.16}$$

La seule chose à faire est d'arranger les termes :

$$\begin{aligned}
(u)_{i,j}^{n+1} (1 + \Delta t (g_N^n + g_S^n + g_O^n + g_E^n)) &= (u)_{i,j}^n \\
&+ \Delta t g_N^n ((u)_{i-1,j}^{n+1}) \\
&+ \Delta t g_S^n ((u)_{i+1,j}^{n+1}) \\
&+ \Delta t g_O^n ((u)_{i,j-1}^{n+1}) \\
&+ \Delta t g_E^n ((u)_{i,j+1}^{n+1})
\end{aligned} \tag{3.17}$$

3.5.3 Forme matricielle

Alors, l'équation (3.17) montre l'équation à résoudre pour une position de pixel (i, j) . Nous pouvons écrire les équations du système au format matrice/vecteur, couvrant l'image entière, comme indiqué en (3.19). Maintenant, les composantes du vecteur sont u_I où $I \in \{1, \dots, N\}$ et N est le nombre de pixels dans l'image. Nous utilisons I, J pour marquer les positions dans la matrice A du système donnée en (3.19). Ceci est fait pour véhiculer clairement l'idée que les domaines des images discrétisées et des matrices système sont différents. Si le domaine de l'image discrétisée est $\Omega : [1, m] \times [1, n]$ (image discrète avec m colonnes et n lignes), la matrice système A est définie sur $[m \cdot n] \times [m \cdot n]$ (ici \cdot désigne la multiplication). Maintenant, nous pouvons écrire la formulation d'Euler, semi-implicite, dans un format vectoriel / matriciel comme suit :

$$\frac{u^{n+1} - u^n}{\Delta t} = A(u^n) u^{n+1} \tag{3.18}$$

d'où $u := u_I$ avec $I = [1 \dots N]$. La matrice du système $A(u^n)$ est définie comme suit :

$$A(u^n) = [a_{I,J}^n]$$

$$a_{I,J}^n := \begin{cases} -\Delta t g_{J \sim I}^n & [J \in N(I)], \\ 1 + \sum_{\substack{J \in N^-(I) \\ J \in N^+(I)}} \Delta t g_{J \sim I}^n & (J = I), \\ 0 & \text{Autrement} \end{cases} \quad (3.19)$$

où $g_{J \sim I}^n$ fait référence au poids de «diffusion» entre les pixels J et I à l'instant n . En d'autres termes, ils se réfèrent au $g_{\{W;N;E;S\}}$ vu précédemment. L'équation (3.20) donne un exemple de la façon dont la matrice de système A ressemblerait à une image de taille 3×3 . Ici, C et N sont des matrices de blocs qui se réfèrent aux matrices «centrales» et «voisines», de manière correspondante.

$$\begin{aligned} A &= \begin{bmatrix} C & N & 0 \\ N & C & N \\ 0 & N & C \end{bmatrix} \\ C &= \begin{bmatrix} \sum_{\substack{J \in N^-(I) \\ J \in N^+(I)}} \Delta t g_{J \sim I}^n & -\Delta t g_{J \sim I}^n & 0 \\ -\Delta t g_{J \sim I}^n & \sum_{\substack{J \in N^-(I) \\ J \in N^+(I)}} \Delta t g_{J \sim I}^n & -\Delta t g_{J \sim I}^n \\ 0 & -\Delta t g_{J \sim I}^n & \sum_{\substack{J \in N^-(I) \\ J \in N^+(I)}} \Delta t g_{J \sim I}^n \end{bmatrix} \\ N &= \begin{bmatrix} -\Delta t g_{J \sim I}^n & 0 & 0 \\ 0 & -\Delta t g_{J \sim I}^n & 0 \\ 0 & 0 & -\Delta t g_{J \sim I}^n \end{bmatrix} \end{aligned} \quad (3.20)$$

A partir de (3.19) nous pouvons voir à quoi ressemble la matrice A Pour une image de taille 3×3 : c'est une matrice carrée tridiagonale, de taille 9×9 , avec des composantes non nulles seulement sur la diagonale principale et sur les diagonales adjacentes à cela. Par conséquent, à moins que l'image soit très petite, il est impossible de résoudre le système

en inversant A directement. Au lieu de cela, nous recherchons une solution en utilisant des méthodes itératives.

Dans ce qui suit, nous pouvons voir comment A ressemble à une «notation» différente. Il est intéressant de voir comment changer l'ordre de déplacement, les voisins sur les diagonales à côté de la diagonale principale. Les sous-matrices avec fond gris se réfèrent à la sous-matrice C donnée dans l'équation (3.20)

C	S	0	E	0	0	0	0	0	0	0	0
N	C	S	0	E	0	0	0	0	0	0	0
0	N	C	0	0	E	0	0	0	0	0	0
W	0	0	C	S	0	E	0	0	0	0	0
0	W	0	N	C	S	0	E	0	0	0	0
0	0	W	0	N	C	0	0	E	0	0	0
0	0	0	W	0	0	C	S	0	0	0	0
0	0	0	0	W	0	N	C	S	0	0	0
0	0	0	0	0	W	0	N	C	0	0	0

(a) Traversée en colonne

C	E	0	S	0	0	0	0	0	0	0	0
W	C	E	0	S	0	0	0	0	0	0	0
0	W	C	0	0	S	0	0	0	0	0	0
N	0	0	C	E	0	S	0	0	0	0	0
0	N	0	W	C	E	0	S	0	0	0	0
0	0	0	0	W	C	0	0	S	0	0	0
0	0	0	N	0	0	C	E	0	0	0	0
0	0	0	0	N	0	W	C	E	0	0	0
0	0	0	0	0	0	0	W	C	0	0	0

(b) Traversée en ligne

3.5.4 Résultat numérique

Alors après avoir fait le bruitage sur la figure (3.3) on filtre cette image en utilisant le filtre de Perona-Malik

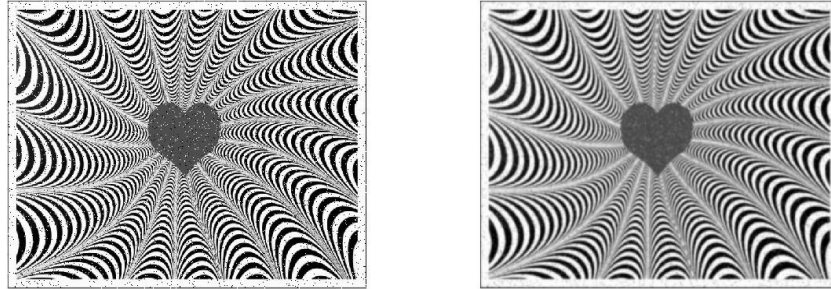


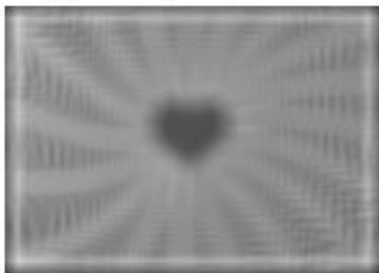
Figure 3. 5 Image bruitée «poivre et sel» (à gauche), Image filtrée par Perona-Malik (à droite)

3.6 Comparaison entre l'EDP de Perona-Malik et l'équation de chaleur

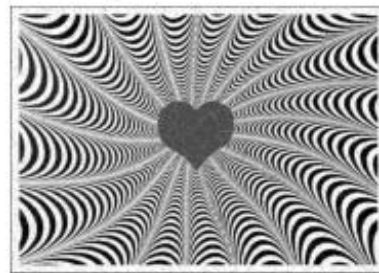
L'image originale



L'image filtrée par un filter linéaire



L'image filtrée par un filter nonlinéaire (PM)



On remarque que la diffusion non linéaire est mieux que la diffusion linéaire

Conclusion

Dans ce mémoire, nous avons étudié en détails le modèle de Perona-Malik.

Théoriquement, nous avons étudié les propriétés qualitatives de la solution du problème dans le cas unidimensionnel , en donnant quelques résultats théoriques. Un résultat numérique est ensuite réalisé concernant le modèle de Perona-Malik et l'équation de la chaleur en 2-D, en mettant en évidence la pertinence du modèle de Perona-Malik par rapport à celui de la chaleur.

Bibliographie

- [1] S. ABED, *Image processing by the partial derivative equations*, Thesis of Magister, University of EL HADJ LAKHDAR BATNA, 2012.
- [2] B. BENHAMOUDA, *Parameter adaptation for nonlinear diffusion in image processing*, master thesis, Dept. of Mathematics, University of Kaiserslautern, P.O. Box 3049, 67653, Kaiserslautern, Germany, 1994.
- [3] T. BROX. *From Pixels to Regions : Partial Differential Equations in Image Analysis*. PhD thesis, Saarland University, Saarbrücken, Germany, 2005.
- [4] M.J. BLACK, G. SAPIRO, D.H. MARIMONT AND D. HEEGER, (1998) '*Robust anisotropic diffusion*', IEEE Trans, on Image Process, Vol.7, pp.421-432.
- [5] P. CHARBONNIER, L. BLANC-FERAUD, G AUBERT and M. BARLAUD, (1994) '*Two deter-ministic half-quadratic regularization algorithms for computed imaging*', Proc. IEEE Int. Conf. Image Processing (ICIP-94, Austin, Nov.13-16, 1994) IEEE Computer Society Press, Los Alamitos, Vol.2, pp.168-172.
- [6] R. CHOUDER. *Auto-similarité et contour d'image*, thèse de doctorat, Université Mohamed Boudiaf M'sila, 2018.
- [7] *Cours de traitement d'image Master EDPs et Applications*, Université de M'sila. BEN-HAMIDOUCHE. N, 2017.
- [8] S. KH. DZHU MAGAZIVA, *Numerical Study of a P.D.E, U.S.S.R. Comput, Math. Phys.*, vol. 23, pp. 45-49, 1983.
- [9] B.FISCHL, AND E.L. SCHWARTZ, (1997) '*Learning an integral equation approximation to nonlinear anisotropic diffusion in image processing*'. IEEE Trans, on Pattern Anal. Machine Intelligence Vol.19, pp.342-352.

- [10] P. PERONA and J. MALIK, (1987) '*Scale-space and edge detection using anisotropic diffusion*', IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.12, No.7, pp.629–639.
- [11] M. H. PROTTER and H. F. WEINBERGER, *Maximum principles in Differential Equations*, Prentice-Hall, 1967.
- [12] J. RALLI. *Fusion and Regularisation of Image Information in Variational Correspondence Methods*. PhD thesis, University of Granada, Spain, 2011.
- [13] U. TROTTEBERG, C. OOSTERLEE, and A. SCHULLER. *Multigrid*. Academic Press, 2001.
- [14] J. WEICKERT, *Anisotropic Diffusion in Image Processing*, Ph.D. thesis University of Kaiserslautern, to be filed, 1994, 639-739.
- [15] J. WEICKERT, (1998) '*Anisotropic Diffusion in Image Processing*'. Stuttgart : BG : Teubner Stuttgart.
- [16] J. WEICKERT, B. M. TER HAAR ROMENY, and M. A. VIERGEVER. *Efficient and reliable schemes for nonlinear diffusion filtering*. IEEE Transactions on Image Processing, 7 :398-410, 1998.
- [17] YOU, Y., XU, W., TANNENBAUM, A. and KAVEH, M. (1996) '*Behavioral analysis of anisotropic diffusion in image processing*'. IEEE Trans, on Image Process Vol.5, pp.1539-1552.

Annexe

Dans ce mémoire on a utilise des méthodes numériques pour résoudre le problème de Perona-Malik et le modèle de l'équation de la chaleur et on a utilise la programmation de ces méthodes avec matlab tel que :

Programme 1 : Résolution du modèle de Perona-Malik en 2-D

```
function diff_im = anisodiff2D(im, num_iter, delta_t, kappa, option)
%ANISODIFF2D Conventional anisotropic diffusion
% DIFF_IM = ANISODIFF2D(IM, NUM_ITER, DELTA_T, KAPPA, OPTION) per-
foms
% conventional anisotropic diffusion (Perona & Malik) upon a gray scale
% image. A 2D network structure of 8 neighboring nodes is considered for
% diffusion conduction.
%
% ARGUMENT DESCRIPTION :
% IM - gray scale image (MxN).
% NUM_ITER - number of iterations.
% DELTA_T - integration constant (0 <= delta_t <= 1/7).
% Usually, due to numerical stability this
% parameter is set to its maximum value.
% KAPPA - gradient modulus threshold that controls the conduction.
% OPTION - conduction coefficient functions proposed by Perona & Malik :
% 1 -  $c(x,y,t) = \exp(-(nabla I/kappa).^2)$ ,
% privileges high-contrast edges over low-contrast ones.
% 2 -  $c(x,y,t) = 1./(1 + (nabla I/kappa).^2)$ ,
% privileges wide regions over smaller ones.
```

```

%
% OUTPUT DESCRIPTION :
% DIFF_IM - (diffused) image with the largest scale-space parameter.
%
% Example
% -----
% s = phantom(512) + randn(512);
% num_iter = 15;
% delta_t = 1/7;
% kappa = 30;
% option = 2;
% ad = anisodiff2D(s,num_iter,delta_t,kappa,option);
% figure, subplot 121, imshow(s,[]), subplot 122, imshow(ad,[])
%
% See also anisodiff1D, anisodiff3D.
% References :
% P. Perona and J. Malik.
% Scale-Space and Edge Detection Using Anisotropic Diffusion.
% IEEE Transactions on Pattern Analysis and Machine Intelligence,
% 12(7) :629-639, July 1990.
%
% G. Grieg, O. Kubler, R. Kikinis, and F. A. Jolesz.
% Nonlinear Anisotropic Filtering of MRI Data.
% IEEE Transactions on Medical Imaging,
% 11(2) :221-232, June 1992.
%
% MATLAB implementation based on Peter Kovese's anisodiff(.) :
% P. D. Kovese. MATLAB and Octave Functions for Computer Vision and Image Pro-
cessing.
% School of Computer Science & Software Engineering,
% The University of Western Australia. Available from :
% <http://www.csse.uwa.edu.au/~pk/research/matlabfns/>.

```



```

%
% Credits :
% Daniel Simoes Lopes
% ICIST
% Instituto Superior Tecnico - Universidade Tecnica de Lisboa
% danlopes (at) civil ist utl pt
% http ://www.civil.ist.utl.pt/~danlopes
%
% May 2007 original version.
% Convert input image to double.
im = double(im);
% PDE (partial differential equation) initial condition.
diff_im = im;
% Center pixel distances.
dx = 1;
dy = 1;
dd = sqrt(2);
% 2D convolution masks - finite differences.
hS = [0 0 0; 0 -1 0; 0 1 0];
hE = [0 0 0; 0 -1 1; 0 0 0];
hW = [0 0 0; 1 -1 0; 0 0 0];
hNE = [0 0 1; 0 -1 0; 0 0 0];
hSE = [0 0 0; 0 -1 0; 0 0 1];
hSW = [0 0 0; 0 -1 0; 1 0 0];
hNW = [1 0 0; 0 -1 0; 0 0 0];
% Anisotropic diffusion.
for t = 1 :num_iter
% Finite differences. [imfilter(...,'conv') can be replaced by conv2(...,'same')]
nablaN = imfilter(diff_im,hN,'conv');
nablaS = imfilter(diff_im,hS,'conv');
nablaW = imfilter(diff_im,hW,'conv');
nablaE = imfilter(diff_im,hE,'conv');

```

```

nablaNE = imfilter(diff_im,hNE,'conv');
nablaSE = imfilter(diff_im,hSE,'conv');
nablaSW = imfilter(diff_im,hSW,'conv');
nablaNW = imfilter(diff_im,hNW,'conv');
% Diffusion function.
if option == 1
cN = exp(-(nablaN/kappa).^2);
cS = exp(-(nablaS/kappa).^2);
cW = exp(-(nablaW/kappa).^2);
cE = exp(-(nablaE/kappa).^2);
cNE = exp(-(nablaNE/kappa).^2);
cSE = exp(-(nablaSE/kappa).^2);
cSW = exp(-(nablaSW/kappa).^2);
elseif option == 2
cN = 1./(1 + (nablaN/kappa).^2);
cS = 1./(1 + (nablaS/kappa).^2);
cW = 1./(1 + (nablaW/kappa).^2);
cE = 1./(1 + (nablaE/kappa).^2);
cNE = 1./(1 + (nablaNE/kappa).^2);
cSE = 1./(1 + (nablaSE/kappa).^2);
cSW = 1./(1 + (nablaSW/kappa).^2);
cNW = 1./(1 + (nablaNW/kappa).^2);
end
% Discrete PDE solution.
diff_im = diff_im + ...
delta_t*(...
(1/(dy^2))*cN.*nablaN + (1/(dy^2))*cS.*nablaS + ...
(1/(dx^2))*cW.*nablaW + (1/(dx^2))*cE.*nablaE + ...
(1/(dd^2))*cNE.*nablaNE + (1/(dd^2))*cSE.*nablaSE + ...
(1/(dd^2))*cSW.*nablaSW + (1/(dd^2))*cNW.*nablaNW );
% Iteration warning.
fprintf('\rIteration %d\n',t);

```

end

Programme 2

```
clear;clc;clf;
%K=imread('4.jpg');
K=imread('cameraman.png');
%K=imread('3.jpg');
%K=rgb2gray(K);
K=im2double(K);
K=imnoise(K,'salt & pepper',0.05);%K=imnoise(K,'salt & pepper',0.05);
%K = imnoise(K,'gaussian',0,0.005);
figure(1); subplot(1,2,1); imshow(K)
%diff=anisodiff2D(K, 20, 0.025, 80, 2)
diff = anisodiff(K, 50, 20, 0.025, 2);
figure(1); subplot(1,2,2); imshow(diff)
```

Programme 3 : Résolution du modèle de la chaleur en 2-D

```
clear;clc;clf;
K=imread('4.jpg');
%K=imread('cameraman.png');
%K=imread('3.jpg');
K=rgb2gray(K);
K=im2double(K);
K=imnoise(K,'salt & pepper',0.05);%K=imnoise(K,'salt & pepper',0.05);
%K = imnoise(K,'gaussian',0,0.005);
figure(1); subplot(2,2,1); imshow(K); title('L"image originale');
% La diffusion nonlineaire
%diff=anisodiff2D(K, 20, 0.025, 80, 2)
diff = anisodiff(K, 50, 20, 0.025, 2);
% La diffusion lineaire
```

```
A=fspecial('gaussian',[50 50],150);  
Ldiff=imfilter(K,A);  
figure(1); subplot(2,2,3);  
imshow(Ldiff);title('L"image filtrée par un filter linéaire');
```