

Delphine Reynard

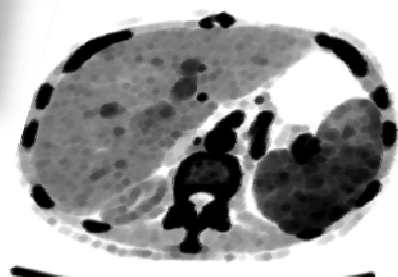
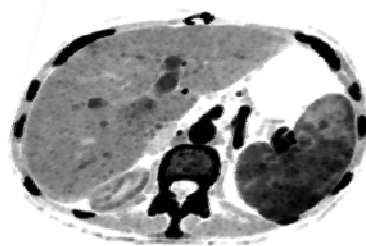
Sylvain Chagué

Ecole Nationale
Supérieure des Mines
SAINT-ETIENNE

Mini-projet UP1 : Imagerie et modélisation

IMAGERIE

TRAITEMENT DES IMAGES PAR EDP APPLICATION A LA RESTAURATION D'IMAGES



Novembre 2008

Tuteur : Johan Debayle

Table des matières

INTRODUCTION ET OBJECTIFS	3
1. SYNTHÈSE DES EDP UTILISÉES EN TRAITEMENT D'IMAGE	4
1.1. DIFFUSION ISOTROPE	4
1.2. DIFFUSION ANISOTROPE	5
1.3. DIFFUSION DÉGÉNÉRÉE	8
1.3.1. DILATATION ET ÉROSION	8
1.3.2. DIFFUSION ANISOTROPE SELON LES LIGNES DE NIVEAU	10
1.4. EDP ET CARTE DE DISTANCES	11
2. APPLICATION A LA RESTAURATION D'IMAGE	11
2.1. ANALYSE QUALITATIVE	12
2.2. PROTOCOLE DE MESURE DU DÉBRUITAGE	13
2.3. RESULTATS	13
2.3.1. IMAGE LENA BRUITÉE AVEC UN BRUIT GAUSSIEN	13
2.3.2. IMAGE LENA BRUITÉE PAR UN BRUIT POIVRE ET SEL	15
2.3.3. COUPE HISTOLOGIQUE D'UNE JAMBE BRUITÉE PAR UN BRUIT GAUSSIEN	16
2.3.4. DIFFUSION LINÉAIRE SELON LES LIGNES DE NIVEAUX	17
2.4. CRITIQUES DE LA MÉTHODE UTILISÉE	19
CONCLUSION ET PERSPECTIVES	19
REFERENCES BIBLIOGRAPHIQUES	20
ANNEXE 1 : SCHEMAS NUMÉRIQUES UTILISÉS	21
ANNEXE 2 : CODE MATLAB (DIFFUSION LINÉAIRE, MALIK ET PERONA)	23
ANNEXE 3 : CODE MATLAB (DILATATION ET ÉROSION)	24

Introduction et objectifs

Les images numériques sont des objets complexes qui sont difficile à analyser (détection, reconnaissance) d'une manière complètement automatique. Les objets présents peuvent être à des échelles très différentes et les détails fins viennent perturber les objets de plus grande taille. Les procédés d'acquisition et de stockage sont sources de bruit, d'une quantification et éventuellement de déformations dues à la compression.

C'est pourquoi, pour simplifier l'analyse des images, il est parfois nécessaire de dégrader volontairement les images afin de ne garder que les détails d'une certaine échelle. C'est ainsi qu'est apparu la notion d'espaces d'échelle. L'utilisation des équations aux dérivées partielles en traitement d'images est très souvent liée à ces problématiques. Le champ applicatif des équations aux dérivées partielles est très grand : filtrage, restauration, rehaussement, agrandissement, suppression d'objets indésirables, etc. Dans le cadre de ce mini-projet, nous nous sommes essentiellement intéressés à l'utilisation des EDP dans un but de restauration d'images. Le but de la restauration d'image est d'améliorer la qualité de cette image en atténuant/annulant les dégradations. Nous nous intéresserons aux dégradations dues au bruit.

Dans une première partie, nous présenterons les différentes EDP qui peuvent être utilisées en restauration ainsi que la façon de les traiter de manière informatique. Ensuite, nous nous intéresserons aux effets des ces différentes EDP : sur des images normales puis bruitées. Cela nous permettra de comparer ces différentes techniques dans un but de restauration. Enfin nous conclurons cette étude et nous présenterons rapidement d'autres applications des EDP en traitement d'images.

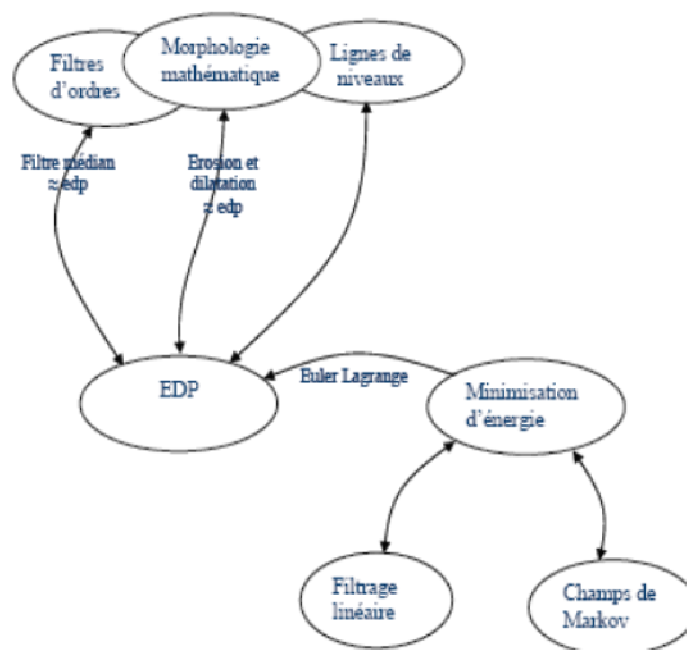


Figure 1 : Les EDP dans le traitement d'images

1. Synthèse des EDP utilisées en traitement d'image

Cette partie a été réalisée à partir des travaux d'**Abdelmounim Belahmidi** [1].

1.1. Diffusion isotrope

Dans la théorie classique de détection des bords, l'information essentielle contenue dans une image est constituée par les frontières des objets. Cette théorie définit ces frontières là où la variation du niveau de gris de l'image est importante. Or dans une image, le bruit peut générer des variations sans signification et des bords qui n'existent pas visuellement. C'est de là que naît la nécessité de filtrer l'image avant d'extraire ses bords.

Le lissage linéaire consiste en une opération de convolution linéaire:

$$u(x, y, t) = f(x, y) * G_t(x, y) = \int G_t(x - \zeta, y - \eta) \times f(\zeta, \eta) d\Omega$$

Avec $f(x, y)$: Signal de départ,

$u(x, y, t)$: Image restaurée

t : paramètre qui contrôle l'importance du lissage

G : opérateur Gaussien : $G_t(x, y) = \frac{1}{\sqrt{4\pi t}} e^{-(x^2+y^2)/4t}$, de variance $\sigma^2=2t$

Koenderink [2] a montré que cette opération de convolution par un opérateur Gaussien est équivalente à la diffusion de f par l'équation de la chaleur :

$$\frac{\partial u}{\partial t} = \text{div}(\nabla(u(x, y, t))) = \begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \\ u(x, y, 0) = f(x, y) \end{cases}$$

Cette EDP parabolique linéaire permet une diffusion isotrope.

Dans cette méthode, seules les basses fréquences sont conservées : cela correspond à une destruction des détails de l'image, le bruit y compris, en faveur des grandes tendances.

Nous avons implémenté ce filtre, ainsi que le filtrage Gaussien correspondant. L'image d'origine utilisée est celle de Léna :

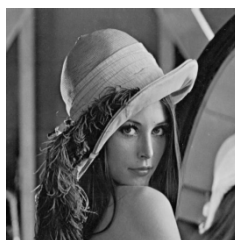


Figure 2 : Léna en niveau de gris



Figure 3 : Comparaison entre EDP linéaire et filtrage gaussien

Nous observons bien que, pour un pas de temps dt , la $k^{\text{ème}}$ itération de l'image filtrée par l'EDP linéaire, et celle filtrée par convolution Gaussienne de variance $\sigma^2 = 2 \cdot k \cdot dt$, sont les mêmes.

Le problème rencontré pour construire le filtrage Gaussien a été le choix du noyau de convolution ; en effet, selon la valeur de la variance, tous les noyaux ne seront pas adaptés. Nous avons ainsi augmenté la taille de ce noyau ([11, 11], puis [19, 19] et [25, 25]).

L'inconvénient du lissage linéaire est ainsi que dans des régions présentant des discontinuités importantes au niveau de l'intensité du niveau de gris, celles-ci seront lissées, et il y aura une diminution du contraste visuel rendant difficile une détection de contours.

1.2. Diffusion anisotrope

Les modèles présentés ci après permettent ainsi de combiner extraction et reconstruction d'image.

Ils ont été développés par **Malik et Perona** [3], et le principe est un filtrage conditionnel :

- Forte diffusion dans les zones à faible gradient
- Faible diffusion dans les zones à fort gradient (typiquement les bords)

Pour obtenir cette diffusion conditionnelle, l'équation de la chaleur est remplacée par l'équation aux dérivées partielles suivante :

$$\frac{\partial u}{\partial t} = \text{div}(c(|\nabla u(x, y, t)|)\nabla u(x, y, t))$$

Avec $c(\cdot)$ une fonction régulière à valeur positive qui vérifie les conditions suivantes :

$$C(0)=1 ; \quad \lim_{s \rightarrow \infty} sc(s) = 0 ; \quad c'(s) \leq 0 \quad \text{pour } s > 0$$

Parmi les fonctions recommandées par Malik et Perona, on trouve :

- $c(s) = \frac{1}{1 + \left(\frac{s}{\alpha}\right)^2}$
- $c(s) = \exp\left(-\left(\frac{s}{\alpha}\right)^2\right)$

Si l'on développe l'équation, on trouve :

$$\frac{\partial u}{\partial t} = \underbrace{c(|\nabla u|)|\nabla u| \text{div}\left(\frac{\nabla u}{|\nabla u|}\right)}_{\text{Partie « posée »}} + \underbrace{[c(|\nabla u|) + c'(|\nabla u|)|\nabla u|]\nabla^2 u\left(\frac{\nabla u}{|\nabla u|}, \frac{\nabla u}{|\nabla u|}\right)}_{\text{Partie mal posée}}$$

Le second terme de l'équation peut devenir instable, c'est-à-dire que pour une même donnée initiale, la solution peut être divergente.

C'est pourquoi **Catté, Lions, Morel et Coll [4]** ont proposé un modèle un peu différent :

$$\frac{\partial u}{\partial t} = \text{div}(c(|\nabla G * u(x, y, t)|)\nabla u(x, y, t))$$

Où $\nabla G * u(x, y, t)$ est le gradient de la solution de l'équation de la chaleur.

Les auteurs ont montré que cette équation admet une solution unique, régulière, et un schéma numérique convergent.

Ces modèles non linéaires permettent d'éviter les flous, d'empêcher la délocalisation des bords, et ont un effet régularisant sur les zones homogènes.

Nous avons testé ces 3 modèles non linéaires :

- celui de Malik et Perona avec la fonction $c(s) = \exp\left(-\left(\frac{s}{\alpha}\right)^2\right)$ (**'pm1'**)
- celui de Malik et Perona avec la fonction $c(s) = \frac{1}{1 + \left(\frac{s}{\alpha}\right)^2}$ (**'pm2'**)
- celui de Catté, Lions, Morel et Coll avec la fonction $c(s) = \frac{1}{1 + \left(\frac{s}{\alpha}\right)^2}$ (**'Catt pm2'**)

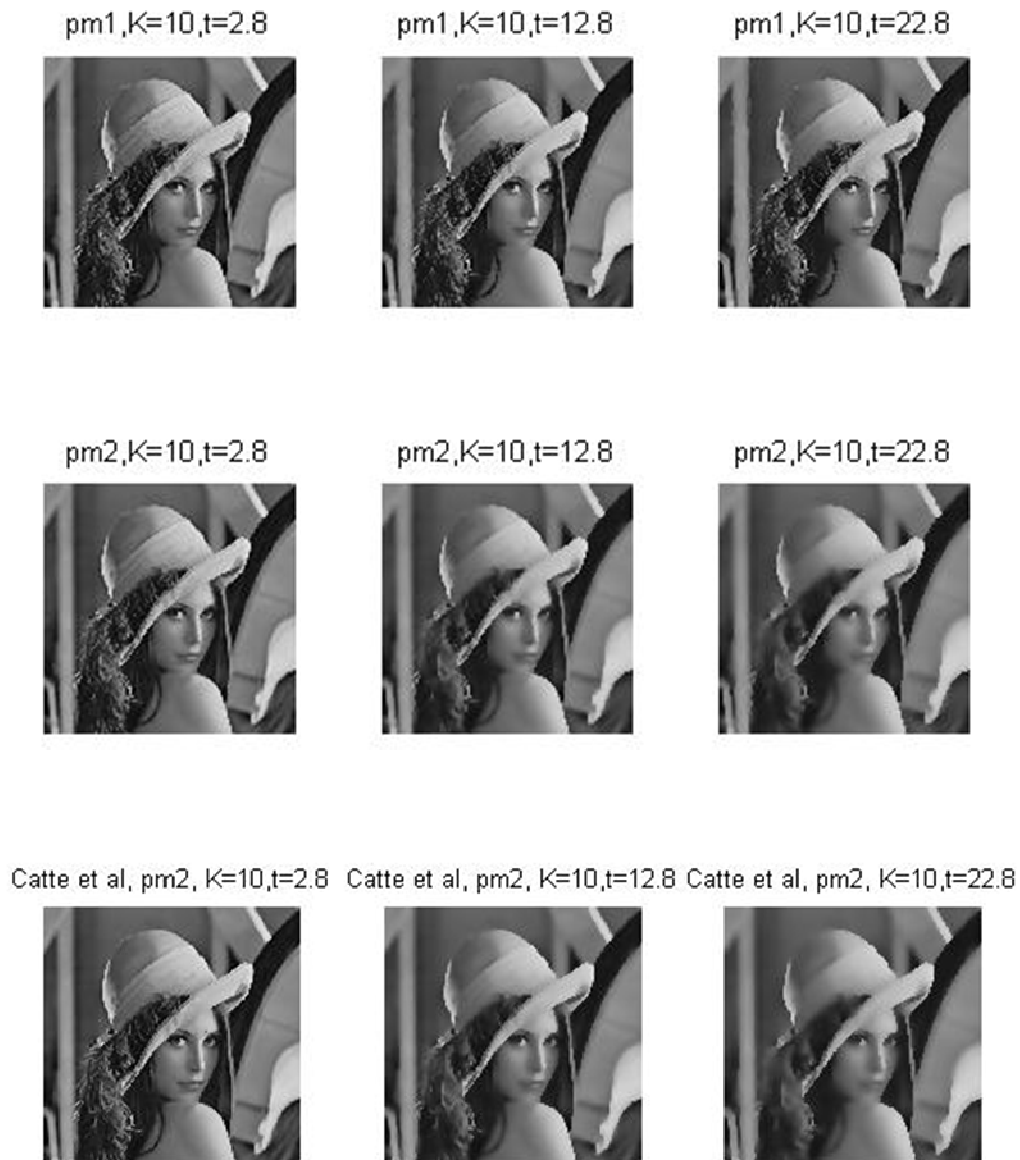


Figure 4 : Effets des EDP non linéaires

Ces réalisations ont été effectuées avec un pas de temps $dt=0.2$.

Nous pouvons constater que dans les 3 cas, l'image se floute, mais les contours sont bien préservés. Pour une même constante K , le modèle 1 floute moins rapidement l'image que le modèle 2. Qualitativement, il est difficile de voir une différence entre le modèle 2 de Malik et Perona simple et celui utilisant le gradient de l'image lissée. En outre, les 3 modèles divergent pour un pas dt trop grand (>0.25).

Les schémas numériques utilisés pour le modèle linéaire et les modèles non linéaires précédents ont été inspirés de Gilboa [9], et sont détaillés en annexe.

1.3. Diffusion dégénérée

Alvarez, Lions et Morel [5] ont proposé le processus de diffusion suivant:

$$\frac{\partial u}{\partial t} = c(|\nabla G * u(x, y, t)|) |\nabla u| \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right)$$

Le terme d'inhibition $\nabla G * u(x, y, t)$ est exactement celui utilisé dans le modèle précédent (Catté, Lions, Morel et Coll), et le terme $|\nabla u| \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right)$ représente l'opérateur de diffusion dégénéré (« opérateur du mouvement par courbure moyenne ») qui diffuse u uniquement dans la direction orthogonale au gradient ∇u . Ce modèle est lié à celui de Malik et Perona puisque le terme $c(|\nabla u|) |\nabla u| \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right)$ correspond à la partie « bien posée » de l'équation de diffusion développée, où l'on remplace ensuite ∇u par $\nabla G * u$.

Ce modèle a pour but de lisser u de chaque côté d'un bord sans altérer les discontinuités de l'image.

Deux cas particuliers de ce modèle avec gradient non lissé (c'est-à-dire avec $c(|\nabla u|)$) sont intéressants:

1.3.1. Dilatation et érosion

Si $c(|\nabla u|) \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) = \pm 1$: on obtient alors les équations de dilatation et d'érosion avec une boule pour élément structurant :

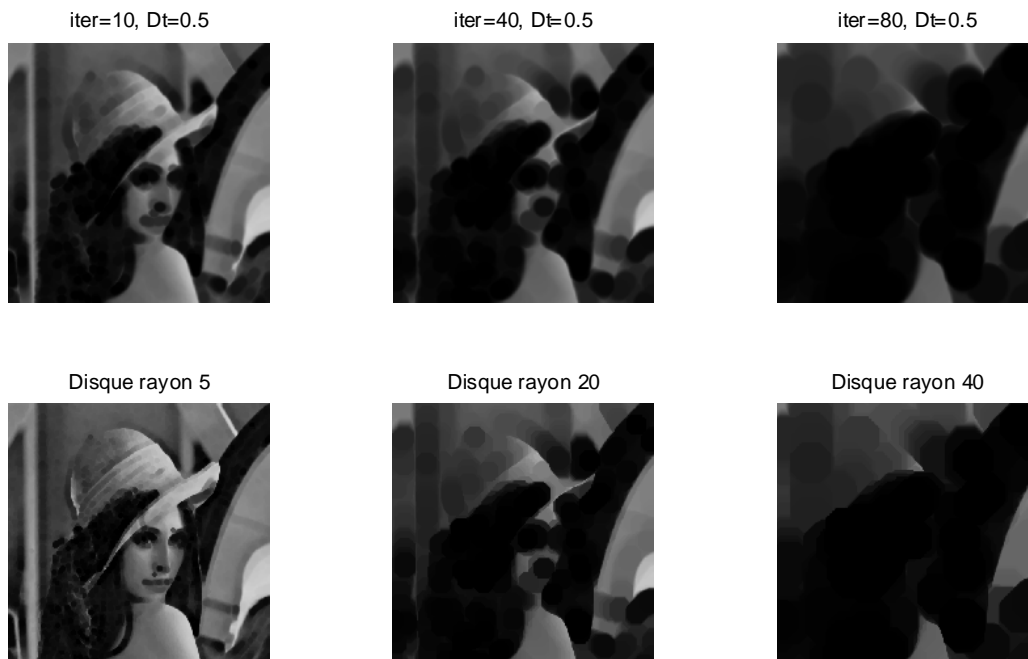
$$\frac{\partial u}{\partial t} = \pm |\nabla u|$$


Figure 5 : Comparaison entre érosion par EDP (haut) et érosion morphologique (bas)

Nous avons comparé les résultats obtenus avec l'EDP d'érosion avec ceux obtenus par érosion morphologique. Appliquer l'EDP d'érosion sur n itérations avec un pas de temps Dt semble correspondre à une érosion morphologique avec pour élément structurant un disque de rayon $R=n.Dt$. La seule différence est que les images obtenues par EDP sont plus lisses que les images érodées morphologiquement.

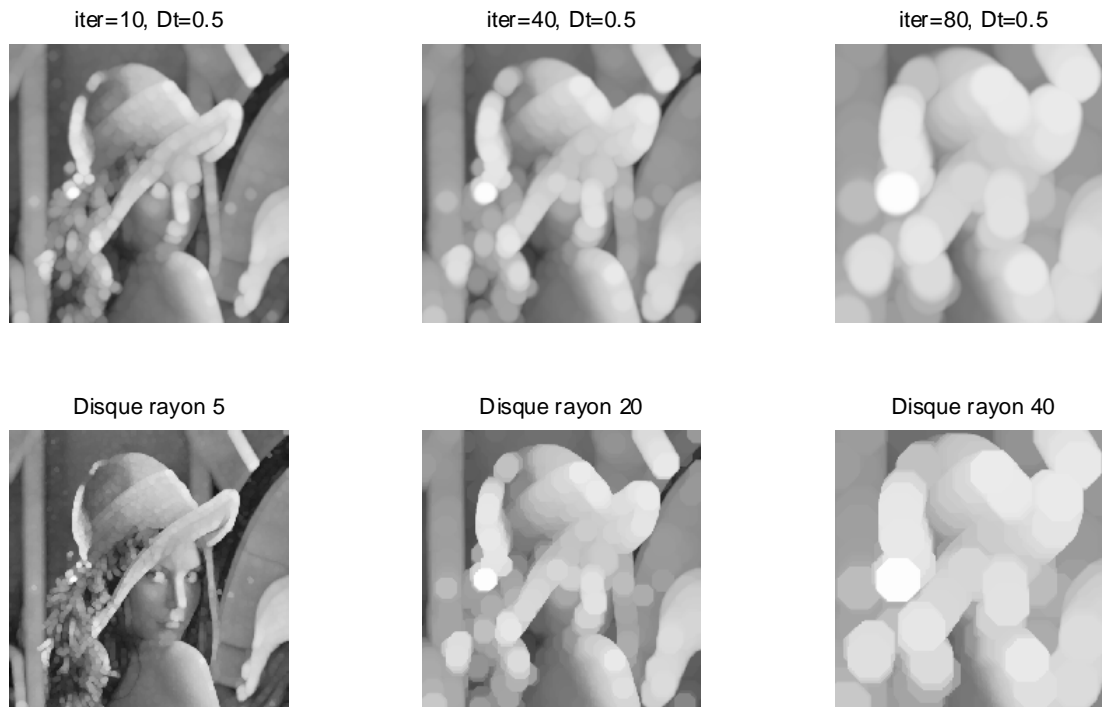


Figure 6 : Comparaison entre dilation par EDP et dilatation morphologique

Les comparaisons entre les images obtenues avec l'EDP de dilatation et celles obtenues par dilatation morphologiques sont les mêmes que dans le cas de l'érosion : appliquer l'EDP de dilatation sur n itérations avec un pas de temps Dt correspond à une dilatation morphologique avec pour élément structurant un disque de rayon $R=n.Dt$. La seule différence est que les images obtenues par EDP sont plus lisses que les images érodées morphologiquement.

L'aspect plus lisse de la morphologie via EDP peut ainsi présenter un avantage par rapport aux opérations de morphologie classique qui font apparaître de manière très nette l'élément structurant.

A partir de ces 2 EDP, il est possible de définir une ouverture et une fermeture à l'aide d'EDPs. Le rendu, plus lisse, des EDP peut être un avantage dans certains cas.

L'implémentation de ces deux EDP sous Matlab a été faite à partir d'un schéma aux différences finies, inspiré de l'article de P. Maragos et M. Akmal Butt [7]. Le détail du code Matlab est donné en annexe **REF.**

1.3.2. Diffusion anisotrope selon les lignes de niveau

Si $c(s) = t$: cas d'une diffusion anisotrope dans la direction des lignes de niveau :

$$\frac{\partial u}{\partial t} = t |\nabla u| \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right)$$



Figure 7 : Diffusion anisotropique selon les lignes de niveau

Le terme $\operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right)$ est la dérivée seconde de u dans la direction orthogonale au gradient

C'est un lissage anisotropique conditionnel, mais seulement le long des courbes de niveau de l'image. La fonction $c(|\nabla u|)$ permet de contrôler la vitesse de diffusion (dans notre cas, $c(|\nabla u|) = t$)

Dans les zones où le gradient est faible, le terme $\operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right)$ est grand et permet une diffusion anisotropique le long de la direction orthogonale au gradient, alors que dans les zones où le gradient est fort, la pondération est faible et annule la diffusion.

1.4. EDP et carte de distances

L'EDP de dilatation peut être résolu en utilisant un algorithme d'évolution des courbes [8]. Plus généralement, si la distance locale entre deux pixels voisins est pondérée par une fonction de l'intensité de l'image, la GWDT (Gray Weighted Distance Transform) est la plus petite valeur parmi tous les chemins reliant le pixel p à une source S . Elle peut être implémentée en utilisant la méthode récursive de « min-sum difference » [8].

On peut voir la GWDT comme une évolution de courbe : au temps $t=0$, la frontière de la source S est modélisée par la courbe $\Gamma(0)$, qui va ensuite se propager à la vitesse $\beta(x,y)=c_0/\eta(X,y)$. La courbe qui se propage est assimilée au niveau zéro de la courbe de la fonction $\Phi(x,y,t)$, laquelle évolue selon l'EDP :

$$\Phi_t = \beta ||\nabla\Phi||.$$

La valeur de GWDT au pixel (x,y) est le temps qu'il faut à la courbe pour atteindre ce pixel : c'est le plus petit t tel que $\Phi(x,y,t) \geq 0$.

Il est alors équivalent de résoudre l'équation de la frontière correspondante :

$$||\nabla T|| = 1/\beta.$$

C'est l'EDP Eikonal. Cette EDP peut être utilisée dans plusieurs applications :

- Les transformations de distance Euclidienne
- Gridless Hoftoning : pour percevoir visuellement les intensités d'une image. Par exemple : $||\nabla T|| = \text{const} - I(x,y)$
- Shape-from-shading : pour trouver la fonction de hauteur z d'une surface d'un objet 3D.

2. Application à la restauration d'image

L'une des applications du filtrage est la restauration d'images détériorées.

Notre objectif était de comparer les différentes EDP, et leur efficacité, dans le cadre de la restauration d'image.

Pour cela, nous avons bruité artificiellement 4 images : une première image binaire très simple, l'image de Lena et une image de coupe de jambe ont été bruitées avec un bruit gaussien. L'image Lena a aussi été bruitée avec un bruit poivre et sel.

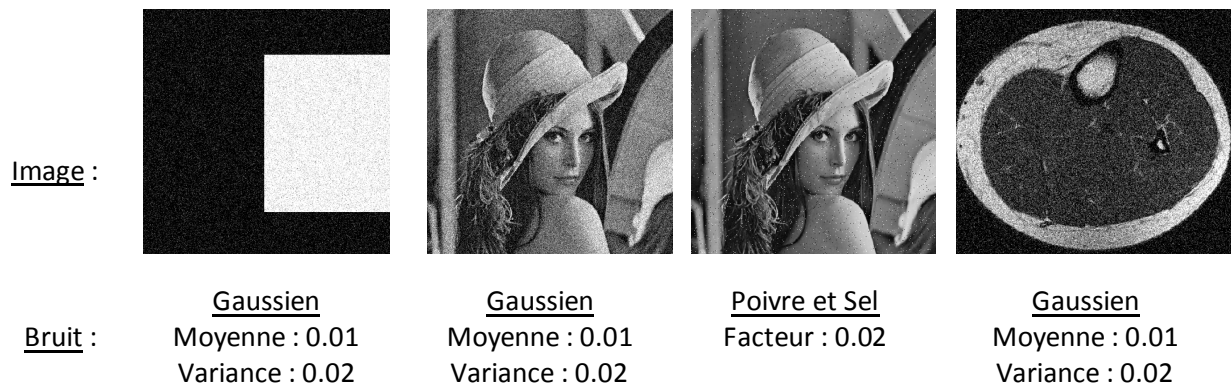


Figure 8 : Images tests bruitées

2.1. Analyse qualitative

Afin de tester l'efficacité des différentes EDP sur des images bruitées, nous avons appliqué 100 pas de diffusion avec un intervalle de temps $dt=0.2$ à l'image binaire contenant un bruit gaussien.

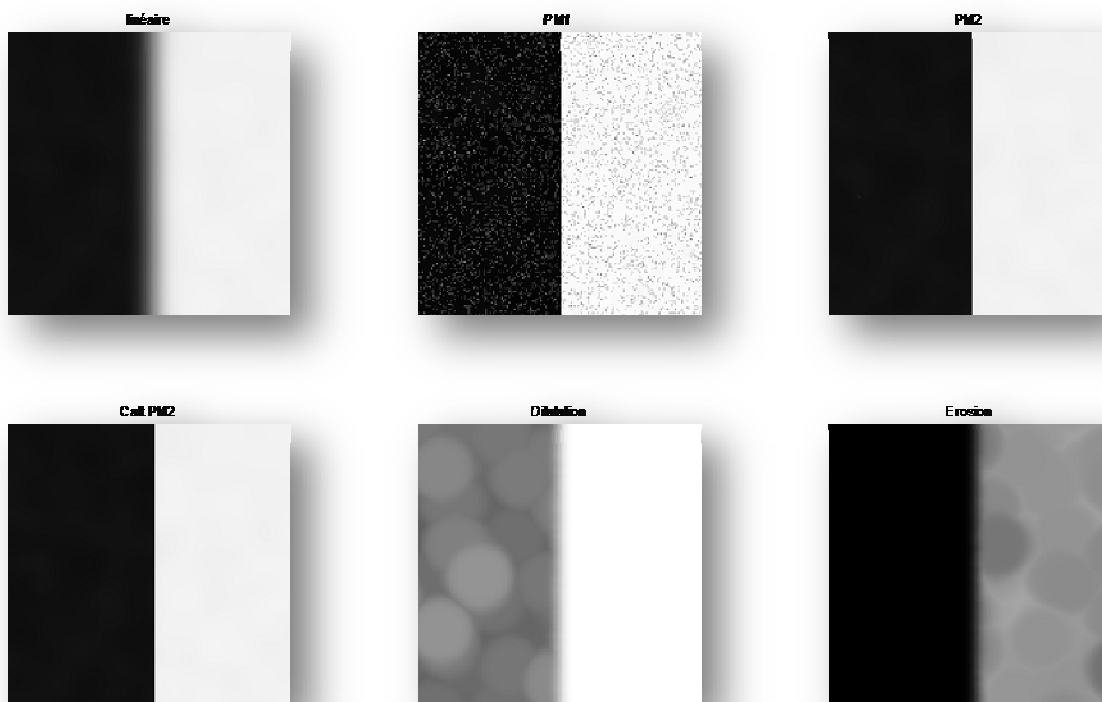


Figure 9 : Image binaire (zoom) bruité après 100 pas de diffusion (

Les résultats présentés ci-dessus mettent en valeur les différences entre les méthodes. La diffusion linéaire supprime bien le bruit mais floute complètement la frontière. La diffusion anisotropique selon le modèle PM1 n'a pas beaucoup amélioré l'image après 100 pas de temps. Il faudrait plus de pas pour que l'on puisse voir une différence. Les méthodes PM2 et

Catt PM2 supprime le bruit tout en conservant la frontière nette. Comme prévu, la dilatation et l'érosion déforment complètement l'image. Cependant, ces observations ne sont que qualitatives, et il paraît intéressant de mettre en place une mesure pour comparer ces différentes EDP

2.2. Protocole de mesure du débruitage

Notre objectif était de pouvoir quantifier le fait que l'image finale (obtenue après filtrage par l'une des EDP) était « proche » ou non de l'image originale non bruitée. La mesure que nous avons choisie de mettre en place est le PSNR ; celui-ci se calcule à partir de la moyenne des erreurs entre chaque pixel des deux images :

$$MSE(I_1, I_2) = \frac{1}{n \cdot m} \sum_i \sum_j (I_1(x_i, y_j) - I_2(x_i, y_j))^2$$

Et s'obtient par la formule suivante :

$$PSNR(I_1, I_2) = 20 \cdot \log_{10}(M / \sqrt{MSE(I_1, I_2)})$$

Avec M la valeur maximale de I_1 et I_2 (dans notre cas, $M=255$).

Avec cette mesure, plus le PSNR sera élevé, plus l'image finale sera « proche » de l'image initiale non bruitée. Par exemple, le PSNR de l'image Léna dégradée par un bruit Gaussien de moyenne 0.01 et de variance 0.02 est de 17,2 dB.

2.3. Résultats

2.3.1. Image Léna bruitée avec un bruit gaussien



Figure 10 : Image Léna bruitée par un bruit gaussien après 30 pas de temps

On retrouve les mêmes propriétés que sur l'image binaire (Figure 9). La méthode la plus adaptée pour ce type de bruit semble être celle de Malik et Perona avec la fonction n^2 ('pm2').

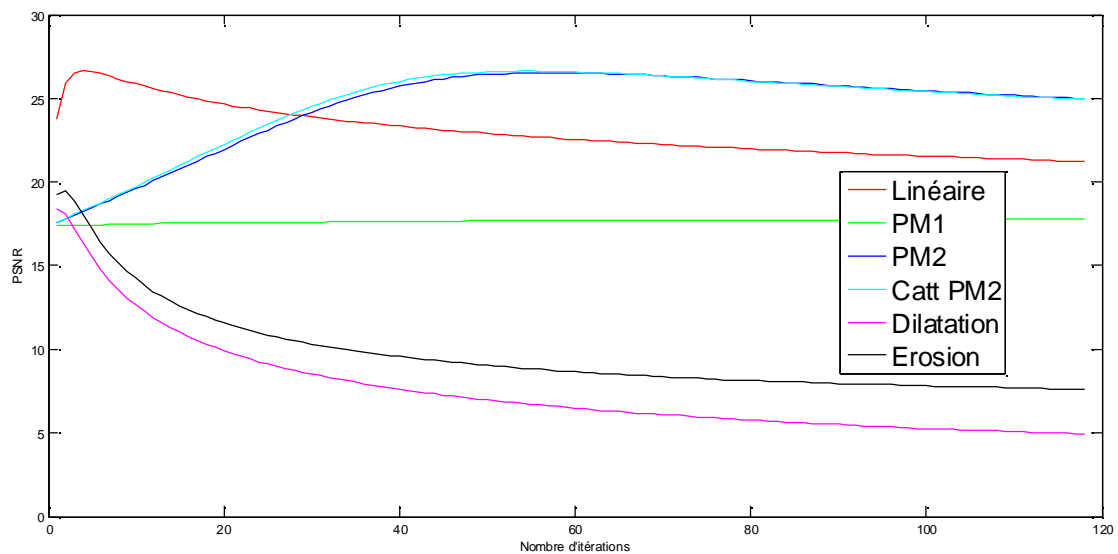


Figure 11: PSNR pour Lena (bruit gaussien moyenne : 0,01 variance : 0,02)

Dans l'ensemble, la mesure du PSNR en fonction du nombre de pas de temps rejoint les observations qualitatives précédentes.

La diffusion linéaire améliore sensiblement le PSNR sur les premiers pas de temps, mais pour un nombre d'itérations plus important, son effet lissant diminue le PSNR. On obtient un maximum de 26 dB après 3 itérations.

Les méthodes de diffusion anisotropiques permettent une nette amélioration du PSNR (à l'exception de 'pm1' dont le processus de diffusion est beaucoup plus long, et qui par conséquent ne modifie que très légèrement l'image). On atteint un maximum de 27 dB au bout de 45 pas de temps.

Voici un zoom sur les images obtenues au meilleur PSNR avec la diffusion Linéaire et la diffusion anisotropique :



Figure 12 : Images au meilleur PSNR pour Diffusion Linéaire et Diffusion anisotropique

Pour un PSNR à peu près identique (26 et 27 dB), on constate que les deux images sont très différentes, ce qui nous fera interrogé plus loin sur la pertinence de ce critère. L'image obtenue par diffusion anisotropique semble être la plus satisfaisante.

La dilatation et l'érosion, comme prévue, n'apporte aucune amélioration sur la qualité de l'image et donc sur le PSNR.

2.3.2. Image Léna bruitée par un bruit Poivre et Sel

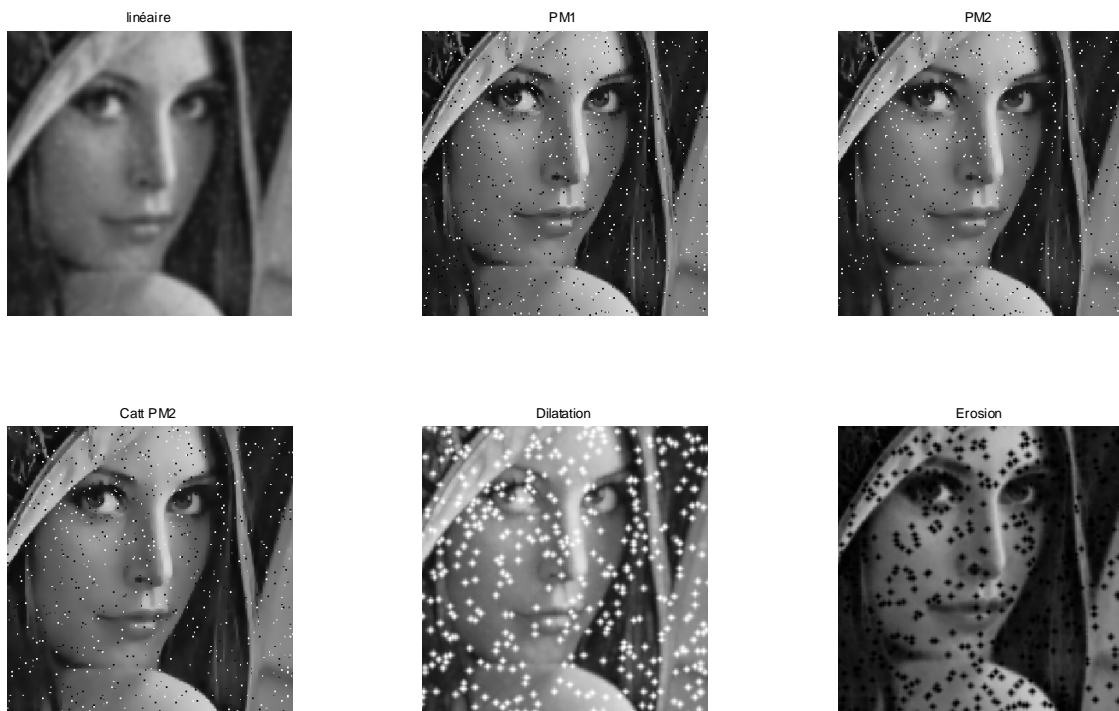


Figure 13 : Image Léna bruitée par un bruit poivre et sel après 10 pas de temps

Visuellement, même si elle floute l'image, seule la diffusion linéaire atténue le bruit. Les EDP basées sur le modèle de Malik et Perona préservant les zones de fort contraste ne modifient pas du tout le bruit qui consiste, dans le cas du bruit poivre et sel, à des zones de très fort contraste, là où la diffusion n'est donc pas importante.

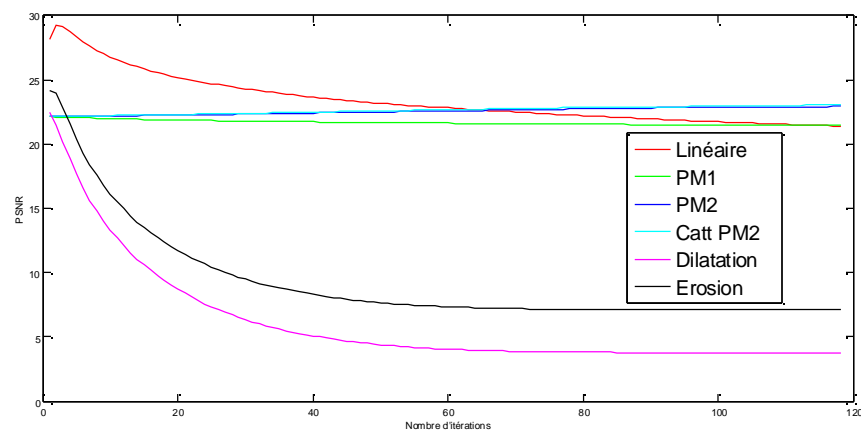


Figure 14 : PSNR Image Léna - bruit poivre et sel

La mesure du PSNR confirme ce que l'on a pu observer visuellement. Seule la diffusion linéaire permet au bout de 3-4 pas de temps d'améliorer le PSNR.

2.3.3. Coupe histologique d'une jambe bruitée par un bruit gaussien

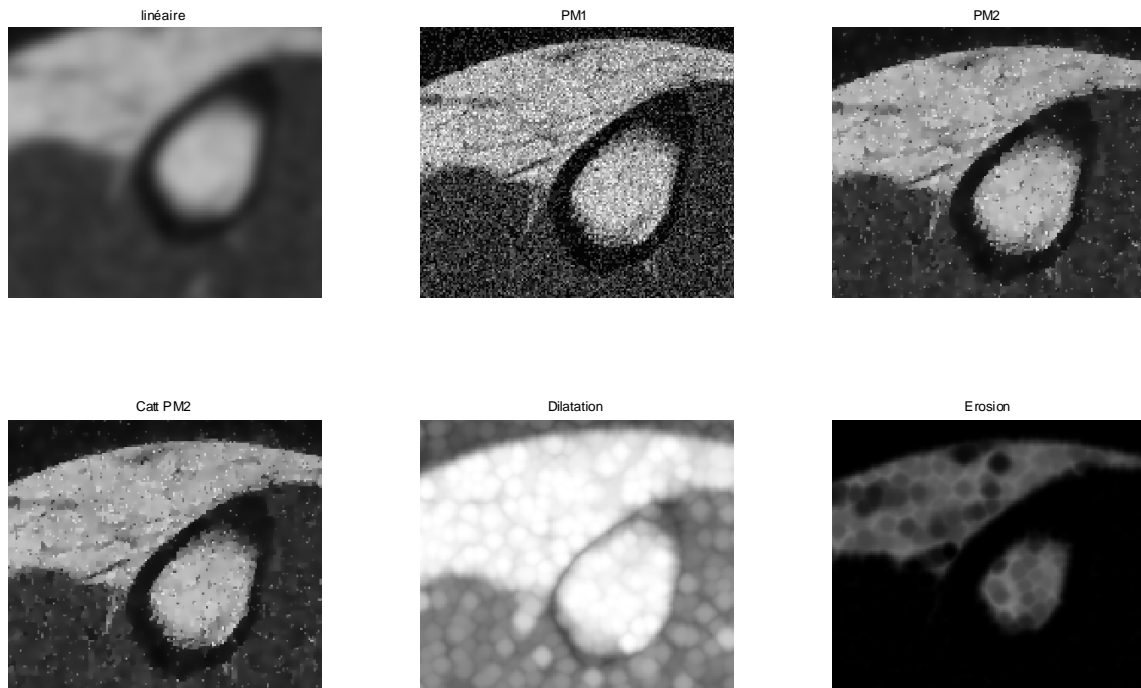


Figure 15 : Jambe (bruitage gaussien) après 30 pas de temps

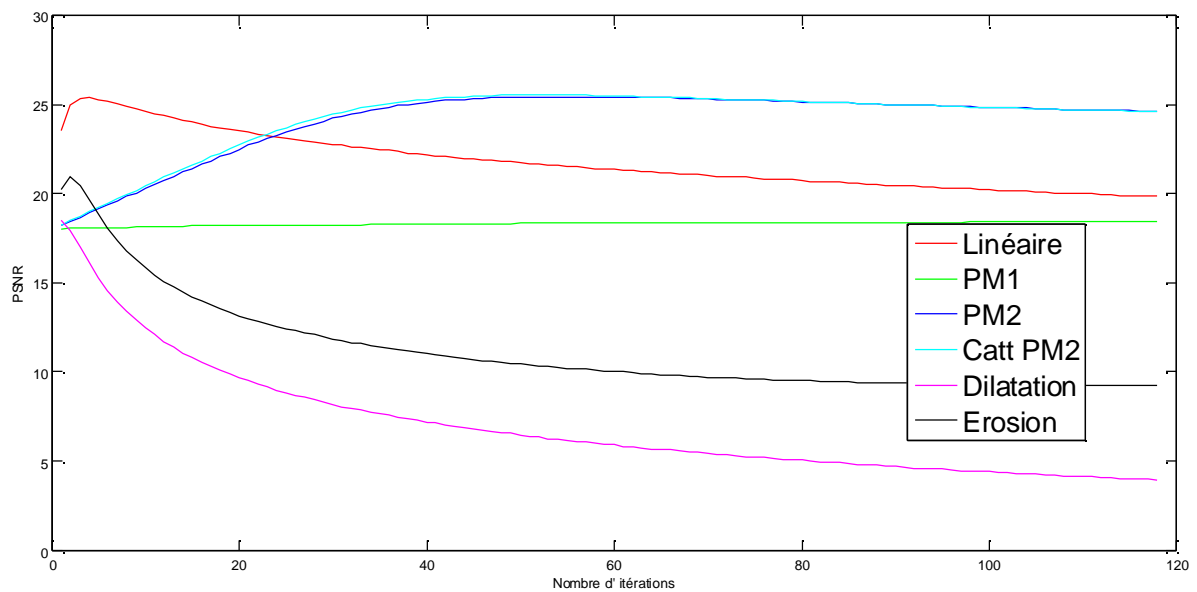


Figure 16 : PSNR Jambe (Bruitage gaussien)

Ce dernier test a été effectué afin de voir si les résultats obtenus sur l'image Léna sont indépendants de l'image utilisée.

Nous constatons que, aussi bien qualitativement que quantitativement (calcul du PSNR), les mêmes observations peuvent être faites sur cette coupe histologique de jambe que sur l'image Léna.

2.3.4. Diffusion linéaire selon les lignes de niveaux

Nous avons effectués chacun des tests décrits ci-dessus avec la méthode de diffusion selon les lignes de niveaux. Cette méthode nécessitant un pas de temps très petit par rapport aux autres méthodes (0.0001 au lieu de 0.2), nous n'avons pas pu la réaliser en même temps que les autres.

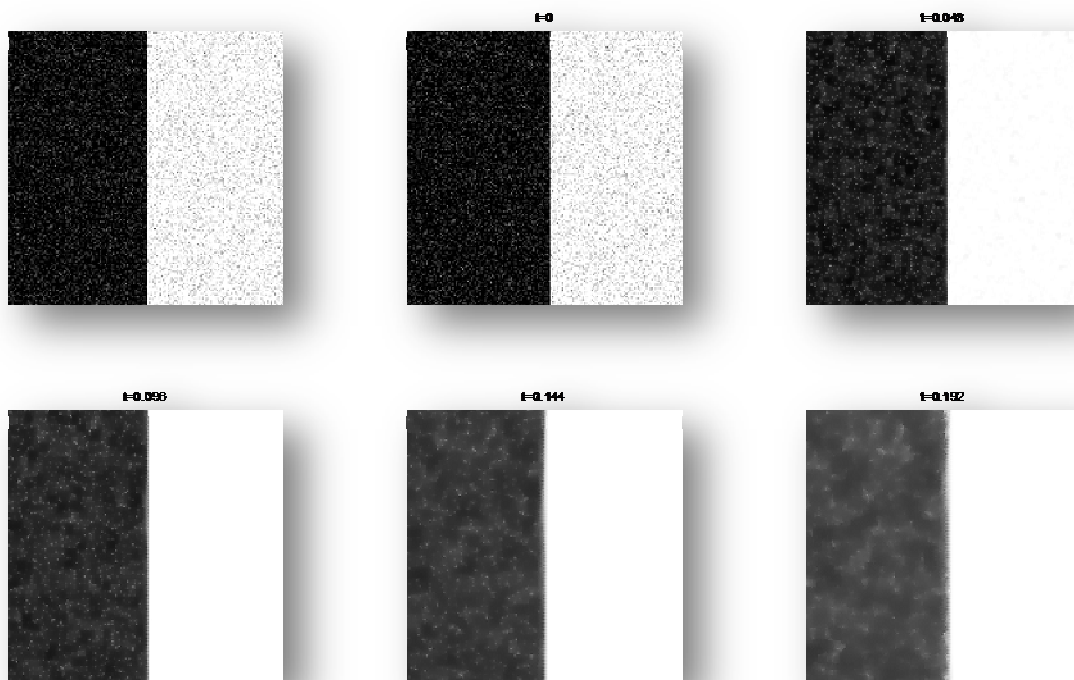


Figure 17 : diffusion anisotrope selon les lignes de niveau

Sur l'image binaire, cette EDP permet de très bien enlever le bruit sur la partie blanche, mais laisse des « traces » blanches sur la partie noire, et le résultat semble ainsi moins satisfaisant que le modèle 2 de Malike et Perona par exemple.



Figure 18 : diffusion anisotrope selon les lignes de niveau

Ces défauts se retrouvent sur des images plus compliquées (comme Léna). Visuellement, il est difficile de dire quel modèle, entre celui-ci et celui de Malik et Perona, est le meilleur ; cela dépendra de l'application et des objectifs voulus.

La valeur maximale du PSNR obtenue avec cette méthode est moins élevée (19,2 dB : cf. Figure 19) que celles obtenues pour les modèles de diffusion linéaire et anisotropique. En effet, l'image correspondant au PSNR maximal est encore très bruitée (cf. Figure 20 : Léna après 20 pas de temps).

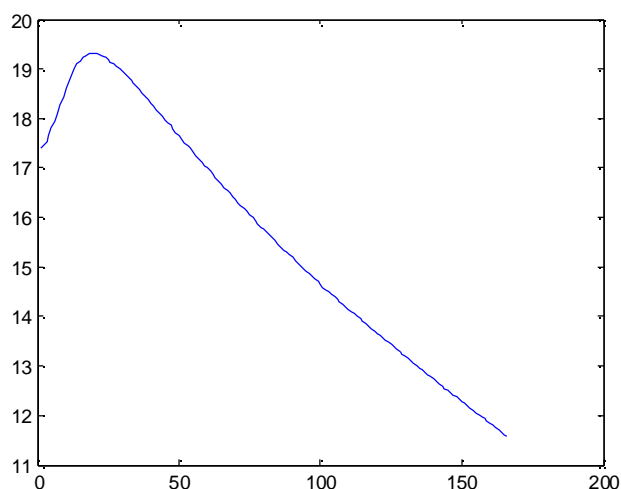


Figure 19: PSNR Diffusion anisotropique selon les lignes de niveau



Figure 20 : Léna après 20 pas de temps

2.4 Critiques de la méthode utilisée

Notre méthode de comparaison des différentes EDP est basée sur le critère PSNR. Celui-ci est globalement cohérent avec les observations qualitatives. Cependant, 2 images ayant un même PSNR peuvent être visuellement très différentes. Il est compliqué de classer l'efficacité des différentes méthodes en se basant sur ce seul critère.

Ce critère permet tout de même de déterminer pour une méthode donnée le pas de temps correspondant à l'image la plus proche de celle d'origine. On peut ensuite comparer visuellement les « meilleures » images de chaque méthodes et obtenir un classement sensoriel des EDP.

Il existe dans la littérature de nombreux autres critères permettant de comparer deux images ou de quantifier le débruitage. Par exemple, certaines méthodes utilisent les propriétés du système visuel humain tel que des comparaisons de contraste. Cependant, ce n'était pas l'objet de cette étude.

Il aurait aussi pu être pertinent de comparer les temps d'exécution de chaque méthode et de l'utiliser comme un critère de performance.

Conclusion et perspectives

Cette étude sur l'utilisation des Equations aux dérivées partielles dans un but de restauration d'images a montré que la méthode proposée par Malik et Perona permet d'obtenir les meilleurs résultats (parmi les EDP étudiées) : le bruit est réduit tout en conservant les zones de fort contraste (les contours). Cependant, pour certains types de bruit (poivre et sel par exemple) cette méthode ne fonctionne pas du tout. L'intérêt de l'utilisation des EDP pour la restauration d'images dépend donc des applications considérées.

Nous avons montré qu'il est possible de redéfinir certains opérateurs de morphologie mathématique à partir d'EDP, ceux-ci présentant l'avantage d'être plus lisse.

L'utilisation des EDP n'est pas restreinte au débruitage d'images, mais peut présenter de nombreux avantages pour d'autres problématiques. La diffusion peut être par exemple utilisée pour des problèmes d'inpainting : (restaurer une partie masquée de l'image) ou alors servir à augmenter artificiellement la résolution d'une image. Les utilisations des EDP en traitement d'image sont donc nombreuses et permettent de résoudre de manière efficace des problèmes dont la résolution pourrait être plus compliquée avec des outils classiques.

Références bibliographiques

- [1] Abdelmounim Belahmidi, **Equations aux dérivées partielles appliquées à la restauration et à l'agrandissement des images**. *Thèse, Université de PARIS IX-Dauphiné*, Juin 2003.
- [2] J.J. Koenderink , **The structure of Image**, *Biol. Cybern*, 50:363 -370, 1984.
- [3] J.Malik et P. Perona, **A scale space and edge detection using anisotropic diffusion**, *IEEE, Comp. Soc. Workshop on Comp. Vision, Miami*, p.16-22, 1987.
- [4] F. Catté, P.L. Lions, J.M. Morel et T. Coll, **Image selective smoothing and edge detection by nonlinear diffusion**. *SIAM J. Numer. Anal.*, 29:182-193, 1992.
- [5] L. Alvarez,P.L. Lions et J.M. Morel, Image selective smoothing and edge detection by nonlinear diffusion ii. *SIAM J. Numer. Anal.*, 29:845-866, 1992. .
- [6] R. Deriche et O. Faugeras, *Les EDP en traitement des images et Vision par ordinateur*, INRIA, Nov 1995
- [7] P. Maragos and M. Akmal Butt, **Partial Differential Equation in Image Analysis: Continuous modeling, discrete processing**. .
- [8] P. Maragos, **Differential Morphology and Image Processing**. *IEEE Transactions on Image Processing*,vol 5, n°6, June 1996.
- [9] <http://www.math.ucla.edu/~gilboa/PDE-filt/diffusions.html>
- [10] Foued DERRAZ, Mohamed BELADGHAM, M'hamed KHELIF, **Mesure Objective de la Qualité d'Image Médicale Dérivée de l'Index de Similarité Structurale**, Laboratoire de Génie Biomédicale –TLEMCEN-ALGERIE

Annexe 1 Schémas numériques utilisés

Notations :

$$\text{Div}(w_1, w_2) = 1/h * (I_{x,-} w_1 + I_{y,-} w_2)$$

$$\nabla w = 1/h(I_{x,+} w, I_{y,+} w)$$

Avec $I_{x,+} w = w_{i+1,j} - w_{i,j} = I_s$;

$$I_{x,-} w = w_{i,j} - w_{i-1,j} = -I_n ;$$

$$I_{y,+} w = w_{i,j+1} - w_{i,j} = I_e ;$$

$$I_{y,-} w = w_{i,j} - w_{i,j-1} = -I_w ;$$

Equation de diffusion linéaire:

$$\frac{\partial u}{\partial t} = \text{div}(\nabla(u(x, y, t))) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

Le schéma numérique correspondant est:

$$\frac{u^{t+1} - u^t}{\Delta t} = \frac{u_{i+1,j}^t - 2u_{i,j}^t + u_{i-1,j}^t}{h^2} + \frac{u_{i,j+1}^t - 2u_{i,j}^t + u_{i,j-1}^t}{h^2} = \frac{I_n + I_s + I_e + I_w}{h^2}$$

Equations de diffusion non linéaire :

$$\frac{\partial u}{\partial t} = \text{div}(c(|\nabla u(x, y, t)|) \nabla u(x, y, t))$$

Dans chaque direction, le gradient est multiplié par une fonction qui dépend de la valeur du gradient dans cette direction.

L'expression du gradient peut ainsi s'écrire :

$$\nabla w = 1/h(c(|I_{x,+} u|) * I_{x,+} u, c(|I_{y,+} u|) * I_{y,+} u)$$

On applique ensuite la divergence à ce gradient :

$$I_{x,-} w_1 = I_{x,-} [c(|I_{x,+} u|) * I_{x,+} u] = c(I_n) * I_n + c(I_s) * I_s$$

D'où l'expression finale du schéma numérique:

$$\frac{u^{t+1} - u^t}{\Delta t} = \frac{c(|I_n|) * I_n + c(|I_s|) * I_s + c(|I_e|) * I_e + c(|I_w|) * I_w}{h^2}$$

Equation de dilatation :

$$\frac{\partial u}{\partial t} = +|\nabla u|$$

Le schéma numérique est celui utilisé dans [7] :

$$\frac{u^{t+1} - u^t}{\Delta t} = + \frac{\sqrt{\min^2(0, -In) + \max^2(0, Is) + \min^2(0, -Iw) + \max^2(0, Ie)}}{h^2}$$

Equation d'érosion

$$\frac{\partial u}{\partial t} = -|\nabla u|$$

$$\frac{u^{t+1} - u^t}{\Delta t} = + \frac{\sqrt{\max^2(0, -In) + \min^2(0, Is) + \max^2(0, -Iw) + \min^2(0, Ie)}}{h^2}$$

Equation de diffusion selon les lignes de niveau :

$$\frac{\partial u}{\partial t} = t|\nabla u| \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right)$$

Pour le calcul de $|\nabla u|$ nous utilisons le schéma utilisé pour l'érosion et la dilatation :

$$G = |\nabla u| = \sqrt{\max^2(0, -In) + \min^2(0, Is) + \max^2(0, -Iw) + \min^2(0, Ie)}$$

Puis :

$$\frac{u^{t+1} - u^t}{\Delta t} = \frac{1}{h^2} tG \left(\frac{In}{\operatorname{abs}(In) + \varepsilon} + \frac{Is}{\operatorname{abs}(Is) + \varepsilon} + \frac{Iw}{\operatorname{abs}(Iw) + \varepsilon} + \frac{Ie}{\operatorname{abs}(Ie) + \varepsilon} \right)$$

Le terme $+\varepsilon$ est ajouté pour éviter une division par zéro dans l'algorithme.

Annexe 2

Code Matlab (diffusion Linéaire, Malik et Perona)

```
function Jd=diffusion(I,method,N,dt,K,sigma2)

% Simulates N iterations of diffusion, parameters:
% J = source image (2D gray-level matrix) for diffusion
% method = 'lin': Linear diffusion (constant c=1).
%           'pml': perona-malik, c=exp{-(|grad(J)|/K)^2} [PM90]
%           'pm2': perona-malik, c=1/{1+(|grad(J)|/K)^2} [PM90]
%
% K      edge threshold parameter
% N      number of iterations
% dt     time increment (0 < dt <= 0.25, default 0.2)
% sigma2 - if present, calculates gradients of diffusion coefficient
%           convolved with gaussian of var sigma2 (Catte et al [CLMC92])

if ~exist('N')
    N=1;
end
if ~exist('K')
    K=1;
end
if ~exist('dt')
    dt=0.2;
end
if ~exist('sigma2')
    sigma2=0;
end

[Ny,Nx]=size(I);
J=cell(N);
J{1}=I;

if (nargin<3)
    error('not enough arguments (at least 3 should be given)');
end

for i=1:N;
    % gaussian filter with kernel 5x5 (Catte et al)
    if (sigma2>0)
        Jo = J{i}; % save J original
        J{i}=gauss(J{i},5,sigma2);
    end

    % calculate gradient in all directions (N,S,E,W)
    In=[J{i}(1,:); J{i}(1:Ny-1,:)]-J{i}; % -I_(x,-) w=w_(i,j)-w_(i-1,j)
    Is=[J{i}(2:Ny,:); J{i}(Ny,:) ]-J{i}; % I_(x,+) w=w_(i+1,j)-w_(i,j)
    Ie=[J{i}(:,2:Nx) J{i}(:,Nx)]-J{i}; % I_(y,+) w=w_(i,j+1)-w_(i,j)
    Iw=[J{i}(:,1) J{i}(:,1:Nx-1)]-J{i}; % -I_(y,-) w=w_(i,j)-w_(i,j-1)

    % calculate diffusion coefficients in all directions according to
    method
    if (method == 'lin')
        Cn=K; Cs=K; Ce=K; Cw=K;
    elseif (method == 'pml')
        Cn=exp(-(abs(In)/K).^2);
        Cs=exp(-(abs(Is)/K).^2);
    end
end
```

```

        Ce=exp(-(abs(Ie)/K).^2);
        Cw=exp(-(abs(Iw)/K).^2);
    elseif (method == 'pm2')
        Cn=1./(1+(abs(In)/K).^2);
        Cs=1./(1+(abs(Is)/K).^2);
        Ce=1./(1+(abs(Ie)/K).^2);
        Cw=1./(1+(abs(Iw)/K).^2);
    else
        error(['Unknown method ' method '']);
    end
    if (sigma2>0) % calculate real gradients (not smoothed)
        In=[Jo(1,:); Jo(1:Ny-1,:)]-Jo;
        Is=[Jo(2:Ny,:); Jo(Ny,:)]-Jo;
        Ie=[Jo(:,2:Nx) Jo(:,Nx)]-Jo;
        Iw=[Jo(:,1) Jo(:,1:Nx-1)]-Jo;
        J{i}=Jo;
    end
    % Next Image J
    J{i+1}=J{i}+dt*(Cn.*In + Cs.*Is + Ce.*Ie + Cw.*Iw);
end;
Jd = J;

```

```

function Ig=gauss(I,ks,sigma2)
% Ig=gauss(I,ks,sigma2)
% ks - kernel size (odd number)
% sigma2 - variance of Gaussian
[Ny,Nx]=size(I);
hks=(ks-1)/2; % half kernel size
if (Ny<ks) % 1d convolution
    x=(-hks:hks);
    flt=exp(-(x.^2)/(2*sigma2)); % 1D gaussian
    flt=flt/sum(sum(flt)); % normalize
    % expand
    x0=mean(I(:,1:hks)); xn=mean(I(:,Nx-hks+1:Nx));
    eI=[x0*ones(Ny,ks) I xn*ones(Ny,ks)];
    Ig=conv(eI,flt);
    Ig=Ig(:,ks+hks+1:Nx+ks+hks); % truncate tails of convolution
else
    %% 2-d convolution
    x=ones(ks,1)*(-hks:hks); y=x';
    flt=exp(-(x.^2+y.^2)/(2*sigma2)); % 2D gaussian
    flt=flt/sum(sum(flt)); % normalize
    % expand
    if (hks>1)
        xL=mean(I(:,1:hks))'; xR=mean(I(:,Nx-hks+1:Nx))';
    else
        xL=I(:,1); xR=I(:,Nx);
    end
    eI=[xL*ones(1,hks) I xR*ones(1,hks)];
    if (hks>1)
        xU=mean(eI(1:hks,:)); xD=mean(eI(Ny-hks+1:Ny,:));
    else
        xU=eI(1,:); xD=eI(Ny,:);
    end
    eI=[ones(hks,1)*xU; eI; ones(hks,1)*xD];
    Ig=conv2(eI,flt,'valid');
end

```


Annexe 3 Code Matlab (Dilatation et Erosion)

```
%Fonction qui calcul le résultat de l'EDP de dilatation
% pour 'iter' itérations avec un pas de temps 'dt'
function J=dilatation2(I,iter,dt)

J=double(I);
[Ny,Nx]=size(J);

for i=1:iter;
    % Calcul du gradient dans toutes les directions (N,S,E,W)
    ln=[J(1,:); J(1:Ny-1,:)]-J;
    ls=[J(2:Ny,:); J(Ny,:)]-J;
    le=[J(:,2:Nx) J(:,Nx)]-J;
    lw=[J(:,1) J(:,1:Nx-1)]-J;

    %Calcul de la norme du gradient
    G=sqrt(power(min(0,-ln),2)+power(max(0,ls),2)+power(min(0,-lw),2)+power(max(0,le),2));

    J=J+dt*G;
    J=(J<256).*J;
end

end
```

```
%Fonction qui calcul le résultat de l'EDP d'érosion
%pour 'iter' itérations avec un pas de temps 'dt'
function J=erosion2(I,iter,dt)

J=double(I);
[Ny,Nx]=size(J);

for i=1:iter;
    % Calcul du gradient dans toutes les directions (N,S,E,W)
    ln=[J(1,:); J(1:Ny-1,:)]-J;    % -l_(x,-) w=w_(i,j)-w_(i-1,j)
    ls=[J(2:Ny,:); J(Ny,:)]-J;    % l_(x,+) w=w_(i+1,j)-w_(i,j)
    le=[J(:,2:Nx) J(:,Nx)]-J;    % l_(y,+) w=w_(i,j+1)-w_(i,j)
    lw=[J(:,1) J(:,1:Nx-1)]-J;    % -l_(y,-) w=w_(i,j)-w_(i,j-1)

    %Calcul de l'opposé de la norme du gradient
    G=sqrt(power(max(0,-ln),2)+power(min(0,ls),2)+power(max(0,-lw),2)+power(min(0,le),2));

    J=J-dt*G;
    J=(J>0).*J;
end

end
```