People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University Djilali Bounaama Khemis Miliana
Faculty of Sciences and of the Technology
Department of Mathematics

Master thesis

Presented for obtaining the Master diploma in Mathematics

Specialty : Mathematical Analysis and Applications

by

ARDJOUNE Abdelkadir

Fractional-Order Anisotropic Diffusion for Image Denoising

Defended on July 20, 2019

Committee members

Pr. M. Hachama	**Supervisor**
Dr. O. Benniche	**Jury President**
Pr. M. Benbachir	**Examiner**
Dr. B. Chaouchi	**Examiner**

2018-2019

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University Djilali Bounaama Khemis Miliana
Faculty of Sciences and of the Technology
Department of Mathematics

Master thesis

Presented for obtaining the Master diploma in Mathematics

Specialty : Mathematical Analysis and Applications

by

ARDJOUNE Abdelkadir

Fractional-Order Anisotropic Diffusion for Image Denoising

Defended on July 20, 2019

Committee members

Pr. M. Hachama	**Supervisor**
Dr. O. Benniche	**Jury President**
Pr. M. Benbachir	**Examiner**
Dr. B. Chaouchi	**Examiner**

2018-2019

Acknowledgements

I would first like to thank my God Allah who gave me the will and the courage to complete this work.

I would also like to express my deepest thanks to my Master thesis director, Professor Mohammed Hachama. For his encouragement, advices and remarks allowed me to progress and to complete my work of Master thesis.

My special thanks also go to Dr. O. Benniche, Dr. B. Chaouchi and Pr. M. Benbachir for accepting to evaluate this work. Big thanks to all the team of teachers from the Department of mathematics who contributed to our supervision during of years.

I would like to express my deep gratitude to all my friends for their support. I will always be grateful to all my family, my parents, sisters and brothers, for their encouragement, prayers and support. I thank them from the bottom of my heart and dedicate this Master thesis to them.

Finally, thank you to all those who participated, near or far, in the realization of this work.

Abstract

This work studies the use of a class of fractional-order anisotropic diffusion equations for noise removal. These equations are Euler–Lagrange equations of a cost functional which is an increasing function of the absolute value of the fractional derivative of the image intensity function. This work has three chapters. Chapter one presents some basic notions as the fractional derivative, some PDEs as Perona-Malik equation, Euler-Lagrange equation. In chapter 2, we concentrate about the numerical part (numerical algorithm) of the proposed equation. In chapter 3, we consider some simulations are conducted to evaluate and compare the results.

Keywords : Anisotropic diffusion. Image smoothing. Fractional-order partial differential equation. Fractional-order difference. Image denoising.

Résumé

Ce travail étudie l'utilisation d'une classe d'équations de diffusion anisotropes d'ordre fractionnaire pour l'élimination du bruit. Ces équations sont des équations d'Euler-Lagrange d'un coût fonctionnel qui est un fonction croissante de la valeur absolue du dérivé fractionnaire de la fonction d'intensité de l'image. Ce travail comporte trois chapitres. Le premier chapitre présente quelques notions de base comme la dérivée fractionnaire, certaines EDP, l'équation de Perona-Malik et l'équation d'Euler-Lagrange. Au chapitre 2, nous nous concentrons sur la partie numérique (algorithme numérique) de l'équation proposée. Au chapitre 3, nous considérons que certaines simulations sont effectuées pour évaluer et comparer les résultats.

Mots-clés : Diffusion anisotrope. Lissage d'image. Équation différentielle partielle d'ordre fractionnaire. Différence d'ordre fractionnaire. Débruitage d'image.

Contents

List of Symbols

D	:= The Riemann-Liouville fractional derivative
${}^{C}D$	:= The Caputo fractional derivative
${}^{GL}D$	:= The Grunwald-Letnikov fractional derivative
$\Gamma(\cdot)$	:= The gamma function
Ω	:= Open set of $\mathbb{R}^n$
$\partial\Omega$	:= The borders of Ω
$L^p(\mathbb{R}^n)$	$:= \{u : \Omega \to \mathbb{R}^n \mid \int_\Omega \lvert u(x)\rvert^p dx < +\infty\}$
$C^p(\mathbb{R}^n)$	:= {$u : \Omega \to \mathbb{R}$, continuous derivatives from order 0 to p}
$H^s(\mathbb{R}^2)$	:= The Sobolev space ($s \in \mathbb{R}$)
$x \in \mathbb{R}^n$	$:= x = (x_1, ..., x_n)$
t	:= The time, $t \in [0, +\infty[$
i, j	:= Location (node numbers)
Δx, Δy	:= The grid size
Δt	:= The step size
$c(\cdot)$	:= The diffusion coefficient
α	:= The scale space ($\in \mathbb{R}^+$)
$D_\alpha u$	$:= (D_{\alpha x}, D_{\alpha y})$ The fractional derivative operator
∇u	$:= \left(\frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, ..., \frac{\partial u}{\partial x_n}\right)$ The gradient operator
$\mathrm{div}(\vec{u})$	$:= \nabla\vec{u} := \sum_{i=1}^{n} \frac{\partial u}{\partial x_i}$
$\langle\cdot,\cdot\rangle$	:= Inner product
$\leftrightarrow$	:= Denotes the Fourier transform pair
$\mathcal{F}$	:= The 2-D continuous Fourier transform operator(2-D CFT)
$\mathcal{F}^{-1}$	:= The inverse 2-D continuous Fourier transform operator(2-D ICFT)
F	:= The 2-D discrete Fourier transform operator(2-D DFT)
F^{-1}	:= The inverse 2-D discrete Fourier transform operator(2-D IDFT)
SNR	:= Signal-to-noise ratio
σ	:= The noise level
k	:= Iteration number

Introduction

As a preliminary introduction. The image is one of the most important ways that a person can communicate with others. It addresses all human beings, educated and ignorant, small and large, breaking the language barrier, so it is the most widespread. Russian novelist Ivan Turgenev says that one picture has been exposed to what a writer could not say in 100 pages.

Image definition

The image is a representation of a scene, representation of a person or object by painting, sculpture, drawing, photography, film, etc. It's also a structured whole of informations that, after being displayed on the screen, has a meaning for the human eye.

Representation of digital images

- **Discrete representation** : A digital image (in gray level) is a matrix of integer or real values. A pixel is the position of indices of a matrix coefficient, moreover. A color image, is given by three grayscale images representing the red, green, and blue channels, i.e. a pixel is coded by 3 components $(\mathtt{r}, \mathtt{g}, \mathtt{b})$.

- **Continuous representation** : Another "continuous" approach is to represent the image by a function defined on a field $\Omega \subset \mathbb{R}^2$, and with values in $\mathbb{R}$. This representation is better in domain of partial differential equations (PDEs).

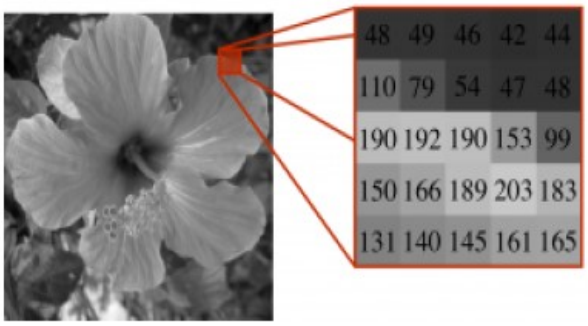


Figure 1: Discrete image.

Image processing

Image processing is the set of methods and techniques operating on them, with the aim of improving the visual appearance of the image and extracting information deemed relevant. In fact, we can distinguish several domains related to images.

- Image processing consists of modifying (filtering) an image. Its purpose is to make visible or hide things in the image.
- Image analysis is about measuring things through an image.
- Computer vision makes it possible to interpret the content of an image.

In the last two types of actions, many processing techniques are used (e.g. denoising, restoration, etc).

Image restoration

Analog or digital images often suffer damage that affects their quality. An observed image u_0 is, in fact, a modified version of an unknown ideal image u that one would like to find or approach by an image $\hat{u}$.
A general model (linear) to represent these degradations is as follows

$$u_0 = Mu + \delta,$$

where δ is a noise (often assumed to mean zero and variance σ^2), and M is a continuous linear operator defined on the space of the images.

Image denoising

The observed image $\mathfrak{u}_0$ is a noisy version of $\mathfrak{u}$. This noise can be additive or multiplicative. Using the previous generic model, we take the identity $\mathsf{M} = \mathrm{Id}$.

A real image always contains a noise. This one can be more or less important. In some cases, you may want to remove it or, at least, reduce it. This is an operation called denoising.

Figure 2: Image denoising problem.

Isotropic diffusion

Is method often used to reduce noise within an image (for image denoising, restoration, and enhancement). But it can damage image features such as edges, lines, and textures. To avoid the damage, it has to be adaptively controlled by the amount of smoothing or the direction of smoothing.

Problem : diffusion occurs in all direction, regardless of edges.

Anisotropic diffusion

Is a technique aiming at reducing image noise without removing significant parts of the image content (Smoothing + Preserve the image features).

Result : diffusion in inhibited when the gradient gets more important (edges).

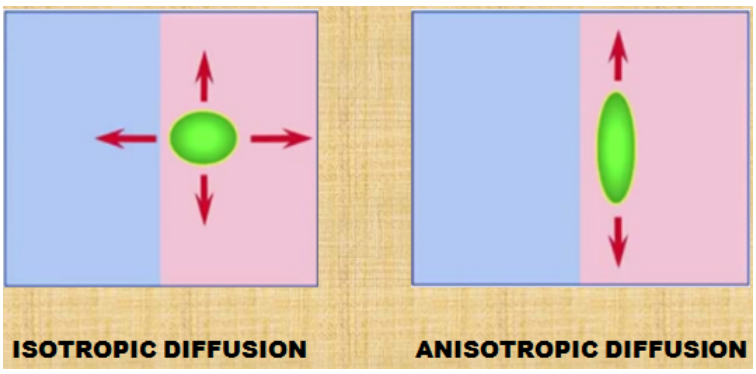


Figure 3: Isotropic vs Anisotropic diffusion.

Chapter 1

Fractional-order anisotropic diffusion

Contents

1.1 Reminders

Differentiability

Definition 1. *[1] Given* $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ *and* $\boldsymbol{v}$ *a unit vector in* $\mathbb{R}^n$*. The directional derivative of* f *in the direction* $\mathbf{v}$ *at* $\mathbf{x}$ *is*

$$\frac{\partial f}{\partial \mathbf{v}}(\mathbf{x}) = D_{\mathbf{v}}(\mathbf{x}) = \lim_{\substack{h \to 0 \\ h > 0}} \frac{f(\mathbf{x} + h\mathbf{v}) - f(\mathbf{x})}{h},$$

when this limit exists.

Definition 2. *[1] Let* $\Omega \subset \mathbb{R}^n$ *be open,* $f : \Omega \to \mathbb{R}^m$*, and* $\mathbf{x} \in \Omega$*.* f *is differentiable at* $\mathbf{x}$ *if there exists* $T_{\mathbf{x}} \in L(\mathbb{R}^n, \mathbb{R}^m)$ *such that*

$$\lim_{\substack{h \to 0 \\ h \in \mathbb{R}^n}} \frac{\|f(\mathbf{x} + h) - (f(\mathbf{x}) + T_{\mathbf{x}}(h))\|}{\|h\|} = 0,$$

f *is differentiable if it is differentiable at all* $\mathbf{x} \in \Omega$*.* $T_{\mathbf{x}}$ *called differential (or derivative) of* f *at* $\mathbf{x}$*, is denoted* $df_{\mathbf{x}}$*.*

Adjoint operators

Let $T : V_1 \to V_2$ be a linear operator on an inner product space V_1 (are Hilbert spaces).

Definition 3. *[2] The adjoint of* T *is a transformation* $T^* : V_2 \to V_1$ *satisfying*

$$\langle T(\vec{x}), \vec{y} \rangle = \langle \vec{x}, T^*(\vec{y}) \rangle,$$

for all $\vec{x} \in V_1, \vec{y} \in V_2$*.*

Remark 1. *The adjoint of* T *may not exist.*

Theorem 1. *[2] Let* V_1, V_2 *be a finite-dimensional inner product spaces, and let* T *be a linear operator on* V_1*. Then there exists a unique function* $T^* : V_2 \to V_1$ *such that*

$$\langle T(\vec{x}), \vec{y} \rangle = \langle \vec{x}, T^*(\vec{y}) \rangle,$$

for all $\vec{x} \in V_1, \vec{y} \in V_2$*. Furthermore,* T^* *is linear.*

Self-adjoint operators

Definition 4. *[2] An operator* $T : V_1 \to V_2$ *is self-adjoint if* $T = T^*$*.*

Remark 2. *If the vector space is Euclidean, we also speak of symmetric operator.*

Theorem 2. *[2] Let V_1, V_2 be a finite dimensional inner product spaces over a field $\mathbb{F}$, and let T and U be linear operators on V_1 having adjoints. Then*

a) $(TU)^* = U^*T^*$

b) $(T^*)^* = T$

c) $I^* = I$

Let $A = (a_{ij})$ be an $m \times n$ matrix with complex entries.

Definition 5. *[2] The adjoint matrix of A is the $n \times m$ matrix $A^* = (b_{ij})$ such that $b_{ij} = \overline{a_{ji}}$.*

1.2 Fractional calculus

Fractional calculus has been used to model physical and engineering processes, which are found to be best described by fractional differential equations. In the recent years, fractional calculus has played a very important role in various fields such as signal and image processing.

1.2.1 Gamma function

Definition 6. *[3] The Gamma function $\Gamma(\alpha)$ is defined by the following integral*

$$\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt \quad (\Re(\alpha) > 0).$$

By integrating in parts, we can see

$$\Gamma(\alpha + 1) = \alpha\Gamma(\alpha) \quad \Re(\alpha) > 0.$$

In particular

$$\Gamma(n + 1) = n! \quad \forall n \in \mathbb{N}.$$

In the follows, we consider $I = [a, b] \quad (-\infty < a < b < +\infty)$ be a finite interval on the real axis $\mathbb{R}$.

1.2.2 Fractional integral

Definition 7. *[3] Let* $f \in L^1([a,b])$. *Riemann-Liouville's fractional integral of the function* f *of order* $\alpha \in \mathbb{R}$ *noted* $I_a^\alpha f$ *is defined by*

$$\left(I_a^\alpha f\right)(x) = \frac{1}{\Gamma(\alpha)} \int_a^x (x-t)^{\alpha-1} f(t)\,dt \quad x > a. \tag{1.1}$$

When $\alpha \in \mathbb{N}$, *the fractional integral generalizes the classical integral, e.g.*

For $\alpha = 1$

$$\left(I_a^1 f\right)(x) = \int_a^x f(t)\,dt.$$

Theorem 3. *[3] If* $f \in L^1([a,b])$, *then* $I_a^\alpha f$ *exists for almost all* $x \in [a,b]$ *and more* $I_a^\alpha f \in L^1([a,b])$.

Definition 8. *[4] Let* $\theta = (0,0)$ *and* $\alpha = (\alpha_1, \alpha_2)$ *where* $0 < \alpha_1, \alpha_2 \leqslant 1$. *Also, put* $J_a \times J_b = [0,a] \times [0,b]$ *where* a *and* b *are positive constants. The Riemann-Liouville fractional partial integral of* $u \in L^1(J_a \times J_b)$ *is defined by*

$$\left(I_\theta^\alpha u\right)(x,y) = \frac{1}{\Gamma(\alpha_1)\Gamma(\alpha_2)} \int_0^x \int_0^y (x-s)^{\alpha_1-1} (y-t)^{\alpha_2-1} u(s,t)\,dt\,ds.$$

1.2.3 Fractional-order derivative

1.2.3.1 Riemann-Liouville fractional derivation

Definition 9. *[3] Let* $f \in \boldsymbol{L}^1([a,b])$, *the fractional derivative in the sense of Riemann-Liouville of* f *of order* $\alpha \in \mathbb{R}$ *noted* $D_a^\alpha f$ *is defined by*

$$\begin{aligned} \left(D_a^\alpha f\right)(x) &= \left(\frac{d}{dx}\right)^n \left(I_a^{n-\alpha} f\right)(x) \\ &= \frac{1}{\Gamma(n-\alpha)} \left(\frac{d}{dx}\right)^n \int_a^x (x-t)^{n-\alpha-1} f(t)\,dt, \end{aligned} \tag{1.2}$$

where $n-1 < \alpha < n$ *and* $x > a$.
In particular, for $\alpha = m \in \mathbb{N}$, *we have*

$$\left(D_a^0 f\right)(x) = \frac{1}{\Gamma(1)} \left(\frac{d}{dx}\right) \int_a^x f(t)\,dt = f(x),$$

$$\left(D_a^m f\right)(x) = \frac{1}{\Gamma(1)} \left(\frac{d^{m+1}}{dx^{m+1}}\right) \int_a^x f(t)\,dt = \frac{d^m}{dx^m} f(x).$$

As a result the fractional derivative in the sense of Riemann-Liouville coincides with the classic derivative for $\alpha \in \mathbb{N}$.

Definition 10. *[4] Whenever the integral* $\left(I_\theta^\alpha u\right)(x,y)$ *exists. The Riemann-Liouville partial derivative of fractional order* α *for a function* $u \in L^1\left(J_a \times J_b\right)$ *is defined by*

$$\left(D_\theta^\alpha u\right)(x,y) = D_{xy}^2\left(I_\theta^{1-\alpha} u\right)(x,y) = \frac{\partial^2}{\partial x \partial y}\int_0^x \int_0^y \frac{(x-s)^{-\alpha_1}(y-t)^{-\alpha_2}}{\Gamma\left(1-\alpha_1\right)\Gamma\left(1-\alpha_2\right)} dt ds.$$

1.2.3.2 Caputo fractional derivation

Definition 11. *[3] Let* $\alpha \in]n-1, n[, n \in \mathbb{N}$ *and* $f \in C^n([a,b])$*, we call derivative of order* $\alpha \in \mathbb{R}$ *in Caputo sense the function defined by*

$$\begin{aligned}\left({}^C D_a^\alpha f\right)(x) &= \left(I_a^{n-\alpha} f^{(n)}\right)(x) \\ &= \frac{1}{\Gamma(n-\alpha)}\int_a^x (x-t)^{n-\alpha-1} f^{(n)}(t) dt.\end{aligned}$$

Proposition 1. *[3] If* $0 \leqslant \alpha, \beta \leqslant 1$ *with* $\alpha + \beta \leqslant 1$ *and* f *of class* C^1 *we have*

$$\left({}^C D_a^\alpha \, {}^C D_a^\beta f\right)(x) = \left({}^C D_a^\beta \, {}^C D_a^\alpha f\right)(x) = \left({}^C D_a^{\alpha+\beta} f\right)(x).$$

1.2.3.3 Relationship with derivatives in the sense of Riemann-Liouville

$$\left({}^C D_a^\alpha f\right)(x) = \left(D_a^\alpha f\right)(x) - \sum_{j=0}^{n-1} \frac{f^{(j)}(a)(x-a)^{j-\alpha}}{\Gamma(j+1-\alpha)}.$$

We can also write

$$\left({}^C D_a^\alpha f\right)(x) = D_a^\alpha\left[f - \sum_{j=0}^{n-1} \frac{f^{(j)}(a)(x-a)^j}{j!}\right].$$

We deduce that if $f^{(j)}(a) = 0$ for $j = 0, 1, 2, \ldots, n-1$ we will have ${}^C D_a^\alpha = D_a^\alpha$.

1.2.3.4 Grünwald-Letnikov fractional derivative

Definition 12. *[3] Let* $\alpha \in]n-1, n[$ *and* $f \in C^0([a,b])$*, we call derivative of order* $\alpha \in \mathbb{C}$ $(\Re(\alpha) \geqslant 0)$ *in Grünwald-Letnikov sense the function defined by*

$$\left({}^{GL} D_a^\alpha f\right)(x) = \lim_{h \to 0} \frac{1}{h^\alpha} \sum_{j=0}^{n} \frac{\Gamma(j-\alpha)}{\Gamma(j+1)\Gamma(-\alpha)} f(x-jh).$$

Proposition 2. *[3] If* $0 \leqslant n-1 < \alpha < n$ *and* $0 \leqslant m-1 < \beta < m$ *we have*

$$\left({}^{GL} D_a^\alpha \, {}^{GL} D_a^\beta f\right)(x) = \left({}^{GL} D_a^\beta \, {}^{GL} D_a^\alpha f\right)(x) = \left({}^{GL} D_a^{\alpha+\beta} f\right)(x),$$

only if $f^{(j)}(a) = 0$ *for all* $j = 0, 1, \ldots, r-2$. *with* $r = \max(m, n)$.

1.2.3.5 Relationship with derivatives in the sense of Riemann-Liouville.

If f is of class C^n, and $0 \leqslant n-1 \leqslant \alpha < n$ by doing part integrations and repeated differentiations we obtain

$$\left(D_a^\alpha f\right)(x) = \sum_{j=0}^{n-1} \frac{f^{(j)}(a)(x-a)^{j-\alpha}}{\Gamma(j+1-\alpha)} + \frac{1}{\Gamma(n-\alpha)} \int_a^x (x-t)^{n-\alpha-1} f^{(n)}(t)dt = \left({}^{GL}D_a^\alpha f\right)(x).$$

In this case the Grünwald-Letnikov approach and the Riemann-Liouville approach are equivalent.

1.3 Fourier transform

To avoid convergence issues, the Fourier integral is first defined over the space $\mathbf{L^1}(\mathbb{R})$ of integrable functions. It is then extended to the space $\mathbf{L^2}(\mathbb{R})$ of finite energy functions.

Definition 13. *[6] Let f be a function satisfying $\int_{-\infty}^{+\infty} |f(t)|dt < +\infty$ ($f \in \mathbf{L^1}(\mathbb{R})$). The Fourier transform is defined by*

$$\mathcal{F}\{f(t)\}(w) = \int_{-\infty}^{+\infty} f(t)e^{-jwt}dt = \hat{f}(w).$$

If $f \in \mathbf{L^1}(\mathbb{R})$ and $\hat{f} \in \mathbf{L^1}(\mathbb{R})$ then, the inverse transform is given by the following formula

$$\mathcal{F}^{-1}\{f(t)\}(w) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} f(t)e^{jwt}dt.$$

Definition 14. *[6] Two-dimensional (2D) Fourier transform of the function $f \in \mathbf{L^1}(\mathbb{R}^2)$ is defined by*

$$\hat{f}(w_1, w_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y)e^{-j(w_1x+w_2y)}dxdy.$$

If $f \in \mathbf{L^1}(\mathbb{R}^2)$ and $\hat{f} \in \mathbf{L^1}(\mathbb{R}^2)$ then, the two-dimensional (2D) inverse Fourier transform is given by the following formula

$$f(x,y) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \hat{f}(w_1, w_2)e^{j(w_1x+w_2y)}dw_1dw_2.$$

Fourier transform properties

Property 1. *[6] Let $f \in \mathbf{L^1}(\mathbb{R})$ and $c \in \mathbb{R}$, then*

$$\mathcal{F}\{f(t-c)\}(w) = e^{-jwc}\mathcal{F}\{f(t)\}(w) = e^{-jwc}\hat{f}(w).$$

Property 2. *[6] Let be two functions* f *and* g *of* $\mathbb{R}$ *in* $\mathbb{R}$*, with* $(f, g \in \mathbf{L^1}(\mathbb{R}))$*, if* $\mathcal{F}\{f\}$ *and* $\mathcal{F}\{g\}$ *are the Fourier transforms of* f *and* g*, respectively. The Fourier transform of the convolution product* $f \star g$[1] *given by*

$$\mathcal{F}\{f \star g(t)\}(w) = \mathcal{F}\{f(t)\}(w) \cdot \mathcal{F}\{g(t)\}(w),$$

and often, a reciprocal formula

$$\mathcal{F}\{(f \cdot g)(t)\}(w) = \mathcal{F}\{f(t)\}(w) \star \mathcal{F}\{g(t)\}(w).$$

Property 3. *[6] Let* $f \in \mathbf{L^1}(\mathbb{R})$*, suppose that* f *is differentiable and that* $f' \in \mathbf{L^1}(\mathbb{R})$*, then the Fourier transform of the first-order derivative is*

$$\mathcal{F}\{f'(t)\}(w) = (jw)\mathcal{F}\{f(t)\}(w) = (jw)\hat{f}(w).$$

Proof. Use integration by parts:

$$\int u dv = [uv] - \int v du,$$

with

$$u = e^{-jwt} \qquad dv = f'(t)dt$$

$$du = -jwe^{-jwt}dt \qquad v = f(t)$$

This yields

$$\begin{aligned}
\mathcal{F}\{f'(t)\}(w) &= \int_{-\infty}^{\infty} f'(t)e^{-jwt}dt \\
&= f\,(t)e^{-jwt}\Big|_{t=-\infty}^{\infty} - \int_{-\infty}^{\infty} -jwf(t)e^{-jwt}dt \\
&= jw\int_{-\infty}^{\infty} f(t)e^{-jwt}dt \\
&= jw \cdot \mathcal{F}\{f(t)\}(w).
\end{aligned}$$

The first term consists of an oscillating function times $f(t)$. But if the function is bounded so that

$$\lim_{x \to \pm\infty} f(t) = 0,$$

then the term vanishes. ■

Remark 3. *[6] More generally, if* $f, ..., f^{(n)} \in \mathbf{L^1}(\mathbb{R})$*, we get*

$$\mathcal{F}\{f^{(n)}(t)\}(w) = (jw)^n\mathcal{F}\{f(t)\}(w) = (jw)^n\hat{f}(w) \quad (n \in \mathbb{N}).$$

[1] We call convolution product of f and g, denoted $f \star g$, the function, if it exists, defined by

$$(f \star g)(t) = \int_{\mathbb{R}} f(t-\tau)g(\tau)d\tau.$$

Property 4. *[3] Let* $f \in \mathbf{L}^1([a,b])$ *be a function such that* $\mathcal{F}(f)$ *and* $\mathcal{F}(I_a^\alpha f)$ *exist and* $\alpha \in \mathbb{R}$. *The Fourier transform of the Riemann-Liouville integral satisfies the relation*

$$\mathcal{F}\{I_a^\alpha f(t)\}(w) = (jw)^{-\alpha}\mathcal{F}\{f\}(w) = (jw)^{-\alpha}\hat{f}(w).$$

$$(n-1 < \alpha < n), n \in \mathbb{N}.$$

Proof. Let's calculate the Fourier transform of the integral fractional Riemann-Liouville.

$$\begin{aligned} I_a^\alpha f(t) &= \frac{1}{\Gamma(\alpha)}\int_a^t (t-\tau)^{\alpha-1} f(\tau)d\tau \\ &= \frac{t^{\alpha-1}}{\Gamma(\alpha)} \star f(t) \end{aligned} \tag{1.3}$$

Let's start with the Fourier transform of the function

$$h(t) = \begin{cases} \frac{t^{\alpha-1}}{\Gamma(\alpha)}, & (t > 0) \\ 0, & (t \leqslant 0) \end{cases}$$

Then

$$\begin{aligned} \mathcal{F}\{h(t)\}(w) &= \int_{-\infty}^{+\infty} h(t)e^{-jwt}dt \\ &= \int_{-\infty}^{0} h(t)e^{-jwt}dt + \int_0^{+\infty} h(t)e^{-jwt}dt \\ &= 0 + \int_0^{+\infty} h(t)e^{-jwt}dt = \int_0^{+\infty} \frac{t^{\alpha-1}}{\Gamma(\alpha)}e^{-jwt}dt \\ &= \frac{1}{\Gamma(\alpha)}\int_0^{+\infty} \left(\frac{y}{jw}\right)^{\alpha-1} e^{-y}\frac{dy}{jw} \quad (y = jwt, dy = jwdt) \\ &= \frac{1}{\Gamma(\alpha)}\int_0^{+\infty} \frac{y^{\alpha-1}}{(jw)^\alpha}e^{-y}dy = \frac{(jw)^{-\alpha}}{\Gamma(\alpha)}\int_0^{+\infty} y^{\alpha-1}e^{-y}dy \\ &= \frac{(jw)^{-\alpha}}{\Gamma(\alpha)}\Gamma(\alpha) = (jw)^{-\alpha}. \end{aligned}$$

Now we can find the Fourier transform of the fractional integral of Riemann-Liouville 1.3, which can be written as the convolution of the functions $h(t)$ and $f(t)$, i.e.

$$I_a^\alpha g(t) = h(t) \star f(t).$$

Using the second property 2 we obtain

$$\mathcal{F}\{I_a^\alpha f(t)\} = (jw)^{-\alpha}\hat{f}(w).$$

■

Property 5. *[3] Similarly, let* $f \in \mathbf{L^1}([a,b])$ *be a function for which* $\mathcal{F}(f)$ *and* $\mathcal{F}\left(I_a^{\alpha} f\right)$ *are defined and* $\alpha \in \mathbb{R}$ *. Then the Fourier transform of the derivative in the Riemann-Liouville sense is defined by*

$$\mathcal{F}\{D_a^{\alpha} f(t)\}(w) = (jw)^{\alpha}\mathcal{F}\{f(t)\}(w) = (jw)^{\alpha}\hat{f}(w).$$

$$(n-1 < \alpha < n), n \in \mathbb{N}.$$

If $f \in \mathbf{L^1}([a,b])$ *and* $\hat{f} \in \mathbf{L^1}([a,b])$ *then, the inverse transform is given by the following formula*

$$D^{\alpha} f(t) = \mathcal{F}^{-1}\left\{(jw)^{\alpha}\mathcal{F}\{f(t)\}\right\}.$$

Proof. Let's calculate the Fourier transform of the fractional derivative in Caputo sense.

$$\begin{aligned}\left({}^{C}D_a^{\alpha} f\right)(t) &= \frac{1}{\Gamma(n-\alpha)}\int_a^t (t-x)^{n-\alpha-1} f^{(n)}(x)\,dt \\ &= \frac{t^{n-\alpha-1}}{\Gamma(n-\alpha)} \star f^{(n)}(t)\end{aligned}$$

Taking the Fourier transform of this Caputo's derivative results in the following expression

$$\mathcal{F}\{{}^{C}D_a^{\alpha} f(t)\}(w) = \mathcal{F}\left\{\frac{t^{n-\alpha-1}}{\Gamma(n-\alpha)} \star f^{(n)}(t)\right\}(w).$$

Now, from the convolution property of the Fourier transform 2, we get

$$\begin{aligned}\mathcal{F}\{{}^{C}D_a^{\alpha} f(t)\}(w) &= \mathcal{F}\left\{\frac{t^{n-\alpha-1}}{\Gamma(n-\alpha)}\right\}(w) \cdot \mathcal{F}\{\, f^{(n)}(t)\}(w) \\ &= \frac{\mathcal{F}\{t^{n-\alpha-1}\}(w)}{\Gamma(n-\alpha)} \cdot \mathcal{F}\{f^{(n)}(t)\}(w) \\ &= \frac{(jw)^{\alpha-n+1-1}}{\Gamma(n-\alpha)}\Gamma(n-\alpha) \cdot (jw)^{n}\mathcal{F}\{f(t)\}(w) \\ &= (jw)^{\alpha-n} \cdot (jw)^{n}\mathcal{F}\{f(t)\}(w) \\ &= (jw)^{\alpha}\mathcal{F}\{f(t)\}(w),\end{aligned}$$

where $\mathcal{F}\{t^{-\mu}\} = (j\omega)^{\mu-1}\Gamma(1-\mu)$.

The same result for $D_a^{\alpha} f(t)$ and ${}^{GL}D_a^{\alpha} f(t)$.

■

Definition 15. *[6] For a two-dimensional, periodic function (e.g., an intensity image)* $u(x,y)$ *of size* $m \times n$*, the discrete Fourier transform (2-D DFT) is defined as*

$$\hat{u}(w_1, w_2) = \frac{1}{\sqrt{mn}}\sum_{x=0}^{m-1}\sum_{y=0}^{n-1} u(x,y) \cdot e^{-j2\pi\left(\frac{w_1 x}{m} + \frac{w_2 y}{n}\right)},$$

for the spectral coordinates $w_1 = 0 \dots m-1$ *and* $w_2 = 0 \dots n-1$. *Similarly, the inverse 2-D DFT is defined as*

$$u(x,y) = \frac{1}{\sqrt{mn}} \sum_{w_1=0}^{m-1} \sum_{w_2=0}^{n-1} \hat{u}(w_1,w_2) \cdot e^{j2\pi\left(\frac{w_1 x}{m}+\frac{w_2 y}{n}\right)},$$

for the image coordinates $x = 0 \dots m-1$ *and* $y = 0 \dots n-1$.

Property 6. *[10] The translation property of the 2-D DFT is*

$$u\left(x-x_0, y-y_0\right) \leftrightarrow e^{-j2\pi\left(\frac{w_1 x_0}{m}+\frac{w_2 y_0}{n}\right)} \hat{u}\left(w_1,w_2\right).$$

1.4 Anisotropic diffusion

1.4.1 Introducing the Perona-Malik equation

The Perona-Malik model, first proposed in 1987, is a nonlinear partial diffusion equation that uses an inhomogeneous diffusivity coefficient. It is widely used in image processing for purposes like smoothing, restoration, segmentation, filtering or detecting edges. It is usually presented in a general form

$$\partial_t u(x,t) = \nabla(c(u(x,t))\nabla u(x,t)). \tag{1.4}$$

Where $c : \mathbb{R} \to [0,1]$ is a decreasing and continuous function (Edge stopping function or diffusion coefficient, is called also diffusivity) vanishing on the edges (high gradients), and close to 1 on regular regions (low gradients), such that

$$c(0) = 1, \text{ and } \lim_{x \to +\infty} c(s) = 0.$$

The typical choices of the function $c(\cdot)$ are

$$c\left(s^2\right) = \frac{1}{1+\frac{s^2}{\lambda^2}},$$

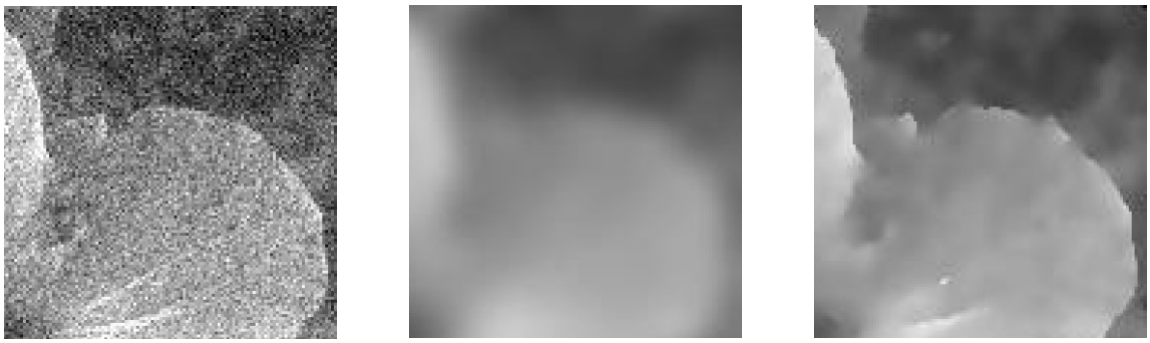
and

$$c\left(s^2\right) = \exp\left(-\frac{s^2}{2\lambda^2}\right),$$

with the parameter $\lambda > 0$.

The Perona-Malik equation is associated with the following energy functional

$$E(u) = \int_\Omega f(|\nabla u|) d\Omega,$$



(a) Image noisy. (b) Heat equation. (c) Perona-Malik equation.

Figure 1.1: Edges preserving by Perona-Malik model.

where $f(\cdot) \geqslant 0$ is an increasing function associated with the diffusion coefficient as

$$c(s) = \frac{f'(\sqrt{s})}{\sqrt{s}}.$$

Anisotropic diffusion equations is then shown to be an energy-dissipating process that seeks the minimum of the energy functional. We consider the following functional defined in the space of continuous images over a support of Ω.

$$E(u) = \int_\Omega f\left(|D_\alpha u|\right) d\Omega. \tag{1.5}$$

$$D_\alpha u = (D_{\alpha x} u, D_{\alpha y} u) \text{ and } |D_\alpha u| = \sqrt{D_{\alpha x}^2 + D_{\alpha y}^2}.$$

We can formally compute the Euler–Lagrange equation for this minimization problem as follows.

Take any test function $\eta \in C^\infty(\Omega)$. Define

$$\Phi(a) = E(u + a\eta) = \int_\Omega f\left(|D_\alpha(u + a\eta)|\right) dxdy = \int_\Omega f\left(|D_\alpha u + aD_\alpha \eta|\right) dxdy.$$

We obtain

$$\begin{aligned}
\Phi'(0) &= \lim_{a\to 0}\frac{\Phi(a)-\Phi(0)}{a} = \lim_{a\to 0}\frac{E(u+a\eta)-E(u)}{a} = \left.\frac{d}{da}\right|_{a=0} E(u+a\eta)\\
&= \left.\frac{d}{da}\right|_{a=0}\int_\Omega f\left(|D_\alpha u + aD_\alpha\eta|\right)dxdy = \int_\Omega \left.\frac{d}{da}\right|_{a=0} f\left(|D_\alpha u + aD_\alpha\eta|\right)dxdy\\
&= \int_\Omega\left(\left.\frac{d}{da}\right|_{a=0}|D_\alpha u + aD_\alpha\eta|\, f'\left(|D_\alpha u|\right)\right)dxdy = \int_\Omega\left(D_\alpha\eta\frac{(D_\alpha u)^T}{|D_\alpha u|}f'\left(|D_\alpha u|\right)\right)dxdy\\
&= \int_\Omega\left(f'\left(|D_\alpha u|\right)\frac{D_{\alpha x}u}{|D_\alpha u|}D_{\alpha x}\eta + f'\left(|D_\alpha u|\right)\frac{D_{\alpha y}u}{|D_\alpha u|}D_{\alpha y}\eta\right)dxdy\\
&= \int_\Omega\left(c\left(|D_\alpha u|^2\right)D_{\alpha x}uD_{\alpha x}\eta + c\left(|D_\alpha u|^2\right)D_{\alpha y}uD_{\alpha y}\eta\right)dxdy\\
&= \int_\Omega\left(\left\langle c\left(|D_\alpha u|^2\right)D_{\alpha x}u, D_{\alpha x}\eta\right\rangle + \left\langle c\left(|D_\alpha u|^2\right)D_{\alpha y}u, D_{\alpha y}\eta\right\rangle\right)dxdy\\
&= \int_\Omega\left(\left\langle D^*_{\alpha x}\left(c\left(|D_\alpha u|^2\right)D_{\alpha x}u\right),\eta\right\rangle + \left\langle D^*_{\alpha y}\left(c\left(|D_\alpha u|^2\right)D_{\alpha y}u\right),\eta\right\rangle\right)dxdy\\
&= \int_\Omega\left(D^*_{\alpha x}\left(c\left(|D_\alpha u|^2\right)D_{\alpha x}u\right) + D^*_{\alpha y}\left(c\left(|D_\alpha u|^2\right)D_{\alpha y}u\right)\right)\eta dxdy,
\end{aligned}$$

for all $\eta\in C^\infty(\Omega)$, where $D^*_{\alpha x}$ is the adjoint of $D_{\alpha x}$ and $D^*_{\alpha y}$ is the adjoint of $D_{\alpha y}$. Thus, the Euler–Lagrange equation is

$$D^*_{\alpha x}\left(c\left(|D_\alpha u|^2\right)D_{\alpha x}u\right) + D^*_{\alpha y}\left(c\left(|D_\alpha u|^2\right)D_{\alpha y}u\right) = 0. \tag{1.6}$$

The Euler–Lagrange equation may be solved through the following gradient descent procedure

$$\frac{\partial u}{\partial t} = -D^*_{\alpha x}\left(c\left(|D_\alpha u|^2\right)D_{\alpha x}u\right) - D^*_{\alpha y}\left(c\left(|D_\alpha u|^2\right)D_{\alpha y}u\right). \tag{1.7}$$

- When $\alpha = 1$, 1.7 is precisely the Perona–Malik equation[7], namely

$$\frac{\partial u}{\partial t} = \operatorname{div}\left(c\left(|\nabla u|^2\right)\nabla u\right). \tag{1.8}$$

- When $\alpha = 2$, 1.7 is precisely the fourth-order anisotropic diffusion equation[8], namely

$$\frac{\partial u}{\partial t} = -\nabla^2\left[c\left(|\nabla^2 u|^2\right)\nabla^2 u\right]. \tag{1.9}$$

- When $1\leqslant\alpha\leqslant 2$, 1.7 leads to a "natural interpolation" between 1.8 and 1.9, e.g.

The third-order pseudo-PDE $\alpha = 3/2 = 1.5$.

$$\begin{aligned}
\frac{\partial u}{\partial t} &= -D^{*}_{3/2x}\left(c\left(\left|D_{3/2}u\right|^{2}\right)D_{3/2x}u\right) - D^{*}_{3/2y}\left(c\left(\left|D_{3/2}u\right|^{2}\right)D_{3/2y}u\right) \\
&= -(-1)^{2}\,{}^{C}D_{3/2x}\left(c\left(\left|D_{3/2}u\right|^{2}\right)D_{3/2x}u\right) - (-1)^{2}\,{}^{C}D_{3/2y}\left(c\left(\left|D_{3/2}u\right|^{2}\right)D_{3/2y}u\right) \\
&= -c\left(\left|D_{3/2}u\right|^{2}\right){}^{C}D_{3/2x}\left(D_{3/2x}u\right) - c\left(\left|D_{3/2}u\right|^{2}\right){}^{C}D_{3/2y}\left(D_{3/2y}u\right) \\
&= -c\left(\left|D_{3/2}u\right|^{2}\right)\left[{}^{C}D_{3/2x}\left(D_{3/2x}u\right) + {}^{C}D_{3/2y}\left(D_{3/2y}u\right)\right].
\end{aligned}$$

Note that in this Euler–Lagrange equation, for the unknown $u \in H^{2\alpha}(\Omega)$, we will first prolongate u to $Eu \in H^{2\alpha}(R^2)$[2] by the prolongation operator E when we work with the Fourier transform $\hat{u}$ of u.

[2]For any $s \in R$, the Sobolev space $H^{s}(R^2)$ can be defined thanks to the Fourier transform

$$H^{s}(\mathbb{R}^{n}) = \left\{u \in L^{2}(\mathbb{R}^{n}) \,|\, \int_{\mathbb{R}^{n}} \left(1+|w|^{2}\right)^{s} |\hat{u}(w)|^{2}\, dw < +\infty\right\}.$$

The space $H^{s}(R^2)$ is a Hilbert space equipped with the norm

$$\|u\|^{2}_{H^{s}} = \int_{\mathbb{R}^{n}} \left(1+|w|^{2}\right)^{s} |\hat{u}(w)|^{2} dw, \quad w = (w_1, w_2).$$

Chapter 2

Approximations and the numerical algorithm

Contents

2.1 Discretizations of the problem

For practical applications, we first assume that the initial discrete image u is $m \times m$ pixels, and that it has been sampled from its continuous version at uniformly spaced points starting at $(0,0)$, i.e., $u(x,y,t) = u(i\Delta x, j\Delta y, k\Delta t)$ for $i,j = 0,\ldots,m-1$ and $k = 0,1,\ldots$ $/\Delta x = \Delta y = 1$ (chosen). The approximate values of u will be denoted by

$$U^k_{i,j} \approx u\left(x_i, y_j, t_k\right).$$

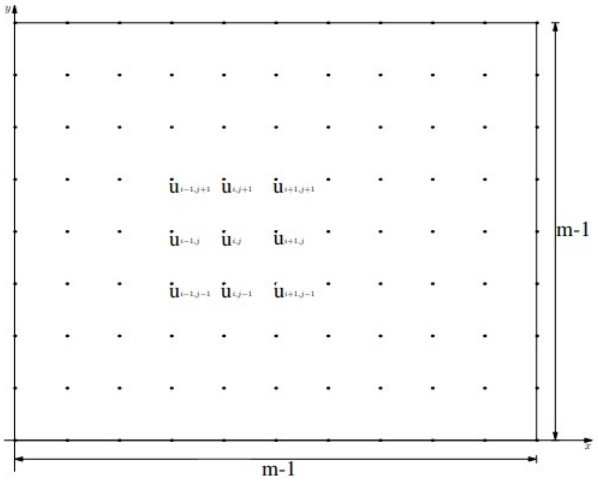


Figure 2.1: Finite-Difference Scheme.

We shall approximate the derivatives by finite differences (explicit scheme).

The simplest difference scheme based at the mesh point (x_i, y_j, t_k) uses a forward difference for the time derivative, this gives

$$\frac{u^{k+1}_{i,j} - u^k_{i,j}}{\Delta t} = \frac{u\left(x_i, y_j, t_{k+1}\right) - u\left(x_i, y_j, t_k\right)}{\Delta t} \approx \frac{\partial u}{\partial t}\left(x_i, y_j, t_k\right),$$

for any function u with a continuous t-derivative. The scheme uses a centred second difference for the second order space derivative

$$\frac{u^k_{i-1,j} - 2u^k_{i,j} + u^k_{i+1,j}}{(\Delta x)^2} = \frac{u\left(x_{i-1}, y_j, t_k\right) - 2u\left(x_i, y_j, t_k\right) + u\left(x_{i+1}, y_j, t_k\right)}{(\Delta x)^2} \approx \frac{\partial^2 u}{\partial x^2}\left(x_i, y_j, t_k\right),$$

then

$$\begin{aligned}\frac{u^{k+1}_{i,j} - u^k_{i,j}}{\Delta t} &= \alpha\left[\left(\frac{u^k_{i-1,j} - 2u^k_{i,j} + u^k_{i+1,j}}{(\Delta x)^2}\right) + \left(\frac{u^k_{i,j-1} - 2u^k_{i,j} + u^k_{i,j+1}}{(\Delta y)^2}\right)\right] \\ &= \alpha\left(u^k_{i,j-1} + u^k_{i-1,j} - 4u^k_{i,j} + u^k_{i+1,j} + u^k_{i,j+1}\right).\end{aligned}$$

Thus

$$\begin{aligned} u_{i,j}^{k+1} &= u_{i,j}^{k} + \Delta t\alpha \left(u_{i,j-1}^{k} + u_{i-1,j}^{k} - 4u_{i,j}^{k} + u_{i+1,j}^{k} + u_{i,j+1}^{k} \right) \\ &= \left(1 - 4\Delta t\alpha\right) u_{i,j}^{k} + \Delta t\alpha \left(u_{i,j-1}^{k} + u_{i-1,j}^{k} + u_{i+1,j}^{k} + u_{i,j+1}^{k} \right). \end{aligned}$$

Coefficient on $u_{i,j}^{k}$ must be non-negative for **stability**[9]. Hence,

$$\left(1 - 4\Delta t\alpha\right) \geqslant 0,$$

so

$$\Delta t \leqslant \frac{1}{4\alpha}.$$

2.2 Anisotrpic Diffusion Algorithm

2.2.1 Approximation of D_α

We use the 2-D discrete Fourier transform (2-D DFT) to compute the **fractional-order difference** D_α. It is one important aspect of the algorithm that it considers the input image as a periodic image, which is equivalent to imposing a period boundary condition on1.7.

The discrete Fourier basis functions are defined on a square domain $\{0, ..., m-1\} \times \{0, ..., m-1\}$, so we need not prolongate the discrete image in practical computations.

The first-order partial difference in the frequency domain is

$$\begin{aligned} D_x^1 u(x,y) &= \frac{\partial u}{\partial x}(x,y) \\ &\simeq \frac{u(x,y) - u(x-\Delta x, y)}{\Delta x} \simeq u(x,y) - u(x-1,y) \\ &\leftrightarrow \left(1 - e^{-j2\pi w_1/m}\right) \hat{u}\left(w_1, w_2\right). \end{aligned}$$

More generally, we define the fractional-order partial difference in the frequency domain

$$D_{\alpha x} u \leftrightarrow \left(1 - e^{-j2\pi w_1/m}\right)^{\alpha} \hat{u}\left(w_1, w_2\right) \quad \alpha \in \mathbb{Z}.$$

Now we use the central difference scheme to compute the fractional-order difference. This is equivalent to translating $D_{\alpha x} u$ by $\alpha/2$ units. Usually, $\alpha/2$ is not an integer, the trigonometric interpolation can be automatically implemented by the translation property 6 of the 2-D DFT, namely

$$\begin{aligned} \mathcal{F}\left\{ u\left(x + \frac{\alpha}{2}, y\right); w_1, w_2 \right\} &= e^{-j\pi 2 w_1 \cdot \left(\frac{-\alpha}{2}\right)/m} \hat{u}\left(w_1, w_2\right) \\ &= e^{j\pi\alpha w_1/m} \hat{u}\left(w_1, w_2\right) \end{aligned} \tag{2.1}$$

where u is the continuous interpolated image by the trigonometric interpolation. Since the formula 2.1 makes sense for all $\alpha/2$ and not only for integer $\alpha/2$, the continuous image is unambiguously defined not just at integer x in the range $0 \leqslant x < m$, but in fact for all real number x. Thus, we obtain the central difference

$$\begin{aligned}\tilde{D}_{\alpha x} u &= D_{\alpha x}\left(u\left(x+\frac{\alpha}{2}, y\right)\right) \\ &\leftrightarrow \left(1-e^{-j2\pi w_1/m}\right)^{\alpha} \times e^{j\pi\alpha w_1/m}\hat{u}\left(w_1, w_2\right),\end{aligned}$$

we can also write

$$\tilde{D}_{\alpha x} u = F^{-1}\left(\left(1-e^{-j2\pi w_1/m}\right)^{\alpha} \times e^{j\pi\alpha w_1/m} F(u)\right) \quad \alpha \in \mathbb{Z}, \tag{2.2}$$

where $\tilde{D}_{\alpha x} u$ is the fractional-order partial difference.

Proposition 3. *[10] When m is an **odd integer**, $\tilde{D}_{\alpha x} u$ is real value.*

Proof. Let

$$p\left(w_1\right) = (1-e^{-j2\pi w_1/m})^{\alpha} \times e^{j\pi\alpha w_1/m}, \quad -\frac{m-1}{2} \leqslant w_1 \leqslant \frac{m-1}{2}.$$

Then we have

$$\begin{aligned}&\operatorname{conj}\left(p\left(-w_1\right)\right) \\ &\quad = \operatorname{conj}\left(\left(1-e^{j2\pi w_1/m}\right)^{\alpha} e^{-j\pi\alpha w_1/m}\right) \\ &\quad = \operatorname{conj}\left(\left(1-e^{j2\pi w_1/m}\right)^{\alpha}\right) \times \operatorname{conj}\left(e^{-j\pi\alpha w_1/m}\right) \\ &\quad = \left(1-e^{-j2\pi w_1/m}\right)^{\alpha} e^{j\pi\alpha w_1/m} \\ &\quad = p\left(w_1\right)\end{aligned}$$

where $\operatorname{conj}(\cdot)$ is the complex conjugation[1], so the function $p\left(w_1\right)$ is conjugate-symmetric with respect to $w_1 = 0$. Thus, the matrix

$$p\left(w_1\right)\hat{u}\left(w_1, w_2\right), \quad -\frac{m-1}{2} \leqslant w_1 \leqslant \frac{m-1}{2}.$$

is conjugate-symmetric with respect to $w_1, w_2 = 0$, so we have

$$\tilde{D}_{\alpha x} u = F^{-1}\left(p\left(w_1\right)\hat{u}\left(w_1, w_2\right)\right) \quad \alpha \in \mathbb{Z},$$

is real value[2]. ∎

When m is an ***even integer***, the domain of definition of the function $p\left(w_1\right)$ is not symmetric with respect to $w_1 = 0$. So usually, there is a complex component in $\tilde{D}_{\alpha x} u$. However, when image size is enough large, this complex component is very small so that it can be ignored in computations.

[1] $\operatorname{conj}(z_1 = i) = -i$, $\operatorname{conj}(z_1 \cdot z_2) = \operatorname{conj}(z_1)\operatorname{conj}(z_2)$, $\quad z_1, z_2 \in \mathbb{C}$.
[2] $\operatorname{conj}(z) = z$, if and only if z is real value, $\quad z \in \mathbb{C}$.

2.2.2 Approximation of D_α^*

Now we turn to compute the adjoint of $\tilde{D}_{\alpha x}$. Let K_1 be a purely diagonal operator[3] in the frequency domain, defined by

$$K_1 = \operatorname{diag}\left(\left(1 - e^{-j2\pi w_1/m}\right)^\alpha e^{j\pi\alpha w_1/m}\right).$$

We get

$$\tilde{D}_{\alpha x} = F^{-1} \circ K_1 \circ F.$$

Let $\tilde{D}_{\alpha x}^*$ be the adjoint of $\tilde{D}_{\alpha x}$, we get

$$\begin{aligned}\tilde{D}_{\alpha x}^* &= \left(F^{-1} \circ K_1 \circ F\right)^* = (K_1 \circ F)^* \circ \left(F^{-1}\right)^* \\ &= F^* \circ K_1^* \circ \left(F^{-1}\right)^* \\ &= F^{-1} \circ K_1^* \circ F.\end{aligned}$$

Since K_1 is a purely diagonal operator, K_1^* is the complex conjugation of K_1, we have

$$\tilde{D}_{\alpha x}^* u \leftrightarrow \operatorname{conj}\left(\left(1 - e^{-j2\pi w_1/m}\right)^\alpha \times e^{j\pi\alpha w_1/m}\right) \hat{u}\left(w_1, w_2\right).$$

The same algorithm is used for calculations of $\tilde{D}_{\alpha y}$ and $\tilde{D}_{\alpha y}^*$.

Proposition 4. *[10] When α is an **even integer**, $\tilde{D}_{\alpha x}$ and $\tilde{D}_{\alpha y}$ are symmetric; when α is an **odd integer**, $\tilde{D}_{\alpha x}$ and $\tilde{D}_{\alpha y}$ are anti-symmetric.*

Proof. Let $\varepsilon = -2\pi w_1/m$, then we have

$$\begin{aligned}K_1\left(w_1, w_1\right) &= (1 - e^{j\varepsilon})^\alpha e^{-\frac{\alpha}{2}j\varepsilon} = (1 + (-e^{j\varepsilon}))^\alpha e^{-\frac{\alpha}{2}j\varepsilon} \\ &= \left[1 + \alpha(-e^{j\varepsilon}) + \frac{\alpha(\alpha-1)}{2!}(-e^{j\varepsilon})^2 + \cdots + \frac{\alpha(\alpha-1)\cdots(\alpha-n+1)}{n!}(-e^{j\varepsilon})^n\right. \\ &\quad \left. + O\left((-e^{j\varepsilon})^{n+1}\right)\right] \times e^{-\frac{\alpha}{2}j\varepsilon} \\ &= \sum_{r=0}^{\alpha} (-1)^r C_\alpha^r e^{(j\varepsilon)^r} e^{-\frac{\alpha}{2}j\varepsilon}\end{aligned}$$

Where $C_\alpha^r = \frac{\alpha!}{r!(\alpha-r)!} = \frac{\alpha(\alpha-1)\ldots(\alpha-r+1)}{r!}$ and $\lim_{x\to 0} O(x) = 0$.

[3] A diagonal operator in the broad sense of the word is an operator D of multiplication by a complex function λ in the direct integral of Hilbert spaces

$$H = \int_M \bigoplus H(t)\,d\mu(t),$$

i.e.

$$(Df)(t) = \lambda(t) f(t), \quad t \in M, \quad f \in H.$$

- When α is an even integer ($\alpha = 2r$), we have

$$(-1)^r C_\alpha^r = (-1)^{\alpha - r} C_\alpha^{\alpha - r},$$

so

$$\begin{aligned}
&K_1(w_1, w_1) \\
&= \sum_{r=0}^{\alpha} (-1)^r C_\alpha^r e^{j\varepsilon\left(r - \frac{\alpha}{2}\right)} = C_\alpha^{\frac{\alpha}{2}} (-1)^{\frac{\alpha}{2}} \\
&+ \sum_{r=0}^{\frac{\alpha}{2}-1} (-1)^r C_\alpha^r e^{j\varepsilon\left(r - \frac{\alpha}{2}\right)} + \sum_{r=\frac{\alpha}{2}+1}^{\alpha} (-1)^r C_\alpha^r e^{j\varepsilon\left(r - \frac{\alpha}{2}\right)} \\
&= C_\alpha^{\frac{\alpha}{2}} (-1)^{\frac{\alpha}{2}} + \sum_{r=0}^{\frac{\alpha}{2}-1} (-1)^r C_\alpha^r \left(e^{j\varepsilon\left(r - \frac{\alpha}{2}\right)} + e^{j\varepsilon\left(\frac{\alpha}{2} - r\right)} \right) \\
&= C_\alpha^{\frac{\alpha}{2}} (-1)^{\frac{\alpha}{2}} + 2 \sum_{r=0}^{\frac{\alpha}{2}-1} (-1)^r C_\alpha^r \frac{\left(e^{j\varepsilon\left(\frac{\alpha}{2} - r\right)} + e^{-j\varepsilon\left(\frac{\alpha}{2} - r\right)} \right)}{2} \\
&= C_\alpha^{\frac{\alpha}{2}} (-1)^{\frac{\alpha}{2}} + 2 \sum_{r=0}^{\frac{\alpha}{2}-1} (-1)^r C_\alpha^r \cos\left(\varepsilon \left(\frac{\alpha}{2} - r \right) \right) \in \mathbb{R}.
\end{aligned}$$

Thus, we have $K_1^*(w_1, w_2) = \mathrm{conj}(K_1(w_1, w_2)) = K_1(w_1, w_2)$, so $\tilde{D}_{\alpha x}^* = \tilde{D}_{\alpha x}$. Thus the symmetric.

- When α is an odd integer ($\alpha = 2r - 1$), we have

$$(-1)^r C_\alpha^r = -(-1)^{\alpha - r} C_\alpha^{\alpha - r},$$

so

$$\begin{aligned}
&K_1(w_1, w_1) \\
&= \sum_{r=0}^{\alpha} (-1)^r C_\alpha^r e^{j\varepsilon\left(r - \frac{\alpha}{2}\right)} \\
&= \sum_{r=0}^{\frac{\alpha-1}{2}} (-1)^r C_\alpha^r e^{j\varepsilon\left(r - \frac{\alpha}{2}\right)} + \sum_{r=\frac{\alpha+1}{2}}^{\alpha} (-1)^r C_\alpha^r e^{j\varepsilon\left(r - \frac{\alpha}{2}\right)} \\
&= \sum_{r=0}^{\frac{\alpha-1}{2}} (-1)^r C_\alpha^r \left(e^{j\varepsilon\left(r - \frac{\alpha}{2}\right)} - e^{j\varepsilon\left(\frac{\alpha}{2} - r\right)} \right) \\
&= -\sum_{r=0}^{\frac{\alpha-1}{2}} (-1)^r C_\alpha^r \left(e^{j\varepsilon\left(\frac{\alpha}{2} - r\right)} - e^{j\varepsilon\left(r - \frac{\alpha}{2}\right)} \right) \\
&= -2j \sum_{r=0}^{\frac{\alpha-1}{2}} (-1)^r C_\alpha^r \frac{\left(e^{j\varepsilon\left(\frac{\alpha}{2} - r\right)} - e^{-j\varepsilon\left(\frac{\alpha}{2} - r\right)} \right)}{2j} \\
&= -2j \sum_{r=0}^{\frac{\alpha-1}{2}} (-1)^r C_\alpha^r \sin\left(\varepsilon \left(\frac{\alpha}{2} - r \right) \right) \in \mathbb{C}.
\end{aligned}$$

Thus, we have $K_1^*(w_1, w_2) = \text{conj}(K_1(w_1, w_2)) = -K_1(w_1, w_2)$, so $\tilde{D}_{\alpha x}^* = -\tilde{D}_{\alpha x}$. Thus the anti-symmetric.

The same method is used to prove the symmetry of $\tilde{D}_{\alpha y}$. ■

2.2.3 Numerical algorithm

Having algorithms of $\tilde{D}_{\alpha x}$ and $\tilde{D}_{\alpha x}^*$, we can easily obtain our noise removal algorithm. Let

$$g = \tilde{D}_{\alpha x}^*(c(|\tilde{D}_\alpha u|^2)\tilde{D}_{\alpha x} u) + \tilde{D}_{\alpha y}^*(c(|\tilde{D}_\alpha u|^2)\tilde{D}_{\alpha y} u),$$

where $\tilde{D}_\alpha u = \left(\tilde{D}_{\alpha x} u, \tilde{D}_{\alpha y} u\right)$, then

$$\hat{g} = K_1^* \circ F(c(|\tilde{D}_\alpha u|^2)\tilde{D}_{\alpha x} u) + K_2^* \circ F((|\tilde{D}_\alpha u|^2)\tilde{D}_{\alpha y} u),$$

where

$$K_1^* = \text{diag}\left(\text{conj}\left((1 - e^{-j2\pi w_1/m})^\alpha \times e^{j\pi\alpha w_1/m}\right)\right),$$
$$K_2^* = \text{diag}\left(\text{conj}\left((1 - e^{-j2\pi w_2/m})^\alpha \times e^{j\pi\alpha w_2/m}\right)\right).$$

We compute the evolution of the initial image u, along flow 1.7.

To summarize, our noise removal approach is done in following steps.

Algorithm 1 Anisotropic Diffusion algorithm

Input: u the input image. **Output:** $\hat{u}$

1: Initialization: $n = 1$, $u_n = u$, k, Δt, $t = k\Delta t$.
2: Compute the 2-D DFT $\hat{u}_n$ of u_n.
3: Compute α-order partial differences $\tilde{D}_{\alpha x} u_n$ and $\tilde{D}_{\alpha y} u_n$ using 2.2.
4: Compute $h_{xn} = c(|\tilde{D}_\alpha u_n|^2)\tilde{D}_{\alpha x} u_n$ and $h_{yn} = c(|\tilde{D}_\alpha u_n|^2)\tilde{D}_{\alpha y} u_n$ in the spatial domain.
5: Compute $\hat{g}_n = K_1^* \circ F(h_{xn}) + K_2^* \circ F(h_{yn})$.
6: Compute $\hat{u}_{n+1} = \hat{u}_n - \hat{g}_n \times \Delta t$.
7: **if** $n = k$ **then**
8: Compute the 2-D IDFT of $\hat{u}_n$.
9: $n = n + 1$.
10: **stop**.
11: **else** go to 3:

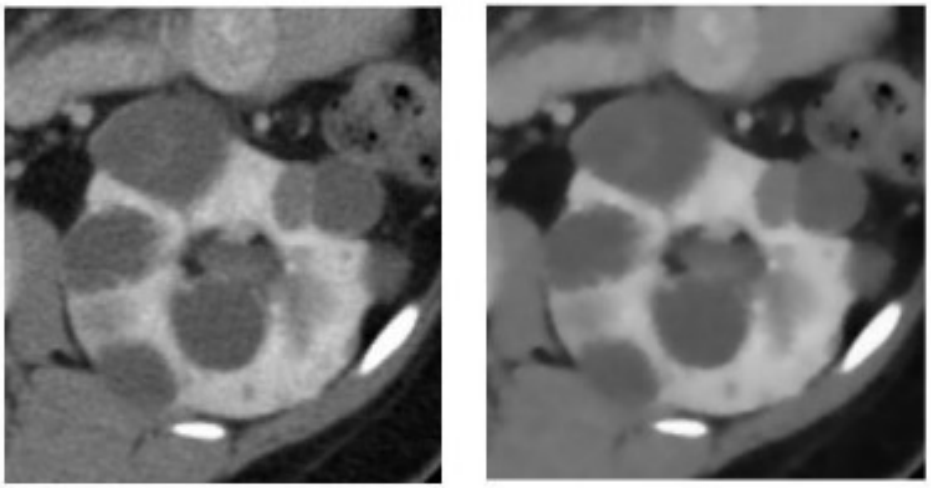


Figure 2.2: An image before and after applying the Anisotropic Diffusion algorithm.

When image size m is 512, the complex component is very small. If we want to eliminate this small complex component, we can extend the observed image using the following method

$$\begin{aligned} uo(x, y) &= u(x, y), \quad 0 \leqslant x, y < m \\ uo(x, m) &= u(x, m-1), \quad 0 \leqslant x < m \\ uo(m, y) &= u(m-1, y), \quad 0 \leqslant y < m \\ uo(m, m) &= u(m-1, m-1). \end{aligned}$$

The size of the extended image uo is $m + 1 = 513$ which is an odd integer. Thus, the complex component goes away in practical computations when applying the proposed algorithm to uo.

Chapter 3

Simulations: Evaluation and Comparison

Contents

We present here the numerical results of the anisotropic diffusion equations based on the minimization of energy E 1.5 for image denoising.

Four different images are used (Lena, Peppers, Clown and Sixteen) with size 256×256. To generate the degraded images, we applied the operator D^{α} and added an additive Gaussian noise σ (black and white) with the condition $\Delta t = 0.02$.

The Gaussian filter $g(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}$ and evaluate the $\mathrm{SNR}_{\mathrm{dB}} = 20 \log_{10}\left(\frac{A_{\mathrm{signal}}}{A_{\mathrm{noise}}}\right)$.

Table 3.1: The statistics of alpha and noise level that we add to the images.

Image	**alpha α**	**G. noise σ**
Lena	$\{1, 1.2, 1.4, 1.6, 1.8, 2\}$	$\{0.001, 0.002, 0.005\}$
Peppers	$\{1, 1.2, 1.4, 1.6, 1.8, 2\}$	$\{0.001, 0.002, 0.005\}$
Clown	$\{1, 1.2, 1.4, 1.6, 1.8, 2\}$	$\{0.001, 0.002, 0.005\}$
Sixteen	$\{1, 1.2, 1.4, 1.6, 1.8, 2\}$	$\{0.001, 0.002, 0.005\}$

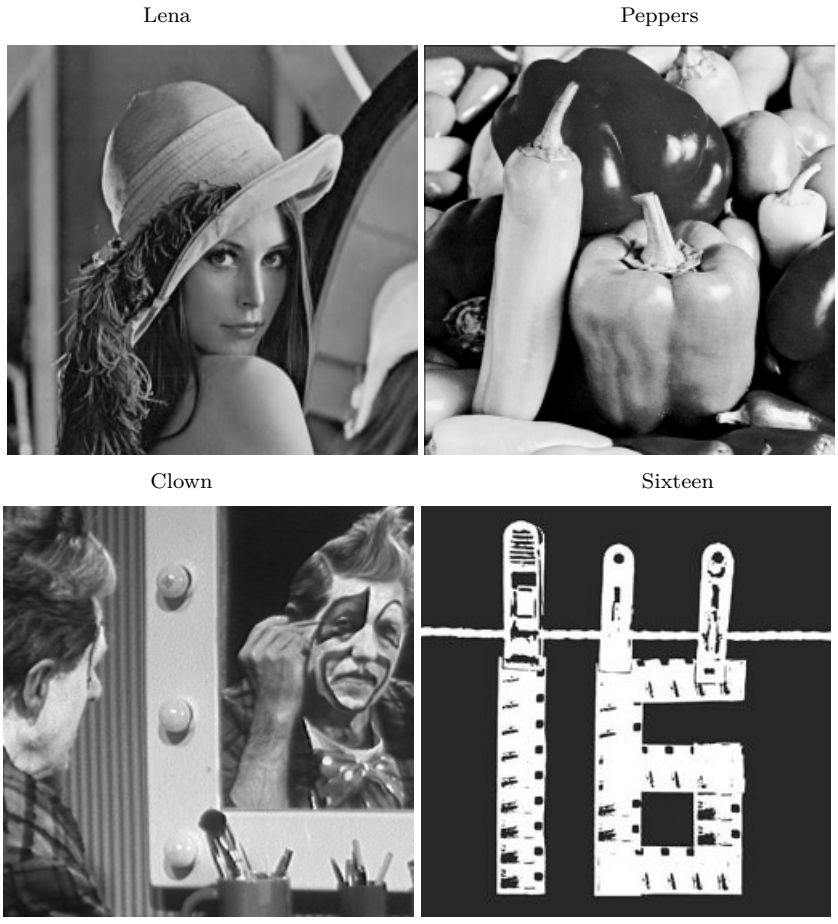


Figure 3.1: Original images.

3.1 Experiment 1

In this experiment, we take $c(s^2) = \exp\left(-\frac{s^2}{2\lambda^2}\right)$, $\lambda = 0.002$ and $k = 5$.

Figure 3.2: From left to right and from top to bottom : Original image. Noised image with ($\sigma = 0.002$). First line : Denoised image with $\alpha = 1$, $\alpha = 1.2$, $\alpha = 1.4$, $\alpha = 1.6$. Second line : $\alpha = 1.8$, $\alpha = 2$.

Figure 3.3: From left to right and from top to bottom : Original image. Noised image with ($\sigma = 0.001$). First line : Denoised image with $\alpha = 1$, $\alpha = 1.2$, $\alpha = 1.4$, $\alpha = 1.6$. Second line : $\alpha = 1.8$, $\alpha = 2$.

Figure 3.4: From left to right and from top to bottom : Original image. Noised image with ($\sigma = 0.002$). First line : Denoised image with $\alpha = 1$, $\alpha = 1.2$, $\alpha = 1.4$, $\alpha = 1.6$. Second line : $\alpha = 1.8$, $\alpha = 2$.

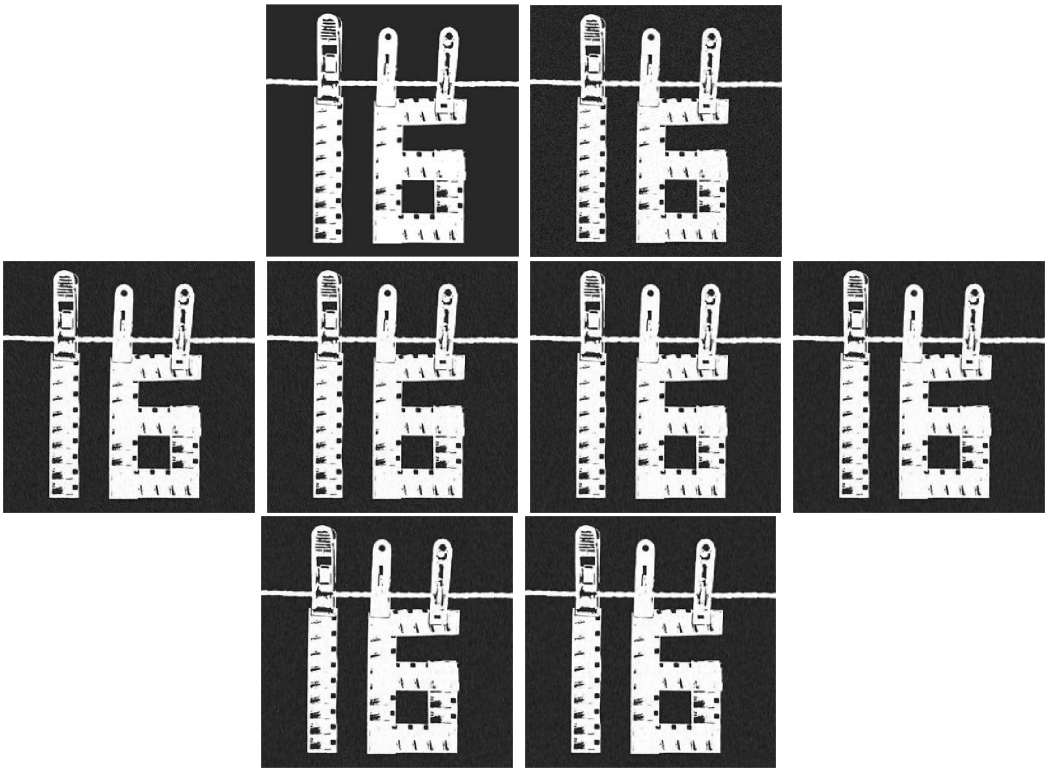


Figure 3.5: From left to right and from top to bottom : Original image. Noised image with ($\sigma = 0.001$). First line : Denoised image with $\alpha = 1$, $\alpha = 1.2$, $\alpha = 1.4$, $\alpha = 1.6$. Second line : $\alpha = 1.8$, $\alpha = 2$.

Image	**Lena**						**Peppers**					
α	1	1.2	1.4	1.6	1.8	2	1	1.2	1.4	1.6	1.8	2
$\sigma = 0.001$	24.04	24.26	25.07	25.47	25.53	25.47	26.33	26.61	27.33	27.92	28.14	28.20
$\sigma = 0.002$	20.97	21.13	21.68	21.99	22.03	21.84	23.41	23.61	24.20	24.61	24.78	24.70
$\sigma = 0.005$	16.61	16.65	16.79	16.95	16.86	16.82	19.26	19.26	19.53	19.66	19.67	19.55
Image	**Clown**						**Sixteen**					
α	1	1.2	1.4	1.6	1.8	2	1	1.2	1.4	1.6	1.8	2
$\sigma = 0.001$	25.57	25.82	26.52	26.98	27.03	27.03	25.90	26.20	27.10	27.71	28.03	28.05
$\sigma = 0.002$	22.52	22.71	23.23	23.59	23.65	23.44	22.98	23.21	23.77	24.17	24.38	24.13
$\sigma = 0.005$	18.15	18.18	18.34	18.52	18.50	18.41	18.73	18.77	18.90	19.10	19.03	18.96

Table 3.2: SNR with different choices of α across three noise levels.

3.2 Experiment 2

In this next experiment, we compare anisotropic diffusion algorithm $\alpha = 3/2 = 1.5$ with the common Perona–Malik algorithm $\alpha = 1$ in various iterations k.

Figure 3.6: From left to right and from top to bottom : Original image. Noised image with ($\sigma = 0.001$). First line : Denoised image $\alpha = 1$ with $k = 50$, $k = 20$, and $\alpha = 1.5$ with the same values of k.

Figure 3.7: From left to right: Original image (enlargement of the Lena's face), denoised image $\alpha = 1$ ($k = 20$), $\alpha = 1.5$ ($k = 20$), and $\alpha = 2$ ($k = 20$).

Figure 3.8: From left to right and from top to bottom : Original image. Noised image with ($\sigma = 0.002$). First line : Denoised image $\alpha = 1$ with $k = 50$, $k = 20$, and $\alpha = 1.5$ with the same values of k.

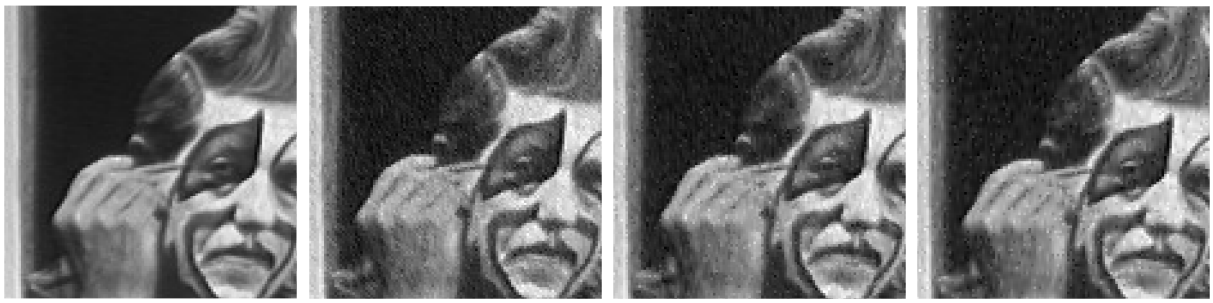


Figure 3.9: From left to right: Original image (enlargement of the Clown's face), denoised image $\alpha = 1$ ($k = 10$), $\alpha = 1.5$ ($k = 10$), and $\alpha = 2$ ($k = 10$).

Table 3.3: SNR with different choices of α across the noise level σ and k iterations.

Image	**Lena**		**Clown**	
σ	0.001		0.002	
α	1	1.5	1	1.5
$k = 50$	22.53	24.74	21.50	23.75
$k = 20$	23.98	25.28	22.50	23.52

3.3 Experiment 3

Now, we compare the various results of the diffusion coefficient $c_1(s) = \frac{1}{1+s}$, $c_2(s^2) = \frac{1}{1+(x/\lambda)^2}$, $c_3(s^2) = \exp\left(-\frac{s^2}{2\lambda^2}\right)$, $\lambda = 0.002$ and $k = 20$.

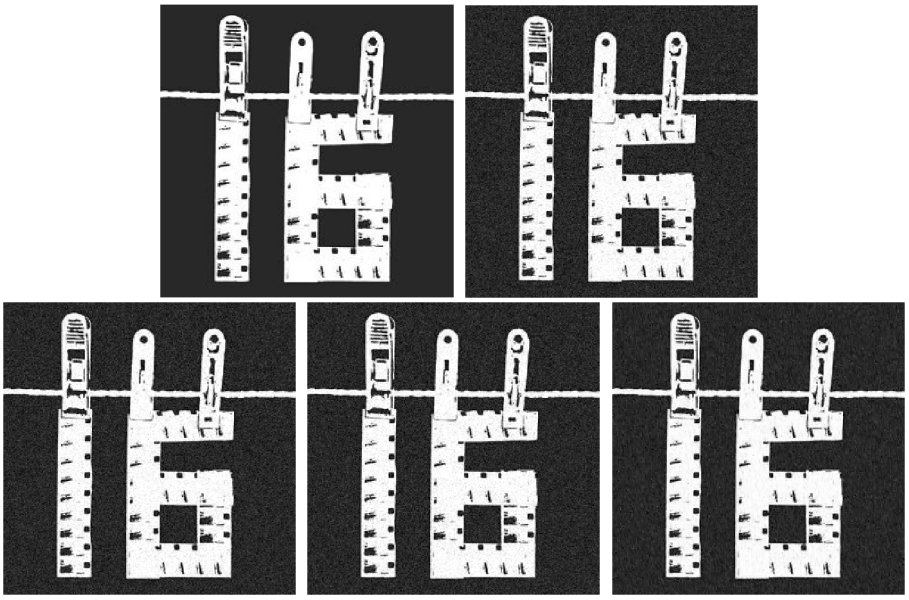


Figure 3.10: From left to right and from top to bottom : Original image. Noised image with ($\sigma = 0.002$). First line : Denoised image $\alpha = 1.5$ with $c_1, SNR = 21.89dB$, $c_2, SNR = 21.94dB$, $c_3, SNR = 24.07dB$.

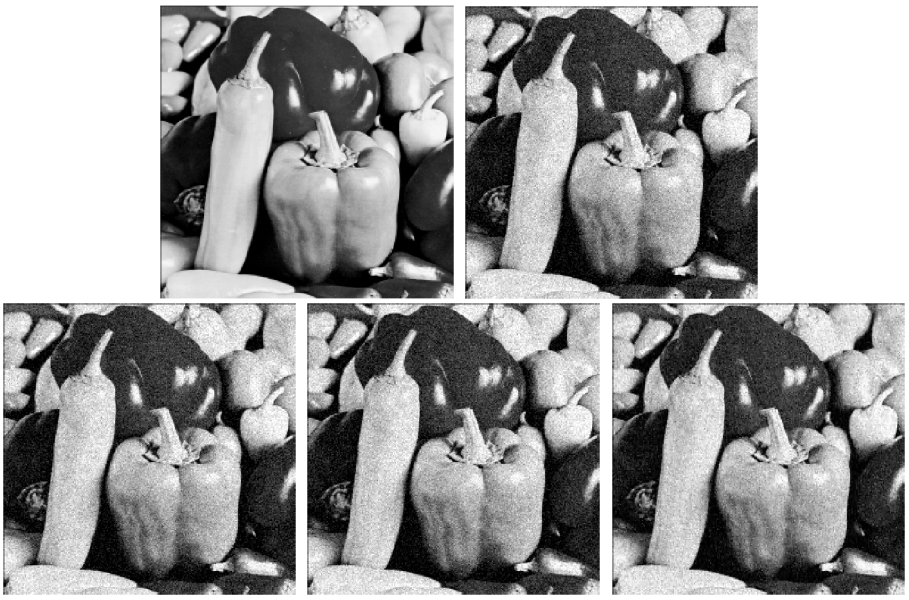


Figure 3.11: From left to right and from top to bottom : Original image. Noised image with ($\sigma = 0.005$). First line : Denoised image $\alpha = 1.5$ with $c_1, SNR = 18.66dB$, $c_2, SNR = 18.74dB$, $c_3, SNR = 19.59dB$.

3.4 Discussion

We note that the proposed fractional-order pseudo-PDEs obtain better SNR and visual effects. i.e. Notice the growing of SNR and almost the same while the α takes the non-integer value, however, the SNR decreases with $\alpha = 1$, $\alpha = 2$.

We also note that the noninteger-order $\alpha = 1.2, 1.4, 1.6, 1.8$ models make clear improvements because the average is less sensitive to noise.

We intentionally choosed a large value for iteration number so that the smoothing effects are easy to see. The images resulting from the second-order PDE $\alpha = 1$ (Perona-Malik equation) look "blocky" and produce false edges especially on Lena's face. The denoised images using the fourth-order PDE $\alpha = 2$ contain speckle artifacts which are isolated black and white speckles (see the lena's face and exactly in clown's hair). The images resulting from the proposed third-order pseudo-PDE $\alpha = 1.5$ look natural and do not produce false edges. Speckle artifacts are also avoided.

In the experiment 3 we tested the application of three types diffusion coefficients c_1, c_2, c_3, that we noted the c_3 which defined by a function exponential gives better result, clear look, plus grow of SNR.

We may consider the best parameters as $\alpha = 1.5, \sigma = 0.001$, $c_3\left(s^2\right) = \exp\left(-\frac{s^2}{2\lambda^2}\right)$, $\lambda = 0.002$, $k = 20$.

Figure 3.12: Best result SNR $= 25.28$dB.

Conclusion

As a conclusion. This work included a mention of the relationship between fractional derivation and image enhancement.

In a first chapter, we presented basic notions of fractional-order derivative (R-L, Caputo, G-L derivatives). We also presented a class of fractional-order anisotropic diffusion models, like the second-order (Perona-Malik model) and fourth-order anisotropic diffusion models.

In a second chapter, we give an iterative algorithm in the frequency domain using the 2-D DFT and apply it to image denoising. Although the use of the 2-D DFT leads to consider the input image as a periodic image. The fractional derivative operator $\mathsf{D}_{\alpha}\mathfrak{u}$ $= (\mathsf{D}_{\alpha x}, \mathsf{D}_{\alpha y})$ can be take any type of three derivatives.

In a third chapter, through the some algorithm executions in **Matlab**, we cited the main advantage of anisotropic diffusion algorithm is that the proposed pseudo-PDEs exhibit higher perceptual quality than second-order and fourth-order PDEs. The second-order PDE ($\alpha = 1$) preserves edges but has the sometimes undesirable staircase effect, the fourth-order PDE ($\alpha = 2$) avoids the blocky effect but suffers from speckle artifacts, the proposed third-order pseudo-PDE ($\alpha = 1.5$) avoids the blocky effect and has no speckle artifacts. Another advantage is that the noninteger-order anisotropic diffusion can improve the SNR, however, the SNR cannot be improved if the integer-order models are used.

Appendix

The MATLAB code

```
%Anisotrpic Diffusion Algorithm.
clear all; clc

%%
% Initialization.
alpha = 1.5;
dx = 1; dy = 1;
dt = 1./(4.^alpha);
%c = @(x) 1./(1000.^10+x);
%c = @(x) 1./(1+(x/0.002)^2);
c = @(x) exp(-x.^2/2*(0.002).^2);
%%
% 1) Input image : squared matrix.
u = imread('rice.png'); %Input image u.
subplot(1,3,1);imshow(u); title('Original image');
[m,~] = size(u);
u(m+1,:) = u(m,:); u(:,m+1) = u(:,m);
m=m+1;
uref = u;
u = imnoise(u,'gaussian',0,0.001);
x = dx * 0:m-1; y = dy * 0:m-1;

%%
%w1 = exp(-2*pi*1i./x); w2 = exp(-2*pi*1i./y);
w = -(m-1)/2:(m-1)/2;
w1 = repmat(w',1,m);
w2 = repmat(w,m,1);
```

```
28
29 %%
30 K1 = diag( conj((1-exp(-1i*2*pi*w/m) ).^alpha .*exp(1i*pi*alpha
       *w/m) ));
31 %K2 = diag( conj((1-exp(-1i*2*pi*w/m) ).^alpha .*exp(1i*pi*alpha
       *w/m) ));
32 K2 = K1;
33
34 NIter = 20;
35 u0 = u;
36 subplot(1,3,2);imshow(u0); title('Noisy image');
37 %2) Compute DFT of the image u.
38 Fu = fft2(u0); Fu = fftshift(Fu); %imagesc(log ( + abs(Fu)))
39 Filtx = (1-exp(-1i*2*pi*w1/m)).^(alpha).* exp(1i*pi*alpha*w1/m);
40 Filty = (1-exp(-1i*2*pi*w2/m)).^(alpha).* exp(1i*pi*alpha*w2/m);
41 for i=1:NIter
42
43     %%
44     %3) Compute alpha-order partial differences.
45     D_alphaX=ifft2(ifftshift(Filtx.*Fu));
46     D_alphaY=ifft2(ifftshift(Filty.*Fu));
47
48     %%
49     % 4) Compute hx, hy.
50     hx = c(D_alphaX.*D_alphaX).*D_alphaX;
51     hy = c(D_alphaY.*D_alphaY).*D_alphaY;
52
53     g = K1* fftshift(fft2(hx)) + K2* fftshift(fft2(hy));
54
55     Fu = Fu - g * dt;
56
57     v = ifft2(ifftshift(Fu)); v = uint8(round(v));
58     subplot(1,3,3);imshow(v); title('Denoised image');drawnow();
59
60 end
61  % Measure signal-to-noise ratio (SNR).
62 SNR = snr( double(uref), double(uref)-double(v) );
63 fprintf('\n The SNR value is %0.4f \n', SNR);
```

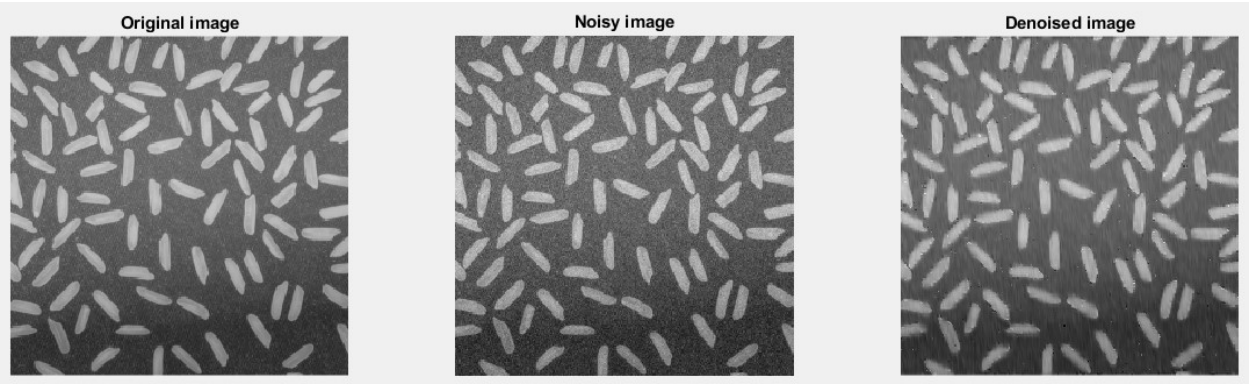


Figure 13: $\mathsf{SNR} = 22.32\mathrm{dB}$.

Bibliography

[1] C. Canuto, A. Tabacco, Mathematical Analysis II, Springer-Verlag Italia, Milano 2008.

[2] J. B. Conway, A Course in Functional Analysis, second edition, Graduate texts in mathematics 96, Springer-Verlag New York, Inc 1990.

[3] A. A. Kilbas, H. M. Srivastava, J. J. Trujillo, Theory and Applications of Fractional Differential Equations, North-Holland Mathematics Studies 204, editor: Jan van Mill, 2006.

[4] S. Etemad, Sh. Rezapour, On a Two-Variables Fractional Partial Differential Inclusion Via Riemann-Liouville Derivative, Novi Sad J. Math, Vol. 46, No. 2, 45-53, 2016.

[5] I. Podlubny, Y. Chen, Adjoint Fractional Differential Expressions and Operators, DETC2007-35005, Las Vegas, Nevada, USA, September 4-7, 2007.

[6] S. Mallat, A Wavelet Tour of Signal Processing, October 9, 2008.

[7] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," IEEE Trans. Pattern Anal. Mach. Intell., vol. 12, no. 7, pp. 629–639, Jul. 1990.

[8] Y. L. You and M. Kaveh, "Fourth-order partial differential equations for noise removal," IEEE Trans. Image Process., vol. 9, no. 10, pp. 1723–1730, Oct. 2000.

[9] D. M. Causon, C. G. Mingham, Introductory Finite Difference Methods for PDEs, Ventus Publishing ApS ISBN 978-87-7681-642-1, 2010.

[10] J. Bai and X. C. Feng, "Fractional-order anisotropic diffusion for image denoising," IEEE Trans. Image Process., vol. 16, no. 10, pp. 2492–2502, 2007.