DEMOCRATIC AND POPULAR REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
Mohamed Boudiaf University of Msila
Faculty of Mathematics and Computer Sciences
Department of Mathematics

Master Thesis

Field : Mathematics and Computer Sciences
Branch : Mathematics
Option: Partial Differential Equations and applications

Theme

Osmosis Filtering for Shadow Removal in Images

Presented by :
GUETTAF Amira

Before the jury composed of :

Mr MERZOUGUI Abdelkrim	Prof,	University of M'sila	**President**.
Mr CHOUDER Rafaa	M.C.B,	University of M'sila	**Supervisor**.
Mr BENHAMIDOUCHE Noureddine	Prof,	University of M'sila	**Examiner**.

University year 2022/2023

Achnowledgments

First and foremost, I would like to extend my sincere thanks to Dr. CHOUDER Rafaa for his assistance and support throughout the completion of this work. Especially for his careful reading of the final version and insightful discussions and constructive comments, which have been of great value.

I would like to thank all my loved ones, my family, and my friends, for their love and support at all times.

I express my gratitude to the members of the jury, each by name, for accepting to examine my work. Not forgetting my parents for their contribution, support, and patience.

I thank the administration of the Faculty of Mathematics and Computer Science, the Mathematics Department, for their technical support.

Lastly, I would like to thank everyone who has helped me in any way, the teachers, colleagues, and friends.

Guettaf Amira

Dedication

I dedicate this work:
To my dear mother, who stayed up, raised me, and worked tirelessly. May God preserve her and grant her a long life.

To my compassionate father, who toiled and endured hardships in order to take care of me. May God protect and bless him.

To my siblings:Omar,Chaomaa,Aymen , Chourouk and to my family,who bestows upon me love and vitality.

To all my friends who have always encouraged me, and to whom I wish continued success.

To all those who have helped me, near or far, and to those who have shared the emotional moments during the completion of this work and warmly supported and encouraged me throughout my journey.

Thanks

Guettaf Amira 

Contents

Notations

$\mathbf{x}$	vector $\mathbf{x}$ of spatial coordinates in an image
x, y, z	abscissa, ordinate, side of a point of an image
$\|x\|$	absolute value of the real x
$\|\mathbf{v}\|, \|\mathbf{v}\|$	vector norm of $\mathbf{v}$
∇	gradient operator
div	divergence operator
Δ	Laplacian operator
$*$	convolution operator
T	transpose operator
$\times$	matrix product
.	scalar product
$\mathbb{N}$	The set of natural numbers.
$\mathbb{R}, \mathbb{R}^d$	the set of real numbers and its d–Cartesian product, endowed with the Euclidean norm $\|.\|_2$
U_x, U_y	derived of U respectively according to the variables x and y
$PDEs$	partial differential equations
AOS	additive operator splitting
ADI	alternating direction implicit
MOS	multiplicative operator splitting
$AMOS$	additive and multiplicative operator splitting
PR	the peaceman-rachford scheme
D	the douglas scheme

Introduction

Partial differential equations (PDEs) provide a natural framework for modeling various processes in real-world applications, including physics, life sciences, and economics. It is therefore not surprising that they have made significant contributions to the mathematical foundations of signal and image analysis. For example, PDEs appear as Euler-Lagrange equations when solving continuous optimization problems arising from variation models [2] [5] or regularization of ill-posed problems [3]. They have also been recognized as the natural framework for scale-spaces [1] and have been successfully employed in image enhancement [20], inpainting [16], and image compression [9]. PDE-based models have benefited from decades of research on their theoretical foundations and efficient numerical algorithms. Due to their continuous nature, it is also straightforward to incorporate useful properties such as rotation invariance. Recently, some researchers proposed osmosis models for shadow removal [18], [6]. Osmosis is non symmetric diffusion-like filters which lead to non-constant steady states [18], [17], [12]. Osmosis models are invariant to multiplicative perturbations and therefore are adapted to shadow removal. On the other hand, osmosis can be applied as a shadow removal filter in a number of imaging challenges: image reconstruction from compressed data or seamless image fusion, image fusion, art diagnostic and virtual restoration.
The osmosis filtering method is based on the mathematical theory of partial differential equations (PDEs). The basic idea is to modify the image using an iterative process that is guided by a diffusion equation. This equation models the behavior of a substance that diffuses through a medium, such as water diffusing through a membrane.
To summarize, in light of the aforementioned points, it is evident that osmosis filtering offers significant advantages over other shadow removal techniques. Its ability to eliminate shadows without introducing artifacts or blurring the image, while preserving the edges and details, sets it apart. Additionally, its capacity to handle complex shadows commonly encountered in real-world scenarios adds to its appeal.
Consequently, osmosis filtering emerges as a powerful and promising approach for shadow removal in digital images. By considering the local structure and tex-

ture of the image, it achieves impressive results while maintaining the integrity of the image. Its potential extends across various fields, including computer vision, medical imaging, and photography, where it can significantly improve image quality and enhance visual analysis and interpretation.

In this work, we study some linear and non-linear filters and their effect on images processing. This work is divided into four chapters. The first chapter deals with the image from a mathematical point of view (definitions and applications of the image in reality), we ended this chapter by some PDEs used in image processing field. The second chapter devoted with isotropic and anisotropic diffusion, where we have studied it in terms of finite difference approximations, stability of types (Explicit-Implicit schemes) to its applications in image processing. In the next chapter, we dealt with linear osmosis, where we studied discretesation by the ADI method. In the last chapter we discuss the application of anisotropic osmosis filtering for shadow removal. We use the AOS method to solve numerically this model.

CHAPTER 1

PDEs IN IMAGE PROCESSING

Partial differential equations (PDEs) are one of the most important mathematical tools in image processing and computer vision. As they are appear in diffusion methods as well as in the Euler-Lagrange equations of variational models, they have been considered appear in many image processing tasks. Consequently, this chapter cannot give a complete overview about the field of PDEs based image processing. Instead we pick out mainly those methods that will be utilized in the other chapters of this thesis.

1.1 Images and pictures in our world

Human beings are predominantly visual creatures: we rely heavily on our vision to make sense of the world around us. We not only look at things to identify and classify them, but we can scan for differences, and obtain an overall rough "feeling" for a scene with a quick glance. Humans have evolved very precise visual skills: we can identify a face in an instant; we can differentiate colours; we can process a large amount of visual information very quickly. However, the world is in constant motion: stare at something for long enough and it will change in some way. Even a large solid structure, like a building or a mountain, will change its appearance depending on the time of day (day or night); amount of sunlight (clear or cloudy), or various shadows falling upon it. We are concerned with single images: snapshots, if you like, of a visual scene. Although image processing can deal with changing scenes, we shall not discuss it in any detail in this text. For our purposes, an image is a single picture which represents something. It may be a picture of a person, of people or animals, or of an outdoor scene, or a microphotograph of an electronic component, or the result of medical imaging.

Even if the picture is not immediately recognizable, it will not be just a random blur. It captures a moment, an event that it freezes and offers us to analysis, to ameliorate their quality, enhance some characteristics to efficiently combine different pieces of information. Advertising, Photography, video games, etc..., our daily lives of images.

1.2 What is image processing?

Image processing involves changing the nature of an image in order to either
• improve its pictorial information for human interpretation.
• render it more suitable for autonomous machine perception.
We shall be concerned with digital image processing, which involves using a computer to change the nature of a digital image (see below). It is necessary to realize that these two aspects represent two separate but equally important aspects of image processing. A procedure which satisfies condition (1)—a procedure which makes an image "look better"—may be the very worst procedure for satisfying condition (2). Humans like their images to be sharp, clear and detailed; machines prefer their images to be simple and uncluttered. Examples of (1) may include:

1.2.1 Imaging Problems

A digitally sampled image u° ,which is assumed to belong to mathematical space Y,is obtained as a modified version of the desired image u,which belongs to another space X.where X and Y ara two Banach spaces .u° image corrupted by noise,is very often modelled by a bounded linear transformation operator $S; X \to Y$,reading as: see [12]

$$u^{\circ}(x) = Su(x) \tag{1.1}$$

The clean image is recoved from the observed corrupted u°:this problem is called *inverse problem*.
Different operator S lead to different imaging problems.

• **Denoising**:Removing noise from a noisy image,there are types of noise:
$*$ Gaussian noise: it is an additive random noise of the type

$$u^{\circ}(x) = u(x) + \eta(x) \tag{1.2}$$

Where $\eta(x)$ is modelled by a Gaussian white noise distribution.
$*$ Poission noise:it is an intensity-dependent random noise.
$*$ speckle noise:it is multiplicative random noise of the type

$$u^{\circ}(x) = u(x) + n(x)\,u(x) \tag{1.3}$$

Where $n(x)$ is unformly distributed.
$*$ salt and peper noise: it is an impulsive random noise of the type

$$u^{\circ}(x) = (1 - s(x))\, u(x) + s(x)\, c(x) \tag{1.4}$$

where $c(x)$ and $s(x)$ are independent random fielrs of independent 0-1 binary random variable.

• **Deblurring**:Reconstruction of a visually better image from a blurred image.
$x = (x1, \ldots, xd)$
$u^{\circ} = K_{\varsigma} * u$

$$K_{\varsigma}(x) = \frac{1}{(\sqrt{2\pi}\varsigma)^d} \exp\left(-\frac{|x_1^2 + \ldots + x_d^2|}{2\varsigma^2}\right) \tag{1.5}$$

• **Inpainting**: In brief, the domain of the image u° can be divided into two parts: the occluded part, denoted as O, or the inpainting domain, which consists of values from another overlapped image f that need to be replaced, and the intact part, denoted as O^C, where the image should remain unchanged. The fundamental condition is that the entire image domain Ω is the union of O and O^C. As a result, the image u° can be represented by the following transformation:

$$u^{\circ}(x) = \chi_{\Omega\emptyset} u(x) + \chi_O f(x) \tag{1.6}$$

where χ_O is a binary mask indicator function, indicating if a pixel belongs or not to O:

$$\chi_O = \begin{cases} 1 & \text{if } x \in O \\ 0 & \text{otherwise} \end{cases} \tag{1.7}$$

• **Zoming** : we take part of the image in ordre to enlarge it,i.e reconstructing data in pixels is enlarged by a certain amount in ordre to clarify the details of the image.

• **Edge dectection**:Detect the edges of the image.

• **Segmentation**:dividing the image.

•**Shadow removal,de-hasing and other challenges**:In the shadow removal problem a shadow is projected over an object and the take is to recover the illumination.In the de-hazing problem the image is affected by a certain amount of the image.

$$u^{\circ}(x) = c(x)\, u(x)\, forc(x) \in [0, 1] \tag{1.8}$$

1.3 Some applications

Image processing has an enormous range of applications; almost every area of science and technology can make use of image processing methods. Here is a

short list just to give some indication of the range of image processing applications. see Figure 1.1

1. Medicine

• Inspection and interpretation of images obtained from X-rays, MRI or CAT scans.

• analysis of cell images, of chromosome karyotypes.

2. Agriculture

• Satellite/aerial views of land, for example to determine how much land is being used for different purposes, or to investigate the suitability of different regions for different crops.

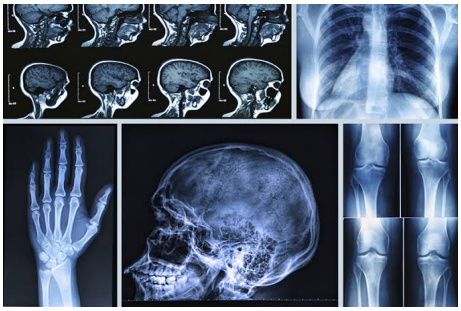
• inspection of fruit and vegetables—distinguishing good and fresh produce from old.

3. Industry

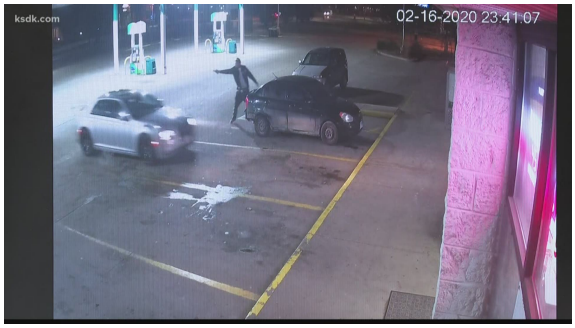
• Automatic inspection of items on a production line, inspection of paper samples.

4. Law enforcement

• Fingerprint analysis, sharpening or deblurring of speed-camera images.



(a) X-Ray, CT, and MRI Scans



(b) Surveillance Camera Scans

(c) A fingerprint Image

(d) Pregnancy Follow-Up Picture

(e) Satellite farmland image

Figure 1.1: Figures showing some applications of image processing

1.4 PDEs in Image Processing

1.4.1 Mathematical description of an image

Digital images can be rephrased as a vectorial graph function, denoted as u°, which operates on a discrete two-dimensional rectangular imaging domain Ω with dimensions $M \times N$. In this context, M represents the height of the image, and N represents the width. Each element within this collection of matrices is referred to as a pixel, and it corresponds to an integer interval within the range $[0, 2^r - 1]^C$.

$$u^\circ : \Omega \to [0, 2^r - 1]^C$$

where the colour information is discretized over a number of C parallel retangular matrice and r is length. We can distinguish between grey-scal image when $C = 1$,denoted by $u^\circ$and vectorial colour in bold when $C > 1$.
the discrete grid

$$(x, y) \longmapsto (M + 1 - y, x) = (i, j)\ for x, j = 1, \ldots, N\ and\ y, i = 1, \ldots, M$$

1.4.2 Linear diffusion filtering

The first step to use PDEs for smoothing images was done in the beginning of eighties, when the idea of scale-space filtering has introduced by Witkin [24] and further developed by Koenderink [11].
The essential idea of this approach is quite simple: embed the original image in a family of derived images $u(x, y, t)$ obtained by convolving the original image $u_0(x, y)$ with a Gaussian kernel $G_{\sqrt{2t}}(x, y)$ of variance t:

$$u(x, y, t) = \left(G_{\sqrt{2t}} * u_0\right)(x, y),$$

where

$$G_\sigma(x, y) = \frac{1}{2\pi\delta^2} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right).$$

This family of derived images may equivalently be viewed as the solution of the following heat equation or the linear diffusion equation

$$\begin{cases} \frac{\partial u}{\partial t} = \triangle u \\ u(x, y, 0) = u_0(x, y) \end{cases},$$

for a function $u(x, y, t)$ is the smoothed image at t, and $\triangle = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ denote Laplacian operator, with the initial condition $u(x, y, 0) = u_0(x, y)$, the original image.
Koenderink [11] motivates the diffusion equation formulation by stating two criteria:

- **Causality**: Any feature at a coarse level of resolution is required to possess a (not necessarily unique) "cause" at a finer level of resolution although the reverse need not be true. In other words, no spurious detail should be generated when the resolution is diminished.

- **Homogeneity and Isotropy**: The blurring is required to be space invariant.

The linear diffusion filter has its limitations: whether we smooth uniformly by a rotational symmetric Gaussian kernel, or diffuse the data equally in all directions, the process not only removes undesirable local extrema (noise) but also deforms important features of the image, blurs and dislocates edges. To overcome these drawbacks, we have to move to nonlinear filters; nonlinear diffusion offers an excellent alternative.

1.4.3 Nonlinear diffusion filtering

Overcoming the undesirable effects of linear smoothing filtering, such as blurring or dislocating the semantically meaningful edges of the image, nonlinear diffusion equations can be used because the nonlinear diffusion technique not only preserves the edge sharpness, it may also enhance it. This technique was firstly proposed by Perona and Malik [13] by stating three criteria:

- **Causality:** no spurious detail should be generated passing from finer to coarser scales.

- **Immediate localization:** boundaries should be sharp and coincide with the semantically meaningful boundaries at that resolution.

- **Piecewise smoothing:** intra-region smoothing should be preferred to inter-region smoothing.

To satisfy the second and third criteria, Perona and Malik proposed to change the diffusion coefficient D, (D is constant in linear diffusion); by introducing a space-time-variant diffusion coefficient. Therefore the Perona-Malik model can be written as

$$\begin{cases} \frac{\partial u}{\partial t} = div(g\left(|\nabla u|\right)\nabla u) \\ u(x,y,0) = u_0(x,y) \end{cases},$$

where div denotes the divergence operator, $u(x,y,t)$ is the smoothed image at time step t, $|\nabla u|$ is the gradient magnitude of u, and $g(x)$ is the diffusivity function. $g(x)$ should be a nonnegative, monotonically decreasing function with $g(0) = 1,$ so that the diffusion is maximal within uniform regions, and approaching $zero$ at infinity, so that the diffusion is stopped across edges.

Nonlinear diffusion filtering can successfully smooth noise while respecting the region boundaries and small structures within the image, as long as some of

its crucial parameters are determined or estimated correctly. According to Perona and Malik the choice of functions g leads to the desirable result of edges enhancement.

1.4.4 Osmosis filtering

The isotropic drift-diffusion PDE, which is called osmosis for its analogies with the physical process, compared to standard plain diffusion models, the osmosis model considers an additional drift term, making the process asymmetric [18, 17, 12].
For a regular domain $\Omega \subset \mathbb{R}^2$, a given vector field $d : \Omega \to \mathbb{R}^2$ and a given image $f \in L^\infty(\Omega, \mathbb{R})$, the isotropic osmosis model reads [18]

$$\begin{cases} \partial u = \Delta u - div(du) \quad \text{on } \Omega \times (0,T], \\ u(0,x) = f(x) \quad \text{on } \Omega, \\ \langle \nabla u - du, n \rangle = 0 \ \text{ on } \ \partial\Omega \times (0,T], \end{cases} \tag{1.9}$$

where $\langle ., . \rangle$ denotes the Euclidean scalar and n the outer normal vector on $\partial\Omega$.

1.4.5 Finite-Difference Approximation to Derivatives

Let a function U derivatives and single-valued, finite and continuous functions of x, then by Taylor's theorem.

$$U(x+h) = U(x) + hU'(x) + \frac{1}{2}h^2U''(x) + \frac{1}{6}h^3U^{(3)}(x) + \frac{1}{4!}h^4U^{(4)}(x) + \dots \tag{1.10}$$

and

$$U(x-h) = U(x) - hU'(x) + \frac{1}{2}h^2u''(x) - \frac{1}{6}h^3U^{(3)}(x) + \frac{1}{4!}h^4U^{(4)}(x) + \dots \tag{1.11}$$

Addition of these expansions gives

$$U(x+h) + U(x-h) = 2U(x) + h^2U''(x) + O\left(h^4\right)$$

where $O(h^4)$ denotes terms containing fourth and higher powers of h. Assuming these are negligible in comparison with lower powers of h it follows that,

$$U''(x) = \left(\frac{\mathrm{d}^2U}{\mathrm{d}x^2}\right)_{x=x} \approx \frac{1}{h^2}\{U(x+h) - 2U(x) + U(x-h)\} \tag{1.12}$$

with a leading error on the right-hand side of order h^2.
Subtraction of equation (1.11) from equation (1.10) and neglect of terms of order h^3 leads to

$$U'(x) = \left(\frac{\mathrm{d}U}{\mathrm{d}x}\right)_{x=x} \approx \frac{1}{2h}\{U(x+h) - U(x-h)\} \tag{1.13}$$

with an error of order h^2. equation (1.13) clearly approximates the slope of the tangent at the point P by the slope of the chord AB, and is called a central-difference approximation (Figure 1.2). We can also approximate the slope of the tangent at P by either the slope of the chord PB, giving the forward-difference formula,

$$U'(x) \approx \frac{1}{h}\{U(x+h) - U(x)\},$$

or the slope of the chord AP giving the backward-difference formula,

$$U'(x) \approx \frac{1}{h}\{U(x) - U(x-h)\},$$

with an error of order h.

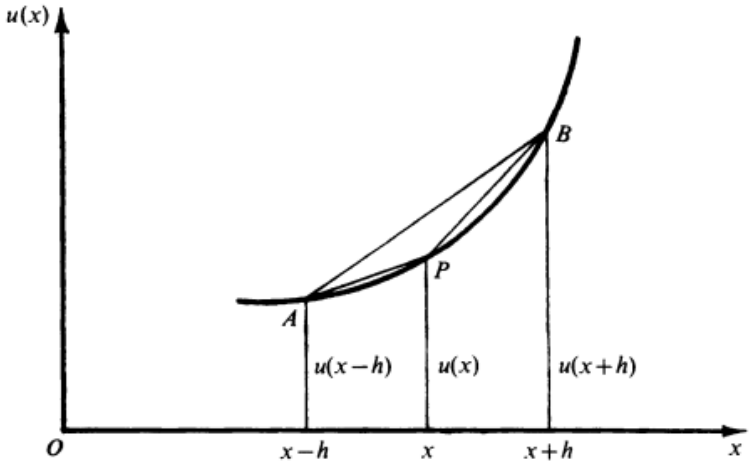


Figure 1.2: Approximations of the slope of the tangent at point P by the slope of the chord AB, and by the slope of the chord PB.

Notation for functions of several variables

Assume U is a function of the independent variables x and t. Subdivide the $x-t$ plane into sets of equal rectangles of sides $\Delta x = h, \Delta t = k$, by equally spaced grid lines parallel to Oy, defined by $x_i = ih, i = 0, \pm 1, \pm 2, \ldots$, and equally spaced grid lines parallel to Ox, defined by $t_j = jk, j = 0, 1, 2, \ldots$, as shown in figure 1.3. Denote the value of U at the representative mesh point $P(ih, jk)$ by

$$U_{i,j} = U(ih, jk).$$

Then by equation(3.4),

$$\left(\frac{\partial^2 U}{\partial x^2}\right)_{i,j} \approx \frac{U\{(i+1)h, jk\} - 2U\{ih, jk\} + U\{(i-1)h, jk\}}{h^2}.$$

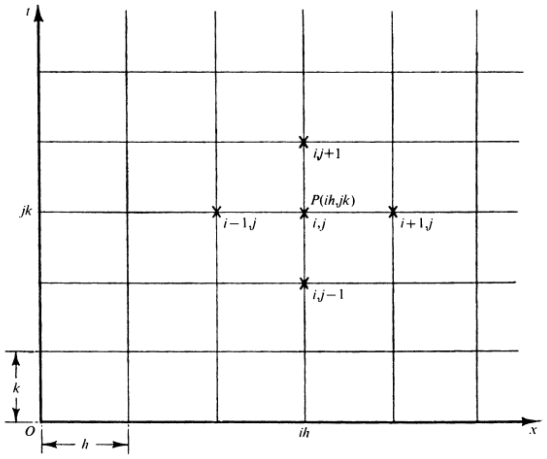


Figure 1.3: Approximated value of U at the representative mesh point P.

i.e.

$$\left(\frac{\partial^2 U}{\partial x^2}\right)_{i,j} \approx \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h^2},$$

with a leading error of order h^2.

With this notation the forward (resp. backward) difference approximation for $\partial U/\partial x$ at P is

$$\left(\frac{\partial U}{\partial x}\right)^{+}_{i,j} \approx \frac{U_{i+1,j} - U_{i,j}}{h}, \left(\frac{\partial U}{\partial x}\right)^{-}_{i,j} \approx \frac{U_{i,j} - U_{i-1,j}}{h},$$

with a leading error of $O(h)$.
The forward-difference approximation for $\partial U/\partial t$ at P is

$$\frac{\partial U}{\partial t} \approx \frac{U_{i,j+1} - U_{i,j}}{k},$$

with a leading error of $O(k)$.

CHAPTER 2

DIFFUSION FOR IMAGE PROCESSING

The diffusion partial differential equations are one of the most important mathematical tools in image processing. In this chapter, we will discuss isotropic and anisotropic diffusion models by addressing the numerical solutions of this models. Numerical simulations will be presented in order to ilustrate thier impact in the field of image processing.

2.1 Isotropic Diffusion

The first PDE that have been used in image processing is the Heat equation that realizes a spatial diffusion of the gray values of a given image. It is a parabolic equation that reads, see [20]. For any bounded function $f \in \mathbb{C}^2$ and any fixed value $T > 0$, the heat equation can be expressed as follows:

$$\begin{cases} \frac{\partial u(t,x)}{\partial t} & = \Delta u(t,x) \qquad \text{on } \mathbb{R}^2 \times (0,T] \\ u(0,x) & = f(x) \qquad \text{for } \mathbb{R}^2 \end{cases} \tag{2.1}$$

where Δ is the Laplacian operator.

2.1.1 Discretization of 2D heat equation

The explicite scheme in 2D

In the case of two dimensions the explicite scheme:

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

After simplification:

$$u_{i,j}^{n+1} = \frac{\Delta t}{\Delta x^2}(u_{i+1,j}^n + u_{i-1,j}^n) + \frac{\Delta t}{\Delta y^2}(u_{i,j+1}^n + u_{i,j-1}^n) + (1 - 2\frac{\Delta t}{\Delta x^2} - 2\frac{\Delta t}{\Delta y^2})u_{i,j}^n$$

or write $\alpha = \frac{\Delta t}{\Delta x^2}$ and $\beta = \frac{\Delta t}{\Delta y^2}$

$$u_{i,j}^{n+1} = \alpha(u_{i+1,j}^n + u_{i-1,j}^n) + \beta(u_{i,j+1}^n + u_{i,j-1}^n) + (1 - 2\alpha - 2\beta)u_{i,j}^n$$

In this case the stability condition: $\alpha + \beta \leq \frac{1}{2}$ this stability condition a limite on the time step:

$$\Delta t \leq \frac{\Delta x^2 \Delta y^2}{2(\Delta x^2 + \Delta y^2)})$$

The implicite scheme in 2D

In the case of two dimensions the implicite scheme:

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

After simplification we get:

$$u_{i,j}^{n+1} = -\frac{\Delta t}{\Delta x^2}(u_{i+1,j}^n + u_{i-1,j}^n) - \frac{\Delta t}{\Delta y^2}(u_{i,j+1}^n + u_{i,j-1}^n) + (1 + 2\frac{\Delta t}{\Delta x^2} + 2\frac{\Delta t}{\Delta y^2})u_{i,j}^n$$

or write $\alpha = \frac{\Delta t}{\Delta x^2}$ and $\beta = \frac{\Delta t}{\Delta y^2}$

$$u_{i,j}^{n+1} = -\alpha(u_{i+1,j}^n + u_{i-1,j}^n) - \beta(u_{i,j+1}^n + u_{i,j-1}^n) + (1 + 2\alpha + 2\beta)u_{i,j}^n$$

this scheme is unconditionally stable.

2.1.2 Numerical Experiments

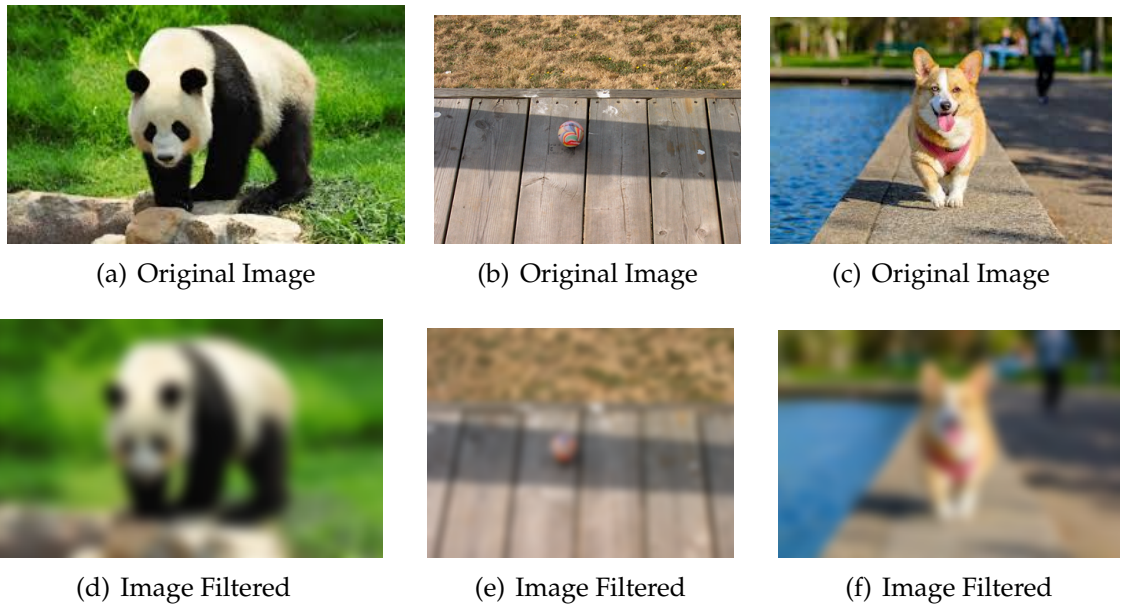
The isotropic diffusion filter has its limitations: whether we smooth uniformly by a rotational symmetric Gaussian kernel, or diffuse the data equally in all directions, the process deforms important features of the image, blurs and dislocates edges. (See Figure 2.1).

2.2 Anisotropic Diffusion

2.2.1 1D Perona-Malik equation

Let's consider a continuous-scale 1-D signal $u(x,t)$, where x and t represent space and time, respectively. The diffusion partial differential equation (PDE) with a positive diffusivity function $g(x,t)$ can be written in $1D$ as follows:

$$\partial_t u = \partial_x(g\partial_x u) \tag{2.2}$$



(a) Original Image (b) Original Image (c) Original Image

(d) Image Filtered (e) Image Filtered (f) Image Filtered

Figure 2.1: Images filtered by isotropic diffusion

The diffusion equation should be supplemented with an initial condition $u(x,0) = f(x)$, which represents the initial state of the signal at time $t = 0$. Additionally, if the domain is bounded, boundary conditions should also be specified to determine the behavior of the signal at the boundaries of the domain.
In a discrete setting, we use a spatial mesh width Δx and define the pixel location x_i as $x_i = (i - 1/2)\Delta x$ for $i \in 1, \ldots, N$. Similarly, we introduce a time discretization $t_n = n\Delta t$, where n represents the time step index and Δt is the time step size. This allows us to obtain a discrete signal $u_{ni} \approx u(x_i, t_n)$. Then, a standard finite difference discretization of the diffusion equation can be achieved using the explicit scheme.

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{1}{\Delta x}\left(g_{i+1/2}^n \frac{u_{i+1}^n - u_i^n}{\Delta x} - g_{i-1/2}^n \frac{u_i^n - u_{i-1}^n}{\Delta x}\right) \tag{2.3}$$

Where $g_{i+1/2}^n$ denotes the diffusivity between the computational cells i and $i+1$. Using the mesh ratio $r = \frac{\Delta t}{(\Delta x)^2}$, our scheme can be rewritten as:

$$u_i^{n+1} = u_i^n + r g_{i+1/2}^n u_i^n - r g_{i-1/2}^n u_i^n - r g_{i-1/2}^n u_{i+1}^n - r g_{i-1/2}^n u_{i-1}^n \tag{2.4}$$

This equation represents the update step for computing the value of the signal at the next time step $n+1$ based on the current values at time step n.
It is convenient to express this as a matrix-vector multiplication in the form $u^{n+1} = Q^n u^n$, where Q^n is an $(N \times N)$ matrix with entries

$$q^n i, j = \begin{cases} 1 - r g^n i - 1/2 - r g^n i + 1/2 & \text{if } j = i \\ r g_{i-1/2}^n & \text{if } j = i-1 \\ r g_{i+1/2}^n & \text{if } j = i+1 \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

Let's briefly review some important properties of the matrix Q^n. It is evident that the matrix is symmetric. The stability of the iterative scheme 2.4 can be demonstrated if the entries of Q^n are nonnegative. Since the diffusivity is positive, all off-diagonal entries contain nonnegative values, except for the diagonal entries, which require further clarification. Therefore, it is necessary to ensure that all diagonal entries satisfy the following condition:

$$q^n_{i,i} = 1 - rg^n i - 1/2 - rg^n i + 1/2 \geq 0. \tag{2.6}$$

This condition implies a stability requirement on the size of the time step Δt.
To implement homogeneous Neumann boundary conditions $(\partial_x u = 0)$, we need to modify the entries for $q^n_{1,1}$ and $q^n_{N,N}$ in the following way:

$$q^n_{1,1} = 1 - rg^n_{3/2} \text{ and } q^n_{N,N} = 1 - rg^n_{N-1/2} \tag{2.7}$$

This interpretation involves setting the missing terms $g^n_{1/2}$ and $g^n_{N+1/2}$ to 0, which effectively implements homogeneous Neumann boundary conditions. It is worth mentioning that it is also possible to implement other boundary conditions such as Dirichlet boundary conditions or periodic boundary conditions. Diffusion can

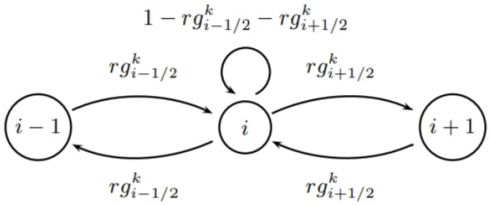


Figure 2.2: The diffusion process can be visualized using a Markov chain model.

also be represented using Markov chains, which utilize stochastic matrices incorporating transition probabilities. A stochastic matrix consists of nonnegative entries and has unit column sums. By considering the positive diffusivity and selecting a mesh ratio r that satisfies equation 2.6 for all i, we can ensure that the matrix Q^n only contains nonnegative entries. Furthermore, all column sums are equal to 1. Hence, Q^n is a stochastic matrix, and the entries $q^n_{i,j} \geq 0$ can be interpreted as transition probabilities. In the context of Markov chains, it is convenient to employ a graph-based representation of the diffusion model, as illustrated in Figure 2.2.

2.2.2 2D Perona-Malik equation

The two-dimensional Perona-Malik equation is written as follows:[12]

$$\begin{cases} u_t = \text{ div } (g\,(|\nabla u|)\nabla u) & \text{in } \Omega \times (0, +\infty), \\ \dfrac{\partial u}{\partial n} = 0 & \text{in } \partial\Omega \times (0, +\infty), \\ u(x, 0) = u_0(x) & \text{in } \Omega. \end{cases} \tag{2.8}$$

Discretised Diffusion Equation

In order to solve Equation 2.8 of Implicit Scheme we need to discretise the divergence operator. We start by marking the positions of the pixels of interest with (i, j) :

$$\frac{(I_k)_{i,j}^{n+1} - (I_k)_{i,j}^{n}}{\Delta t} = \text{div}\left(g^n \nabla I_k^{n+1}\right)$$

After this we discretise the div operator:

$$\begin{aligned}(I_k)_{i,j}^{n+1} - (I_k)_{i,j}^{n} =& \Delta t g_N^n \left((I_k)_{i-1,j}^{n+1} - (I_k)_{i,j}^{n+1}\right) \\ &+ \Delta t g_S^n \left((I_k)_{i+1,j}^{n+1} - (I_k)_{i,j}^{n+1}\right) \\ &+ \Delta t g_W^n \left((I_k)_{i,j-1}^{n+1} - (I_k)_{i,j}^{n+1}\right) \\ &+ \Delta t g_E^n \left((I_k)_{i,j+1}^{n+1} - (I_k)_{i,j}^{n+1}\right)\end{aligned}$$

2.2.3 AOS (Additive Operator Splitting)

With the vector/matrix format in place, we can now formulate the 'additive operator splitting' scheme proposed by Weickert et al [19]. In order to simplify the notation, we write A instead of $A\left((\mathbf{I}_k)^n\right)$. Id refers to the identity matrix. Therefore, we have:

$$(\mathbf{I}_k)^{n+1} = (\mathbf{I}_k)^n + \Delta t A \mathbf{I}_k^{n+1}$$

From which $(\mathbf{I}_k)^{n+1}$ can be solved as follows:

$$(\mathbf{I}_k)^{n+1} = (Id - \Delta t A)^{-1} \mathbf{I}_k^n$$

Now, we 'decompose' A so that $A = \sum_{l=1}^{m} A_l$, which allows as to write the above equation as:

$$(\mathbf{I}_k)^{n+1} = \left(\sum_{l=1}^{m} \frac{1}{m} Id - \Delta t \sum_{l=1}^{m} A_l\right)^{-1} \mathbf{I}_k^n$$

where m is the number of dimensions (in our case $m = 2$). Previous equation can be written, using only a single summation operator, as:

$$(\mathbf{I}_k)^{n+1} = \left(\sum_{l=1}^{m} \frac{1}{m} \left(Id - \Delta t m A_l\right)\right)^{-1} \mathbf{I}_k^n \tag{2.9}$$

Equation 2.9 has interesting 'form' in the sense that the 'system matrix' is decomposed. The problem is that the decomposed system matrix is inside the $()^{-1}$ operator. Instead, we would like to construct the solution in parts as follows:

$$(\mathbf{I}_k)^{n+1} = \sum_{l=1}^{m} \left(\frac{1}{m} \left(Id - \Delta t m A_l\right)\right)^{-1} \mathbf{I}_k^n \tag{2.10}$$

The problem is that the right hand sides of Equations 2.9 and 2.10 are not equal, as can be easily verified. Therefore, we pose the question if there exists a simple variable x, when used to multiply the right hand side of 2.10, would make these equal:

$$(\sum_{l=1}^{m} \frac{1}{m} \underbrace{(Id - \Delta t m A_l)}_{B})^{-1} \mathbf{I}_k^n = x \sum_{l=1}^{m} (\frac{1}{m} \underbrace{(Id - \Delta t m A_l)}_{B})^{-1} \mathbf{I}_k^n \tag{2.11}$$

The above can be simplified into:

$$B^{-1} = x m^2 B^{-1}$$

And, thus we have:

$$x = \frac{1}{m^2}$$

Based on this, in order to use the 'additive operator splitting' scheme given by Equation 2.10., we multiply the right hand side with $\frac{1}{m^2}$, and we obtain the following equation:

$$(\mathbf{I}_k)^{n+1} = \frac{1}{m^2} \sum_{l=1}^{m} \left(\frac{1}{m} (Id - \Delta t m A_l) \right)^{-1} \mathbf{I}_k^n$$

which is the same as:

$$(\mathbf{I}_k)^{n+1} = \sum_{l=1}^{m} \left(m Id - \Delta t m^2 A_l \right)^{-1} \mathbf{I}_k^n$$

As an example, if $l = 2(2\mathrm{D})$, then we would have:

$$(\mathbf{I}_k)^{n+1} = (2Id - 4\Delta t A_1)^{-1} \mathbf{I}_k^n + (2Id - 4\Delta t A_2)^{-1} \mathbf{I}_k^n$$

Introducing the notations $V = (2Id + 4\Delta t A_x)\, u^n$ and $W = (2Id + 4\Delta t A_y)\, u^n$ the solution is simply

$$u^{n+1} = V + W$$

Now, as it can be understood, the whole idea of this scheme is to bring the equations to a 'simpler' form, allowing us to use efficient block-wise solvers.

2.2.4 Numerical Experiments

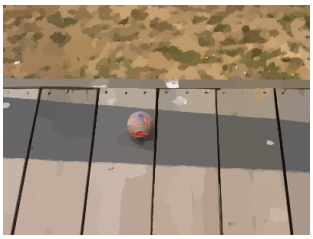
As seen in Figure 2.4, compared to the isotropic diffusion filtering which blurs the edges, the anisotropic diffusion filtering preserves edges sharpness.

(a) Original Image

(b) Original Image

(c) Original Image

(d) Image Filtered after 200 Iteration



(e) Image Filtered after 200 Iteration

(f) Image Filtered after 200 Iteration

Figure 2.3: Images filtered by anisotropic diffusion

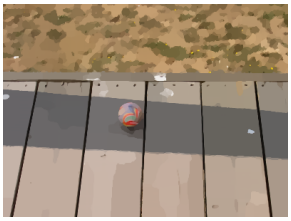
(a) Original Image

(b) Image Filtered by isotropic diffusion

(c) Image Filtered by anisotropic diffusion

(d) Original Image

(e) Image Filtered by isotropic diffusion



(f) Image Filtered by anisotropic diffusion

Figure 2.4: Images filtered by isotropic and anisotropic diffusion

CHAPTER 3

LINEAR OSMOSIS MODELS FOR VISUAL COMPUTING

In this chapter, our objective is to extend a linear parabolic partial differential equation (PDE) introduced in the previous chapter. The steady state of this equation proves beneficial for addressing various imaging problems, hence earning the name "osmosis filter" due to its resemblance to physical processes.

3.1 Linear Osmosis Filter in 1D

Osmosis can be considered as a generalization of diffusion filters. In osmosis, we introduce semi-permeable membranes between adjacent pixels, allowing for selective transport of particles with different transition probabilities depending on the orientation. This means that the transition probability from pixel i to pixel $i+1$ may differ from the probability in the reverse direction.
In the Markov model, this leads to a loss of symmetry in the graph represented in Figure 2.2. This is achieved by introducing different diffusivities, known as osmoticities, in different orientations. The forward osmoticity from pixel i to $i+1$ at time level n is denoted by $g^{+,n}_{i+1/2}$, while the backward osmoticity from pixel $i+1$ to i is denoted by $g^{-,n}_{i+1/2}$. These osmoticities are chosen such that a normalization condition is satisfied.[18]

$$g^{+,n}_{i+1/2} + g^{-,n}_{i+1/2} = 2 \tag{3.1}$$

Figure 3.1 illustrates a graph-based representation of the osmosis process. This

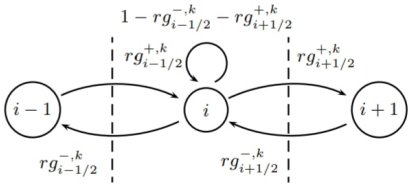


Figure 3.1: Modeling the Osmosis Process using a Markov Chain Approach

schematic diagram depicts the newly introduced process.

$$u_i^{n+1} = u_i^n \underbrace{-rg^{+,n}_{i+1/2}u_i^n - rg^{-,n}_{i-1/2}u_i^n}_{"outflow"} \underbrace{+rg^{-,n}_{i-1/2}u_{i+1}^n + rg^{+,n}_{i-1/2}u_{i-1}^n}_{"inflow"} \tag{3.2}$$

The scheme 3.2, along with the normalization condition 3.1, provides an approximation of the 1-D osmosis partial differential equation (PDE) on a fixed, given mesh with a size of Δt.

$$\partial_t u + \partial_x \left(\frac{g^+ - g^-}{\Delta x} u \right) = \partial_{xx} u \tag{3.3}$$

The equation represents an advection-diffusion equation or drift-diffusion equation, where g^+ and g^- are continuous-scale representations of the osmoticities.
Extending osmosis to higher dimensions and color images is a straight-forward process. Furthermore, osmosis has proven to be a versatile framework for addressing various visual computing problems, including but not limited to clustering, data integration, focus fusion, exposure blending, image editing, shadow removal, and compact image representation.

3.2 2D Linear Osmosis Filter

The researchers, Weickert et al; extensively investigated the traditional theory of linear image osmosis in both continuous and discrete environments. Their study, documented in references [17, **?**], delved deeply into the complexities of this theory, offering a comprehensive comprehension of its principles and practical implementations.
Let $\Omega \subset \mathbb{R}$ be a bounded image domain with Lipschitz boundary $\partial\Omega$, let $u, v, f : \Omega \longrightarrow \mathbb{R}^2$ be positive grey-scale images and $d : \Omega \to \mathbb{R}_+$ be a given vector field.

Definition 3.2.1. (Osmosis energy [18])
The osmosis energy of u with respect to a reference v is defined as:

$$E^v(u) = \int_\Omega v \left| \Delta(\frac{u}{v}) \right|^2 dx \tag{3.4}$$

In the following we consider the relation between the minimisation of the osmosis energy 3.4 and the steady-state solution of the following drift-diffusion

initial value PDE problem, which computes for every $t \in (0,T]$ and $T > 0$ a regularised family of images $\{u(x,t)\}_{t>0}$, starting from f:

$$\begin{cases} \partial u = \Delta u - div(du) \quad \text{on } \Omega \times (0,T] \\ u(0,x) = f(x) \quad \text{on } \Omega \\ \langle \nabla u - du, n \rangle = 0 \text{ on } \partial\Omega \times (0,T] \end{cases} \tag{3.5}$$

where $\langle ., . \rangle$ denotes the Euclidean scalar and n the outer normal vector on $\partial\Omega$.

3.2.1 Continuous theory

In [18], it is proven that solutions to equation 3.5, for any positive time t, maintain the average gray value and non-negativity. Moreover, when the vector field d is defined as $d = \nabla \log v$, with v being a strictly positive reference image, the steady-state solution w of equation 3.5 minimizes a quadratic energy functional and can be represented as a scaled version of v.

Theorem 3.2.2. *[18] The solution of the osmosis model 3.5 enjoys the following properties:* • ***Conservation of the Average Grey Value (AVG):***

$$\frac{1}{|\Omega|}\int_\Omega u(x,t)dx = \frac{1}{|\Omega|}dx, \textit{for all} t > 0$$

• ***Conservation of positivity:***

$$u(x,t) > 0, \textit{for all} x \in \Omega \textit{and} t > 0$$

• ***non-constant steady-states:***
given a stictly positive refernce image v and a comapatible drift $d = \nabla log v$ energy 3.4 is minimised by the steady-states of 3.5 . Furthermore,the stationary solution of (3.2) is given by $w(x) = \frac{\mu_f}{\mu_v} v(x)$,where μ_f and μ_v denote the averge grey value of f and v respectively.

The theorem ensures that the solution to equation 3.5 exhibits a conservation property, and the stationary elliptic problem can be analytically solved using the initial image, v. This distinguishes it from diffusion models, where the steady-states are constant. The non-constant nature of the steady-states in the osmosis model adds an intriguing aspect to the study of convergence, especially in imaging applications, as it represents a non-symmetric variation of the diffusion process.

Proof. [18] • Let $\mu(t) = \frac{1}{|\Omega|}\int_\Omega u(x,t)dx$ denote the average grey value at time $t \geq 0$. Using the divergence theorem and the homogeneous Neumann boundary condition, we obtain:

$$\frac{\partial \mu}{\partial t} = \frac{1}{|\Omega|}\int_\Omega \partial_t u dx = \frac{1}{|\Omega|}\int_\Omega div(\nabla u - du)dx = \int_{\partial\Omega} \langle \nabla u - du, n \rangle dS = 0$$

Thus, the average grey value remains constant over time.
• Assume that $T > 0$ is the smallest time where $\min_{x,t} u(x,t) = 0$, and that this minimum is attained at some inner point ξ. Thus, $\nabla u(\xi, T) = 0$, and we have:

$$\partial_t u(\xi, T) = \Delta u(\xi, T) - \underbrace{u(\xi, T)}_{=0} div d - d^T \underbrace{\nabla u(\xi, T)}_{=0}.$$

This shows that at $(\xi, T)^T$, the osmosis evolution behaves like the diffusion equation $\partial_t u = \Delta u$. It is well-known that for diffusion with homogeneous Neumann boundary conditions, the minimum cannot decrease over time. Thus, the solution of the osmosis process remains positive.
• The energy functional 3.4 can be rewritten as

$$E(u) = \int_\Omega F(u, \nabla u) dx$$

with

$$F(u, \nabla u) = \frac{|v\nabla u - u\nabla v|^2}{v^3}. \tag{3.6}$$

From the calculus of variation, we know that any minimiser of $E(u)$ satisfies the Euler-Lagrange equation:

$$0 = F_u - \partial_x F_{u_x} - \partial_y F_{u_y}$$

with homogeneous Neumann boundary conditions, where $x = (x, y)^T$ and subscripts denote partial derivatives. With F from 3.5, this becomes after some simplifications:

$$0 = -2v div\left(\frac{v\nabla u - u\nabla v}{v^3}\right) - \frac{4\nabla^T v(v\nabla u - u\nabla v)}{v^3}.$$

Using

$$div(v\nabla(\frac{u}{v})) = v^2 div\left(\frac{v\nabla u - u\nabla v}{v^3}\right) + \frac{2\nabla^T v(v\nabla u - u\nabla v)}{v^2}$$

the Euler-lagrange equation can be written as

$$0 = -\frac{2}{v} div(v\nabla(\frac{u}{v}).$$

This is equivalent to $\Delta u - \text{div}(du) = 0$ when $d = \frac{\nabla v}{v}$ with $v > 0$, as can be easily verified. Additionally, it can be shown through straightforward computations that the boundary condition on $\partial\Omega \times (0, T]$ is $< \nabla u - du, n >= 0$.
It is worth noting that an image v satisfying $d = \frac{\nabla v}{v}$ also satisfies the steady-state

equation $\Delta u - \text{div}(du) = 0$ with homogeneous Neumann boundary conditions. However, it is important to recognize that cv, where c is any constant, is also a solution to this problem. Since the osmosis evolution preserves the average gray value and the positivity of the initial image, it can only converge to a rescaled version w of v that is positive and has the same average gray value as the original image f. Thus, we have $w(x) = \frac{\mu_f}{\mu_v} v(x)$.

□

3.2.2 Discrete scale-space theory

In this section, we explore discrete numerical scale-space theory, drawing inspiration from [20]. To transition from a continuous to a discrete framework, we consider an image domain Ω, represented as a rectangular grid with dimensions $M \times N$, containing $S = MN$ pixels. The initial image f is a positive element in $\mathbb{R}_+^S$. By using a grid step size of $\Delta x > 0$, we approximate the value of the function u at discretization nodes $x_i = (x_i, y_j)$ using $u_{i,j}$. Similarly, for $n \geq 0$, we denote the value of $u_{i,j}$ at the time discretization node t_n as $u_{i,j}^n$.

In [17, 18] the finite-difference spatial discretization matrix A of Equation 3.5 acts on u in the following way:

$$\begin{aligned} u_{i,j}^{'} = &\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\Delta x)^2} - \frac{1}{\Delta x}\left(d_{1,i+\frac{1}{2},j}\frac{u_{i+1,j}+u_{i,j}}{2} - d_{1,i-\frac{1}{2},j}\frac{u_{i,j}+u_{i-1,j}}{2}\right) \\ &+\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\Delta x)^2} - \frac{1}{\Delta x}\left(d_{2,i,j+\frac{1}{2}}\frac{u_{i,j+1}+u_{i,j}}{2} - d_{2,i,j-\frac{1}{2}}\frac{u_{i,j}+u_{i-1,j}}{2}\right) \end{aligned} \tag{3.7}$$

Here $u_{i,j}$ stands for an approximation of the function in the point $(x_i, y_j) = ((i - 1/2)\Delta x, (j - \frac{1}{2})\Delta x)$ while the approximated quantities correspondent to the discrete field $d = (d_1, d_2)$ where $|d_1|, |d_2| \leq 2(\Delta x)^{-1}$,are defined as:

$$d_{1,i+\frac{1}{2},j} = \frac{2(f_{i+1,j} - f_{i,j})}{\Delta x(f_{i+1,j} + f_{i,j})},\ d_{2,i,j+\frac{1}{2}} = \frac{2(f_{i,j+1} - f_{i,j})}{\Delta x(f_{i,j+1} + f_{i,j})}. \tag{3.8}$$

It is important to note that the discretization implicitly assumes the use of two different grids in order to model the "flux exchange" of d at the interfaces between pixel cells.

Theorem 3.2.3. *[18] For a given $f \in \mathbb{R}_+^S$,consider the fully-discretised problem:*

$$u^0,\ u^{n+1} = Pu^n,\ n \geq 1,$$

where the (non-symmetric)matrix $P \in \mathbb{R}^{S\times S}$ is irreducible and non-negative with strictly positive diagonale entries and unit column sum.Then the following properties hold true:
1. *The evolution preserves positivity and the average grey value of f.*
2. *The eigenvector of Passociated to eigenvalue 1 is the unique steady-state for $n \to \infty$.*

Theorem 3.2.4. *[18] For a given $f \in \mathbb{R}_+^S$,consider the semi-discretised linear osmosis problem:*

$$u(0) = f,\ u^{'}(t) = Au(t),\ t > 0 \tag{3.9}$$

where $A \in \mathbb{R}^{S\times S}$ is an irreducible (non-symmetric) matrix with zero-column sum and non-negative off-diagonal entries.For $n \geq 0$,consider the time-discretisation schemes $u^{n+1} = Pu^n$:
Forward Euler:

$$P = (I + \Delta tA),\ for,\ \Delta t < (max|a_{i,i}|)^{-1} \tag{F.E}$$

Backward Euler:

$$P = (I + \Delta tA),\ forany\ \Delta t > 0 \tag{B.E}$$

Then,in both cases,P is an irreducible,non-negative matrix with strictly diagonal entries and unit column sum and for $u(t)$ the Theorem 3.2.3 hold true.

When $\Delta t > 0$ is large, the numerical implementation using the Backward Euler (B.E) method requires the inversion of a non-symmetric tridiagonal matrix. This operation can be computationally expensive, particularly for large images. In[17, **?**], the authors utilize the Backward Euler (B.E) method to solve equation 3.7. This method is advantageous as it does not have any timestep restrictions. The BiCGStab iterative solver is employed to perform the solution process.

3.2.3 Linear osmosis filter for shadow removal problem

The problem of eliminating shadows from an image $f : \Omega \to \mathbb{R}_+$, while maintaining the original image's geometry and texture, is commonly known as shadow removal. In this task, it is assumed that there are consistent shadows, where the relationship between the image intensity values inside and outside the shadow region remains constant, albeit with an unknown multiplicative constant.
To formulate the problem of shadow removal, let us begin with a concise approach. Consider a positive image $f : \Omega \to \mathbb{R}_+$ that has been affected by a constant shadow. In specific areas of the image, f has undergone rescaling by a constant factor $0 < c < 1$. The objective of shadow removal is to eliminate the shadow while retaining the original geometric and textural details of the image. We can assume that the image domain Ω can be divided into distinct regions, some containing shadows and others that do not.

$$\Omega = \Omega_{out} \cup \Omega_{sb} \cup \Omega_{in} \tag{3.10}$$

where $\Omega_{out}, \Omega_{sb}$ and Ω_{in} are the unshadowed region,the shadow boundaries and the shadowed region of the image,respectively. see Figure 3.2
The task of shadow removal can be viewed as an image reconstruction problem encountered in various practical domains. It entails restoring the correct light intensity in all regions of the image domain Ω. In our exploration, we will delve into this subject in greater detail and examine its significance in relation to other problems in the preservation of cultural heritage.
It is worth mentioning that the shadow removal problem tackled using osmosis

can be formulated as an image inpainting problem. In this context, information is propagated from the outer region Ω_{out} to the inner region Ω_{in} through the shadow boundary Ω_{sb}. To provide clarity, we will employ the following notation to elucidate this statement.

(a) f on Ω (b) f on Ω_{out} (c) f on Ω_{sb} (d) f on Ω_{in}

Figure 3.2: Figures Decomposition of Ω as in 3.10

As previously mentioned, we will introduce the inpainting region as $O = \Omega_{\text{sb}}$, implying that $O^c = \Omega_{\text{out}} \cup \Omega_{\text{in}}$. To establish a more precise connection, let us delve into the details.
Assuming a decomposition, as described in 3.10, is provided (which can be challenging for real images [6]), the osmosis model 3.5 can be straightforwardly modified to address the shadow removal problem. This modification involves defining the vector field d in 3.5 based on the shadowed image f as follows: $d = \nabla \log f$ on $\Omega_{\text{in}} \cup \Omega_{\text{out}}$, and $d = 0$ on Ω_{sb}. Further information regarding this choice can be found in [**?**]. The adapted continuous osmosis model for shadow removal, with $d = \nabla \log f$, is then expressed as follows:

$$\begin{cases} u_t = \Delta u - div(du) \text{ on } \Omega_{in} \cup \Omega_{out} \times (0, T] \\ u_t = \Delta u \text{ on } \Omega_{sb} \times (0, T] \\ u(x, 0) = f(x) \text{ on } \Omega \\ \langle \nabla u - du, n \rangle = 0 \text{ on } \partial\Omega \times (0, T] \end{cases} \tag{3.11}$$

The evolution process at the shadow boundary Ω_{sb} can be interpreted as an inpainting technique, where information is propagated from the outer region Ω_{out} to the inner region Ω_{in} across the shadow boundary Ω_{sb}. The Laplace operator's effect on Ω_{sb} results in isotropic diffusion of image structures from both Ω_{in} and Ω_{out} onto Ω_{sb}, yielding an inpainted result that is free from shadows but may exhibit some blurriness. To mitigate this issue, a post-processing step is commonly employed to refine the inpainted result.

3.3 Splitting methods for linear osmosis filter

Our aim is to improve the computational efficiency of the numerical solution for the discrete linear osmosis model 3.6, particularly for large images. To achieve this, we are exploring different splitting schemes as potential approaches to accelerate the calculation process.

During the 1950s and 1960s, significant progress was made in the development of various splitting methods. These methods included fractional steps or locally one-dimensional methods (LOD) in the Soviet Union, as well as dimensional splitting Alternating Direction Implicit (ADI) methods in the USA. Notable references in this field include [14, 8, 7]. ADI methods have advantages over LOD methods, such as easier application of boundary conditions and the ability to obtain a steady-state solution independent of the time step Δt [21]. For more detailed information, please refer to [25, 26, 27, 28, 29]. In recent years, there have been advancements in unconditionally stable operator splitting schemes for diffusion processes in image processing, as discussed in [20, 4].
In the next,we will consider:
• ADI methods:Peaceman-Rachford (RP) and Douglas (D);
• operator splitting methods:addiitive and multiplicative (AOS,MOS,AMOS).
We assess the effectiveness of these splitting schemes in achieving convergence towards a scaled version of a reference image v, starting from a flat image f, under both the compatible case, where v is encoded as $d = \nabla \log v$, and the non-compatible case, where d is random.

Towards splitting schemes

First and foremost, it is important to note that Theorem 3.2.3 is of significant generality as it allows us to investigate numerous alternative operators, denoted as P, capable of computing solutions for the discretized osmosis problem. These solutions possess similar properties to those derived from continuum models.
As an example, there exists a more comprehensive approach to time discretization called the θ-method, which combines both the Forward Euler (F.E.) and Backward Euler (B.E.) methods. Using a spatial discretization A for the initial value u^0 of Equation 3.5, the θ-method calculates an approximation u^{n+1} for a fixed $\theta \in [0, 1]$, any time step $\Delta t > 0$, and $n \geq 0$, using the following update formula. We start with $u^0 = f$:

$$(I - \Delta t\theta A)u^{n+1} = (I + \Delta t(1 - \theta)A)u^n \qquad (\theta\text{-mth})$$

It is important to highlight that the Forward Euler (F.E.) and Backward Euler (B.E.) methods are specific instances of the θ-method, where θ takes values of 0 and 1, respectively. When θ is set to 0.5, the θ-method becomes a second-order scheme, whereas for any other value of θ, it is a first-order scheme. It is crucial to note that explicit schemes have limitations on the time step size (Δt) to maintain numerical stability. Moreover, for θ values greater than 0, the implementation of the numerical scheme requires inverting a non-symmetric tridiagonal matrix, which can be computationally expensive for large images, even with pre conditioners.
To reduce the computational cost of the semi-discretized problem 3.9, one strategy is to decompose the matrix A into two separate sums, A_1 and A_2, which represent the contributions along the x and y spatial directions, respectively. As a result, the solution to the semi-discretized problem at $t > 0$ can be obtained as

follows:

$$u(t) = exp(t(A_1 + A_2))f$$

In general,A_1 and A_2 do not commute ,therefore the following approximations hold

$u(t) \approx exp(tA_1)exp(tA_2)f$ (Beam-Warming)
$u(t) \approx exp(\frac{t}{2}A_1)exp(tA_2)exp(\frac{t}{2A_1})f,$ (Strang)
$u(t) \approx \frac{1}{2}exp(tA_1)exp(tA_2)f + \frac{1}{2}exp(tA_1)exp(tA_2)f$ (parallel)

The Beam-Warming splitting method is a first-order technique, whereas the Strang and parallel splittings are second-order methods. A splitting scheme is deemed stable if all coefficients in front of the exponentials (A_1 and A_2) are non-negative. It is worth mentioning that, according to [15], there are no stable splitting schemes of order higher than two for linear equations. However, non-stable third-order schemes have been proposed in [**?**] specifically for hyperbolic equations.

3.3.1 ADI methods:Peaceman-Rachford and Douglas

The ADI (Alternating Direction Implicit) splitting method is a time-stepping method used to solve the initial boundary value problem in a domain $\Omega \in \mathbb{R}^S$. In this method, the semi-discretized operator A is decomposed into a sum:

$$A = A_0 + A_1 + \ldots + A_S$$

Each component A_j, where $j = 1, \ldots, S$, represents the linear action of A along the space direction j. The term A_0 may include additional contributions from mixed directions and non-stiff nonlinear terms.

The ADI scheme treats the unidirectional components A_j, for $j \geq 1$, implicitly in time, while the A_0 component, if present, is treated explicitly. This means that the implicit part is solved using an implicit numerical method, such as an implicit finite difference scheme, while the explicit part is solved using an explicit method.

By using the ADI splitting method, the s-dimensional original problem is reduced to s one-dimensional problems. This simplifies the computational complexity of solving the problem by solving each dimension separately.

Please note that the specific details of the numerical methods used to solve each component may vary depending on the problem and the chosen discretization scheme.

For imaging applications, let's consider the case where $S = 2$. We can denote the approximation of u as $u_{i,j}$, which represents the value of u at the grid point $((i - 1/2)\Delta x, (j - 1/2)\Delta x)$, where h is the grid size. By applying the finite difference discretization to Equation 3.7, the splitting $Au = A_1u + A_2u$ simplifies the two-dimensional problem into two separate one-dimensional problems, which can be expressed as follows:

$$\begin{aligned}(A_1(u))_{i,j} &= \frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\Delta x)^2} - \frac{1}{\Delta x}(d_{1,i+\frac{1}{2},j}\frac{u_{i+1,j}+u_{i,j}}{2} - d_{1,i-\frac{1}{2},j}\frac{u_{i,j}+u_{i-1,j}}{2});\\ (A_2(u))_{i,j} &= \frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\Delta x)^2} - \frac{1}{\Delta x}(d_{2,i,j+\frac{1}{2}}\frac{u_{i,j+1}+u_{i,j}}{2} - d_{2,i,j-\frac{1}{2}}\frac{u_{i,j}+u_{i,j-1}}{2})\end{aligned} \quad (3.12)$$

Subsequently, we will explore two widely recognized ADI (Alternating Direction Implicit) schemes that exhibit discrete-scale space similarities to the continuum case, in relation to Theorem 3.2.3.

The Peaceman–Rachford scheme

To start our analysis, let us investigate the *Peaceman-Rachford* ADI scheme [14], which achieves second-order accuracy. Given an initial value u^0, and for each $n \geq 0$ and time step $\Delta t > 0$, we can compute an approximation u^{n+1} using the following update rule:

$$\begin{cases} u^{n+1/2} = u^n + \frac{\Delta t}{2} A_1 u^n + \frac{\Delta t}{2} A_2 u^{n+1/2} \\ u^{n+1} = u^{n+1/2} + \frac{\Delta t}{2} A_1 u^{n+1} + \frac{\Delta t}{2} A_2 u^{n+1/2} \end{cases} \tag{PR}$$

It is worth noting that both steps of the Peaceman-Rachford ADI scheme utilize the trapezoidal rule, also known as the Crank-Nicolson method.
We will illustrate that the Peaceman-Rachford ADI scheme possesses discrete-scale properties that closely resemble those observed in the continuum case.

Proposition 3.3.1. *Let $f \in \mathbb{R}^S_+$ and $\Delta t < 2(maxmax|a^1_{i,i}|, max|a^2_{i,i}|)^{-1}$. Then, the Peaceman–Rachford scheme (PR) with splitting 3.12 preserves the average grey value, the positivity and converges to a unique steady-state.*

Proof. We write the Peaceman–Rachford (PR) iteration as

$$u^{n+1} = (I - \frac{\Delta t}{2} A_1)^{-1}(I + \frac{\Delta t}{2} A_2)(I - \frac{\Delta t}{2} A_2)^{-1}(I + \frac{\Delta t}{2} A_2)u^n \tag{3.13}$$

and observe that for$i = 1, 2$ the matrices $P_i^- = (I - \frac{\Delta t}{2} A_i)^{-1}$ and $P_i^+ = (I + \frac{\Delta t}{2} A_i)$,are non-negative and irreducible with unit column sum and positive diagonal entries being each A_i a one-dimensional implicit/explicit discretised osmosis operator satisfying the assumptions of Theorem 3.2.3. Therefore, at every implicit/explicit half-step and using the restriction on Δt, the average grey-value and positivity are conserved. Furthermore, the unique steady-state is the eigenvector of the operator$P = P_1^- P_2^+ P_2^- P_1^+$ associated to the eigenvalue to one. □

The Douglas scheme

Another dimensional splitting method is the Douglas scheme, as described in [8, 7]. For $n \geq 0$, $\Delta t > 0$, and $\theta \in [0, 1]$, the updating rule for the Douglas scheme is as follows:

$$\begin{cases} y^0 = u^n + \Delta t A u^n \\ y^i = y^{i-1} + \theta \Delta t (A_j y^j - A_j u^n), \;\; j = 1, 2 \\ u^{n+1} = y^2 \end{cases} \tag{D}$$

To calculate the numerical approximation at each time step, the Peaceman-Rachford ADI scheme follows a specific procedure. It begins by employing a forward predictor, which is subsequently stabilized by intermediate steps. These intermediate steps exclusively involve the unidirectional components A_j of the splitting,

denoted as $Au = A_1u + A_2u$. Additionally, these intermediate steps are weighted by a parameter θ, which determines the balance between implicit and explicit behavior.
When $\theta = \frac{1}{2}$, the scheme attains a time-consistency order of two. For all other values of θ, the scheme achieves a time-consistency order of one.
The following lemma will be helpful in establishing a result similar to Theorem 3.2.3 for the operator (D).

Lemma 3.3.2. *If $C = (c_{i,j}) \in \mathbb{R}^{S\times S}$ and $D = (d_{i,j}) \in \mathbb{R}^{S\times S} s.t. \sum_{i=1}^{S} c_{i,j} = c$ and $\sum_{i=1}^{S} d_{i,j} = d$ for every $j = 1, \ldots, S$,then,the matrix $B = CD$ has column sum equal to cd.*

Proof. Writing each element $b_{i,j}$ in terms of the elements of C and D then for every j:

$$\sum_{i=1}^{S} b_{i,j} = \sum_{i=1}^{S}\sum_{K=1}^{S} c_{i,K}d_{K,j} = \sum_{K=1}^{S}(d_{K,j}\sum_{i=1}^{S} c_{i,K}) = \sum_{K=1}^{S}(d_{K,j}.c) = c\sum_{K=1}^{S} d_{K,j} = cd \tag{3.14}$$

this concludes the proof. □

Proposition 3.3.3. *Let $f \in \mathbb{R}_+^S, \Delta t > 0$ and $\theta \in [0, 1]$. Then, the Douglas scheme (D) applied to the split semi-discretised scheme 3.7 preserves the average grey value.*

Proof. For every $n \geq 0$, we write the Douglas iteration (D) a

$$\begin{aligned} u^{n+1} &= (I - \theta\Delta tA_2)^{-1}[(I - \theta\Delta tA_1)^{-1}((I - \theta\Delta tA_1) + \Delta tA) - \theta\Delta tA_2]u^n \\ &= (I + \Delta t(I - \theta\Delta tA_2)^{-1}(I - \theta\Delta tA_1)^{-1}A)u^n \\ &= (I + \Delta tP_2P_1A)u^n \end{aligned}$$

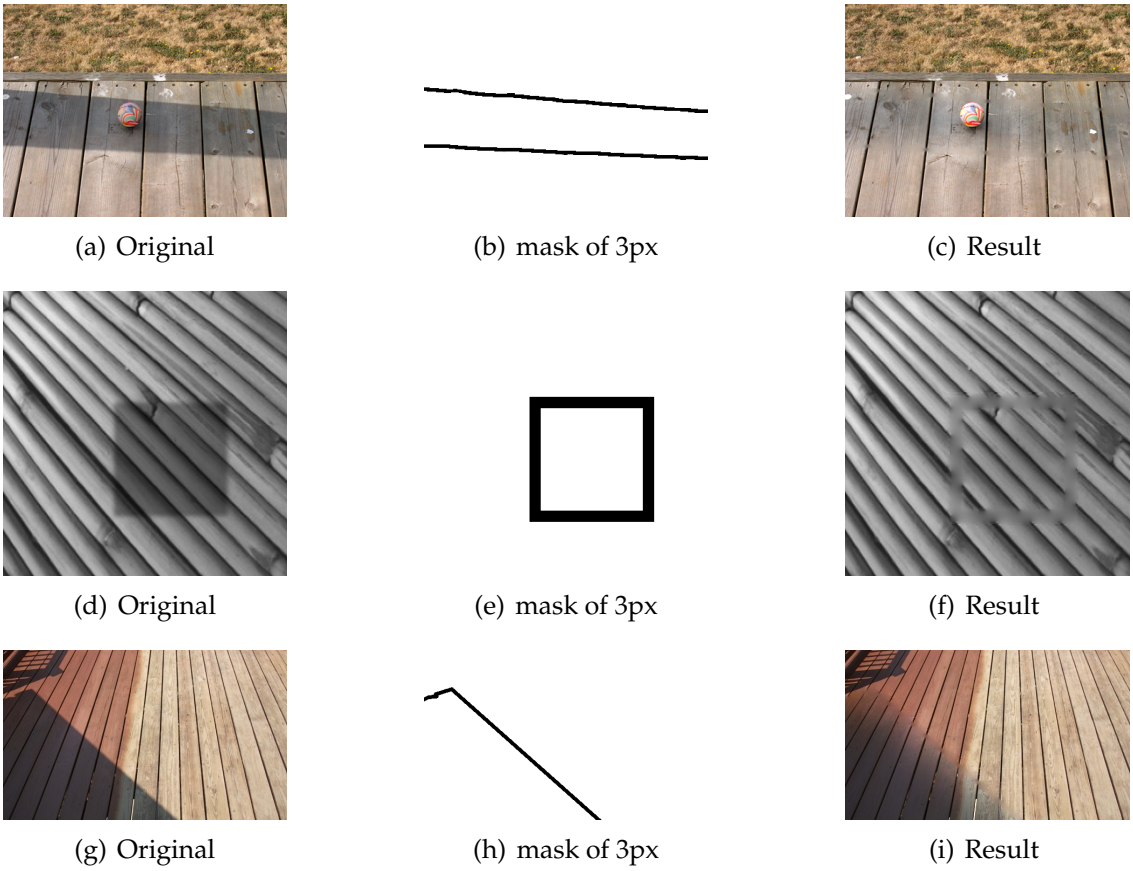
where both P_1 and P_1 are non-negative, irreducible, with unit column sum and strictly positive diagonal entries by Theorem 3.2.4, while $A = A_1 + A_2$ is zero-column sum being the standard discretised osmosis operator 3.7. By Lemma 4.6, the operator $B := P_2P_1A$ has zero column sum and the operator $P = I + \Delta tB$ has unit column sum. Thus, for every $n \geq 0$

$$\frac{1}{S}\sum_{i=1}^{S} u_i^{n+1} = \frac{1}{S}\sum_{i=1}^{S}\sum_{j=1}^{S} p_{i,j}u_j^n = \frac{1}{S}\sum_{j=1}^{S}(\sum_{i=1}^{S} p_{i,j})u_j^n = \frac{1}{S}\sum_{j=1}^{S} u_j^n \tag{3.15}$$

this concludes the proof. □

Remark 3.3.4. The off-diagonal entries of matrix B not being non-negative implies that we cannot directly apply Theorem 3.2.3 to establish that the iterates $(u^n)_{n\geq 0}$ remain positive and converge to a unique steady-state. However, based on our numerical tests, it appears that both properties hold true.

Remark 3.3.5. Both the Peaceman-Rachford scheme (PR) and the scheme (D) are unconditionally stable when the dimensional splitting number is set to s = 2. However, using large time steps in these schemes may result in reduced time-accuracy due to the inclusion of explicit steps.



(a) Original (b) mask of 3px (c) Result

(d) Original (e) mask of 3px (f) Result

(g) Original (h) mask of 3px (i) Result

Figure 3.3: Shadow removal for different images by the isotropic model 3.11.

CHAPTER 4

ANISOTROPIC OSMOSIS FILTERING FOR SHADOW REMOVAL

In this chapter, we enhance the isotropic osmosis model 3.5 and its application for shadow removal.
We introduce the anisotropic osmosis model and its analytical properties, next we explore space and time discretisation schemes. In last section, the anisotropic model is applied to address the shadow removal problem.

4.1 Anisotropic Osmosis Filter

In this section, we enhance the classical osmosis model by incorporating local directional information of the image into the diffusion process. This modification allows for the propagation of geometric structures primarily along the preferred local directions. This variant is referred to as the "anisotropic osmosis model," which contrasts with the isotropic osmosis model 3.5 that does not consider such directional information.

4.1.1 Definitions and modelling

Definition 4.1.1. (Anisotropy matrix) Let $b : \Omega \to \mathbb{R}^2_+$ be a vector field with positive entries such that $b(x) = (b_1(x), b_2(x)) \in \mathbb{L}^\infty(\Omega, \mathbb{R}^2_+)$ and $z(x) = (z_1(x), z_2(x)) : \Omega \to \mathbb{R}^2$ a smooth unitary vector field, i.e.$|z| = 1$. Then, for every $x \in \Omega$ we define the contraction matrix$\Lambda_b(x)$ associated to the vector $b(x)$ and the rotation

matrix$R_z(x)$ associated to the field z as:

$$\Lambda_b(x) = \begin{pmatrix} b_1(x) & 0 \\ 0 & b_2(x) \end{pmatrix}, \qquad R_z(x) = \begin{pmatrix} z_1(x) & -z_2(x) \\ z_2(x) & z_1(x) \end{pmatrix} \tag{4.1}$$

With these matrices we define the anisotropy matrix $M(x)$ in a point$x \in \Omega$as

$$M(x) = \Lambda_b(x) R_z^\top(x)$$

Remark 4.1.2. It should be noted that the matrix R_z represents a rotation matrix, where its columns correspond to the unit vectors z and $z^\perp$ respectively. For any point x in the domain Ω, we can equivalently interpret this operator in terms of the angle $\theta : \Omega \to [0, 2\pi)$ formed by the vector z relative to the Cartesian axes x and y. This allows us to identify $z(x) = (z_1(x), z_2(x)) = (\cos(\theta(x)), \sin(\theta(x)))$.
Next, we denote the symmetric and positive definite anisotropic diffusion matrix as $W = M^\top M$, where M is a matrix representing the anisotropy.

$$W = \begin{pmatrix} b_1^2 z_1^2 + b_2^2 z_2^2 & b_1^2 z_1 z_2 - b_2^2 z_1 z_2 \\ b_1^2 z_2 z_1 - b_2^2 z_2 z_1 & b_1^2 z_2^2 + b_2^2 z_1^2 \end{pmatrix} = b_1^2 (z \otimes z) + b_2^2 (z^\perp \otimes z^\perp). \tag{4.2}$$

It is worth mentioning that the effect of the operator M on the gradient can be expressed more conveniently by utilizing directional vectors z and $z^\perp$. These vectors aid in distinguishing the two components of directional transport.

$$M \nabla u = \begin{pmatrix} b_1 \nabla u \cdot z \\ b_2 \nabla u \cdot z^\perp \end{pmatrix} \tag{4.3}$$

Similarly, in equation 4.2, the matrix W has been formulated in order to distinguish between the different components that arise from z and $z^\perp$.

We can define the energy of the anisotropic osmosis model as follows.

Definition 4.1.3. (Anisotropic osmosis energy) Let $\Omega \subset \mathbb{R}^2$,$u, v \in \mathbb{H}^1(\Omega, \mathbb{R}_+)$ be two positive images and let $W : \Omega \to \mathbb{R}^2$ be a positive definite symmetric matrix field. We define the anisotropic osmosis energy of u with respect to W and the reference image v as

$$E(u) = \int_\Omega v(x) \nabla^\top \left(\frac{u(x)}{v(x)} W(x) \nabla \left(\frac{u(x)}{v(x)} dx. \right.\right. \tag{4.4}$$

We will also use the following alternative notation for E:

$$E(u) = \int_\Omega v(x) \| \nabla \left(\frac{u(x)}{v(x)} \right\|_W^2 dx, \tag{4.5}$$

where $\|e\|_W = \sqrt{\langle e, We \rangle}$.

Remark 4.1.4. (Isotropic case). If we set W to be the identity matrix, equation 4.4 corresponds to the isotropic osmosis energy 3.4 that was studied in reference [18].

Next, we introduce an anisotropic osmosis evolution process that aims to minimize the anisotropic osmosis energy until it reaches a steady-state.

Proposition 4.1.5. *Let $v : \Omega \to \mathbb{R}_+$ be a positive image,$d \in \mathbb{R}^2$the vector field defined as $d = \nabla log v$ and $W : \Omega \to \mathbb{R}^2$ be a positive definite symmetric matrix field. Then, for a given positive image $f \in \mathbb{L}^\infty(\Omega, \mathbb{R}_+)$ the solution of the Euler–Lagrange equation of the functional$E$ defined in Definition 4.1.3 is the steady-state of the anisotropic image osmosis model*

$$\begin{cases} u_t = div(W(\nabla u - du)) \ \ on\Omega \times (0,T] \\ u(x,0) = f(x) \ \ on\Omega \\ \langle W(\nabla u - du), n\rangle \ \ on\partial\Omega \times (0,T] \end{cases} \tag{4.6}$$

Proof. We obtain the optimality condition for any critical point u of E by computing it for any test function $\varphi \in \mathbb{C}_c^\infty(\Omega)$. This is expressed as follows:

$$\begin{aligned} \frac{\partial u}{\partial \tau} E(u + \tau\varphi|_{\tau=0} &= 2\int_\Omega v\langle W\nabla(\tfrac{u}{v}), \nabla(\tfrac{\varphi}{v})\rangle dx \\ &= -2\int_\Omega div(vW\nabla(\tfrac{u}{v}))\tfrac{\varphi}{v} dx \\ &= -2\int_\Omega \tfrac{1}{v} div(vW(\tfrac{\nabla u}{v} - \tfrac{u\nabla u}{v^2}))\varphi dx = 0 \end{aligned} \tag{4.7}$$

By applying the divergence theorem and using Neumann boundary conditions in 4.6, we obtain the following expression. For any $x \in \Omega$ and due to the positivity of v and the fact that φ is compactly supported in Ω, we have:

$$0 = div(vW(\frac{\nabla u}{v} - \frac{u\nabla u}{v^2})) = div(W(\nabla u - \frac{\nabla v}{v}u)). \tag{4.8}$$

By defining $d = \frac{\nabla v}{v}$, we observe that the above equation corresponds to the partial differential equation (PDE).

$$div(W(\nabla u - du)) = 0$$

This equation represents the steady-state solution of equation 4.6. □

The anisotropic osmosis PDE 4.6 exhibits certain properties that are shared with the isotropic osmosis PDE 3.5, making it well-suited for various imaging applications. The convergence of the solution to its steady-state can be established by employing suitable Lyapunov functional, as described in reference **[?]**, which has already been done for the linear isotropic osmosis model. The analysis of the anisotropic model can be conducted using similar arguments.

Theorem 4.1.6. *The solution $u : \Omega \to \mathbb{R}$of the anisotropic osmosis model 4.6 satisfies the following properties:*

***1. conservation of the average grey value**:*

$$\frac{1}{|\Omega|}\int_\Omega u(x,t)dx = \frac{1}{|\Omega|}dx, for\ all t > 0$$

***2. preservation of non-negativity**:*

$$u(x,t) > 0, for\ all x \in \Omega and t > 0$$

3. non-constant steady-states: *The steady-state of 4.6 is given by*

$$w(x) = \frac{\mu_f}{\mu_v} v(x).$$

Proof. [18]
1. Let $\mu u(t) = \frac{1}{|\Omega|}\int_\Omega u(x,t), dx$ denote the average grey value of the image u at time $t \geq 0$. By applying the divergence theorem and utilizing the homogeneous Neumann boundary conditions to equation 4.6, we can derive the following expression:

$$\begin{aligned} \frac{d\mu_u}{dt} = \frac{1}{|\Omega|}\int_\Omega u_t dx = \frac{1}{|\Omega|}\int_\Omega div(W(\nabla u - du))dx \\ = \int_{\partial\Omega}\langle W(\nabla u - du), n\rangle dS = 0 \end{aligned} \tag{4.9}$$

The desired result can be obtained by following the steps outlined above.
2. To rewrite the anisotropic osmosis PDE in an extended diffusion-transport-reaction form for any $(x,t) \in \Omega \times (0,T]$, we can express it as follows.

$$\begin{aligned} u_t(x,t) &= div(W\nabla u(x,t)) - div(Wdu(x,t)) \\ &= W \cdot D^2u(x,t) + (div(W) - Wd) \cdot \nabla u(x,t) - div(Wd)u(x,t) = Lu \end{aligned} \tag{4.10}$$

Given that W is a symmetric positive definite matrix, we can observe that the operator L is uniformly strongly elliptic. Along with the Neumann-type boundary conditions and the positivity of the initial data f, this implies, as stated in reference [22], that the solution u remains positive for all $t > 0$. Similar conclusions can be drawn for the isotropic model, as discussed in references [23].
3.For any $c \in \mathbb{R}$, the steady-state equation of the anisotropic PDE system can be solved by considering $w = cv$ and applying the Neumann-type boundary conditions as specified in equation 4.6. It can be verified that the equation is satisfied when $w = cv$.

$$div(W(\nabla w - \frac{\nabla v}{v}w)) = div(W(c\nabla v - \frac{\nabla v}{v}cv)) = 0 \tag{4.11}$$

The conservation laws of mass and non-negativity ensure that the process reaches a non-negative steady-state solution w that has the aforementioned form. Furthermore, the constant $c \in \mathbb{R}$ can be easily determined by noting that:

$$c\mu_v = \frac{1}{|\Omega|}\int_\Omega cv(x)dx = \frac{1}{|\Omega|}\int_\Omega w(x)dx = \frac{1}{|\Omega|}\int_\Omega f(x)dx = \mu_f \tag{4.12}$$

Therefore, the constant c can be determined by setting $c = \frac{\mu_f}{\mu_v}$, and this assignment is well-defined since v is strictly positive throughout the region Ω. □

4.1.2 Anisotropic diffusion inpainting

Anisotropic diffusion inpainting, which utilizes a diffusion tensor, was first introduced in [29] and has shown successful applications in inpainting-based compression [**?**, **?**]. This method takes advantage of the edge-enhancing anisotropic

diffusion filter proposed for image denoising [20]. To propagate structures from specified image regions into inpainting regions, the differential operator $div(W(\nabla u_\sigma)\nabla u)$ is employed, where u_σ represents the convolution of u with a Gaussian filter having a standard deviation of σ. The diffusion tensor W consists of eigenvectors parallel and perpendicular to ∇u_σ, with corresponding eigenvalues given by:

$$\mu_1(|\nabla u_\sigma|) = \frac{1}{\sqrt{1 + |\nabla u_\sigma|^2/\lambda^2}} \text{ and } \mu_2(|\nabla u_\sigma|) = 1 \tag{4.13}$$

The aim is to perform inpainting by aligning it with the orientation of a structure while reducing inpainting in the perpendicular direction, controlled by a high contrast parameter $\lambda > 0$. These methods have proven effective in inpainting edge-like structures, even in scenarios with sparse available data and large gaps to be filled [**?**]. However, they are not suitable for shadow removal applications due to the presence of non-physical edges caused by shadow boundaries. To overcome this limitation, a modification is required using more sophisticated structure descriptors such as tensor voting. Tensor voting can provide more robust local structure directions, even in the presence of shadow boundaries. The details of this modification will be discussed in the following step.

4.1.3 Computation of structure directions via tensor voting

In this section, we present a workflow that aims to achieve local robustness against lighting variations caused by shadows. This approach allows us to introduce anisotropy in the specific directions that are considered suitable within the domain Ω_{sb}.
A widely used technique for estimating the orientation of the local gradient in an image u is to compute the eigenvector e_1 associated with the largest eigenvalue λ_1 of the structure tensor $J_\rho(u)$, as described in Section 2.3.4.
The concept of *Tensor Voting* was initially introduced in [**?**] as a technique for extracting curves from noisy images by grouping local features that exhibit consistency within a neighborhood of the measurements. This framework enhances the robustness of structure tensor estimation in the presence of noise and image artifacts [**?**]. Given a 2-tensor B in $\mathbb{R}^2$, it can be represented by the following matrix:

$$W = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \lambda_1(e_1 \otimes e_1) + \lambda_2(e_2 \otimes e_2). \tag{4.14}$$

Equation 4.14 can be rewritten in an equivalent form, where λ_1 and λ_2 represent the eigenvalues corresponding to the eigenvectors e_1 and e_2, respectively. The expression is as follows:

$$B = (\lambda_1 - \lambda_2)(e_1 \otimes e_1) + \lambda_2(e_1 \otimes e_1 + e_2 \otimes e_2) \tag{4.15}$$

Now, we can distinguish between the following two quantities:

- $(\lambda_1 - \lambda_2)$ is referred to as *saliency* or *stickiness*. It represents an estimation of the confidence in the direction e_1 and is also known as *orientation certainty* or *anisotropy measure*;

- λ_2 is referred to as *ballness* and indicates the size of the minor axis of the anisotropy ellipse. It quantifies the consistency in estimating the main direction and is often referred to as *orientation uncertainty* or *junctionness*.

The tensor voting process involves incorporating the influences of neighboring tensors into the tensor field at each iteration. This iterative update improves the tensor field by taking into account the saliency parameter. However, the original approach can be computationally demanding, especially for larger images. To address this concern, the authors propose an efficient implementation of the tensor voting framework using steerable filters theory, which utilizes complex-valued convolutions. This implementation offers computational advantages and allows for faster processing of the tensor voting algorithm.
When it comes to shadow removal, accurately estimating the structure direction e_1 can be challenging due to the presence of shadow edges that do not align with actual image structures. As a result, both the structure tensor and tensor voting methods face difficulties in accurately estimating local directions in such scenarios.
To address the challenge of estimating local directions in the presence of shadow edges, we propose a modification to the tensor voting framework. In our approach, we consider shadow edges as a bias in the estimation of image structure and adjust the computation of saliency and orientation accordingly. The modified framework is then applied to the shadowed image, where the saliency is set to zero and the orientation is randomized at the shadow boundaries. Further details about the algorithm and its results can be found in Section 4.1.3.

4.1.4 Anisotropic osmosis-inpainting model for shadow removal

We propose a structure-preserving osmosis model to solve the problem of shadow removal for a positive grey-scale image $f : \Omega \to \mathbb{R}^+$ with a constant shadow, where Ω is decomposed as shown in equation 3.10.

$$\begin{cases} u_t = div(W(\nabla u - du)) \ \text{on}\Omega \times (0,T] \\ u(x,0) = f(x) \ \text{on}\Omega \\ \langle W(\nabla u - du), n\rangle = 0 \ \text{on}\partial\Omega \times (0,T] \end{cases} \tag{4.16}$$

Here, we introduce the definitions of the discontinuous vector field d and the discontinuous matrix field W as follows:

$$d(x) = \begin{cases} \nabla log\text{f} & \text{if} \quad x \in \Omega \backslash \Omega_{sb}, \\ 0 & \text{if} \quad x \in \Omega_{sb} \end{cases} \tag{4.17}$$

$$W(x) = \begin{cases} I & \text{if} \quad \Omega \backslash \Omega_{sb}, \\ \epsilon(e_1 \otimes e_1) + 1(e_1 \otimes e_1) & \text{if} \quad x \in \Omega. \end{cases} \tag{4.18}$$

In this equation, the symbol I denotes the 2×2 identity matrix, while the directions e_1 and e_2 are derived from the modified tensor voting method, which is applied to the original image f.
To summarize, our proposed method combines isotropic osmosis in the non-shadowed region (Ω_{sb}^c) and anisotropic inpainting on the shadow boundary (Ω_{sb}). This approach ensures a balanced intensity in the shadowed region by applying classical osmosis, while preserving structures and avoiding blurring at the shadow boundary through interpolation. By performing joint osmosis and inpainting, the need for additional post-processing steps is eliminated.

4.2 Space-time discretisation

To discretize the anisotropic osmosis model while preserving its properties, we consider a rectangular image domain with $M \times N$ pixels. The positive initial image f is represented as a vector in $\mathbb{R}_+^S$. The approximation of the function u is denoted as $u = (u_{i,j})_{i,j}$, where $u_{i,j}$ represents its approximated value at suitable discretization nodes $((i-\frac{1}{2})\Delta x, (j-\frac{1}{2})\Delta x)$, with $i = 1, \ldots, M$ and $j = 1, \ldots, N$. At the time node $t_k = k\tau$, the value of $u_{i,j}^n$ is denoted as$u_{i,j}^k$. For $x \in \Omega$, the discretized eigenvalues $\lambda_1(x)$ and $\lambda_2(x)$ are denoted as $\lambda_1, \lambda_2 \in \mathbb{R}_+^S$, and the discretized orientation $\theta(x)$ is denoted as $\theta \in [0, 2\pi)$.

4.2.1 Operator splittings : AOS, MOS and AMOS

The operator splitting approach, by decomposing the operator as $A = A_1 + A_2$, allows for the development of two unconditionally stable schemes: the additive operator splitting (AOS) and the multiplicative operator splitting (MOS). These schemes were proposed in [20, 4] and have proven to be highly effective in efficiently solving nonlinear diffusion models.

Additive and multiplicative operator splitting: AOS and MOS

The methods called additive operator splitting (AOS) and multiplicative operator splitting (MOS) are iterative techniques that can be applied for any non-negative integer n and a positive time-step Δt. These methods are defined as follows:

$$u^{n+1} = \frac{1}{2}\sum_{k=1}^{2}(I - 2\Delta t A_k)^{-1}u^n \qquad \text{(AOS)}$$

$$u^{n+1} = \prod_{k=1}^{2}(I - \Delta t A_k)^{-1} \qquad \text{(MOS)}$$

which are stable for any $\Delta t > 0$ and first-accurate in time. The tridiagonal structure of $(I - \Delta t A_k)^{-1}, k = 1, 2$ allows for efficient inversion, e.g. by LU factorisation.
The following proposition guarantees that when the AOS and MOS iterations are

applied to the osmosis problem, they preserve the average gray value and positivity of the solution, and eventually converge to a unique steady-state solution:

The AOS and MOS iterative methods used for solving the osmosis problem have the property of preserving the average gray value and maintaining positivity at each iteration. Additionally, these methods exhibit convergence towards a single steady-state solution.

Proposition 4.2.1. *For a starting $f \in \mathbb{R}_+^S$ the schemes (AOS) and (MOS) with splitting 3.12 preserve the average grey value, positivity and converges to a unique steady-state for any $\Delta t > 0$.*

Proof. Let $S = MN$. We first notice that the operators $P_i^- = (I - \Delta t A_i)^{-1}$,with $i = 1, 2$ are non-negative and irreducible with unit column sum and positive diagonal entries being eachA_ii one-dimensional osmosis discrete space operator fulfilling Theorem 3.2.4.Thus the operator $P_{AOS} = \frac{1}{2}(P_1^- + P_2^-)$ is non-negative, irreducible with unit column sum and positive diagonal entries. Similarly, the operator$P_{MOS} = P_1^- P_2^-$ is trivially non-negative and irreducible, with unit column sum by Lemma 3.3.2 and with strictly positive elements on the diagonal:

$$\begin{aligned} P_{i,i} &= \sum_{K=1}^{S} (P_1^-)_{i,K} (P_2^-)_{K,i} \\ &= (P_1^-)_{i,i}(P_2^-)_{i,i} + \sum_{K=1, K\neq i}^{N} (P_1^-)_{i,K}(P_2^-)_{K,i} > 0 \text{ for all } i = 1, \ldots, S \end{aligned} \tag{4.19}$$

For any $n > 0$, the AOS time-stepping $u^{n+1} = P_{AOS} u^n$ and MOS time-stepping $u^{n+1} = P_{MOS} u^n$ satisfy Theorem 3.2.3. Therefore, in both cases, the iterates u^{n+1} preserve the average gray value and positivity for any $\Delta t > 0$. Finally, the unique steady-state solution of the AOS and MOS iterations corresponds to the eigenvector associated with the eigenvalue one of P_{AOS} and P_{MOS}, respectively. □

Additive–multiplicative operator splitting (AMOS)

In the reference [4], a refined variant of the iterative approach, called the additive-multiplicative operator splitting (AMOS), is introduced. The AMOS method combines the additive operator splitting (AOS) and multiplicative operator splitting (MOS) in a more precise manner. The update rule for the variable u^{n+1} in this method can be expressed as:

$$u^{n+1} = \frac{1}{2} \sum_{k=1}^{2} ((I - \Delta t A_{jn})^{-1} (I - \Delta t A_{in})^{-1}) u^n, \text{ with } i = \{1, 2\}, j = \{2, 1\} \quad \text{(AMOS)}$$

Result of Proposition 4.2.1 holds trivially for (AMOS) by combining (AOS) and (MOS).

Corollary 4.2.2. *When starting with an initial value $f \in \mathbb{R}_+^S$, the (AMOS) scheme with the given splitting rule (referring to equation 38 in the reference) ensures the preservation of the average gray value and maintains positivity at each iteration. Furthermore, the scheme converges to a unique steady-state solution for any positive time-step $\Delta t > 0$.*

4.3 Numerical results

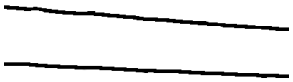
In this section, we present the utilization of isotropic and anisotropic osmosis models to address shadow removal in synthetic and real-world images. We provide various numerical examples to showcase the effectiveness of these models. Specifically, the anisotropic model 4.16 is employed for images containing predominantly uniform shadows, allowing for the simultaneous removal of shadows and inpainting of shadow edges. As obtaining ground truth is often impractical in this scenario, we evaluate the reconstruction quality of the anisotropic model through visual inspection and compare it to the isotropic approach.

4.3.1 On the thickness of the shadow boundary

In the following experiments, we make the assumption that a rough segmentation of the shadow boundary has already been provided, for the sake of simplicity. However, obtaining an accurate shadow boundary segmentation can be challenging in the case of natural real images, as the boundaries may not be well-defined due to various factors such as noise, blurring, and compression artifacts. Consequently, standard segmentation methods relying on edge detection may not yield satisfactory results. Previous studies have tackled the task of shadow segmentation separately, employing techniques such as brightness-based or clustering approaches. In many practical scenarios, the user manually approximates the shadow boundary by using a brush that includes pixels from both inside and outside the shadowed area (referred to as Ω_{sb} in Section 4.1.1).
Figure 4.1 presents these findings. We conducted a comparison by applying the isotropic model 3.11 to a real image with varying thickness of the shadow boundary. It is evident that a thicker Ω_{sb} leads to a more effective removal of the shadow, particularly for the same extended final time T.

4.3.2 Synthetic examples

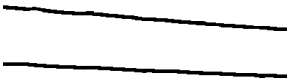
In this experiment, we apply the anisotropic osmosis model to synthetic and real-world images. The aim of this experiment is to test if this method can effectively remove constant shadows. We present the results for an image with parallel grey-scale, and for color concentric circles with an angle drawn tangentially to the circumference. Both images have a constant shadow, but a small transition zone with blurred shadows was added for a more realistic effect. We compare the anisotropic model described in this chapter with the isotropic osmosis model described in chapter 3, which results in homogeneous diffusion inpainting at the shadow boundary. The anisotropic shadow removal method shows clear advantages at the shadow boundaries, since it does not suffer from blurring artifacts, as can be seen in the visual comparison of both methods.



(a) mask of 3px

(b) Isotropic Osmosis

(c) Final Result

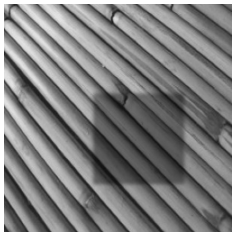


(d) mask of 3px

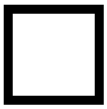
(e) Result

(f) Final Result

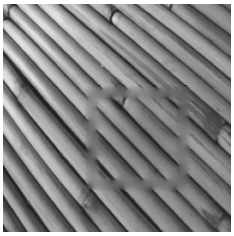
Figure 4.1: Figures Comparision between different thicknessees of the shadow boundary for the solution of the isotropic and the anisotropic model.



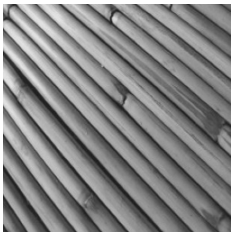
(a) Shadowed image



(b) Mask



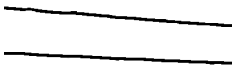
(c) Isotropic osmosis



(d) Anisotropic osmosis

Figure 4.2: Shadow removal via isotropic and anisotropic osmosis.

(a) Shadowed image



(b) Mask

(c) Isotropic

(d) Anisotropic

Figure 4.3: Shadow removal via isotropic and anisotropic osmosis.

Conclusion

Osmosis filtering has proven to be a powerful tool for image restoration and enhancement, and in this paper, we have presented an isotropic and anisotropic osmosis model for shadow removal in images. The models studied in this work, are based on the principles of osmosis filtering and uses the isotropic and the anisotropic diffusion matrix to incorporate edge information in the filtering process.

We have discuss the effectiveness of this models through several experiments on both synthetic and natural images. The results show that the anisotropic osmosis model outperforms existing methods for shadow removal in terms of visual quality and quantitative metrics.

Finally, we have discussed the connection between the anisotropic diffusion term in the model and specific diffusion-based inpainting models.

Overall, this work illustrates the potential of osmosis filtering as a versatile tool for image restoration and enhancement, and we expect that this model will find application in various imaging applications beyond shadow removal.

Bibliography

[1] L. Alvarez,F. Guichard, P.L. Lions,and J.M. Morel, *Axioms and fundamental equations of image processing*, Archive for Rational Mechanics and Analysis **123**, 199–257 (1993).

[2] G. Aubert, P. Kornprobst, Mathematical Problems in Image Processing, Partial Differential Equations and the Calculus of Variations, Applied Mathematical Sciences147,(2006).

[3] M. Bertero, T.A. Poggio, and V. Torre, *"Ill-posed problems in early vision"*, Proceedings of the IEEE, vol. *76*, pp. 869–889, 1988.

[4] D. Barash, M. Israeli, and R. Kimmel. An accurate operator splitting scheme for nonlinear difusion filtering. In M. Kerckhove, editor, *Scale-Space and Morphology in Computer Vision*, pages 281–289, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[5] T.F. Chan and J. Shen. *Image Processing and Analysis: Variational, PDE, Wavelet,and Stochastic Methods*. Society for Industrial and Applied Mathematics,(2005).

[6] X. Chunxia, S. Ruiyun, X. Donglin, and M. Kwanâ-Liu. Fast shadow removal using adaptive multi-scale illumination transfer. *Computer Graphics Forum*,32(8):207–218, 2013. doi:10.1111/cgf.12198.

[7] J. Jr. Douglas and H.H. Jr. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82(2):421–439, 1956. doi:10.2307/ 1993056.

[8] J. Douglas, Jr. On the numerical integration of $\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial^2 y} = \frac{\partial u}{\partial t}$by implicit methods. *Journal of the Society for Industrial and Applied Mathematics*, 3(1):42–65, 1955. doi:10.1137/0103004.

[9] I. Galic, J. Weickert, M. Welk, A. Bruhn, A. Belyaev and H.P. Seidel, *Towards PDE-based image compression*. Variational Geometric and Level-Set Methods in Computer Vision, **3752**, 37–48 (2005).

[10] R. Glowinski, T.W Pan, and X.C Tai. *Some Facts About Operator Splitting and Alternating Direction Methods*, pages 19–94. Springer International Publishing, Cham, 2016.

[11] J. Koendernik, *'The structure of images'*, Biol. Cybern, **50**, pp. 363-370 (1984).

[12] S. Parisotto, Anisotropic variational models and PDEs for inverse imaging problems,December 2018

[13] P. Perona and J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence,1987.

[14] D. Peaceman and H. Rachford, Jr. The numerical solution of parabolic and elliptic differential equations. Journal of the Society for Industrial and Applied Mathematics, 3(1):28–41, 1955. doi:10.1137/0103003.

[15] Q. Sheng. Solving linear partial differential equations by exponential splitting. *IMA Journal of Numerical Analysis*, 9(2):199–212, 04 1989. doi:10.1093/imanum/9.2.199.

[16] C.B. Schonlieb, *Partial Differential Equation Methods for Image Inpainting*. Cambri dgeUniversity Press **29**, (2015).

[17] O. Vogel, K. Hagenburg, J. Weickert, and S. Setzer. A fully discrete theory for linear osmosis filtering. *Scale Space and Variational Methods in Computer Vision*, pages 368–379, 2013.

[18] J. Weickert, K. Hagenburg, M. Breuß and O. Vogel, Linear osmosis models for visual computing. In:A. Heyden, F. Kahl, C. Olsson, M. Oskarsson, X.C. Tai (eds.) Energy Minimization Methods in Computer Vision and Pattern Recognition, pp. 26–39. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). DOI 10.1007/978-3-642-40395-8 3

[19] J. Weickert, Romeny, M., and Viergever, A. , Efficient and Reliable Schemes for Nonlinear Diffusion Filtering,1998.

[20] J. Weickert, *Anisotropic Diffusion in Image Processing*, Stuttgart Teubner, (1998).

[21] R. M. Beam and Warming R. F. An implicit finite-difference algorithm for hyperbolic systems in conservation-law form. *Journal of Computational Physics*, 22(1):87–110, 1976. doi 10.1016/0021-9991(76)90110-8. (Cited on page 143.)

[22] Cantrell R. S. and Cosner C. *Spatial Ecology via Reaction-Diffusion Equations*. Wiley Series in Mathematical and Computational Biology, Wiley,2003. doi:10.1002/0470871296. (Cited on page 159.)

[23] M. Schmidt. *Linear Scale-Spaces in Image Processing: Drift-Diffusion and Connections to Mathematical Morphology*. PhD thesis, Dept. of Mathematics,Saarland University, Saarbrücken, Germany, 2018. (Cited on pages 158 and 159.)

[24] A.P. Witken, *'Scale-space filtering'*, Presented at 8th int. Joint conf. Art. intell., Karlsruhe, Germany, (1983).

[25] G. Strang. Accurate partial difference methods i: Linear cauchy problems.*Archive for Rational Mechanics and Analysis*, 12(1):392–402, Jan 1963. URL:https://doi.org/10.1007/BF00281235, doi:10.1007/BF00281235. (Cited on page 143.)

[26] G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968. doi:10.1137/0705041.(Cited on page 143.)

[27] W. Hundsdorfer and J.G. Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 2007. doi:10.1007/ 978-3-662-09017-6. (Cited on pages 143 and 147.)

[28] Robert I. McLachlan and G. Reinout W. Quispel. Splitting methods. *Acta Numerica*, 11:341—-434, 2002. doi:10.1017/S0962492902000053. (Cited on page 143.)

[29] Roland Glowinski, Tsorng-Whay Pan, and Xue-Cheng Tai. *Some Facts About Operator Splitting and Alternating Direction Methods*, pages 19–94. Springer International Publishing, Cham, 2016.

Abstract :

The use of (PDEs) has a long tradition in mathematical image processing. In particular, a very special place is a transport-diffusion PDE describing the physical phenomenon of osmosis for imaging application.

The goal of this work is to study the foundations of osmosis filtering for imaging applications.

Keywords:

Image Processing, Edge Enhancement, Nonlinear Diffusion, Linear Diffusion, Finite Difference, osmosis filter, shadow removal.

Résumé:

L'utilisation des (PDE) a une longue tradition dans le traitement mathématique des images. En particulier, une place toute particulière est occupée par une EDP de transport-diffusion décrivant le phénomène physique d'osmose pour une application en imagerie.

L'objectif de ce travail est d'étudier les fondements du filtrage par osmose pour les applications d'imagerie.

Mots-clés :

Traitement d'image, Amélioration des contours, Diffusion non linéaire, Diffusion linéaire, Différences finies, Filtre d'osmose, Suppression des ombres.

ملخص

استخدام المعادلات التفاضلية الجزئية لديه تقليد طويل في مجال معالجة الصور الرياضية. وعلى وجه التحديد، تحتل المعادلة التفاضلية الجزئية للنقل والانتشار مكانة خاصة جدا في وصف ظاهرة فيزيائية لتناضح لتطبيق التصوير.

الهدف من هذا العمل هو دراسة أسس تصفية التناضح لتطبيقات التصوير.

الكلمات المفتاحية

معالجة الصور، تحسين الحواف، انتشار غير خطي، انتشار خطي، الفروق المحدودة، مرشح التناضح، إزالة الظل.