People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University of Saad Dahlab Blida 1
Faculty of Science
Department of Mathematics

Dissertation

For the degree of

MASTER

Domain : Mathematics and Computer Science
Branch : Mathematics
Option : Analysis and Applied Mathematics

Image Fusiom using a joint-variational Osmosis model

Presented by :

Lakraa Redouane

In 20th Jully 2022 infront of the following Jury :

Mr. Benbachir Maamar	Professor, USD-BLIDA1	Presedent of jury
Mr. HachamaMohammed	Professor, USD-BLIDA1	Superviser
Mr.Boudjemaa Redouane	Doctor, USD-BLIDA1	Examiner

2021/2022

Image Fusion

using a joint-variational Osmosis model

Lakraa Redouane

Abstract

The main purpose of this work is the fusion of multiple images to a single composite that offers more information than the individual input images. We focus the approach within a variational framework. First, we present the most basic variational model which is the Poisson editing and follow it up by Osmosis. Osmosis is a transport phenomenon that is omnipresent in nature. It differs from diffusion by the fact that it allows nonconstant steady states. Then we study a proposed modification to this model that is called joint-variational Osmosis that makes the overall term non-convex. The minimization of this new non-convex model gives plausible image data fusion.

We minimize it using the inertial Porixmal algorithm for non convex optimization algorithm (iPiano), we apply the resulting minimization scheme to solve multi-modal face fusion, color transfer and cultural heritage conservation problems.
Comparing this result with famous models visualy or quantitatively using error mesures shows the superiority and flexibility of this method.

Keywords: **Image fusion, Variational image fusion, Osmosis model, drfit-diffusion, non-convex optimization, gradient descent algorithms, proximal algorithms**

Acknowledgements

I want to express my grateful thanks to all the people that supported me in doing this work. Without them, this thesis would certainly not have been possible. First of all, I would like to express my gratitude to Prof. Benbachir Maamarr for giving us the opportunity to join the new AMA branch and be the first wave of students to come out of this formation. Next I would like to thank Prof. Hachama Mohammed first for his teaching troughout the two years, and for building the love and intrest in us for this great field of applied mathematics, and second for his supervision, guidance and patience while making this thesis.

I want also to gratefully thank Dr. Boudjemaa Redouane for being the reviewer of this thesis.
Certainly, my thanks also go to Dr. Simone Parisotto and Luca Calatroni, author and co-author of the article "Variational Osmosis for Non-linear Image Fusion". As well as Dr. Marco Caliari co-author of "Anisotropic osmosis filtering for shadow removal in images" for thier great help, and for Prof. Joachim Weickert for his brief but meaningful assistance.
Last but not least, I want to thank my family for the continuous and unconditional support throughout my whole studies. Thank you very much!

Contents

1 Introduction to mage fusion **1**
1.1 Image fusion 1
1.2 Applications of image fusion 2
1.2.1 Underwater image enhancement 2
1.2.2 Exposure Fusion 2
1.2.3 Image and Video Decolorization by Fusion 3
1.2.4 Focus Fusion 3

2 PDEs and variational models **5**
2.1 Inputs and Notations 5
2.2 Calculus of Variations and Numerical Solution 5
2.3 Poisson editing 6
2.3.1 Motivation 6
2.3.2 Poisson editing using guidance 7
2.3.3 Seamless cloning 8
2.4 Image Osmosis 9
2.4.1 Continuous Linear Osmosis Filtering 9

3 The joint variational Osmosis model **12**
3.1 Inputs of the model 12
3.2 The new fusion term 13
3.3 Fidelity and regularisation 14
3.4 Numerical resolution: The iPiano algorithm 15
3.5 Computing algorithm components 16
3.5.1 Gradient of Osmosis term 16
3.5.2 Proximal operator of the fidelity and regularisation terms 18
3.6 Algorithm 19
3.6.1 Results 20

4 Conclusion **23**

Chapter 1

Introduction to mage fusion

1.1 Image fusion

In the late 1970s, with the emergence and development of image sensors, image fusion emerged as a new research field in the intersection of sensors, signal and image processing, and artificial intelligence. Image fusion combines the image information about the same scene obtained either by multiple image sensors or by the same image sensor in different working modes to obtain a new and more accurate description of the scene as illustrated on Figure (1.1).

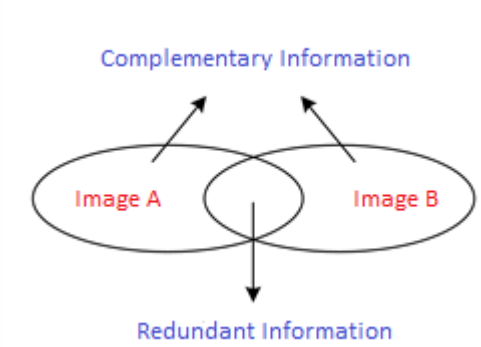


Figure 1.1: Image fusion principle

Using complementary images allows to construct a more complete image and improve reliability by exploiting redundant information.. Albeit this is just one type of fusion where we fuse a selected region, other methods can fuse different exposures from multiple images or multiple color channels from one single image as will be shown in the next sections.

Significant research efforts have accordingly been dedicated to the development of image fusion, with a great number of image fusion algorithms having been proposed in the literature. Figure (1.2) shows that the number of scientific journals, conferences and papers published on the topic of image fusion has increased dramatically since 2010.

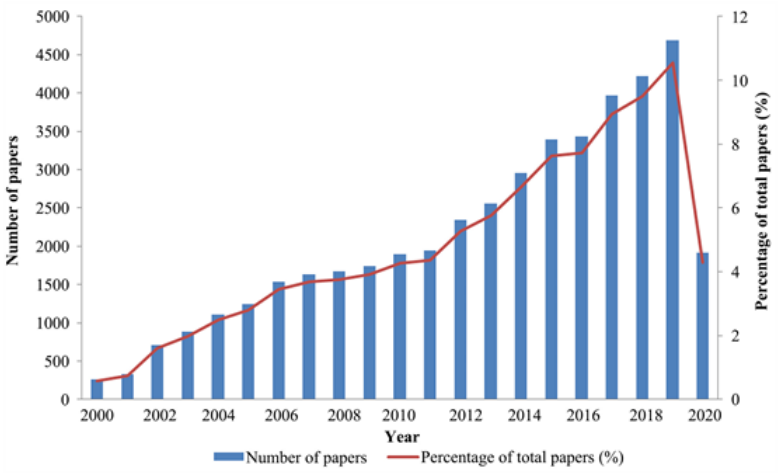


Figure 1.2: Statistics on Web of Science from 2000 to 2019. Statistics of 2020 are not full.

1.2 Applications of image fusion

1.2.1 Underwater image enhancement

Underwater imaging is challenging due to the physical properties existing in such environments. There have been several attempts to restore and enhance the visibility of such degraded images but each of them has a drawback. The enhanced image version is obtained by fusing some inputs. Next figure shows a comparison against the standard Histogram equalization method.

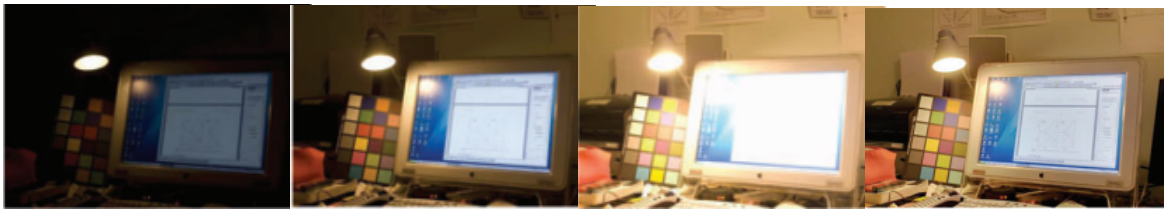
(a) Original image (b) Histogram equalization (c) Fusion result

Figure 1.3: Enhancing underwater image using a fusion method

1.2.2 Exposure Fusion

Exposure Fusion is a simple and practical alternative to High Dynamic Range Photography (HDR). It fuses multiple exposures into a high-quality, low dynamic range image. Guided by some quality measures proposed we select the 'good' pixels from the sequence and combine them into the final result. With standard consumer cameras is it not always possible to capture all the bright and dark details of a real-world scene with a single acquisition. Taking several images with a standard camera while changing the exposure

settings allows to produce an overall well-exposed image in a post-processing step. Figure (1.4) shows the result of the fusion approach proposed by Mertens et al. [15].

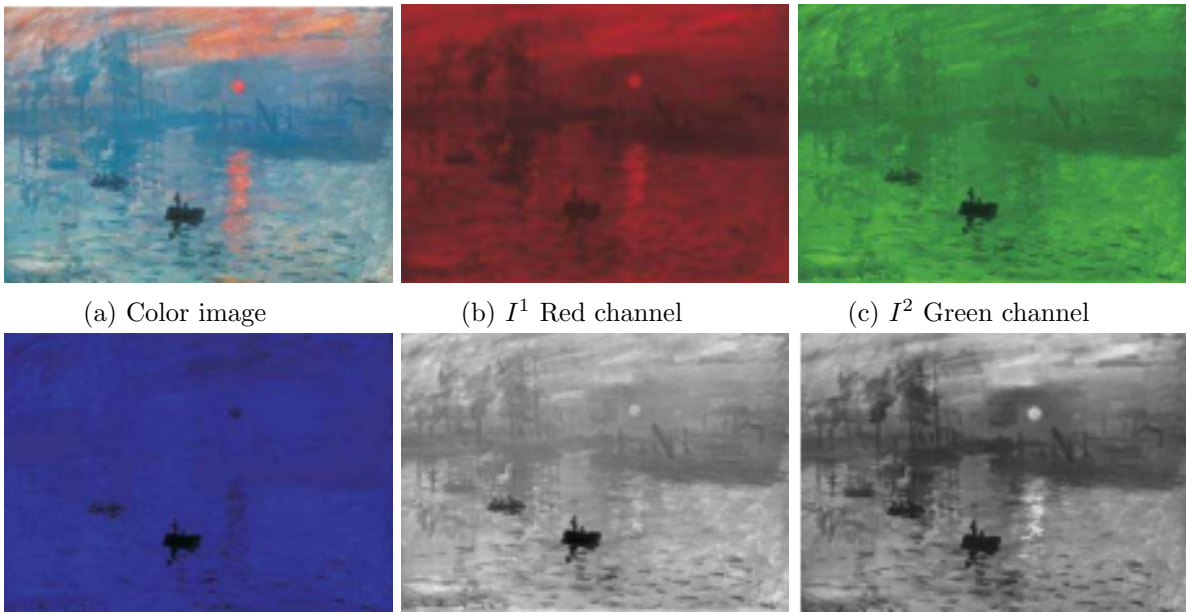


(a) under-exposed (b) poorly exposed (c) over-exposed (d) Result

Figure 1.4: Exposure Fusion

1.2.3 Image and Video Decolorization by Fusion

Although color plays an important role in images, applications such as compression, visualization of medical imaging, aesthetical stylization, displaying data on monochrome devices such as electronic book readers and printings require reliable decolorized image versions. Decolorization is the conversion of a color image to its greyscale representation. The main challenge of decolorization is to preserve as much information as possible.



(a) Color image (b) I^1 Red channel (c) I^2 Green channel

(d) I^3 Blue channel (e) I^4 Helmholtz-Kohlrausch (f) The fusion method result

Figure 1.5: Color, input images and fused decolorized image

1.2.4 Focus Fusion

A typical problem in small scale photography and optical microscopy is the restricted depth of field of common cameras and microscopes. Objects only appear sharp at a certain

distance range to the imaging device. Hence, it is often not possible to capture a single image that is sharp everywhere. A common remedy is to take several photographs while varying the focal settings. Focus fusion describes the task of combining the acquired focal stack to an all-in-focus composite that is desirably sharp in every image region. Figure (1.6) shows different focus distances and the obtained sharpness from focus stacking. Fusing this focal stack results in a composite image that provides the desired sharpness.

(a) Near focus (b) Middle focus (c) Far focus (d) Focus stacking

Figure 1.6: Different focus distances and focus stacking

Chapter 2

PDEs and variational models

2.1 Inputs and Notations

The goal of variational fusion is to obtain an optimal image u as a to an PDE or the minimization of an energy functional. The observed (initial) image we look to improve will be refereed as f. Both f and u are functions defined on $S \in \mathbb{R}^2$ with values in $\mathbb{R}$. We also note b the background image ; we are partially interested in its domain S, which we will fuse with image f. This part of S we note Ω (See Figure(2.1)). We will use a drift

(a) Image f (b) Image b has better soil Ω (c) Fused image u

Figure 2.1: Image fusion example

vector field, noted $\mathbf{d}$. After establishing our PDE, descretizing it is the next step. So S and Ω become finite point sets defined on an discrete grid, we note them S_d and Ω_d.

2.2 Calculus of Variations and Numerical Solution

Often we have that our operations such as cloning, shadow removal or inpainting are performed by applying partial differential equations, that are the solution of a minimization problem. So how did we converge from a minimization problem such as $(\min_u \int \|\boldsymbol{\nabla} u\|_2^2 \, dx)$ to a PDE. And how do we implement these linear PDEs to our images that are discrete objects.

Variational Framework Because images are two-dimensional functions, we will restrict ourselves to functions of $\boldsymbol{x} = (x_1, x_2)^\intercal \in \Omega \subset \mathbb{R}^2$. Also, we consider vector-valued functions $\boldsymbol{u} = (u_1, ..., u_n)^\intercal$. The energy functional is given by

$$E(u) = \int_\Omega F(u_1, ..., u_n, \boldsymbol{\nabla} u_1, ..., \boldsymbol{\nabla} u_n)\ dx\ , \tag{2.1}$$

where u is the desired solution from minimizing $E(u)$.
According to the calculus of variations, the necessary conditions for a minimiser of the energy (2.1) are given by the so-called *Euler-Lagrange* equations

$$F_{u_i} - \partial_{x_1} F_{\partial x_1 u_i} - \partial_{x_2} F_{\partial x_2 u_i} = 0 \quad (i = 1, ..., n) \tag{2.2}$$

Whith $\boldsymbol{n}$ as the outer normal vector on the boundary $\partial\Omega$, the corresponding boundary conditions read

$$\boldsymbol{n}^\intercal \begin{pmatrix} F_{\partial x_1 u_i} \\ F_{\partial x_2 u_i} \end{pmatrix} = 0 \quad (i = 1, ..., n) \tag{2.3}$$

2.3 Poisson editing

2.3.1 Motivation

The gradient of images could be directly modified to perform useful operations; this operation is called gradient-based image processing or Poisson editing. For example operations such as seamless cloning, contrast enhancement or shadow removal can be performed in an efficient way by modifying the image gradients. The idea was proposed by Perez et al. [20] and proved to be a turning point in image processing. Lets say we want to clone part of image b (region Ω) on an image f seamlessly (see Figure(2.2)).

- The eye is more sensitive to color differences than absolute color values.
- We thus try to find image u that preserves gradients difference:
  $\min_u \int_\Omega \|\boldsymbol{\nabla} u - \boldsymbol{\nabla} b\|_2^2\ dx$
- This leads to the equation: $\boldsymbol{\Delta} u = \boldsymbol{\Delta} b$ in Ω and $u = f$ on $\partial\Omega$. Its a PDE called the Poisson equation with Dirichlet boundary conditions.

Figure 2.2: Poisson editing components: from left to right we have f, b and Ω.

2.3.2 Poisson editing using guidance

Guided Interpolation In this method of image editing, a guided interpolation framework is proposed, with the guidance being specified by the user. Figure(2.3) shows an unknown function u interpolates in domain Ω the destination function f, under guidance of vector field $\mathbf{d}$, which might be or not the gradient field of a source function b. For poisson editting. The vector field is indeed chosen as the gradient of b.

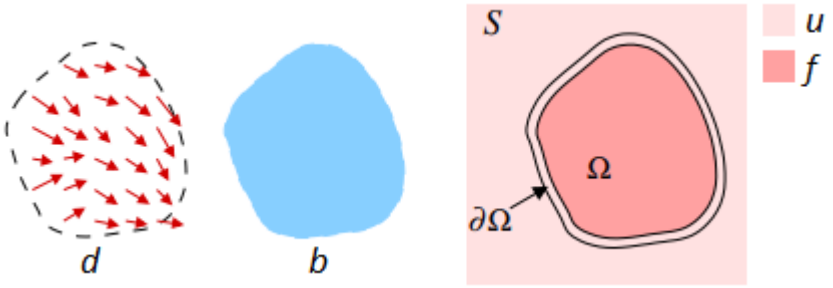


Figure 2.3: Guided interpolation notation

We are going to take a look at image interpolation using a guidance vector field. As it is enough to solve the interpolation problem for each color component separately, Figure (2.2) illustrates the notations. Let S, a closed subset of $\mathbb{R}^2$, be the image definition domain, and let Ω be a closed subset of S with boundary $\partial\Omega$.
Let f be a known scalar function defined over S minus the interior Ω of and let u be an unknown scalar function defined over the interior of Ω. Finally, let $\mathbf{d}$ be a vector field defined over Ω. The simplest interpolant u of f over is the membrane interpolant defined as the solution of the minimization problem:

$$\min_u \int_\Omega |\boldsymbol{\nabla} u|^2 d\Omega \quad \text{with} \quad u\big|_{\partial\Omega} = f\big|_{\partial\Omega} \tag{2.4}$$

The minimizer must satisfy the associated Euler-Lagrange equation over Ω:

$$\boldsymbol{\Delta} u = 0 \quad \text{with} \quad u\big|_{\partial\Omega} = f\big|_{\partial\Omega} \tag{2.5}$$

This simple method produces an unsatisfactory result, as it does nothing but solve the Laplace equation in Ω which produces blurred images, see Figure (2.4).
This can be overcome in a variety of ways. One way is to use a guidance field, a vector field we note $\mathbf{d}$ used in an extended version of the minimization problem (2.4) above:

$$\min_u \int_\Omega |\nabla u - \mathbf{d}|^2 d\Omega \quad \text{with} \quad u\big|_{\partial\Omega} = f\big|_{\partial\Omega} \tag{2.6}$$

whose solution is the unique solution of the following Poisson equation with Dirichlet boundary conditions, over Ω we have:

$$\boldsymbol{\Delta} u = \mathbf{div}(\mathbf{d}) \quad \text{with} \quad u\big|_{\partial\Omega} = f\big|_{\partial\Omega} \tag{2.7}$$

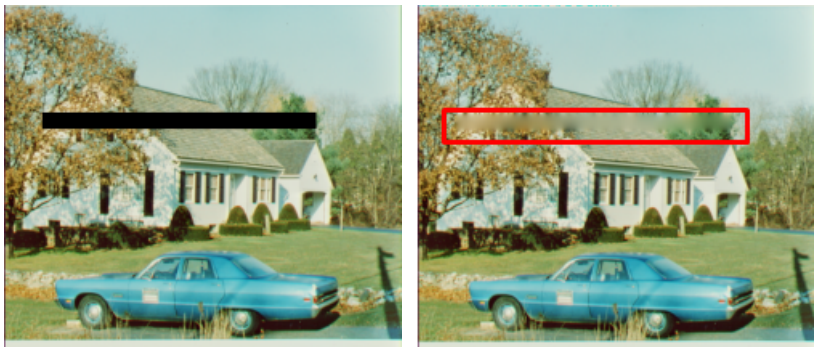


Figure 2.4: A simple case of Poisson editing

When the guidance field $\mathbf{d}$ is conservative, meaning it is the gradient of some function b, a helpful alternative way of understanding what Poisson interpolation does is to define the correction function $\tilde{f}$ on Ω such as $u = b + \tilde{f}$.

The Poisson equation (2.7) then becomes the following Laplace equation with boundary conditions, we have over Ω:

$$\Delta \tilde{f} = 0 \quad \text{with} \quad \tilde{f}\big|_{\partial\Omega} = (f - b)\big|_{\partial\Omega}. \tag{2.8}$$

Therefore, inside Ω. The additive correction $\tilde{f}$ is a membrane interpolant of the mismatch $(f - b)$ between the source and the destination along the boundary $\partial\Omega$. This particular instance of guided interpolation is used for seamless cloning in the next subsection.

2.3.3 Seamless cloning

Importing gradients The basic choice for the guidance field $\mathbf{d}$ is a gradient field taken directly from a source image. Denoting by b this source image, the interpolation is performed under the guidance of:

$$\mathbf{d} = \nabla b, \tag{2.9}$$

and (2.7) now reads over Ω :

$$\mathbf{\Delta} u = \mathbf{\Delta} b \quad \text{over } \Omega \quad \text{with} \quad u\big|_{\partial\Omega} = f\big|_{\partial\Omega}. \tag{2.10}$$

The seamless cloning tool thus obtained ensures the compliance of source and destination

Figure 2.5: Concealment of objects by importing a piece of the background.

boundaries. It can be used to conceal undesirable image features or to insert new elements in an image.

2.4 Image Osmosis

Osmosis filters [25] are based on drift–diffusion processes using a *drift vector*. They offer nontrivial steady states with a number of interesting applications. The drift vector field introduces an active transport mechanism in the diffusion model. Apart from the average grey value, the steady state is fully determined by the drift vector field.

2.4.1 Continuous Linear Osmosis Filtering

We consider a rectangular image domain $\Omega \in \mathbb{R}^2$ with boundary $\partial\Omega$, and a positive greyscale image $f : \Omega \to \mathbb{R}_+$. Moreover, assume we are given some *drift vector field* $\mathbf{d} : \Omega \to \mathbb{R}^2$. Then a (linear) *Osmosis filter* computes a processed version $u(x,t)$ of $f(x)$ by solving the drift-diffusion PDE :

$$\partial_t u = \boldsymbol{\Delta} u - \operatorname{div}(\mathbf{d}u) \quad \text{on } \Omega \times (0, \mathrm{T}], \tag{2.11}$$

With f as initial condition,

$$u(x,0) = f(x) \quad \text{on } \Omega, \tag{2.12}$$

and homogeneous Neumann boundary conditions.

$$\langle \nabla u - \mathbf{d}u, \mathbf{n} \rangle = 0 \quad \text{on } \partial\Omega \times (0, \mathrm{T}]. \tag{2.13}$$

Here $\langle ., . \rangle$ denotes the Euclidean inner product, and $\mathbf{n}$ is the outer normal vector to the image boundary $\partial\Omega$.

When setting the drift vector to satisfy $\mathbf{d} := \ln(\boldsymbol{\nabla} v) = \frac{\boldsymbol{\nabla} v}{v}$ for a given reference image $v > 0$, then the steady state equation :

$$\Delta u - \operatorname{div}(\mathbf{d}u) = 0 \tag{2.14}$$

is equivalent to the Euler-Lagrange equation of the energy functional

$$E(u) = \int_\Omega v(x) \left| \boldsymbol{\nabla} \left(\frac{u(x)}{v(x)} \right) \right|^2 dx \tag{2.15}$$

Indeed, the energy functional (2.15) can be rewritten as

$$E(u) = \int_\Omega F(u, \boldsymbol{\nabla} u)\ dx. \tag{2.16}$$

Using the following gradient property ($\nabla(\frac{a}{b}) = \frac{b\nabla a - a\nabla b}{b^2}$) on F we get

$$F(u, \boldsymbol{\nabla} u) = \frac{|v\boldsymbol{\nabla} u - u\boldsymbol{\nabla} v|^2}{v^3} \tag{2.17}$$

which can be simplified to

$$F(u, \boldsymbol{\nabla} u) = \frac{v^2 \nabla u.\nabla u - 2u.v\nabla u.\nabla v + u^2 \nabla v.\nabla v}{v^3}$$

with this abuse of notation

$$\nabla u.\nabla v = \nabla u^{\intercal}.\nabla v = \nabla u.\nabla v^{\intercal} = \nabla v.\nabla u$$

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

$$F_u - \partial_x F_{u_x} - \partial_y F_{u_y} = 0 \tag{2.18}$$

Now we compute the components of (2.18):

$$F_u = \frac{-2v.\nabla u.\nabla v + 2u.\nabla v.\nabla v}{v^3}$$

$$F_{u_x} = \frac{2v^2.u_x - 2u.v.v_x}{v^3}, F_{u_y} = \frac{2v^2.u_y - 2u.v.v_y}{v^3}$$

We know that Euler-Lagrange equation can be written as $F_u - \text{div}(F_{u_x}, F_{u_y}) = 0$ so (2.18) becomes :

$$\frac{-2v.\nabla u.\nabla v + 2u.\nabla v.\nabla v}{v^3} - 2\text{div}\left(v.\frac{v.\nabla u - u.\nabla v}{v^3}\right) = 0 \tag{2.19}$$

Using this divergence identity $\big(\text{div}(A\vec{B}) = (\nabla A).\vec{B} + A.\ \text{div}(\vec{B})\big)$ on the second part of (2.19) to get :

$$-\frac{2v.\nabla u.\nabla v}{v^3} + \frac{2u.\nabla v.\nabla v}{v^3} - 2\nabla v.\big(\frac{v.\nabla u - u.\nabla v}{v^3}\big) - 2v.\text{div}\ \big(\frac{v.\nabla u - u.\nabla v}{v^3}\big) = 0$$

$$\implies \underbrace{-\frac{2v.\nabla u.\nabla v}{v^3}}_{(1)} + \underbrace{\frac{2u.\nabla v.\nabla v}{v^3}}_{(2)} - \underbrace{\frac{2v.\nabla v.\nabla u}{v^3}}_{(3)} + \underbrace{\frac{2u.\nabla v.\nabla v}{v^3}}_{(4)} - 2v.\text{div}\ \big(\frac{v.\nabla u - u.\nabla v}{v^3}\big) = 0$$

Combining (1)+(3) with (2)+(4) we finally get:

$$-2v.\text{div}\ \big(\frac{v.\nabla u - u.\nabla v}{v^3}\big) - \frac{4\nabla v(v.\nabla u - u.\nabla v)}{v^3} = 0 \tag{2.20}$$

By using this equation which we will prove later

$$\text{div}\ \left(v.\nabla\left(\frac{u}{v}\right)\right) = v^2.\text{div}\ \left(\frac{v.\nabla u - u.\nabla v}{v^3}\right) + \frac{2\nabla v(v.\nabla u - u.\nabla v)}{v^2} \tag{2.21}$$

we can see that equation (2.21) multiplied by $\big(\frac{-2}{v}\big)$ is equal to our Euler-Lagrange equation (2.20), therefore (2.20) becomes:

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

$$\implies \text{div}\ \left(v.\nabla\left(\frac{u}{v}\right)\right) = 0$$

and using again the gradient property ($\nabla(\frac{a}{b}) = \frac{b\nabla a - a\nabla b}{b^2}$) we get

$$\text{div}\left(v.\nabla\left(\frac{u}{v}\right)\right) = \text{div}\left(\frac{v.\nabla u - u.\nabla v}{v}\right)$$

Because the divergence operator is linear and $\text{div}(\nabla u) = \Delta u$, and by recalling that $\mathbf{d}:= \frac{\boldsymbol{\nabla v}}{v}$ it results in (2.14) :

$$\boldsymbol{\Delta} u - \text{div}(\mathbf{d}u) = 0$$

The equation (2.21) we used can be proven by putting $\text{div}\left(v.\nabla\left(\frac{u}{v}\right)\right) = \text{div}\left(v^2\left(\frac{v.\nabla u - u.\nabla v}{v^3}\right)\right)$ and applying the divergence identity $\left(\text{div}(A\vec{B}) = (\nabla A).\vec{B} + A.\ \text{div}(\vec{B})\right)$. Thus we got the steady state (2.14).

Straightforward computations also show that one obtains (2.13) as boundary condition on $\partial\Omega$. Its also clear that v (and k.v) is a solution of the steady state (2.14) because

$$\boldsymbol{\Delta} v - \text{div}(\frac{\boldsymbol{\nabla v}}{v}.v) = \boldsymbol{\Delta} v - \boldsymbol{\Delta} v = 0$$

The solution of the steady state is a function w where $w = k.v$ and $k = \frac{\mu_f}{\mu_v}$ (μ_f and μ_v are the average of f and v over Ω respectively), its easly proven :

$$k\mu_v = \frac{1}{|\Omega|}\int_\Omega k.v(x)dx = \frac{1}{|\Omega|}\int_\Omega w(x)dx = \frac{1}{|\Omega|}\int_\Omega f(x)dx = \mu_f.$$

Chapter 3

The joint variational Osmosis model

3.1 Inputs of the model

Our main objective is to smoothly fuse two images, one image for the foreground noted f and one background image b. To force the fusion in the boundary of the foreground image $A \subset \Omega$, suh as A represents the domain of f which we want to fuse (its a face in our first example). We use an α-map associated to A. Let $\Omega = A_f \cup A_b \cup A_{mix}$, with A_f be a region where the foreground data is desired, ($u(x) = f(x)$ on A_f), A_b where the background data is desired ($u(x) = b(x)$ on A_b), and A_{mix} the uncertain zone, i.e. where $f(x)$ and $b(x)$ are mixed.

We can choose the α-map as follows:

$$\alpha(x) = \begin{cases} 1 & \text{if } x \in A_f \\ \alpha \in (0,1) & \text{if } x \in A_{mix} \\ 0 & \text{if } x \in A_b \end{cases} \tag{3.1}$$

An alternative choice of the α-map is defined as the convolution of the indicator function with a compactly supported kernel $\varphi \in C_c^\infty(\Omega)$ as follows:

$$\alpha(x) = (\varphi * I_d)(x). \tag{3.2}$$

and our indicator function is defined as

$$I_d(x) = \begin{cases} 0 & \text{if } x \notin A \\ 1/2 & \text{if } x \in \partial A \\ 1 & \text{if } x \in A \end{cases} \tag{3.3}$$

This ensures a smooth transition from 0 to 1 in the boundary ∂A.
However, in this work we use a manual user-supplied α-maps to identify the source images f and b.

The following Figure (3.1) gives an example of each image with a user made α-map and identical size f and b images. From the left to the right. Foreground f, background b and α-map.

Figure 3.1: Portraits of Leonhard Euler and Louis Lagrange

3.2 The new fusion term

Based on Osmosis varriational energy montioned in the previous chapter, we propose the new fusion model witch aims to smoothly fuse the information contained in f and b in a new image v.

The first change we make is for the drift vector in (2.11), by choosing v whose associated drift contains informations from both f and b.
Noted $\tilde{\mathbf{d}}$ such as :

$$\tilde{\mathbf{d}} = \boldsymbol{\nabla}\, log(f^{\alpha}.\, b^{1-\alpha}). \tag{3.4}$$

Note that if $\alpha(x) \equiv \alpha \in [0, 1]$ then this choice of $\tilde{\mathbf{d}}$ is simply the convex combination of the drifts associated to f and b.

And to enforce v to stay locally close to such reference image $f^{\alpha}.b^{1-\alpha}$, a penalisation term is added, weighted by a parameter $\mu > 0$.

Combining this altogether, the osmosis-driven fusion term, witch now depends on u and v reads :

$$\mathcal{O}(u,v) = \frac{1}{2}\int_{\Omega} v(x) \left| \boldsymbol{\nabla}\left(\frac{u(x)}{v(x)}\right)\right|^2 dx + \frac{\mu}{2} \left\| v - (f^{\alpha}.\ b^{1-\alpha})\right\|_2^2 \tag{3.5}$$

While the original Osmosis fusion term reads:

$$E(u) = \int_{\Omega} v(x) \left\| \boldsymbol{\nabla}\left(\frac{u(x)}{v(x)}\right)\right\|_W^2 dx \tag{3.6}$$

The new proposed model (3.5) combines the original Osmosis term (3.6) with a quadratic term which acts as a fidelity term that forces v to stay consistent with the foreground and background images f and b.

NB: Although the quadratic term has a convex combination, the general term $\mathcal{O}(u,v)$ is non-convex. Its also important to montion that (3.5) is the base fusion term, however its not final. Additional terms will be added to control the fusion in the next part.

3.3 Fidelity and regularisation

After we established the fusion term, we add two further terms. The first term is a fidelity term imposing u to stay close to f in the foreground, for each $x \in \Omega$ and $\alpha(x) > 0$:

$$D(u) = \frac{1}{2}\int_{\Omega} \alpha(x)\big(u(x) - f(x)\big)^2 dx \tag{3.7}$$

Note that we can write D along with the L^2 norm in (3.5) because the integral operator is linear, but the term is written separately to highlight the importance of the quadratic regularisation term. Therefore this decomposition will be respected in numerical schemes.

Whilst the combination of these two terms gives good results, an inhancing term is added for further regularisation acting on v denoted as *R(v)* such as:

$$R(v) = \int_{\Omega} |\boldsymbol{\nabla} v(x)| dx \tag{3.8}$$

This is called **Total variation regularisation**. We could alternatively use smooth regularisation function like **Tikhonov**-type regularisation which uses L^2 norme:

$$R(v) = \frac{1}{2}\|\boldsymbol{\nabla} v(x)\|^2 = \frac{1}{2}\int_\Omega |\boldsymbol{\nabla} v(x)|^2 dx \tag{3.9}$$

These three terms are merged in the final proposed model along with two more weights γ and ϵ. By varring them aswell as the weight μ of $\mathcal{O}(u, v)$, the term reads:

$$\min_{u,v} \mathcal{E}(u, v) = \mathcal{O}(u, v) + \gamma D(u) + \epsilon R(v). \tag{3.10}$$

These weights as well as the initial condition play a huge role in the behaviour of the final images in a way that we will discover in the numerical part of this thesis. The solution of the model is computed as the minimising of this energy functional.
We summarize our framework with this flowchart: Note that kNN matting is a method

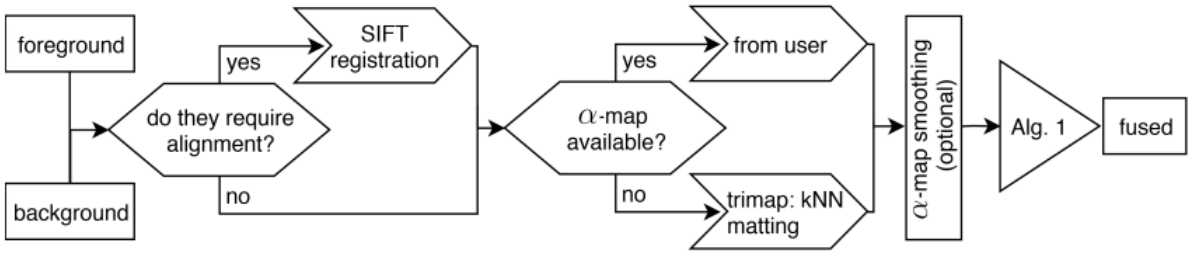


Figure 3.2: Flowchart of the proposed model for image fusion

to automaticly compute the α-map and (Alg. 1) is the algorithm which we will use to fuse the faces of f and b. Now we move to minimising the energy whilst we don't have the luxury we had in chapter (2) of using the *Euler-Lagrange* equation (2.2) which is the standard in most variational models.

3.4 Numerical resolution: The iPiano algorithm

The *Inertial Proximal Algorithm for Non-convex Optimization* [18] is an algorithm proposed for solving a minimization problem composed of a differentiable (possibly non-convex) and a convex (possibly non-differentiable) function. We have established that our Osmosis energy we ought to minimize (3.10) is composed of a differentiable non-convex term and convex non-differentiable term. That makes our approach ideal. Let's recall our general problem (**??**)

$$\begin{cases} \min : F(x) = f(x) + g(x) \\ \quad \text{st} : x \in \mathcal{X} \end{cases}$$

where g is a convex (possibly non-smooth) and f is a smooth (possibly non-convex) function. Arbitrarily, the main update of our algorithm is

$$x_{t+1} = \mathbf{prox}_{\eta g}\big(x_t - \eta \nabla f(x_t)\big) + \zeta(x_t - x_{t-1}) \tag{3.11}$$

We recall that the *proximal operator* $\mathbf{prox}_h : \mathbb{R}^n \to \mathbb{R}^n$ of h is defined by $\mathbf{prox}_h(v) = \arg\min_x \big(h(x) + (1/2)\|x - v\|_2^2\big)$, where $\|.\|_2^2$ is the usual Euclidean norm.

3.5 Computing algorithm components

The iterative scheme of (3.10) for $t \geq 0$ reads

$$\begin{cases} u_{t+1} = \mathbf{prox}_{\eta_{1,t}\gamma D}(u_t - \eta_{1,t}\partial_u \mathcal{O}(u_t, v_t)) + \zeta_1(u_t - u_{t-1}), \\ v_{t+1} = \mathbf{prox}_{\eta_{2,t}\epsilon R}(v_t - \eta_{2,t}\partial_v \mathcal{O}(u_t, v_t)) + \zeta_2(v_t - v_{t-1}). \end{cases} \tag{3.12}$$

3.5.1 Gradient of Osmosis term

The gradient of this term is given by $\boldsymbol{\nabla}\mathcal{O}(u, v) = (\partial_u \mathcal{O}(u, v), \partial_v \mathcal{O}(u, v))^\top$, the derivatives of $\mathcal{O}(u, v)$ are computed next.

- Deriving $\partial_v \mathcal{O}(u, v)$

$$\partial_v \mathcal{O}(u, v) = \frac{\partial}{\partial_v}\left(\frac{1}{2}\int_\Omega v(x)\left|\boldsymbol{\nabla}\left(\frac{u(x)}{v(x)}\right)\right|^2 dx\right) + \frac{\partial}{\partial_v}\left(\frac{\mu}{2}\left\|v - (f^\alpha .\, b^{1-\alpha})\right\|_2^2\right).$$

  Using the same gradient property ($\nabla(\frac{a}{b}) = \frac{b\nabla a - a\nabla b}{b^2}$) we previously used in (2.17) to get

$$\partial_v \mathcal{O}(u, v) = \frac{\partial}{\partial_v}\frac{1}{2}\left(\int_\Omega \frac{|v\boldsymbol{\nabla}u - u\boldsymbol{\nabla}v|^2}{v^3} dx\right) + \frac{\partial}{\partial_v}\left(\frac{\mu}{2}\left\|v - (f^\alpha .\, b^{1-\alpha})\right\|_2^2\right),$$

  with the same abuse of notation

$$\nabla u.\nabla v = \nabla u^\intercal.\nabla v = \nabla u.\nabla v^\intercal = \nabla v.\nabla u.$$

We get

$$\partial_v \mathcal{O}(u,v) = \underbrace{\frac{1}{2}\int_\Omega \frac{\partial}{\partial_v}\left(\frac{v^2\nabla u.\nabla u - 2u.v\nabla u.\nabla v + u^2\nabla v.\nabla v}{v^3}\right)dx}_{(a)+(b)+(c)}$$
$$+ \frac{\partial}{\partial_v}\left(\frac{\mu}{2}\ \left\|v - (f^\alpha.\ b^{1-\alpha})\right\|_2^2\right),$$
$$(a) = \frac{\partial}{\partial_v}\frac{1}{2}\int_\Omega \frac{\boldsymbol{\nabla} u.\boldsymbol{\nabla} u}{v}dx = -\frac{1}{2}\int_\Omega \frac{\boldsymbol{\nabla} u.\boldsymbol{\nabla} u}{v^2}dx = -\frac{1}{2v^2}|\boldsymbol{\nabla} u|^2,$$
$$(b) = -\frac{1}{2}\frac{\partial}{\partial_v}\int_\Omega \frac{2u}{v^2}\boldsymbol{\nabla} u.\boldsymbol{\nabla} v\ dx$$
$$= -\frac{1}{2}\int_\Omega \frac{\partial}{\partial_v}\left(\frac{2u}{v^2}\boldsymbol{\nabla} u.\boldsymbol{\nabla} v\right) + \frac{2u}{v^2}\frac{\partial}{\partial_v}\left(\boldsymbol{\nabla} u.\boldsymbol{\nabla} v\right)\ dx$$
$$= -\frac{1}{2}\int_\Omega -\frac{4u}{v^3}\boldsymbol{\nabla} u.\boldsymbol{\nabla} v + \frac{2u}{v^2}\boldsymbol{\nabla} u.\boldsymbol{\nabla}\boldsymbol{I}\ dx \quad \text{(Neumann B.C and Green furmula)}$$
$$= \int_\Omega \frac{2u}{v^3}\boldsymbol{\nabla} u.\boldsymbol{\nabla} v + \mathrm{div}\left(\frac{u}{v^2}\boldsymbol{\nabla} u\right)\ dx,$$
$$(c) = \frac{1}{2}\frac{\partial}{\partial_v}\int_\Omega \frac{u^2}{v^3}\boldsymbol{\nabla} v.\boldsymbol{\nabla} v\ dx$$
$$= \frac{1}{2}\int_\Omega \frac{\partial}{\partial_v}\left(\frac{u^2}{v^3}\right)\boldsymbol{\nabla} v.\boldsymbol{\nabla} v + \frac{u^2}{v^3}\frac{\partial}{\partial_v}\left(\boldsymbol{\nabla} v.\boldsymbol{\nabla} v\right)\ dx$$
$$= \int_\Omega -\frac{3u^2}{2v^4}\boldsymbol{\nabla} v.\boldsymbol{\nabla} v - \frac{v}{2}\mathrm{div}\left(\frac{u^2}{v^3}\right)\ dx.$$

Finally, we have

$$\partial_v \mathcal{O}(u,v) = -\frac{1}{2v^2}|\boldsymbol{\nabla} u|^2 + \frac{2u}{v^3}\boldsymbol{\nabla} u.\boldsymbol{\nabla} v + \mathrm{div}\left(\frac{u}{v^2}\boldsymbol{\nabla} u\right)$$
$$- \frac{3u^2}{2v^4}|\boldsymbol{\nabla} v|^2 - \frac{v}{2}\mathrm{div}\left(\frac{u^2}{v^3}\right)$$
$$+ \mu(v - f^\alpha b^{1-\alpha}).$$

The same way we have

$$\partial_u \mathcal{O}(u,v) = -\frac{(\boldsymbol{\Delta} u - \mathrm{div}(\tilde{\mathbf{d}}u))}{v}$$

where $\tilde{\mathbf{d}}$ is the drift vector from (3.4), i.e $\tilde{\mathbf{d}} = \boldsymbol{\nabla}\ log(v) = \boldsymbol{\nabla}\ log(f^\alpha.\ b^{1-\alpha})$.

3.5.2 Proximal operator of the fidelity and regularisation terms

The iPiano scheme (3.12) requires the computation of two proximal operators.

- Proximal operator of $D(u)$

$$\mathbf{prox}_{\eta_{1,t}\gamma D}(x) = \arg\min_u \frac{\gamma}{2}\big(\|\sqrt{\alpha}(u-f)\|_2^2\big) + \frac{1}{2\eta_{1,t}}(\|u-x\|_2^2).$$

From the optimality condition, u is the optimal solution if and only if

$$0 \in \nabla\big(\frac{\gamma}{2}\big(\|\sqrt{\alpha}(u-f)\|_2^2\big)\big) + \nabla\big(\frac{1}{2\eta_{1,t}}\|u-x\|_2^2\big) \iff 0 \in \gamma(\sqrt{\alpha}(u-f)) + \frac{1}{\eta_{1,t}}(u-x)$$

then optimal solution u^* is obtained by

$$\gamma(\sqrt{\alpha}(u^*-f)) + \frac{1}{\eta_{1,t}}(u^*-x) = 0 \iff u^* = \big(\gamma\alpha f - \frac{1}{\eta_{1,t}}\big).\big(\gamma\alpha + \frac{1}{\eta_{1,t}}x\big)^{-1}$$

and finally

$$\mathbf{prox}_{\eta_{1,t}\gamma D}(x) = \big(\gamma\alpha f - \frac{1}{\eta_{1,t}}\big).\big(\gamma\alpha + \frac{1}{\eta_{1,t}}x\big)^{-1}.$$

- Proximal operator of $R(v)$

The Total Variation regularization ($\|\nabla v\|_1$) has a singularity when $v = 0$, one way arround this is the classical choice

$$\|\nabla v\|_1 = \int_\Omega \sqrt{\delta^2 + (u_x)^2 + (u_y)^2}\; dx\ ,$$

so when deriving there would be no singularity when the denominator equals 0, as we always have the small positive constant ($0 < \delta \leq 1$) we added.
However, for numerical stability a better choice was given in the work of Chambolle and Pock [3] where it is better to consider a smooth version of the TV. Which is obtained by replacing TV with its Huberised version (Huber regularization). Let

$$\mathcal{H}_\delta(x) = \begin{cases} \frac{x^2}{2\delta} & \text{if } x \leq \delta \\ |x| - \frac{\delta}{2} & \text{else.} \end{cases} \tag{3.13}$$

This promotes quadratic regularisation for areas where $\|\nabla v\|_1 \leq \delta$ and L^1 regularization when $\|\nabla v\|_1 > \delta$. The proximal step is given by

$$\mathbf{prox}_{\eta_{2,t}\epsilon R}(x) = \arg\min_v\ \epsilon\mathcal{H}_\delta(\boldsymbol{\nabla} v) + \frac{1}{2\eta_{2,t}}(\|v-x\|_2^2).$$

3.6 Algorithm

Applying our model $\min_{u,v} \mathcal{E}(u,v) := \mathcal{O}(u,v) + \gamma D(u) + \epsilon R(v)$ to fuse two given images f and b for different purposes. Another input we need is the $\alpha-$map we defined in chapter 3. The resulting algorithm dubbed **Alg 1** is given by

Algorithm 1 *Iterative scheme for the joint osmosis model* (Alg 1)

- **Input** image f, b and the $\alpha-$map;
- **Output** fused image u;
- **Parameters** tolerance (tol), maximum number of iterations (max-it), weights (μ, γ, ϵ) and step-sizes $\eta_{1,0} = 0.99(1-2\zeta_1)/\beta_{1,0}$.
  $\eta_{2,0} = 0.99(1-2\zeta_2)/\beta_{2,0}$ and $\zeta_1 = \zeta_2 = 0.4$, $\beta_{1,0} = \beta_{2,0} = 1$, algorithm's function name (fun-iPiano).

Function fun-iPiano
Require: $u_0 = f$ and $v_0 = f^{\alpha} b^{-1-\alpha}$ ▷ Initialisation
  for $t = 0, ...,$max-it **do**
    $u_0^* = u_t - \eta_{1,t}\partial_u \mathcal{O}(u_t, v_t) + \zeta_1(u_t - u_{t-1})$
    $v_0^* = v_t - \eta_{2,t}\partial_v \mathcal{O}(u_t, v_t) + \zeta_2(v_t - v_{t-1})$
    for $t = 0, ...,$max-it **do**
      $p_1 = \mathbf{prox}_{\eta_{1,t}\gamma D}(u_0^*)$;
      $p_2 = \mathbf{prox}_{\eta_{2,t}\epsilon R}(v_0^*)$;
      gap-u$=\mathcal{O}(p_1, v_t) - \mathcal{O}(u_t, v_t) - < \partial_u \mathcal{O}(u_t, v_t), (p_1 - u_t) > -(\beta_{1,t}/2)\|p_1 - u_t\|_2^2$;
      gap-v$=\mathcal{O}(u_t, p_2) - \mathcal{O}(u_t, v_t) - < \partial_v \mathcal{O}(u_t, v_t), (p_2 - v_t) > -(\beta_{2,t}/2)\|p_2 - v_t\|_2^2$;
      if gap-u < 0 and **gap-v** < 0 **then**
        $u_{t+1} = p_1$; $v_{t+1} = p_2$; **break**; ▷ accept and update the variabl
      end if
      update $\eta_{1,t}, \eta_{2,t}, \beta_{1,t}, \beta_{2,t}$
    end for
    if $|\mathcal{E}(u_{t+1}, v_{t+1}) - \mathcal{E}(u_t - v_t)|/|\mathcal{E}(u_{t+1}, v_{t+1})|$ <tol **then**
      break; ▷ Exit condition
    end if
  end for

This algorithm is based on a trial and error strategy, its easier to code as the gradient can be numerically computed explicitly. Thus, one can try to estimate a numerical Lipschitz constant and then check if that is a good estimate or not (and in case adapt / reject).

3.6.1 Results

Before we look at the different applications of (Alg 1) to imaging problems, let's check our choice of the weight. For the Huber term we choose the constant $\delta = 0.005$, while the exit condition could be either max-it=10000 iterations or the relative error on the energy $|\mathcal{E}(u_{t+1}, v_{t+1}) - \mathcal{E}(u_t - v_t)|/|\mathcal{E}(u_{t+1}, v_{t+1})|$ is less then a given value (tol). A a reasonable choice for μ according to tests is the value 100 whilst 0.1 and 1 are taken for γ and ϵ respectively. The weight μ is taken very large with respect to γ and ϵ to force the effect of regularisation and preserve structural information at every iteration. Last, ϵ controls the amount of the foreground information to be preserved.

Results in face fusion

Using the inputs from Figure (3.1). We can plot the result and compare it with previous methods used in chapter 2, the Osmosis and Poisson models. The proposed joint-variational mode with (ϵ, μ, γ)=(0.01, 10, 1) shows similar result to the Osmosis model but has better color contrast. From left to right, Seamless Poisson editing, Osmosis, Joint-variational Osmosis.

Figure 3.3: Joint-variational Osmosis

Results in editting and cloning

For this application we use the inputs of Figure(5.2), the result is shown in Figure(3.5) with (ϵ, μ, γ)=(0.5, 100, 1).

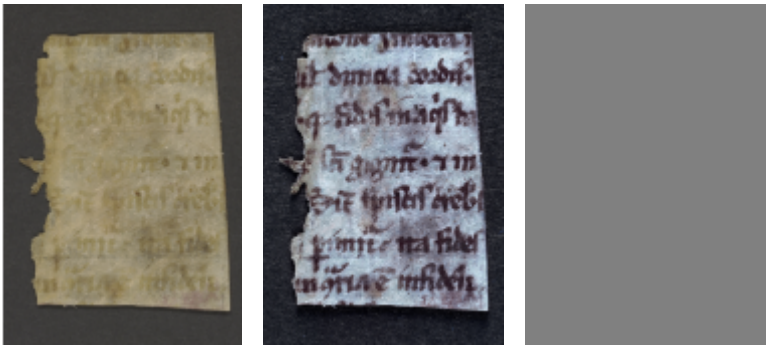
(a) Foreground (b) Background (c) $\alpha-$map

Figure 3.4: Inputs of the fusion

Figure 3.5: Seamless cloning with joint-variational Osmosis

Cultural heritage

One way to improve old decayed cultural heritage images with invisible text is to take an infrared image and use this model to fuse them. Figure (5.4) shows a manuscript from Biblioteca Capitolare (Italy), where the text is only visible after infrared inspection. Applying this model with a constant $\alpha-$map and (ϵ, μ, γ)=(0.1, 10, 0.1) gives a good result.



(a) Original (b) Infrared (c) $\alpha-$map=0.5

Figure 3.6: Inputs of the fusion

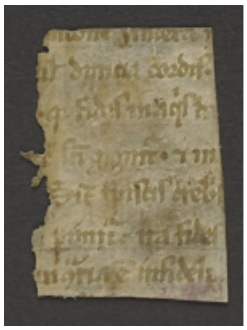


Figure 3.7: Seamless cloning with joint-variational Osmosis

Error

We will use two error mesures to see the difference between input image f and output image u, for the first mesure let's recall the definition of the e Geometric Chromaticity Mean of an image u. $\mathbf{GCM}(u) = \sqrt[3]{u_R u_G u_B}$ with u_R, u_G, u_B the values of u in each color channel.
Therefore, the so called chromaticity error between two given images u^1 and u^2 is defined as

$$\text{err}(u^1, u^2) = \left| \frac{u^1}{\mathbf{GCM}(u^1)} - \frac{u^2}{\mathbf{GCM}(u^2)} \right|. \tag{3.14}$$

The second error mesure is called Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [14]. We use the two errors to compare the result of this model with the models from chapter 2.

error table (smaller is better)		
Models	$\|\text{err}(u, f)\|$	BRISQUE [14]
Poisson [20]	0.0111	30.2
Osmosis [25]	0.0452	30.2
This model	**0.0062**	**30.1**

Using these two types of errors metrics, we can see that the joint variational Osmosis model result shows the most consistency with the original image f while imporving it.
Ploting error (3.14) of the input image f from Figure (3.1) and the output image u of our model from Figure (5.1) gives the following result.

Figure 3.8: Error (3.14)

Chapter 4

Conclusion

In this thesis, we have looked at image fusion using variational methods. We started with the most basic and important model called Poisson editing, then we studied the Osmosis model presented in [25] which is s a novel concept for visual computing. Its an important process in nature that is as occurring as heat diffusion but widely ignored. It creates nonconstant steady states that can be controlled in a transparent way by the drift vector field. This offers many application areas, we have seen mainly face fusion but it can be expanded to other applications like shadow removal [2] and image editing. Afterward, we focused our work on Parisotto et al. [19]' modification. It took this consept further by proposing a convex combination for the drift vector proposed in [25] and adding two terms of fidelity and regularization, the later being the non-smooth Total Variation regularization. The resulting model was over all non convex. Mininmizing it was not possible using conventional methods due to the non convexity barrier, this barrier was broken with the exceptional work of Ochs et al. [18] and the algorithm iPiano that allowed us to minimize our model. Finally, we saw some real life results in face fusion, cloning and cultural heritage.

Refrences

1) *Adrian Galdran, David Pardo, Artzai Pic´on, and Aitor Alvarez-Gila. Automatic red- channel underwater image restoration. Journal of Visual Communication and Image Representation, 26:132–145, 2015.*
2) *Antonin Chambolle and Thomas Pock. An introduction to continuous optimization for imaging. Acta Numerica, 25:161–319, 2016.*
3) *Arkadij Semenoviˇc Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.*
4) *Boris T Polyak. Gradient methods for the minimisation of functionals. USSR Com- putational Mathematics and Mathematical Physics, 3(4):864–878, 1963.*
5) *Codruta O Ancuti, Cosmin Ancuti, Chris Hermans, and Philippe Bekaert. Image and video decolorization by fusion. pages 79–92, 2010.*
6) *Dmitrii Konstantinovich Faddeev and Vera Nikolaevna Faddeeva. Computational methods of linear algebra. Zapiski Nauchnykh Seminarov POMI, 54:3–228, 1975.*
7) *Farong Gao, Kai Wang, Zhangyi Yang, Yejian Wang, and Qizhong Zhang. Underwater image enhancement based on local contrast correction and multi-scale fusion. Journal of Marine Science and Engineering, 9(2):225, 2021.*
8) *Franck Iutzeler and J´erˆome Malick. On the proximal gradient algorithm with al-ternated inertia. Journal of Optimization Theory and Applications, 176(3):688–710, 2018.*
9) *Huimin Lu, Yujie Li, Yudong Zhang, Min Chen, Seiichi Serikawa, and Hyoungseop Kim. Underwater optical image processing: a comprehensive review. Mobile networks and applications, 22(6):1204–1211, 2017.*
10) *Joachim Weickert, Kai Hagenburg, Michael Breuß, and Oliver Vogel. Linear osmosis models for visual computing. pages 26–39, 2013.*
11) *John Y Chiang and Ying-Ching Chen. Underwater image enhancement by wavelength ompensation and dehazing. IEEE transactions on image processing, 21(4):1756–1769, 2011.*
12) *Jupo Ma, Jinjian Wu, Leida Li, Weisheng Dong, Xuemei Xie, Guangming Shi, and Weisi Lin. Blind image quality assessment with active inference. IEEE Transactions on Image Processing, 30:3650–3663, 2021.*

13) *Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. SIAM Journal on Opti-mization, 26(1):57–95, 2016.*
14) *Naum Z Shor. Subgradient and ε-subgradient methods. In Nondifferentiable Opti-mization and Polynomial Problems, pages 35–70. Springer, 1998.*
15) *Oliver Vogel, Kai Hagenburg, Joachim Weickert, and Simon Setzer. A fully discrete theory for linear osmosis filtering. pages 368–379, 2013.*
16) *Pascal Getreuer. Rudin-osher-fatemi total variation denoising using split bregman. Image Processing On Line, 2:74–95, 2012.*
17) *Patrick L Combettes and Val´erie R Wajs. Signal recovery by proximal forward- backward splitting. Multiscale modeling & simulation, 4(4):1168–1200, 2005.*
18) *Patrick P´erez, Michel Gangnet, and Andrew Blake. Poisson image editing. Pages 313–318, 2003.*
19) *Peter Ochs, Yunjin Chen, Thomas Brox, and Thomas Pock. ipiano: Inertial proximal algorithm for nonconvex optimization. SIAM Journal on Imaging Sciences, 7(2):1388–1419, 2014.*
20) *R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. SIAM journal on control and optimization, 14(5):877–898, 1976.*
21) *S Benalia and Mohammed Hachama. A nonlocal method for image shadow removal. Computers & Mathematics with Applications, 107:95–103, 2022.*
22) *Simone Parisotto, Luca Calatroni, Aur´elie Bugeau, Nicolas Papadakis, and Carola- Bibiane Sch¨onlieb. Variational osmosis for non-linear image fusion. IEEE Transac-tions on Image Processing, 29:5507–5516, 2020.*
23) *SK Zavriev and FV Kostyuk. Heavy-ball method in nonconvex optimization problems. Computational Mathematics and Modeling, 4(4):336–341, 1993.*
24) *Stanley P Frankel. Convergence rates of iterative treatments of partial differential quations. Mathematics of Computation, 4(30):65–75, 1950.*
25) *Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion. pages 382–390, 2007.*
26) *Yurii Nesterov. Introductory lectures on convex optimization: A basic course, vol- ume 87. Springer Science & Business Media, 2003.*