Second order variational models for image texture analysis

Maïtine Bergounioux

Second order variational models for image texture analysis

Maïtine Bergounioux

UMR 7349-MAPMO Fédération Denis Poisson
Université d'Orléans
BP 6759 F-45067 Orleans Cedex 2

Abstract

We present variational models to perform texture analysis and/or extraction for image processing. We focus on second order decomposition models. Variational decomposition models have been studied extensively during the past decades. The most famous one is the Rudin-Osher-Fatemi model.
We first recall most classical first order models . Then we deal with second order ones : we detail the mathematical framework, theoretical models and numerical implementation. We end with two 3D applications. Eventually, an appendix includes the mathematical tools that are used to perfom this study and Matlab© codes are provided.

Keywords: Image processing, texture analysis, variational models, second order models.

Contents

Email address: maitine.bergounioux@univ-orleans.fr (Maïtine Bergounioux)
URL: http://maitinebergounioux.net/PagePro/Accueil.html (Maïtine Bergounioux)

1. Introduction

The question of texture in image processing is an important issue. The definition itself is not clear : some people define it as a random or periodic structure from an image. We rather define it as an inner structure that adds to the image informations which are not necessarily fundamental at a first glimpse. The figure 1.1 of a hut gives an illustration : basic information is given by the contours and image dynamics (one sees a hut in the country). There is no ambiguity. The texture adds a second level of information: the nature of the roof (which gives information on geographical or cultural buildings), nature of the leaves of the tree (winter or summer) and so on. This is comparable to exercises that are proposed to young children studying some language grammar. A complex sentence is given and the pupil has to remove words that are not necessary to understand the basic meaning of the sentence. For example

A small white house, with a red roof is build in a dark oaks forest

becomes

A house is build in a forest.

We just keep the first-level information : everything else is just details, namely texture.

(a) Original Image (b) Cartoon (c) Texture

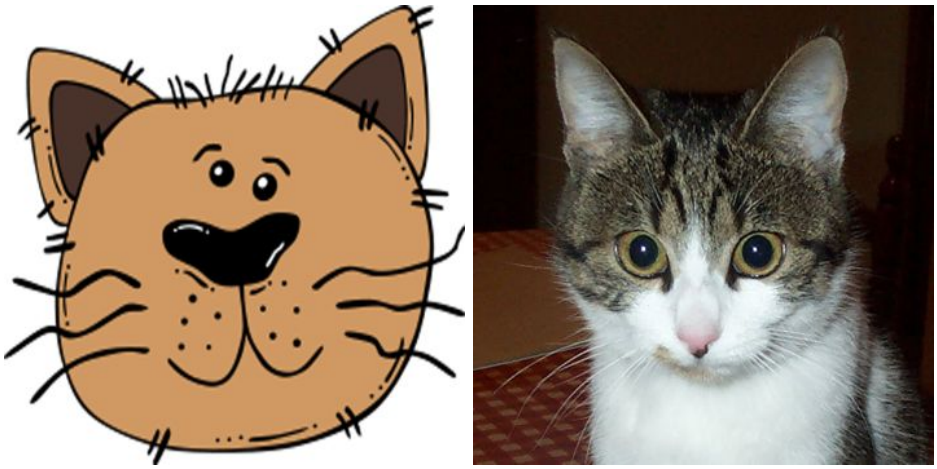
Figure 1.1: Decomposition cartoon + texture

It is now more and more important to take textures into account into applications. This makes the difference between 50's cartoons that did not involve much texture and nowadays animation movies that need these textures for the sake of realistic rendering.

Besides the traditional questions of segmentation (contours and/or regions), denoising or restoration, texture management (identification, synthesis) has become an important issue. One may think, of course of computer graphics (animated movies, "realistic" video games). However, textures often contain important "second-level" information: this is the case in medical imaging (detection of tumors in mammograms, bone abnormalities identification in radiograph, automatic differentiation of tissues in MRI).

There are various techniques to study of textures. Especially noteworthy

- statistical methods as in Khelifi and Jiang (2011), Wen et al. (2011), Karoui et al. (2010), Portilla and Simoncelli (2000), Bar-Joseph et al. (2001) or
- probabilistic ones (Galerne et al. (2011), Grzegorzek (2010), Paget and Longstaff (1998), Mumford and Gidas (2001), Zhu et al. (1998))
- image decomposition methods: one can refer to Gilles and Meyer (2010), Buades et al. (2010), Duval et al. (2010), Shahidi and Moloney (2009), Aubert and Aujol (2005) for example.
- wavelets theory as in Eckley et al. (2010), Ramrishnan and Selvan (2008), Aujol et al. (2003), Peyré (2010), De Bonet (1997), Portilla and Simoncelli (2000) and morphological component analysis Elad et al. (2005), Fadili et al. (2007)
- graph cuts techniques (Kwatra et al. (2003))



(a) Cartoon (b) Image with texture

Figure 1.2: Cartoon versus texture

These techniques are then used in different contexts as, for example

- inpainting (Casaburi et al. (2011), Aujol et al. (2010)),
- texture synthesis, Maurel et al. (2011), Peyré (2009, 2010), Ashikhmin (2001), Lewis (1984), Lefebvre and Hoppe (2005), Efros and Leung (1999), Kwatra et al. (2005),
- similarity analysis, Clarke et al. (2011), De Bonet (1997)
- specific structures modelling, Chen and Zhu (2006), Bargteil et al. (2006), Kwatra et al. (2007), Foster and Metaxas (1996), and
- 3D textures and dynamical textures Grzegorzek (2010), Ashikhmin (2001), Doretto et al. (2003).

In this work, we focus on **variational decomposition models** for texture extraction and analysis. The use of such methods has been initiated in Osher et al. (1992) for denoising purpose. The basic philosophy is the following: consider a noisy image $f = u + b$. We want to get rid of the noise b. The most natural way to do is to minimize this noise which is usually assumed to be gaussian. Unfortunately, this is not sufficient to get a unique solution : we have to add priors on the image we want to recover to get uniqueness (and usually stability). The general form of such models is a problem of minimizing an energy functional

$$\mathcal{F}(u) = \|u - f\|_X + \mathcal{R}(u), \ u \in Y \subset X,$$

where X, Y are (real) linear spaces, $\mathcal{R}$ is a regularization operator, f is the observed (or measured) noisy image and u is the image to recover. The first term is the fitting data term and the second one is a prior that permits to get a problem which has a unique solution.

During the last decade, many methods have been developed using different regularization and/or data fidelity terms. Let us mention the use of Sobolev-type spaces (including Besov spaces or BMO) in the regularization term (see Osher et al. (2003), Garnett et al. (2011), Kim and Vese (2009), Le et al. (2009), Lieu and Vese (2008), Garnett et al. (2007), Le and Vese (2005), Tadmor et al. (2004) for example) and/or the use of Meyer space (Meyer (2001), Aujol et al. (2005), Aubert and Aujol (2005), Strong et al. (2006), Aujol (2009), Gilles and Meyer (2010), Duval et al. (2010)).

These methods have in common to involve only first-order terms, that deal with first order (generalized) derivative. In this work, we focus on higher order methods, namely second-order ones (see Bergounioux and Piffet (2010, 2013), Bergounioux and Tran (2011), Tran et al. (2012), Bergounioux (2011), Demengel (1984), Hinterberger and Scherzer (2006), Bredies et al. (2010, 2011), Knoll et al. (2011)) that seem promising to deal with image texture.

In next section we define what a variational decomposition model is, and focus on first order ones, with a detailed presentation of the so-called Rudin-Osher-Fatemi model. In section 3 we present second-order models, together with the functional framework and compare with the point of view of Bredies et al. (2010). We present numerical algorithms as well. Section 4 is devoted to the 3D case. We propose in section 5, examples and applications. We end with an appendix to provide the main mathematical tools we used and some MATLAB© codes.

2. First order variational decomposition models

2.1. Variational models principle

Variational models in image processing have been extensively studied during the past decade. There are used for segmentation processes (geodesic or geometric contours) and restoration purpose as well. We are mainly interested in the last item which involves denoising or deblurring methods and texture extraction. Shortly speaking, image restoration problems are usually severely ill posed and a Tychonov-like regularization process is needed. The general form of such models consists in the mimization of an energy functional :

$$\mathcal{F}(u) = \|u - u_d\|_X + \mathcal{R}(u) \ , u \in Y \subset X \ ,$$

where X, Y are (real) Banach spaces, $\mathcal{R}$ is a regularization operator, u_d is the observed (or measured) image and u is the image to recover or denoise. The first

term is the fitting data term and the second one permits to get a problem which is no longer ill posed via a regularization process.

Let us give an example of such a regularization process. Let be Ω a open bounded subset of $\mathbb{R}^2$ and $X = H^1(\Omega)$ and $H = L^2(\Omega)$ the usual Sobolev spaces (see Appendix A.3) endowed with the usual norms:

$$\|u\|_2 := \|u\|_{L^2(\Omega)} \text{ and } \|u\|_X^2 := \|u\|_2^2 + \|\nabla u\|_2^2 \ .$$

We consider the following (original) fitting data problem :

$$(\mathcal{P}) \qquad \min_{u\in X} \|u - u_d\|_2^2,$$

where $u_d \in L^2(\Omega)$. It is easy to see that the functional $u \mapsto \|u - u_d\|_2^2$ is not coercive on X: let be $\Omega =]0,1[$, $u_n(x) = x^n$, $u_d = 0$ for instance. Then $\|u_n\|_2 = \dfrac{1}{\sqrt{2n}}$, $\|u_n'\|_2 = \dfrac{n}{\sqrt{2n-1}}$. So

$$\lim_{n\to+\infty} \|u_n\|_X + \infty \text{ and } \lim_{n\to+\infty} \|u_n\|_2 = 0.$$

Therefore, we do not even know if $(\mathcal{P})$ has (at least) a solution. Let us define the regularized problem as

$$(\mathcal{P}_\alpha) \qquad \min_{u\in X} \|u - u_d\|_2^2 + \alpha\|\nabla u\|_2^2$$

where $\alpha > 0$. We want u to fit the data u_d, but ask for the gradient to be small (it depends on α).

Proposition 2.1. *For every $\alpha > 0$, problem $(\mathcal{P}_\alpha)$ has a unique solution u_α. Moreover, assuming that $(\mathcal{P})$ has at least a solution, then one can extract a subsequence of the family (u_α) that weakly converges in X to a solution u^* of $(\mathcal{P})$ as $\alpha \to 0$.*

Proof - Problem $(\mathcal{P}_\alpha)$ has a unique solution u_α because the functional

$$u \mapsto \mathcal{J}_\alpha(u) = \|u - u_d\|_2^2 + \alpha\|\nabla u\|_2^2$$

is coercive, continuous and strictly convex (it is the X-norm up to an affine part) and we may use Theorem A.3. Let us prove that the family (u_α) is uniformly bounded in X with respect to α.

$$\forall u \in X \qquad \mathcal{J}_\alpha(u_\alpha) \le \mathcal{J}_\alpha(u).$$

We have assumed that $(\mathcal{P})$ has at least a solution $u = \tilde{u}$. So

$$\underbrace{\|\tilde{u}\|_2^2 \le \|u_\alpha\|_2^2}_{\tilde{u} \text{ solution to } (\mathcal{P})} \le \underbrace{\mathcal{J}_\alpha(u_\alpha) = \|u_\alpha\|_2^2 + \alpha\|\nabla u_\alpha\|_2^2 \le \mathcal{J}_\alpha(\tilde{u})}_{u_\alpha \text{ solution to } (\mathcal{P}_\alpha)} \tag{2.1}$$
$$= \|\tilde{u}\|_2^2 + \alpha\|\nabla\tilde{u}\|_2^2.$$

So, $\mathcal{J}_\alpha(u_\alpha)$ is bounded independently of $\alpha \leq \alpha_o$. This implies the boundedness of $(u_\alpha)_{\alpha\leq\alpha_o}$ in $L^2(\Omega)$. In addition, we get with (2.1)

$$\alpha\|\nabla u_\alpha\|_2^2 \leq \|\tilde{u}\|_2^2 + \alpha\|\nabla\tilde{u}\|_2^2 - \|u_\alpha\|_2^2 \leq \|\tilde{u}\|_2^2 + \alpha\|\nabla\tilde{u}\|_2^2 - \|\tilde{u}\|_2^2 = \alpha\|\nabla\tilde{u}\|_2^2 \ ;$$

consequently $(u_\alpha)_{\alpha\leq\alpha_o}$ is bounded in X. Therefore, one can extract a subsequence weakly convergent in X to some u^*. We refer to Attouch et al. (2006), Brezis (1987) for the weak convergence notion. On the other hand equation (2.1) gives

$$\lim_{\alpha\to 0} \mathcal{J}_\alpha(u_\alpha) = \|\tilde{u}\|_2^2 = \inf(\mathcal{P}).$$

With the lower semi-continuity of the L^2-norm, we obtain

$$\|u^*\|_2^2 \leq \liminf_{\alpha\to 0} \|u_\alpha\|_2^2 = \liminf_{\alpha\to 0} \mathcal{J}_\alpha(u_\alpha) \leq \inf(\mathcal{P}),$$

so that u^* is a solution to $(\mathcal{P})$. □

We want to compute u_α numerically. As $\mathcal{J}_\alpha$ is strictly convex, u_α satisfies the necessary and sufficient optimality condition :

$$\mathcal{J}'_\alpha(u_\alpha) = 0.$$

A classical computation gives

$$\begin{aligned} \forall u \in X \qquad \frac{1}{2}\mathcal{J}'_\alpha(u_\alpha)\cdot u &= \int_\Omega (u_\alpha - u_d)(x)u(x)dx + \int_\Omega \nabla u_\alpha(x)\nabla u(x)dx \\ &= \int_\Omega (u_\alpha - u_d - \Delta u_\alpha)(x)u(x)dx. \end{aligned}$$

Thus, the solution u_α satisfies the Euler equation:

$$u_\alpha - u_d - \Delta u_\alpha = 0, \ u_\alpha \in H_0^1(\Omega).$$

One usually uses a dynamic formulation and rather solves

$$\frac{\partial u}{\partial t} - \Delta u + u = u_d. \tag{2.2}$$

This dynamic approach is equivalent to calculating a minimizing sequence with a gradient method. Indeed, the basic gradient algorithm with constant step δt writes

$$\frac{u_{t+\delta t} - u_t}{\delta_t} = -\mathcal{J}'_\alpha(u_t);$$

Passing to the limit as $\delta t \to 0$ gives

$$\frac{\partial u}{\partial t} = -\mathcal{J}'_\alpha(u) = \Delta u - u + u_d.$$

The most simple regularization term $L(u) := \|\nabla u\|_2^2$ (Tychonov regularization) is not well adapted to image restoration : the reconstructed image is too smoothed because the Laplacian is an isotropic diffusion operator. In particular, edges are degraded which is not acceptable to perform a good segmentation. It is not surprising, however, since the dynamic heat equation (2.2) is related to a gaussian convolution filter. It is well known that using such a filter adds blur to the result.

2.2. The Rudin-Osher-Fatemi model

A better approach is the use of a regularization term that preserves contours. This implies to deal with functions that can be discontinuous (the jump-set describes the contours). Such functions cannot belong to $H^1((\Omega)$ any longer since their distributional derivative may be Dirac measures. So we have to considerer a less restrictive functional space.

2.2.1. The space of bounded variation functions

Let Ω be an open bounded subset of $\mathbb{R}^n$, $n \geq 2$ (practically $n = 2$ or $n = 3$) smooth enough (with the cone property and $\mathcal{C}^1$ for example). We first recall the definition and the main properties of the space $BV(\Omega)$ of bounded variation functions (see Ambrosio et al. (2000), Aubert and Kornprobst (2006), Attouch et al. (2006) for example). It s defined as

$$BV(\Omega) = \{u \in L^1(\Omega) \mid \Phi_1(u) < +\infty\},$$

where

$$\Phi_1(u) := \sup \left\{ \int_\Omega u(x) \operatorname{div} \xi(x)\, dx \mid \xi = (\xi_1, \cdots, \xi_n) \in \mathcal{C}_c^1(\Omega, \mathbb{R}^n) \parallel |\xi| \parallel_\infty \leq 1 \right\}. \tag{2.3}$$

Here $\mathcal{C}_c^1(\Omega, \mathbb{R}^n)$ denotes the space of $\mathbb{R}^n$ valued, $\mathcal{C}^1$ functions with compact support in Ω endowed with the uniform (L^∞) norm, $|\xi| := \sqrt{\xi_1^2 + \cdots + \xi_n^2}$ and

$$\operatorname{div} \xi = \frac{\partial \xi_1}{\partial x_1} + \cdots + \frac{\partial \xi_n}{\partial x_n}.$$

The space $BV(\Omega)$, endowed with the norm $\|u\|_{BV(\Omega)} = \|u\|_{L^1} + \Phi_1(u)$, is a Banach space. The derivative in the sense of the distributions of every $u \in BV(\Omega)$ is a bounded Radon measure, denoted Du, and $\Phi_1(u) = \int_\Omega |Du|$ is the total variation of u. We next recall standard properties of bounded variation functions (Ambrosio et al. (2000), Attouch et al. (2006)).

Proposition 2.2. *Let Ω be an open subset of $\mathbb{R}^n$ with Lipschitz boundary.*

1. *For every* $u \in BV(\Omega)$, *the Radon measure* Du *can be decomposed into* $Du = \nabla u\, dx + D^s u$, *where* $\nabla u\, dx$ *is the absolutely continuous part of* Du *with respect of the Lebesgue measure and* $D^s u$ *is the singular part.*
2. *The mapping* $u \mapsto \Phi_1(u)$ *is lower semi-continuous from* $BV(\Omega)$ *to* $\mathbb{R}^+$ *for the* $L^1(\Omega)$ *topology.*
3. $BV(\Omega) \subset L^2(\Omega)$ *with continuous embedding, if* $n = 2$.
4. $BV(\Omega) \subset L^p(\Omega)$ *with compact embedding, for every* $p \in [1, 2)$, *if* $n = 2$.

We end this section with a "density" result in $BV(\Omega)$ (Attouch et al. (2006) Theorem 10.1.2. p 375 for example):

Theorem 2.1. *The space* $\mathcal{C}^\infty(\overline{\Omega})$ *is dense in* $BV(\Omega)$ *in the following sense :* $\forall u \in BV(\Omega)$ *there exist a sequence* $(u_n)_{n\geq 0} \in \mathcal{C}^\infty(\overline{\Omega})$ *such that*

$$\lim_{n\to+\infty} \|u_n - u\|_{L^1} = 0 \qquad \textit{and} \quad \lim_{n\to+\infty} \Phi_1(u_n) = \Phi_1(u)\ .$$

A useful corollary is a Poincaré-Wirtinger inequality in the BV- space

Theorem 2.2. *Let* $\Omega \subset \mathbb{R}^n$ *be an open connected, bounded set of class* $\mathcal{C}^1$. *Then there exists a constant* $C > 0$ *such that*

$$\forall u \in BV(\Omega) \qquad \|u - m(u)\|_{L^1(\Omega)} \leq C\Phi_1(u)\ ,$$

where $m(u) := \dfrac{1}{|\Omega|}\displaystyle\int_\Omega u(x)dx$ *is the mean-value of* u.

Proof - Let $u \in BV(\Omega)$ and $(u_n)_{n\geq 0} \in \mathcal{C}^\infty(\overline{\Omega})$ be a sequence such that

$$\lim_{n\to+\infty} \|u_n - u\|_{L^1} = 0 \qquad \text{and} \quad \lim_{n\to+\infty} \Phi_1(u_n) = \Phi_1(u)\ .$$

It is clear that $m(u_n) \to m(u)$. In addition $u_n \in W^{1,1}(\Omega)$ since Ω is bounded. We use the Poincaré-Witinger inequality (Attouch et al. (2006), Corollary 5.4.1 p180 for example) to infer

$$\forall n \qquad \|u_n - m(u_n)\|_{L^1(\Omega)} \leq C\|\nabla u_n\|_{L^1} = \Phi_1(u_n)\ .$$

Passing to the limit gives the result. □

2.2.2. The Rudin-Osher-Fatemi model

The most famous model is the Rudin-Osher-Fatemi denoising model (see Acar and Vogel (1994), Osher et al. (1992)). This model involves a regularization term that preserves the solution discontinuities, what a classical H^1-Tychonov regularization method does not. The observed image to recover is splitted in two parts $u_d = u + v$ where v represents the oscillating component (noise or texture)

and u is the smooth part (oftenly called the *cartoon* component). So we look for the solution as $u + v$ with $u \in BV(\Omega)$ and $v \in L^2(\Omega)$. The regularization term involves only the cartoon component u, while the remainder term $v = u_d - u$ represents the noise to be minimized. We get

$$\min_{u \in BV(\Omega)} \mathcal{F}_1(u) := \frac{1}{2}\|u_d - u\|^2_{L^2(\Omega)} + \lambda \Phi_1(u), \tag{$\mathcal{P}_1$}$$

where $\Phi_1(u)$ is the total variation of u and $\lambda > 0$.

Theorem 2.3. *Problem ($\mathcal{P}_1$) has a unique solution in $BV(\Omega)$.*

Proof - Let $u_n \in BV(\Omega)$ be a minimizing sequence. As u_n is bounded in $L^2(\Omega)$ one may extract a subsequence (denoted similarly) that weakly converges to u^* in $L^2(\Omega)$. As the L^2-norm is lower semi-continuous and convex we have

$$\|u_d - u^*\|_2^2 \leq \liminf_{n \to +\infty} \|u_d - u_n\|_2^2.$$

Moreover u_n is bounded in $L^1(\Omega)$ since Ω is bounded. As $\Phi_1(u_{n)}$ is bounded as well, then u_n is bounded in $BV(\Omega)$. As $BV(\Omega)$ is compactly embedded in $L^1(\Omega)$ (proposition 2.2) this implies that u_n strongly converges (up to a subsequence) in $L^1(\Omega)$ to $u^* \in BV(\Omega)$.
In addition, Φ_1 is lower semi-continuous with respect to the L^1 strong topology (proposition 2.2), so that

$$\Phi_1(u^*) \leq \liminf_{n \to +\infty} \Phi_1(u_n).$$

Eventually

$$\frac{1}{2}\|u_d - u^*\|_2^2 + \lambda \Phi_1(u^*) \leq \liminf_{n \to +\infty} \frac{1}{2}\|u_d - u_n\|_2^2 + \lambda \Phi_1(u_n) = \inf(\mathcal{P}_1).$$

So u^* is a solution to problem ($\mathcal{P}_1$). As the cost functional is strictly convex we get uniqueness. □

Now, we want to set optimality conditions to compute the solution. Unfortunately Φ_1 is not (Gâteaux) differentiable and we need non smooth analysis tools (see Appendix A.2).

2.2.3. First order optimality condition

The functional $\mathcal{F}_1$ is convex. Therefore $\bar{u}$ is solution to ($\mathcal{P}_1$) if and only if $0 \in \partial \mathcal{F}_1(\bar{u})$ where $\partial \mathcal{F}_1(\bar{u})$ denotes the subdifferential of $\mathcal{F}_1$ at $\bar{u}$ (Appendix A.2.2). We use Theorem A.8 to compute $\partial \mathcal{F}_1(u)$. Indeed the function $u \mapsto \|u - u_d\|_2^2$ is

continuous on $L^2(\Omega)$ and Φ_1 is finite on $BV(\Omega)$ with values in $\mathbb{R} \cup \{+\infty\}$. As $u \mapsto \|u - u_d\|_2^2$ is Gâteaux-differentiable on $L^2(\Omega)$ as well we get

$$0 \in \partial \mathcal{F}_1(\bar{u}) = \bar{u} - u_d + \partial(\lambda \Phi_1(\bar{u})) = \bar{u} - u_d + \lambda \partial \Phi_1(\bar{u}) \ ,$$

that is

$$\frac{u_d - \bar{u}}{\lambda} \in \partial \Phi_1(\bar{u}) \ .$$

It remains to compute $\partial \Phi_1(\bar{u})$. Using corollary A.4, it comes

$$\frac{u_d - \bar{u}}{\lambda} \in \partial \Phi_1(\bar{u}) \Longleftrightarrow \bar{u} \in \partial \Phi_1^*(\frac{u_d - \bar{u}}{\lambda}) \ ,$$

where Φ_1^* is the Legendre-Fenchel conjugate of Φ_1 that we compute now.

Theorem 2.4. *The Legendre-Fenchel conjugate Φ_1^* of the total variation Φ_1 is the indicatrix function of the L^2-closure $\overline{\mathcal{K}_1}$ of the set $\mathcal{K}_1$, where*

$$\mathcal{K}_1 := \left\{ \xi = div\ \varphi \mid \varphi \in \mathcal{C}_c^1(\Omega)^n, \ \| \, |\varphi| \, \|_\infty \leq 1 \right\}. \tag{2.4}$$

Proof - The result is well known (Aujol (2009), Chambolle (2004) for example) but we give a proof anyway for convenience.
As Φ_1 is a semi-norm, it is positively homogeneous and the conjugate Φ_1^* is the indicatrix function of a closed convex set $\tilde{K}$ (Proposition A.4).
We first show that $\mathcal{K}_1 \subset \tilde{K}$: let be $u \in \mathcal{K}_1$. The definition of Φ_1 gives

$$\Phi_1(u) = \sup_{\xi \in \mathcal{K}_1} (\xi, u) \, . \tag{2.5}$$

Therefore $(\xi, u)) - \Phi_1(u) \leq 0$ for every $\xi \in \mathcal{K}_1$ and $u \in L^2(\Omega)$ (Note that if $u \in L^2(\Omega) \backslash BV(\Omega)$ then $\Phi_1(u) = +\infty$). We deduce that

$$\forall u^* \in \mathcal{K}_1 \quad \Phi_1^*(u^*) = \sup_{u \in L^2(\Omega)} (u^*, u) - \Phi_1(u) = \sup_{u \in BV(\Omega)} (u^*, u) - \Phi_1(u) \leq 0.$$

As Φ_1^* takes only one finite value then $\Phi_1^*(u^*) = 0$ and $u^* \in \tilde{K}$. Therefore $\mathcal{K}_1 \subset \tilde{K}$; as $\tilde{K}$ is closed then

$$\overline{\mathcal{K}_1} \subset \tilde{K}.$$

Eventually,

$$\Phi_1(u) = \sup_{\xi \in \mathcal{K}_1} (u, \xi) \leq \sup_{\xi \in \overline{\mathcal{K}_1}} (u, \xi) \leq \sup_{\xi \in \tilde{K}} (u, \xi) = \sup_{\xi \in \tilde{K}} (u, \xi) - \Phi_1^*(\xi) = \Phi_1^{**}(u).$$

As $\Phi_1^{**} = \Phi_1$, then

$$\sup_{\xi \in \mathcal{K}_1} (u, \xi) \leq \sup_{\xi \in \tilde{K}} (u, \xi) \leq \sup_{\xi \in \mathcal{K}_1} (u, \xi) \, ,$$

and

$$\sup_{\xi\in\mathcal{K}_1}(u,\xi)=\sup_{\xi\in\overline{\mathcal{K}_1}}(u,\xi)=\sup_{\xi\in\tilde{K}}(u,\xi)\,. \tag{2.6}$$

Assume there exists $u^*\in\tilde{K}$ such that $u^*\notin\overline{\mathcal{K}_1}$. With Hahn-Banach Theorem A.6, one can strictly separate u^* and the closed convex set $\overline{\mathcal{K}_1}$. There exists $\alpha\in\mathbb{R}$ and $u_0\in L^2(\Omega)$ such that

$$(u_0,u^*)>\alpha\geq\sup_{v\in\overline{\mathcal{K}_1}}(u_0,v)\,.$$

With (2.6) we obtain

$$\sup_{\xi\in\tilde{K}}(u_0,\xi)\geq(u_0,u^*)>\alpha\geq\sup_{v\in\overline{\mathcal{K}_1}}(u_0,v)=\sup_{v\in\tilde{K}}(u_0,v)\,.$$

We get a contradiction. Therefore $\tilde{K}=\overline{\mathcal{K}_1}$. □

Finally, $\bar{u}$ is solution to $(\mathcal{P}_1)$ if and only if

$$\bar{u}\in\partial 1_{\overline{\mathcal{K}_1}}(\frac{u_d-\bar{u}}{\lambda}).$$

Using proposition A.2 gives

$$\bar{u}=c\left[\frac{u_d-\bar{u}}{\lambda}+\frac{\bar{u}}{c}-\Pi_{\mathcal{K}_1}(\frac{u_d-\bar{u}}{\lambda}+\frac{\bar{u}}{c})\right]$$

for every $c>0$. Here and in the sequel $\Pi_{\mathcal{K}_1}$ denotes the L^2 projection on $\overline{\mathcal{K}_1}$. Now, set $c=\lambda$ to obtain :

$$\bar{u}=u_d-\lambda\Pi_{\mathcal{K}_1}\left(\frac{u_d}{\lambda}\right)\,.$$

As $\Pi_{\lambda\mathcal{K}_1}=\lambda\Pi_{\mathcal{K}_1}\left(\frac{u_d}{\lambda}\right)$ (with corollary A.3) we have the following result:

Theorem 2.5. *The function $\bar{u}$ is the solution to $(\mathcal{P}_1)$ if and only if*

$$\bar{u}=u_d-\Pi_{\lambda\mathcal{K}_1}(u_d)$$

where $\Pi_{\lambda\mathcal{K}_1}$ is the L^2-projection on $\overline{\lambda\mathcal{K}_1}$.

We shall perform the numerical realization in section 2.4. This model is used for denoising purpose but the use of the total variation implies numerical perturbations. The computed solution turns to be piecewise constant and artificial contours are generated: this is the *staircasing effect* (Buades et al. (2006)). Therefore, though noise can be succesfully removed, the solution is not satisfactory. This variational model has been improved using different functional spaces, for the data fitting term and/or the regularizing term.

2.3. Some generalizations

Recently people considered that an image can be decomposed into many components, each component describing a particular property of the image (Aujol et al. (2005), Aubert and Aujol (2005), Garnett et al. (2011), Le et al. (2009), Le and Vese (2005) and references therein for example). It is assumed that the image to be recovered from the data u_d can be decomposed as $f = u + v$ or $f = u + v + w$ where u, v and w are functions that characterize different parts of f (see Aujol et al. (2005), Osher et al. (2003), Yin et al. (2007) for example). We cannot present every model since there are too many. We focus on the Meyer model and improved variants by Aujol and al. (Aujol et al. (2005), Aubert and Aujol (2005), Aujol and Chambolle (2005)).

2.3.1. The Meyer model

Assume we want to decompose the image as $u_d = u + v$ where $u \in BV(\Omega)$ is the cartoon part. The remainder term $v = u_d - u$ should involve the oscillating component (as noise and/or texture). Such decompositions have been performed in Aubert and Aujol (2005), Aujol et al. (2005) using the Meyer-space of oscillating functions G (see Meyer (2001)). This space is defined as follows

$$G(\Omega) := \{\ f = \mathrm{div}(g) \mid g = (g_1, g_2) \in L^\infty(\Omega) \times L^\infty(\Omega)\ \}\ . \tag{2.7}$$

This space equipped with the norm

$$\|f\|_G := \inf\{\ \|\,|g|\,\|_\infty \mid f = \mathrm{div}(g),\ g = (g_1, g_2) \in L^\infty(\Omega) \times L^\infty(\Omega)\}$$

is a Banach space. In addition, if $\mathcal{BV}$ is the closure of the Schwartz class in BV then G is the dual space $\mathcal{BV}^*$ of $\mathcal{BV}$. The G - norm is a tool that measures the oscillations. More precisely the more f is oscillating, the less is $\|f\|_G$. Nevertheless, non oscillating functions may have a small G-norm.

In Meyer (2001), the following result is proved, that gives a characterization of the solutions of the Rudin-Osher-Fatemi model ($\mathcal{P}_1$)) with respect to the parameter λ.

Theorem 2.6. *Let u_d, u and v three functions in $L^2(\Omega)$. If $\|u_d\|_G > \lambda$ then the (unique) ROF decomposition $u_d = u + v$ is characterized by*

$$\|v\|_G = \lambda \text{ and } (u, v)_2 = \lambda \|u\|_{BV}\ .$$

As already mentionned, oscillating functions have a small G-norm and textures and/or noise may be viewed as the oscillating parts of the image u_d. So, the ROF model may be improved by replacing the L^2-norm by the G-norm in the data fitting term. This model has been investigated in Meyer (2001) :

$$\min_{u \in BV(\Omega)} \mathcal{F}_G(u) := \frac{1}{2}\|u_d - u\|_G + \lambda \Phi_1(u), \tag{$\mathcal{P}_G$}$$

One can find numerics in Osher and Vese (2003) for example.

2.3.2. Generalized $u+v+w$ decomposition models

In Aubert and Aujol (2005), Aujol et al. (2005), the authors investigate a new decomposition model : $u_d = u + v + w$ where

- $u \in BV(\Omega)$ is the cartoon part,
- $v \in G_\mu(\Omega)$ is an oscillating part (texture). Here, $\mu > 0$ and

$$G_\mu(\Omega) := \{v \in G(\Omega) \mid \|v\|_G \leq \mu \} ,$$

- $w = u_d - u - v \in L^2(\Omega)$ is the remainder part (noise).

The model writes

$$\min_{(u,v)\in BV(\Omega)\times G_\mu(\Omega)} \frac{1}{2}\|u_d - u - v\|_2^2 + \lambda\Phi_1(u), \qquad (\mathcal{P}_{G_\mu})$$

The discretized problem $(\mathcal{P}_{G_\mu})$ has a unique solution and the authors propose an algorithm to solve it in Aujol et al. (2005) . The link to the Meyer model is done and numerical tests are performed. For more details one can refer to Aubert and Aujol (2005), Aujol et al. (2005, 2003), Strong et al. (2006), Aujol and Chambolle (2005)

2.4. Numerical computation

2.4.1. Rudin-Osher-Fatemi discrete model

We now consider discrete 2D images (with finite number of pixels) which is the practical case. Such a discrete image is identified to a matrix $N \times M$ that we may view as a vector of length NM. We denote $X = \mathbb{R}^{N\times M}$ and $Y = X \times X$. The Hilbert space X is endowed with the usual scalar product

$$(u, v)_X = \sum_{1\leq i\leq N} \sum_{1\leq j\leq M} u_{ij} v_{ij},$$

and the associated norm $\|\cdot\|_X$.

We now give a discrete formulation of what we have described previously. In particular, we define the discrete total variation which is the ℓ^1 -norm of the usual gradient. More precisely, for every $u \in X$, the gradient ∇u is a vector in Y:

$$(\nabla u)_{i,j} = ((\nabla u)^1_{i,j}, (\nabla u)^2_{i,j}),$$

defined with classical a finite difference scheme, for example

$$(\nabla u)^1_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if} \quad i < N \\ 0 & \text{si} \quad i = N \end{cases} , \qquad (2.8a)$$

$$(\nabla u)^2_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if} \quad j < M \\ 0 & \text{si} \quad j = M \end{cases} \tag{2.8b}$$

The discrete total variation writes

$$J_1(u) = \sum_{1 \le i \le N} \sum_{1 \le j \le M} |(\nabla u)_{i,j}|, \tag{2.9}$$

where $|(\nabla u)_{i,j}| := \sqrt{|(\nabla u)^1_{i,j}|^2 + |(\nabla u)^2_{i,j}|^2}$. We use a discrete version of the divergence operator as well, setting

$$\text{div} \ = -\nabla^*,$$

where ∇^* is the adjoint operator of ∇, that is

$$\forall p \in Y, \forall u \in X \qquad (-\text{div } p, u)_X = (p, \nabla u)_Y = (p^1, \nabla^1 u)_X + (p^2, \nabla^2 u)_X.$$

One can verify that the discrete divergence writes

$$\begin{aligned} (\text{div } p)_{i,j} = &\begin{cases} p^1_{i,j} - p^1_{i-1,j} & \text{if } 1 < i < N \\ p^1_{i,j} & \text{if } i = 1 \\ -p^1_{i-1,j} & \text{if } i = N \end{cases} \\ + &\begin{cases} p^2_{i,j} - p^2_{i,j-1} & \text{if } 1 < j < M \\ p^2_{i,j} & \text{if } j = 1 \\ -p^2_{i,j-1} & \text{if } j = M \end{cases} \end{aligned} \tag{2.10}$$

The discrete laplacian operator is defined as

$$\Delta u = \ \text{div } (\nabla u).$$

Once this discretization is performed, problems $(\mathcal{P}_1)$ turns to be

$$\min_{u \in X} F_1(u) := \|u - u_d\|^2_X + \lambda J_1(u). \tag{2.11}$$

We can prove as in section 2.2 that the discretized problem has a unique solution that we are going to characterize. Similarly

$$J_1(u) = \sup_{\xi \in K_1} (u, \xi)_X \,,$$

where

$$K_1 = \{\xi = \text{div } g \,|\, g \in Y, |g_{i,j}| \le 1, 1 \le i \le N, 1 \le j \le M\}, \tag{2.12}$$

and

$$\forall g = (g^1, g^2) \in Y \qquad |g_{i,j}| = \sqrt{(g^1_{i,j})^2 + (g^2_{i,j})^2}.$$

As in section 2.2 we have

Theorem 2.7. *Chambolle (2004)The Legendre-Fenchel conjugate J_1^* of J_1 is the indicatrix function of K_1 given by (2.12).*
Moreover, $\bar{u}$ is the solution to problem (2.11) is and only if

$$\bar{u} = u_d - P_{\lambda K_1}(u_d), \tag{2.13}$$

where P_C is the orthogonal projector from X on the closed convex set C.

Compute the solution to problem (2.12) is equivalent to compute the projection on the set λK_1. Of course, this is not straighforward. We report here two algorithms: the first one (Chambolle (2004)) is a fixed-point type algorithm. The second one is a Nesterov type algorithm (Nesterov (2005)) that has been adapted to the context by Weiss et al. (2009).

2.4.2. Chambolle algorithm
We have to compute

$$P_{\lambda K_1}(u_d) = \text{argmin } \{ \|\lambda \text{ div } (p) - u_d\|_X^2 \mid |p_{i,j}| \leq 1,\ i = 1, \cdots, N, j = 1, \cdots, M \}.$$

Following Chambolle (2004) we use a fixed-point method :

Algorithm 1 Chambolle algorithm

Initialization : $n = 0$; $p^0 = 0$
Iteration n : set

$$p_{i,j}^{n+1} = \frac{p_{i,j}^n + \rho \left(\nabla[\text{div } p^n - u_d/\lambda]\right)_{i,j}}{1 + \rho \left|\left(\nabla[\text{div } p^n - u_d/\lambda]\right)_{i,j}\right|}.$$

Stopping criterion.

If the parameter ρ satisfies $\rho \leq 1/8$, then $\lambda \text{ div } p^n \to P_{\lambda K_1}(u_d)$ and the solution writes

$$\bar{u} = u_d - \lambda \text{ div } p^\infty \quad \text{where } p^\infty = \lim_{n \to +\infty} p^n.$$

2.4.3. Nesterov type algorithms
The previous method works well but is rather slow. We now present a faster algorithm. It is derived from a method by Y. Nesterov (Nesterov (2005)).The original goal was to solve

$$\inf_{q \in Q} E(q) \tag{2.14}$$

where E is convex, differentiable with Lipschitz derivative and Q is a closed set. Let d be a convex function , $x_0 \in Q$ and $\sigma > 0$ such that

$$\forall x \in Q \quad d(x) \geq \frac{\sigma}{2}\|x - x_0\|^2.$$

The algorithm writes

Algorithm 2 Nesterov algorithm

Initialization : $k = 0$; $G_0 = 0$; $x_k \in Q$ and L is the Lipschitz constant of ∇E.

Iteration k :

for $0 \leq k \leq J$ **do**

(a) Set $\eta_k = \nabla E(x_k)$.

(b) Compute y_k the solution to

$$\min_{y \in Q} \left\{ (\eta_k, y - x_k)_X + \frac{1}{2} L \|y - x_k\|_X^2 \right\} .$$

(c) $G_k = G_{k-1} + \frac{k+1}{2}\eta^k$.

(d) Compute z_k the solution to

$$\min_{z \in Q} \left\{ \frac{L}{\sigma} d(z) + (G_k, z)_X \right\} .$$

(e) Set $x_k = \frac{2}{k+3} z_k + \frac{k+1}{k+2} y_k$.

end for

It has been proved that if $\bar{u}$ is the solution to (2.14) then

$$0 \leq E(y_k) - E(\overline{u}) \leq \frac{4Ld(\overline{u})}{\sigma(k+1)(k+2)} .$$

In our case, Weiss et al. (2009) have adapted the method to solving the dual problem of (2.11). Using Theorem A.10 (A.2.4) gives

$$\begin{aligned} \min_{u \in X} J_1(u) + \frac{1}{2\lambda}\|u - u_d\|_X^2 &= \max_{v \in X} (-J_1^*(-v) - \mathcal{N}_\lambda^*(v)) \\ &= -\min_{q \in X} (J_1^*(-v) + \mathcal{N}_\lambda^*(v)) \ , \end{aligned}$$

where $\mathcal{N}_\lambda(u) = \frac{1}{2\lambda}\|u - u_d\|_X^2$. We already noticed that J_1^* is the indicatrix of the set K_1 defined by (2.12). Let us compute $\mathcal{N}_\lambda^*$:

$$\mathcal{N}_\lambda^*(v) = \sup_{u \in X}((u, v)_X - \mathcal{N}_\lambda(u)) = \sup_{u \in X}((u, v)_X - \frac{1}{2\lambda}\|u - u_d\|_X^2) .$$

The supremum is achived at $u = \lambda v + u_d$ and

$$\mathcal{N}_\lambda^*(v) = \frac{\lambda}{2}\|v\|_X^2 + vu_d = \frac{1}{2\lambda}\|\lambda v + u_d\|_X^2 - \frac{\|u_d\|^2}{2\lambda} \ .$$

The dual problem writes

$$\min_{v\in K} \|\lambda v + u_d\|_X^2 = \min_{p\in\mathcal{B}_\lambda} \| - \mathrm{div}(p) + u_d\|_X^2 \ , \tag{2.15}$$

where

$$\mathcal{B}_\lambda := \ \{\ p = (p^1, p^2) \in X \times X |\ |p_{i,j}| \leq \lambda,\ 1 \leq i \leq N, 1 \leq j \leq M\ \} \ .$$

The solution $\bar{u}$ of the primal problem (2.11) is obtained as follows

$$\bar{u} = u_d - \lambda \bar{v} \ , \tag{2.16}$$

where $\bar{v} = \mathrm{div}\ \bar{p}$ is solution to (2.15). Now, we may use algorithm 2 to solve(2.15). We set

$$E(p) = \frac{1}{2}\| - \mathrm{div}(p) + u_d\|_X^2 \text{ and } Q = \mathcal{B}_\lambda,$$

and choose $d(x) = \frac{1}{2}\|x\|_X^2$ with $x_0 = 0$ and $\sigma = 1$.

- Step (a) gives $\eta_k = \nabla E(p_k) = \nabla(-\mathrm{div}(p_k) + u_d)$
- Step (b) : as

  $$(\eta_k, y - x_k)_X + \frac{L}{2}\|y - x_k\|_X^2 = \frac{L}{2}\left\|y - x_k + \frac{\eta_k}{L}\right\|_X^2 - \frac{\|\eta_k\|_X^2}{2L}$$

  we need to compute the solution to

  $$\min_{y\in\mathcal{B}_\lambda} \left\|y - p_k + \frac{\eta_k}{L}\right\|_X^2 .$$

  Step (b) turns to calculate q_k the ℓ^2 (euclidean) projection on the ℓ^∞-ball $\mathcal{B}_\lambda$ (see A.1.4) of $p_k - \frac{\eta_k}{L}$:

  $$q_k = \Pi_{\mathcal{B}_\lambda}\left(p_k - \frac{\eta_k}{L}\right) \ .$$

- Similarly, step (d) is equivalent to the computation of

  $$z_k = \Pi_{\mathcal{B}_\lambda}\left(-\frac{G_k}{L}\right) \ .$$

We eventually obtain

Algorithm 3 Modified Nesterov algorithm (Weiss et al. (2009))

Input : the maximal number of iterations I_{max} and an initial guess $p_0 \in \mathcal{B}_\lambda$ are given.

Output : $\tilde{q} := q_{I_{max}}$ approximates $\bar{q}$ solution to (2.15)

Let $L = \|\mathrm{div}\|_2^2$ be the discrete divergence operator norm.
Set $G_{-1} = 0$
for $0 \leq k \leq I_{max}$ **do**
$\eta_k = \nabla(-\mathrm{div}(p_k) + u_d)$
$q_k = \Pi_{\mathcal{B}_\lambda}\left(p_k - \dfrac{\eta_k}{L}\right)$.
$G_k = G_{k-1} + \dfrac{k+1}{2}\eta_k,\ z_k = \Pi_{\mathcal{B}_\lambda}\left(-\dfrac{G_k}{L}\right)$.
$p_{k+1} = \dfrac{2}{k+3}z_k + \dfrac{k+1}{k+3}q_k$
end for

The solution of problem (2.11) is approximated by $\tilde{u}$:

$$\tilde{u} = u_d - \lambda \mathrm{div}(\tilde{q}) \ . \tag{2.17}$$

3. Second order models (2D case)

The ROF variational model is a good tool to perform denoising while preserving contours (what a Gaussian filter does not achieve). However, there are undesired effects that come from the use of first-order (generalized) derivative (total variation or more complicated terms). Roughly speaking, the solution should have a very small first order derivative. Concerning the total variation, which is also the total length of contours, it gives satisfactory denoising but the solution turns to be (more or less) piecewise constant. Therefore, original contours are kept but artificial ones may be created which is not acceptable. This is called the *staircasing effect* (Caselles et al. (2007), Ring (2000)). We give an example below (Bergounioux and Piffet (2010)).

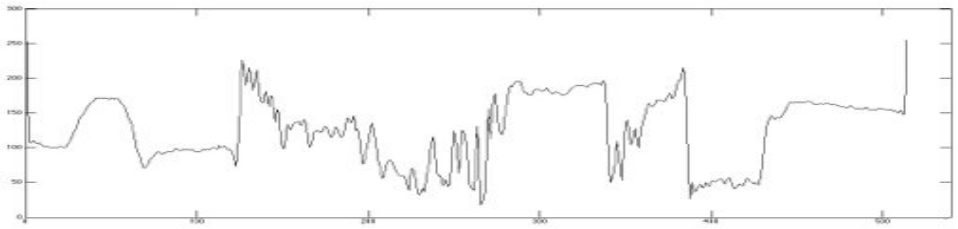
(a) Original Image (b) Noisy Image (c) Denoised Image

Figure 3.1: ROF denoising process - Gaussian noise with standard deviation $\sigma = 0.25$ and λ=50. Staircasing effect

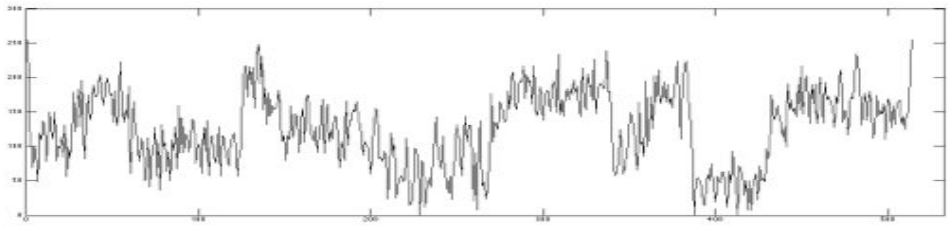
(a) Zoomed area (b) Noisy Image

(c) Denoised

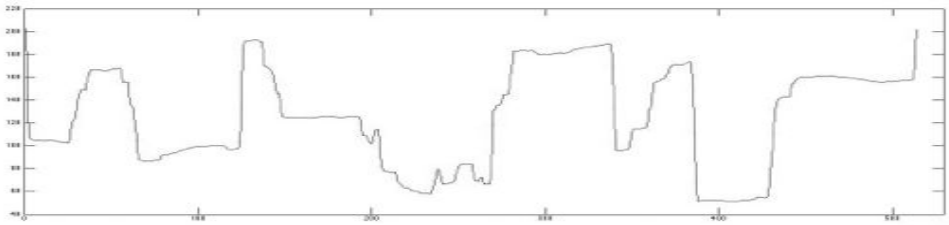
Figure 3.2: ROF denoising process - Zoom -Staircasing effect



(a) Original slice



(b) Noisy slice



(c) Denoised slice

Figure 3.3: ROFdenoising process - Extraction of a slice -Staircasing effect

We infer that the use of a second order penalization term leads to piecewise affine solutions so that there is no staircasing effect any longer. In this section, we present a second order decomposition model for 2D- denoising and texture extraction. We present the functional framework (space BV^2) and compare with the Total Generalized Variation introduced by Bredies et al. (2010).Then, we give numerical hints and improved variants. We end with a comparison between ROF and the second-order methods.

3.1. The space $BV^2(\Omega)$

3.1.1. General properties

We extend the concept of (first-order) variation definition to the second derivative (in the distributional sense). Recall that the Sobolev space $W^{1,1}(\Omega)$ is defined as

$$W^{1,1}(\Omega) = \{\ u \in L^1(\Omega) \mid \nabla u \in L^1(\Omega)\ \}$$

where ∇u stands for the first order derivative of u (in the sense of distributions). Full results can be found in Demengel (1984), Hinterberger and Scherzer (2006), Bergounioux and Piffet (2010).

Definition 3.1. *A function $u \in W^{1,1}(\Omega)$ is **Hessian bounded** if*

$$\Phi_2(u) := \sup \left\{ \int_\Omega \langle \nabla u, div(\xi) \rangle_{\mathbb{R}^n} \ \mid \xi \in \mathcal{C}_c^2(\Omega, \mathbb{R}^{n\times n}),\ \|\xi\|_\infty \leq 1 \right\} < \infty, \quad (3.1)$$

where

$$div(\xi) = (div(\xi_1), div(\xi_2), \ldots, div(\xi_n)),$$

with

$$\forall i,\ \xi_i = (\xi_i^1, \xi_i^2, \ldots, \xi_i^n) \in \mathbb{R}^n \quad and \quad div(\xi_i) = \sum_{k=1}^n \frac{\partial \xi_i^k}{\partial x_k}.$$

$BV^2(\Omega)$ is defined as following

$$BV^2(\Omega) := \{u \in W^{1,1}(\Omega) \mid \Phi_2(u) < +\infty\}.$$

We recall that if $X = \mathbb{R}^{n\times n}$, $\|\xi\|_\infty = \sup_{x\in\Omega} \sqrt{\sum_{i,j=1}^n \left|\xi_i^j(x)\right|^2}$.

We give thereafter many useful properties of $BV^2(\Omega)$ (proofs can be found in Demengel (1984), Bergounioux and Piffet (2010)).

Theorem 3.1. *The space $BV^2(\Omega)$ endowed with the following norm*

$$\|f\|_{BV^2(\Omega)} = \|f\|_{W^{1,1}(\Omega)} + \Phi_2(f) = \|f\|_{L^1} + \|\nabla f\|_{L^1} + \Phi_2(f), \quad (3.2)$$

where Φ_2 is given by (3.1), is a Banach space.

Proposition 3.1. *A function u belongs to $BV^2(\Omega)$ if and only if $u \in W^{1,1}(\Omega)$ and $\dfrac{\partial u}{\partial x_i} \in BV(\Omega)$ for $i \in \{1, \ldots, n\}$. In particular*

$$\Phi_2(u) \leq \sum_{i=1}^n \Phi_1\left(\frac{\partial u}{\partial x_i}\right) \leq n\ \Phi_2(u).$$

Remark 3.1. *The previous result shows that*

$$BV^2(\Omega) = \left\{ u \in W^{1,1}(\Omega) \mid \forall i \in \{1, \ldots, n\},\ \frac{\partial u}{\partial x_i} \in BV(\Omega) \right\}.$$

We get a lower semi-continuity result for the semi-norm Φ_2 as well.

Theorem 3.2. *The operator Φ_2 is lower semi-continuous from $BV^2(\Omega)$ endowed with the strong topology of $W^{1,1}(\Omega)$ to $\mathbb{R}$. More precisely, if $\{u_k\}_{k\in\mathbb{N}}$ is a sequence of $BV^2(\Omega)$ that strongly converges to u in $W^{1,1}(\Omega)$ then*

$$\Phi_2(u) \leq \liminf_{k\to\infty} \Phi_2(u_k).$$

Remark 3.2. *In particular, if $\liminf_{k\to\infty} \Phi_2(u_k) < \infty$, then $u \in BV^2(\Omega)$.*

We have embedding results as well:

Theorem 3.3. *(Demengel (1984)) Assume $n \geq 2$. Then*

$$BV^2(\Omega) \hookrightarrow W^{1,q}(\Omega) \quad with \quad q \leq \frac{n}{n-1}, \tag{3.3}$$

with continuous embedding. Moreover the embedding is compact if $q < \frac{n}{n-1}$. In particular

$$BV^2(\Omega) \hookrightarrow L^q(\Omega) \quad for \quad q \leq \frac{n}{n-2} \quad if \quad n > 2 \tag{3.4}$$

$$BV^2(\Omega) \hookrightarrow L^q(\Omega), \quad \forall q \in [1, \infty[, \quad if \quad n = 2. \tag{3.5}$$

In the sequel, we set $n = 2$ and Ω is a Lipschitz bounded, open subset of $\mathbb{R}^2$, so that $BV^2(\Omega) \subset H^1(\Omega)$ with continuous embedding and $BV^2(\Omega) \subset W^{1,1}(\Omega)$ with compact embedding. Let us define the space $BV_0(\Omega)$ as the space of functions of bounded variation that vanish on the boundary $\partial\Omega$ of Ω. More precisely as Ω is bounded and $\partial\Omega$ is Lipschitz, functions of $BV(\Omega)$ have a trace of class L^1 on $\partial\Omega$ (see Ziemer (1989), Ambrosio et al. (2000)), and the trace mapping $T : BV(\Omega) \to L^1(\partial\Omega)$ is linear, continuous from $BV(\Omega)$ equipped with the intermediate convergence to $L^1(\partial\Omega)$ endowed with the strong topology (Attouch et al. (2006), Theorem 10.2.2 p 386). The space $BV_0(\Omega)$ is then defined as the kernel of T. It is a Banach space, endowed with the induced norm. Note that if $u \in BV^2(\Omega)$ the trace $u_{|\partial\Omega}$ belongs to $H^{1/2}(\partial\Omega) \subset L^2(\partial\Omega)$:

$$BV_0(\Omega) := \{u \in BV(\Omega) \mid u_{|\partial\Omega} = 0 \} ,$$

Next we define similarly

$$BV_0^2(\Omega) := \{u \in BV^2(\Omega) \mid u_{|\partial\Omega} = 0 \} ,$$

$$BV_m(\Omega) := \{u \in BV(\Omega) \mid \int_\Omega u(x)\, dx = 0 \ i = 1, \cdots, n\} ,$$

and

$$BV_m^2(\Omega) := \{u \in BV^2(\Omega) \mid \int_\Omega \frac{\partial u}{\partial x_i}\, dx = 0 \ i = 1, \cdots, n\} .$$

At last we shall use the following result of Bergounioux (2011):

Lemma 3.1 (Poincaré-Wirtinger inequalities). *Let $\Omega \subset \mathbb{R}^n$ be an open Lipschitz bounded set. There exist generic constants only depending on Ω, $C_i > 0$ such that*

$$\begin{aligned}
\forall u \in BV_0(\Omega) \qquad & \|u\|_{L^1(\Omega)} \leq C_1 \Phi_1(u), \\
\forall u \in BV_m(\Omega) \qquad & \|u\|_{L^1(\Omega)} \leq C_2 \Phi_1(u), \\
\forall u \in BV_0^2(\Omega) \qquad & \Phi_1(u) \leq C_1 \Phi_2(u) \\
\forall u \in BV_m^2(\Omega) \qquad & \Phi_1(u) \leq C_2 \Phi_2(u)
\end{aligned}$$

We end with a remark related to next subsection. Let us call

$$\mathcal{K} := \left\{ \xi \in \mathcal{C}_c^2(\Omega, \mathbb{R}^{n \times n}),\ \|\xi\|_\infty \leq 1 \right\}.$$

Then, for every function $u \in W^{1,1}(\Omega)$ an integration by parts gives

$$\int_\Omega u \operatorname{div}^2 \xi \, dx = - \int_\Omega (\nabla u, \operatorname{div} \xi)_{\mathbb{R}^d} \, dx \ .$$

so that

$$\Phi_2(u) := \sup \left\{ \int_\Omega u \operatorname{div}^2 \xi \, dx, \xi \in \mathcal{K} \right\}. \tag{3.6}$$

3.1.2. The Total Generalized Variation

Another definition for second-order total variation spaces has been set in Bredies et al. (2010, 2011). The main difference lies in the choice of test functions in the variational formulation. The authors define the *Total Generalized Variation* $TGV^2(u)$ as the supremum of the duality product between u and symmetric tests functions that are bounded together with their derivative. Let be $\alpha = (\alpha_0, \alpha_1) > 0$, we call

$$TGV_\alpha^2(u) = \sup \left\{ \int_\Omega u \operatorname{div}^2 \xi \, dx, \xi \in \mathcal{K}_\alpha \right\},$$

where

$$\mathcal{K}_\alpha := \{ \xi \in \mathcal{K},\ \xi_{ij} = \xi_{ji} \ \ \forall i, j,\ \|\xi\|_\infty \leq \alpha_0,\ \|\operatorname{div} \xi\|_\infty \leq \alpha_1 \}.$$

The BGV_α^2 space is defined as following

$$BGV_\alpha^2(\Omega) = \left\{ u \in L^1(\Omega)\ , TGV_\alpha^2(u) < +\infty \right\}. \tag{3.7}$$

Recall that

$$BV^2(\Omega) := \{ u \in W^{1,1}(\Omega) \mid \Phi_2(u) < +\infty \},$$

where

$$\Phi_2(u) := \sup \left\{ \int_\Omega u \operatorname{div}^2 \xi \, dx, \xi \in \mathcal{K} \right\}.$$

These two spaces are different : indeed $BGV^2(\Omega)$ functions do not necessarily belong to $W^{1,1}(\Omega)$ so that $BGV^2(\Omega)$ includes less regular function than $BV^2(\Omega)$. More precisely :

Proposition 3.2. *Let be $\alpha = (\alpha_0, \alpha_1) > 0$. For every function u in $W^{1,1}(\Omega)$ we get*

$$TGV^2_\alpha(u) \leq \alpha_0 TV^2(u) \ .$$

Therefore

$$\forall \alpha > 0 \qquad BV^2(\Omega) \subset BGV^2_\alpha(\Omega)$$

with continuous embedding.
Proof - As $\mathcal{K}_\alpha \subset \mathcal{K}$ the first relation is obvious. Moreover if $u \in BV^2(\Omega)$, then $u \in W^{1,1}$ and $TGV^2_\alpha(u) < +\infty$. In addition

$$\|u\|_{BVG^2_\alpha} = \|u\|_{L^1} + TGV^2_\alpha(u) \leq \|u\|_{W^{1,1}} + \alpha_0 TV^2(u) \leq \max(1, \alpha_0)\|u\|_{BV^2} \ ,$$

which gives the embedding continuity. □

Corollary 3.1. *For $u \in BV^2(\Omega)$, $TV_2(u) = 0$ if and only if u is a polynomial function of order 1.*
Proof - For $u \in BV^2(\Omega)$, $TV_2(u) = 0 \Longrightarrow TGV^2_\alpha(u) = 0$. We use Proposition 3.3 of Bredies et al. (2010, 2011). □

3.2. A partial second order model

3.2.1. The ROF2 model

We now assume (as in the models of subsection 2.3.2) that the image we want to recover from the data u_d can be decomposed as $u_d = u + w$ where $u \in BV^2(\Omega)$ and $w := u_d - u \in L^2(\Omega)$. We consider the following cost functional defined on $BV^2(\Omega)$:

$$\mathcal{F}_2(u) = \frac{1}{2}\|u_d - u\|^2_{L^2(\Omega)} + \lambda \Phi_2(u), \tag{3.8}$$

where $\lambda > 0$. We are looking for a solution to the optimisation problem

$$\inf\{ \ \mathcal{F}_2(u) \mid u \in BV^2_0(\Omega) \ \} \tag{$\mathcal{P}_2$}$$

The first term $\|u_d - u\|^2_{L^2(\Omega)}$ of $\mathcal{F}_2$ is the fitting data term. Here we have chosen the L^2-norm for simplicity but any L^p norm can be used ($p \in [2, +\infty)$). Let us mention that Bredies et al. (2011) have investigated the very case where $p = 1$ with TGV_2 instead of Φ_2.

If the image is noisy, the noise is considered as a texture and will be involved in the remainder term $u_d - u$: more precisely u should be the part of the image without the oscillating component, that is the denoised part. Such an approach has already been used by Hinterberger and Scherzer (2006) with the $BV^2(\Omega)$ space. Their algorithm is different from the one we use here.

Theorem 3.4. *Assume that $\lambda > 0$. Problem $(\mathcal{P}_2)$ has at least a solution u.*

Proof - *Proof* - Let $u_n \in BV_0^2(\Omega)$ be a minimizing sequence, i.e.

$$\lim_{n\to+\infty} \mathcal{F}_2(u_n) = \inf(\mathcal{P}_2) < +\infty.$$

Therefore $\Phi_2(u_n)$ is bounded and with Lemma 3.1, $\|\nabla u_n\|_{L^1} = \Phi_1(u_n)$ is bounded as well. As u_n is L^2 -bounded, it is L^1 -bounded as well. This yields that u_n is bounded in $W^{1,1}(\Omega)$. Therefore the sequence u_n is bounded in $BV^2(\Omega)$.

With the compactness result of Theorem 3.3, we deduce that $(u_n)_{n\in\mathbb{N}}$ strongly converges (up to a subsequence) in $W^{1,1}(\Omega)$ to $u^* \in BV_0^2\Omega)$(because the trace operator is continuous). With theorem 3.2 we get

$$\Phi_2(u^*) \leq \liminf_{n\to+\infty} \Phi_2(u_n).$$

So

$$\mathcal{F}_2(u^*) \leq \liminf_{n\to+\infty} \mathcal{F}_2(u_n) = \inf(\mathcal{P}_2),$$

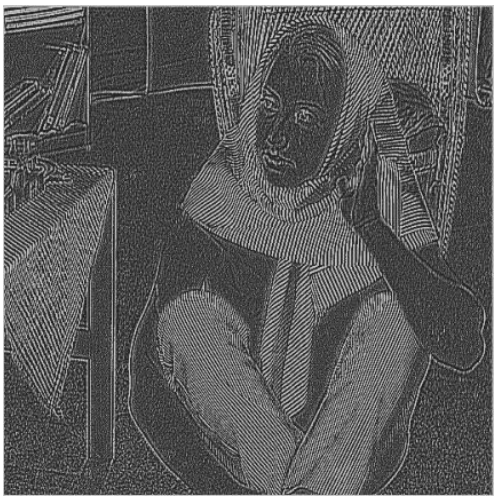
and u^* is a solution to $(\mathcal{P}_2)$. □

3.2.2. Anisotropic improvment (Piffet (2011))

We observe (see section 3.5) that the second order model $(\mathcal{P}_2)$ removes the staircasing effect. However, as the solution is close to a piecewise affine one, the model generates a blur effect on the BV^2 -part. This means that contour lines are still partly involved in the oscillating component. As a result, this decomposition model is not efficient for texture extraction. To improve the result, a local modification of the Hessian operator is performed and a local anisotropic strategy is performed, which depends on each pixel and is consistent with the contours. We have noticed that cancelling one or more coefficients of the (local) Hessian matrix permits to get rid of the contours along the corresponding direction. In Figure 3.4 the coefficients $(Hv)^{1,1}$ and $(Hv)^{2,2}$ of the Hessian matrix have been globally set to 0. We can see that horizontal and vertical contours are not involved in the texture part any longer. However, this method has to been improved since there are two major inconveniences :

- First, the same transform is performed at every pixel, so that the image is globally treated. All the vertical and horizontal lines are removed;
- Second, the transform depends on the chosen (fixed) cartesian axis and it is not possible to remove contours that are not horizontal, vertical or diagonal.

(a) Original image (Barbara)



(b) Texture part without anisotropic strategy

(c) Texture part without horizontal and vertical contours

Figure 3.4: Effects of anisotropic improvement strategy -Piffet (2011)

Therefore, we perform a local rotation which is driven by the gradient direction, to make the contour direction, horizontal (or vertical). Then we cancel the corresponding terem in the new rotated Hessian matrix. The whole process is detailed in (section 4) .

(a) (b)

Figure 3.5: Illustration of the method with "Barbara" example

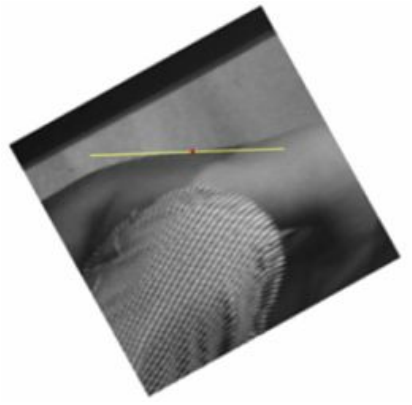
The different steps are the following :

- **Step 1.** Detect points of interest which are pixels of contours that appear in the texture component and that need to be removed. This step can be performed using (for example) a thresholding on the image gradient norm. The other pixels are treated with the original model ($\mathcal{P}_2$) without any anisotropic strategy.

Figure 3.6: Pixels of interest which are concerned by the anisotropic strategy.

- **Step 2.** Compute the image gradient at every point of interest: this gives the angle α between the normal direction at the point and (for example) the horizontal line.

(a) Choice of a significant pixel



(b) Rotation of the thumbnail

Figure 3.7: Once the angle α between the direction of the contour (given by the gradient) and the horizontal direction is perform a rotation and set the Hessian component that correspond to the gradient diection to 0 ($H^{2,2}$ for example)

- **Step 3.** Extract a neighborhood (patches of size $p \times p$) centered at every interest point, and perform a rotation of angle α. Then, compute the new Hessian matrix at the considered pixel, setting either the horizontal or vertical component to 0. Large enough patches must be considered to avoid boundary effects (for example $p = 5$).

Next figures illustrates the result.

(a) Texture part obtained with ($\mathcal{P}_2$) model

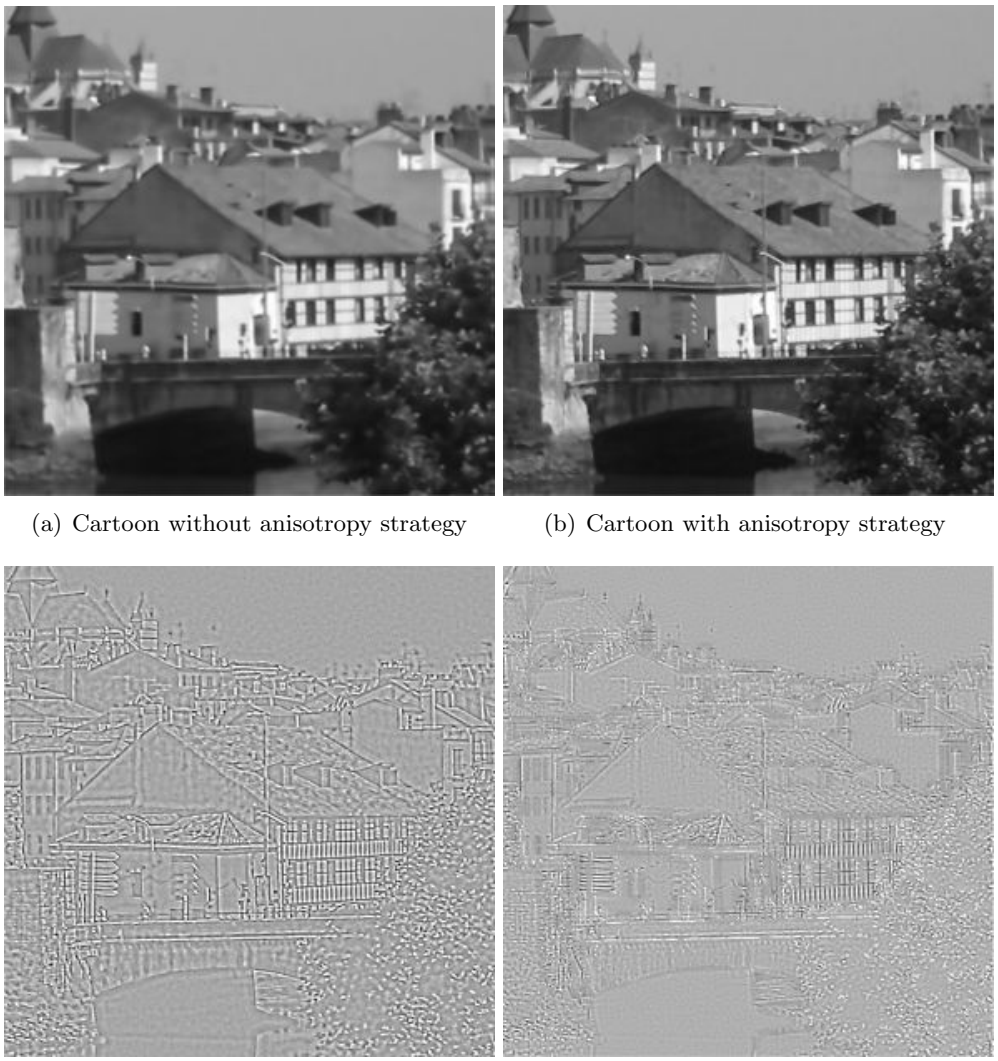
(b) Texture part obtained with locally anisotropic strategy

Figure 3.8: Comparison between ($\mathcal{P}_2$) model and ($\mathcal{P}_2$) with a local anisotropic strategy.

We shall detail the strategy and the implementation in the 3D case (section 4) but we give an example however.

Figure 3.9: Original image

We observe in Figures 3.10 and 3.11 that the contour lines of the BV^2-component are well preserved using the anisotropic strategy, so that we may use the concept of *cartoon* t as for the ROF model (see Aujol et al. (2005)). Moreover, we can see on the texture component that contours and edges disappear when using anisotropy strategy. Even if we can notice that the locally anisotropic model gives pretty good results for texture extraction, we still have to carefully analyze this strategy, which is related to the penalization of image curvature. Additional comments, details and examples can be found in Piffet (2011).



(a) Cartoon without anisotropy strategy (b) Cartoon with anisotropy strategy

(c) Texture without anisotropy strategy (d) Texture with anisotropy strategy

Figure 3.10: ($\mathcal{P}_2$) model with and without anisotropy strategy - $\lambda = 20$

(a) Cartoon without anisotropy strategy (b) Cartoon with anisotropy strategy

Figure 3.11: ($\mathcal{P}_2$) model with and without anisotropy strategy - $\lambda = 100$ - Zoom on cartoons

3.3. Numerical experiments

3.3.1. Discretization of problem ($\mathcal{P}_2$)

We assume once again that the image is squared with size $N \times M$. We note $X := \mathbb{R}^{N\times M} \simeq \mathbb{R}^{NM}$ endowed with the usual inner product and the associated euclidean norm and use the discretization process of section 2.4.1. To define a discrete version of the second order total variation Φ_2 we have to introduce the discrete Hessian operator. For any $v \in X$, the Hessian matrix of v, denoted Hv is identified to a X^4 vector: $(Hv)_{i,j} = \left((Hv)^{11}_{i,j}, (Hv)^{12}_{i,j}, (Hv)^{21}_{i,j}, (Hv)^{22}_{i,j}\right)$, with, for every $i = 1, \dots, N,\ j = 1, \dots, M$

$$(Hv)^{11}_{i,j} = \begin{cases} v_{i+1,j} - 2v_{i,j} + v_{i-1,j} & \text{if } 1 < i < N, \\ v_{i+1,j} - v_{i,j} & \text{if } i = 1, \\ v_{i-1,j} - v_{i,j} & \text{if } i = N, \end{cases}$$

$$(Hv)^{12}_{i,j} = \begin{cases} v_{i,j+1} - v_{i,j} - v_{i-1,j+1} + v_{i-1,j} & \text{if } 1 < i \leq N,\ 1 \leq j < M, \\ 0 & \text{if } i = 1, \\ 0 & \text{if } i = N, \end{cases}$$

$$(Hv)^{21}_{i,j} = \begin{cases} v_{i+1,j} - v_{i,j} - v_{i+1,j-1} + v_{i,j-1} & \text{if } 1 \leq i < N,\ 1 < j \leq M, \\ 0 & \text{if } i = 1, \\ 0 & \text{if } i = N, \end{cases}$$

$$(Hv)^{22}_{i,j} = \begin{cases} v_{i,j+1} - 2v_{i,j} + v_{i,j-1} & \text{if } 1 < j < M, \\ v_{i,j+1} - v_{i,j} & \text{if } j = 1, \\ v_{i,j-1} - v_{i,j} & \text{if } j = M. \end{cases}$$

The discrete second order total variation corresponding to $\Phi_2(v)$ is defined as

$$J_2(v) = \sum_{i=1}^{N} \sum_{j=1}^{M} \|(Hv)_{i,j}\|_{\mathbb{R}^4}, \tag{3.9}$$

where $\|x\|_{\mathbb{R}^4} = \sqrt{\sum_{i=1}^{4} x_i^2}$ for every $x = (x_1, x_2, x_3, x_4) \in \mathbb{R}^4$. The discretized problem writes

$$\inf_{u \in X} F_2(u) := \frac{1}{2}\|u_d - u\|_X^2 + \lambda J_2(u). \tag{P_2}$$

Theorem 3.5. *Problem* (P_2) *has a unique solution for every* $\lambda > 0$.

Proof - The cost functional F_2 is continuous and coercive because of the term $\|u_d - v\|_X^2$. In addition it is strictly convex so that we get the result. □

3.3.2. Optimality conditions

We follow the steps of section 2.5 to get optimality conditions for the solution to (P_2). For the sake of simplicity, we perform the study in the finite dimensional case only.

We first compute the Legendre-Fenchel conjugate function of J_2. As J_2 is positively homogeneous, the Legendre-Fenchel conjugate J_2^* is the characteristic function of a closed, convex set K. As $J_2^{**} = J_2$, we get

$$J_2(v) = \sup_{u \in K} \langle v, u \rangle_X .$$

We use the inner scalar product of X^4 :

$$\langle p, q \rangle_{X^4} = \sum_{1 \le i \le N} \sum_{1 \le j \le M} \left(p_{i,j}^1 q_{i,j}^1 + p_{i,j}^2 q_{i,j}^2 + p_{i,j}^3 q_{i,j}^3 + p_{i,j}^4 q_{i,j}^4 \right),$$

for every $p = \left(p^1, p^2, p^3, p^4\right)$, $q = \left(q^1, q^2, q^3, q^4\right) \in X^4$. So, for every $v \in X$,

$$J_2(v) = \sup_{p \in \mathcal{C}} \langle p, Hv \rangle_{X^4}, \tag{3.10}$$

where the feasible set is

$$\mathcal{C} := \{\, p \in X^4 \mid \|p_{i,j}\|_{\mathbb{R}^4} \le 1, \ \forall\, 1 \le i \le N, 1 \le j \le M \,\} .$$

Let us compute the adjoint operator of H (which is the discretized "second divergence" operator) :

$$\forall p \in X^4, \ \forall v \in X \qquad \langle H^* p, v \rangle_X = \langle p, Hv \rangle_{X^4}.$$

We verify that $H^* : X^4 \to X$ satisfies for every $p = (p^{11}, p^{12}, p^{21}, p^{22}) \in X^4$

$$
\begin{aligned}
(H^*p)_{i,j} \;=\; & \begin{cases} p^{11}_{i-1,j} - 2p^{11}_{i,j} + p^{11}_{i+1,j} & \text{if } \; 1 < i < N \\ p^{11}_{i+1,j} - p^{11}_{i,j} & \text{if } \; i = 1, \\ p^{11}_{i-1,j} - p^{11}_{i,j} & \text{if } \; i = N, \end{cases} \\
+ \; & \begin{cases} p^{22}_{i,j-1} - 2p^{22}_{i,j} + p^{22}_{i,j+1} & \text{if } \; 1 < j < M, \\ p^{22}_{i,j+1} - p^{22}_{i,j} & \text{if } \; j = 1, \\ p^{22}_{i,j-1} - p^{22}_{i,j} & \text{if } \; j = M, \end{cases} \\
+ \; & \begin{cases} p^{12}_{i,j-1} - p^{12}_{i,j} & \\ \quad -p^{12}_{i+1,j-1} + p^{12}_{i+1,j} & \text{if } \; 1 < i < N, \; 1 < j < M, \\ p^{12}_{i+1,j} - p^{12}_{i+1,j-1} & \text{if } \; i = 1, \; 1 < j < M, \\ p^{12}_{i,j-1} - p^{12}_{i,j} & \text{if } \; i = N, \; 1 < j < M, \\ p^{12}_{i+1,j} - p^{12}_{i,j} & \text{if } \; 1 < i < N, \; j = 1, \\ p^{12}_{i,j-1} - p^{12}_{i+1,j-1} & \text{if } \; 1 < i < N, \; j = M, \\ p^{12}_{i+1,j} & \text{if } \; i = 1, \; j = 1, \\ -p^{12}_{i+1,j-1} & \text{if } \; i = 1, \; j = M, \\ -p^{12}_{i,j} & \text{if } \; i = N, \; j = 1, \\ p^{12}_{i,j-1} & \text{if } \; i = N, \; j = M, \end{cases} \\
+ \; & \begin{cases} p^{21}_{i-1,j} - p^{21}_{i,j} & \\ \quad -p^{21}_{i-1,j+1} + p^{21}_{i,j+1} & \text{if } \; 1 < i < N, \; 1 < j < M, \\ p^{21}_{i,j+1} - p^{21}_{i,j} & \text{if } \; i = 1, \; 1 < j < M, \\ p^{21}_{i-1,j} - p^{21}_{i-1,j+1} & \text{if } \; i = N, \; 1 < j < M, \\ p^{21}_{i,j+1} - p^{21}_{i-1,j+1} & \text{if } \; 1 < i < N, \; j = 1, \\ p^{21}_{i-1,j} - p^{21}_{i,j} & \text{if } \; 1 < i < N, \; j = M, \\ p^{21}_{i,j+1} & \text{if } \; i = 1, \; j = 1, \\ -p^{21}_{i,j} & \text{if } \; i = 1, \; j = M, \\ -p^{21}_{i-1,j+1} & \text{if } \; i = N, \; j = 1, \\ p^{21}_{i-1,j} & \text{if } \; i = N, \; j = M, \end{cases}
\end{aligned}
\tag{3.11}
$$

Finally, we obtain

Theorem 3.6. *The Legendre-Fenchel conjugate of J_2 is $J_2^* = \mathbf{1}_{K_2}$ where*

$$K_2 := \{H^*p \mid p \in X^4, \; \|p_{i,j}\|_{\mathbb{R}^4} \leq 1, \; \forall 1 \leq i \leq N, \; 1 \leq j \leq M\} \subset X. \tag{3.12}$$

Proof - The proof is the same as the one of theorem 2.4. Alternatively, one may note that J_2 is the support function of K_2 which is the conjugate function of the indicator function 1_{K_2} of K_2 (see Ekeland and Temam (1999) p. 19). Therefore, as K_2 is closed and convvex and J_2 is continuous we get $J_2^* = 1_{K_2}^{**} = 1_{K_2}$. □

Eventually, we get

Theorem 3.7. *The solution $\bar{u}$ of (P_2) verifies*

$$\bar{u} = u_d - P_{\lambda K_2}(u_d),$$

where $P_{\lambda K_2}$ is the orthogonal projector operator on λK_2.

3.3.3. A fixed-point algorithm to compute $P_{\lambda K_2}$

We extend the result of (Chambolle (2004)) that we recalled in section 2.4.2, to the second-order case. To compute $P_{\lambda K_2}(u_d)$ we have to solve

$$\min \left\{ \|\lambda H^* p - u_d\|_X^2 \mid p \in X^4,\ \|p_{i,j}\|_{\mathbb{R}^4}^2 - 1 \leq 0,\ 1 \leq i \leq N,\ 1 \leq j \leq M \right\}.$$

Let us denote $R(p) = \|\lambda H^* p - u_d\|_X^2$ and

$$g_{i,j}(p) = \|p_{i,j}\|_{\mathbb{R}^4}^2 - 1 = (p_{i,j}^{11})^2 + (p_{i,j}^{12})^2 + (p_{i,j}^{21})^2 + (p_{i,j}^{22})^2 - 1.$$

First order optimality conditions give the existence of Lagrange multipliers $\alpha_{i,j}$, $(i,j) \in \{1,\dots,N\} \times \{1,\dots,M\}$, such that

$$\nabla R(p) + \sum_{i=1}^{N} \sum_{j=1}^{M} \alpha_{i,j} \nabla g_{i,j}(p) = 0, \tag{3.13a}$$

$$\alpha_{i,j} \geq 0 \ \text{ and } \ \alpha_{i,j} g_{i,j}(p) = 0,\ 1 \leq i \leq N, 1 \leq j \leq M. \tag{3.13b}$$

It is easy to see that $\nabla R(p) = 2\lambda H\left[\lambda H^* p - u_d\right]$ and that

$$\sum_{i=1}^{N} \sum_{j=1}^{M} \alpha_{i,j} \nabla g_{i,j}(p) = 2\alpha_{i,j} \left((p_{i,j}^{11}, p_{i,j}^{22}, p_{i,j}^{12}, p_{i,j}^{21}) \right)_{1 \leq i \leq N, 1 \leq j \leq M}.$$

Therefore relations (3.13) are equivalent to

$$\forall (i,j) \in \{1,\dots,N\} \times \{1,\dots,M\} \qquad \left(H\left[\lambda H^* p - u_d\right]\right)_{i,j} + \alpha_{i,j} p_{i,j} = 0, \tag{3.14a}$$

$$\forall (i,j) \in \{1,\dots,N\} \times \{1,\dots,M\} \qquad \alpha_{i,j} \geq 0 \ \text{ and } \ \alpha_{i,j} g_{i,j}(p) = 0. \tag{3.14b}$$

Let us compute the multipliers $\alpha_{i,j}$ more precisely :

- If $\alpha_{i,j} > 0$ then $\|p_{i,j}\|_{\mathbb{R}^4} = 1$.

- If $\alpha_{i,j} = 0$ then $\left(H\left[\lambda H^* p - u_d\right]\right)_{i,j} = 0$.

In both cases we get

$$\forall (i,j) \in \{1,\dots,N\} \times \{1,\dots,M\} \qquad \alpha_{i,j} = \left\| \left(H\left[\lambda H^* p - u_d\right]\right)_{i,j} \right\|_{\mathbb{R}^4}$$

and we finally obtain the following equality : $\forall (i,j) \in \{1,\dots,N\} \times \{1,\dots,M\}$,

$$\left(H\left[\lambda H^* p - u_d\right]\right)_{i,j} + \left\| \left(H\left[\lambda H^* p - u_d\right]\right)_{i,j} \right\|_{\mathbb{R}^4} p_{i,j} = 0. \tag{3.15}$$

We use a semi-implicit gradient method to solve these equations : this gives

Algorithm 4 Second order Chambolle-type algorithm

Initialization : $n = 0$; $p^0 = 0$

Iteration n : set

$$p_{i,j}^{n+1} = \frac{p_{i,j}^n - \tau \left(H\left[H^* p^n - u_d/\lambda\right]\right)_{i,j}}{1 + \tau \left\| \left(H\left[H^* p^n - u_d/\lambda\right]\right)_{i,j} \right\|_{\mathbb{R}^4}} \ . \tag{3.16}$$

Stopping criterion.

The algorithm step $\tau > 0$ is related to the adjoint operator H^* norm that we call κ in the sequel. We first give a κ estimate:

Lemma 3.2. *The adjoint operator H^* norm, κ satisfies $\kappa \leq 8$.*

Proof - The definition of κ gives : $\kappa = \sup_{\|p\|_{X^4} \leq 1} \|H^* p\|_X$. As

$$\|H^* p\|_X = \sup_{q \in \bar{\mathcal{B}}_X(0,1)} \langle H^* p, q \rangle_X = \sup_{q \in \bar{\mathcal{B}}_X(0,1)} \langle p, Hq \rangle_{X^4} \leq \sup_{q \in \bar{\mathcal{B}}_X(0,1)} \|Hq\|_{X^4} \|p\|_{X^4},$$

we get

$$\|H^* p\|_X \leq |||H||| \ \|p\|_{X^4}, \tag{3.17}$$

where

$$|||H||| = \sup_{\|q\|_X \leq 1} \|Hq\|_{X^4} \ .$$

For any $q \in X$

$$
\begin{aligned}
\|Hq\|_{X^4}^2 = \sum_{i=1}^{N}\sum_{j=1}^{M} \Big[& (q_{i+1,j} - 2q_{i,j} + q_{i-1,j})^2 + (q_{i,j+1} - q_{i,j} - q_{i-1,j+1} + q_{i-1,j})^2 \\
& + (q_{i+1,j} - q_{i,j} - q_{i+1,j-1} + q_{i,j-1})^2 + (q_{i,j+1} - 2q_{i,j} + q_{i,j-1})^2 \Big] \\
\leq 4 \sum_{i=1}^{N}\sum_{j=1}^{M} & [q_{i+1,j}^2 + q_{i,j}^2 + q_{i,j}^2 + q_{i-1,j}^2 + q_{i,j+1}^2 + q_{i,j}^2 + q_{i-1,j+1}^2 + q_{i-1,j}^2 \\
& + q_{i+1,j}^2 + q_{i,j}^2 + q_{i+1,j-1}^2 + q_{i,j-1}^2 + q_{i,j+1}^2 + q_{i,j}^2 + q_{i,j}^2 + q_{i,j-1}^2] \\
\leq 4 \times 16 \ \|q\|_X^2 = 64 \ \|q\|_X^2 .
\end{aligned}
$$

Finally $|||H||| \leq 8$, and with relation (3.17), $\|H^* p\|_X \leq 8 \ \|p\|_{X^4}$. We deduce that $\kappa \leq 8$. □

Theorem 3.8. *Let be* $\tau \leq 1/64$. *Then* $\lambda \left(H^* p^n\right)_n$ *converges to* $P_{\lambda K_2}(u_d)$ *as* $n \to +\infty$.

Proof - We refer to Bergounioux and Piffet (2010). □

3.3.4. Nesterov type algorithms

Algorithm 2 is a generic one. As in section 2.4.3, we apply it to solving the dual problem. We set

$$
E(p) = \frac{1}{2}\|H^* p - u_d\|_X^2 \text{ and } Q = \mathcal{B}_\lambda,
$$

and choose $d(x) = \dfrac{1}{2}\|x\|_X^2$ with $x_0 = 0$ and $\sigma = 1$.

Algorithm 5 Modified Nesterov algorithm fo (P_2)

Input : the maximal number of iterations I_{max} and an initial guess $p_0 \in \mathcal{B}_\lambda$ are given.

Output : $\tilde{q} := q_{I_{max}}$ approximates $\bar{q}$ solution to (2.15)

Set $L = \|H^*\|^2$.
Set $G_{-1} = 0$
for $0 \leq k \leq I_{max}$ **do**
$\eta_k = H(H^* p_k - u_d)$
$q_k = \Pi_{\mathcal{B}_\lambda}\left(p_k - \dfrac{\eta_k}{L}\right)$.
$G_k = G_{k-1} + \dfrac{k+1}{2}\eta_k,\ z_k = \Pi_{\mathcal{B}_\lambda}\left(-\dfrac{G_k}{L}\right)$.
$p_{k+1} = \dfrac{2}{k+3} z_k + \dfrac{k+1}{k+3} q_k$
end for

3.4. A full second order model

The variational model we have previously studied, involves a single second order term Φ_2. The motivation was to get rid of the staircasing effect while restoring noisy data. We infered that the use of a second order penalization term leads to piecewise affine solutions so that there is no staircasing any longer. However, we observed that the contours were not kept as well as we wanted and that the resulting image was slightly blurred. To overcome this difficulty, we now consider a full second order model involving both first and second order penalization terms. Furthermore, we focus on texture extraction; indeed denoising can be handled in a similar way, considering that noise is a very fine texture.

3.4.1. The model

Specifically, we assume that the image we want to recover from data can be decomposed as $u_d = w + u + v$ where u, v and w are functions that characterize the various structures of u_d. In the sequel $u_d \in L^2(\Omega)$. We consider the following cost functional defined on $BV(\Omega) \times BV^2(\Omega)$:

$$\mathcal{F}_{\lambda,\mu}(u,v) = \frac{1}{2}\|u_d - u - v\|^2_{L^2(\Omega)} + \lambda\Phi_1(u) + \mu\Phi_2(v), \tag{3.18}$$

where $\lambda, \mu > 0$. We are looking for a solution to the optimization problem

$$\inf\{\ \mathcal{F}_{\lambda,\mu}(u,v) \mid (u,v) \in \mathcal{X} \times \mathcal{Y}\ \} \tag{$\mathcal{P}_{\lambda,\mu}$}$$

where $\mathcal{X} = BV_0(\Omega)$ or $BV_m(\Omega)$ and $\mathcal{Y} = BV_0^2(\Omega)$ or $BV_m^2(\Omega)$. In other words we expect

- v to be the smooth *colored* part of the image (that should be piecewise affine),
- u to be a $BV(\Omega)\backslash BV^2(\Omega)$ function which derivative is a measure supported by the image contours,
- $w := u_d - u - v \in L^2$ is the noise and/or fine textures (we detail this point later).

First we give an existence result for problem $(\mathcal{P}_{\lambda,\mu})$.

Theorem 3.9 (Existence). *Assume that $\lambda > 0$ and $\mu > 0$. The problem $(\mathcal{P}_{\lambda,\mu})$ has at least an optimal solution $(u^*, v^*) \in \mathcal{X} \times \mathcal{Y} \subset BV(\Omega) \times BV^2(\Omega)$.*

Proof - Let $(u_n, v_n) \in BV_0(\Omega) \times BV_0^2(\Omega)$ be a minimizing sequence, i.e.

$$\lim_{n\to+\infty} \mathcal{F}_{\lambda,\mu}(u_n, v_n) = \inf(\mathcal{P}_{\lambda,\mu}) < +\infty.$$

Therefore

- $\Phi_2(v_n)$ is bounded and with Lemma 3.1, $\|\nabla v_n\|_{L^1}$ is bounded as well.
- $\Phi_1(u_n)$ is bounded. Using once again Lemma 3.1 this yields that u_n is bounded in $L^1(\Omega$. Therefore the sequence u_n is bounded in $BV(\Omega)$.
- As u_n+v_n is L^2 -bounded, it is L^1 -bounded as well so that v_n is L^1 bounded. As $\|\nabla v_n\|_{L^1}$ and $\Phi_2(v_n)$ are bounded this means that the sequence v_n is bounded in $BV^2(\Omega)$.

With the compactness result of Theorem 3.3, we infer that $(v_n)_{n\in\mathbb{N}}$ strongly converges (up to a subsequence) in $W^{1,1}(\Omega)$ to $v^* \in BV_0^2\Omega)$(because the trace operator is continuous) and $(u_n)_{n\in\mathbb{N}}$ strongly converges (up to a subsequence) in $L^1(\Omega)$ to $u^* \in BV_0(\Omega)$. Moreover $u_n + v_n$ weakly converges to $u^* + v^*$ in $L^2(\Omega)$. With Theorem 3.2 we get

$$\Phi_1(u^*) \leq \liminf_{n\to+\infty} \Phi_1(u_n),\ \Phi_2(v^*) \leq \liminf_{n\to+\infty} \Phi_2(v_n).$$

So

$$\mathcal{F}_{\lambda,\mu}(u^*, v^*) \leq \liminf_{n\to+\infty} \mathcal{F}_{\lambda,\mu}(u_n, v_n) = \min(\mathcal{P}_{\lambda,\mu}),$$

and (u^*, v^*) is a solution to $(\mathcal{P}_{\lambda,\mu})$. □

It is easy to see that (u^*, v^*) is a solution to $(\mathcal{P}_{\lambda,\mu})$ if and only if

$$\begin{aligned} u^* &= \operatorname{argmin}\left\{\frac{1}{2}\|\mathrm{u_d} - \mathrm{v}^* - \mathrm{u}\|^2 + \lambda\Phi_1(\mathrm{u})\ , \mathrm{u} \in \mathrm{BV_0}(\Omega)\right\}, \\ v^* &= \operatorname{argmin}\left\{\ \frac{1}{2}\|\mathrm{u_d} - \mathrm{u}^* - \mathrm{v}\|^2 + \mu\Phi_2(\mathrm{v}), \mathrm{v} \in \mathrm{BV_0^2}(\Omega)\right\}. \end{aligned} \tag{3.19}$$

and we may derive optimality conditions in a standard way :

Theorem 3.10. *(u^*, v^*) is a solution to $(\mathcal{P}_{\lambda,\mu})$ if and only if*

$$u_d - u^* - v^* \in \lambda \partial \Phi_1(u^*), \tag{3.20a}$$

$$u_d - u^* - v^* \in \mu \partial \Phi_2(v^*). \tag{3.20b}$$

The proof is straightforward since Φ_1 and Φ_2 are convex and continuous and variables u and v can be decoupled.

3.4.2. Numerical realization and algorithm

We use the same discretization process as in the previous section. The discretized problem writes

$$\inf_{(u,v)\in X\times X} F_{\lambda,\mu}(u,v) := \frac{1}{2}\|u_d - u - v\|_X^2 + \lambda J_1(u) + \mu J_2(v). \tag{$P_{\lambda,\mu}$}$$

Theorem 3.11. *Assume $\lambda > 0, \mu > 0$. Problem ($P_{\lambda,\mu}$) has a unique solution (u^*, v^*) .*

Proof - The proof is obvious since the cost functional is strictly convex and coercive. □

Using the subdifferential properties and decoupling u^* and v^* gives the following necessary and sufficient optimality conditions :

Proposition 3.3. *(u^*, v^*) is a solution to $(P_{\lambda,\mu})$ if and only if*

$$u_d - u^* - v^* \in \lambda \partial J_1(u^*), \tag{3.21a}$$

$$u_d - u^* - v^* \in \mu \partial J_2(v^*). \tag{3.21b}$$

We can perform an explicit computation to get the following result :

Theorem 3.12. *(u^*, v^*) is a solution to $(P_{\lambda,\mu})$ if and only if*

$$u^* = u_d - v^* - \Pi_{\lambda K_1}(u_d - v^*) \ , \tag{3.22a}$$

$$v^* = u_d - u^* - \Pi_{\mu K_2}(u_d - u^*), \tag{3.22b}$$

where K_1 and K_2 are the following convex closed subsets :

$$K_1 = \{\text{div } p \mid p \in X^2,\ \|p_{i,j}\|_{\mathbb{R}^2} \leq 1\ \forall i = 1,\dots,N, j = 1,\dots,M\}, \tag{3.23a}$$

$$K_2 = \{H^* p \mid p \in X^4,\ \|p_{i,j}\|_{\mathbb{R}^4} \leq 1,\ \forall i = 1,\dots,N, j = 1,\dots,M\}, \tag{3.23b}$$

and Π_{K_i} denotes the orthogonal projection on K_i.

Proof - We refer to Bergounioux and Piffet (2010), Chambolle (2004). We use Theorems 2.7 and 3.7. □

We may write relations (3.22) as a fixed point equation $(u,v) = G(u,v)$, where

$$\begin{array}{rcl} G : X^2 & \to & X^2 \\ (u,v) & \mapsto & \left(\begin{array}{c} u_d - v - \mathit{\Pi}_{\lambda K_1}(u_d - v) \\ (u_d - u - \mathit{\Pi}_{\mu K_2}(u_d - u)) \end{array} \right). \end{array} \tag{3.24}$$

We use a gradient-type algorithm to compute the solution: for every $\alpha > 0$, set

$$\begin{pmatrix} u_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} u_n \\ v_n \end{pmatrix} + \alpha \left(G(u_n, v_n) - \begin{pmatrix} u_n \\ v_n \end{pmatrix} \right).$$

This leads to the following :

Algorithm 6 Fixed-point algorithm for $(P_{\lambda,\mu})$

Initialization step : Choose u_0 and v_0 (for example $u_0 = 0$ and $v_0 = u_d$) and $0 < \alpha < 1/2$. Set $n = 0$.

Iteration n : Define the sequences $(u_n, v_n)_{n\in\mathbb{N}}$ as

$$\begin{cases} u_{n+1} = u_n + \alpha\left(u_d - v_n - \mathit{\Pi}_{\lambda K_1}(u_d - v_n) - u_n\right) \\ v_{n+1} = v_n + \alpha\left(u_d - u_n - \mathit{\Pi}_{\mu K_2}(u_d - u_n) - v_n\right). \end{cases}$$

Stopping criterion.

We may give a convergence result :

Theorem 3.13. *If $\alpha > 0$ is small enough, the sequence (u_n, v_n) converges to the (unique) fixed point of G.*

Proof - The above algorithm is a descent method with step α and direction $D(u,v) = (u,v) - G(u,v)$. We have to prove that $D = (D_1, D_2)$ is Lipschitz continuous. We set $S_1(w) := w - \Pi_{\lambda K_1}(w)$ so that $D_1(u,v) = u - S_1(u_d - v)$. For every $(w_1, w_2), \in X^2$, we have

$$\begin{aligned} \|S_1(w_1) - S_1(w_2)\|_X^2 = \|w_2 - w_1\|_X^2 &+ \|\Pi_{\lambda K_1}(w_2) - \Pi_{\lambda K_1}(w_1)\|_X^2 \\ &+ 2\left(w_1 - w_2, \Pi_{\lambda K_1}(w_2) - \Pi_{\lambda K_1}(w_1)\right) . \end{aligned}$$

With Proposition A.1 (Appendix A.1.4) we get

$$\begin{aligned} \|S_1(w_1) - S_1(w_2)\|_X^2 &\le \|w_2 - w_1\|_X^2 - \|\Pi_{\lambda K_1}(w_2) - \Pi_{\lambda K_1}(w_1)\|_X^2 \\ &< \|w_2 - w_1\|_X^2 = \|v_2 - v_1\|_X^2. \end{aligned}$$

Therefore : $\|D_1(u_1,v_1)-D_1(u_2,v_2)\|_X < \|u_2-u_1\|_X + \|v_2-v_1\|_X$. A similar computation gives $\|D_2(u_1,v_1)-D_2(u_2,v_2)\|_X < \|u_2-u_1\|_X + \|v_2-v_1\|_X$, and finally

$$\|D(u_1,v_1)-D(u_2,v_2)\|_{1,X} < 2(\|u_1-u_2\|_X + \|v_1-v_2\|_X).$$

Therefore D is 2-Lipschitz continuous and we use a general convergence result to conclude. □

For the numerical realization a (standard) relaxed version of the algorithm is used.

Algorithm 7 Relaxed fixed-point algorithm for $(P_{\lambda,\mu})$

Initialization step : Choose u_0 and v_0 (for example $u_0 = 0$ and $v_0 = u_d$) and $0 < \alpha < 1/2$. Set $n = 0$.

Iteration n : Define the sequences $((u_n,v_n))_n$ as

$$\begin{cases} u_{n+1} = u_n + \alpha\left(u_d - u_n - v_n - \Pi_{\lambda K_1}\left(u_d - v_n\right)\right) \\ v_{n+1} = v_n + \alpha\left(u_d - u_{n+1} - v_n - \Pi_{\mu K_2}\left(u_d - u_{n+1}\right)\right). \end{cases}$$

Stopping criterion.

We perform the computation of projections $\Pi_{\lambda K_1}$ and $\Pi_{\mu K_2}$ using (for example) Algorithms 3 and 5.

3.5. Numerical results

We performs numerical tests to investigate the behavior of the partial second order model $(\mathcal{P}_2)$ and the full second order $(\mathcal{P}_{\lambda,\mu})$, then compare to the basic first order model $(\mathcal{P}_1)$. The results we report here can be found in Bergounioux and Piffet (2010, 2013). Numerical computation has been done using Nesterov-type Algorithms 3 and 5.

3.5.1. Denoising

Throughout this section, we consider the following image that is corrupted by a white Gaussian noise with standard deviation $\sigma = 0.15$ or $\sigma = 0.25$. We report on $(\mathcal{P}_2)$ which is rather a denoising model. We shall report on $(\mathcal{P}_{\lambda,\mu})$ in next subsection (devoted to texture extraction). The stopping criterion has been set to a maximal number of iterations that can be chosen arbitrary large.

(a) Original (b) Noisy image $\sigma = 0.15$ (c) Noisy image $\sigma = 0.25$

Figure 3.12: Test images

Sensibility with respect to λ parameter . We note that we lose details information when parameter λ increases, what was expected. However, especially when the data is very noisy, we have a blur (subjective) feeling, that we do not have when restoration is performed with the standard ROF model.

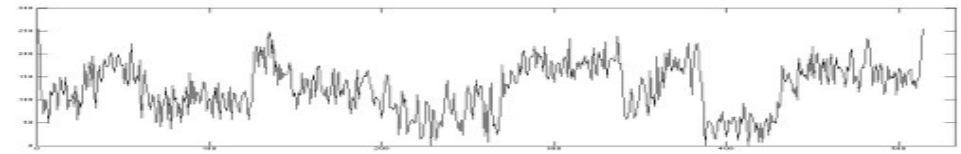
(a) $\lambda = 1$ (b) $\lambda = 5$ (c) $\lambda = 15$

(d) $\lambda = 25$ (e) $\lambda = 100$ (f) $\lambda = 500$

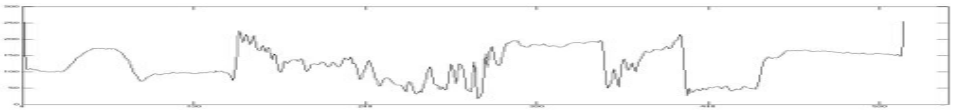
Figure 3.13: Solution - Standard deviation $\sigma = 0.15$

As expected, we see on Figure 3.13 that the smoothing process is more efficient when λ is large. Checking what happens precisely on slices (lines) of the image (Figure 3.15 for example), we remark that the $(\mathcal{P}_2)$ model keeps contour information pretty well, anyway better than expected watching the image.

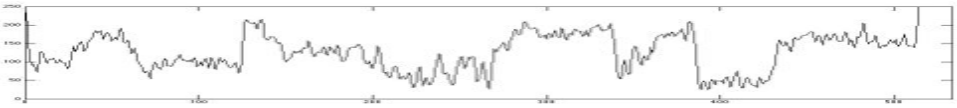
Sensitivity with respect to iterations number `itmax` *in Algorithm 5.* We fix $\lambda = 25$ and choose $\sigma = 0.25$.



(a) Noisy slice ($\sigma = 0.25$)



(b) Original slice



(c) 5 iterations

(d) 10 iterations

(e) 20 iterations

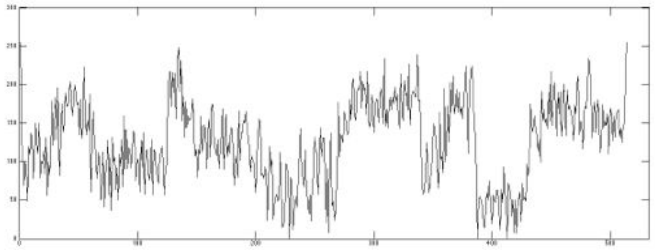
(f) 50 iterations

Figure 3.14: Sensitivity with respect to the number of iterations - $\sigma = 0.25$, $\lambda = 25$ - Slice of "Lena" image.

Figure 3.14 shows the behavior of a slice (line) during iterations (we can see more easily how noise is removed). The algorithm converges well: the quality of restoration is improved as the number of iterations grows. Noise is removed and contours are preserved.

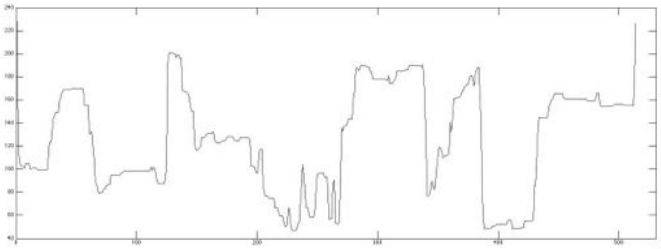
Comparison with Rudin-Osher-Fatemi model $(\mathcal{P}_1)$. We compare the two models on the noisy image with $\sigma = 0.25$. As expected, piecewise constant areas appear with $(\mathcal{P}_1)$, while it is not the case with $(\mathcal{P}_2)$. We still focus on a line that meets contours.

(a) Noisy image



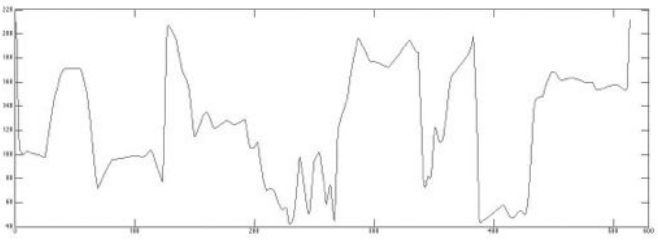
(b) Noisy slice

(c) $(\mathcal{P}_1)$ model solution



(d) $(\mathcal{P}_1)$ model slice

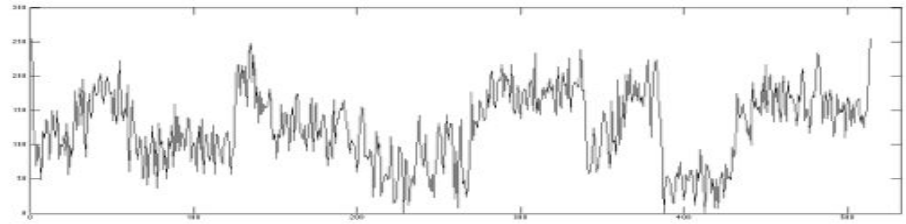
(e) $(\mathcal{P}_2)$ model solution



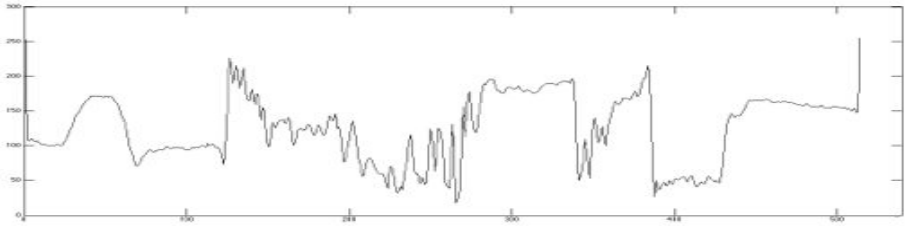
(f) $(\mathcal{P}_2)$ model slice

Figure 3.15: Comparison between $(\mathcal{P}_1)$ and $(\mathcal{P}_2)$ models - $\sigma = 0.25$, $\lambda = 25$

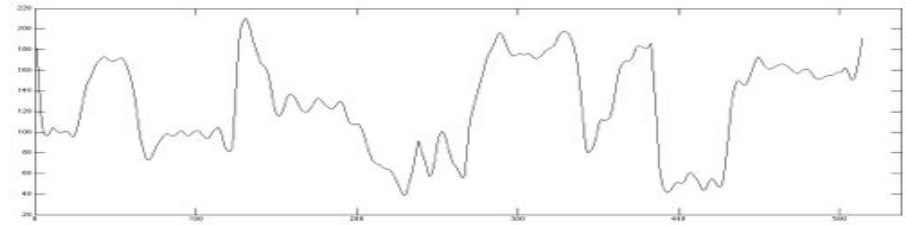
Figures 3.15 and 3.16 are obtained for $\lambda = 25$ and $\lambda = 50$ respectively and 200 iterations.



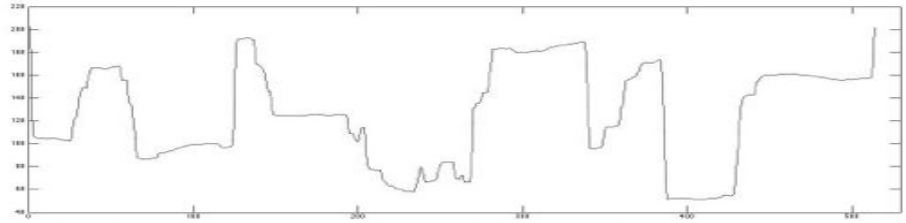
(a) Noisy slice



(b) Original slice



(c) $(\mathcal{P}_2)$ model .



(d) $(\mathcal{P}_1)$ model

Figure 3.16: Zoom on "Lena" slices- $\sigma = 0.25$, $\lambda = 50$, 50 iterations

Figure 3.17 is obtained for a large number of iterations and $\lambda = 50$ to show how we deal with the staircasing effect : the image restored with $(\mathcal{P}_1)$ is clearly piecewise constant while the $(\mathcal{P}_2)$ one seems to be blurred. However, this is

an optical effect: considering a slice shows that the $(\mathcal{P}_2)$ model removes noise significantly and contours are better preserved: the amplitude of high peaks that correspond to contours is not changed, which is not the case in ROF-model (Figure 3.16).

(a) Original image (b) Noisy image (c) $(\mathcal{P}_1)$ model image (d) $(\mathcal{P}_2)$ model image

Figure 3.17: Staircasing effect - $\sigma = 0.25$, $\lambda = 50$.

Though the $(\mathcal{P}_{\lambda,\mu})$ model is rather a texture analysis tool, it can been used for denoising as well: indeed noise (and/or fine textures) is included in the L^2 (w) part and the denoised image is $v + u$. We give an example with Figure 3.18. Additional experiments will be performed in section 5.

(a) Noisy image ($\sigma = 0.25$) (b) $(\mathcal{P}_1)$ - $\lambda = 50$ (c) $(\mathcal{P}_2)$ - $\lambda = 50$ (d) $(\mathcal{P}_{\lambda,\mu})$ – $\lambda = 5$ $\mu = 10$

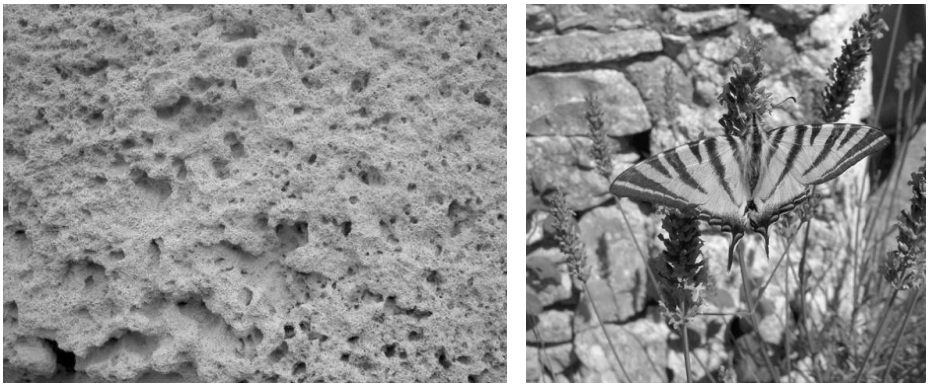
Figure 3.18: Comparison of the different models for denoising purpose.

3.5.2. Texture analysis

In this section, we do not report on texture extraction process for $(\mathcal{P}_2)$. Numerical tests can be found in Bergounioux and Piffet (2010). We focus on $(\mathcal{P}_{\lambda,\mu})$ which can be viewed as a multiscale model for texture extraction. We have performed numerical experimentation on the two (natural) images of Figure 3.19. More results can be found in Bergounioux and Piffet (2013).

- Image (a) is a picture of an old damaged wall which can be considered as pure texture.

- Image (b) involves both sharp contours and small details.



(a) Wall (b) Butterfly

Figure 3.19: Examples

The stopping criterion is based on the difference between two consecutive iterates that should be less than 10^{-3} coupled with a maximal number of iterations (here 175).

Sensitivity with respect to λ . We can see that the ratio $\rho := \frac{\lambda}{\mu}$ is significant : indeed if $\mu >> \lambda$ the second-order term is more weighted than the first order one and the BV^2 component has a small second derivative. This means that there are less and less details as the ratio ρ grows and the resulting image is more and more blurred.

The ratio ρ is less significant for the BV-component u which is sensible to the λ parameter. One sees that the larger λ is, the more u looks piecewise constant. This is consistent with the fact that the optimal value for $\Phi_1(u)$ should be smaller as λ grows.

Moreover, if λ is large enough then $u = 0$ (Figure 3.21 (d)). Indeed we have noticed that the optimal solution (u^*, v^*) satisfies (3.19). This means that u^* is the solution to the classical Rudin-Osher-Fatemi problem

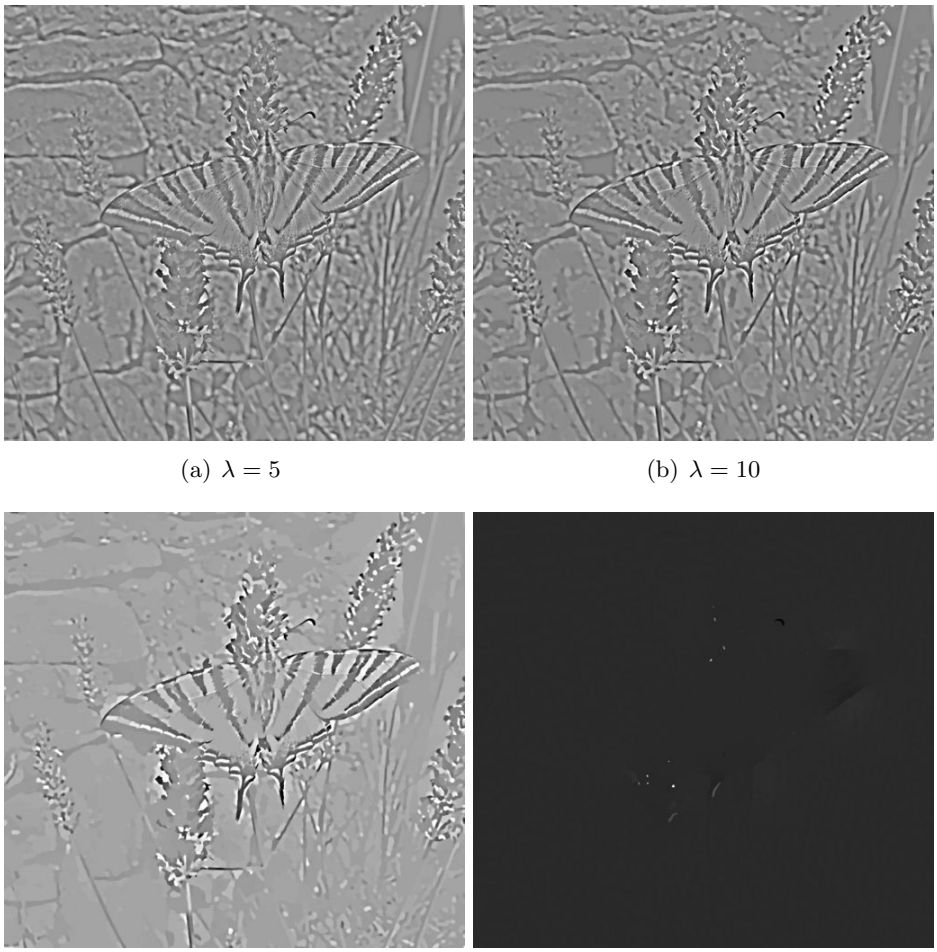
$$u^* = \text{argmin}\{\frac{1}{2}\|\text{f} - \text{u}\|^2 + \lambda\Phi_1(\text{u}) \ , \text{u} \in \text{BV}(\Omega)\}$$

with $f := u_d - v^*$. With a result by Meyer (Meyer (2001), Lemma 3, p.42) we know that $u^* = 0$ if $\lambda > \|u_d - v^*\|_G$, where $\|\cdot\|_G$ denotes the G-norm (see section 2.7).

(a) $\lambda = 5 \quad (\rho = 10)$ (b) $\lambda = 10 \quad (\rho = 5)$

(c) $\lambda = 20 \quad (\rho = 2.5)$ (d) $\lambda = 50 \quad (\rho = 1)$

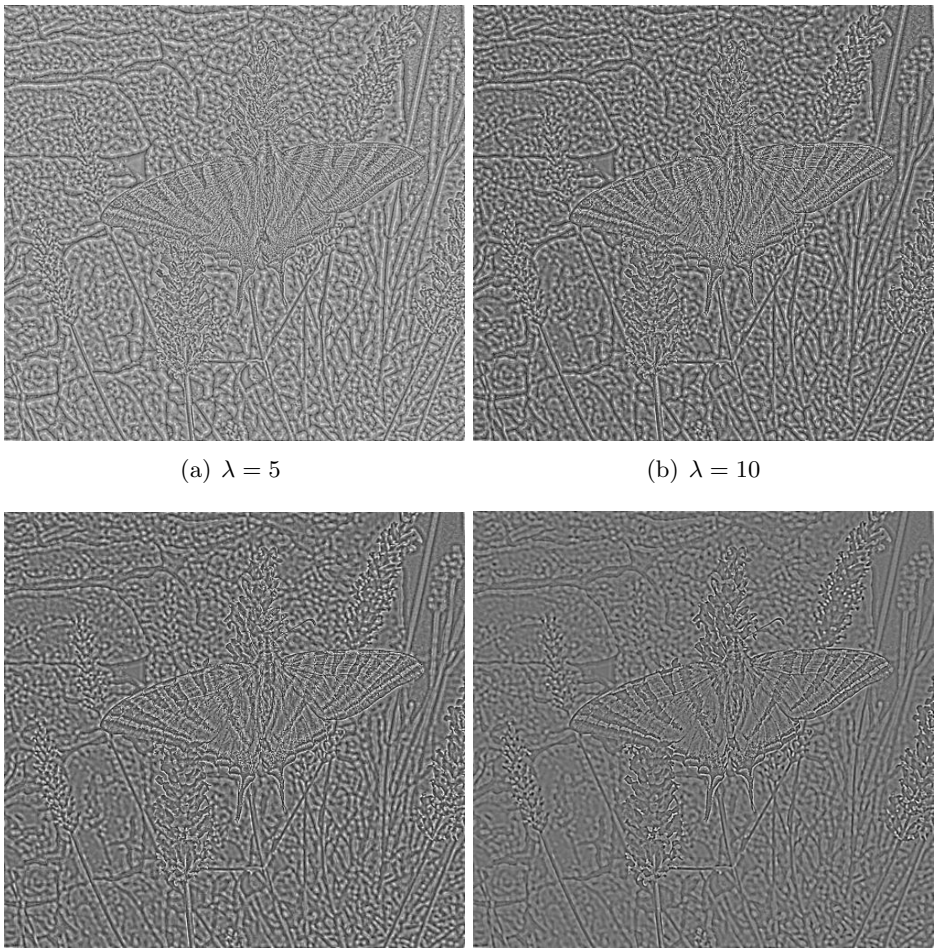
Figure 3.20: BV^2 component - v - $\mu = 50$ - $\rho := \dfrac{\lambda}{\mu}$



(a) $\lambda = 5$ (b) $\lambda = 10$

(c) $\lambda = 20$ (d) $\lambda = 50$

Figure 3.21: BV component - u - $\mu = 50$



(a) $\lambda = 5$ (b) $\lambda = 10$

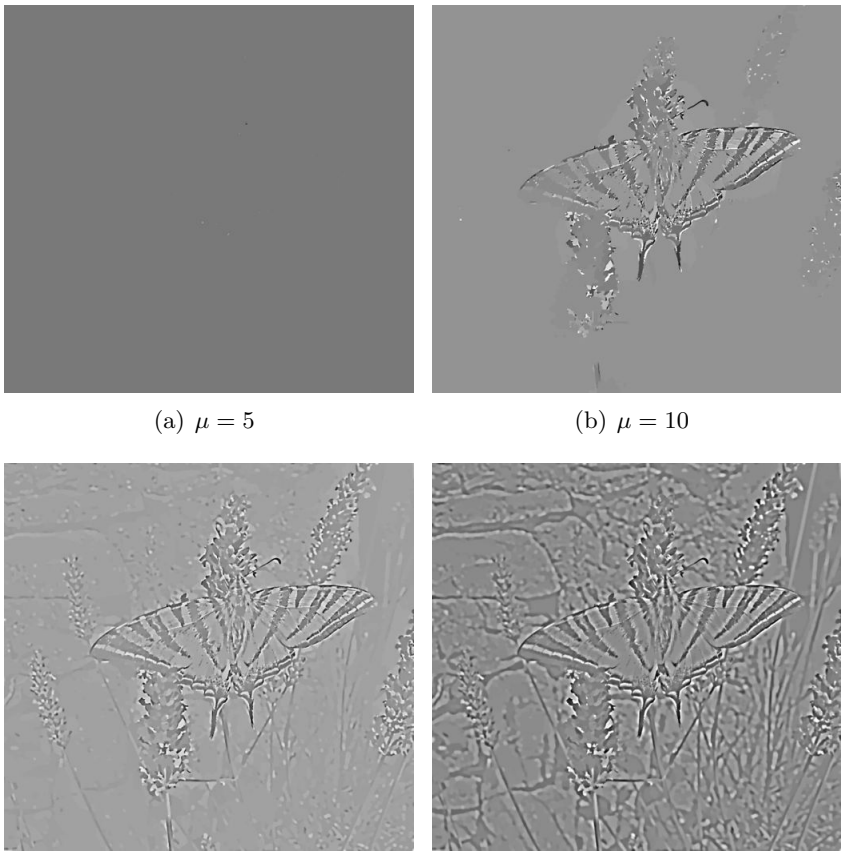
(c) $\lambda = 20$ (d) $\lambda = 50$

Figure 3.22: L^2 component - $w = u_d - u - v$- $\mu = 50$

Sensitivity with respect to μ . The same comments hold : the ratio ρ is the significant quantity with respect to the behaviour of the BV^2 component. The effect of μ on the remainder term w seems more significant than the effect of λ.

(a) $\mu = 5$ (b) $\mu = 10$ (c) $\mu = 20$ (d) $\mu = 50$

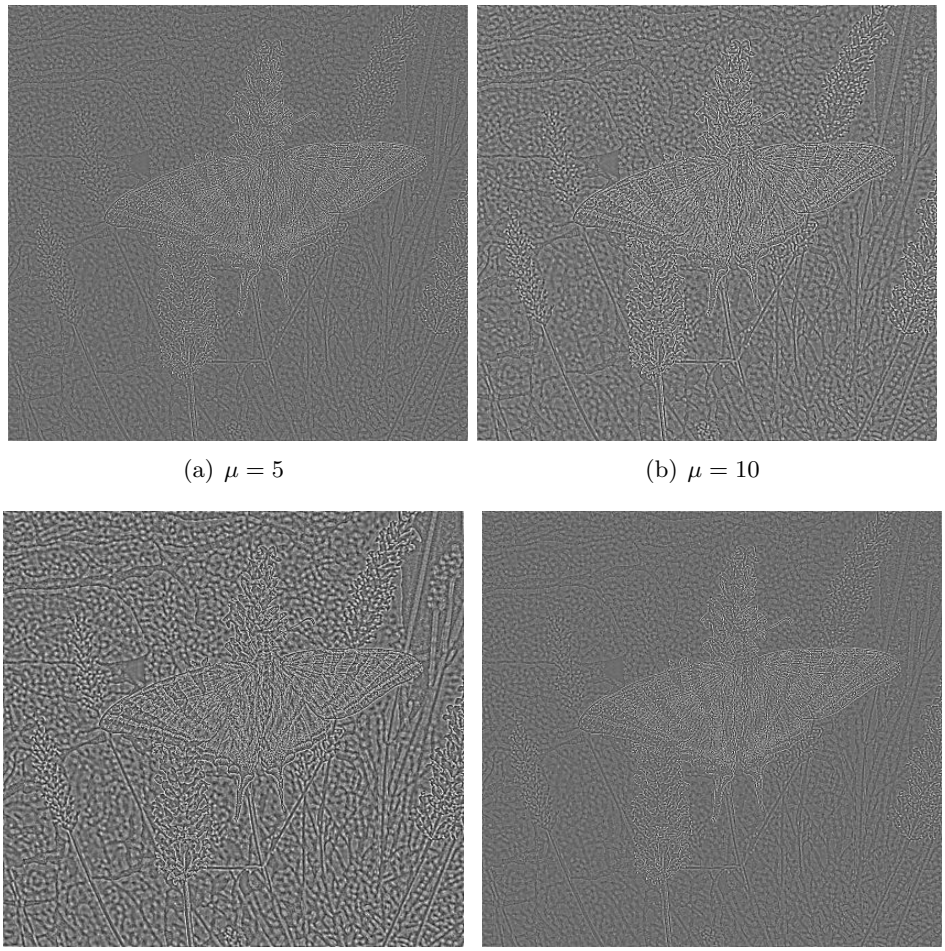
Figure 3.23: BV^2 component - v - $\lambda = 10$



(a) $\mu = 5$ (b) $\mu = 10$

(c) $\mu = 20$ (d) $\mu = 50$

Figure 3.24: BV component - u - $\lambda = 10$ -

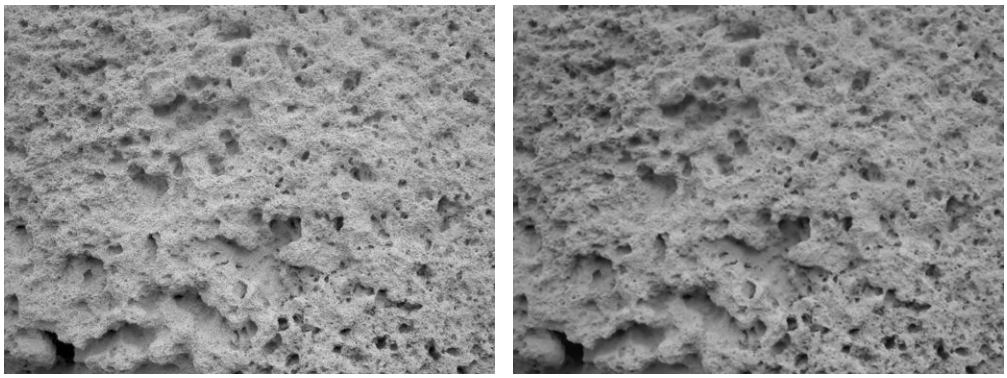


(a) $\mu = 5$ (b) $\mu = 10$

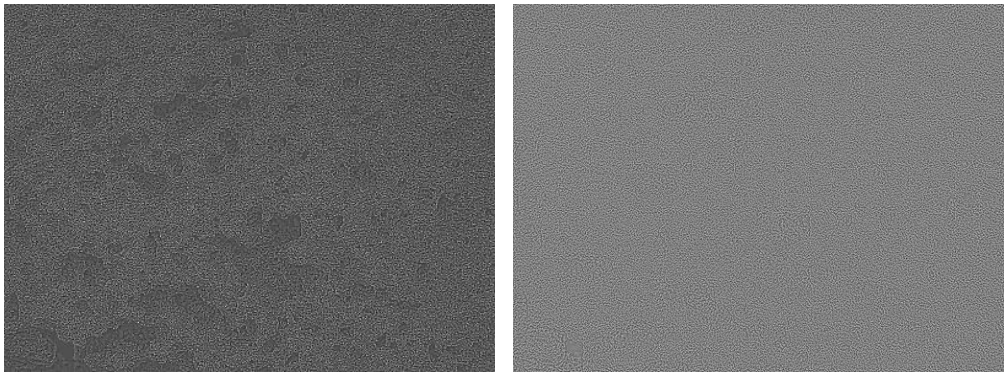
(c) $\mu = 20$ (d) $\mu = 50$

Figure 3.25: L^2 component - $w = u_d - u - v$- $\lambda = 10$

Decomposition as three components. We present the three components together for image (a) and different values of λ and μ.This image may be considered as pure texture. We clearly see that the BV^2-component involves the image dynamic, the BV-component u extracts a macro-texture and the remainder term w a micro-structure. The scaling between u and w is tuned via parameters λ.

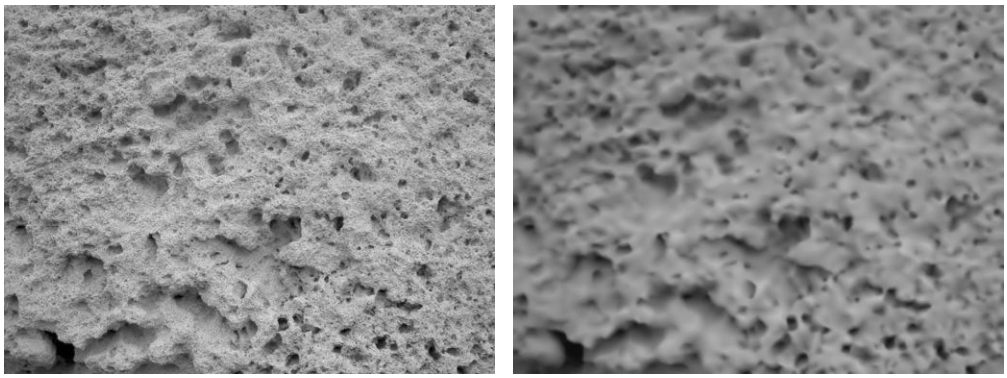


(a) Original u_d (b) BV^2-component v

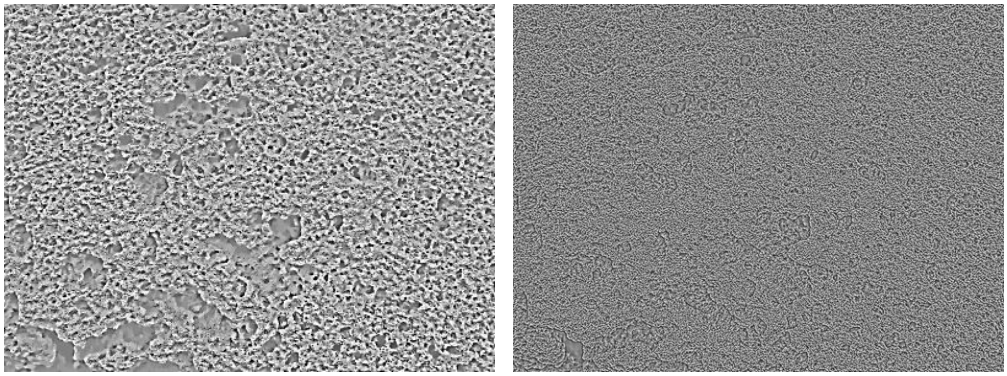


(c) BV-component u (d) L^2 -component : w

Figure 3.26: Wall for $\lambda = 1$ and $\mu = 1$ - $\rho = 1$

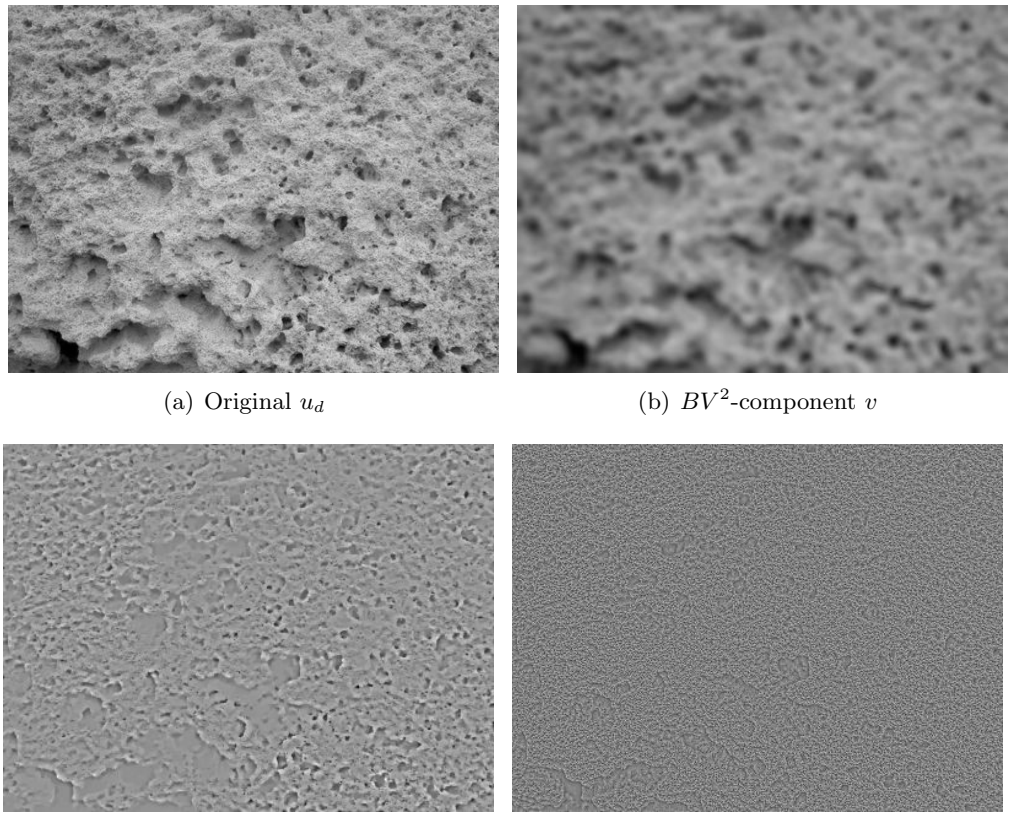


(a) Original u_d (b) BV^2-component v



(c) BV-component u (d) L^2 -component : w

Figure 3.27: Wall for $\lambda = 5$ and $\mu = 10$ - $\rho = 0.5$

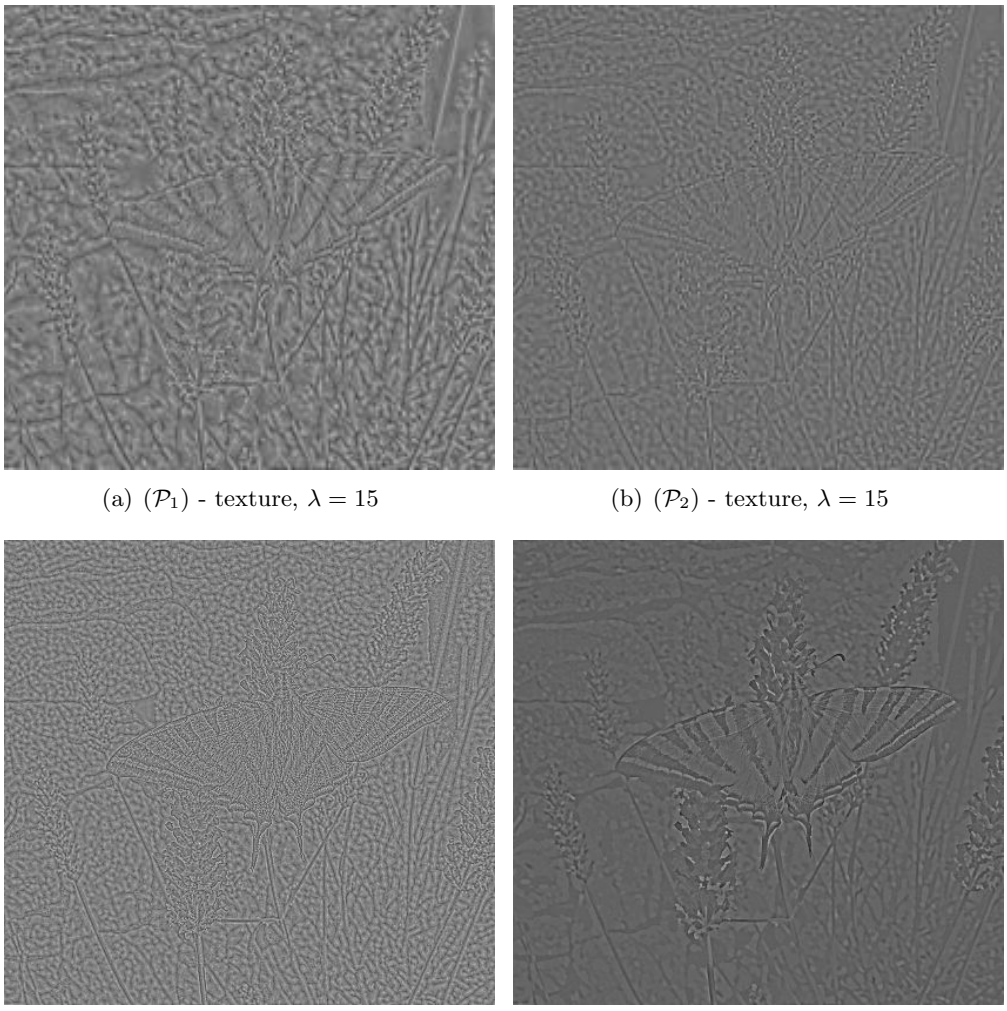


(a) Original u_d (b) BV^2-component v

(c) BV-component u (d) L^2 -component : w

Figure 3.28: Wall for $\lambda = 10$ and $\mu = 50$ - $\rho = 0.5$

We end this section with a comparison between $(\mathcal{P}_1)$, $(\mathcal{P}_2)$ and $(\mathcal{P}_{\lambda,\mu})$ for texture analysis.



(a) $(\mathcal{P}_1)$ - texture, $\lambda = 15$

(b) $(\mathcal{P}_2)$ - texture, $\lambda = 15$

(c) $(\mathcal{P}_{\lambda,\mu})$ - L^2 part w, $\lambda = 5$ and $\mu = 15$

(d) $(\mathcal{P}_{\lambda,\mu})$ - full texture $u + w$, $\lambda = 5$ and $\mu = 15$

Figure 3.29: Comparison for texture extraction. The textures in $(\mathcal{P}_1)$ and $(\mathcal{P}_2)$ are defined as the remainder term "data - solution".The texture in $(\mathcal{P}_{\lambda,\mu})$ is defined as w and we present $u+w$ as well.

4. 3D second order models

We have investigated the different models in a continuous setting and the discretized problems in the 2D-case. This section devoted to the 3D numerical realization. A detailed analysis can be found in Bergounioux and Tran (2011).

4.1. Resolution of problems (P_1) and (P_2) in the 3D-case

Results of section 2.4 (for $(\mathcal{P}_1)$) and section 3 (for $(\mathcal{P}_2)$) can be extended to the 3D-case in a straightforward way. We first begin with the discretization process. In the sequel, the dimension space is $n = 3$ and the image size is $N_1 \times N_2 \times N_3$. The generic component of u is $u_{i,j,k}$ and we denote similarly the continuous function and the corresponding (discretized) tensor.
We set $X = \mathbb{R}^{N_1 \times N_2 \times N_3}$ endowed with inner product and norm

$$\langle u, v \rangle_X = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sum_{k=1}^{N_3} u_{i,j,k} v_{i,j,k} \text{ and } \|u\|_X = \sqrt{\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sum_{k=1}^{N_3} u_{i,j,k}^2}$$

and set $Y = X \times X \times X$.

(a) Computation of the discrete gradient $\nabla u \in Y$ of the image $u \in X$ and discretization of the term $\Phi_1(u)$:

$$(\nabla u_{i,j,k}) = (\nabla u^1_{i,j,k}, \nabla u^2_{i,j,k}, \nabla u^3_{i,j,k})$$

where

$$\nabla u^1_{i,j,k} = \begin{cases} u_{i+1,j,k} - u_{i,j,k} & i < N_1 \\ 0 & i = N_1 \end{cases}$$

$$\nabla u^2_{i,j,k} = \begin{cases} u_{i,j+1,k} - u_{i,j,k} & j < N_2 \\ 0 & j = N_2 \end{cases}$$

$$\nabla u^3_{i,j,k} = \begin{cases} u_{i,j,k+1} - u_{i,j,k} & k < N_3 \\ 0 & k = N_3 \end{cases}$$

Then, using the notations of the previous sections the 3D discrete total variation writes

$$J_1(u) = \sum_{1 \le i \le N_1} \sum_{1 \le j \le N_2} \sum_{1 \le k \le N_3} |(\nabla u)_{i,j,k}|, \tag{4.1}$$

where $|(\nabla u)_{i,j,k}|) := \sqrt{\sum_{p=1}^{3} (\nabla u^p_{i,j,k})^2}$.

(b) Computation of the adjoint operator of the discrete gradient: the discrete

divergence writes

$$
\begin{aligned}
(\text{div } p)_{i,j,k} = &\begin{cases} p^1_{i,j,k} - p^1_{i-1,j,k} & \text{if } 1 < i < N_1 \\ p^1_{i,j,k} & \text{if } i = 1 \\ -p^1_{i-1,j,k} & \text{if } i = N_1 \end{cases} \\
&+ \begin{cases} p^2_{i,j,k} - p^2_{i,j-1,k} & \text{if } 1 < j < N_2 \\ p^2_{i,j,k} & \text{if } j = 1 \\ -p^2_{i,j-1,k} & \text{if } j = N_2 \end{cases} \\
&+ \begin{cases} p^3_{i,j,k} - p^2_{i,j,k-1} & \text{if } 1 < k < N_3 \\ p^3_{i,j,k} & \text{if } k = 1 \\ -p^3_{i,j,k-1} & \text{if } k = N_3 \end{cases}
\end{aligned} \tag{4.2}
$$

(c) Computation of the discrete Hessian and computation of $J_2(v)$.
We have

$$\langle \nabla u, \text{div}\phi \rangle = - \langle \phi, \nabla^2 u \rangle \ .$$

Then,

$$J_2(v) := \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sum_{k=1}^{N_3} \|(Hv)_{i,j,k}\|_{\mathbb{R}^9}$$

where

$$
\begin{aligned}
(Hv)_{i,j,k} =&(Hv^{11}_{i,j,k}, Hv^{12}_{i,j,k}, Hv^{13}_{i,j,k}, Hv^{21,}_{i,j,k} \\
&Hv^{22}_{i,j,k}, Hv^{23}_{i,j,k}, Hv^{31}_{i,j,k}, Hv^{32}_{i,j,k}, Hv^{33}_{i,j,k}).
\end{aligned}
$$

For every $i = 1, ..., N_1$, $j = 1, ..., N_2$ and $k = 1, ..., N_3$, the computation of Hv gives

$$
(Hv)^{11}_{i,j,k} = \begin{cases} v_{i+1,j,k} - 2v_{i,j,k} + v_{i-1,j,k} & 1 < i < N_1 \\ v_{i+1,j,k} - v_{i,j,k} & i = 1 \\ v_{i,j,k} - v_{i-1,j,k} & i = N_1 \end{cases}
$$

$$
(Hv)^{12}_{i,j,k} = \begin{cases} v_{i,j+1,k} - v_{i,j,k} - v_{i-1,j+1,k} + v_{i-1,j,k} & 1 < i \leq N_1 \\ & 1 \leq j < N_2 \\ 0 & j = N_2 \\ 0 & i = 1 \end{cases}
$$

$$
(Hv)^{13}_{i,j,k} = \begin{cases} v_{i,j,k+1} - v_{i,j,k} - v_{i-1,j,k+1} + v_{i-1,j,k} & 1 < i \leq N_1 \\ & 1 \leq k < N_3 \\ 0 & i = 1 \\ 0 & k = N_3 \end{cases}
$$

$$(Hv)^{21}_{i,j,k} = \begin{cases} v_{i+1,j,k} - v_{i,j,k} - v_{i+1,j-1,k} + v_{i,j-1,k} & 1 \le i < N_1 \\ & 1 < k \le N_3 \\ 0 & i = N_1 \\ 0 & k = 1 \end{cases}$$

$$(Hv)^{22}_{i,j,k} = \begin{cases} v_{i,j+1,k} - 2v_{i,j,k} + v_{i,j-1,k} & 1 < j < N_2 \\ v_{i,j+1,k} - v_{i,j,k} & j = 1 \\ v_{i,j,k} - v_{i,j-1,k} & j = N_2 \end{cases}$$

$$(Hv)^{23}_{i,j,k} = \begin{cases} v_{i,j,k+1} - v_{i,j,k} - v_{i,j-1,k+1} + v_{i,j-1,k} & 1 < j \le N \\ & 1 \le k < N_3 \\ 0 & j = 1 \\ 0 & k = N_3 \end{cases}$$

$$(Hv)^{31}_{i,j,k} = \begin{cases} v_{i+1,j,k} - v_{i,j,k} - v_{i+1,j,k-1} + v_{i,j,k-1} & 1 < k \le N_3 \\ & 1 \le i < N_1 \\ 0 & k = 1 \\ 0 & i = N_1 \end{cases}$$

$$(Hv)^{32}_{i,j,k} = \begin{cases} v_{i,j+1,k} - v_{i,j,k} - v_{i+,j+1,k-1} + v_{i,j,k-1} & 1 \le j < N \\ & 1 < k \le N_3 \\ 0 & j = N_2 \\ 0 & k = 1 \end{cases}$$

$$(Hv)^{33}_{i,j,k} = \begin{cases} v_{i,j,k+1} - 2v_{i,j,k} + v_{i,j,k-1} & 1 < k < N_3 \\ v_{i,j,k+1} - v_{i,j,k} & k = 1 \\ v_{i,j,k} - v_{i,j,k-1} & k = N_3 \end{cases}$$

(d) Computation of the adjoint operator of discrete Hessian.
Let us consider $H^* : X^9 \to X$ defined as follows (H^* is the adjoint of operator H): for every $p = (p^{11}, p^{12}, p^{13}, p^{21}, p^{22}, p^{23}, p^{31}, p^{32}, p^{33}) \in X^9$,

$$\begin{aligned}(H^*p)_{i,j,k} = \sigma^{11}_{i,j,k} + \sigma^{12}_{i,j,k} + \sigma^{13}_{i,j,k} + \sigma^{21}_{i,j,k} + \sigma^{22}_{i,j,k} \\ + \sigma^{23}_{i,j,k} + \sigma^{31}_{i,j,k} + \sigma^{32}_{i,j,k} + \sigma^{33}_{i,j,k}\end{aligned}$$

where

$$\sigma^{11}_{i,j,k} = \begin{cases} p^{11}_{i+1,j,k} - 2p^{11}_{i,j,k} + p^{11}_{i-1,j,k} & 1 < i < N_1 \\ p^{11}_{i+1,j,k} - p^{11}_{i,j,k} & i = 1 \\ p^{11}_{i-1,j,k} - p^{11}_{i,j,k} & i = N_1 \end{cases}$$

$$\sigma^{22}_{i,j,k} = \begin{cases} p^{22}_{i,j+1,k} - 2p^{22}_{i,j,k} + p^{22}_{i,j-1,k} & 1 < j < N_2 \\ p^{22}_{i,j+1,k} - p^{22}_{i,j,k} & j = 1 \\ p^{22}_{i,j-1,k} - p^{22}_{i,j,k} & j = N_2 \end{cases}$$

$$\sigma^{33}_{i,j,k} = \begin{cases} p^{33}_{i,j,k+1} - 2p^{33}_{i,j,k} + p^{33}_{i,j,k-1} & 1 < k < N_3 \\ p^{33}_{i,j,k+1} - p^{33}_{i,j,k} & k = 1 \\ p^{33}_{i,j,k-1} - p^{33}_{i,j,k} & k = N_3 \end{cases}$$

$$\sigma^{12}_{i,j,k} = \begin{cases} p^{12}_{i+1,j,k} & i = 1, j = 1 \\ -p^{12}_{i+1,j-1,k} & i = 1, j = N_2 \\ p^{12}_{i+1,j,k} - p^{12}_{i+1,j-1,k} & i = 1, 1 < j < N_2 \\ -p^{12}_{i,j,k} & i = N_1, j = 1 \\ p^{12}_{i,j-1,k} & i = N_1, j = N_2 \\ p^{12}_{i,j-1,k} - p^{12}_{i,j,k} & i = N_1, 1 < j < N_2 \\ p^{12}_{i+1,j,k} - p^{12}_{i,j,k} & 1 < i < N_1, j = 1 \\ p^{12}_{i,j-1,k} - p^{12}_{i+1,j-1,k} & 1 < i < N_1, j = N_2 \\ p^{12}_{i,j-1,k} - p^{12}_{i,j,k} - p^{12}_{i+1,j-1,k} + p^{12}_{i+1,j,k} & 1 < i < N_1, 1 < j < N_2 \end{cases}$$

$$\sigma^{13}_{i,j,k} = \begin{cases} p^{13}_{i+1,j,k} & i = 1, k = 1 \\ -p^{13}_{i+1,j,k-1} & i = 1, k = N_3 \\ p^{13}_{i+1,j,k} - p^{13}_{i+1,j,k-1} & i = 1, 1 < j < N_3 \\ -p^{13}_{i,j,k} & i = N_1, k = 1 \\ p^{13}_{i,j,k-1} & i = N_1, k = N_3 \\ p^{13}_{i,j,k-1} - p^{13}_{i,j,k} & i = N_1, 1 < k < N_3 \\ p^{13}_{i+1,j,k} - p^{13}_{i,j,k} & 1 < i < N_1, k = 1 \\ p^{13}_{i,j,k-1} - p^{13}_{i+1,j,k-1} & 1 < i < N_1, k = N_3 \\ p^{13}_{i,j,k-1} - p^{13}_{i,j,k} - p^{13}_{i+1,j,k-1} + p^{13}_{i+1,j,k} & 1 < i < N_1, 1 < k < N_3 \end{cases}$$

$$\sigma^{21}_{i,j,k} = \begin{cases} p^{21}_{i,j+1,k} & j=1, i=1 \\ -p^{21}_{i-1,j+1,k} & j=1, i=N_1 \\ p^{21}_{i,j+1,k} - p^{21}_{i-1,j+1,k} & j=1, 1<i<N_1 \\ -p^{21}_{i,j,k} & j=N_2, i=1 \\ p^{21}_{i-1,j,k} & j=N_2, i=N_1 \\ p^{21}_{i-1,j,k} - p^{21}_{i,j,k} & j=N_2, 1<i<N_1 \\ p^{21}_{i,j+1,k} - p^{21}_{i,j,k} & 1<j<N_2, i=1 \\ p^{21}_{i-1,j,k} - p^{21}_{i-1,j+1,k} & 1<j<N_2, i=N_1 \\ p^{21}_{i-1,j,k} - p^{21}_{i,j,k} - p^{21}_{i-1,j+1,k} + p^{21}_{i,j+1,k} & 1<j<N_2, 1<i<N_1 \end{cases}$$

$$\sigma^{23}_{i,j,k} = \begin{cases} p^{23}_{i,j+1,k} & j=1, k=1 \\ -p^{23}_{i,j+1,k-1} & j=1, k=N_3 \\ p^{23}_{i,j+1,k} - p^{23}_{i,j+1,k-1} & j=1, 1<k<N_3 \\ -p^{23}_{i,j,k} & j=N_2, k=1 \\ p^{23}_{i,j,k-1} & j=N_2, k=N_3 \\ p^{23}_{i,j,k-1} - p^{23}_{i,j,k} & j=N_2, 1<k<N_3 \\ p^{23}_{i,j+1,k} - p^{23}_{i,j,k} & 1<j<N_2, k=1 \\ p^{23}_{i,j,k-1} - p^{23}_{i,j+1,k-1} & 1<j<N_2, k=N_3 \\ p^{23}_{i,j,k-1} - p^{23}_{i,j,k} - p^{23}_{i,j+1,k-1} + p^{23}_{i,j+1,k} & 1<j<N_2, 1<k<N_3 \end{cases}$$

$$\sigma^{31}_{i,j,k} = \begin{cases} p^{31}_{i,j,k+1} & k=1, i=1 \\ -p^{31}_{i-1,j,k+1} & k=1, i=N_1 \\ p^{31}_{i,j,k+1} - p^{31}_{i-1,j,k+1} & k=1, 1<i<N_1 \\ -p^{31}_{i,j,k} & k=N_3, i=1 \\ p^{31}_{i-1,j,k} & k=N_3, i=N_1 \\ p^{31}_{i-1,j,k} - p^{31}_{i,j,k} & k=N_3, 1<i<N_1 \\ p^{31}_{i,j,k+1} - p^{31}_{i,j,k} & 1<k<N_3, i=1 \\ p^{31}_{i-1,j,k} - p^{31}_{i-1,j,k+1} & 1<k<N_3, i=N_1 \\ p^{31}_{i-1,j,k} - p^{31}_{i,j,k} - p^{31}_{i-1,j,k+1} + p^{31}_{i,j,k+1} & 1<k<N_3, 1<i<N_1 \end{cases}$$

$$\sigma^{32}_{i,j,k} = \begin{cases} p^{32}_{i,j,k+1} & k=1, j=1 \\ -p^{32}_{i,j-1,k+1} & k=1, j=N_2 \\ p^{32}_{i,j,k+1} - p^{32}_{i,j-1,k+1} & k=1, 1<j<N_2 \\ -p^{32}_{i,j,k} & k=N_3, j=1 \\ p^{32}_{i,j-1,k} & k=N_3, j=N_2 \\ p^{32}_{i,j-1,k} - p^{32}_{i,j,k} & k=N_3, 1<j<N_2 \\ p^{32}_{i,j,k+1} - p^{32}_{i,j,k} & 1<k<N_3, j=1 \\ p^{32}_{i,j-1,k} - p^{32}_{i,j-1,k+1} & 1<k<N_3, j=N_2 \\ p^{32}_{i,j-1,k} - p^{32}_{i,j,k} - p^{32}_{i,j-1,k+1} + p^{32}_{i,j,k+1} & 1<k<N_3, 1<j<N_2 \end{cases}$$

The algorithms to compute the projections are the same. Let us detail the second order case for example. The solution to problem (P_2) verifies:

$$v = u_d - P_{\lambda K_2}(u_d)$$

where $P_{\lambda K}$ is the orthogonal projector operator on λK_2 and

$$K_2 := \{H^* p \mid p \in X^9,\ \|p_{i,j,k}\|_{\mathbb{R}^9} \leq 1,\ 1 \leq i \leq N_1, 1 \leq j \leq N, 1 \leq k \leq N_3\}.$$

To compute $P_{\lambda K}(u_d)$ we have to solve the following problem:

$$\begin{cases} \min \|\lambda H^* p - u_d\|^2_X \\ p \in X^9 \\ \|p_{i,j,k}\|^2_{\mathbb{R}^9} \leq 1,\ 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \end{cases}$$

The difference lies in the definition of K_1 and K_2.

4.2. Anisotropic variant for (P_2) in the 3D case

We detail the method that we have presented in section 3.2.2. We perform two rotations r_α and r_β to compute a modified Hessian matrix H' at a voxel (i, j, k). More precisely, we perform a change of variables (with the rotations) to compute the Hessian matrix and the adjoint matrix as in the previous section: the local axis (with the gradient vector as z-axis) are considered instead of the original fixed cartesian axis. Then, we may cancel the Hessian matrix terms corresponding to the gradient direction (for example), to get rid of the corresponding contour (if it is significant) in the extracted texture. Finally we go back to the original axis with the inverse rotations. Let us detail the process :

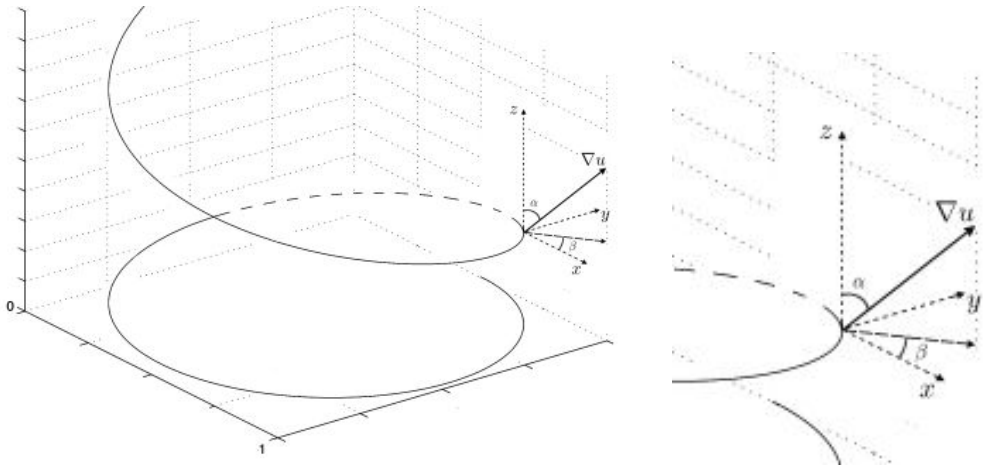


Figure 4.1: Definition of local axis and angles α and β

The angles α and β are defined at point $X_o = (x_o, y_o, z_o)$ as follows : α is the (azimuthal) angle between the gradient $\nabla u(x_o, y_o, z_o)$ and the z-axis . β is the angle between the orthogonal projection of

$$\nabla u(x_o, y_o, z_o) := \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} (x_o, y_o, z_o)$$

(on the xOy plane) and the x -axis. Note that we can perform this transformation with axis Ox or Oy instead of Oz . Let us define the two rotations : r_α and r_β which matrices are :

$$R_\alpha = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \text{ and } R_\beta = \begin{pmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{pmatrix} ,$$

with

$$\alpha = \text{atan} \left(\frac{u_z}{\sqrt{u_x^2 + u_y^2}} \right) (X_o), \quad \beta = \text{atan} \left(\frac{u_y}{u_x} \right) (X_o) .$$

The change of variables from the fixed basis to the local one is given par

$$\tilde{X} = R_\beta R_\alpha X, \text{ with } X = (x, y, z) \in \mathbb{R}^3 .$$

Moreover

$$X = (R_\beta R_\alpha)^{-1} \tilde{X} = R_\alpha^{-1} R_\beta^{-1} \tilde{X} = R_{-\alpha} R_{-\beta} \tilde{X} .$$

In the sequel, we set $\tilde{u}(\tilde{X}) := u(X)$ and $R_{\alpha,\beta} \stackrel{def}{:=} R_{-\alpha}R_{-\beta}$ and we compute the first and second order derivative of $\tilde{u}$:

$$\nabla\tilde{u} = \begin{pmatrix} \dfrac{\partial\tilde{u}}{\partial\tilde{x}} \\ \dfrac{\partial\tilde{u}}{\partial\tilde{y}} \\ \dfrac{\partial\tilde{u}}{\partial\tilde{z}} \end{pmatrix} \text{ and } \tilde{H} := \begin{pmatrix} \dfrac{\partial^2\tilde{u}}{\partial\tilde{x}^2} & \dfrac{\partial^2\tilde{u}}{\partial\tilde{x}\partial\tilde{y}} & \dfrac{\partial^2\tilde{u}}{\partial\tilde{x}\partial\tilde{z}} \\ \dfrac{\partial^2\tilde{u}}{\partial\tilde{x}\partial\tilde{y}} & \dfrac{\partial^2\tilde{u}}{\partial\tilde{y}^2} & \dfrac{\partial^2\tilde{u}}{\partial\tilde{y}\partial\tilde{z}} \\ \dfrac{\partial^2\tilde{u}}{\partial\tilde{x}\partial\tilde{z}} & \dfrac{\partial^2\tilde{u}}{\partial\tilde{y}\partial\tilde{z}} & \dfrac{\partial^2\tilde{u}}{\partial\tilde{z}^2} \end{pmatrix} .$$

A short computation gives

$$\frac{\partial\tilde{u}}{\partial\tilde{x}} = \frac{\partial u}{\partial x}\frac{\partial\tilde{x}}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial\tilde{y}}{\partial x} + \frac{\partial u}{\partial z}\frac{\partial\tilde{z}}{\partial x} = \nabla u \cdot \frac{\partial\tilde{X}}{\partial x} = \nabla u \cdot R(:,1) \ ,$$

where $\cdot$ denotes the $\mathbb{R}^3$ scalar product and $R(:,1)$ is the first column of R. Finally, we get

$$\nabla\tilde{u} = R_{\alpha,\beta}\nabla u \ . \tag{4.3}$$

Now we compute $\tilde{H}$; we set $\tilde{v} = \dfrac{\partial\tilde{u}}{\partial\tilde{x}}$ and estimate $\nabla\tilde{v}$ as above : this will be the first column of $\tilde{H}$.

$$\nabla\tilde{v} = R_{\alpha,\beta}\nabla v = R_{\alpha,\beta}\begin{pmatrix} \dfrac{\partial^2 u}{\partial x^2} \\ \dfrac{\partial^2 u}{\partial y\partial x} \\ \dfrac{\partial^2 u}{\partial z\partial x} \end{pmatrix} .$$

Finally

$$\tilde{H} = R_{\alpha,\beta}H \ . \tag{4.4}$$

As already mentioned, the idea is to cancel some terms of the Hessian matrix to get rid of (or to keep) the contours. However, without performing the rotations, there would be only few possible directions, for example vertical, horizontal and diagonal in the 2D-case so that many contours are not considered. Performing the change of variables allows to identify the gradient direction (that is the contour direction if the gradient is large enough) with the z-axis and then cancel corresponding terms of the matrix $\tilde{H}$. Of course, we have to get back to the original situation. Let us denote by $\mathcal{L}$ the (linear) transformation that assigns 0 to some coefficients of $\tilde{H}$ (this is a projection). The whole process is described by

$$H \to \tilde{H} = R_{-\alpha}R_{-\beta}H \to \mathcal{L}(\tilde{H}) := \tilde{H}' \to [R_{\alpha,\beta}]^{-1}\mathcal{L}(\tilde{H}) = R_\beta R_\alpha \mathcal{L}(\tilde{H}) \ ,$$

that is

$$H \to [R_\beta R_\alpha \mathcal{L} R_{-\alpha} R_{-\beta}]\, H \ . \tag{4.5}$$

So, algorithm is modified as follows

Algorithm 8 Anisotropic strategy for (P_2)

Choose $\tau > 0, \mu > 0$ and compute ∇u_d.
Use a thresholding process to identify the contours $(\|\nabla u_d\| \geq \rho)$.
Set I_ρ the set of voxels corresponding to these significant contours.
For voxels in I_μ, modify H with the following rule

$$H \to \tilde{H} = R_{-\alpha} R_{-\beta} H \to \mathcal{L}(\tilde{H}) = [\mathcal{L} R_{-\alpha} R_{-\beta}] H := H'$$

and compute $(H')^*$
Perform algorithms (4) or (5) with H' instead of H.

5. Examples and applications

We end this paper with two examples in biology and material science. The full second order model behaves well as soon as we have tuned the parameters. This is the most challenging issue of this model. We are not able to provide any automatic tuning of parameters λ and μ by now. Nevertheless, the 3D-images we study have been obtained in the same experimental conditions. Therefore it is possible to tune the parameters with one 2D slice using PSNR or user expertise. Then these parameters can be used for the whole image stack.

Though we have not yet performed a quantitative, theoretical sharp analysis of these second-order models, we get some hints however. Parameter λ should be less that μ and the ratio $\dfrac{\lambda}{\mu}$ gives the scale between the cartoon and the noise parts if the images are pure textures (this is the case for some bone micro-radiographs see Jennane et al. (2013)). In addition, we know that the larger λ is, the better is the cartoon part. We give examples thereafter.

5.1. X-ray imaging - Material science

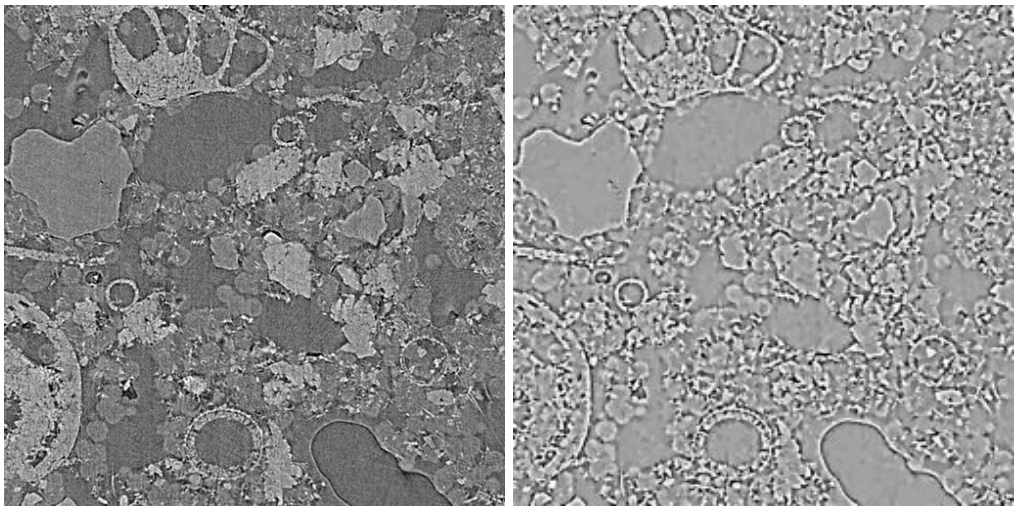
The first application concerns X-ray microtomography images on healthy and deteriorated building stones (Guillot et al. (2009)). Geomaterials (Tuffeau stone) studied here are sedimentary limestones widely utilized during the last centuries for historical monuments construction (chateaux, churches, cathedrals, and houses) along the Loire valley between Orlans and Nantes. Today, Tuffeau stone is mainly used to restore these monuments. This stone is a yellowish-white porous sedimentary limestone, mainly composed of calcite (40 to 70 %), silica (20 to 60 %) in the form of opal cristobalite-tridymite and quartz and some secondary

minerals such as clays and micas. Tuffeau stones are extremely porous (40 to 50 %) with equivalent pore size distribution ranged from 103 to 10-2 μm in size.

The analysis will make it possible to identify the mineralogical phases and the three-dimensional morphology and structure of the porous and solid phases. The images are 8 bits grey level images: they involve several areas corresponding to various materials composing the stone. In addition the porous media under study structure results in elements of texture we have to analyze and restore. The texture corresponds to pores at a micrometric scale, each pore being represented by few pixels. The segmentation and the restoration thus carried out, it will then be necessary to develop areas segmentation tools. We have here three areas to determine, each one corresponding to the various phases of the material (silica, quartz and pore). Once these three areas are identified, one can get the 3D representation of each one. The study has been done with laboratory ISTO [1].

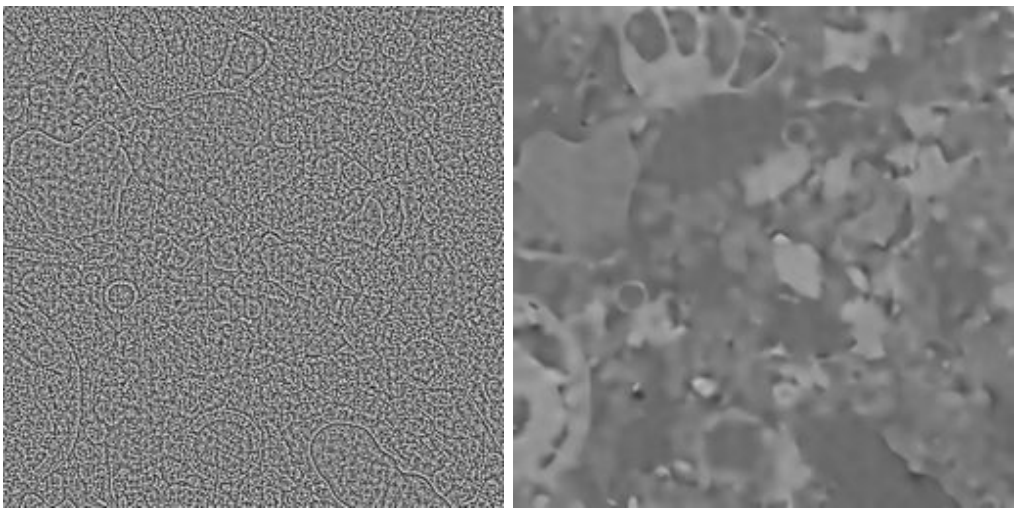
It is impossible to perform segmentation of such images without any preprocessing. Indeed, images are noisy and involve fine texture areas (due to the micritic calcite part) as well. The denoising process should preserve the texture which involves relevant physical information. As we want to recover the vacuum area we have to perform a contour segmentation and if possible regions classification to recover the different physical components of the stone. The decomposition model we propose, can be used as a preprocessing to separate the noise and fine texture component w from the macro-texture component u and perform a classical segmentation method on u.

[1] We thank Olivier Rozenbaum at ISTO(http://www.isto.cnrs-orleans.fr/) who performed acquisition.



(a) Original image

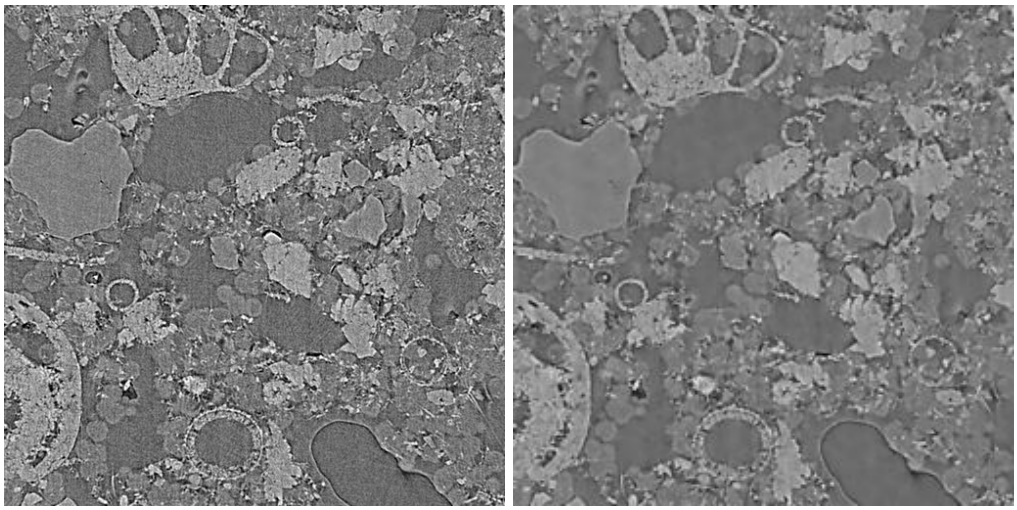
(b) BV-component : u



(c) L^2 -component : $w = u_d - u - v$

(d) BV^2-component : v

Figure 5.1: Decomposition - $\lambda = 10,\ \mu = 20$ (in Bergounioux and Piffet (2013))



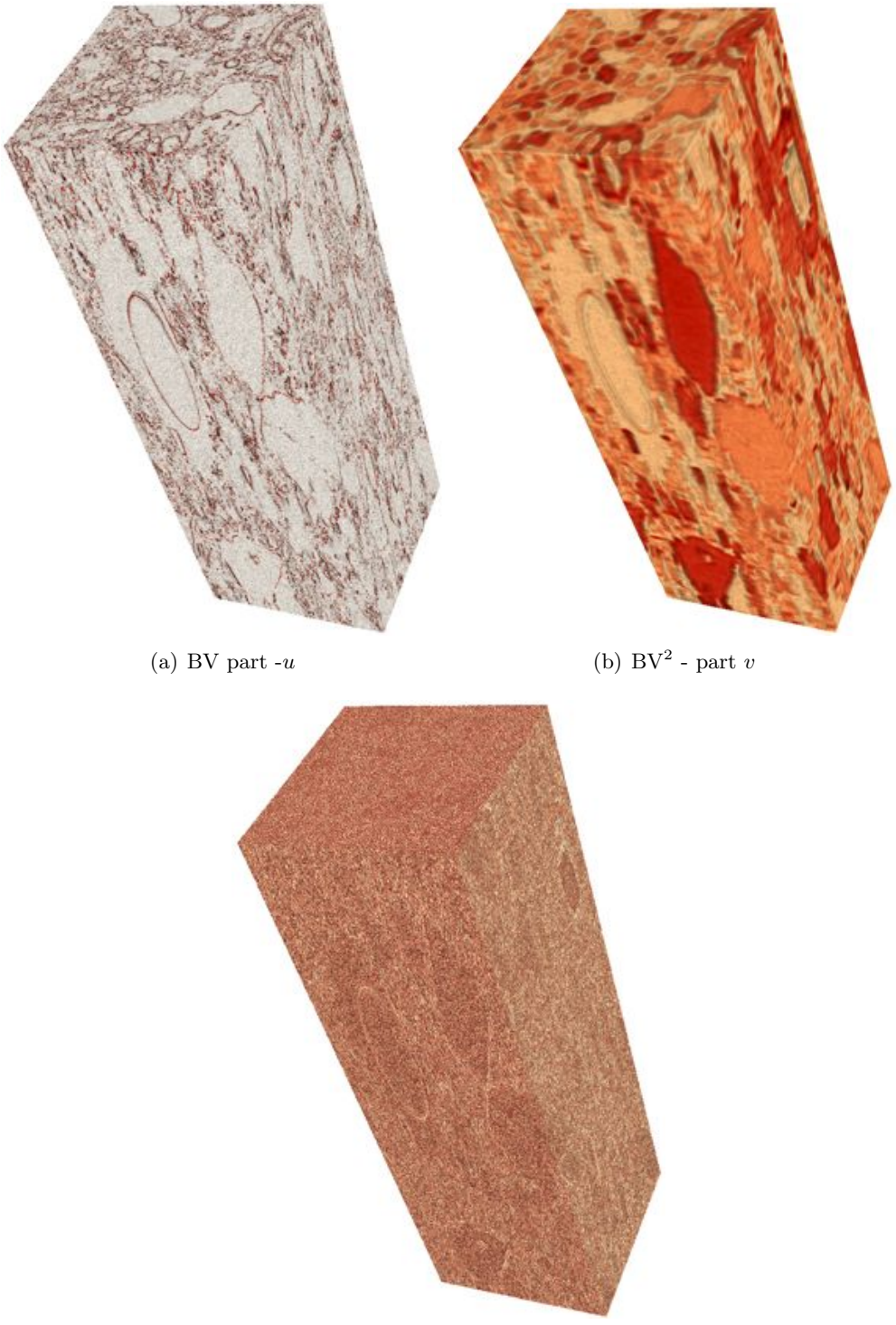
(a) Original (noisy) image u_d (b) Component $u + v$ (denoised image)

Figure 5.2: Original and denoised image - $\lambda = 10,\ \mu = 20$ (in Bergounioux and Piffet (2013))

The original data is made of 800 2D-images

(a) Original Image

Figure 5.3: 3D X-ray stack of Tuffeau



(a) BV part -u

(b) BV^2 - part v

(c) L^2 - part w

Figure 5.4: Decomposition of a 3D X-ray stack - Tuffeau - $\lambda = 5,\ \mu = 10$

We show on the same 3D picture the effect of parameters λ and μ

(a) BV part -u

(b) BV^2 - part v

Figure 5.5: The original data is made of 800 2D-images . The volume corresponding to the first 200 slices has been decomposed with $\lambda = 5$ $\mu = 10$, the volume 201 to 400 with $\lambda = 1$ $\mu = 10$, the volume 401 to 600 with $\lambda = 10$ $\mu = 10$ and the volume 601 to 800 , with $\lambda = 20$ $\mu = 50$.

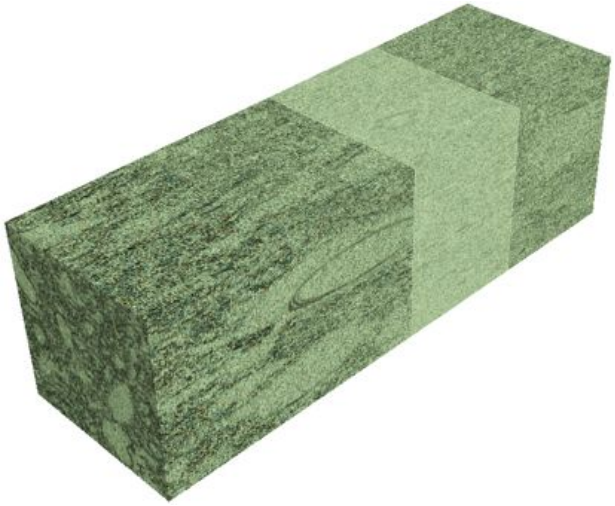
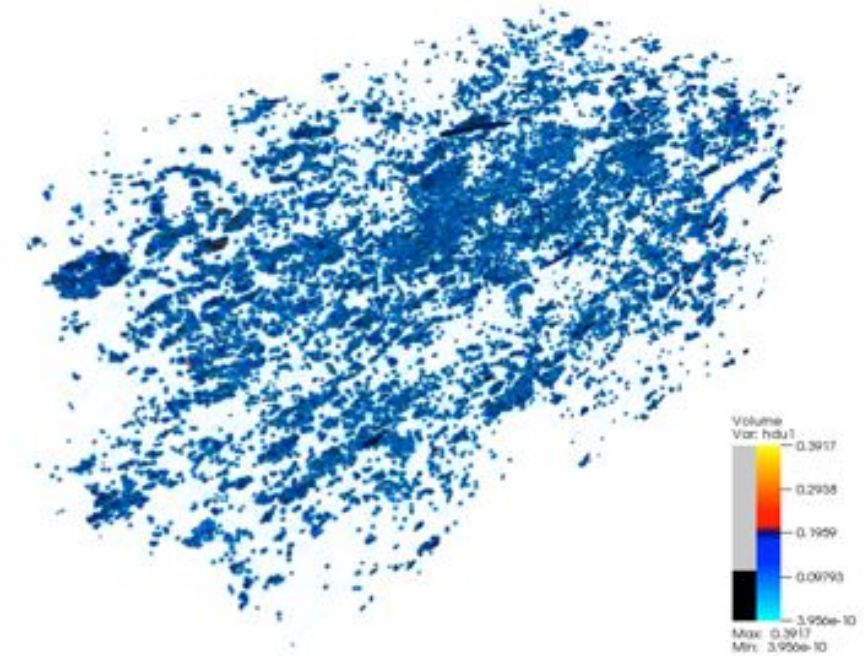
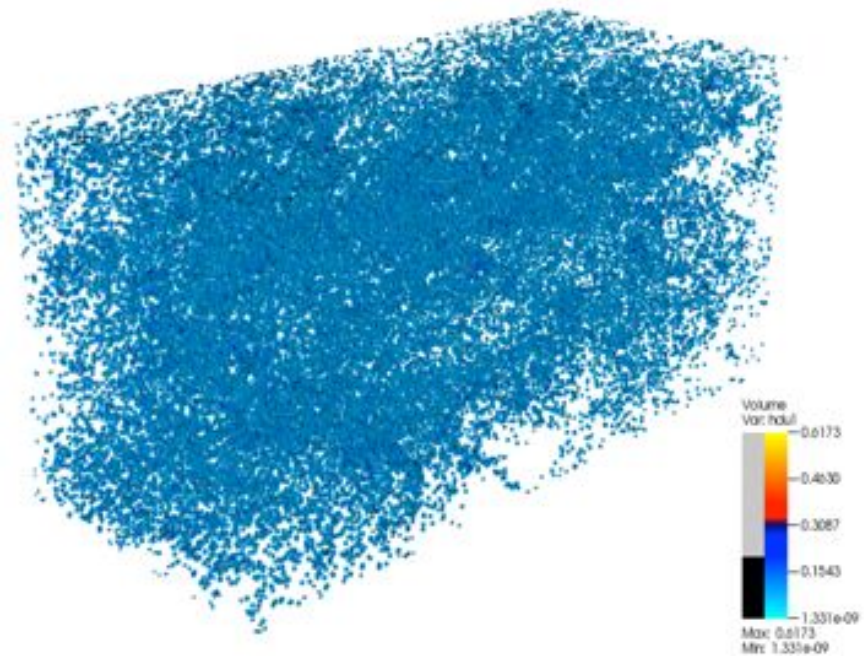


Figure 5.6: (c) L^2 - part w - The volume corresponding to the first 200 slices has been decomposed with $\lambda = 5$ $\mu = 10$, the volume 201 to 400 with $\lambda = 1$ $\mu = 10$, the volume 401 to 600 with $\lambda = 10$ $\mu = 10$ and the volume 601 to 800 , with $\lambda = 20$ $\mu = 50$.

We have computed the decomposition using two methods: first, we used a "false" 3D method by performing a 2D model on every slice. Second, we used the direct 3D method (which is more memory consuming of course). As expected, the solutions are different. The use of a direct 3D method is much better since we use informations in the three directions to compute the gradients and the hessians. We present below the difference between the solutions.



(a) BV part -u

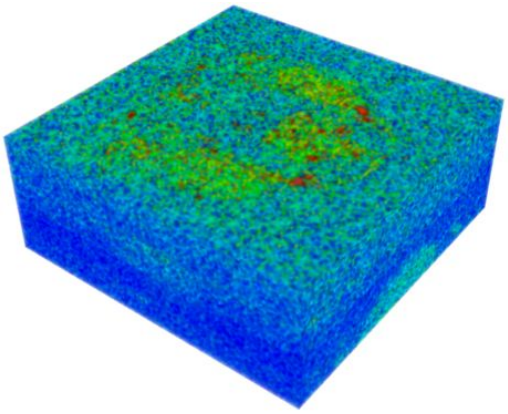


(b) L^2 - part w

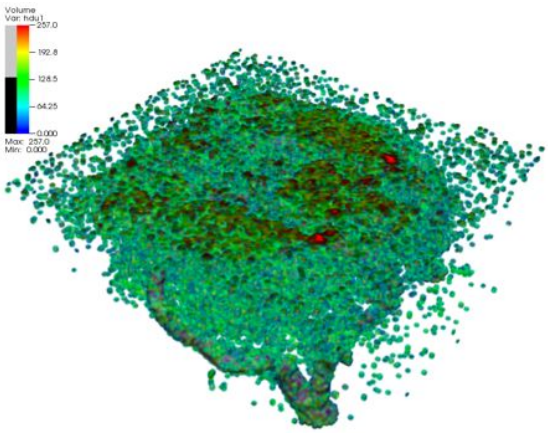
Figure 5.7: Difference between the solutions given by a 2D- slice by slice strategy and a full 3D-strategy - $\lambda = 5$ $\mu = 10$

5.2. MRI imaging -Biology

This example deals with MRI 3D-images of mice brain vessel network.



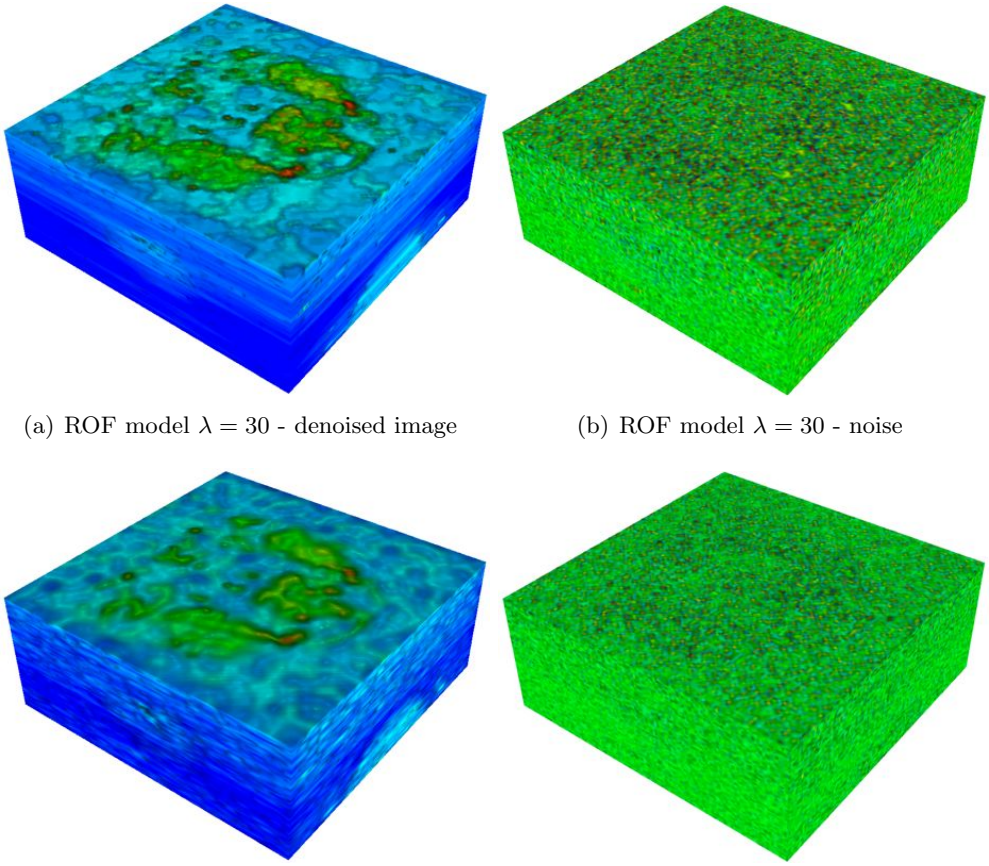
(a) Original Image



(b) Original Image with thresholding

Figure 5.8: Original 3D- MRI stack with and without thresholding

Mice have been genetically modified : some are sick (malaria, cancer) and some are healthy. The goal of the segmentation process is to recover the complete network to get useful indicators as the nodes number, the total volume or the mean size of vessels. Segmentation is quite challenging since images are undersampled and very noisy. Indeed animals are quite small and magnetic fields have to be quite high. Moreover, very small vessels are embedded in noise so that it is quite difficult to recover them. However, these thin structures are of high interest since they are the first to be destroyed during the disease process.



(a) ROF model $\lambda = 30$ - denoised image

(b) ROF model $\lambda = 30$ - noise

(c) ROF2 model $\mu = 10$ - denoised image

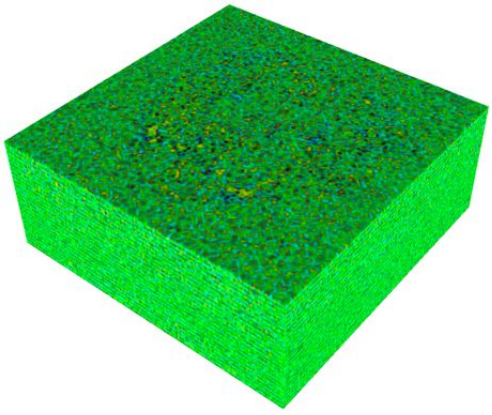
(d) ROF2 model $\mu = 10$ - noise

Figure 5.9: Denoising with ROF ($\lambda = 30$) and ROF2 ($\mu = 10$) models.

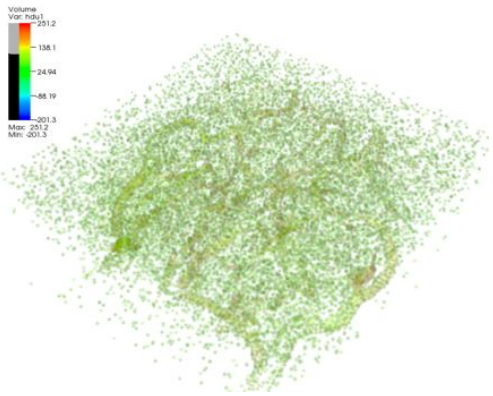
The MRI experiments were performed using Manganese Mn2+ by the researchers of the CBM[2] in Orléans. The full decomposition model acts as a preprocessing tool to isolate the big vessels in the smooth BV^2 part and the small ones in the BV part. Noise is stored in the L^2 part. Figure (5.9) illustrates the $(\mathcal{P}_1)$ and $(\mathcal{P}_2)$ denoising processes respectively.

Figure (5.10) gives the decomposition that is obtained using the full second order model. Parameters have not been optimized : these are clearly not the best ones. However, one can see that usual (histogram based) thresholding process gives interesting results on the BV^2 and BV parts while it is useless on the L^2 part.

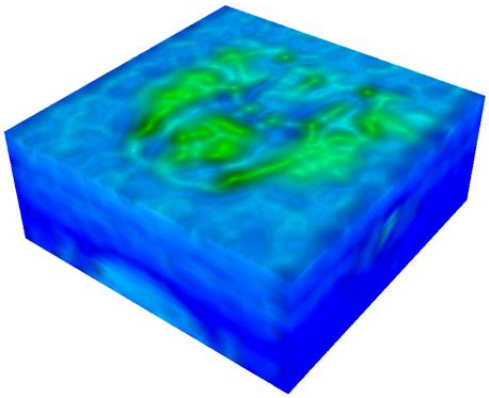
[2]We thank Jean-Claude Belœil, Sandra Même and Frédéric Szeremeta at CBM http://cbm.cnrs-orleans.fr/



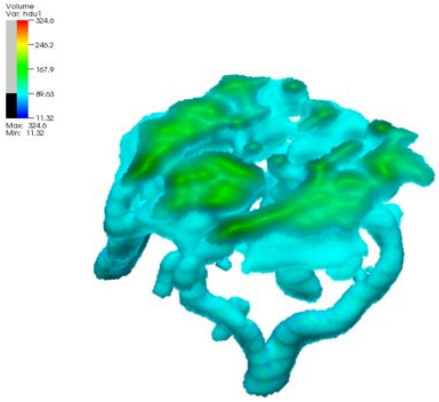
(a) BV part -u



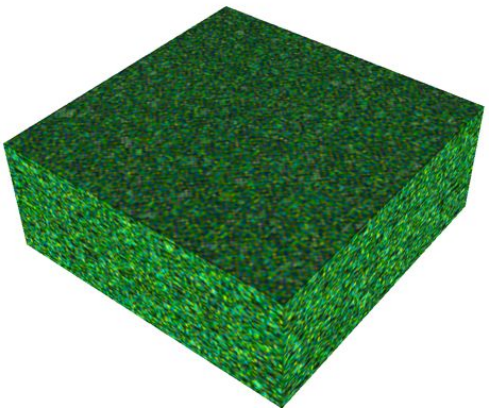
(b) BV part -u with thresholding



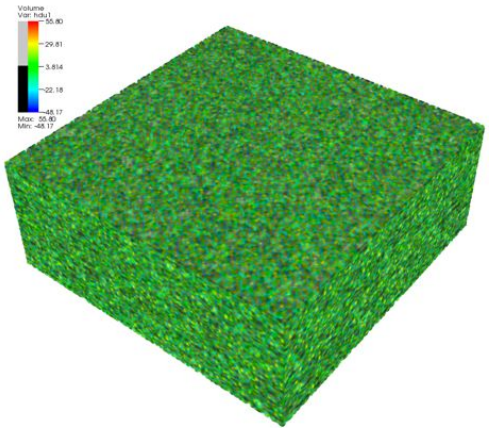
(c) BV^2 - part v



(d) BV^2 - part v with thresholding



(e) L^2 - part w



(f) L^2 - part w with thresholding

Figure 5.10: Decomposition of a 3D MRI stack with the full second order model - $\lambda = 10$, $\mu = 30$ with and without thresholding

A. Mathematical tools

In this section, we recall the main mathematical results that are used in this paper. In the sequel V is a Banach space (that is a normed linear space such that every Cauchy sequence is convergent) assumed to be reflexive. We call V' the V-topological dual space. We denote $\| \ \|_V$ the V-norm and $\langle \cdot, \cdot \rangle$ the duality bracket between V and V':

$$\forall v \in V, \forall \varphi \in V' \qquad \langle \varphi, v \rangle := \varphi(v) \ .$$

A.1. Optimization in Banach spaces

For details on the results presented in this subsection one can refer to Attouch et al. (2006), Brezis (1987) (weak topology, basic functional analysis), Azé (1997), Ekeland and Temam (1999), Barbu and Precupanu (1978), Hiriart-Urruty (1998) (convex analysis, optimization problems).

A.1.1. Semi-continuity and convexity

Definition A.1 (Semi-continuity). *A functional $J : V \to \mathbb{R} \cup \{+\infty\}$ is lower semi-continuous (lsc) on V if one of the following equivalent conditions is satisfied:*

- $\forall a \in \mathbb{R}, \qquad \{ \ u \in V \ \mid \ J(u) \leq a \ \}$ *is a closed subset of* V
- $\forall \bar{u} \in V, \qquad \liminf_{u \to \bar{u}} J(u) \geq J(\bar{u})$.

Theorem A.1. *Every convex function lower semi-continuous for the norm topology (strong) is lower semi-continuous for the weak topology of V.*

From a practical point of view, we use the above result to infer

Corollary A.1. *Let be J a convex, strongly lower semi-continuous from V to $\mathbb{R} \cup \{+\infty\}$. Let be v_n a sequence weakly convergent to some v in V. Then*

$$J(v) \leq \liminf_{n \to +\infty} J(v_n).$$

A.1.2. Gâteaux-differentiability

Definition A.2 (Gâteaux-differentiability). *The function $J : V \to \mathbb{R} \cup \{+\infty\}$ is Gâteaux-differentiable at $u \in$ dom (J) if*

$$J'(u; v) = \lim_{t \to 0^+} \frac{J(u + tv) - J(u)}{t},$$

exists for every $v \in V$ and the mapping

$$v \mapsto J'(u; v)$$

is linear and continuous.

We denote $\nabla J(u)$ the Gâteaux derivative of J at u. It belongs to V'.
When V is a Hilbert space we use the Riesz Theorem (see Brezis (1987) for example) to identify V and V'. Then

$$\forall (u,v) \in V \times V \qquad J'(u;v) = (\nabla J(u), v)\,,$$

where $(\cdot,\cdot)$ stands for the inner V-product. The element $\nabla J(u) \in V$ is the *gradient* of J at u.

Theorem A.2. *Assume that $J : \mathcal{C} \subset V \to \mathbb{R}$ is Gâteaux -differentiable on $\mathcal{C}$ where $\mathcal{C}$ is convex. Then J is convex if and only if*

$$\forall (u,v) \in \mathcal{C} \times \mathcal{C} \qquad J(v) \geq J(u) + \langle \nabla J(u), v-u \rangle \tag{A.1}$$

or equivalently

$$\forall (u,v) \in \mathcal{C} \times \mathcal{C} \qquad \langle \nabla J(u) - \nabla J(v), u-v \rangle \geq 0. \tag{A.2}$$

We may define similarly the second (Gâteaux) derivative of J at u as the (Gâteaux) derivative of the (vector) function $u \to \nabla J(u)$. We note $D^2 J(u)$ and call it *Hessian function*. When $V = \mathbb{R}^n$ the Hessian function can be identified to a square matrix $n \times n$.

A.1.3. Minimization in a reflexive Banach space
We begin with the most useful result for minimization in Banach spaces.

Definition A.3 (Coercivity). *The function $J : V \to \mathbb{R}$ is coercive if*

$$\lim_{\|x\|_V \to +\infty} J(x) = +\infty.$$

Theorem A.3. *Assume V is a reflexive Banach space. Let be $J : V \to \mathbb{R} \cup \{+\infty\}$, lower semi-continuous for the weak topology of V. Let K be a nonempty, weakly closed subset of V. We assume there exists $v_o \in K$ such that $J(v_o) < +\infty$. Then, the minimization problem*

$$(\mathcal{P}) \quad \left\{ \begin{array}{c} \textit{Find } u \textit{ such that} \\ J(u) = \inf \{\ J(v) \mid v \in K\ \}, \end{array} \right. \tag{A.3}$$

has at least a solution if either J is coercive or K is bounded.

An important corollary holds in the convex case :

Corollary A.2. *Assume V is a reflexive Banach space. Let be $J : V \to \mathbb{R} \cup \{+\infty\}$, convex, lower semi-continuous for the strong topology of V. Let K be a nonempty, closed convex subset of V. Assume again there exists $v_o \in K$ such that $J(v_o) < +\infty$. Then, if J is coercive or if K is bounded, $(\mathcal{P})$ has at least a solution. Moreover, if J est strictly convex the solution is unique.*

Let us end with necessary first order optimality condition:

Theorem A.4. *Let K be a non empty, convex subset of V and $J : K \to \mathbb{R}$ be Gâteaux-differentiable on K. If $u \in V$ is a solution to problem $(\mathcal{P})$ then*

$$\forall v \in K, \qquad < \nabla J(u), v - u > \geq 0. \tag{A.4}$$

A.1.4. Example: projection on a closed convex set
In what follows V is an Hilbert space endowed with the inner product $(\cdot, \cdot)$ and the associated norm $\| \cdot \|$, and C is a closed, convex (non empty) subset of V.

Theorem A.5. *Let C be a closed, convex (non empty) subset of V and $x \in V$. Then the problem*

$$\min \ \{ \ \|x - y\|^2, \ y \in C \ \}$$

has a unique solution $x^* \in C$ which is characterized as

$$\forall y \in C \qquad (x - x^*, y - x^*) \leq 0. \tag{A.5}$$

or equivalently

$$\forall y \in C \qquad (x^* - y, y - x) \leq 0. \tag{A.6}$$

The mapping $P_C : V \to C$ that associates x^* to x is the (orthogonal) projection on C. Therefore $P_C(x)$ is the element of C the nearest of x. If we define the distance function as

$$d(x, C) = \inf_{y \in C} \|x - y\|. \tag{A.7}$$

then $d(x, C) = \|x - P_C(x)\|$ when C is a non empty, closed, convex subset of V. We shall use the following corollary

Corollary A.3. *Let C be a a closed, convex set and $\alpha > 0$. Then, for every $x \in V$*

$$P_{\alpha C}(x) = \alpha P_C(\frac{x}{\alpha}) \ .$$

Proof - Let be $x \in V$.

$$\begin{aligned}
x^* = P_{\alpha C}(x) &\iff \forall y \in \alpha C \qquad (x - x^*, y - x^*) \leq 0 \\
&\iff \forall y \in C \qquad (x - x^*, \alpha y - x^*) \leq 0 \\
&\iff \forall y \in C \qquad \left(\frac{x}{\alpha} - \frac{x^*}{\alpha}, y - \frac{x^*}{\alpha} \right) \leq 0 \\
&\iff \frac{x^*}{\alpha} = P_C(\frac{x}{\alpha}) \\
&\iff x^* = \alpha P_C(\frac{x}{\alpha})
\end{aligned}$$

Proposition A.1. *The projection P_C is Lipschitz-continuous from V to C. More precisely:*

$$\forall (x,y) \in V \times V \qquad \|P_C(x) - P_C(y)\| \leq \|x - y\|.$$

In addition

$$\forall (x_1,x_2) \in V \times V \qquad (x_1 - x_2, P_C(x_2) - P_C(x_1)) \leq -\|P_C(x_2) - P_C(x_1)\|^2 \ .$$

A.2. Non smooth Analysis

The results we present here are detailed in Ekeland and Temam (1999), Barbu and Precupanu (1978), Hiriart-Urruty (1998). We first recall one of the main tools of convex analysis.

A.2.1. The Hahn -Banach separation Theorem

In what follows, $\mathcal{X}$ is a (not necessarily reflexive) real Banach space. The geometrical form of Hahn-Banach theorem allows to separate convex sets. For more details we refer to Brezis (1987) .

Definition A.4 (Affine hyperplan). *An closed affine hyperplan is defined as*

$$H = \{\ x \in \mathcal{X} \ \mid \ \alpha(x) + \beta = 0\ \},$$

where $\alpha \in \mathcal{X}'$ and $\beta \in \mathbb{R}$.

In the case where $\mathcal{X}$ is an Hilbert space, the affine closed hyperplans are

$$H = \{\ x \in \mathcal{H} \ \mid (\alpha, x) + \beta = 0\ \},$$

where $\alpha \in \mathcal{X},\ \alpha \neq 0$ and $\beta \in \mathbb{R}$.

Definition A.5 (Separation). *Let A and B be two non empty subsets of $\mathcal{X}$. The affine hyperplan H whose analytical form is $\alpha(x) + \beta = 0$, separates A and B if*

$$\forall x \in A \quad \alpha(x) + \beta \leq 0 \quad et \quad \forall y \in B \quad \alpha(y) + \beta \geq 0.$$

The separation is strict if there exists $\varepsilon > 0$ such that

$$\forall x \in A \quad \alpha(x) + \beta \leq -\varepsilon \quad et \quad \forall y \in B \quad \alpha(y) + \beta \geq \varepsilon.$$

The separation Hahn-Banach theorem (geometrical form) writes

Theorem A.6. *Let A and B be two non empty, convex subsets of $\mathcal{X}$ such that $A \cap B = \emptyset$.*
• Assume that A is an open set. Then, there exists a closed affine hyperplan that separates A and B.
• Assume that A is closed and B is compact. Then, there exists a closed affine hyperplan that strictly separates A and B.

A.2.2. Subdifferential

Definition A.6 (Subdifferential). *Let be $f : V \to \mathbb{R} \cup \{+\infty\}$ and $u \in$ dom f (i.e. $f(u) < +\infty$). The subdifferential of f at u is the set $\partial f(u)$ (possibly empty) defined as follows*

$$\partial f(u) := \{\, u^* \in V' \mid \forall v \in V,\ f(v) \geq f(u) + \langle u^*, v - u \rangle \ \}.$$

Elements $u^* \in \partial f(u)$ are called subgradients. If $\partial f(u) \neq \emptyset$, f is sub differentiable at u.

Remark A.1. *1. $f : V \to \mathbb{R} \cup \{+\infty\}$ achieves its minimum at $u \in$ dom f if and only if $0 \in \partial f(u)$.*
2. If $f, g : V \to \mathbb{R} \cup \{+\infty\}$ and $u \in$ dom $f \cap$ dom g, then

$$\partial f(u) + \partial g(u) \subset \partial(f + g)(u).$$

3.As

$$\partial f(u) = \bigcap_{v \in V} \{u^* \in V' \ \mid \ \langle u^*, v - u \rangle \leq f(v) - f(u) \ \},$$

$\partial f(u)$ is a convex, weakly star closed subset of V'. .
4. For every $\lambda > 0$ we have $\partial(\lambda f)(u) = \lambda \partial f(u)$.

Theorem A.7 (Relation with Gâteaux-differentiability). *Let $f : V \to \mathbb{R} \cup \{+\infty\}$ be a convex function.*
If f is Gâteaux-differentiable at $u \in$ dom f, it is subdifferentiable at u and $\partial f(u) = \{f'(u)\}$.
Conversely, if f is finite, continuous at u and $\partial f(u)$ is a singleton, then f is Gâteaux-differentiable at u and $\partial f(u) = \{\nabla(u)\}$.

Theorem A.8 (Subdifferential of the sum of two functions). *Let f and g be convex, lower semi-continuous from V to $\mathbb{R} \cup \{+\infty\}$. Assume there exists $u_0 \in$dom $f \cap$ dom g such that f is continuous at u_0. Then*

$$\forall u \in V \qquad \partial(f + g)(u) = \partial f(u) + \partial g(u).$$

We end with a chain rule result for subdifferentiability :

Theorem A.9. *Let Λ be a linear continuous operator from V to W (both Banach spaces). Let f be convex, lower semi-continuous from V to $\mathbb{R} \cup \{+\infty\}$. Assume, there exists $u_0 \in$dom f such that f is continuous at u_0. Then*

$$\forall u \in V \qquad \partial(f \circ \Lambda)(u) = \Lambda^* \partial f(\Lambda u),$$

where Λ^* $(W' \to V')$ is the adjoint operator of Λ.

We give now an important example.

A.2.3. Case where f is a set indicatrix.
When f is the indicatrix function of a non empty subset K of V:

$$f(u) \stackrel{def}{=} 1_K(u) = \begin{cases} 0 & \text{if } u \in K, \\ +\infty & \text{else} \end{cases}$$

the subdifferential of f en $u \in V$ is the **normal cone** of K at u:

$$\partial 1_K(u) := N_K(u) = \{\ u^* \in V' \ \mid\ \forall v \in K \quad \langle u^*, v-u\rangle \leq 0\ \}.$$

Assume V is an Hilbert space and K is a non empty, closed, convex subset of V. We describe the subdifferential of 1_K at u :

Proposition A.2. *Let be $u \in K$, where K is a closed, convex (non empty) subset of the Hilbert space V. Then*

$$\lambda \in \partial 1_K(u) \Longleftrightarrow \lambda = c[u + \frac{\lambda}{c} - P_K(u + \frac{\lambda}{c})]$$

for every $c > 0$ where P_K is the projection of V on K.

Proof - We first note that $\partial 1_K(u) \subset V$ (since $V = V'$). Relation (A.5) of Theorem A.5 gives

$$\forall v \in K \qquad (w - P_K(w), v - P_K(w))_V \leq 0.$$

As

$$\lambda \in \partial 1_K(u) \iff \forall v \in K \qquad (\lambda, v-u)_V \leq 0$$

we get for every $c > 0$

$$\forall v \in K \qquad \left(u + \frac{\lambda}{c} - u, v - u\right)_V \leq 0.$$

So, letting $w = u + \dfrac{\lambda}{c}$ we obtain

$$\lambda \in \partial 1_K(u) \Longleftrightarrow u = P_K(u + \frac{\lambda}{c}) \Longleftrightarrow \lambda = c[u + \frac{\lambda}{c} - P_K(u + \frac{\lambda}{c})].$$

A.2.4. Legendre-Fenchel transformation
Definition A.7 (Legendre-Fenchel transformation). *Let be $f : V \to \mathbb{R} \cup \{+\infty\}$. The Legendre-Fenchel conjugate of f is the function $f^* : V' \to \bar{\mathbb{R}}$ defined as*

$$\forall u^* \in V' \qquad f^*(u^*) = \sup_{u \in V}\{\ \langle u^*, u\rangle - f(u)\ \}. \tag{A.8}$$

Proposition A.3. *For any function $f : V \to \mathbb{R} \cup \{+\infty\}$, the conjugate f^* is convex and lower semi-continous for the weak star topology.*

The following result is very useful when dealing with norms or semi-norms:

Proposition A.4. *Let f be a function positively homogeneous (taking at least one finite value) from V to $\mathbb{R} \cup \{+\infty\}$, that is*

$$\forall \lambda \in \mathbb{R}, \forall x \in V \qquad f(\lambda x) = |\lambda| f(x) \ . \tag{A.9}$$

Then, there exists a closed, convex set $K \subset V'$ such that $f^* = 1_K$.

Proof - Let f be a function positively homogeneous (taking at least one finite value) from V to $\mathbb{R} \cup \{+\infty\}$. Let be $u^* \in V'$.
• If there exists $u_o \in V$ such that $\langle u^*, u_o \rangle - f(u_o) > 0$. With (A.9) we get for every $\lambda > 0$

$$\langle u^*, \lambda u_o \rangle - f(\lambda u_o) = \lambda [\langle u^*, u_o \rangle - f(u_o)] \leq f^*(u^*).$$

Passing to the limit as $\lambda \to +\infty$ gives $f^*(u^*) = +\infty$.
• Otherwise,

$$\forall u \in V \qquad \langle u^*, u \rangle - f(u) \leq 0,$$

so $f^*(u^*) \leq 0$. The definition of f^* gives

$$\langle u^*, 0 \rangle - f(0) \leq f^*(u^*) \ .$$

Moreover (A.9) implies $f(0) = f(n \cdot 0) = nf(0)$ for every $n \in \mathbb{N}$. So $f(0) = 0$ and finally $f^*(u^*) = 0$.
Let us set $K = \{u^* \in V^* \mid f^*(u^*) = 0 \}$. We have just proved that $f^* = 1_K$. As f^* is convex and lower semi-continuous then K is convex and closed.

Next theorem is one of the most important result for the convex duality theory. It makes the relation between the so-called primal problem (find the infimum of $f + g$) with the dual one which deals with maximization of $f^* + g^*$.

Theorem A.10. *Let $f, g : V \to \mathbb{R} \cup \{+\infty\}$ be convex functions such that there exists $u_0 \in$ dom g and f continuous at u_0. Then*

$$\inf_{u \in V} (f(u) + g(u)) = \max_{u^* \in V'} (-f^*(u^*) - g^*(-u^*)) \, ,$$

where f^* and g^* are the Legendre-Fenchel conjugates of f and g respectively.

Finally we have an "inversion" result:

Theorem A.11. *Let $f : V \to \mathbb{R} \cup \{+\infty\}$ be a lower semi-continuous, convex function, with at least one finite value. Then, for every $u \in V$*

$$f(u) = \max_{u^* \in V'} (< u^*, u > - f^*(u^*)) \, .$$

This means that $f^{**} = f$.

A.2.5. Relation with subdifferentiablity

Theorem A.12. *Let be $f : V \to \mathbb{R} \cup \{+\infty\}$ and f^* its conjugate function. Then*

$$u^* \in \partial f(u) \Longleftrightarrow f(u) + f^*(u^*) = \langle u^*, u \rangle \, .$$

Corollary A.4. *If $f : V \to \mathbb{R} \cup \{+\infty\}$ is a lower semi-continuous , convex function, with at least one finite value, then*

$$u^* \in \partial f(u) \Longleftrightarrow u \in \partial f^*(u^*).$$

A.3. Sobolev spaces

This subsection gives basic results on Sobolev spaces. For more details, one can refer to Adams (1978), Attouch et al. (2006).
Let Ω be a bounded, open subset of $\mathbb{R}^n$, $(n \leq 3)$ with a smooth boundary Γ. We call $\mathcal{D}(\Omega)$ the space of $\mathcal{C}^\infty$ functions with compact support in Ω. The dual space $\mathcal{D}'(\Omega)$ is the space of **distributions** on Ω.
Fore every distribution $u \in \mathcal{D}'(\Omega)$, the derivative $\dfrac{\partial u}{\partial x_i}$ is defined (by duality) as following:

$$\forall \varphi \in \mathcal{D}(\Omega) \qquad \left\langle \frac{\partial u}{\partial x_i}, \varphi \right\rangle_{\mathcal{D}'(\Omega),\mathcal{D}(\Omega)} \stackrel{def}{=} - \left\langle u, \frac{\partial \varphi}{\partial x_i} \right\rangle_{\mathcal{D}'(\Omega),\mathcal{D}(\Omega)} \, .$$

The derivative of u in the distribution sense writes $D_i u, \dfrac{\partial u}{\partial x_i}$ or $\partial_i u$.
If $\alpha \in \mathbb{N}^n$, we note $D^\alpha u = \partial_1^{\alpha_1} u \cdots \partial_n^{\alpha_n} u$ et $|\alpha| = \alpha_1 + \cdots + \alpha_n$; we get

$$\forall \varphi \in \mathcal{D}(\Omega) \qquad \langle D^\alpha u, \varphi \rangle_{\mathcal{D}'(\Omega),\mathcal{D}(\Omega)} = (-1)^{|\alpha|} \langle u, D^\alpha \varphi \rangle_{\mathcal{D}'(\Omega),\mathcal{D}(\Omega)} \, .$$

Definition A.8 (Sobolev spaces). *The Sobolev spaces $W^{p,m}(\Omega)$, $H^m(\Omega)$ are defined as:*

$$W^{p,m}(\Omega) = \{ \, u \in L^p(\Omega) \mid D^\alpha u \in L^p(\Omega), \ |\alpha| \leq m \, \} \, ,$$

$$H^m(\Omega) := W^{2,m}(\Omega) \{ \, u \in \mathcal{D}'(\Omega) \mid D^\alpha u \in L^2(\Omega), \ |\alpha| \leq m \, \} \, ,$$

$$H^1(\Omega) = \{ \, u \in L^2(\Omega) \mid \frac{\partial u}{\partial x_i} \in L^2(\Omega), \ i = 1 \cdots n \, \} \, .$$

Remark A.2. $H^0(\Omega) = L^2(\Omega)$.

Let us give main basic properties of the above Sobolev spaces:

Proposition A.5. *$H^m(\Omega)$ endowed with the inner product*

$$(u,v)_m = \sum_{|\alpha|\leq m} \int_\Omega D^\alpha u(x)\ D^\alpha v(x)\ dx\ ,$$

is an Hilbert space.

Proposition A.6.

$$H^m(\Omega) \subset H^{m'}(\Omega)$$

with continuous embedding for $m \geq m'$.

Definition A.9 (Functions with null trace).

$$H_o^1(\Omega) = \{\ u \in H^1(\Omega)\ |\ u_{|\Gamma} = 0\ \}\ .$$

$$H_o^m(\Omega) = \{\ u \in H^1(\Omega)\ |\ \frac{\partial^j u}{\partial n^j}_{|\Gamma} = 0,\ j = 1, \cdots, m-1\}\ ,$$

where $\dfrac{\partial}{\partial n}$ is the outer normal derivative of u on Γ :

$$\frac{\partial u}{\partial n} = \sum_{i=1}^{n} \frac{\partial u}{\partial x_i} \cos(\vec{n}, \vec{e_i})\ ,$$

where $\vec{n}$ is the outer normal vector to Γ.

Definition A.10 (Duality). *For every $m \in \mathbb{N}$, we denote $H^{-m}(\Omega)$ the dual space of $H_o^m(\Omega)$.*

Theorem A.13 (Rellich). *If Ω is a bounded open subset of $\mathbb{R}^n$, then for every $m \in \mathbb{N}$, the embedding of $H_o^{m+1}(\Omega)$ in $H_o^m(\Omega)$ is compact.*

In particula, $H_o^1(\Omega)$ is compactly embedded in $L^2(\Omega)$. From a pratical point of view, this means that any sequence whose $H_o^1(\Omega)$ norm is bounded weakly converges in $H_o^1(\Omega)$ and strongly in $L^2(\Omega)$ (up to a subsequence).

B. 2D-MATLAB © codes

B.1. Problem ($\mathcal{P}_1$)

```
%========================================
function [usol]= ROFNest(imag, lambda,itmax);
%========================================
% Denoising with ROF model using   Nesterov algorithm
% usol =argmin lambda * J(u) + | udata-u|^2

udata=double(imag);
rho= 0.25;
tol =1e-5
    usol= udata- proj_nesterov(udata,itmax,lambda);
end

%****************** subfunctions************

%---------------------------------------------------
function [w]=proj_nesterov(imag,itmax,lambda)
%---------------------------------------------------
% Compute the projection w  on lambda K
%  with  Nesterov -Weiss

udata=double(imag);
L=8;

%initialization

g1=zeros(size(imag)) ;
g2=zeros(size(imag)) ;
x1=zeros(size(imag)) ;
x2=zeros(size(imag)) ;
itnest=0;

%iteration

for k=0:itmax-1
    x=[x1,x2];
    [eta1,eta2]=grad(-divdiscret(x1,x2)+udata/lambda);
    normg=sqrt((x1-eta1/L).^2+(x2-eta2/L).^2);
    y1=(x1-eta1/L)./max(1,normg);
```

```
    y2=(x2-eta2/L)./max(1,normg);
    g1=g1+((k+1)/2)*eta1;
    g2=g2+((k+1)/2)*eta2;
    normh=sqrt(g1.^2+g2.^2)/L;
    v1=(-g1/L)./max(1,normh);
    v2=(-g2/L)./max(1,normh);
    x1=(2/(k+3))*v1+((k+1)/(k+3))*y1;
    x2=(2/(k+3))*v2+((k+1)/(k+3))*y2;
    itnest=itnest+1
end

%solution
w=lambda*divdiscret(x1,x2);
end
%-----------------------------
function [u1, u2]= grad(u)
%-----------------------------
% compute the gradient  with forward finite difference

[n1 n2] = size(u);

for i=1:n1-1
    u1(i,:)= u(i+1,:)-u(i,:);
end
u1(n1,:)= 0;

for j=1:n2-1
    u2(:,j)= u(:,j+1)-u(:,j);
end
u2(:,n2)= 0;
end

%---------------------------------
function  q = divdiscret(p1, p2)
%---------------------------------
% compute the divergence  of( p1 p2)
% according the finite difference scheme for gradient

[n1,n2] = size(p1);

for i=2:n1-1
```

```
   q1(i,:)= p1(i,:)-p1(i-1,:);
 end
q1(1,:) = p1(1,:);
q1(n1,:) = - p1(n1-1,:);

for j=2:n2-1
   q2( :,j)= p2(:,j)-p2(:,j-1);
 end
q2(:,1) = p2(:,1);
q2(:,n2) = - p2(:,n2-1);

q=q1+q2;
end
```

B.2. Problem $\mathcal{P}_2$

```
%=========================================
function [usol]= ROF2Nest(imag,lambda,itmax)
%=========================================
%Denoising with ROF2 model using   Nesterov algorithm
%usol =arg min lambda * J_2(u) + | udata-u|^2
udata=double(imag); rho= 0.25;
usol= udata- proj2_nesterov(udata,itmax,lambda);
 end
%****************** subfunctions************
%-----------------------------------------------------------
 function [w2]= proj2_nesterov(imag,itmax,lambda)
%-----------------------------------------------------------
% Compute the projection  w2  on lambda K_2
%  with  Nesterov -Weiss
 udata=double(imag);   L=64;
% Initialization
 k=0;
g1=zeros(size(imag)) ; g2=g1;g3=g1;g4=g1;
x1=zeros(size(imag)) ;x2=x1;x3=x1;x4=x1;
% Iteration
while  (k< itmax)
  x=[x1,x2,x3,x4];
  eta=gradsec2(divdiscret2(x)-udata/lambda);
  [h1,h2]=size(eta);
  eta1=eta(:,1:h2/4);eta2=eta(:,h2/4+1:2*h2/4);
  eta3=eta(:,2*h2/4+1:3*h2/4); eta4=eta(:,3*h2/4+1:h2);
```

```
  normg=sqrt((x1-eta1/L).^2+(x2-eta2/L).^2+...
     (x3-eta3/L).^2+(x4-eta4/L).^2);
  y1=(x1-eta1/L)./max(1,normg); y2=(x2-eta2/L)./max(1,normg);
  y3=(x3-eta3/L)./max(1,normg); y4=(x4-eta4/L)./max(1,normg);
  g1=g1+((k+1)/2)*eta1; g2=g2+((k+1)/2)*eta2;
  g3=g3+((k+1)/2)*eta3;  g4=g4+((k+1)/2)*eta4;
  normh=sqrt(g1.^2+g2.^2+g3.^2+g4.^2)/L;
  v1=(-g1/L)./max(1,normh);v2=(-g2/L)./max(1,normh);
  v3=(-g3/L)./max(1,normh);v4=(-g4/L)./max(1,normh);
  x1=(2/(k+3))*v1+((k+1)/(k+3))*y1;x2=(2/(k+3))*v2+((k+1)/(k+3))*y2;
  x3=(2/(k+3))*v3+((k+1)/(k+3))*y3 x4=(2/(k+3))*v4+((k+1)/(k+3))*y4;
  x=[x1,x2,x3,x4];
  k=k+1;
end
%Solution
w2=lambda*divdiscret2(x);
end
%----------------------------
 function H=gradsec2(phi)
%----------------------------
%  compute the Hessian  H of  phi

[n1, n2]= size(phi);
g11=0*phi; g22=0*phi; g12=0*phi; g21 = 0*phi;

for i=2:n1-1
    g11(i,:)= phi(i+1,:)-2*phi(i,:)+ phi(i-1,:);
end
g11(1,:)=phi(2,:)-phi(1,:);
g11(n1,:)=phi(n1-1,:)-phi(n1,:)
for j=2:n2-1
    g22(:,j)= phi(:,j+1)-2*phi(:,j)+ phi(:,j-1);
 end
g22(:,1)=phi(:,2)-phi(:,1);
g22(:,n2)=phi(:,n2-1)-phi(:,n2);
for i=2:n1
    for j=1:n2-1
      g12(i,j)=  (phi(i,j+1)- phi(i,j)- phi(i-1,j+1)+phi(i-1,j));
    end
end
for i=1:n1-1
```

```
    for j=2:n2
      g21(i,j)=(phi(i+1,j)- phi(i,j)- phi(i+1,j-1)+phi(i,j-1));
    end
end
g21=g12;
H=[g11 g22 g12 g21];
end
%--------------------------------
 function DV=divdiscret2(p)
%--------------------------------
%  Compute the adjoint of the Hessian : H^*
%  at p=( p11 p12 p21 p22)

[l,c]=size(p); n=c/4;

p11=p(:,1:n) ; p22=p(:,n+1:2*n);
p12=p(:,2*n+1:3*n);p21=p(:,3*n+1:4*n);

DV1=zeros(l,n);
for i=2:l-1
     DV1(i,:)= p11(i-1,:)-2*p11(i,:)+p11(i+1,:);
end
DV1(1,:)=p11(2,:)-p11(1,:);
DV1(l,:)=p11(l-1,:)-p11(l,:);
DV2=zeros(l,n);
for j=2:n-1
     DV2(:,j)= p22(:,j-1)-2*p22(:,j)+p22(:,j+1);
end
DV2(:,1)=p22(:,2)-p22(:,1);
DV2(:,n)=p22(:,n-1)-p22(:,n);
DV3=zeros(l,n);
for i=2:l-1
 for j=2:n-1
    DV3(i,j)= p12(i,j-1)-p12(i,j)-p12(i+1,j-1)+p12(i+1,j) ;
  end
  DV3(i,1)= p12(i+1,1)-p12(i,1); DV3(i,n)= p12(i,n-1)-p12(i+1,n-1) ;
end
for j=2:n-1
    DV3(1,j)=p12(2,j)-p12(2,j-1);
    DV3(l,j)=p12(l,j-1)-p12(l,j);
end
```

```
DV3(1,1)= p12(2,1); DV3(1,n)= - p12(2,n-1);
DV3(l,1)= - p12(l,1); DV3(l,n)= p12(l,n-1);
DV4=zeros(l,n);
for i=2:l-1
    for j=2:n-1
         DV4(i,j)= p21(i-1,j)-p21(i,j)-p21(i-1,j+1)+p21(i,j+1);
    end
    DV4(i,1)= p21(i,2)-p21(i-1,2);
    DV4(i,n)= p21(i-1,n)-p21(i,n)   ;
end
for j=2:n-1
    DV4(1,j)=p21(1,j+1)-p21(1,j);
    DV4(l,j)=p21(l-1,j)-p21(l-1,j+1);
end
DV4(1,1)= p21(1,2);
DV4(1,n)= - p21(1,n);
DV4(l,1)= - p21(l-1,2);
DV4(l,n)= p21(l-1,n);
DV=DV1+DV2+DV3+DV4;
end
```

B.3. Problem $(\mathcal{P}_{\lambda,\mu})$

```
function [u,v,w]= mixte(udata,lam,mu,kmax);
% Fixed point algorithm for the full second order model
%Input : udata: image to decompose
%          lambda (lam) and mu - kmax : maximal number of iterations
% Output : u -BV part, v- BV2 part, w- L2 part
ud =double(udata);
% Number of iterations od Nesterov algorithms and descent step
 itmax=30 ;
 ro=0.25;
% Normalisation  of the image
% ud  is double and normalized : between  0 and 1
 a=min(min(ud));
 b=max(max(ud));
ud= (ud-a)/(b-a);
% Initialization
v0=ud; u0= 0*ud;
k=1;err=1;
%Iteration
while  (k < kmax) & (err > 1e-05)
```

```
    u=(1-ro)*u0+ ro*(ud-v0 -proj_nesterov(ud-v0,itmax,lam));
    v= (1-ro)*v0+ ro*(ud-u- proj2_nesterov(ud-u,itmax,mu));
erru=max(max (abs(u-u0)))/max(max (abs(u0))) ;
errv=max(max (abs(v-v0)))/max(max (abs(v0))) ;
err=max(erru,errv) ;
u0=u;v0=v;
k=k+1
end
% Solution
w=ud-u-v;
end
```

The related functions are described in the two previous subsections.

References

Acar, R., Vogel, C., 1994. Analysis of bounded variation penalty methods for ill-posed problems. Inverse Problems 10 6 (1217-1229).

Adams, R. A., 1978. Sobolev spaces. Academic Press, Springer Verlag.

Ambrosio, L., Fusco, N., Pallara, D., 2000. Functions of Bounded Variation and Free Discontinuity Problems. Oxford Science Publications.

Ashikhmin, M., 2001. Synthesizing natural textures. In: SI3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics. ACM Press, New York, NY, USA, pp. 217–226.

Attouch, H., Buttazzo, G., Michaille, G., 2006. Variational analysis in Sobolev and BV spaces. Vol. 6 of MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, applications to PDEs and optimization.

Aubert, G., Aujol, J.-F., 2005. Modeling very oscillating signals. Application to image processing. Appl. Math. Optim. 51 (2), 163–182.

Aubert, G., Kornprobst, P., 2006. Mathematical Problems in Image Processing. Vol. 147 of Applied Mathematical Science. Springer.

Aujol, J.-F., 2009. Some first-order algorithms for total variation based image restoration. J. Math. Imaging Vision 34 (3), 307–327.

Aujol, J.-F., Aubert, G., Blanc-Féraud, L., 2003. Wavelet-based level set evolution for classification of textured images. IEEE Trans. Image Process. 12 (12), 1634–1641.

Aujol, J.-F., Aubert, G., Blanc-Féraud, L., Chambolle, A., 2005. Image decomposition into a bounded variation component and an oscillating component. J. Math. Imaging Vision 22 (1), 71–88.

Aujol, J.-F., Chambolle, A., 2005. Dual norms and image decomposition models. International Journal of Computer Vision 63 (1), 85–104.

Aujol, J.-F., Ladjal, S., Masnou, S., 2010. Exemplar-based inpainting from a variational point of view. SIAM J. Math. Anal. 42 (3), 1246–1285.

Azé, D., 1997. Eléments d'analyse convexe et variationnelle. Ellipses.

Bar-Joseph, Z., El-Yaniv, R., Lischinski, D., Werman, M., 2001. Texture mixing and texture movie synthesis using statistical learning. IEEE Transactions on Visualization and Computer Graphics 7 (2), 120–135.

Barbu, V., Precupanu, T., 1978. Convexity and Optimization in Banach Spaces. Sijthoff & Noordhoff.

Bargteil, A. W., Sin, F., Michaels, J. E., Goktekin, T. G., O'Brien, J. F., 2006. A texture synthesis method for liquid animations. In: Proc. of SCA '06. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 345–351.

Bergounioux, M., 2011. On poincaré-wirtinger inequalities in bv - spaces. Control & Cybernetics 4 (40).

Bergounioux, M., Piffet, L., 2010. A second-order model for image denoising. Set-Valued Var. Anal. 18 (3-4), 277–306.

Bergounioux, M., Piffet, L., 2013. A full second order variational model for multiscale texture analysis. Computational Optimization and Applications 54, 215–237.

Bergounioux, M., Tran, M., 2011. Anisotropic second order model for 3d-texture extraction. In: Bergounioux, M. (Ed.), Mathematical Image Processing. Vol. 5. Springer, pp. 41–57.

Bredies, K., Kunisch, K., Pock, T., 2010. Total generalized variation. SIAM J. Imaging Sci. 3 (3), 492–526.

Bredies, K., Kunisch, K., Valkonen, T., 2011. Properties of l^1 - TGV^2 : The one-dimensional case. Submitted for publication 48.

Brezis, H., 1987. Analyse Fonctionnelle. Masson, Paris.

Buades, A., Coll, B., Morel, J.-M., 2006. The staircasing effect in neighborhood filters and its solution. IEEE transaction on medical imaging 15 (6), 1499 – 1505.

Buades, A., Le, T. M., Morel, J.-M., Vese, L. A., 2010. Fast cartoon + texture image filters. IEEE Trans. Image Process. 19 (8), 1978–1986.

Casaburi, D., D'Amore, L., Galletti, A., Marcellino, L., 2011. A numerical algorithm for image sequence inpainting that preserves fine textures. Int. J. Comput. Math. 88 (11), 2331–2347.

Caselles, V., Chambolle, A., Novaga, M., 2007. The discontinuity set of solutions of the tv denoising problem and some extensions. Multiscale Model. Simul. 3 (6), 879894.

Chambolle, A., 2004. An algorithm for total variation minimization and applications. Journal of Mathematical Imaging and Vision 20, 89–97.

Chen, H., Zhu, S. C., 2006. A generative sketch model for human hair analysis and synthesis. IEEE Trans. Pattern Anal. Mach. Intell 28 (7), 1025–1040.

Clarke, A. D. F., Green, P. R., Halley, F., Chantler, M. J., 2011. Similar symmetries: the role of wallpaper groups in perceptual texture similarity. Symmetry 3 (2), 246–264.

De Bonet, J. S., 1997. Multiresolution sampling procedure for analysis and synthesis of texture images. In: Proc. Siggraph '97. ACM Press/Addison-Wesley Publishing Co., pp. 361–368.

Demengel, F., 1984. Fonctions à hessien borné. Annales de l'institut Fourier 34 (2), 155–190.

Doretto, G., Chiuso, A., Wu, Y. N., Soatto, S., Feb. 2003. Dynamic textures. International Journal of Computer Vision 51 (2), 91–109.

Duval, V., Aujol, J.-F., Vese, L. A., 2010. Mathematical modeling of textures: application to color image decomposition with a projected gradient algorithm. J. Math. Imaging Vision 37 (3), 232–248.

Eckley, I. A., Nason, G. P., Treloar, R. L., 2010. Locally stationary wavelet fields with application to the modelling and analysis of image texture. J. R. Stat. Soc. Ser. C. Appl. Stat. 59 (4), 595–616.

Efros, A. A., Leung, T. K., 1999. Texture synthesis by non-parametric sampling. In: ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2. IEEE Computer Society, p. 1033.

Ekeland, I., Temam, R., 1999. Convex Analysis and Variational problems. SIAM Classic in Applied Mathematics, 28.

Elad, M., Starck, J.-L., Donoho, D., Querre, P., 2005. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). Journal on Applied and Computational Harmonic Analysis 19, 340–358.

Fadili, M., Starck, J.-L., Murtagh, F., 2007. Inpainting and zooming using sparse representations. Computer Journal.

Foster, N., Metaxas, D. N., Sep. 1996. Realistic animation of liquids. Graphical Models and Image Processing 58 (5), 471–483.

Galerne, B., Gousseau, Y., Morel, J.-M., 2011. Random phase textures: theory and synthesis. IEEE Trans. Image Process. 20 (1), 257–267.

Garnett, J. B., Jones, P. W., Le, T. M., Vese, L., 2011. Modeling oscillatory components with the homogeneous spaces $B\dot{M}O^{-\alpha}$ and $\dot{W}^{-\alpha,p}$. Pure Appl. Math. Q. 7 (2, Special Issue: In honor of Frederick W. Gehring, Part 2), 275–318.

Garnett, J. B., Le, T. M., Meyer, Y., 2007. Image decompositions using bounded variation and generalized homogeneous Besov spaces. Appl. Comput. Harmon. Anal. 23 (1), 25–56.

Gilles, J., Meyer, Y., 2010. Properties of $\mathrm{BV} - G$ structures + textures decomposition models. Application to road detection in satellite images. IEEE Trans. Image Process. 19 (11), 2793–2800.

Grzegorzek, M., 2010. A system for 3D texture-based probabilistic object recognition and its applications. PAA Pattern Anal. Appl. 13 (3), 333–348.

Guillot, L., Le Trong, E., Rozenbaum, O., Bergounioux, M., Rouet, J.-L., 2009. A mixed model of active geodesic contours with gradient vector flows for x-ray microtomography segmentation. In: Actes de la conférence "Mathématiques pour l'image".

Hinterberger, W., Scherzer, O., 2006. Variational methods on the space of functions of bounded Hessian for convexification and denoising. Computing 76 (1-2), 109–133.

Hiriart-Urruty, J.-B., 1998. Optimisation et analyse convexe. PUF.

Jennane, R., Touvier, J., Bergounioux, M., Lespessailles, E., 2013. A variational model for trabecular bone radiograph characterization.

Karoui, I., Fablet, R., Boucher, J.-M., Augustin, J.-M., 2010. Variational region-based segmentation using multiple texture statistics. IEEE Trans. Image Process. 19 (12), 3146–3156.

Khelifi, F., Jiang, J., 2011. k-NN regression to improve statistical feature extraction for texture retrieval. IEEE Trans. Image Process. 20 (1), 293–298.

Kim, Y., Vese, L. A., 2009. Image recovery using functions of bounded variation and Sobolev spaces of negative differentiability. Inverse Probl. Imaging 3 (1), 43–68.

Knoll, F., Bredies, K., Pock, T., Stollberger, R., 2011. Second order total generalized variation (tgv) for mri. Magnetic Resonance in Medicine 65 (2), 480–491.

Kwatra, V., Adalsteinsson, D., Kim, T., Kwatra, N., Carlson, M., Lin, M. C., 2007. Texturing fluids. IEEE Trans. Vis. Comput. Graph 13 (5), 939–952.

Kwatra, V., Essa, I., Bobick, A., Kwatra, N., Jul. 2005. Texture optimization for example-based synthesis. ACM Transactions on Graphics 24 (3), 795–802.

Kwatra, V., Schödl, A., Essa, I., Turk, G., Bobick, A., Jul. 2003. Graphcut textures: image and video synthesis using graph cuts. ACM Transactions on Graphics 22 (3), 277–286.

Le, T., Lieu, L., Vese, L., 2009. (Φ, Φ^*) image decomposition models and minimization algorithms. J. Math. Imaging Vision 33 (2), 135–148.

Le, T. M., Vese, L. A., 2005. Image decomposition using total variation and div(BMO). Multiscale Model. Simul. 4 (2), 390–423.

Lefebvre, S., Hoppe, H., 2005. Parallel controllable texture synthesis. ACM Trans. Graph. 24 (3), 777–786.

Lewis, J.-P., 1984. Texture synthesis for digital painting. SIGGRAPH Comput. Graph. 18 (3), 245–252.

Lieu, L. H., Vese, L. A., 2008. Image restoration and decomposition via bounded total variation and negative Hilbert-Sobolev spaces. Appl. Math. Optim. 58 (2), 167–193.

Maurel, P., Aujol, J.-F., Peyré, G., 2011. Locally parallel texture modeling. SIAM J. Imaging Sci. 4 (1), 413–447.

Meyer, Y., 2001. Oscillating Patterns in Image Processing and Nonlinear Evolution Equations. University Lecture Series, Vol. 22, AMS.

Mumford, D., Gidas, B., 2001. Stochastic models for generic images. Q. Appl. Math. LIV (1),

85–111.
Nesterov, Y., 2005. Smooth minimization of non-smooth functions. Mathematic Programming, Ser. A 103, 127–152.
Osher, S., Fatemi, E., Rudin, L., 1992. Nonlinear total variation based noise removal algorithms. Physica D 60, 259–268.
Osher, S., Sole, A., L., V., 2003. Image decomposition and restoration using total variation minimization and the H^1 norm. SIAM Journal on Multiscale Modeling and Simulation 1-3, 349–370.
Osher, S., Vese, L., 2003. Modeling textures with total variation minimization and oscillating patterns in image processing. Journal of Scientific Computing 19 (1-3), 553–572.
Paget, R., Longstaff, I. D., Jun. 1998. Texture synthesis via a noncausal nonparametric multiscale markov random field. IEEE Trans. Image Processing 7 (6), 925–931.
Peyré, G., 2009. Sparse modeling of textures. J. Math. Imaging Vision 34 (1), 17–31.
Peyré, G., 2010. Texture synthesis with grouplets. IEEE Trans. Pattern Analysis and Machine Intelligence 32 (4), 733–746.
Piffet, L., 2011. A locally anisotropic model for image texture extraction. In: Bergounioux, M. (Ed.), Mathematical Image Processing. Vol. 5. Springer, pp. 141–58.
Portilla, J., Simoncelli, E. P., 2000. A parametric texture model based on joint statistics of complex wavelet coefficients. Int. J. Comput. Vision 40 (1), 49–70.
Ramrishnan, S., Selvan, S., 2008. Classification of soil texture based on wavelet domain singular values. Int. J. Tomogr. Stat. 9 (S08), 33–50.
Ring, W., 2000. Structural properties of solutions of total variation regularization problems. ESAIM, Math Modelling and Numerical Analysis 34, 799–840.
Shahidi, R., Moloney, C., 2009. Decorrelating the structure and texture components of a variational decomposition model. IEEE Trans. Image Process. 18 (2), 299–309.
Strong, D. M., Aujol, J.-F., Chan, T. F., 2006. Scale recognition, regularization parameter selection, and Meyer's G norm in total variation regularization. Multiscale Model. Simul. 5 (1), 273–303.
Tadmor, E., Nezzar, S., Vese, L., 2004. A multiscale image representation using hierarchical (BV, L^2) decompositions. Multiscale Model. Simul. 2 (4), 554–579 (electronic).
Tran, M., Peteri, R., Bergounioux, M., 2012. Denoising 3d medical images using a second order variational model and wavelet shrinkage. In: ICIAR Conference 2012.
Weiss, P., Aubert, G., Blanc Féraud, L., 2009. Efficient schemes for total variation minimization under constraints in image processing. SIAM Journal on Scientific Computing 31 (3), 2047–2080.
Wen, Y.-W., Chan, R. H., Ching, W.-K., 2011. Simultaneous cartoon and texture reconstruction for image restoration by bivariate function. Appl. Anal. 90 (8), 1275–1289.
Yin, W., Goldfarb, D., Osher, S., 2007. A comparison of three total variation based texture extraction models. J. Vis. Commun. Image 18, 240–252.
Zhu, S. C., Wu, Y., Mumford, D., 1998. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. Int. J. Comput. Vision 27 (2), 107–126.
Ziemer, W., 1989. Weakly Differentiable Functions - Sobolev Space and Functions of Bounded Variation. Indiana University.