Inf-Convolution Model for Image Processing- Numerical Experimentation

Maïtine Bergounioux

INF-CONVOLUTION MODEL FOR IMAGE PROCESSING - NUMERICAL EXPERIMENTATION

M. BERGOUNIOUX

Abstract. This paper is a companion paper of [4] where a second order image decomposition model to perform denoising and texture extraction has been studied. Here we perform some numerical experimentation to make the behavior of the model as clear as possible. For highly textured images the model gives a two-scale texture decomposition

Keywords.- Second order total variation, image decomposition, variational method, inf-convolution, texture extraction

1. Introduction

In [4] we have investigated a inf-convolution type model for texture extraction. More precisely, we assumed that an image (in $L^2(\Omega)$) can be split in three components: a smooth (continuous) part v, a *cartoon* (piecewise constant) part u and an oscillating part w that should involve noise and/or fine textures. The oscillating part of the image is included in the remainder term $w = u_d - u - v$, while v is the smooth part (in $BH(\Omega)$) and u belongs to $BV(\Omega)$: we hope u to be piecewise constant so that its jump set gives the image contours. For highly textured images, the model provides a two-scale texture decomposition: u can be viewed as a *macro-texture* (large scale) whose oscillations are not too large and w is the *micro-texture* (mmuch more oscillating) that contains the noise.

We now assume that u_d belongs to $L^2(\Omega)$ and that the image we want to recover can be decomposed as $u_d = w+u+v$ where u, v and w are functions that characterize different parts of u_d. Components belong to different functional spaces: v is the (smooth) second order part and belongs to $BH(\Omega)$, u is a $BV(\Omega)$ component and $w \in L^2(\Omega)$ is the remainder term. We consider the following cost functional defined on $BV(\Omega) \times BH(\Omega)$:

$$\mathcal{F}_{\lambda,\mu}(u,v) = \frac{1}{2}\|u_d - u - v\|^2_{L^2(\Omega)} + \lambda TV(u) + \mu TV^2(v), \tag{1.1}$$

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

$$\inf\{\ \mathcal{F}_{\lambda,\mu}(u,v) \mid (u,v) \in BV(\Omega) \times BH_0(\Omega)\ \} \tag{$\mathcal{P}_{\lambda,\mu}$}$$

We refer to [4] and the references therein for the different spaces definition.

[0] Université d'Orléans , UFR Sciences, Math., Labo. MAPMO, UMR 7349, Route de Chartres, BP 6759, 45067 Orléans cedex 2,

We expect v to be the smooth *colored* part of the image, u to be a $BV(\Omega)\backslash BH(\Omega)$ function which derivative is a measure supported by the contours and $w := u_d - u - v \in L^2$ is the noise and/or small textures (we shall detail this point later). We have proved in [4] that problem $(\mathcal{P}_{\lambda,\mu})$ has at least an optimal solution (u^*, v^*) in $BV(\Omega) \times BH_0(\Omega)$. Moreover the dual problem to $(\mathcal{P}_{\lambda,\mu})$ writes

$$\inf_{w\in\lambda\mathcal{K}_1\cap\mu\mathcal{K}_2} \frac{1}{2}\|u_d - w\|_2^2. \tag{1.2}$$

where where $\mathcal{K}_1 = \overline{\mathbf{K}_1}$ is the L^2-closure of

$$\mathbf{K}_1 := \left\{\xi = \text{div } \varphi \mid \varphi \in \mathcal{C}_c^1(\Omega),\ \|\varphi\|_\infty \leq 1\right\}. \tag{1.3}$$

and and $\mathcal{K}_2 \supset \overline{\mathbf{K}_2}$ with $\overline{\mathbf{K}_2}$ is the L^2-closure of

$$\mathbf{K}_2 := \left\{\xi = \text{div}^2\psi \mid \psi \in \mathcal{C}_c^2(\Omega, \mathbb{R}^{d\times d}),\ \|\psi\|_\infty \leq 1 \right\}. \tag{1.4}$$

The unique solution w^* is the L^2-projection of u_d on the closed convex set $\lambda\mathcal{K}_1\cap\mu\mathcal{K}_2$:

$$w^* = \Pi_{\lambda\mathcal{K}_1\cap\mu\mathcal{K}_2}(u_d)\ .$$

Next we have a relation between the solutions to $(\mathcal{P}_{\lambda,\mu})$ and the (unique) solution of the dual problem.

Theorem 1.1. *1. Let w^* be the (unique) solution to the dual problem $(\mathcal{P}_{\lambda,\mu})^*$:*

$$w^* = \Pi_{\lambda\mathcal{K}_1\cap\mu\mathcal{K}_2}(u_d)\ .$$

Then there exists $(\bar{u}, \bar{v}) \in BV(\Omega) \times BH_0(\Omega)$ an optimal solution to $(\mathcal{P}_{\lambda,\mu})$ such that

$$w^* = u_d - \bar{u} - \bar{v} \text{ and } w^* \in \partial\Phi_\mu^2(\bar{v}) \cap \partial\Phi_\lambda^1(\bar{u})\ .$$

2. Conversely, if $(\bar{u}, \bar{v}) \in BV(\Omega) \times BH_0(\Omega)$ is any solution to $(\mathcal{P}_{\lambda,\mu})$ then

$$\bar{w} = u_d - \bar{u} - \bar{v} = \Pi_{\lambda\mathcal{K}_1\cap\mu\mathcal{K}_2}(u_d)\ . \tag{1.5}$$

Here

$$\Phi_\lambda^1(u) = \begin{cases} \lambda TV(u) & \text{if } u \in BV(\Omega) \\ +\infty & \text{else.} \end{cases} \quad \text{and} \quad \Phi_\mu^2(v) = \begin{cases} \mu TV^2(v) & \text{if } v \in BH_0(\Omega) \\ +\infty & \text{else.} \end{cases}$$

2. Numerical aspects

2.1. **Discretized problem and algorithm.** We assume that the image is rectangular with size $N \times M$. We note $X := \mathbb{R}^{N\times M} \simeq \mathbb{R}^{NM}$ endowed with the usual (normalized) inner product and the associated Euclidean norm

$$\langle u, v\rangle_X := \frac{1}{NM}\sum_{1\leq i\leq N}\sum_{1\leq j\leq M} u_{i,j}v_{i,j},\ \|u\|_X := \sqrt{\frac{1}{NM}\sum_{1\leq i\leq N}\sum_{1\leq j\leq M} u_{i,j}^2}\ . \tag{2.6}$$

We set $Y = X \times X$. It is classical to define the discrete total variation with finite difference schemes as following (see for example [2]): the discrete gradient of the numerical image $u \in X$ is $\nabla u \in Y$ and may be computed by the following forward scheme for instance:

$$(\nabla u)_{i,j} = \left((\nabla u)_{i,j}^1, (\nabla u)_{i,j}^2\right), \tag{2.7}$$

where

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

Note that the constraint $\dfrac{\partial u}{\partial n} = 0$ is involved in the discretization process of the gradient. Therefore, in a discrete setting, the sets $\mathbf{K_2}$ and $\mathcal{K}_2$ coincide. The (discrete) total variation corresponding to $\Phi_1(u)$ is given by

$$J_1(u) = \frac{1}{NM} \sum_{1 \le i \le N} \sum_{1 \le j \le M} \left\| (\nabla u)_{i,j} \right\|_{\mathbb{R}^2}, \tag{2.8}$$

where $\left\| (\nabla u)_{i,j} \right\|_{\mathbb{R}^2} = \left\| \left(\nabla u^1_{i,j}, \nabla u^2_{i,j} \right) \right\|_{\mathbb{R}^2} = \sqrt{\left(\nabla u^1_{i,j} \right)^2 + \left(\nabla u^2_{i,j} \right)^2}$.
The discrete divergence operator -div is the adjoint operator of the gradient operator ∇:

$$\forall (p, u) \in Y \times X, \ \langle -\text{div } p, u \rangle_X = \langle p, \nabla u \rangle_Y .$$

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:

$$(Hv)_{i,j} = \left((Hv)^{11}_{i,j}, (Hv)^{12}_{i,j}, (Hv)^{21}_{i,j}, (Hv)^{22}_{i,j} \right).$$

We refer to [6, 5] for the detailed expressions of these quantities. The discrete second order total variation corresponding to $\Phi_2(v)$ writes

$$J_2(v) = \frac{1}{NM} \sum_{1 \le i \le N} \sum_{1 \le j \le M} \| (Hv)_{i,j} \|_{\mathbb{R}^4}, \tag{2.9}$$

with

$$\| (Hv)_{i,j} \|_{\mathbb{R}^4} = \sqrt{(Hv^{11}_{i,j})^2 + (Hv^{12}_{i,j})^2 + (Hv^{21}_{i,j})^2 + (Hv^{22}_{i,j})^2} \ .$$

The discretized problem stands

$$\inf_{(u,v) \in X \times X} F_{\lambda,\mu} := \frac{1}{2} \| u_d - u - v \|^2_X + \lambda J_1(u) + \mu J_2(v). \tag{2.10}$$

Problem (2.10) has obviously a solution $\tilde{u}$ and $\tilde{v}$ that satisfies the following necessary and sufficient optimality conditions

$$\tilde{u} = u_d - \tilde{v} - \mathit{\Pi}_{\lambda K_1} (u_d - \tilde{v}) \ , \tag{2.11a}$$

$$\tilde{v} = u_d - \tilde{u} - \mathit{\Pi}_{\mu K_2} (u_d - \tilde{u}) , \tag{2.11b}$$

where K_1 and K_2 are the following convex closed subsets :

$$K_1 = \{ \text{div } p \mid p \in X^2, \ \| p_{i,j} \|_{\mathbb{R}^2} \le 1 \ \forall i = 1, \dots, N, j = 1, \dots, M \}, \tag{2.12a}$$

$$K_2 = \{ H^* p \mid p \in X^4, \ \| p_{i,j} \|_{\mathbb{R}^4} \le 1, \ \forall i = 1, \dots, N, j = 1, \dots, M \}, \tag{2.12b}$$

and $\mathit{\Pi}_{K_i}$ denotes the orthogonal projection on K_i. These projections are computed with a Nesterov-type scheme as in [7]. We refer to [5] for more details. This leads to the following fixed-point algorithm :

Algorithme 1

Initialization step. Choose $u_0,\ v_0$, set $0 < \alpha < 1/2$ and $n = 1$.
Iteration. Define the sequences $((u_n, v_n))_n$ as

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

Stopping test. If $\max(\|u_{n+1} - u_n\|_{L^2}, \|v_{n+1} - v_n\|_{L^2}) \leq \varepsilon$ where $\varepsilon > 0$ is a prescribed tolerance, or if the iterations number is larger than a prescribed maximum number itmax, then STOP.

For any $\alpha \in (0, 1/2)$, the sequence generated by the algorithm converges to a stationary point, solution of (2.11) that we generically denote (u^*, v^*) in the sequel. The tolerance was set to $\varepsilon = 10^{-2}$ so that the stopping criterion is *de facto* the maximum number of iterations itmax. In the sequel, we have set itmax = 10 000 for the 1D case and itmax = 400 for the 2D case.
We do not report on CPU time since all tests have been done with MATLAB© and the code is not optimized. A parallelized C^{++} is version is written that reduces the computational time significantly.

2.2. **Examples.** We use 1D and 2D examples.
For the first (1D) example we set $s = s_0 + s_1 + s_2$ on [0,1] with s_0 a white gaussian noise with standard deviation $\sigma = 0.02$ and

$$s_1 = \begin{cases} 0.4 & \text{on } [\frac{3}{10}, \frac{6}{10}] \\ 0 & \text{elsewhere} \end{cases}, \qquad s_2(x) = \begin{cases} 0.8\,x + 0.2 & \text{on } [0, \frac{1}{2}] \\ -1.2\ (x-1) & \text{elsewhere.} \end{cases}$$

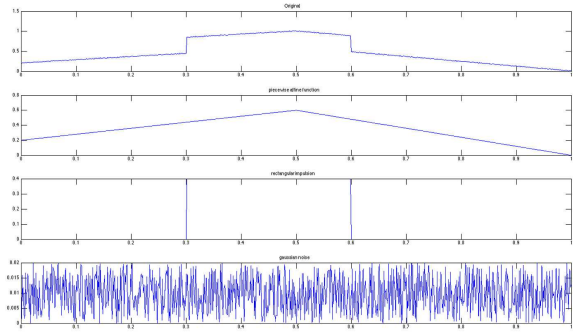
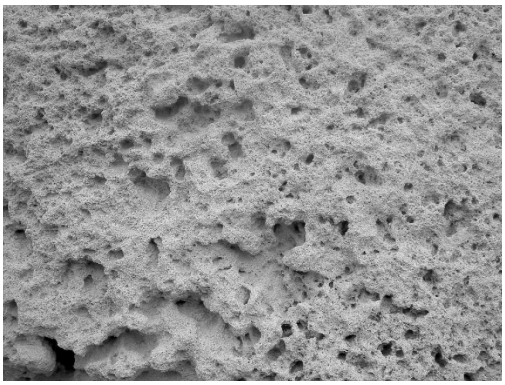


FIGURE 2.1. 1D example - 1000 points

The second example is a 2D picture of a *butterfly* and the third one an highly textured image (old *wall*). We used geometrical images as well but we do not report on them.

(a) Test 2D - Butterfly



(b) Test 2D - Wall

FIGURE 2.2. 2D examples

We present some results and comments in the next subsections[1].

2.3. **Initialization process.** We have tested many initialization choices for algorithm. Indeed, we have not proved uniqueness (though we conjecture it). So the computed solution is only a stationary point. As we may have many, we may think that the initialization process has a significant influence on the generated sequence. More precisely, we used

- $u_0 = 0,\ v_0 = u_d$, that we call initialization (a) in the sequel,
- $u_0 = u_d,\ v_0 = 0$ that we call initialization (a') in the sequel,
- $u_0 = 0,\ v_0 = 0$: initialization (b),
- randomized initializations around u_d mean value.

Initialization (a) (resp. (a')) provides a stationary pair (u^*, v^*) such that u^* (resp. v^*) has null mean value.

Proposition 2.1. *Assume $u_0 = 0$ and $v_0 = u_d$. Then any solution (u^*, v^*) given by the algorithm satisfies $\int_\Omega u^* = 0$. Similarly, if $u_0 = u_d$ and $v_0 = 0$, the pair (u^*, v^*) given by the algorithm satisfies $\int_\Omega v^* = 0$.*

Proof. Though we consider a discrete setting we use a continuous setting notation (using for example a piecewise affine approximation). We first note that

$$w \in K_1 \cup K_2 \Longrightarrow \int_\Omega w = 0\ .$$

We prove the first assertion. Assume that $u_0 = 0$ and $v_0 = u_d$. It is easy to see by induction that

$$\forall n \in \mathbb{N} \qquad \int_\Omega u_n = 0 \text{ and } \int_\Omega (v_n - u_d) = 0\ . \tag{2.13}$$

[1]Complete results (text files, movies, other examples) and MATLAB© code, are available at `http://maitinebergounioux.net/PagePro/Movies.html`

using

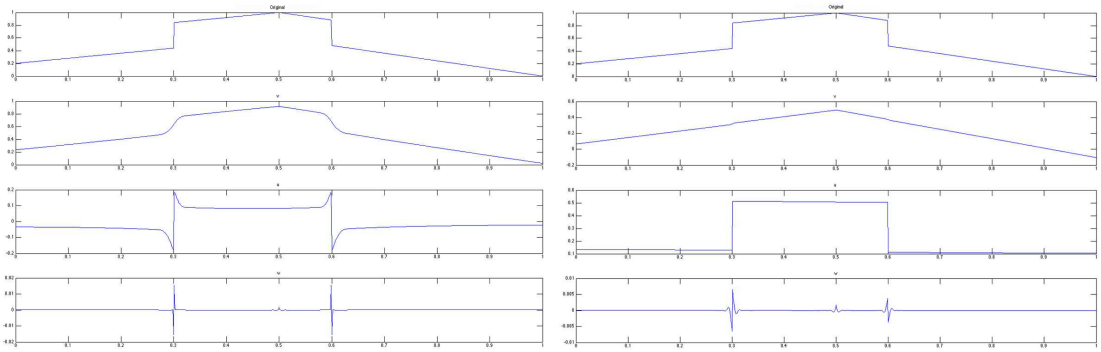
$$\begin{cases} u_{n+1} = u_n + \alpha \left(u_d - u_n - v_n - \Pi_{\lambda K_1} \left(u_d - v_n \right) \right) \\ v_{n+1} = v_n + \alpha \left(u_d - u_n - v_n - \Pi_{\mu K_2} \left(u_d - u_n \right) \right) . \end{cases}$$

Passing to the limit we get

$$\int_\Omega u^* = 0 \text{ and } \int_\Omega (v^* - u_d) = 0 \ .$$

The second assertion is proved similarly.

Proposition 2.1 yields that the BV- part u^* (or the BH- part v^*) belongs to the discrete Meyer space G (see [3]) if we perform the appropriate initialization step. This means it is an oscillating function. More precisely, choosing $u_0 = u_d,\ v_0 = 0$ gives a BH- part that belongs to G. This is not what we want, since the BH- part should not be oscillating. Therefore, we shall never use such an initialization.
Initializations (a) and (a') seem to give different results from initialization (b). We shall see in the sequel that the difference is *small* if the iteration number is large enough. Therefore, we think that the initial guess has no influence on the result, but only on the convergence speed.
We can see on Figure 2.3 (1D example) the oscillating effect of initialization $u_0 = 0,\ v_0 = u_d$:



(a) Initialization $u_0 = 0,\ v_0 = u_d$ (b) Initialization $u_0 = 0,\ v_0 = 0$

FIGURE 2.3. Example 1D without noise, $\lambda = 10^{-2},\ \mu = 5\ 10^{-2}$ and different initializations. Both u^* and w^* have null mean value for init (a). We recover the original decomposition with init (b).

Figure 2.4 and Table 1 gives the computed pairs with initializations (a) (a') (b) and a randomized initialization around the mean value of u_d.

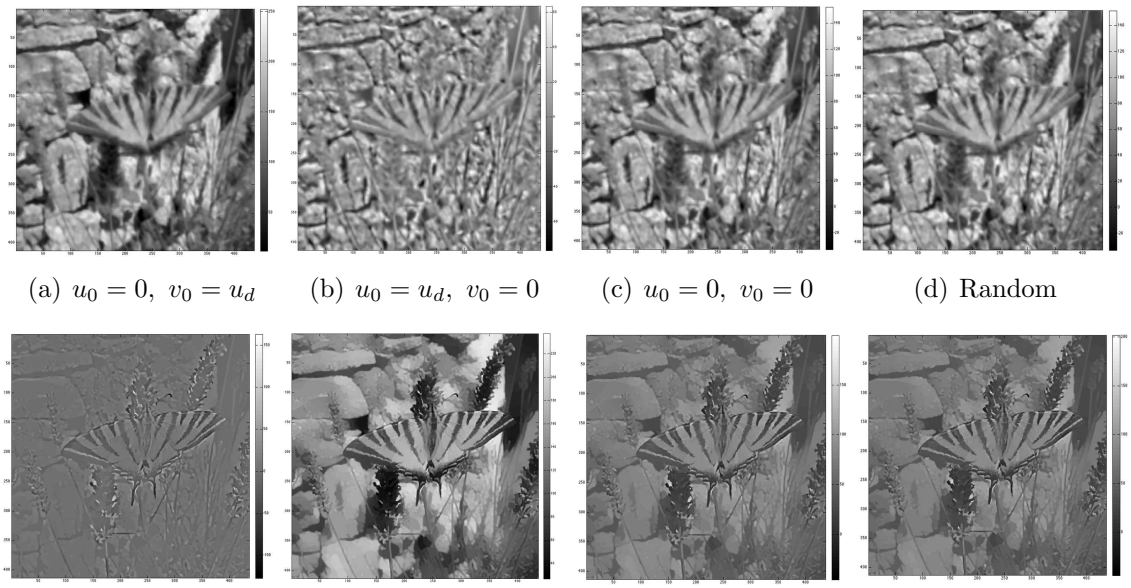
Initialization	$F_{\lambda,\mu}(u^*, v^*)$	$\|w^*\|_{L^2}$	$TV(u^*)$	$TV^2(v^*)$	Error	# it.
$\lambda = 1,\ \mu = 10$						
$u_0 = 0,\ v_0 = u_d$	23.68	1.04	12.70	1.04	2.35	400
$u_0 = u_d,\ v_0 = 0$	18.29	1	13.43	0.43	0.73	400
$u_0 = 0,\ v_0 = 0$	20.36	1.03	12.87	0.69	0.89	400
Random	20.39	1.03	12.88	0.69	0.87	400
$\lambda = 2,\ \mu = 0.1$						
$u_0 = 0,\ v_0 = u_d$	1.5414	2.24 e-01	3.64 e-04	15.15	8.48 e-03	22
$u_0 = u_d,\ v_0 = 0$	8.0239	2.76 e-01	3.31	13.52	4.35	400
$u_0 = 0,\ v_0 = 0$	3.3335	2.45 e-01	0.92	14.65	3.12	400
Random	3.5384	3.18 e-01	1.02	14.62	3.30	400
$\lambda = 5,\ \mu = 7$						
$u_0 = 0,\ v_0 = u_d$	61.7005	4.22	5.71	3.45	1.67	400
$u_0 = u_d,\ v_0 = 0$	62.7803	4.02	7.29	2.60	3.25	400
$u_0 = 0,\ v_0 = 0$	61.6248	4.15	6.34	3.04	1.50	400
Random	61.6331	4.15	6.35	3.04	1.55	400
$\lambda = 7,\ \mu = 7$						
$u_0 = 0,\ v_0 = u_d$	69.6775	5.23	2.29	5.69	9.76 e-01	400
$u_0 = u_d,\ v_0 = 0$	72.6262	4.96	4.09	4.51	4.74	400
$u_0 = 0,\ v_0 = 0$	70.2957	5.13	2.97	5.19	2.45	400
Random	70.3114	5.12	2.98	5.18	2.52	400
$\lambda = 7,\ \mu = 9$						
$u_0 = 0,\ v_0 = u_d$	79.7064	5.42	4.10	4.03	1.33	400
$u_0 = u_d,\ v_0 = 0$	80.1229	5.18	5.40	3.33	4.09	400
$u_0 = 0,\ v_0 = 0$	79.8224	5.33	4.58	3.72	1.86	400
Random	79.8297	5.33	4.58	3.72	1.89	400
$\lambda = 10,\ \mu = 15$						
$u_0 = 0,\ v_0 = u_d$	116.2130	7.04	3.59	3.69	1.39	400
$u_0 = u_d,\ v_0 = 0$	116.9598	6.79	4.33	3.36	4.67	400
$u_0 = 0,\ v_0 = 0$	116.0822	6.95	3.83	3.56	2.02	400
Random	116.0918	6.95	3.84	3.56	2.10	400

TABLE 1. Comparison of different initializations (Butterfly)- itmax=400 -The stationary pair is denoted (u^*, v^*) and $w^* = u_d - u^* - v^*$.

The blue (grey) lines of Table 1 show the *optimal* solution, that is the computed pair whose cost functional value is the lowest. We observed that

- the randomized initialization gives the same result as initialization (b),
- the component $w^* = u_d - u^* - v^*$ is always the same, which is consistent with the theoretical result of uniqueness,

- the values of the cost functional may be quite close and the computed pairs quite different: see for example $\lambda = 5, \mu = 7$ (and figure 2.4),
- initialization (b) gives a pair (u_b, v_b) such that neither u_b nor v_b has null mean value.



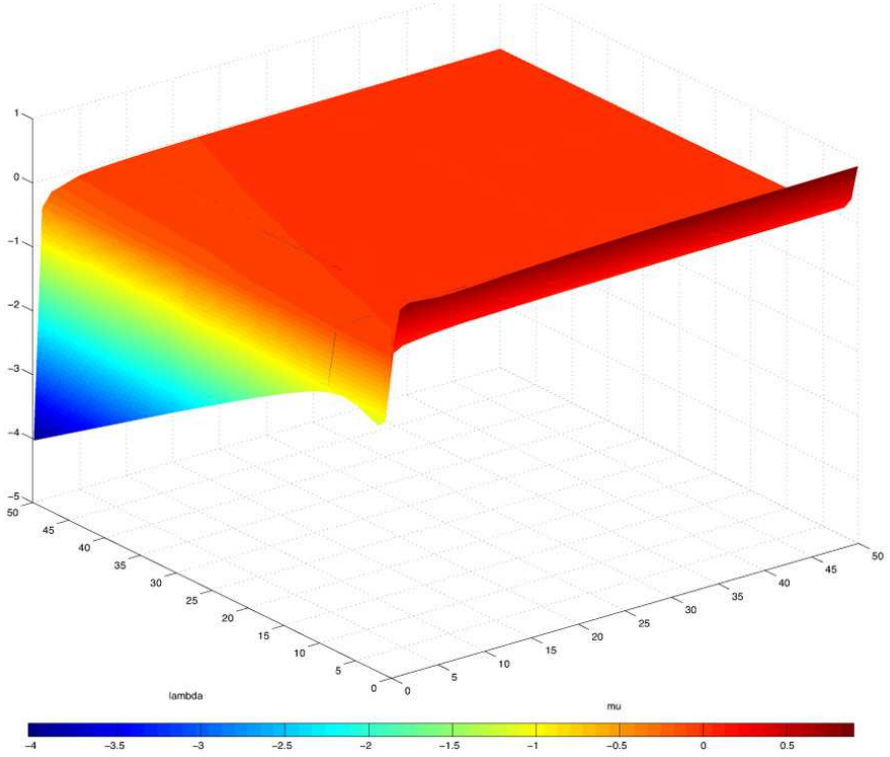
(a) $u_0 = 0,\ v_0 = u_d$ (b) $u_0 = u_d,\ v_0 = 0$ (c) $u_0 = 0,\ v_0 = 0$ (d) Random

(e) $u_0 = 0,\ v_0 = u_d$ (f) $u_0 = u_d,\ v_0 = 0$ (g) $u_0 = 0,\ v_0 = 0$ (h) Random

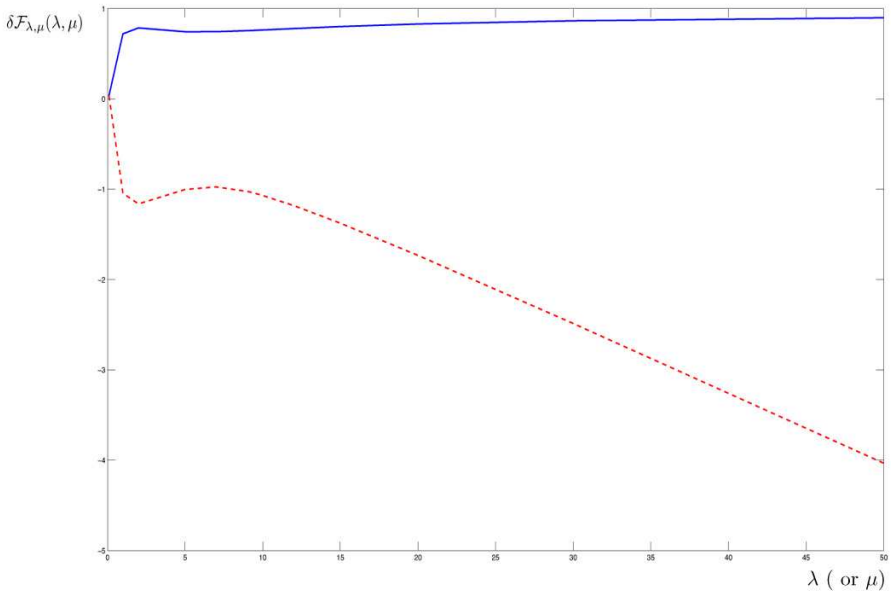
FIGURE 2.4. BH-part v (first line) and BV-part u (second line) given by initializations (a), (a'), (b) and random for $\lambda = 5,\ \mu = 7$ - *Butterfly* example with 400 iterations

In the sequel, (u_a, v_a) denotes the pair given by the algorithm with initialization (a) and (u_b, v_b) the one given by the algorithm with initialization (b). Moreover, we set the signed relative error as

$$\delta\mathcal{F}_{\lambda,\mu} = \frac{F_{\lambda,\mu}(u_a, v_a) - F_{\lambda,\mu}(u_b, v_b)}{\min(F_{\lambda,\mu}(u_a, v_a), F_{\lambda,\mu}(u_b, v_b))}. \tag{2.14}$$



(a) $\delta\mathcal{F}_{\lambda,\mu}(\lambda,\mu)$



(b) $\delta\mathcal{F}_{\lambda,\mu}(0.1,\mu)$ (red dotted line) - $\delta\mathcal{F}_{\lambda,\mu}(\lambda,0.1)$(blue solid line)

FIGURE 2.5. Behavior of $\delta\mathcal{F}_{\lambda,\mu}$ for 400 iterations (*Butterfly* example). If λ and μ are large enough ($\lambda > 0.1$ and $\mu > 0.1$ for example), both optimal values are very close.

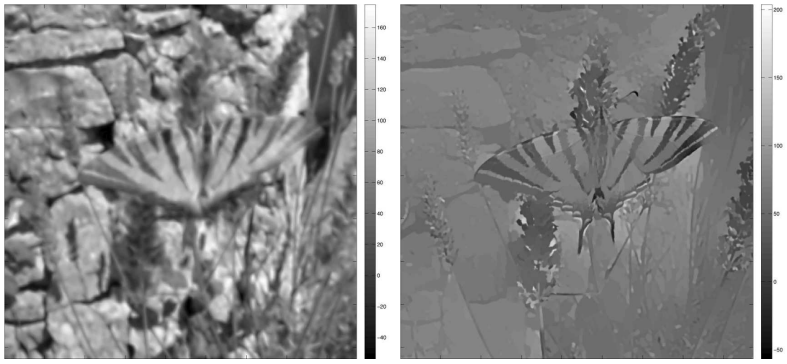
Figure 2.5 shows the behavior of $\delta\mathcal{F}_{\lambda,\mu}$ with respect to λ and μ.

# it.	$F_{\lambda,\mu}(u_a, v_a)$	$F_{\lambda,\mu}(u_b, v_b)$	$\lvert\delta\mathcal{F}_{\lambda,\mu}\rvert$
50	82.38439	81.69328	8 e-03
100	80.9555	80.9579	3 e-05
200	80.18443	80.35509	2 e-03
400	79.83481	79.94497	1.3 e-03
600	79.73564	79.80224	8 e-04
800	79.68948	79.73411	5.6 e-04
1000	79.66213	79.69571	4.2 e-04
1200	79.64396	79.67121	3.4 e-04
1500	79.62567	79.64659	2.6 e-04
5000	79.5738	79.5718	2.5 e-05

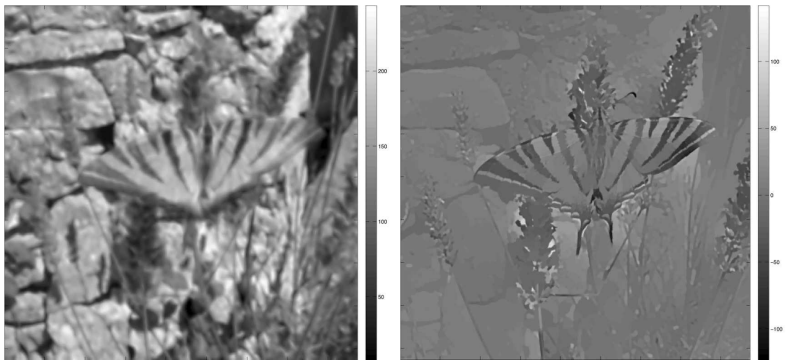
# it.	$TV(u_a)$	$TV(u_b)$	$TV(\varphi)$	$TV^2(v_a)$	$TV^2(v_b)$	$TV^2(\varphi)$	Error (a)	Error (b)
50	2.45	5.62	3.57	5.40	3.20	4.05	7.41	7.14
100	3.29	5.31	2.53	4.69	3.33	2.82	4.88	5.03
200	3.85	4.93	1.66	4.23	3.52	1.87	2.87	3.26
400	4.12	4.60	1.03	4.01	3.70	1.18	1.34	1.87
600	4.19	4.47	0.771	3.95	3.77	0.896	1.02	1.32
800	4.22	4.40	0.628	3.93	3.81	0.736	0.855	1.03
1000	4.23	4.37	0.536	3.92	3.83	0.632	0.735	0.845
1200	4.24	4.35	0.470	3.91	3.84	0.556	0.642	0.723
1500	4.24	4.33	0.396	3.90	3.86	0.472	0.535	0.595
5000	4.28	4.26	0.148	3.88	3.89	0.180	0.207	0.208

TABLE 2. Cost functional, TV and TV^2 for pairs given by initializations (a) and (b) and $\lambda = 7,\ \mu = 9$, as the number of iterations increases. Here $\varphi = u_b - u_a = v_b - v_a$ and the error is given by the stopping criterion of Algorithm.

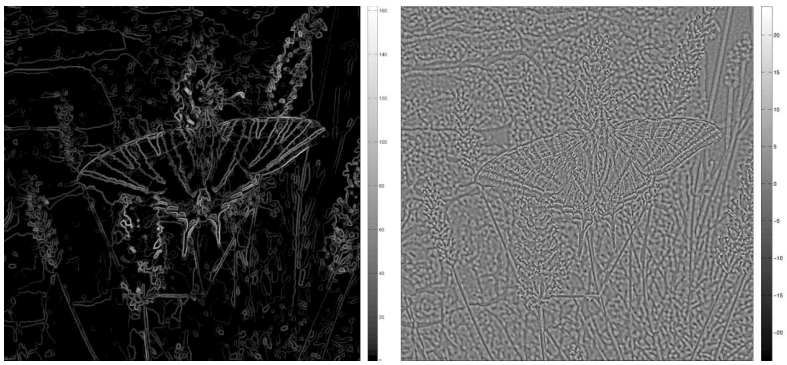
Though $F_{\lambda,\mu}(u_a, v_a) \simeq F_{\lambda,\mu}(u_b, v_b)$ the pairs (u_a, v_a) and (u_b, v_b) may be very different. More precisely, we have $u_b = u_a - \varphi$ and $v_b = v_a + \varphi$. Though the computed function φ_k at iteration k is not a constant function (see Figure 2.6), we infer that φ_k converges to a constant function as the iteration number increases. Indeed, we have numerically observed (see Table 2) that both $TV(\varphi)$ and $TV^2(\varphi)$ decreases to 0 as the iteration number increases. Nevertheless, we can perform only a limited number of iterations. So the computed solutions differ from a (small) piecewise constant function (see Figure 2.6). In addition, it is numerically confirmed that $w_a = u_d - u_a - v_a = w_b$ (what was theoretically proved).



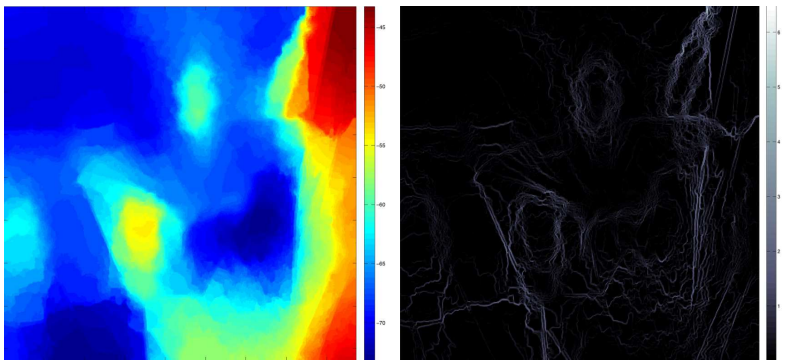
(a) v_a - init : $u_0 = 0,\ v_0 = u_d$ (b) u_a - init : $u_0 = 0,\ v_0 = u_d$



(c) v_b - init : $u_0 = 0,\ v_0 = 0$ (d) u_b - init : $u_0 = 0,\ v_0 = 0$



(e) Norm of u_b gradient (f) $w_b = w_a$



(g) $\varphi = u_b - u_a = v_a - v_b$ (h) Norm of φ gradient

FIGURE 2.6. Difference between the solutions given by initializations (a) and (b) for $\lambda = 7,\ \mu = 9$ - 5000 iterations . $\|\varphi\|_2 = 0.1518,\ TV(\varphi) = 0.1484,\ TV^2(\varphi) = 0.1803$. The function φ seems to be piecewise constant as we see it on the gradient norm.

2.4. **Convergence.** We chose $\alpha = 0.25$ in the fixed point algorithm and we always observed convergence. We set the maximal number of iterations quite large but we noticed that the solution is satisfactory with less iterations (400 for 2D case and 1000 for 1D case).

# it.	$F_{\lambda,\mu}(u_a, v_a)$	$F_{\lambda,\mu}(u_b, v_b)$	$\lvert\delta\mathcal{F}_{\lambda,\mu}\rvert$
$\lambda = 1,\ \mu = 10$			
50	39.132	25.513	5 e-01
100	31.727	23.113	3.7 e-01
200	26.907	21.440	2.5 e-01
400	23.711	20.377	1.6 e-01
600	22.410	19.978	1.2 e-01
800	21.688	19.774	9.6 e-02
$\lambda = 10,\ \mu = 15$			
50	119.448	117.102	2 e-02
100	117.578	116.601	8.3 e-03
200	116.612	116.257	3 e-03
400	116.215	116.083	1.1 e-03
600	116.106	116.031	6.5 e-04
800	116.052	116.006	4 e-04
$\lambda = 10,\ \mu = 2$			
50	25.90989	39.586	5.2 e-01
100	25.91003	33.501	2.9 e-01
200	25.91008	29.512	1.4 e-01
400	25.91009	27.558	6.3 e-02
600	25.91009	26.986	4.1 e-02
800	25.91009	26.699	3 e-02

# it.	$TV(u_a)$	$TV(u_b)$	$TV(\varphi)$	$TV^2(v_a)$	$TV^2(v_b)$	$TV^2(\varphi)$	Error (a)	Error (b)
$\lambda = 1,\ \mu = 10$								
50	10.73	12.12	3.98	2.74	1.27	2.19	14.58	6.51
100	11.81	12.51	3.13	1.92	1	1.50	8.30	3.33
200	12.39	12.74	2.47	1.39	0.81	1.09	4.39	1.72
400	12.70	12.87	1.96	1.04	0.69	0.82	2.37	0. 90
600	12.80	12.92	1.70	0.9	0.65	0.70	1.53	0.58
800	12.85	12.94	1.53	0.83	0.63	0.63	1.09	0.42
$\lambda = 10,\ \mu = 15$								
50	2.59	4.83	2.74	4.42	3.09	3.14	9.04	9.33
100	3.19	4.44	1.85	3.98	3.27	2.14	5.29	6.17
200	3.48	4.07	1.18	3.77	3.45	1.39	2.49	3.77
400	3.59	3.83	0.73	3.69	3.56	0.87	1.39	2.03
600	3.61	3.76	0.55	3.67	3.60	0.67	1.08	1.40
800	3.62	3.72	0.45	3.67	3.62	0.54	0.87	1.07
$\lambda = 10,\ \mu = 2$								
50	8.937 e-03	1.619	1.619	11.214	9.88	1.80	1.15 e-02	18.18
100	8.951 e-03	8.948 e-01	8.94 e-01	11.214	10.566	9.62 e-01	4.72 e-03	12.92
200	8.956 e-03	4.116 e-01	4.11 e-01	11.214	11.010	3.35 e-01	1.93 e-03	8.92
400	8.957 e-03	1.804 e-01	1.80 e-01	11.214	11.183	6.05 e-02	9.30 e-04	4.48
600	8.957 e-03	1.180 e-01	1.17 e-01	11.214	11.207	1.54 e-02	6.23 e-04	2.54
800	8.957 e-03	8.841 e-02	8.77 e-02	11.214	11.212	6.4 e-03	4.42 e-04	1.77

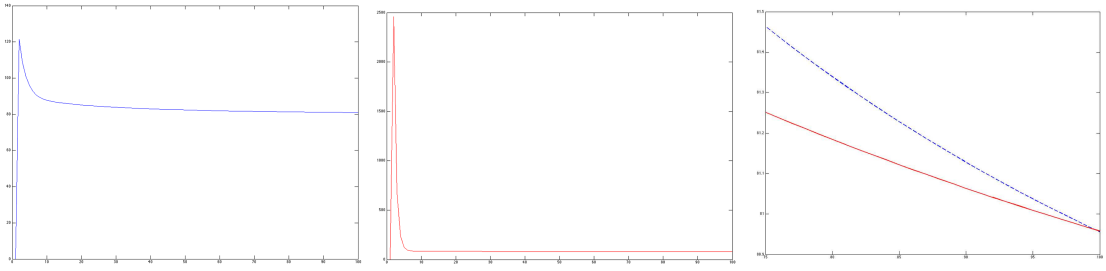
TABLE 3. Sensitivity with respect to number of iterations. Here $\varphi = u_b - u_a = v_b - v_a$ and the error is given by the stopping criterion of Algorithm. On can refer to Table 2 as well.

(a) BV part - $\lambda = 1,\ \mu = 10$ (b) BV part - $\lambda = 10,\ \mu = 15$ (c) BV part -$\lambda = 10,\ \mu = 2$. In this case $u \simeq 0$

(d) φ - $\lambda = 1,\ \mu = 10$ (e) φ - $\lambda = 10,\ \mu = 15$ (f) φ -$\lambda = 10,\ \mu = 2$

FIGURE 2.7. BV component u_a and φcorresponding to Table 3 - 800 iterations

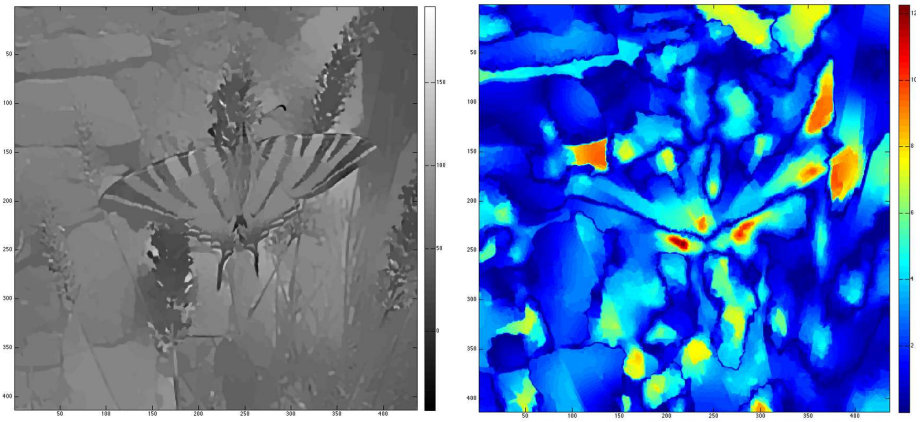
Figure 2.8 illustrates the generic behavior of the cost-functional $\mathcal{F}_{\lambda,\mu}$.[2]



(a) Init (a) : $u_0 = 0,\ v_0 = u_d$ (b) Init (b) : $u_0 = 0,\ v_0 = 0$ (c) Zoom 75-100 iterations

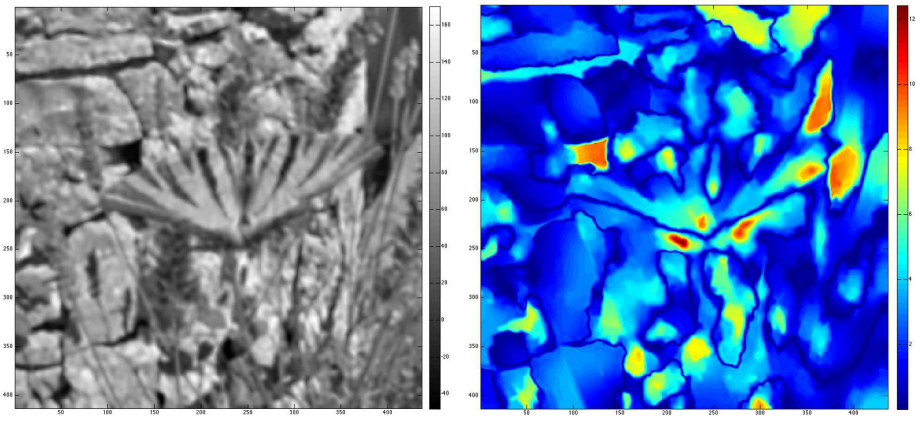
FIGURE 2.8. Behavior of the cost functional for $\lambda = 7,\ \mu = 9$, 100 iterations- Dotted (blue) line is initialization (a) and solid (red) line is initialization (b)

[2]One can look at `http://maitinebergounioux.net/PagePro/Movies.html` to see the convergence process.



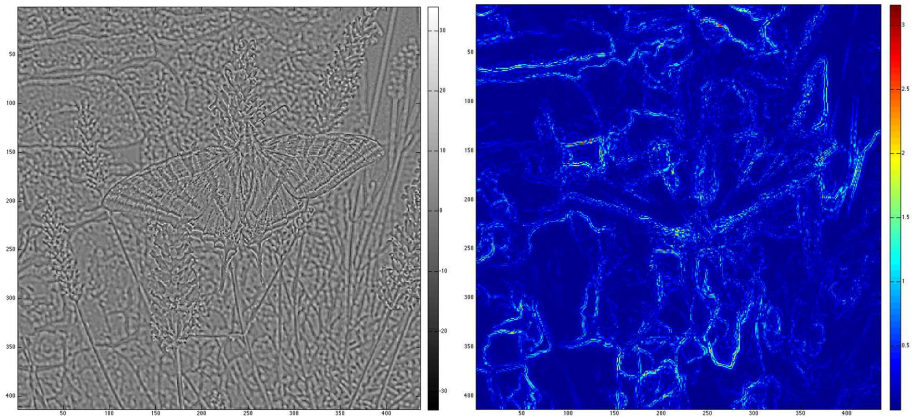
(a) BV part u - 800 iterations

(b) Absolute difference of BV parts between it 400 and it 800



(c) BH part v - 800 iterations

(d) Absolute difference of BH parts between it 400 and it 800



(e) L^2- part w - 800 iterations

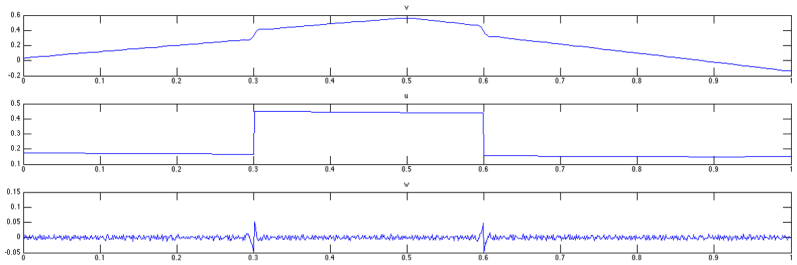
(f) Absolute difference of L^2- parts between it 400 and it 800

FIGURE 2.9. Test 2D - Initialization (b) for $\lambda = 10$, $\mu = 15$- Difference between the computed pairs at iteration 400 and iteration 800.

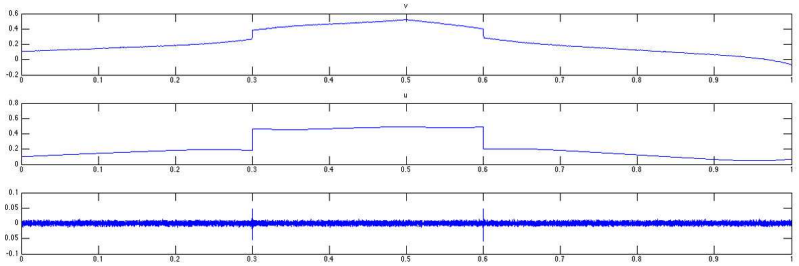
2.5. **Sensitivity with respect to sampling and quantification.** Table 4 and Figure 2.10 show that the model is robust with respect to sampling. Here, we have discretized the analogical signal of example 1D with 10^3, 10^4 and 10^5 points respectively.

λ	μ	$\mathcal{F}_{\lambda,\mu}(u_b, v_b)$	$\|w_b\|_{L^2}$	$TV(u_b)$	$TV^2(v_b)$
		10^3 points 10^4 points 10^5 points	10^3 points 10^4 points 10^5 points	10^3 points 10^4 points 10^5 points	10^3 points 10^4 points 10^5 points
1e -03	1 e-02	6.47 e-06 5.78 e-06 5.73 e-06	1.43 e-03 1.47e-03 1.5 e-03	5.01 e-03 4.37 e-03 4.34 e-03	4.28 e-05 3.19 e-05 3.09 e-05
1e-03	1	4.86 e-05 3.73 e-05 3.63 e-05	1.43 e-03 1.47 e-03 1.47 e-03	5.01 e-03 4.37 e-03 4.34 e-03	4.25 e-05 3.18 e-05 3.09 e-05
1e -02	1e-01	3.27 e-05 2.62 e-05 2.52 e-05	5.07 e-03 5.16 e-03 5.24 e-03	8.80 e-04 1.81 e-04 7.26 e-05	1.10 e-04 1.10 e-04 1.08 e-04
1e -02	1	1.32 e-04 1.26 e-04 1.22 e-04	5.07 e-03 5.16 e-03 5.24 e-03	8.80 e-04 1.81 e-04 7.26 e-05	1.10 e-04 1.10 e-04 1.08 e-04
1e-01	1e-01	1.01 e-04 3.71 e-05 2.73 e-05	6.90 e-03 5.53 e-03 5.38 e-03	5.43 e-04 8.74 e-05 1.20 e-05	2.32 e-04 1.30 e-04 1.16 e-04

TABLE 4. Test 1D (with noise) - sensitivity with respect to sampling - Initialization (b) ($u_0 = v_0 = 0$) and 10 000 iterations



(a) 10^3 points



(b) 10^4 points

FIGURE 2.10. Test 1D (with noise) - Pair given by initialization (b) for $\lambda = 0.1,\ \mu = 1$, 10 000 iterations and different samplings.

We now investigate the sensitivity of the model with respect to quantification. Let u_d a data (with values in [0, 255] for example). Let (λ, μ) be chosen parameters and $(u_{\lambda,\mu}, v_{\lambda,\mu})$ the corresponding computed pair (with the appropriate initialization). Let $\alpha > 0$ and consider the new data αu_d. This is the case, for example, if we get 16 bits images and convert them to 8 bits : in this case $\alpha = (2^8-1)/(2^{16}-1)$. We may want to normalize the data as well: in this case $\alpha = 1/\max(u_d)$. The question is to know what new parameters $(\tilde{\lambda}, \tilde{\mu})$ must be chosen to get $u_{\tilde{\lambda},\tilde{\mu}} = \alpha u_{\lambda,\mu}$ and $v_{\tilde{\lambda},\tilde{\mu}} = \alpha v_{\lambda,\mu}$. For any $(u_{\lambda,\mu}, v_{\lambda,\mu})$ solution to $(\mathcal{P}_{\lambda,\mu})$, we get

$$\begin{aligned} F_{\lambda,\mu}(u_{\lambda,\mu}, v_{\lambda,\mu}) &= \frac{1}{2}\|u_d - u_{\lambda,\mu} - v_{\lambda,\mu}\|^2 + \lambda TV(u_{\lambda,\mu}) + \mu TV^2(v_{\lambda,\mu}) \\ &= \frac{1}{2\alpha^2}\|\alpha u_d - \alpha u_{\lambda,\mu} - \alpha v_{\lambda,\mu}\|^2 + \frac{\lambda}{\alpha}TV(\alpha u_{\lambda,\mu}) + \frac{\mu}{\alpha}TV^2(\alpha v_{\lambda,\mu}) \\ &= \frac{1}{\alpha^2}\left(\frac{1}{2}\|\alpha u_d - u_{\tilde{\lambda},\tilde{\mu}} - v_{\tilde{\lambda},\tilde{\mu}}\|^2 + \alpha\lambda TV(u_{\tilde{\lambda},\tilde{\mu}}) + \alpha\mu TV^2(v_{\tilde{\lambda},\tilde{\mu}})\right) \\ &= \frac{1}{\alpha^2}F_{\tilde{\lambda},\tilde{\mu}}(u_{\tilde{\lambda},\tilde{\mu}}, v_{\tilde{\lambda},\tilde{\mu}}) \text{ with} \end{aligned}$$

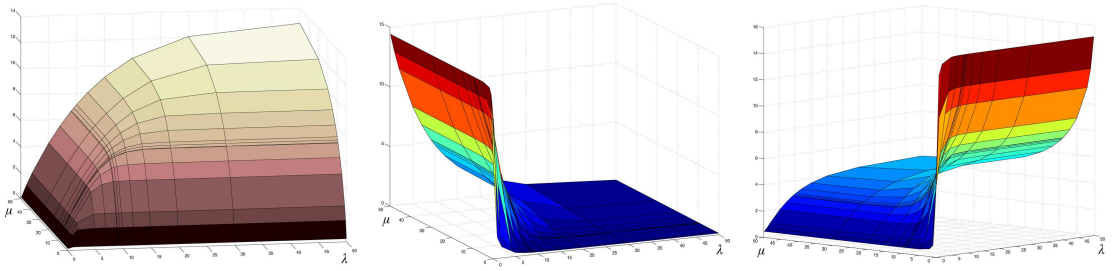
$$\tilde{\lambda} = \alpha\lambda \text{ and } \tilde{\mu} = \alpha\mu.$$

α	1/255	100
$\lambda = 7,\ \mu = 9$ Initialization (a)		
$\|u_{\alpha\lambda,\alpha\mu} - \alpha u_{\lambda,\mu}\|_\infty/\alpha$	3.0291e-01	3.0291e-01
$\|v_{\alpha\lambda,\alpha\mu} - \alpha v_{\lambda,\mu}\|_\infty/\alpha$	3.1291e-01	3.1291e-01
$\lambda = 7,\ \mu = 9$ Initialization (b)		
$\|u_{\alpha\lambda,\alpha\mu} - \alpha u_{\lambda,\mu}\|_\infty/\alpha$	1.8006e-01	1.8006e-01
$\|v_{\alpha\lambda,\alpha\mu} - \alpha v_{\lambda,\mu}\|_\infty/\alpha$	1.7924e-01	1.7924e-01
$\lambda = 10,\ \mu = 2$ Initialization (a)		
$\|u_{\alpha\lambda,\alpha\mu} - \alpha u_{\lambda,\mu}\|_\infty/\alpha$	8.0280e-15	8.4421e-15
$\|v_{\alpha\lambda,\alpha\mu} - \alpha v_{\lambda,\mu}\|_\infty/\alpha$	1.1324e-13	1.4552e-13
$\lambda = 10,\ \mu = 2$ Initialization (b)		
$\|u_{\alpha\lambda,\alpha\mu} - \alpha u_{\lambda,\mu}\|_\infty/\alpha$	3.9216e-03	4.5475e-14
$\|v_{\alpha\lambda,\alpha\mu} - \alpha v_{\lambda,\mu}\|_\infty/\alpha$	1.1324e-13	1.0914e-13

TABLE 5. Sensitivity with respect to quantification- Initialization (b) - itmax = 400

2.6. **Sensitivity with respect to parameters.** As mentioned before the computed stationary pair depends on the initialization guess via the convergence speed. We consider three cases and we illustrate them on test 2D (*Butterfly*).

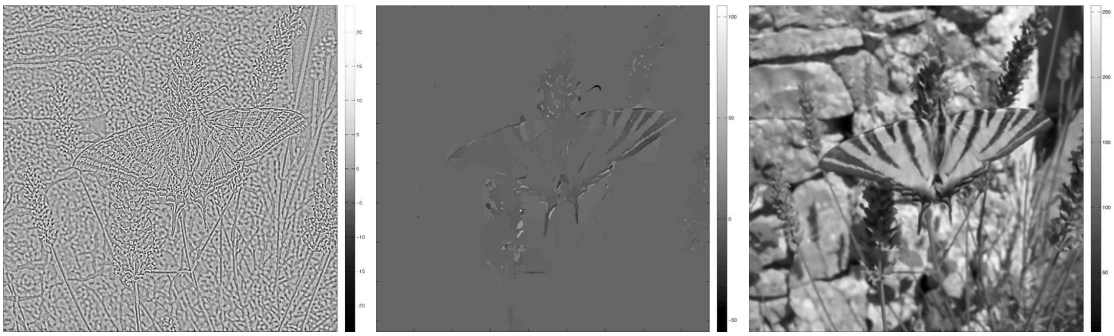
- If $\mu << \lambda$, then initialization (a) : $u_0 = 0$ and $v_0 = u_d$ is the best choice to make the algorithm converge quickly. So we use this initialization to get the *solution* (u^*, v^*). In this case, the BV part is close to 0. However, we note that if we fix μ then $TV(u^*(\lambda, \mu))$ decreases to 0 and $TV^2(v^*(\lambda, \mu))$ increases to become constant (see Figure 2.11) when $\lambda \to +\infty$. This means that if λ is large then u^* is constant. As we know that u^* has a null mean value, then $u^* = 0$.



(a) $\|w^*\|_{L^2}$ (b) $TV(u^*)$ (c) $TV^2(v^*)$

FIGURE 2.11. Generic L^2- norm, TV and TV^2 behavior (μ fixed) 400 iterations - Example 2D (Butterfly).

On can see an example on Figure 2.7 for $\lambda = 10,\ \mu = 2$ and Figure 2.12.



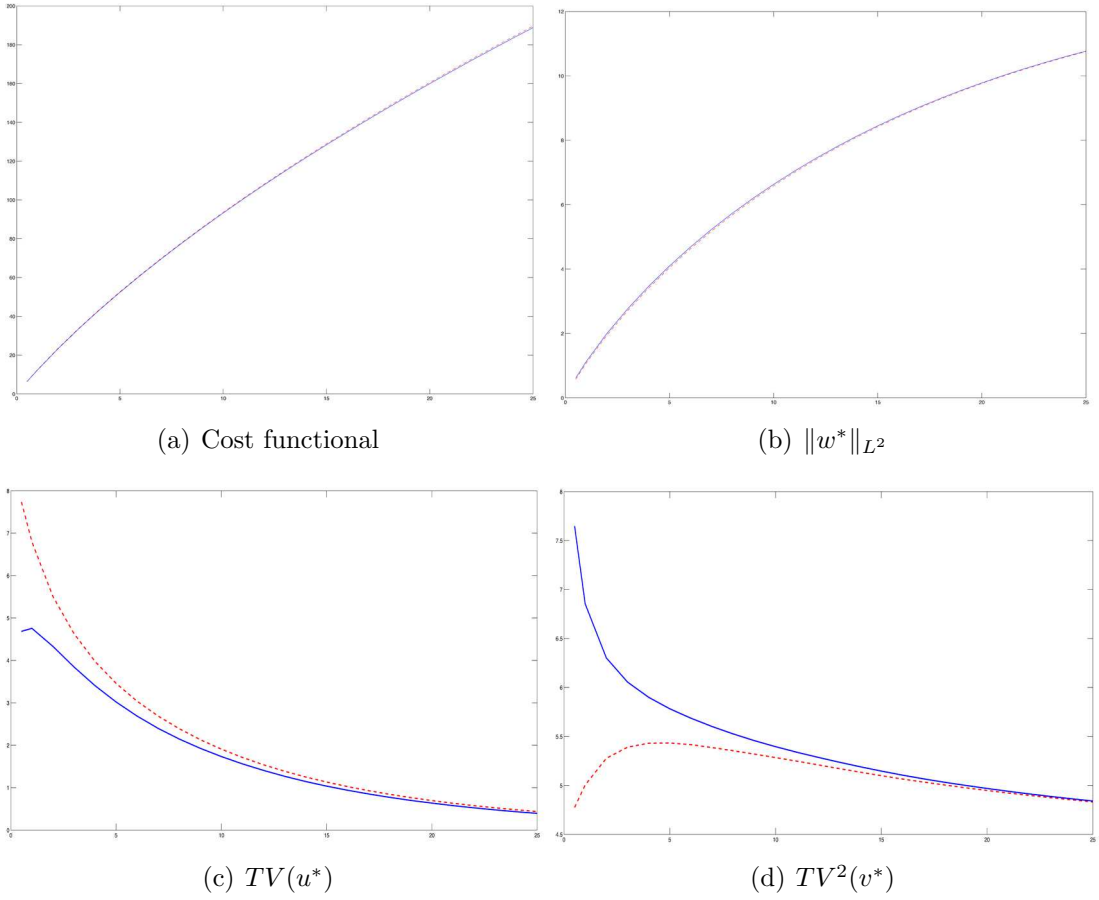
(a) L^2 part (b) BV part (c) BH part

FIGURE 2.12. $\lambda = 7,\ \mu = 5$ - initialization $u_0 = 0$ and $v_0 = u_d$, 400 iterations

- If $\mu \simeq \lambda$, both initializations seem equivalent. For the *Butterfly* test, init (a) remains slightly faster (in this case the minimum value of cost functional is achieved first) while it is the converse for the *Wall* test and small values of λ. Figures 2.13 and 2.14 show the behavior of the cost functional, L^2-norm, TV and TV^2 for both initializations and $\lambda = \mu \in [0.5, 1, 2, 3 \cdots 25]$. We report the behavior of cost functional, L^2- norm, TV and TV^2 in Table 6

$\lambda = \mu$	$\mathcal{F}_{\lambda,\lambda}$	$\|w_a\|_2$	$TV(u_a)$	$TV2(v_a)$	Error
0.5	6.3577	1.459 e-03	4.685	7.646	4.13 e-01
1	12.2448	2.655 e-03	4.756	6.853	5.19 e-01
5	52.4043	9.654 e-03	3.020	5.782	5.83 e-01
10	93.2718	1.562 e-02	1.733	5.395	5.59 e-01
13	114.9198	1.835 e-02	1.268	5.237	5.11 e-01
17	141.3794	2.128 e-02	8.506 e-01	5.066	4.78 e-01
21	165.9126	2.357 e-02	5.768 e-01	4.940	4.36 e-01
25	188.8569	2.537 e-02	3.941 e-01	4.840	3.86 e-01

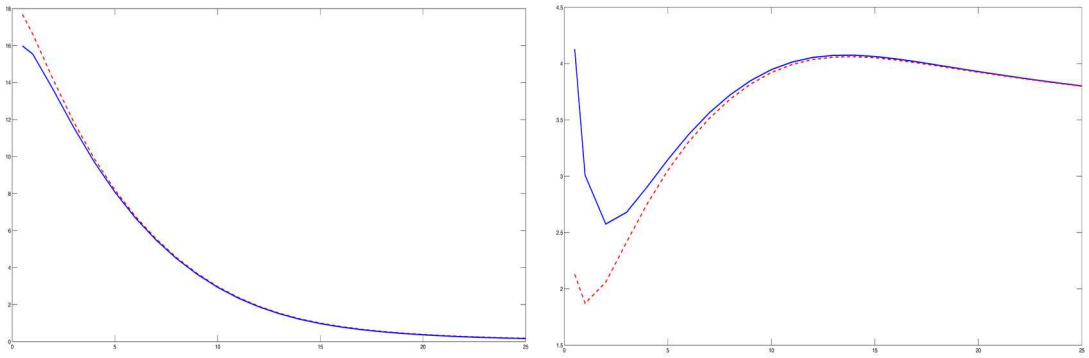
TABLE 6. Cost functional, L^2- norm, TV and TV^2 for $\lambda = \mu = 0.5, 1, 2, 3 \cdots 25$- init (a) - 800 iterations - *Butterfly*



(a) Cost functional (b) $\|w^*\|_{L^2}$

(c) $TV(u^*)$ (d) $TV^2(v^*)$

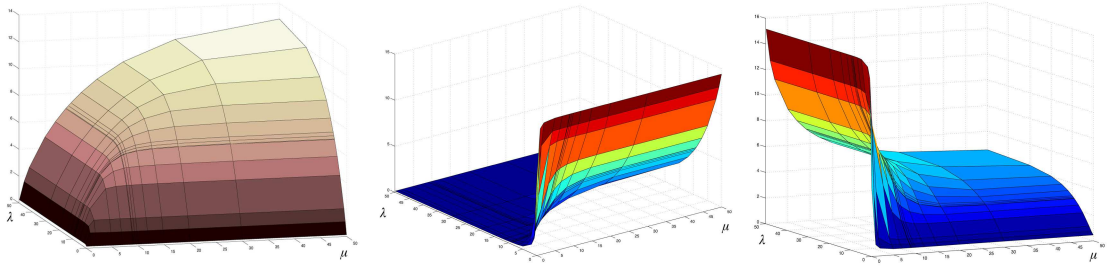
FIGURE 2.13. Cost functional, L^2- norm, TV and TV^2 for $\lambda = \mu = 0.5, 1, 2, 3 \cdots 25$ - Dotted (blue) line is initialization (a) and solid (red) line is initialization (b) - 800 iterations - *Butterfly* test



(a) $TV(u^*)$ (b) $TV^2(v^*)$

FIGURE 2.14. TV and TV^2 for $\lambda = \mu = 0.5, 1, 2, 3 \cdots 25$ - Dotted (blue) line is initialization (a) and solid (red) line is initialization (b) - 800 iterations - *Wall* test

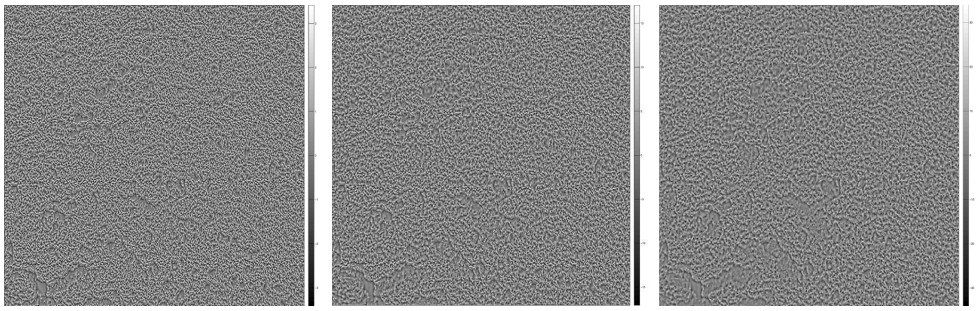
- If $\lambda << \mu$, then we choose initialization (b) : $u_0 = 0$ and $v_0 = 0$ to get the *solution*. The behavior is similar to the case $\mu < \lambda$: if we fix λ, then $TV(u^*(\lambda, \mu))$ increases to a constant value and $TV^2(v^*(\lambda, \mu))$ converges to 0 as $\mu \to +\infty$ (see figure 2.15). This means that if μ is large enough then solution is always the same : v^* is an affine fonction.



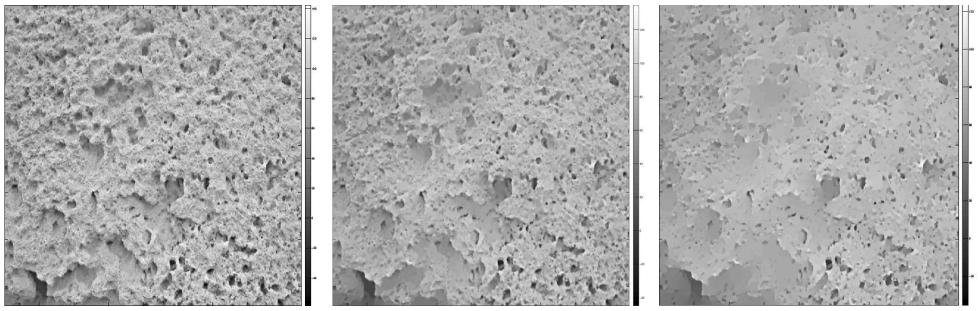
(a) $\|w^*\|_{L^2}$ (b) $TV(u^*)$ (c) $TV^2(v^*)$

FIGURE 2.15. Generic L^2- norm, TV and TV^2 behavior - (λ fixed) 400 iterations - Example 2D (Butterfly).

Examples of solutions are given in Figures 2.4, 2.7 and 2.9. We give another example below on a textured image:



(a) L^2- part $\lambda = 1,\ \mu = 5$ (b) L^2- part $\lambda = 5,\ \mu = 10$ (c) L^2- part $\lambda = 10,\ \mu = 20$



(d) BV part $\lambda = 1,\ \mu = 5$ (e) BV part $\lambda = 5,\ \mu = 10$ (f) BV part $\lambda = 10,\ \mu = 20$

FIGURE 2.16. BV and L^2 components with $\lambda < \mu$ - 800 iterations - Wall example

λ	μ	$\mathcal{F}_{\lambda,\mu}(u^*, v^*)$	$\|w^*\|_{L^2}$	$TV(u^*)$	$TV^2(v^*)$
1	5	21.6334	4.270 e-03	18.453	3.868 e-01
5	10	87.2937	1.759 e-02	10.402	1.413
10	20	152.5461	2.854 e-02	4.922	2.382

TABLE 7. Wall- example, initialization $u_0 = 0,\ v_0 = 0$ - 800 iterations

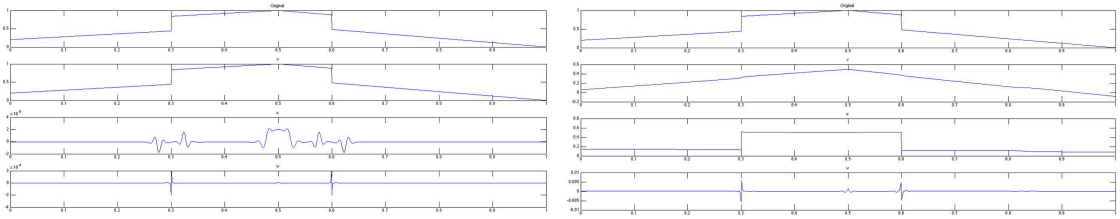
3. Conclusion

The model is well adapted to texture extraction. In the case, where the data is noiseless and/or is not too much textured, the decomposition given par $\lambda \precsim \mu$ and initialization $u_0 = v_0 = 0$, gives a cartoon part which is piecewise constant as expected. This means that $u = \sum_i u_i \mathbf{1}_{\Gamma_i}$ where $\bigcup_i \Gamma_i$ is the contour set. In this case, the remainder L^2 term is the texture and/or noise. The decomposition is robust with respect to quantification, sampling and is always the same for any $\mu >> \lambda$, once λ has been chosen.

In the case where the image is highly textured the model provides a two-scale decomposition. The TV part represents the macro-texture and the L^2 part the micro-texture and/or noise. The scaling is tuned via the ratio $\rho = \dfrac{\lambda}{\mu}$.

The notion of *highly textured* may be quantified par the G-norm. In our 2D examples, the *butterfly* G norm was $\simeq 7.71$ and the *wall* one was $\simeq 4.92$.

Figure 3.17 shows the behavior of the different components with respect to λ and μ. We have chosen the 1D noiseless case, to see the multi-scale effect on components u and w when $\mu < \lambda$ (init (a)).



(a) $\lambda = 5\ 10^{-3},\ \mu = 10^{-3}$- (init (a)) (b) $\lambda = 5\ 10^{-3},\ \mu = 10^{-2}$- (init (b))

FIGURE 3.17. Test 1D without noise (1000 points)

Moreover, the initialization process has no influence on the solution (up to a constant function) but rather on the algorithm speed. The choice has to be made with respect to the parameters: roughly speaking, if $\lambda < \mu$ we choose $u_0 = 0,\ v_0 = 0$ and if $\lambda \geq \mu$ we choose $u_0 = 0,\ v_0 = u_d$. Finally, we have observed (numerically) that the L^2-component w is unique.

Our next issue is to speed up the algorithm and set an automatic parameter tuning with respect to data properties (G norm, Signal to Noise Ratio , and so on.) From the theoretical point of view, we infer that problem $(\mathcal{P}_{\lambda,\mu})$ has a unique solution (up to constant functions) but the question is still open.

REFERENCES

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

[2] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing, Partial Differential Equations and the Calculus of Variations*, volume 147 of *Applied Mathematical Sciences*. Springer Verlag, 2006.

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

[4] M. Bergounioux. Mathematical analysis of a inf-convolution model for image processing. Technical report, hal.archives-ouvertes.fr, 2014. `https://hal.archives-ouvertes.fr/hal-01002958v2`

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

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

[7] P. Weiss, G. Aubert, and L. Blanc-Féraud. Efficient schemes for total variation minimization under constraints in image processing. *SIAM Journal on Scientific Computing*, 31(3):2047–2080, 2009.

E-mail address: maitine.bergounioux@univ-orleans.fr