A Variational Model for Image Texture Identification

R. Echegut and L. Piffet

Université d'Orléans, Département de Mathématiques (MAPMO), UFR Sciences,
Bâtiment de mathématiques, Route de Chartres B.P. 6759, 45067 Orléans cedex 2,
FRANCE
echegut.romain@wanadoo.fr, Loic.Piffet@univ-orleans.fr

Summary. A second order variational model is tested to extract texture from an image. An existence result is given. A fixed point algorithm is proposed to solve the discretized problem. Some numerical experiments are done for two images.

Variational models in image processing have been extensively studied during the past decade. They are used for segmentation processes (geodesic or geometric contours), restoration and textures extraction purpose as well. Roughly speaking image restoration problems are severely ill posed and a Tikhonov-like regularization 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. Here, we are interested in textures extraction and/or image denoising. Recent works were based on the assumption that an image can be decomposed in many components, each component describing a particular property of the image (see [6, 12, 14] for example). We follow this idea and assume that the image f we want to recover from the data u_d can be decomposed as $f = u + v$ where u and v are functions that belong to different functional spaces: u is the "texture" part which involves (periodic or not) details (and noise as well) while v is a more regular part (usually called the "cartoon" component).

In a first section, we present the functional framework, introducing the BV^2 space, and the general variational model we consider. In section 2, we focus on numerical implementation and present the discretization process. Numerical tests are reported in the last section.

M. Diehl et al. (eds.), *Recent Advances in Optimization and its Applications in Engineering*,
DOI 10.1007/978-3-642-12598-0_41, © Springer-Verlag Berlin Heidelberg 2010

1 Functional framework and model

1.1 The $BV^2(\Omega)$ space

Let Ω be an open bounded subset of $\mathbb{R}^n$, $n \geq 2$ (practically $n = 2$). Following Demengel [9], we define the space of Hessian bounded functions that we call $BV^2(\Omega)$. We recall that the space $BV(\Omega)$ of bounded variation functions (see [2, 4, 3]) is defined as

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

where

$$\Phi(u) = \sup\left\{\int_\Omega u(x) \operatorname{div} \xi(x)\, dx \mid \xi \in \mathcal{C}_c^1(\Omega),\ \|\xi\|_\infty \leq 1\right\}. \tag{1}$$

The space $BV(\Omega)$, endowed with the norm $\|u\|_{BV(\Omega)} = \|u\|_{L^1} + \Phi(u)$, 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 $\Phi(u) = \int_\Omega |Du|$ is the total variation of Du. We extend this definition to the second derivative (in the distributional sense). Recall that the Sobolev space

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

where ∇u stands for the first order derivative of u (in the sense of distributions).

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

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

where

$$div(\phi) = (div(\phi_1), div(\phi_2), \ldots, div(\phi_n)),$$

with

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

For more information on the $BV^2(\Omega)$ space, see [9, 13].

1.2 The variational model

We now assume that the image we want to recover from the data u_d can be written as $f = u + v$ where u is in $BV(\Omega)$ and v is in $BV^2(\Omega)$. Such decompositions have already been performed [5, 6, 4] using the “Meyer” space of oscillating function [10] instead of $BV^2(\Omega)$. So far, the model we propose is not the same: the oscillating component will be included in the non regular part u while v involves the cartoon and the contours. We consider the following function defined on $BV(\Omega) \times BV^2(\Omega)$:

$$F(u,v) = \frac{1}{2}\|u_d - u - v\|^2_{L^2(\Omega)} + \lambda |u|_{BV(\Omega)} + \mu |v|_{BV^2(\Omega)} + \delta \|\nabla v\|_{W^{1,1}(\Omega)}, \tag{2}$$

where $\lambda, \mu, \delta \geq 0$ are weigths. We are looking for a solution to the optimisation problem

$$\inf_{(u,v)\in BV(\Omega)\times BV^2(\Omega)} F(u,v) \tag{$\mathcal{P}$}$$

The first term $\|u_d - u - v\|^2_{L^2(\Omega)}$ of F is the fitting data term. Other terms are Tychonov-like regularization terms. Note that the δ-term is not useful from the modelling point of view. It is only a tool that allows to prove existence of solutions. We shall choose $\delta = 0$ for numerical tests.

If the image is noisy, the noise is considered as a texture and will be included in u: more precisely v will be the part of the image without the oscillating component, that is the denoised part. In a previous work, [7], we focused on the denoising process taking only v into account (and assuming that $u = 0$ so that $u_d - v$ is the noise). We now give an existence and uniqueness result for the general problem ($\mathcal{P}$) (see [7] for the proof).

Theorem 1. *Assume that $\lambda > 0, \mu > 0$ and $\delta > 0$. Problem ($\mathcal{P}$) has a unique solution (u,v).*

2 Numerical implementation

2.1 Discretization of the problem

We assume for simplicity that the image is square with size $N \times N$. We denote $X := \mathbb{R}^{N\times N} \simeq \mathbb{R}^{N^2}$ endowed with the usual inner product and the associated euclidean norm

$$\langle u, v\rangle_X := \sum_{1\leq i,j\leq N} u_{i,j} v_{i,j}, \quad \|u\|_X := \sqrt{\sum_{1\leq i,j\leq N} u^2_{i,j}} \; . \tag{3}$$

It is classical to define the discrete total variation as follows (see for example [4]) : the discrete gradient of the numerical image $u \in X$ is $\nabla u \in X^2$ defined by

$$(\nabla u)_{i,j} = \left((\nabla u)^1_{i,j}, (\nabla u)^2_{i,j}\right), \tag{4}$$

where

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

The (discrete) total variation $|u|_{BV(\Omega)}$ is given by

$$J_1(u) = \sum_{1\leq i,j\leq N} \left\|(\nabla u)_{i,j}\right\|_{\mathbb{R}^2}, \tag{5}$$

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 ∇ :

$$\forall (p,u) \in X^2 \times X, \quad \langle -\text{div } p, u\rangle_X = \langle p, \nabla u\rangle_{X^2},$$

so that

$$(\text{div } p)_{i,j} = \begin{cases} p^1_{i,j} - p^1_{i-1,j} & \text{if } 1 < i < N \\ p^1_{i,j} & \text{if } i = 1 \\ -p^1_{i-1,j} & \text{if } i = N \end{cases} + \begin{cases} p^1_{i,j} - p^2_{i,j-1} & \text{if } 1 < j < N \\ p^2_{i,j} & \text{if } j = 1 \\ -p^1_{i,j-1} & \text{if } i = N. \end{cases} \tag{6}$$

To define a discrete version of the second order total variation we have to introduce the discrete Hessian operator. As for the gradient operator, we define it by finite differences. So, for any $v \in X$, the Hessian matrix of v, denoted Hv is identified to a X^4 vector:

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

The discrete second order total variation $|v|_{BV^2(\Omega)}$ of v is defined as

$$J_2(v) = \sum_{1\leq i,j\leq N} \|(Hv)_{i,j}\|_{\mathbb{R}^4}. \tag{7}$$

As in the BV case, we may compute the adjoint operator of H (which is the discretized "second divergence" operator) :

$$\forall p \in X^4,\ \forall v \in X \qquad \langle H^* p, v\rangle_X = \langle p, Hv\rangle_{X^4}. \tag{8}$$

and we deduce a numerical expression for H^* from the equality (8). The discretized problem stands

$$\inf_{(u,v)\in X^2} \frac{1}{2}\|u_d - u - v\|^2_X + \lambda J_1(u) + \mu J_2(v) + \delta(|v| + J_1(v)), \tag{$\mathcal{P}_d$}$$

where

$$|v| := \sum_{1\leq i,j\leq N} |v_{i,j}|\ .$$

In the finite dimensional case we still have an existence result.

Theorem 2. *Problem $\mathcal{P}_d$ has a unique solution for every $\lambda > 0$, $\mu > 0$ and $\delta > 0$.*

For numerical purpose we shall set $\delta = 0$. In fact, we have performed tests with $\delta = 0$ and very small $\delta \neq 0$ (as required by the theory to get a solution to problem $\mathcal{P}_d$) and results where identical. So, to simplify numerical implementation, we consider the following discretized problem :

$$\inf_{(u,v)\in X^2} \frac{1}{2}\|u_d - u - v\|_X^2 + \lambda J_1(u) + \mu J_2(v). \qquad (\tilde{\mathcal{P}}_d)$$

2.2 Algorithm

Using non smooth analysis tools (for convex functions) it is easy to derive (necessary and sufficient) optimality conditions. More precisely (u, v) is a solution of $(\tilde{\mathcal{P}}_d)$ if and only if

$$\begin{cases} 0 \in \partial\left(\lambda J_1(u) + \frac{1}{2}\|u_d - u - v\|^2\right) \\ 0 \in \partial\left(\mu J_2(v) + \frac{1}{2}\|u_d - u - v\|^2\right), \end{cases} \tag{9}$$

where ∂J is the classical subdifferential of J. Using subdifferential properties, we see that (9) is equivalent to

$$\begin{cases} u = u_d - v - \Pi_{\lambda K_1}\left(u_d - v\right) \\ v = u_d - u - \Pi_{\mu K_2}\left(u_d - u\right). \end{cases} \tag{10}$$

where K_1 and K_2 are closed convex sets. Chambolle [8] proved that

$$K_1 = \{\mathrm{div}\ p \mid p \in X^2,\ \|p_{i,j}\|_{\mathbb{R}^2} \leq 1\ \forall i, j = 1, \dots, N\} \tag{11}$$

in the $BV(\Omega)$ setting and we may prove similarly that

$$K_2 = \{H^* p \mid p \in X^4,\ \|p_{i,j}\|_{\mathbb{R}^4} \leq 1,\ \forall i, j = 1, \dots, N\}, \tag{12}$$

(see [7]). Moreover, Chambolle [8] proposed a fixed point algorithm to compute $\Pi_{\lambda K_1}$ and we are able to extend this result to the second order case.

$$p^0 = 0 \tag{13a}$$

$$p_{i,j}^{n+1} = \frac{p_{i,j}^n - \tau(H[H^* p^n - u_d/\lambda])_{i,j}}{1 + \tau\|(H[H^* p^n - u_d/\lambda])_{i,j}\|_{\mathbb{R}^4}}. \tag{13b}$$

which convergence is proved in [7] :

Theorem 3. *Let $\tau \leq 1/64$. Then $\lambda\left(H^* p^n\right)_n$ converges to $\Pi_{\lambda K_2}(u_d)$.*

So, we propose the following algorithm :

- ***Step 1 :*** *We choose u_0 et v_0 (for example, $u_0 = 0$ et $v_0 = u_d$) and* $0 < \alpha < 1$.
- ***Step 2 :*** *define the sequences $\left((u_n, v_n)\right)_n$ as follows:*

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

- ***Step 3 :*** *if a stopping criterion is not satisfied, set $k := k+1$ and go back to 2.*

We can show that the algorithm converges for $\alpha \in]0, 1/2[$. In practice, we observed that the convergence is faster for $\alpha = 0.6$.

3 Numerical tests and comments

We test the model on two images: The first one is a synthetic image where texture has been artificially added, and the second one is the well known "Barbara" benchmark, often used in texture extraction.

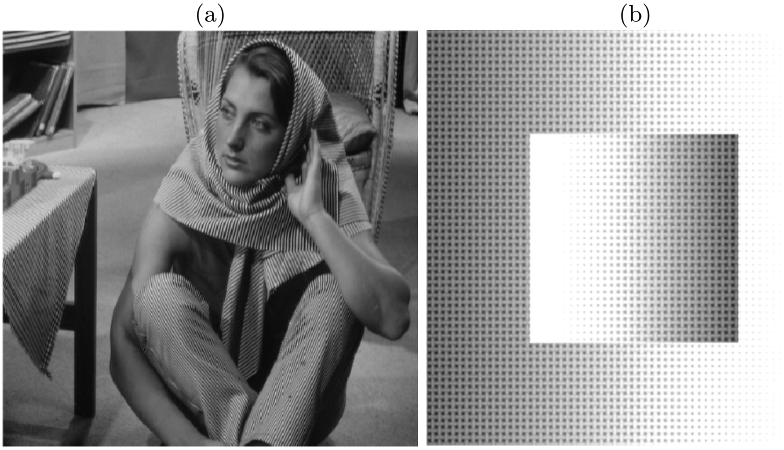

Fig. 1. Original images.

We perform many tests with respect to the different parameters. We only present here the most significant : $\alpha = 0.6$, $\lambda = 0.5$ and $\mu = 100$. Let us first report on the iterations number effect with image (a).

If we are only interested in the texture part, we can observe in fig 2 that we get back all the textures. Unfortunately, most of the geometrical information (that we don't want) is also kept, and we observe that the involved geometric part is getting more important as the iteration number is growing.

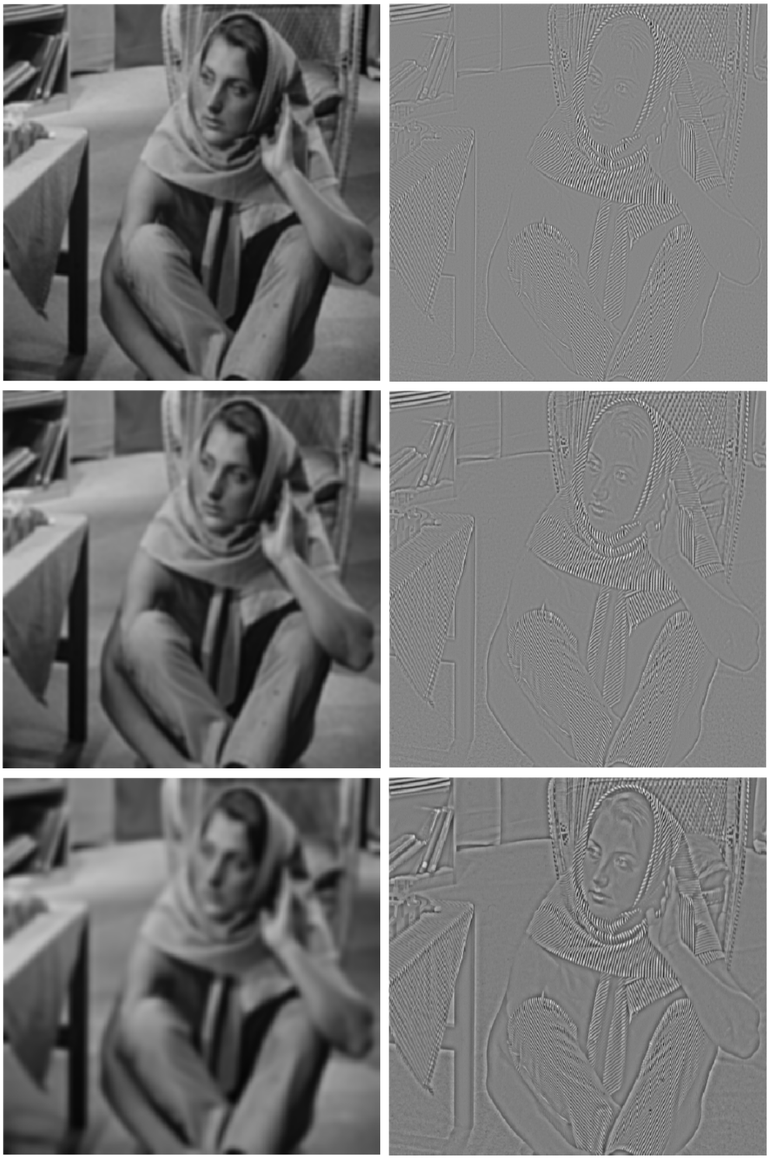

Fig. 2. Number of iterations: first line: 60; second line: 200; third line: 2000.

We see in fig 3 that we can choose a large number of iterations for the texture extraction of image (b) because of its inner structure.

On the other hand, we have to limit this number for the image (a). We give an example image (b) with a very large iterations number.

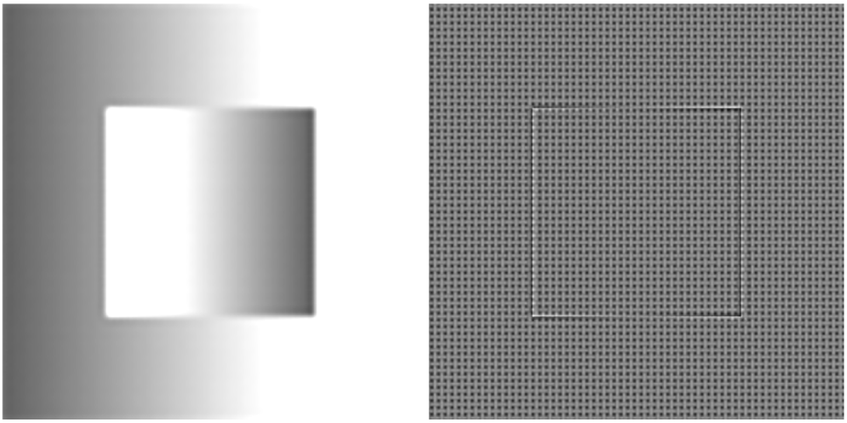

Fig. 3. Number of iterations: 2000.

In addition, we see that too many geometrical information remains together with the texture in the oscillating part: this is a bad point. Nevertheless, our main goal is to locate the texture and we don't need to work with the cartoon part anymore once it has been identified. We do not need to recover all the texture but only a significant part to identify it. In that case, we propose a method that permits to improve the results significantly: we modify the Hessian operator to make it anisotropic. More precisely, we reinforce chosen directions. As texture is made of oscillating information, we hope that we shall keep most of it while many contour lines disappear. We specially act on the vertical and horizontal components of the hessian operator. To deal with non vertical and horizontal lines, we just have to let the image rotate. In the following test, we have replaced the Hessian operator by the operator H' defined for all $v \in X^4$ by :

$$\forall (i,j) \in \{1,...,N\}^2, \quad (H'v)_{i,j} = \big(0, (Hv)^{12}_{i,j}, (Hv)^{21}_{i,j}, 0\big) .$$

We can see on fig 4 that we keep most of the texture without geometrical information. Of course, this method is only efficient on contour lines which are beelines, and permits to deal with only two directions which are necessarily perpendicular. We will propose, in a forthcoming work, a local method to eliminate contour lines in every directions.

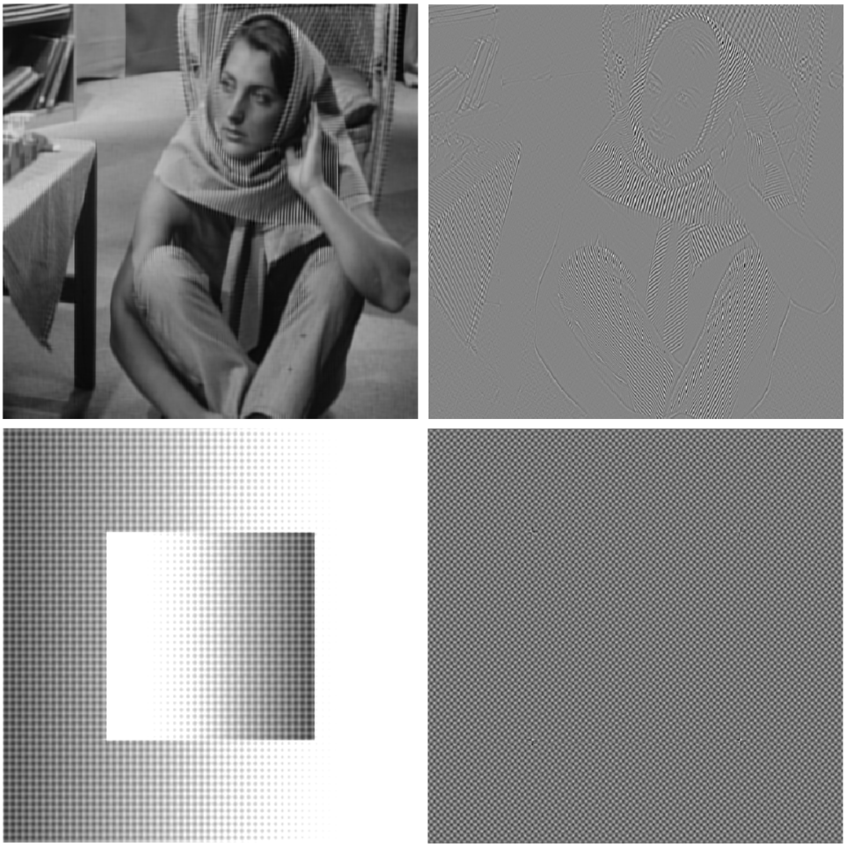

Fig. 4. Test with the anisotropic operator H'. Number of iterations: first line: 60; second line: 2000.

4 Conclusion

The model permits to extract texture from an image, but the texture part still contains too much geometric information. Thus, to recover what we are interested in, we have to use the algorithm with a limited number of iterations. Moreover, we have noticed that we recover too many contour lines as well. The asset of this model is that we can make it anisotropic, modifying the hessian operator in an appropriate way. Therefore we get rid of geometrical information, but we lose part of the texture as well. Nevertheless, if our goal is just to locate texture on an image, this loss remains acceptable.

References

1. Acar R., Vogel C. R. (1994) Analysis of bounded variation penalty methods for ill-posed problems. Inverse Problems, 10: 1217–1229
2. Ambrosio L., Fusco N., Pallara D. (2000) Functions of bounded variation and free discontinuity problems. Oxford mathematical monographs, Oxford University Press.
3. Attouch H., Buttazzo, Michaille G. (2006) Variational analysis in Sobolev and BV spaces : applications to PDEs and optimization. MPS-SIAM series on optimization
4. Aubert G., Kornprobst P. (2006) Mathematical Problems in Image Processing, Partial Differential Equations and the Calculus of Variations. Applied Mathematical Sciences 147, Springer Verlag
5. Aubert G., Aujol J.-F. (2005) Modeling very oscillating signals, application to image processing. Applied Mathematics and Optimization, 51: 163-182
6. Aubert G., Aujol J.-F., Blanc-Feraud L., Chambolle A. (2005) Image decomposition into a bounded variation component and an oscillating component. Journal of Mathematical Imaging and Vision, 22: 71–88
7. Bergounioux M., Piffet L. (2010) A $BV^2(\Omega)$ model for image denoising and/or texture extraction , submitted, Set Valued Analysis
8. Chambolle A. (2004) An algorithm for total variation minimization and applications. Journal of Mathematical Imaging and Vision, 20: 89–97
9. Demengel F. (1984) Fonctions à hessien borné. Annales de l'institut Fourier, 34: 155–190
10. Meyer Y. (2002) Oscillating patterns in image processing and nonlinear evolution equations. Vol. 22 of University Lecture Series, AMS
11. Osher S., Fatemi E., Rudin L. (1992) Nonlinear total variation based noise removal algorithms. Physica D 60:259-268
12. Osher S., Vese L., (2004) Image denoising and decomposition with total variation minimization and oscillatory functions. Special issue on mathematics and image analysis. Journal of Mathematical Imaging and Vision, 20: 7–18
13. Piffet L. (2010) Modèles variationnels pour l'extraction de textures 2D, PhD Thesis, Université d'Orléans
14. Yin W., Goldfarb D., Osher S. (2007) A comparison of three total variation based texture extraction models. Journal of Visual Communication and Image Representation, 18: 240–252
15. Echegut R. (2009) Rapport de stage de Master 1, Université d'Orléans