



**HAL**  
open science

## A new variational method for preserving point-like and curve-like singularities in 2D images

Daniele Graziani, Laure Blanc-Féraud, Gilles Aubert

### ► To cite this version:

Daniele Graziani, Laure Blanc-Féraud, Gilles Aubert. A new variational method for preserving point-like and curve-like singularities in 2D images. International Conference on Acoustics, Speech and Signal Processing, 2011, Prague, Czech Republic. inria-00592603

**HAL Id: inria-00592603**

**<https://hal.inria.fr/inria-00592603>**

Submitted on 13 May 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A NEW VARIATIONAL METHOD FOR PRESERVING POINT-LIKE AND CURVE-LIKE SINGULARITIES IN 2-D IMAGES

*Daniele Graziani, Laure Blanc-Féraud and Gilles Aubert*

ARIANA CNRS/INRIA/UNSA SOPHIA ANTIPOLIS  
Inria, 2004 route des Lucioles BP 93 06902 SOPHIA ANTIPOLIS CEDEX, FRANCE  
e.mail: Daniele.Graziani@sophia.inria.fr

ARIANA CNRS/INRIA/UNSA SOPHIA ANTIPOLIS  
Inria, 2004 route des Lucioles BP 93 06902 SOPHIA ANTIPOLIS CEDEX, FRANCE  
e.mail: Laure.Blanc.Feraud@sophia.inria.fr

LABORATOIRE J.A. DIEUDONNÉ  
Université de Nice SOPHIA ANTIPOLIS, parc valrose 06108 Nice CEDEX 2, FRANCE  
e.mail: gaubert@unice.fr

## ABSTRACT

We propose a new variational method to restore point-like and curve-like singularities in 2-D images. As points and open curves are fine structures, they are difficult to restore by means of first order derivative operators computed in the noisy image. In this paper we propose to use the Laplacian operator of the observed intensity, since it becomes singular at points and curves. Then we propose to restore these singularities by introducing suitable regularization involving the  $l$ -1-norm of the Laplacian operator. Results are shown on synthetic and real data.

**Index Terms**— image processing, non smooth convex optimization,  $l^1$ -minimization, Nesterov scheme, laplacian operator

## 1. INTRODUCTION

The issue of preserving fine structures in image reconstruction, such as points or curves in 2-D, has known an increasing interest for scientific purposes, e.g., in biology and astronomy, for instance filaments and spots detection in biomedical and astronomical images, road and building extraction from satellite images. There is a vast literature on this subject and a lot of different techniques have been proposed in these last years. Without claiming of being exhaustive we refer to [5] for segmentation of blood vessel via morphological reconstruction, [4] for filaments detection in 3D by using active contours approach and [2] for object tracking based on sparsity principles and source separation. In this paper we focus on the variational point of view. Usually these images, possibly corrupted by noise, are characterized by an high intensity value on sets of low dimension, such as points or curves in 2-D, and which decreases to 0 in a neighborhood of the singularities. In this work we propose a new model for point and curve restoration, where these low dimensional sets are considered as singularities in the image, given in term of a proper differential operator defined on the intensity of the image. We provide, in a discrete setting, a new variational formulation for restoring such singularities in noisy images.

---

The research of Daniele Graziani is supported by CNRS under the research project "Gyrovision".

## 2. PRELIMINARIES

This section is devoted to notation, preliminaries results, and to the introduction of fast descent gradient scheme we will use in the sequel.

### 2.1. Discrete setting

We define the discrete rectangular domain  $\Omega$  of step size  $\delta x = 1$  and dimension  $d_1 d_2$ .  $\Omega = \{1, \dots, d_1\} \times \{1, \dots, d_2\} \subset \mathbb{Z}^2$ . In order to simplify the notations we set  $X = R^{d_1 \times d_2}$  and  $Y = X \times X$ .  $u \in X$  denotes a matrix of size  $d_1 \times d_2$ . For  $u \in X$ ,  $u_{i,j}$  denotes its  $(i, j)$ -th component, with  $(i, j) \in \{1, \dots, d_1\} \times \{1, \dots, d_2\}$ . For  $g \in Y$ ,  $g_{i,j}$  denotes the  $(i, j)$ -th component of with  $g_{i,j} = (g_{i,j}^1, g_{i,j}^2)$  and  $(i, j) \in \{1, \dots, d_1\} \times \{1, \dots, d_2\}$ . We endowed the space  $X$  and  $Y$  with standard scalar product and standard norm. For  $u, v \in X$ :

$$\langle u, v \rangle_X = \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} u_{i,j} v_{i,j}.$$

For  $g, h \in Y$ :

$$\langle g, h \rangle_Y = \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \sum_{l=1}^2 g_{i,j}^l h_{i,j}^l.$$

For  $u \in X$  and  $p \in [1, +\infty)$  we set:

$$\|u\|_p := \left( \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} |u_{i,j}|^p \right)^{\frac{1}{p}}.$$

For  $g \in Y$  and  $p \in [1, +\infty)$ :

$$\|g\|_p := \left( \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \sum_{l=1}^2 |g_{i,j}^l|^p \right)^{\frac{1}{p}}.$$

If  $G, F$  are two vector spaces and  $H : G \rightarrow F$  is a linear operator the norm of  $H$  is defined by

$$\|H\| := \max_{\|u\|_G \leq 1} (\|Hu\|_F).$$

**Definition 2.1** A function  $F : X \rightarrow \mathbb{R}$  is said to be  $L$ -lipschitz differentiable if it is differentiable and

$$|\nabla F(u) - \nabla F(v)|_2 \leq L|u - v|_2,$$

for every  $u, v \in X$ .

**Definition 2.2** Let  $\psi : X \rightarrow \mathbb{R}$  be a convex function. The operator

$$\text{prox}_\psi : X \rightarrow X \quad x \mapsto \arg \min_{y \in X} \left\{ \psi(y) + \frac{1}{2}|y - x|_2^2 \right\}$$

is called proximal operator associated to  $\psi$ .

If  $\text{prox}_{\lambda\psi}$  can be computed exactly for every  $\lambda \geq 0$  and every  $x \in X$ , the function  $\psi$  is said to be simple.

If  $u \in X$  the gradient  $\nabla u \in Y$  is given by:

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

where

$$(\nabla u)_{i,j}^1 = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < d_1 \\ 0 & \text{if } i = d_1, \end{cases}$$

$$(\nabla u)_{i,j}^2 = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < d_2 \\ 0 & \text{if } j = d_2. \end{cases}$$

We also introduce the discrete version of the divergence operator defined as the adjoint operator of the gradient:  $\text{div} = -\nabla^*$ . Then we can define the discrete version of the Laplacian operator as  $\Delta u = \text{Div}(\nabla u)$ .

## 2.2. Nesterov algorithm

Here we briefly recall the fast descent gradient Nesterov's algorithm (see [6]) we use to minimize 6. We state it in the formulation proposed in [7]. For further details and general statements we refer the reader to [7] and references therein.

**Proposition 2.1** Let  $F : X \rightarrow \mathbb{R}$  be given by:

$$F(u) = F_1(u) + F_2(u) \quad \text{for } u \in X,$$

where  $F_1$  is a convex  $L$ -Lipschitz differentiable function and  $F_2$  a simple function. Then the following algorithm:

$$\left\{ \begin{array}{l} u_0 \in X \quad A_0 = 0 \quad g = 0 \quad u = 0 \\ \text{do for } k : 1, \dots, K \\ t = \frac{2}{L} \\ a = t + \sqrt{t^2 + 4tA} \\ v = \text{prox}_{AF_2}(u_0 - g) \\ y = \frac{Au + Av}{A+a} \\ u = \text{prox}_{\frac{1}{L}F_2}(y - \frac{1}{L}\nabla F_1(y)) \\ g = g + a\nabla F_1(u) \\ A = A + a \end{array} \right. \quad (1)$$

ensures that:

$$0 \leq F(u_k) - F(u^*) \leq L \frac{|u^* - u_0|_2}{k^2}, \quad (2)$$

where  $u^* \in X$  is a minimum point of  $F$  and  $u_0 \in X$  is an initial data.

## 3. THE VARIATIONAL METHOD

### 3.1. The discrete functionals

From a general point of view in order to restore the given data  $u_0$ , one would like to minimize an energy with an  $L^2$ -fidelity term and a proper regularization criterion  $F$ . To this end one has to solve the following minimum problem:

$$\min_{u \in X} F(u) + \frac{\lambda}{2}|u - u_0|_2^2. \quad (3)$$

$\lambda > 0$  is a positive weight,  $X$  is a suitable space and the criterion  $F$  must be chosen according to the singularities to be restored. In our case it means that  $F$  must be given by a proper differential operator which allows for singularities on points and curves. Moreover in the choice of  $F$  we must take into account that the singularities to be preserved are not jump singularities. It means that, in the continuous setting,  $u$  must belong to the space  $\Delta\mathcal{M}^p(\Omega)$  of functions whose gradient is an  $L^p$ -vector field with distributional divergence given by a Radon measure (see [1] for the precise definition of this space). These considerations lead us to choose the laplacian as differential operator and minimize the following energy:

$$J(u) = \|\Delta u\|_1 + \frac{1}{p}\|\nabla u\|_p^p + \frac{\lambda}{2}|u - u_0|_2^2, \quad (4)$$

where  $\lambda$  is a positive weight and  $1 < p < 2$ . The restriction on  $p$  is due to the fact that when  $p \geq 2$  the distributional laplacian  $\Delta u$  of  $u$  cannot be a measure concentrated on points (see [1, 3] on this issue). Therefore it would not be anymore the right operator to restore the singularities we are interested in. In order to apply algorithm (1) it is necessary to smooth both  $|\cdot|_1$  and  $\|\cdot\|_p$  norm. Indeed the first one is not even differentiable, while the second one is not  $L$ -Lipschitz differentiable for  $p < 2$ . To this purpose we introduce the Huber function defined by:

$$w_\epsilon(x) = \begin{cases} |x| & \text{if } |x| \geq \epsilon \\ \frac{x^2}{2\epsilon} + \frac{\epsilon}{2} & \text{otherwise.} \end{cases} \quad (5)$$

Then we introduce the smoother counterpart of functional (4) as:

$$J_\epsilon(u) = \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} w_\epsilon(|(\Delta u)_{i,j}|) + \frac{1}{p} \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} w_\epsilon(|(\nabla u)_{i,j}|^p) + \frac{\lambda}{2}|u - u_0|_2, \quad (6)$$

where  $\epsilon > 0$  here is a small fixed parameter. We shall consider the minimization problems:

$$\min_{u \in X} J(u), \quad (7)$$

$$\min_{u \in X} J_\epsilon(u). \quad (8)$$

As in [7] we define the notion of  $\delta$ -solution associated to problem (7), which will be used to give an estimation of the number of iterations of the minimization algorithm.

**Definition 3.1** A  $\delta$ -solution of (7) is an element  $u_\delta \in X$  such that

$$J(u_\delta) - J(\bar{u}) \leq \delta,$$

where  $\bar{u}$  is a solution of problem (7).

We minimize  $J_\epsilon$  by applying algorithm (1) with

$$F_1(u) = J_\epsilon(u) \quad F_2(u) = 0.$$

Indeed it is not difficult to check that for  $u \in X$  with  $u = 0$  on the boundary we have

$$\nabla F_1(u) = \Delta(\Psi) - \operatorname{div}(\Phi) + \lambda(u - u_0) \quad (9)$$

where

$$\Psi_{i,j} = \begin{cases} \frac{(\Delta u)_{i,j}}{|(\Delta u)_{i,j}|} & \text{if } |(\Delta u)_{i,j}| \geq \epsilon \\ \frac{(\Delta u)_{i,j}}{\epsilon} & \text{otherwise,} \end{cases}$$

$$\Phi_{i,j} = \begin{cases} \frac{(\nabla u)_{i,j}}{|(\nabla u)_{i,j}|^{2-p}} & \text{if } |(\nabla u)_{i,j}|^p \geq \epsilon \\ \frac{(\nabla u)_{i,j}}{\epsilon} & \text{otherwise.} \end{cases}$$

Then, taking into account that  $\Psi$  and  $\Phi$  are Lipschitz functions with constant  $\frac{1}{\epsilon}$  and  $\frac{1}{\epsilon^p}$  respectively, we infer

$$|\nabla F_1(u) - \nabla F_1(v)|_2 \leq \left( \frac{\|\Delta\|_2^2}{\epsilon} + \frac{\|\operatorname{div}\|_2^2}{\epsilon^{2-p}} + \lambda \right) |u - v|_2.$$

Therefore, by recalling that  $\|\Delta\|_2 \leq 8$  and  $\|\operatorname{div}\|_2 \leq 2\sqrt{2}$ , we conclude that

$$|\nabla F_1(u) - \nabla F_1(v)|_2 \leq \left( \frac{64}{\epsilon} + \frac{8}{\epsilon^{2-p}} + \lambda \right) |u - v|_2. \quad (10)$$

Thanks to inequality (10) we are in position to apply algorithm (1). In our case algorithm (1) ensures that:

$$0 \leq J_\epsilon(u_k) - J_\epsilon(u_\epsilon^*) \leq \left( \frac{64}{\epsilon} + \frac{8}{\epsilon^{2-p}} + \lambda \right) \frac{|u_\epsilon^* - u_0|_2}{k^2}, \quad (11)$$

where  $u_\epsilon^*$  is a minimum of  $J_\epsilon$ .

#### 4. EXAMPLES

Before running our algorithm all the parameters have to be fixed. It is easy to see that for every  $u \in X$  we have

$$0 \leq J_\epsilon(u) - J(u) \leq d_1 d_2 \epsilon. \quad (12)$$

Then by using (12) (11) and the fact that  $u_\epsilon^*$  is a minimum of  $J_\epsilon$  we have

$$J(u_k) \leq J_\epsilon(u_k) \leq J_\epsilon(\bar{u}) + \left( \frac{64}{\epsilon} + \frac{8}{\epsilon^{2-p}} + \lambda \right) \frac{|u_\epsilon^* - u_0|_2}{k^2},$$

where  $\bar{u}$  is a minimum of  $J$ . By applying again bound (12) we deduce

$$J(u_k) \leq J(\bar{u}) + d_1 d_2 \epsilon + \left( \frac{64}{\epsilon} + \frac{8}{\epsilon^{2-p}} + \lambda \right) \frac{|u_\epsilon^* - u_0|_2}{k^2}.$$

Therefore the worst case precision to get a  $\delta$ -solution of (7) is:

$$J(u_k) - J(\bar{u}) = \left( \frac{64}{\epsilon} + \frac{8}{\epsilon^{2-p}} + \lambda \right) \frac{|\bar{u} - u_0|_2}{k^2} + d_1 d_2 \epsilon;$$

then the optimal choices are

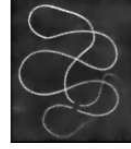
$$\epsilon = \frac{\delta}{d_1 d_2}, \quad K = \left\lceil \sqrt{\left( \frac{64 d_1 d_2}{\delta} + 8 \frac{d_1 d_2}{\delta^{2-p}} + \lambda \right) C} \right\rceil + 1,$$

where  $C := \max_X |\bar{u} - u_0|_2$  and  $K$  the total number of iterations. For images rescaled in  $[0, 1]$ , in the worst case problem, the number of iterations  $K$  needed to get a  $\delta$ -solution of order 1 does not exceed

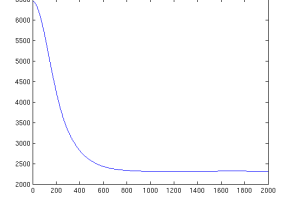


(a) Original image

(b) Noisy image



(c) Restored image



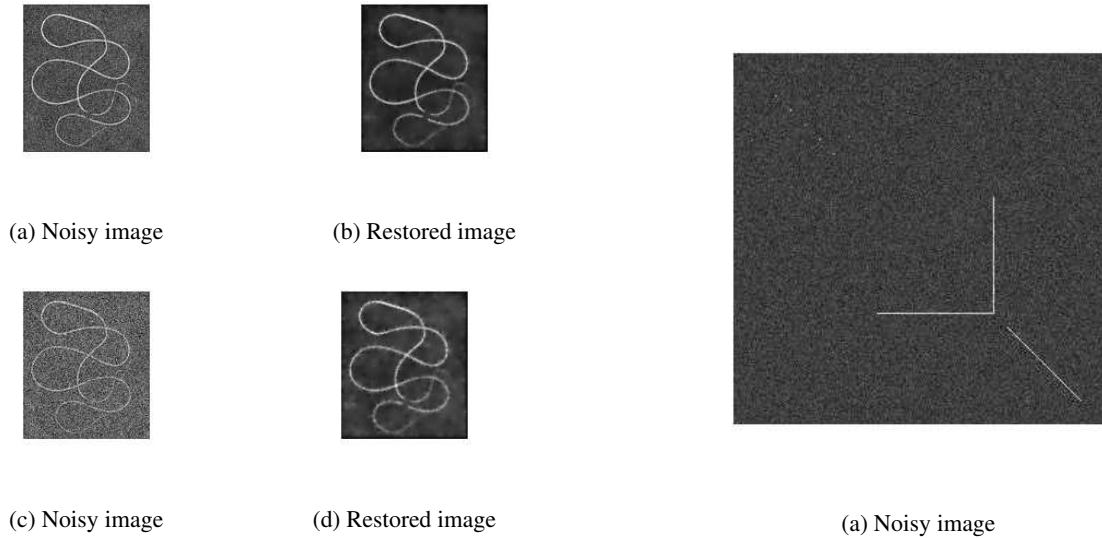
(d) Convergence of the algorithm

**Fig. 1.** Synthetic noisy image PSNR 26.1Db: we test our algorithm on a noisy image containing open curves. Image size  $d_1 \times d_2 = 128 \times 128$ .  $\lambda = 40$ .  $\epsilon = 6.1e - 5$ . Number of iterations  $K = 2000$ .

the value 8000. In all numerical test we let run the algorithm for no more than 4000 iterations. We show a convergence curve with the value of  $J(u_k)$  on the  $y$ -axis and the number of iterations on the  $x$ -axis. The parameter  $\epsilon$  is always fixed in order to get a  $\delta$ -solution of order 1. This choice seems to lead to good restoration results. The parameter  $\lambda$  is tuned according to the level noise and its value is specified on each numerical test. Finally as exponent  $p$  we always take  $p = 1.5$ . In Figure 1 and 2 we test the algorithm against noise on synthetic image containing open curve. A Gaussian Noise is added to the original image. The image domain is of size  $d_1 \times d_2 = 128 \times 128$ . CPU time is about 50 s running on an Intel (R) Xeon(R) CPU 5120 at 1.86GHz. In Figure 3 we test the algorithm against noise on a synthetic image containing points and open curves, which are the singularities we want to be preserved in the restoration process. A Gaussian noise is added to the original image. The image domain is of size  $d_1 \times d_2 = 256 \times 256$ . CPU time is about 8mn s running on an Intel (R) Xeon(R) CPU 5120 at 1.86GHz. Finally in Figure 4 we test our model on real noisy data. The image domain is of size  $d_1 \times d_2 = 256 \times 256$ . CPU time is about 9mn running on an Intel (R) Xeon(R) CPU 5120 at 1.86GHz.

#### 5. CONCLUSION

In this work a new variational model for restoring point and curve-like singularities in images has been developed both from a theoretical and an experimental point of view. We emphasize that, according to our knowledge, this approach seems to be new in the literature. Moreover we stress out that, despite some extra smoothing effect, mainly due to the fact we work with a mesh step size equal to 1, we obtain good experimental results. There are many rooms for improvement from a numerical point of view, such as analysis of functional with more general discrepancy term of type  $|Hu - u_0|^q$ , where  $H$  is a linear operator modeling the blur and  $q \geq 1$ ; as well as performing 3D-numerical simulations on real data. These are subject of our current investigation.



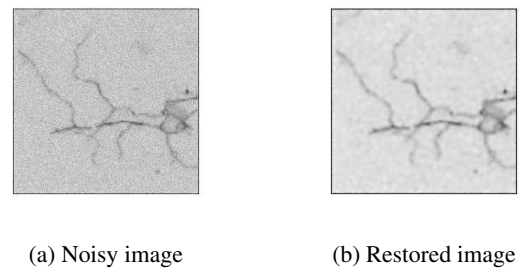
**Fig. 2.** Synthetic noisy images: we test our algorithm on noisy images. Top left: noisy image of size  $d_1 \times d_2 = 128 \times 128$ ; PSNR 20.4Db. Top right: restored image  $\lambda = 25$ ;  $\epsilon = 6.1e - 5$ ; number of iterations  $K = 2000$ . Down left: noisy image of size  $d_1 \times d_2 = 128 \times 128$ , PSNR 14.1Db. Down right: restored image  $\lambda = 10$ ;  $\epsilon = 6.1e - 5$ ; number of iterations  $K = 2000$ .

## 6. REFERENCES

- [1] G.Aubert, D.Graziani. *Variational approximation for detecting point-like target problems*. COCV: Esaim Control Optimisation and Calculus of Variations DOI: 10.1051/cocv/2010029 (2010).
- [2] N.Chenouard S.Vernhettes, I.Bloch, J.-C Olivo-Marin. *Morphological source separation for particle tracking in complex biological environments* 19th IAPR International Conference on Pattern Recognition, ICPR 2008, Tampa, USA, 2008.
- [3] D.Graziani, L.Blanc-Féraud, G.Aubert. *A formal  $\Gamma$ -convergence approach for the detection of points in 2-D images*. Siam Journal of Imaging Science **3** (2010), 578-594.
- [4] M.Jacob, T.Blu, C.Vaillant, J.H. Maddocks, M.Unser *3-D shape estimation of DNA molecules from stereo cryo-electron micrographs using a projection-steerable*. IEEE Transaction on Image Processing, **15** (2006), 214-227.
- [5] A.M. Mendonca, Aurèlio Campilho. *Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction*. IEEE Transaction on Medical Imaging, **25** (2006), 1200-1213.
- [6] Y. Nesterov. *Smooth minimization of non-smooth functions*. Mathematic Programming, Ser. A, **103** (2005), 127-152.
- [7] P. Weiss, L. Blanc-Féraud, G. Aubert. *Efficient schemes for total variation minimization under constraints in image processing*. SIAM journal on Scientific Computing **31** (2009), 2047-2080.



**Fig. 3.** Synthetic noisy image PSNR 19.1Db: we test our algorithm on a noisy image containing 5 points and 2 open lines. Image size  $d_1 \times d_2 = 256 \times 256$ .  $\lambda = 8$ .  $\epsilon = 15e - 6$ . Number of iterations  $K = 4000$ .



**Fig. 4.** Real noisy image: we test our algorithm on a real image of a blood vessels network corrupted by Gaussian noise. Image size  $d_1 \times d_2 = 256 \times 256$ .  $\lambda = 10$ .  $\epsilon = 15e - 6$ . Number of iterations  $K = 4000$ .