Definition of anisotropic de-noising operators through sectional curvature

Wide range of applications from gray-level images to high resolution Doppler spectrum

Stanley Durrleman
stanley.durrleman@polytechnique.org

Frederic Barbaresco
frederic.barbaresco@fr.thalesgroup.com

Thales Air Defence, 7/9 rue des Mathurins
92221 Bagneux Cedex - France

Abstract

Removing noise in measured physical data is a task of great importance in a wide range of scientific fields : satellite imaging, medical imaging, radar signal processing... The known de-noising methods in the literature are often defined only for a particular application and, as far as we know, none are able to define a de-noising process that does not make any assumptions about the type of data. That's why we aim at defining differential operators that could be defined for data of any dimension in the real and complex spaces. These operators will be anisotropic in order to preserve the geometrical information contained in the data like edges or discontinuities. Moreover, there is rarely a canonical way to represent the data and since scientists are used to writing data in several coordinate systems, the operators will be invariant under a change of data parametrization as well.

Our approach is based on a geometrical model of noise resting on the sectional curvature. This geometrical growth enables us to distinguish points of noise from points of edges or discontinuities. Our method consists then in minimizing the total squared sectional curvature in the images. We first apply our ideas in the case of gray-level images for which the sectional curvature is the Gaussian curvature and have therefore well-known geometrical interpretations. We apply afterwards the method to de-noise radar Doppler spectrum, proving how generic it is.

1 Introduction

1.1 The de-noising problem

Let us consider a gray-valued image as a sub-sampling of a real valued function u defined on $\Omega \subset \mathbb{R}^2$. Ω is typically a rectangle. By extension $u(x,y) \in [0,1]$ denotes the gray level at the point (x,y). The main variations of the signal u are due to the edges in the image. In case of noisy data, in addition to the natural variations of u, noise induces some unexpected transitions. The problem, well known in the image processing community, is to remove the variations due to the noise while preserving the edges.

There are two main ways to deal with this problem : in one hand, we can characterize the noise thanks to its statistical properties. We make then some Bayesian estimations of the value of each pixel. These methods rely on a geometrical prior model (such as Ising or Potts models for example) to preserve the "image geometry" that is a certain spatial correlation in the image. In the other hand, we can characterize the noise through its geometrical properties. The state of the art in this approach takes advantage of the family of curvature motion algorithm defined in [2]. Roughly speaking, a point of noise is characterized by level sets of small area. This method is based on a contrast invariant hypothesis. If this hypothesis is relevant in the case of gray-level images, it cannot be applied to data of higher dimension such as color images or images of complex data given by a radar. The median filter, for example, is not applicable to data of higher dimension. For such general data it is more relevant to consider a geometrical invariance : a de-noising method should be independent of data parametrization. Several authors in [1], [4], [5] defined algorithms using the geometrical properties of the sub-manifold defined by the data. We used this approach to introduce the sectional curvature as a geometrical growth which can distinguish between noise and edges, which can be defined for data of any dimension and which does not depend on the parametrization of these data. At that time, it seems the geometrical growth which has the best properties for de-noising data.

1.2 A geometrical idea

We explain our main idea in the particular case of gray-level images.

Consider the graph of the function u, seen as a surface (i.e. a manifold of dimension 2) embedded in $\mathbb{R}^3$. Consider a point of the surface and look how its gray-level varies in its neighborhood : outside noise and edges the gray-level varies smoothly. In a point of edge, the gray-level varies smoothly in one direction, the edge one, and dramatically in another direction, the gradient one. In a point of noise the gray-level varies dramatically in every direction (in the case of a local extremum or a horse saddle for example) : in one sense, there is no spatial coherence. This discussion leads us to consider the principal curvatures [1] of the surface at a given point, and in particular the product of the two principal curvatures called the Gaussian curvature. In fact, at a standard point, both principal curvatures are small in absolute value, by comparison to the other cases. Indeed, at a point of an edge, one absolute curvature is big (in the direction of the gradient) and the other null (in case of a rectilinear edge with linear variation of its gray-level) or at least very small : the product is small in absolute value in comparison to the product of two big absolute principal curvatures at a point of noise. At a standard point, both curvatures are small and its absolute product is also small. The absolute Gaussian curvature appears therefore to be a measure that enables to distinguish between points of noise and points of edge. As we will see in the next section, this measure satisfies our constraint : it can be extended to data of any dimension and independently of the parametrization of these data.

Taking advantage of this intuitive discussion, we will define our model of unnoisy image as an image for which the interpolating function

[1] We denote by principal curvatures at a point p the maximum and the minimum of the curvatures of all curves drawn on the surface and passing by the point p.

u has a null Gaussian curvature at each point. We will thus de-noise an image by changing its pixel values in a way that minimize the total absolute Gaussian curvature of the graph of u.

Remark 1.1 *Obviously this definition depends on the way we interpolate the image. This is the main drawback of methods based on a continuous model. However, from an experimental point of view, the results we will obtain do not rely much on the interpolation since only the relative rough value of the Gaussian curvature is taken into account.*

Remark 1.2 *Such image model allows images with edges. It will not be necessary, thus, to introduce a data fitting criteria during the denoise process. This makes a great difference with many other algorithms that deal with such an additional arbitrary parameter that prevent the image from tending to a constant one.*

Remark 1.3 *In [8] the authors use the mean curvature H (the mean of the two principal curvatures) instead of the gaussian curvature. So they define de-noise algorithm called flow of Beltrami that makes the surface move with a speed proportional to H. Although this method gives better results than a standard isotropic de-noising, we can still observe slightly blurring effect at the edges. Indeed, at an edge, the mean curvature is not equal to 0 and the diffusion is not completely blocked.*

In the following, we first define the generalization of Gaussian curvature called sectional curvature for data of any dimension. Then we define a functional which measure the total sectional curvature and find two ways to minimize it :

- a gradient-descent scheme
- a stochastic relaxation algorithm

2 A geometrical model of noise

2.1 Data space, measure space and induced metric

From now, we will consider images of data of any dimension. We consider a function $u : \Omega \longrightarrow D$ where $\Omega \subset \mathbb{R}^2$ is the measure space, typically a rectangle or a circular sector, and $D \subset \mathbb{R}^d$, $d \geq 1$, is the data space. (In the case of gray-level value images, $d = 1$ and $D = [0,1]$). Since we want to define a metric growth that extends the Gaussian curvature we need to define a proper metric on Ω. For that purpose, we will first define a metric in the fiber bundle $\Omega \times D$ and then define a metric on the "data manifold" thanks to a pullback procedure.

In Ω, we basically measure the distance between two points (two pixels) thanks to the Euclidean distance. Hence, we provide Ω with the Euclidean metric :

$$e = dx^2 + dy^2$$

In D, we much choose a metric that enables to measure the distance between two measures. At that time, let us just suppose that we can provide D with such a metric h. h will be properly defined for each cases.

Given these two metrics e and h, we can now aggregate them into an induced metric g in Ω defined by the following canonical way :

$$g = g_{ij} dx^i dy^j \tag{2.1}$$

with

$$g_{ij} = e \times h\left(\partial_i P(x,y), \partial_j P(x,y)\right) \tag{2.2}$$

for all $(x,y) \in \Omega$ and $(i,j) \in \{0,1\}$. $e \times h$ denotes the metric on the cartesian product $\Omega \times D$, $P(x,y) = (x,y,u(x,y))^t$ and ∂_i holds to be $\partial/\partial x^i$, $x^0 = x$ and $x^1 = y$. This definition is justified by the following example.

Suppose that the data space is the space of gray-levels : $[0,1]$ $(d = 1)$ provided by the Euclidean metric : $h = dz^2$. We compute the induced metric g and we find :

$$\begin{aligned} g = \quad & \left(1 + (\partial_x u)^2\right) dx^2 + \left(1 + (\partial_y u)^2\right) dy^2 \\ & +2\left(\partial_x u\right)\left(\partial_y u\right) dxdy \end{aligned} \tag{2.3}$$

This is the standard metric on the surface defined by the graph of u considered as embedded in $\mathbb{R}^3$. In particular it enables to compute classical geometrical growths of the surface such as its area for example.

Remark 2.1 *Both e and h do not depend on the choice of a coordinate system neither in the acquisition space nor in the data space. In that sense, we say that the method depends only 'weakly' on the way data are parametrized. Nevertheless, the induced metric g do depend on the choice of coordinates.*

2.2 Sectional curvature

Let Ω be an open subset of $\mathbb{R}^2$ and g a metric defined on Ω. In the Riemannian framework, it exists only one torsion free derivation which preserves the metric : the Levi-Civita connection $\mathcal{D}$ (See [7] and the references therein for further details on Riemannian geometry). The Riemannian curvature tensor R is then defined for all vector fields X, Y and Z in Ω by :

$$R(X,Y)Z = \mathcal{D}_X(\mathcal{D}_Y Z) - \mathcal{D}_Y(\mathcal{D}_X Z) - \mathcal{D}_{[X,Y]}Z$$

where $[,]$ denotes the Lie brackets of two vector fields. In two dimension the completely covariant tensor $\overline{R}$ (defined by $\overline{R}(X,Y,Z,T) = g(R(X,Y)Z,T)$ for all vector fields X,Y,Z,T) has only one independent component, let say R_{xyxy} in a given coordinate system $(\partial/\partial x, \partial/\partial y)$. The sectional curvature K is then defined by :

$$K = -\frac{R_{xyxy}}{g_{xx}g_{yy} - g_{xy}^2} \qquad (2.4)$$

and it depends only on the metric g. In particular it does not depend on the choice of the coordinate system (x, y).

Coming back to our example ($D = [0,1]$), we can calculate the sectional curvature given the metric g defined by equation 2.3. We obtain :

$$K(u) = \frac{\partial_{xx}u\partial_{yy}u - (\partial_{xy}u)^2}{\left(1 + (\partial_x u)^2 + (\partial_y u)^2\right)^2} \qquad (2.5)$$

which is the expression of the Gaussian curvature of the surface at the current point (x, y). We thus achieve our goal : K is a geometric generalization of the Gaussian curvature for any kind of data.

2.3 The optimization problem

Taking advantage of our introductory discussion, we define the following functional that is independant of the coordinate system :

$$E(u) = \int_\Omega f(K(u))\sqrt{\det(g)}dxdy \qquad (2.6)$$

where f is a positive convex function. For the applications, we choose $f : x \longrightarrow x^2$, which is also smooth.

Our goal is to find a way to minimize E given an original noisy image u_0.

2.3.1 Gradient descent scheme

We calculate the gradient of E at u by calculating the unique function $L(u) : \Omega \longrightarrow \mathbb{R}^d$ which satisfies for all t small enough and all smooth functions $v : \Omega \longrightarrow \mathbb{R}^d$, equal to 0 at the border of Ω :

$$E(u + tv) = E(u) + t < L(u), v > +o(t)$$

where for all u and $v : \Omega \longrightarrow \mathbb{R}^d$, $< u, v >= \int_\Omega u(x,y)^t v(x,y)dxdy$.

Then we define u_t as the solution at time t of the gradient descent scheme for a given original image u_0 :

$$\begin{cases} u_{t=0} = u_0 \\ \partial_t u_t = -L(u_t) \end{cases} \qquad (2.7)$$

Practically we are able to calculate the gradient $L(u)$ only in the particular case of gray-level images ($D = [0,1]$) provided by the Euclidean metric ($h = dz^2$). In that case, we have the following evolution equation :

$$\partial_t u = \frac{1}{2}\text{div}\left(\frac{K^2\nabla u}{\sqrt{g}}\right) + \text{div}\left(\frac{\left(D^2u(\nabla K)^\perp\right)^\perp}{(\det g)^{3/2}}\right) \qquad (2.8)$$

where $D^2u = (\partial_{xx}u)dx^2 + (\partial_{yy}u)dy^2 + 2(\partial_{xy}u)dxdy$ is the hessian matrix of u, ∇u denotes the gradient of the function u and $n^\perp$ the normal vector of vector n.

Interpreting this equation is a difficult task. However, we can see what happen in two particular cases, a paraboloid of revolution ($u = \alpha(x^2 + y^2)/2$) or a hyperbolic paraboloid ($u =$

$\alpha(x^2 - y^2)/2)$. In the case of a paraboloid of revolution, we have :

$$\partial_t u = \frac{7\alpha^5}{2(\det g)^{11/2}} \left(2 - 7\alpha^2(x^2 + y^2)\right)$$

The paraboloid tends to evolves towards a mean value : the speed at the point $(0,0)$ as the same sign of α (a maximum goes down and a minimum goes up) and is proportional to the value of the Gaussian curvature at this point. In the case of an hyperbolic paraboloid, we have :

$$\partial_t u = -\frac{63\alpha^7}{2\,(\det g)^{11/2}} \left(x^2 - y^2\right)$$

The hyperbolic paraboloid tends to evolve towards the plane $z = 0$, the speed at one point being of the opposite sign of u.

2.3.2 Stochastic relaxation

For almost all cases, we are not able to compute the gradient descent scheme. To overcome this difficulty, we define an iterative stochastic relaxation algorithm. Let initialize the current image u with the original data u_0. Then repeat the following steps until no changes occur anymore :

1. Choose randomly a pixel p in u.
2. Change the value $u(p)$ by $u(p) + v$, where $v \in \mathbb{R}^d$ is chosen thanks to a centered Gaussian law of variance σ^2. Let us call the image obtained u'.
3. If $E(u') < E(u)$, go to step 1. If not, recover u from u' and go to step 1.

Here the value of σ plays a similar role of a data fitting criteria. In a general manner, we look for the local minimum of E which is the closest to the original data u_0. That's why we will choose σ not too small to increase the convergence speed and not too big to evolve towards the closest local minimum.

3 Applications

We write and run our algorithm with the freeware `Megawave` developed at CMLA Cachan[2] by J.Froment and L.Moisan. All algorithms presented here run in about 1 minute on a standard PC. (1 minute for each measure in the case of radar signal, which means 7 minutes for the whole signal). We used centered discrete scheme to estimate derivatives.

3.1 Gray-level valued images

This is the application for which we developed all our examples. Let us recall that in this case $D = [0,1]$, $d = 1$. If we provide D with the Euclidean metric $h = dz^2$, all results were given above.

We applied our two algorithms to a synthetic image (a white ellipse in a black background) noised additively by $Z = X^2 + Y^2$ where X and Y are two independent centered Gaussian variables of same standard deviation $\sigma = 0.1$. (see left image of fig 1) This is classical noise model in radar.

Figure 1 shows the application of the gradient descent scheme at three different times. ΔW and ΔB denotes respectively the variance reduction of the noise in the white and black area.

Figure 2-1 shows the application of the stochastic relaxation algorithm. It tends to gather noise peaks into little constant plateau. In order to avoid this effect, we accept a change not only if E but also the total variation of the image is decreasing. The result is now shown figure 2-2.

As we can see in these previous figures, the edges are perfectly preserved, whereas it remains a relatively high variance of noise in homogeneous areas. To deal with it, we choose to modify the metric h to create a distorsion in the space of gray levels ($D = [0,1]$). Practically, we choose the metric : $h = g(z)dz^2$ where $g : D \longrightarrow]0, \infty[$ is equal to $g(z) = 1 - \exp(-10 * (x - \bar{x}))$, and $\bar{x}$ is calculated at each

[2] CMLA, Ecole Normale Superieure de Cachan, 61 Avenue du President Wilson, 94235 Cachan cedex, France

pixel as the mean of a 3×3 window centered at the pixel weighted by the inverse Gaussian curvature. With this new adaptative metric, we penalized in homogeneous areas pixels which are far from the weighted mean at this point. The calculus of the sectional curvature with this adaptative metric gives :

$$\begin{aligned} K = \quad & g(u)\frac{\partial_{xx}u\partial_{yy}u-(\partial_{xy}u)^2}{\left(1+g(u)(\partial_x u)^2+g(u)(\partial_y u)^2\right)^2} \\ & +\frac{1}{2}g'(u)\frac{(\partial_y u)^2\partial_{xx}u+(\partial_x u)^2\partial_{yy}u-2\partial_x u\partial_y u\partial_{xy}u}{\left(1+g(u)(\partial_x u)^2+g(u)(\partial_y u)^2\right)^2} \end{aligned}$$

The application of this new algorithm combined with the previous one gives better results as we can see figure 3. In both area, we achieve a variance reduction of about 95%.

As shown figure 4, we compare our method with well-known methods : an iterated median filter (see [9] or [10] for further details on median filter or morphological filters) and a "despeckle" algorithm as computed in the freeware `The Gimp`. This last method underestimates the gray-level in the white area whereas the median filter enables to reach greater results. However, the median filter relies on an order relation in the space of gray-levels that can not be applicable in dimension greater than 1 (color images, radar signals,...).

Finally, we compare our algorithm using and adaptative metric and the median filter for an original RSO image. Results are shown figure 5 and 6.

3.2 Images of radar Doppler spectrum

We got a radar signal made of a burst of 8 complex pulses at each pixel of the measure space, a rectangle range-azimuth. (There are 120 measures along the distance axis and 31 azimuths.) These data are records of turbulent atmospheric clutter. We carried out a complex auto-regressive analysis with a regularized Burg algorithm. We thus obtained at each pixel a vector of 7 coefficients in the complex unit disk from which we can recover the whole Doppler spectrum. (see [3], [4] and [5] for further details on Burg algorithm and Doppler analysis). At one azimuth, we show figure 8 the 7 coefficient reflexion return by the Burg algorithm on the top, and the Doppler spectrum along the distance axis at the bottom. We see a major frequency which corresponds to the mean velocity of the cloud noised by parasite peaks of frequency. We would like to reconstruct the main real frequencies and remove the parasite frequencies from the spectrum.

In order to apply our algorithm, we need to define a metric on the data space, the complex unit disk. We would like to use the Poincare's metric but the singularity at the unit circle induces numerical instabilities. That's why we introduce two parameters λ and γ and define the metric :

$$h = \lambda\frac{d\alpha^2 + d\beta^2}{\left(1-\gamma^2\left(\alpha^2+\beta^2\right)\right)}$$

where $\mu = \alpha + i\beta$ is a complex number in the unit disk. For the measure space we keep the euclidean metric. We obtain therefore the following induced metric:

$$\begin{aligned} g = \quad & \left(1+(\partial_x\psi)^2+f(\psi)^2\left(\partial_x\theta\right)^2\right)dx^2 + \\ & \left(1+(\partial_y\psi)^2+f(\psi)^2\left(\partial_y\theta\right)^2\right)dy^2 + \\ & 2\left(\partial_x\psi\partial_y\psi+f(\psi)^2\partial_x\theta\partial_y\theta\right)dxdy \end{aligned}$$

where f, ψ and θ are three functions defined by:

$$\begin{aligned} f(\psi) = \quad & \frac{\lambda}{2\gamma}\sinh\left(\frac{2\gamma\psi}{\lambda}\right) \\ \psi = \quad & \frac{\lambda}{\gamma}\operatorname{atanh}(\gamma\rho) \\ \rho = \quad & \sqrt{\alpha^2+\beta^2} \\ \theta = \quad & \operatorname{atan}\left(\frac{\beta}{\alpha}\right) \pm \pi \end{aligned}$$

We can now compute the sectional curvature K and run a stochastic algorithm which minimizes the squared sectional curvature for each of the 7 complex pulses images.

As we need for the images to avoid some plateaux, we add to this minimization a new constraint : the total variation of the data must decrease as well during the minimization process.

We first apply the so defined algorithm to synthetic data : we synthesized an auto regressive signal of order one noised by a white noise. The data volume is defined along a given azimuth by two frequencies: one for the pixels of range lower than 75; another for the further pixels. We then calculate at each pixel the 7 coefficients thanks to the Burg algorithm. These original data are shown figures 7-1 (blue points) for the coefficients and figure 7-2 (left) for the spectrum. With $\lambda = 1.0$ and $\gamma = 0.7$ we achieve a reduction of 96% of the variance of the signal, as shown in the same figures. The two frequencies are well restaured and all the peaks of noise are removed. The discontinuities are perfecty preserved.

For the data records of atmospheric clutter, results are shown figure 8 for the azimuth 1, and figure 9 for the azimuth 19. In all cases, we set $\lambda = 0.1$ and $\gamma = 0.7$. The main frequency that corresponds to the main velocity of the cloud is well restaured. The peaks of noise are removed except along a curve that experts know as an clipping effect of the wind. These so small secondaries frequencies are not remove thanks to their spatial correlation taken into account by the algorithm.

4 Conclusion

We provide here a generic method to de-noise any kind of data. We illustrate it with two examples : the classical case of gray-level images and another case of high industrial importance : radar Doppler spectrum.

The results obtained, in comparison of the recent state-of-the-art, encourage us to explore new kind of applications such as colored images for which we can define a proper metric that could be de-noise taken into account 3 independent components at the same time. Metric are defined in this case in [8] for example.

References

[1] Ron Kimmel *Numerical Geometry of Images, theory, algorithms and applications*, Springer 2004.

[2] L.Alvarez, F.Guichard, P.L.Lions, J.M.Morel *Axioms and fundamental equations of image processing*, Archive for rational mechanics and analysis, 16(9):200-257

[3] F.Barbaresco *Algorithme de Burg regularise FSDS, comparaison avec l'algorithme de Burg MFE*, Proceedings of XVth coll. GRETSI'95

[4] F.Barbaresco *Information Intrinsic Geometric Flows*, MAXENT'06 Conf. Paris, July 2006 `http://djafari.free.fr/maxent2006`

[5] F.Barbaresco *Etude et extension des flots de Ricci, Kahler-Ricci et Calabi dans le cadre du traitement de l'image et de la geometrie de l'information*, GRETSI'05, Louvain la Neuve, Septembre 2005

[6] J-P.Bourguignon, N.Deruelle, *Relativite Generale*, Cours de l'Ecole Polytechnique, 2003

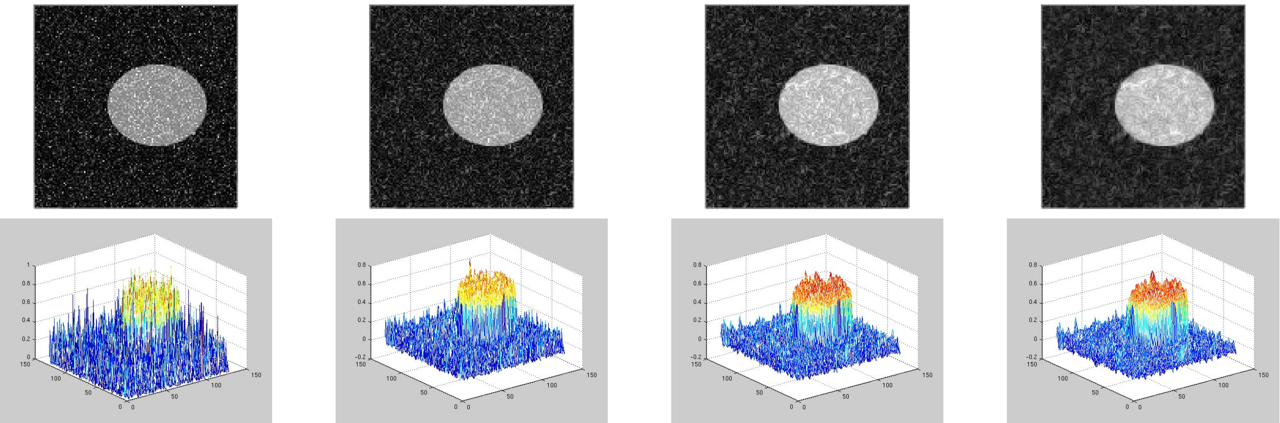
[7] M. Berger *Panoramic View of Riemannian Geometry*, Springer, 2004

[8] A.Sapira, N.Sochen, R.Kimmel *Geometric Filter, Diffusion Flows and Kernels in Image processing*, Handbook of Geometric Computing, Springer-Verlag, Heidelberg, pp 203-230, February 2005

[9] F. Guichard and J.M. Morel, *Image Iterative Smoothing and PDE's*, downloadable manuscript : mailto : fguichard@poseidon-tech.com, 2000

[10] F. Guichard and J.M. Morel *Mathematical Morphology 'almost everywhere'* Proceedings of ISMM 2002, p293-303

5 Figures



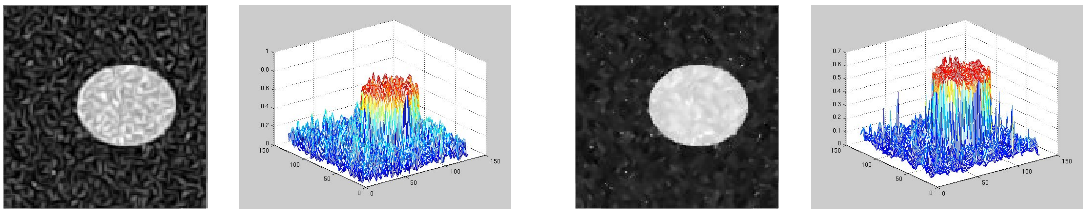
temps $t = 0$ Original Noisy Image

temps $t = 1$ $\Delta B = 46\%$ $\Delta N = 49\%$

temps $t = 10$ $\Delta B = 65\%$ $\Delta N = 67\%$

temps $t = 100$ $\Delta B = 71\%$ $\Delta N = 77\%$

Figure 1: Evolution of the image during the denoising process at 3 time steps



1- Stochastic algo. (100 iter) $\Delta B = 51\%$ $\Delta N = 52\%$

2- Stoch. algo. with regularisation (100 iter) $\Delta B = 99.2\%$ $\Delta N = 90\%$

Figure 2: Results of stochastic de-noising algorithms

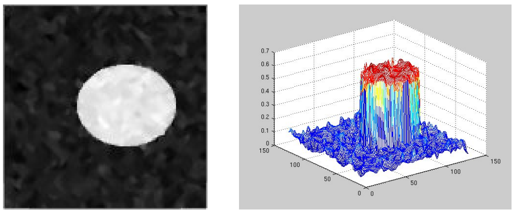
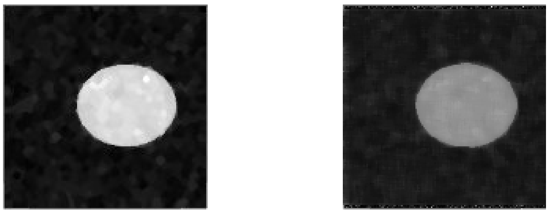


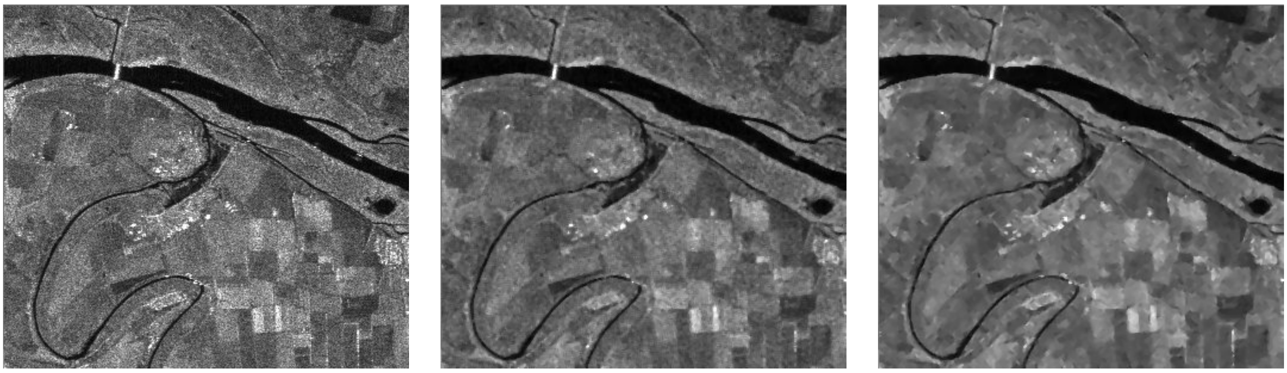
Figure 3: Results for the de-noising algorithm with use of an adaptative metric $\Delta B = 94.5\%$ $\Delta N = 95\%$



1- Median filter
$\Delta B = 94.8\%$ $\Delta N = 96\%$

2- Despeckle with `The Gimp`
$\Delta B = 95\%$ $\Delta N = 85\%$

Figure 4: De-noising with two classical methods

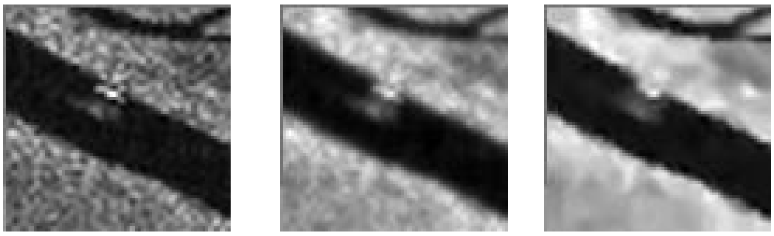


1- Original RSO image

2- median filter

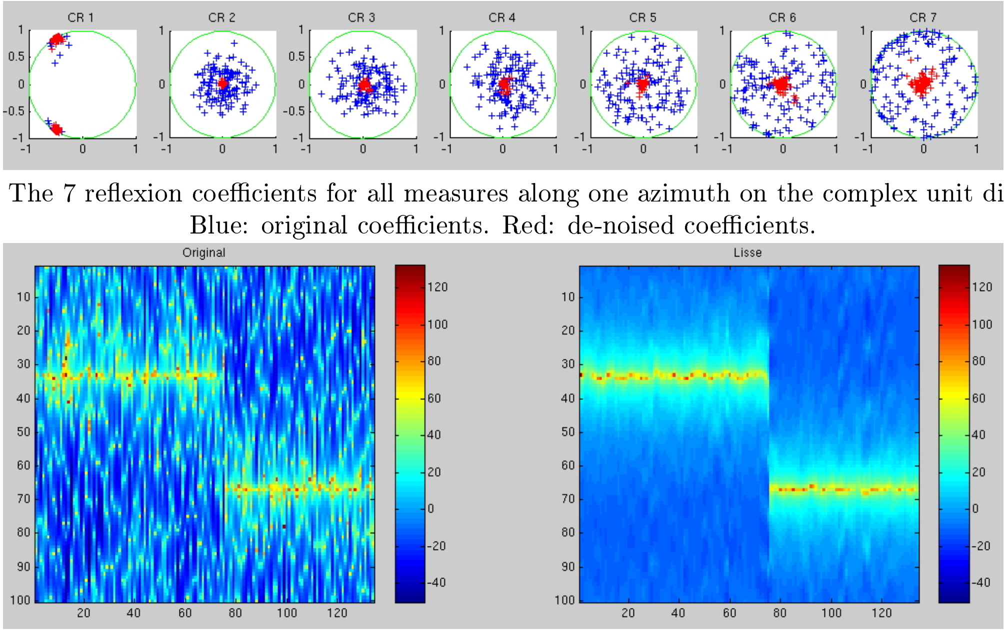
3- adaptative metric

Figure 5: RSO Image : comparison between adpatative metric and median filter



1- Original RSO detail 2- amss (voir [2]) 3- adaptative metric

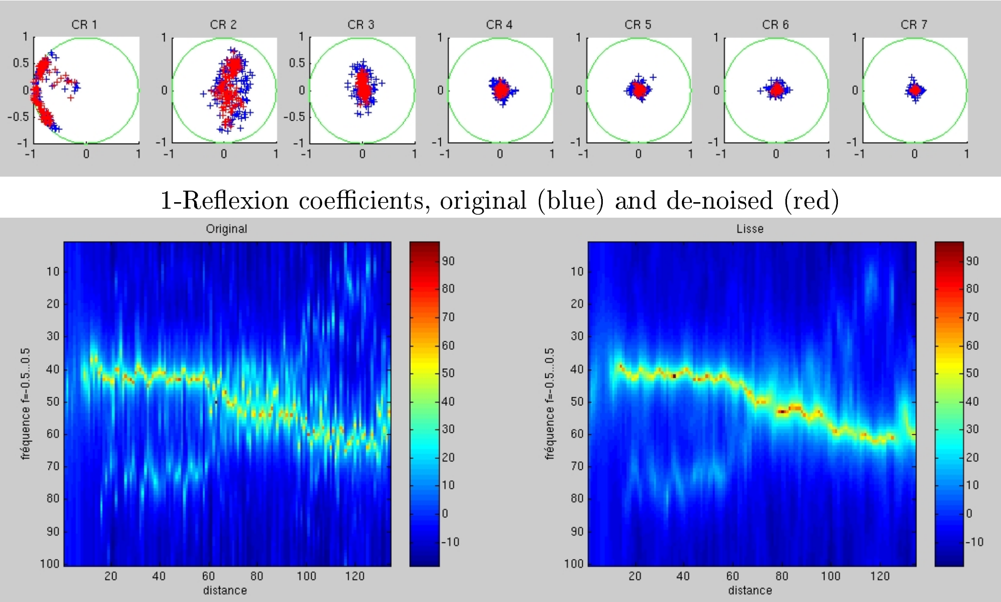
Figure 6: RSO Image : comparison of methods - details



1- The 7 reflexion coefficients for all measures along one azimuth on the complex unit disk
Blue: original coefficients. Red: de-noised coefficients.

2- Power spectral density in the plane range-frequency for one azimuth. Left: original, Right: de-noised
x-axis: range, y-axis: frequency $f = -0.5 \ldots 0.5$

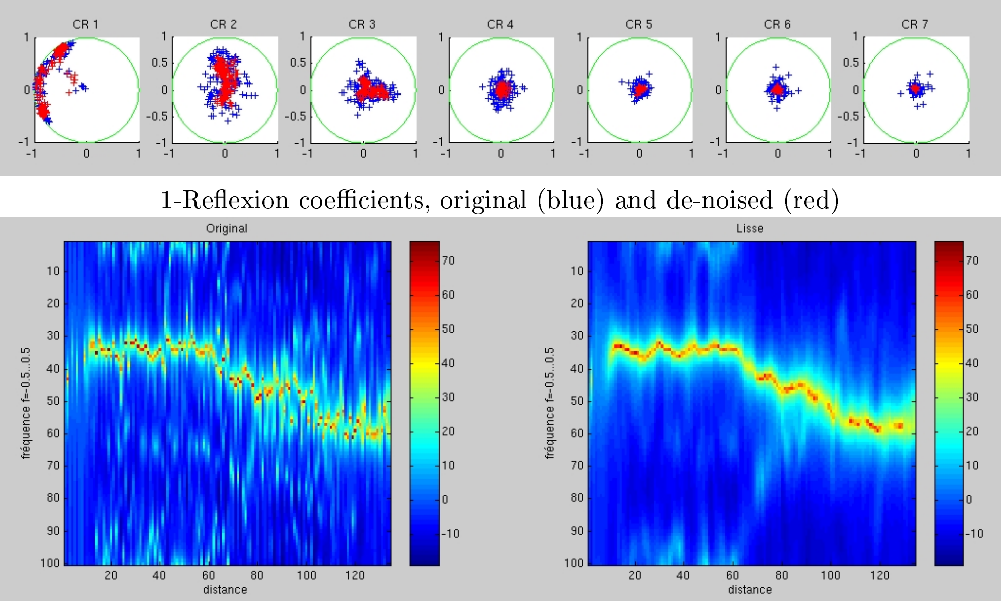
Figure 7: Denoising of simulated data for $\lambda = 1.0$ and $\gamma = 0.7$. Variance reduction : $\Delta Var = 96\%$



1-Reflexion coefficients, original (blue) and de-noised (red)

2-Power spectral density, original (left) and de-noised (right)

Figure 8: Results of the de-noising algorithm with regularisation for azimuth 1. $\lambda = 0.1$, $\gamma = 0.7$



1-Reflexion coefficients, original (blue) and de-noised (red)

2-Power spectral density, original (left) and de-noised (right)

Figure 9: Results of the de-noising algorithm with regularisation for azimuth 19. $\lambda = 0.1$, $\gamma = 0.7$