Adaptive Parameter Estimation for Satellite Image Deconvolution

André Jalobeanu, Laure Blanc-Féraud, Josiane Zerubia

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Adaptive parameter estimation for satellite image deconvolution

A. Jalobeanu — L. Blanc-Féraud — J. Zerubia

N° 3956

June 2000

THÈME 3

Rapport
de recherche

ISSN 0249-6399 ISRN INRIA/RR--3956--FR+ENG

INRIA
SOPHIA ANTIPOLIS

Adaptive parameter estimation for satellite image deconvolution

A. Jalobeanu , L. Blanc-Féraud , J. Zerubia

Thème 3 —Interaction homme-machine,
images, données, connaissances
Projet Ariana

Rapport de recherche n° 3956 —June 2000 —78 pages

Abstract: The deconvolution of blurred and noisy satellite images is an ill-posed inverse problem, which can be regularized within a Bayesian context by using an a priori model of the reconstructed solution. Homogeneous regularization models do not provide sufficiently satisfactory results, since real satellite data show spatially variant characteristics.

We propose here to use an inhomogeneous model, and we study different methods to estimate its space-variant parameters. The chosen estimator is the Maximum Likelihood (ML). We show that this estimator, when computed on the corrupted image, is not suitable for image deconvolution, because it is not robust to noise. Then we show that the estimation is correct only if it is made from the original image. Since this image is unknown, we need to compute an approximation of sufficiently good quality to provide useful estimation results.

Finally we detail an hybrid method used to estimate the space-variant parameters from an image deconvolved by a wavelet-based algorithm, in order to reconstruct the image. The obtained results simultaneously exhibit sharp edges, correctly restored textures and a high SNR in homogeneous areas, since the proposed technique adapts to the local characteristics of the data. A comparison with linear and non-linear concurrent algorithms is also presented to illustrate the efficiency of the proposed method.

Key-words: Deconvolution, Regularization, Hyperparameters, Inhomogeneous models, Complex Wavelet Packets, Markov Random Fields, Local estimation, Maximum Likelihood, Satellite images

Acknowledgements: The authors would like to thank the CNES (French Space Agency) for providing the image of Nîmes (SPOT 5 simulation), and Simon Wilson, from Trinity College Dublin (Ireland), for his kind remarks.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : 04 92 38 77 77 - International : +33 4 92 38 77 77 —Fax : 04 92 38 77 65 - International : +33 4 92 38 77 65

Estimation des paramètres adaptatifs pour la déconvolution d'images satellitaires

Résumé : La déconvolution des images satellitaires floues et bruitées est un problème inverse mal posé, qui peut être régularisé dans un cadre bayésien par l'utilisation d'un modèle a priori de la solution reconstruite. Les modèles de régularisation homogènes ne permettent pas d'obtenir des résultats parfaitement satisfaisants, car les images satellitaires ont des propriétés qui varient spatialement.

Nous proposons d'utiliser un modèle inhomogène, et nous étudions différentes méthodes permettant d'estimer les paramètres adaptatifs. L'estimateur que nous avons retenu est le maximum de vraisemblance (MV). Nous montrons que cet estimateur, lorsqu'il est calculé à partir de l'image dégradée, est inutilisable pour la déconvolution d'images, car il n'est pas robuste au bruit. Nous montrons ensuite que l'estimation n'est correcte que si elle est effectuée sur l'image originale. Comme cette image est inconnue, nous devons en calculer une approximation, dont la qualité doit être suffisante pour que les résultats de l'estimation soient utiles pour la restauration.

Nous détaillons finalement une méthode hybride, permettant d'estimer les paramètres adaptatifs à partir d'une image déconvoluée par un algorithme utilisant des ondelettes, afin de reconstruire l'image. Les résultats obtenus présentent à la fois des bords francs, des textures nettes, et un très bon rapport signal/bruit dans les zones homogènes, dans la mesure où la technique proposée s'adapte localement aux caractéristiques des données. Une comparaison avec des algorithmes concurrents linéaires et non linéaires est aussi effectuée, pour illustrer son efficacité.

Mots-clés : Déconvolution, Régularisation, Hyperparamètres, Modèles inhomogènes, Champs de Markov, Estimation locale, Maximum de vraisemblance, Images satellitaires

Remerciements : Les auteurs souhaitent remercier le CNES pour l'image de Nîmes (simulation SPOT 5), et Simon Wilson de Trinity College Dublin (Irlande), pour ses remarques pertinentes.

Contents

Notation

Images have $N_x \times N_y$ pixels.

X is a vector, made from the pixels of an image in a lexicographic order.
$X_{i,j}$ is the value of the pixel at column i and line j.
X^n is the n^{th} vector of the chain (X^n).

Let us write $P(X)$ instead of $P(X = x)$, where x is a random vector.

$\mathcal{X}$ is the original image, Y is the observed (corrupted) image.

Z_X, Z_Y and K are the partition functions related to the prior law $P(X)$, the posterior law $P(X \mid Y)$, and the likelihood $P(Y \mid X)$.

h is the Point Spread Function (PSF) and has the same dimension as X.

H is the convolution matrix (block-circulant matrix), and h is the generator of H.

B^x, B^y are vectors (same size as X).

D_x and D_y are the derivative operators applied on the images, w.r.t. columns and lines : $(D_x X)_{i,j} = X_{i+1,j} - X_{i,j}$, and are generated by d_x and d_y.

h^4 generates H^4, the convolution matrix applied on the $2N_x \times 2N_y$ symmetrized images. H^4 is obtained by extension of H.

$\mathcal{F}$ stands for the Fourier transform. DCT stands for the Discrete Cosine Transform.

Ω is the state space (set of $N_x \times N_y$ size images with real pixels), defined by
$\Omega = \mathbb{R}^{N_x\,N_y} \cap \{-\kappa \leq \sum_{i,j} x_{ij} \leq \kappa,\}$ where κ is a fixed bound (see appendix A for details).

Chapter 1

Satellite image deconvolution

1.1 Introduction to satellite image deconvolution

The problem presented here is the reconstruction of a satellite image from blurred and noisy data.

The degradation model is represented by the equation

$$Y = H\mathcal{X} + N \quad \text{where } H\mathcal{X} = h \star \mathcal{X} \tag{1.1}$$

where Y is the observed data, and $\mathcal{X}$ the original image. N is the additive noise and is supposed to be Gaussian, white and stationary. H is the convolution operator. The Point Spread Function (PSF) h is positive, and verifies the Shannon property.

We deal with a real satellite image deblurring problem, proposed by the French Space Agency (CNES). This problem is part of a simulation of the future SPOT 5 satellite. Both original and degraded images are provided by CNES. The noise standard deviation σ and the PSF h are also provided ($\sigma = 1.35$ for the example presented in chapter 4, see fig. 1.1 for h). In this case, h is symmetric and separable w.r.t. lines and columns, but the formalism presented in this report can be extended to the more general case where these properties are not satisfied.

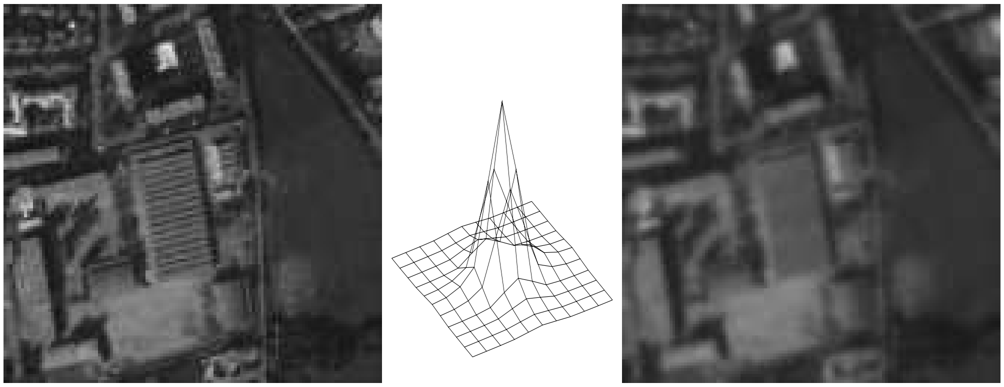


Figure 1.1: *left: original image $\mathcal{X}$ extracted from Nîmes ©CNES, center: PSF h (only a 11×11 area of nonnull coefficients is shown), right: observed image Y*

1.2 Regularization of an ill-posed problem

The deconvolution problem is ill-posed in the sense of Hadamard [18]. Knowing the degradation model is not sufficient to obtain satisfying results : it is necessary to regularize the solution by introducing *a priori* constraints [6, 10, 17]. When the blur operator and the variance of the white Gaussian noise are known, computing the Maximum Likelihood estimate (nonregularized solution) consists of minimizing an energy. The regularization constraint is expressed through a function added to this energy, which represents a roughness penalty on the solution.

This function could be quadratic (as suggested by Tikhonov in [40]) assuming that images are globally smooth, but it yields oversmooth solutions. A more efficient image model assumes that only homogeneous regions are smooth, and that edges must remain sharp. To get this *edge-preserving* regularization, we have used a *non-quadratic* φ-function in [23], called a *potential function*, as introduced in [8] and [15]. Properties of the φ-function have been studied in a variational approach in order to preserve the edges, avoiding noise amplification [1]. It is a symmetric, positive and increasing function, with a quadratic behaviour near 0 to isotropically smooth homogeneous areas, and linear or sub-linear behaviour near ∞ to preserve high gradients, i.e. edges (see fig. 1.2 for an example).

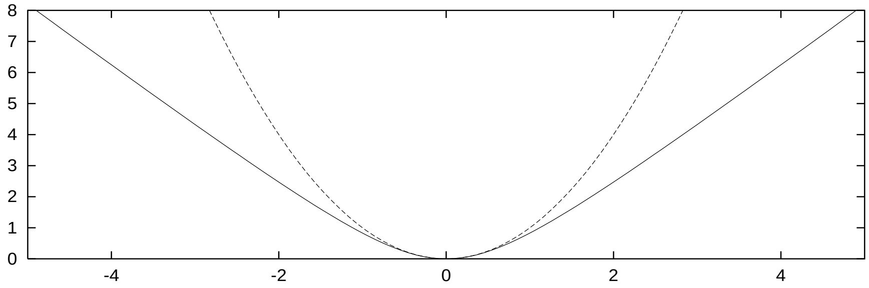


Figure 1.2: *Solid : Hyper Surface φ-function $\varphi(u) = 2\sqrt{1+u^2} - 2$, dashed : quadratic function $\varphi(u) = u^2$*

The regularized solution is computed by minimizing the energy :

$$U(X) = ||Y - HX||^2 / 2\sigma^2 + \Phi(X) \tag{1.2}$$

$$\text{where } \Phi(X) = \lambda^2 \sum_{ij} \left\{ \varphi\left(\frac{(D_x X)_{ij}}{\delta}\right) + \varphi\left(\frac{(D_y X)_{ij}}{\delta}\right) \right\} \tag{1.3}$$

where D_x and D_y are derivative operators previously defined 5. This non-quadratic variational regularization model involves two *homogeneous hyperparameters*, λ and δ [8].

It corresponds to the Maximum A Posteriori (MAP) estimate. The regularization constraint can be expressed as the prior distribution of the unknown image, within a Markov

Random Field (MRF) framework [4, 17] :

$$P(X) = Z_X^{-1} e^{-\Phi(X)} \tag{1.4}$$

while the data term $||Y - HX||^2/2\sigma^2$ corresponds to the likelihood of X, according to the statistics of the stationary Gaussian noise :

$$P(Y \mid X) = K^{-1} e^{-||Y-HX||^2/2\sigma^2} \tag{1.5}$$

Thus, the energy $U(X)$ is minimized within a variational framework. This is equivalent to maximizing the posterior distribution $P(X \mid Y)$, within a stochastic approach [21]. The posterior distribution is defined by :

$$P(X \mid Y) = Z_Y^{-1} e^{-||Y-HX||^2/2\sigma^2 - \Phi(X)} \tag{1.6}$$

The "hyper surface" convex function $\varphi(u) = 2\sqrt{1+u^2} - 2$ can be used, assuming the uniqueness of the solution. In this case, the restoration is made by a deterministic minimization algorithm [7].

1.3 Estimation of the homogeneous parameters

1.3.1 MCMC estimation for non-quadratic models

The visual quality of the reconstructed image strongly depends on the values of the hyperparameters (λ, δ) and so they must be accurately determined, excluding an empirical estimation. The parameter λ weights the regularization term versus the data term. Too high values of λ yield oversmooth solutions, and too small values lead to noisy images. The parameter δ is related to a threshold below which the gradients (due to the noise) are smoothed, and above which they are preserved. A high value of this threshold filters the edges as well as the noise, yielding over-regularized images. On the other hand, a small value of δ provides insufficient noise filtering.

We have proposed in [23] to use a stochastic approach, using a Markov Chain Monte Carlo (MCMC) technique, based on the *Maximum Likelihood Estimator* (MLE).

The MLE w.r.t. λ and δ is computed with the probability of the observed data :

$$(\hat{\lambda}, \hat{\delta}) = \arg\max_{\lambda,\delta} P(Y \mid \lambda, \delta) \tag{1.7}$$

To calculate (1.7), the joint distribution $P(X, Y)$ is integrated on X. Then Bayes law is used to reduce it to the prior and likelihood distributions w.r.t. X :

$$P(Y \mid \lambda, \delta) = \int_\Omega P(Y \mid X, \lambda, \delta)\, P(X \mid \lambda, \delta)\, dX \tag{1.8}$$

which leads to $P(Y \mid \lambda, \delta) = Z_Y \,/\, Z_X\, K$, where Z_X and Z_Y are respectively the partition functions related to the prior (1.4) and posterior (1.6) distributions of X. Most of the difficulties of the parameter estimation come from these functions, which are impossible to directly evaluate, because of the size of the state space Ω ($M^{N_x N_y}$ for images with M grey levels). We have optimized this criterion without explicitly computing Z_X and Z_Y, by evaluating its derivatives w.r.t. the hyperparameters. We refer to [22] for more detailed calculations.

We have used a gradient descent algorithm to optimize the log-likelihood. All the derivative evaluations w.r.t. the parameters $\theta = (\lambda, \delta)$ are done by estimating the expectations as follows :

$$\frac{\partial \log Z_X}{\partial \theta^i} = -E_{X \sim P(X)} \left[\frac{\partial \Phi(X)}{\partial \theta^i} \right] \tag{1.9}$$

giving a stochastic expression of the derivatives :

$$\frac{\partial \log P(Y \mid \theta)}{\partial \theta^i} = E_X \left[\frac{\partial \Phi}{\partial \theta^i} \right] - E_Y \left[\frac{\partial \Phi}{\partial \theta^i} \right] \tag{1.10}$$

where E_X and E_Y are respectively the expectations computed w.r.t. the prior and posterior laws. The expectations are approximated by empirical mean estimates, computed over n samples. We get :

$$\frac{\partial \log P(Y \mid \theta)}{\partial \theta^i} \simeq \frac{1}{n} \sum_{k=1}^{n} \frac{\partial \Phi(X_k \sim P(X))}{\partial \theta^i} - \frac{1}{n} \sum_{k=1}^{n} \frac{\partial \Phi(X_k \sim P(X \mid Y))}{\partial \theta^i} \tag{1.11}$$

This expression exhibits two types of expectation, one for the prior law $P(X)$, and the other for the posterior law $P(X \mid Y)$. So we sample from prior and posterior distributions to optimize the likelihood (1.7), by using a modified Geman & Yang sampler [16, 23].

Let us recall the MCMCML estimation algorithm presented in [22, 23]. It has been successfully applied on simulated blurred satellite data and real high resolution military images.

ALGORITHM 1.3.1 (MCMCML)

- *Initialization : The ratio λ/δ corresponds to the best Wiener filter [19] and $\delta \simeq 5\sigma$.*
- *Compute $\hat{X}$ with a deterministic algorithm, by minimizing $U(X)$ (1.2) with θ_n.*
- *Compute $E_X[\,]$ and $E_Y[\,]$ with θ_n : Generate 2 Markov Chains (prior and posterior distributions), then compute the expectations needed in (1.11).*
- *Iteration from θ_n to θ_{n+1} : $\theta^i_{n+1} = \theta^i_n - \alpha \frac{\partial \log P(Y \mid \theta)/\partial \theta^i}{\partial^2 \log P(Y \mid \theta)/\partial (\theta^i)^2}$ (where $\alpha < 1$)
The second derivative is numerically approximated.*
- *Stopping criterion : we stop the algorithm if $\frac{|\partial \log P(Y \mid \theta)/\partial \theta^i|}{\theta^i} < \epsilon$.*

1.3.2 Estimation for a Gaussian model

Using a quadratic regularization model (i.e. $\varphi(u) = u^2$) yields oversmooth solutions by filtering the edges as well as the noise (see fig. 1.3 for a comparison between quadratic and non-quadratic regularization).

$$\Phi_g(X) = b \sum_{ij} \left\{ (D_x X)^2_{ij} + (D_y X)^2_{ij} \right\} \tag{1.12}$$

where b is the homogeneous parameter (equivalent to λ^2 in (1.3)).

Compared to non-quadratic ones, such a model has the advantage that it provides full Gaussian prior and posterior laws :

$$\begin{array}{rcl} P(X) & = & Z_X^{-1} e^{-b\,(||D_x X||^2 + ||D_y X||^2)} \\ P(X\,|\,Y) & = & Z_Y^{-1} e^{-||Y-HX||^2/2\sigma^2 - b\,(||D_x X||^2 + ||D_y X||^2)} \end{array} \tag{1.13}$$

Then the normalizing functions Z_X and Z_Y can be computed :

$$Z_Y = e^{-U(\hat{X})} \sqrt{\frac{(2\pi)^{N_x N_y}}{\det \Sigma_Y^{-1}}} \quad \text{and} \quad Z_X = 2\kappa \sqrt{\frac{(2\pi)^{N_x N_y - 1}}{\det' \Sigma_X^{-1}}}, \tag{1.14}$$

see appendix A for details and demonstration. The constant κ is a bound related to the state space Ω, and $\det'$ is the product of all non-null eigenvalues. $\hat{X}$ is the restored image with the current parameters and $U(\hat{X})$ is its energy, with Φ defined in (1.2). Σ_X and Σ_Y are the covariance matrices of the laws $P(X)$ and $P(X\,|\,Y)$:

$$\begin{array}{rcl} \Sigma_X^{-1} & = & 2b(D_x^t D_x + D_y^t D_y) \\ \Sigma_Y^{-1} & = & H^t H/\sigma^2 + 2b\,(D_x^t D_x + D_y^t D_y) \end{array} \tag{1.15}$$

The quadratic forms Σ_X and Σ_Y are diagonalized by a Fourier transform, because they are defined by block-circulant matrices, since we consider that images are extrapolated by periodization. Indeed, derivative and convolution operators are block-circulant. Therefore, the above determinants can be exactly evaluated (see appendix A).

Thus, each step of the previously presented descent method is deterministic (because expectations are given by analytical expressions) and reduces to a sum over pre-computed quantities. The log-likelihood gradient can be computed exactly, therefore no MCMC method is needed anymore. The computation is fast, because the only limiting factor is the reconstruction algorithm, which is based on a deterministic descent algorithm due to the convexity of the criterion to be optimized.

1.3.3 A Gaussian approximation of non-quadratic models

We can significantly optimize the MCMCML algorithm by approximating the prior and posterior densities when φ is non-quadratic, with carefully chosen Gaussian distributions

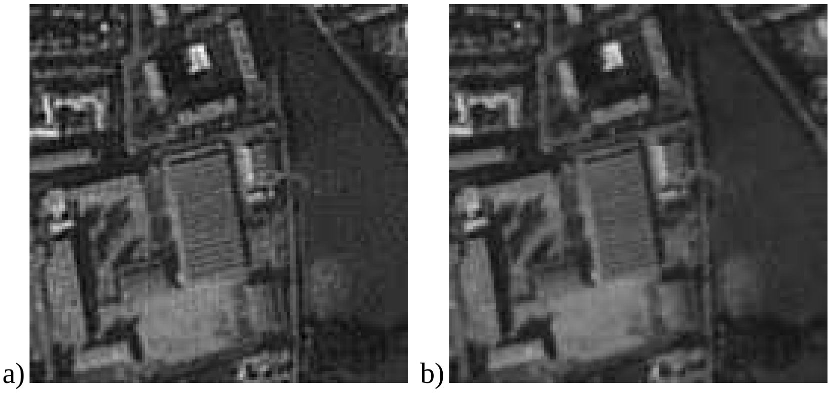


Figure 1.3: *Restored image of fig. 1.1 :a) with homogeneous Gaussian prior and estimated* $b = 0.0005$*, b) with non-quadratic* φ*-function based prior and estimated* $\lambda = 0.5$ *and* $\delta = 10$

(see fig. 1.4). By studying the behaviour of the function $e^{-\lambda^2 \varphi(u/\delta)}$, where u is a difference between pixels, we remark that it can be approximated by $e^{-\alpha u^2}$, whose variance $\frac{1}{2\alpha^2}$ depends on the λ and δ values. α is evaluated by numerical studies.

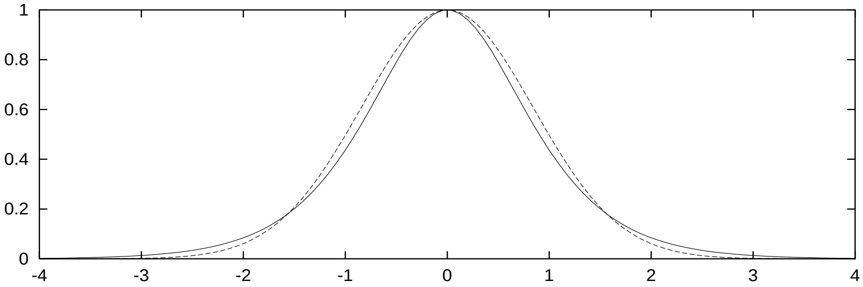


Figure 1.4: *Gaussian approximation of* $e^{-\lambda^2 \varphi(u/\delta)}$ *(solid) by* $e^{-\alpha u^2}$ *(dashed), with* $\lambda = \delta = 1$.

Several methods are available to calculate α, for instance: either the integral or the variance of the functions $e^{-\lambda^2 \varphi(u/\delta)}$ and $\frac{1}{2\alpha^2}$ are constrained to be equal. From numerical results, we propose the following relation, where $K \simeq 6$:

$$\alpha = \frac{\lambda^2}{\delta^2} \left(\frac{1}{1 + \frac{1}{K\lambda^2}} \right) \tag{1.16}$$

The proposed optimization consists of three steps :

- Estimate the parameter α for the quadratic model ;
- Convert α to λ by inverting equation (1.16), with a fixed $\delta = 5\sigma$;
- Reconstruct the image with the new (λ, δ) and a non-quadratic φ-function.

Images reconstructed with parameters estimated this way are very satisfying, as their SNR is maximum 0.1 dB (0.5%) lower compared to the images reconstructed with the MCMC algorithm presented in the previous section. The computation time of the estimation step is negligible compared to the stochastic method.

1.4 Insufficiency of homogeneous models

In practice, to optimize the computing time, the estimation is made on a small area extracted from the image. Indeed, as the model is homogeneous, we assume the same (λ, δ) is convenient for the entire (large size) image. This enables a faster estimation, because sampler speed is conditioned by a Cosine Transform.

In fact, textured areas and homogeneous areas correspond to different estimated values of λ (see fig. 1.5), so they would be better reconstructed if processed separately. Real images cannot be efficiently represented by homogeneous models, since they are made from various textures, sharp edges and constant areas. So the prior parameters (λ, δ) should adapt to the local structure of the image in order to enable a better reconstruction (i.e. less noisy in homogeneous regions, and with sharper details in other areas).

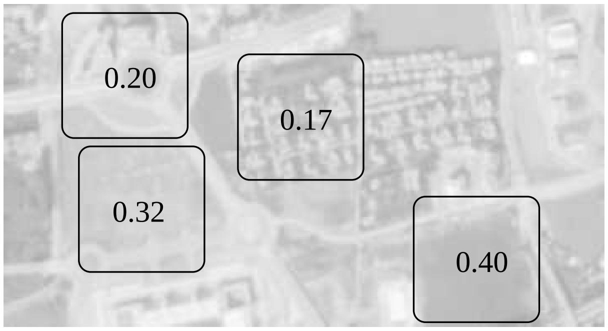


Figure 1.5: *Different estimated values of λ correponding to different areas on the same image, extracted from Nîmes © CNES*

Chapter 2

Adaptive parameter estimation

2.1 Towards adaptive regularization models

2.1.1 An inhomogeneous Gaussian model

A possible choice for the prior model is the Gaussian Markov Random Field (GMRF), corresponding to a quadratic φ-function in equation (1.3). To avoid obtaining globally oversmooth or noisy solutions, different λ parameters have to be used for different regions. As suggested in [41], the deconvolution results can be improved by partitioning the image into regions with constant parameters. But it is difficult to accurately determine such an image segmentation (i.e. to split this image into homogeneous and texture areas) because the provided data are blurred and noisy. So we prefer to define a model with a different parameter for each gradient of the image. A few authors have already studied such inhomogeneous MRF models (IGMRF), see [2, 3, 38] for instance.

The regularizing function Φ corresponding to IGMRF is then defined as :

$$\Phi(X) = \sum_{ij} \left\{ b^x_{ij} (D_x X)^2_{ij} + b^y_{ij} (D_y X)^2_{ij} \right\} \tag{2.1}$$

where b^x_{ij} and b^y_{ij} are the inhomogeneous parameters, w.r.t. columns and lines. The variables b^x_{ij} and b^y_{ij} are analogous to continuous line process [15, 17], as low values of b^x_{ij} or b^y_{ij} correspond to an edge located between two pixels.

Homogeneous GMRFs generally apply a too high penalty to edges, because the square differences of pixel values have the same weight λ^2/δ^2. The inhomogeneous model defined by (2.1) enables edge preserving if the b values related to high pixel differences are small compared to those related to constant areas.

The choice of the IGMRF is motivated by the half-quadratic expansion used in [8, 15] to optimize the homogeneous criterion (1.2). To get a quadratic form, φ is expressed as :

$$\varphi(u) = \inf_{b \in [0,1]} [bu^2 + \psi(b)]$$

Alternate minimizations are done w.r.t. X and B to optimize the criterion with a regularizing function $\Phi^\#$ rewritten as :

$$\Phi^\#(X, B^x, B^y) = \lambda^2 \sum_{ij} \left\{ B^x_{ij} (D_x X)^2_{ij}/\delta^2 + \psi(B^x_{ij}) + B^y_{ij} (D_y X)^2_{ij}/\delta^2 + \psi(B^y_{ij}) \right\}$$

This defines a convergent algorithm for a convex φ function. At the end, the B^x and B^y fields are related to the edges of the image (a small value corresponds to an edge), and the restored image corresponds to the previously defined inhomogeneous regularizing model (2.1), with $b_{ij} = (\lambda^2/\delta^2)B_{ij}$ for each direction. Thus, estimating (λ, δ) and using a φ-function in the regularization model as in [23] leads to a b field, which gives quite good reconstruction results, as regards edges, excepted that textures are generally oversmoothed, and homogeneous areas remain somewhat noisy. The parameter choice, either automatic or not, consists in a compromise between the correct restoration of the details and of constant areas. The model is in practice not sufficiently adaptive, even if it provides anisotropic smoothing (which adapts to the direction of the edges).

We have seen previously that reducing the size of the estimation window enables to fit better to the local characteristics of the image (see fig. 1.5). It gives, for example, larger values of b in homogeneous areas, providing better noise filtering.

So, we will use an IGMRF as a prior distribution of the unknown image, as it seems to give a sufficiently good representation of satellite images.

The difficulty of such models is the estimation of the b parameter values, providing at the same time noise filtering within constant areas and detail preserving in other areas. Therefore, the space varying parameters have to adapt to the local characteristics of the image. As the number of parameters to estimate is twice the number of pixels in the image, the estimation problem remains complex. We will study in section 2.2 the choice of the estimator, and we will discuss the difficulties of such an estimation when the original image is unknown.

2.1.2 An inhomogeneous Generalized Gaussian model

One could extend the previous model by using in the regularizing function an arbitrary power p instead of the square of the pixel differences :

$$\Phi_{gg}(X) = \sum_{ij} \left\{ b^x_{ij} |(D_x X)_{ij}|^p + b^y_{ij} |(D_y X)_{ij}|^p \right\} \tag{2.2}$$

This defines an Inhomogeneous Generalized Gaussian Markov Random Field (IGGMRF) model, as proposed in [5, 39]. It is a more general class of models which includes the Gaussian model (p=2).

IGGMRFs have been used in [38] within a multiresolution framework. We will discuss later the estimation method presented in this work. The corresponding estimation scheme seems to give better results with a fixed value of p close to 1, than with a Gaussian model.

We did not choose such a model, because our estimation method, which will be detailed in section 2.3.2, seems not to give sufficient SNR improvement by setting a p value different from 2. Non-Gaussian models lead to a non-quadratic functional to be optimized, which can be achieved by a half-quadratic method, but whose increase in computational complexity is probably not justified by the enhancement of the results.

2.1.3 Other possible regularization models

We could construct many other adaptive regularization models, based on the homogeneous models found in the literature (see [7] for a review), by replacing the homogeneous parameters by a different parameter for each pixel difference. A possible model can be intuitively derived from the φ-function regularizing model (non-quadratic), leading to the following expression :

$$\Phi_2(X) = \sum_{ij} \left\{ (\lambda^x)^2_{ij}\, \varphi\left(\frac{(D_x X)_{ij}}{\delta^x_{ij}}\right) + (\lambda^y)^2_{ij}\, \varphi\left(\frac{(D_y X)_{ij}}{\delta^y_{ij}}\right) \right\} \tag{2.3}$$

The main difficulty of estimating the parameters of this model is the multiplicity of solutions, as there are two parameters for each pixel difference (λ and δ).

It has been experimentally shown [22, 23] that an infinity of optimal couples (λ, δ) exist in the homogeneous case. It could also be true in the adaptive case, which could considerably complicate the estimation problem.

The model related to (2.3), which provides both adaptive regularization through λ and adaptive model selection through δ, seems to be too computationally heavy, both for estimation and reconstruction. The gain in SNR does probably not justify this increase of complexity.

2.2 Choice of an estimator

How to determine the optimal parameters to obtain the best reconstructed image? The chosen estimator has to be adapted to the local statistical characteristics of the data, while remaining robust to noise. In this section, we detail some classical estimators, used for both homogeneous and adaptive models. Then, in next section, we discuss the difficulty of implementing such estimators in the inhomogeneous case. After evaluating the results, we finally propose in section 2.3 a new estimation method, fully adaptive, using experimental results from original satellite images.

There are three essential kinds of statistical estimators which have been applied on parameter estimation problems. Let θ represent all the parameters.

- $P(Y \mid \theta)$: MLE on Y, the observed data [42]. This estimator has been successfully applied to the case of homogeneous models involving φ-functions, by using a MCMC method [23]. It needs sampling from both prior and posterior laws.

- $P(X, Y \mid \theta)$: MLE for the joint law of X and Y [31]. Only an approximate optimization method can be used in practice, consisting of alternate optimizations w.r.t. X and θ. Optimizing w.r.t. X is equivalent to the MAP criterion. Optimizing w.r.t. θ consists of computing the MLE $P(\hat{X} \mid \theta)$ on the complete data $\hat{X}$, where $\hat{X}$ is the image restored with the current value of $\hat{\theta}$. It is a suboptimal method, and the convergence is not sure. The advantage of this method is that only prior sampling is required, which substantially reduces the computation time in the case of deblurring.

- $P(\mathcal{X} \mid \theta)$: MLE computed on the original image. The problem is that the image $\mathcal{X}$ is unknown. It cannot be approximated by its degraded version Y, because Y is blurred and noisy, so it has different properties. Parameters estimated on a blurred image have a too high value, and therefore lead to oversmooth solutions. Parameters estimated on a noisy image are too low, and provide insufficient regularization, leading to noisy solutions. This estimator is only significant in the case of complete data, and it supplies good parameter estimates for image segmentation [11]. As the previous method, it only requires sampling from the prior law.

In the case of one global parameter, the first estimator gives the best results. Both prior and posterior sampling are achieved in the frequency space by diagonalizing the covariance matrix, by using a half-quadratic expansion of φ [23].

The two other estimators are easier to implement, as there is no need for posterior sampling. It is possible to use a table of precomputed values of prior expectations (and to interpolate them to get an approximation for any parameter value) to speed up the algorithm. The problem raised by the second method is that estimation takes not really into account the data image, but a restored version of it, whose shape strongly depends on the current values of the parameters. Thus, a bad initialization often leads to degenerated solutions.

The last estimator $P(\mathcal{X} \mid \theta)$ is the fastest one, but needs an image close to the orignial one, $\mathcal{X}$, which is unknown. Therefore, an approximation of $\mathcal{X}$ has to be accurately determined, for example by a non-parametric reconstruction algorithm, if we want the parameters obtained this way to be significant for regularization. We try to determine such an approximation in the next chapters.

In the following section, we detail how to estimate the inhomogeneous parameters from complete data. This is a very efficient method, preferable to other ones because of its simplicity and its accuracy. Then, in the next sections, we show that the only correct estimator is this one, by mentioning the difficulties related to the computation and the variance of the MLE on the observed data.

2.3 Estimation from the original image

2.3.1 The Maximum Likelihood Estimator on complete data

Since the estimation from the noisy and blurred data Y provides too noisy parameters within an IGMRF framework (this will be discussed in section 2.5), let us examine the case of the parameters directly estimated from the original image $\mathcal{X}$. Suppose this image is known. We have to test if the parameters estimated this way provide a good regularization when applied to deconvolution, removing the noise while keeping a maximum number of details.

However, only an approximate version of the original image is known (obtained with a deconvolution algorithm). In section 2.4, we study the estimation error of such a method and its effects.

The MLE w.r.t. the image $\mathcal{X}$ related to the prior distribution is :

$$\hat{b}_{ij}^{x,y} = \arg\max_{b_{ij}^{x,y}} \Big[\log P(X \mid \{b^x\}, \{b^y\}) \Big] = \arg\max_{b_{ij}^{x,y}} \Big[-\log Z_X - \Phi(\mathcal{X}) \Big] \tag{2.4}$$

The log-likelihood derivatives are :

$$\frac{\partial \log P(X \mid \{b^x\}, \{b^y\})}{\partial b_{ij}^{x,y}} = E_X \left[(D_{x,y} X)_{ij}^2 \right] - (D_{x,y}\mathcal{X})_{ij}^2 \tag{2.5}$$

The estimation problem consists of solving the nonlinear system :

$$\left\{ E_X \left[(D_{x,y} X)_{ij}^2 \right] = (D_{x,y}\mathcal{X})_{ij}^2 \right\} \tag{2.6}$$

The complete data formulation simplifies the estimation problem, since the expectation term only depends on the parameters and the other terms only depends on $\mathcal{X}$. To compute one step of the descent method to perform the estimation, it is sufficient to compute the variance of each pixel difference w.r.t. the prior law (the mean is null). This can be achieved by sampling from this law with an optimized Gibbs sampler as presented in appendix C, or by finding an approximate analytic expression of this variance.

An approximation of the MLE

We propose to use the simplest approximation of the local variance, which is not accurate, but provides good results, as shown on figure 2.2. It consists of considering that the variance of a gradient is only given by the corresponding local b. Thus, we suppose that the variance of a gradient $(D_{x,y}X)_{ij}$ is equal to the variance of the same gradient in the homogeneous case, i.e. when all the parameters are equal to the corresponding $b_{ij}^{x,y}$. Then we have to compute this variance for a homogeneous GMRF.

As is demonstrated in appendix A, the partition function of the homogeneous prior distribution (with parameter b) is :

$$Z_X = 2\kappa \sqrt{\frac{(2\pi)^{N_x N_y - 1}}{\prod_{(i,j)\neq(0,0)} 2b\, c_{ij}}} \tag{2.7}$$

(see appendix A for the definitions of κ and c_{ij}). We have already seen in equation (1.9) that

$$-\frac{\partial \log Z_X}{\partial b} = E_X \left[\frac{\partial \Phi_g(X)}{\partial b} \right] = E_X \left[\sum_{ij} \left\{ (D_x X)_{ij}^2 + (D_y X)_{ij}^2 \right\} \right] \tag{2.8}$$

if Φ_g is defined by (1.12). The sum and the expectation can be swapped, and the variances $E_X[(D_{x,y}X)^2_{ij}]$ are all equal to v since the model is homogeneous. Then the expectation is equal to $2N_xN_y\,v$. We also remark that Z_X is proportional to $(\sqrt{b})^{-(N_xN_y-1)}$, so that its log derivative is equal to $-\frac{N_xN_y-1}{2b}$. Since $N_xN_y \gg 1$, equation (2.8) is finally simplified and gives the expression of the variance v :

$$v = E_X[(D_{x,y}X)^2_{ij}] \simeq \frac{1}{4b} \tag{2.9}$$

We will also suppose the local variance to be equal to $1/4b$, which gives :

$$\hat{b}^{x,y}_{ij} \simeq \frac{1}{4(D_{x,y}X)^2_{ij}} \tag{2.10}$$

Experimental studies have shown that this expression gives the same estimation results as the inhomogeneous MLE, computed using a Gibbs sampler as described in appendix C, and a Newton-Raphson descent algorithm.

Within a Bayesian framework, the deconvolution achieved by computing the MAP estimate is correct only if the chosen prior law accurately models the unknown image. The adaptive Gaussian law seems to fit to satellite images. Therefore the complete data MLE provides efficient adaptive parameters for image deconvolution, when the original image is known. Furthermore, this estimator is efficiently approximated by a simple expression (2.10).

In the following section, we use this type of estimator and we try to find the optimal weighting to apply to the squared pixel differences of the regularizing term Φ.

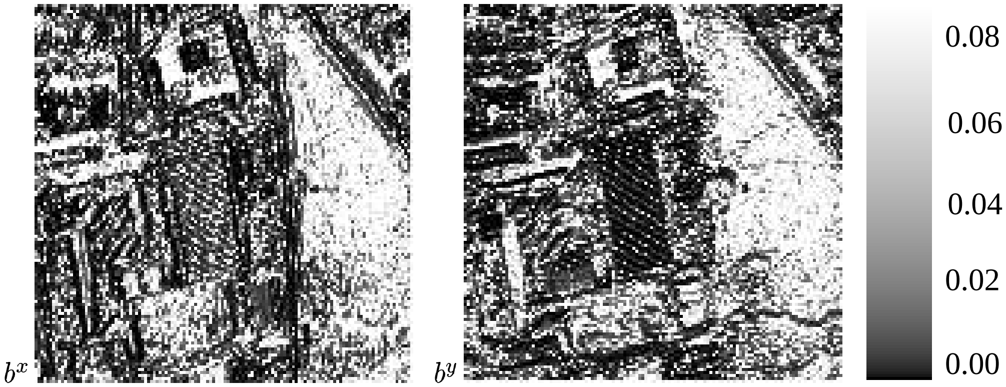


Figure 2.1: *Parameter fields b^x and b^y estimated with MLE on complete data $\mathcal{X}$*

Comparison with other approximations

In [37], another approximation is proposed to estimate the inhomogeneous parameters within a Generalized Gaussian distribution context. By taking the case of a Gaussian law, and after some simplifications, we can write :

$$(\hat{b}^x_{ij})^{-1} \simeq \frac{1}{2} \sum_{k,l \in V(i,j)} (X_{ij} - X_{kl})^2 + \frac{1}{2} \sum_{k,l \in V(i+1,j)} (X_{ij} - X_{kl})^2 \tag{2.11}$$

where $V(i,j)$ represents the 4 nearest neighbor pixels of (i,j). This intuitive estimate comes from the MLE in the homogeneous case :

$$\hat{b} = \left[\frac{2}{N_x N_y} \sum_{i,j} \left((D_x X)^2_{ij} + (D_y X)^2_{ij} \right) \right]^{-1} \tag{2.12}$$

Here, the local pixel differences are averaged over the entire image. So a local estimate could be given by averaging in the neighbourhood of the concerned pixels only.

This method does not give satisfactory results for satellite image deconvolution (see fig. 2.2). For the tests, a representative area has been extracted from a large size image. This area exhibits textures, constant areas and edges. The homogeneous parts are efficiently denoised, but the edges and the textures remain too noisy. This is linked to the average operator of equation (2.11), which associates to $(\hat{b}^x_{ij})^{-1}$ a smoothed version of the data X. It gives low regularizing parameters w.r.t. both directions x and y even if there is an edge w.r.t. a single direction. Furthermore, the noise is not smoothed enough along the edge. We should expect the b corresponding to a vertical or horizontal edge to be nearly equal to zero, which is impossible because of this smoothing.

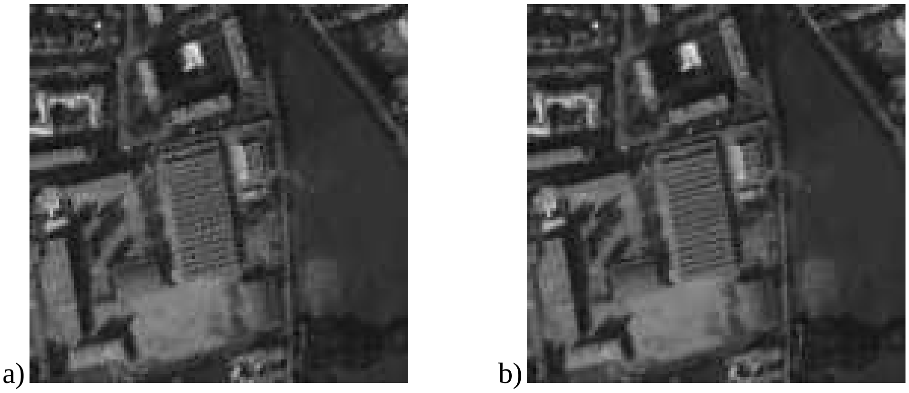


Figure 2.2: *Restored image of figure 1.1 with IGMRF and parameters estimated from complete data $\mathcal{X}$: a) using the nearest neighbour approximation (2.11) (SNR=24.8 dB), b) using the single gradient approximation (2.10) (SNR=28.8 dB) - the same results are obtained with an MCMC method*

2.3.2 Optimal weighting

The expression (2.10) could be generalized, by changing the factor 4 or the exponent of the gradient. We have studied the following estimator :

$$\hat{b}_{ij}^{x,y} \simeq \lambda \, (D_{x,y}X)_{ij}^{-p} \tag{2.13}$$

with various λ and p to find the optimal values of these new parameters (in the sense of a maximum SNR).

Experiments have shown that the best exponent is $p = 2$, since the results are not very sensitive to p and it is equivalent to MLE for homogeneous models. But the value of λ depends on the image and has to be estimated as a homogeneous parameter. We have :

$$\hat{b}_{ij}^{x,y} = \frac{\lambda}{1 + (D_{x,y}X)_{ij}^2} \tag{2.14}$$

The one added to the squared gradient ensures that the value of b remains in $[0, \lambda]$ and enables to return to a φ-model as explained below.

To estimate λ, we should return to a classical φ-model defined by (1.3), with $\delta = 1$. Consider that the result $\hat{X}$ reconstructed from Y is nearly equal to the original image. In this case, at the end of the restoration algorithm, we have $b = \lambda\varphi'(u)/2u$ for each gradient u of the solution, and this solution is obtained by minimizing the energy Φ with these fixed values of b.

It is equivalent to use the IGMRF regularizing model with the same values of b. The expression (2.14) can be rewritten as $\hat{b} = \varphi'(u)/2u$ with a φ function having the right properties. The function $\varphi(u) = log(1 + u^2)$ fits to the problem.

The estimation of λ can be made by algorithm 1.3.1, for various images. Then, instead of using the solution of the homogeneous nonlinear regularizing algorithm to compute the values of b, we use the original image $\mathcal{X}$. Thus, we use equation (2.14) with an estimated value of λ. However, the estimated values of λ have not provided better reconstruction results than $\lambda = \frac{1}{4}$. Even if this estimation is made adaptive, it does not improve the quality of the results.

2.4 Robustness w.r.t. approximations

We keep the complete data MLE in the rest of the work and we try to determine its accuracy when the original image is only known through an approximation.

We have to check the sensitivity of the complete data MLE to the variations of the pixel differences of the image X used for estimation. We first focus on the robustness to the noise, then we study the effect of the blur. Indeed, the true image is not provided and we have to test if it is possible to obtain correct parameter values in this more realistic case.

Let Δg be the gradient error, i.e. the gradient of the residual noise (difference between the approximated and the true images). This random variable is not necessary Gaussian. We also define the associated relative error of the gradients, $\Delta g/g$.

Let us define the function $b(g) = 1/4g^2$ which gives the MLE of b for a given pixel difference value g. The error Δg induces an estimation error Δb. To compute this error, let us take the Taylor series expansion of the function b around g_0 (gradient computed from the original image). We have :

$$b(g_0 + \Delta g) \simeq b(g_0) + \Delta g \frac{\partial b(g_0)}{\partial g} \tag{2.15}$$

hence $\Delta b = b(g_0 + \Delta g) - b(g_0) \simeq -\Delta g/2g^3$, therefore the relative errors are linked by $\Delta b/b \sim \Delta g/g$.

The variance of b is then given by :

$$\text{var}(b) = E\left[(\Delta b))^2\right] \simeq 4\, b_0 \left(\frac{\Delta g}{g}\right)^2 \tag{2.16}$$

2.4.1 Robustness w.r.t. noise

The relative fluctuations $\Delta b/b$ can become very high for small values of g, because they are of the same order as $\Delta g/g$. It corresponds to constant areas. If some noise is present on the approximation image, it induces an under-estimation of b in these areas, and finally an insufficient regularization. It means that the provided approximation X of $\mathcal{X}$ has to be very smooth in these regions.

To get a robust estimate, it is necessary to evaluate the magnitude of the error Δg. This quantity depends on the method chosen to approximate $\mathcal{X}$ (here, a wavelet-based deconvolution algorithm), it will be computed in chapter 4. We set to zero all the gradients below $|\Delta g|$ to ensure a maximum regularization in constant areas.

2.4.2 The case of blurred edges

On the other hand, for high values of the gradient g, the estimator becomes sufficiently accurate since $\Delta g/g$ remains small. Consequently, the edges are detected and correctly located, as they usually correspond to values of g from 30 to 255.

How sensitive is the MLE to the blur? If an edge is blurred, the parameters have to adapt to enable the reconstruction of a sharp edge, by smoothing the noise only along the edge.

Near an edge, i.e. for a high value of g, the blur induces an error Δg, but the relative variation $\Delta g/g$ remains small. Therefore, the estimated value of b remains correct. But if the extent of the blur is large, an edge can affect the gradient values at a long distance, leading to under-estimated parameter values. Does this really mean insufficient regularization in the whole neighborhood? The answer is no, because the blur does not affect the pixel differences taken in a direction perpendicular to the edge. The gradients in this direction remain low, and provide high b values to guarantee efficient noise filtering.

This can be true only if the edges are well-defined (boundary between two constant areas). However, it is not valid for textures and small details, because the directions they define are scrambled by the blur. Therefore, the approximation has to be as good as possible for these features to allow a correct estimation. That is why the parameters cannot be evaluated from the observed image Y, as proposed in [31] using a suboptimal algorithm.

2.4.3 Conclusion

Finally, we find that the optimal estimator for inhomogeneous Gaussian models is the MLE related to the *prior model*, computed using an approximation of the true image. This method is robust if the noise remains small in constant areas, and can accept some blur on the edges. This is the only ML method which is both robust to degraded observations and computationally efficient.

2.5 Difficulties of adaptive MLE on observed data

The regularization model which is used in the following is an inhomogeneous Gaussian model, whose regularizing function is given by equation (2.1).

If we want to estimate inhomogeneous regularization parameters with a method similar to algorithm 1.3.1, we have to address some problems raised by adaptive model sampling.

2.5.1 Computational complexity

Necessity of using classical samplers

We first consider the MLE computed on Y. Sampling from the posterior distribution is intractable when using classical samplers as Gibbs [17] or Metropolis [33]. This distribution is expressed as :

$$P(X \mid Y, \{b^x\}, \{b^y\}) = Z_Y^{-1} e^{-||Y-HX||^2/2\sigma^2 - \sum_{ij} \left\{ b^x_{ij} (D_x X)^2_{ij} + b^y_{ij} (D_y X)^2_{ij} \right\}} \tag{2.17}$$

The diagonalization of the covariance matrix of this distribution in the Fourier basis is no more possible, because the parameters b^x and b^y are space varying. Therefore, a classical sampler has to be used to explore the posterior and prior laws.

The low order neighborhood of the prior Gaussian MRF fits well to Gibbs sampler [17]. The local conditional distribution $P(X_{x,y} \mid X_{i \neq x, j \neq y}, \{b^x\}, \{b^y\})$ for the site (x, y) can be computed, taking advantage of the Gaussian form of the probability (see appendix C.1). The estimation method based on such a sampling scheme has been detailed in section 2.3.1.

The posterior Gaussian MRF takes into account the data, through the blur operator H, inducing a high order neighborhood (for example, the size of the convolution kernel of SPOT 5 simulations is 11 × 11). Sampling becomes really intractable by means of classical samplers. It is possible to approximate the local distribution in order to take into account the data, as seen in appendix C.2.

Number of samples

How many samples are needed to compute the expectations to evaluate the derivatives of the likelihood? Equation (1.11) has to be rewritten for the IGMRF (2.1) :

$$\frac{\partial \log P(Y \mid \{b^x\}, \{b^y\})}{\partial b_{ij}^{x,y}} = \frac{1}{n}\sum_{k=1}^{n}\Big[D_{x,y}(X_k \sim P(X))\Big]_{ij}^2 - \frac{1}{n}\sum_{k=1}^{n}\Big[D_{x,y}(X_k \sim P(X \mid Y))\Big]_{ij}^2 \tag{2.18}$$

The gradients $(D_{x,y}X)_{ij}$ of the samples X are Gaussian variables, and the above expectations represent their variance. If we consider the prior distribution, which is zero-mean, the relative fluctuation of the variance estimator is equal to $\sqrt{2/n}$, where n is the number of samples (the samples are supposed to be independent).

The number of needed samples depend on the expectations accuracy $\Delta E/E$, which has a fixed upper bound ϵ to ensure the convergence of the gradient algorithm. Descent methods which use (2.18) to optimize the parameters are quite sensitive to the noise. An acceptable value for ϵ, coming from experiments, is 1%. We experimentally show that the resulting accuracy of estimated parameter values is sufficient to give good quality reconstructed images.

So we have :

$$\frac{\Delta E}{E} = \sqrt{\frac{2}{n}} < \epsilon = 0.01 \tag{2.19}$$

which gives $n > 20,000$.

This is the minimum number of samples. As a sample is extracted from a Markov Chain, it is not independent from the previous one. It means that in practice a larger number of samples is needed to reduce the fluctuations and to obtain the right accuracy.

This number has to be compared with the number of pixel differences averaged to estimate an expectation for an homogeneous φ model like the one presented in the first chapter. Usually a 64×64 sample size is taken to optimize the computing time, therefore a sum is computed on about 4000 pixel differences for each sample (see equation (1.11)). As only half of these random variables are independent, we take into account only 2000 of them to compute the accuracy. To reach the wanted 1%, about 10 to 15 samples (each containing 2000 independent variables) are sufficient, this verifies equation (2.19), and is compatible with observations.

For an inhomogeneous model, as seen previously, a single pixel difference has to be averaged over many samples to estimate each parameter. Thus, for a 64×64 image, the computation of expectations needs about 2000 times more iterations for each sampler! For a real satellite image, whose minimum size is 3000×3000 pixels, the estimation step becomes more than $3,000,000$ times longer compared to the homogeneous equivalent model.

Using Gaussian MRF enables to speed up the sampler by a factor 4 (30 operations per pixel for each iteration for IGMRF versus 120 for homogeneous φ model). In this case, the local conditional distribution is a Gaussian whose mean and variance are easy to compute (see appendix C for a detailed algorithm). But the inhomogeneous estimation step remains about $25,000$ times longer.

So, even in the favorable case of the prior law, which is related to a nearest neighbour MRF, this method is time consuming and practically not suitable for parameter estimation.

2.6 Non-robustness of the MLE on observed data

2.6.1 Significance of the local Maximum Likelihood

Maximum Likelihood is often used in parameter estimation problems [21], because it is asymptotically optimal, since it is unbiased and efficient [29, 34] for large data records. Its probability density function is Gaussian, centered on the true optimal parameter value. Indeed, observations Y are noisy, therefore estimations made over Y using the Likelihood $P(Y \mid \theta)$ are also corrupted by noise. An estimator is said to have good properties if its variance remains sufficiently small to guarantee a good accuracy for the estimated value of θ.

How accurate have to be the parameters? As they are used to deconvolve images, they have to provide a small distance between the reconstructed solution X and the original image $\mathcal{X}$. This distance is often estimated by a mean square difference, which gives an estimate of the SNR. But other distances can be used, see chapter 4 for quality criterion definitions.

If this distance is evaluated for different values of θ, we first expect the distribution of $\hat{\theta}$ (the estimator of θ) to be centered on the value which provides the optimal SNR. Second, we hope that the flucutations of θ remain small, so that the SNR is not affected by the variations of the regularizing parameters.

Thus, an experimental verification on a set of satellite images has to be done to ensure that the MLE of θ leads to a near-optimal value of the SNR. This has been verified for homogeneous non-linear models for a particular case [23], but there is no formal proof that MLE should work in any case.

2.6.2 Computation of the CRLB

To check the robustness of the MLE, its variance has to be estimated. As the computation of the Cramer-Rao Lower Bound (CRLB) is intractable in this case, we will use another method to approximate this variance. Indeed, the CRLB is defined in the following way [34] :

$$\text{var}(\hat{\theta}) > E_\theta \left[\left(\frac{\partial \log P(Y \mid \theta)}{\partial \theta} \right)^2 \right]^{-1} \tag{2.20}$$

where E_θ denotes the expectation taken w.r.t. the law $P(Y \mid \theta)$. The observation Y is a random variable since it is a noisy observation of X, where X is also a random variable following the prior law $P(X \mid \theta)$. To estimate the CRLB, we have to sample X and Y from the joint law :

$$P(Y \mid \theta) = \int_{X \in \Omega} P(X, Y \mid \theta) \, dX \tag{2.21}$$

and then compute an empirical mean of the log-likelihood derivative. Such a sampling can be achieved by sampling X from Y and Y from X alternately, respectively from the posterior law (1.6) and the likelihood of Y (1.5). As posterior sampling is not easy, such an alternate sampling is time consuming, because the conditional sampling steps have to be repeated many times to ensure the convergence.

In fact, we did not use the CRLB for another reason. This bound only depends on the parameters of the prior and posterior laws (noise, convolution and regularizing parameters). It obviously does not take into account the real image $\mathcal{X}$. So it gives the accuracy of the MLE, by considering that X is governed by its prior law. However, if X is not correctly modeled by this law (which is often the case for complex satellite images), it would be preferable to consider that X is known, and equal to $\mathcal{X}$.

2.6.3 Variance of the MLE

In this section, we evaluate the variance of the MLE on observed data Y, to show this estimator is not robust to noise. First, we define a general framework to achieve such a computation, valid for both homogeneous and adaptive models. Then we consider the homogeneous case of nonlinear and Gaussian models to show that the MLE is very accurate in these cases. Finally, by using some results of the Gaussian model, we show that the variance is high in the adaptive case, and that this estimator has wrong properties w.r.t. image deconvolution.

We assume that to estimate the performance of the MLE we know the original image $\mathcal{X}$ and Y is a realization of the random process (1.1). We do not know the convolution product $H\mathcal{X}$, but a noisy version of it. Then, the MLE $\hat{\theta}$ defined by (1.7) is a random variable, which explicitely depends on the satellite image $\mathcal{X}$ we deal with. This is illustrated by figure 2.3. We have :

$$\mathcal{X} \mapsto \left\{ Y^i = H\,\mathcal{X} + N^i \right\} \mapsto \left\{ \hat{\theta}^i = \arg\max_{\theta} P(Y^i \mid \theta) \right\} \tag{2.22}$$

where N^i, Y^i and $\hat{\theta}^i$ represent realizations of the noise, the observation and the estimated parameters.

The expected value of $\hat{\theta}$ is $E[\hat{\theta}]$. It is the ideal parameter value, equal to the value estimated from the complete data Y^0, with the complete data MLE related to $P(\mathcal{X} \mid \theta)$. It is not the parameter value corresponding to the non-noisy image Y^0, as illustrated on Fig. 2.3.

The variance var$(\hat{\theta})$ gives access to the estimation error. This is the quantity we try to determine in the following sections, if we want to evaluate the accuracy of the MLE.

In the following, we denote Y^0 the non-noisy observation given by $Y^0 = H\,\mathcal{X}$, $E[\hat{\theta}]$ the mean of $\hat{\theta}$ estimated from $Y = H\,\mathcal{X} + N$, and $\hat{\theta}^0$ the parameters estimated from Y^0.

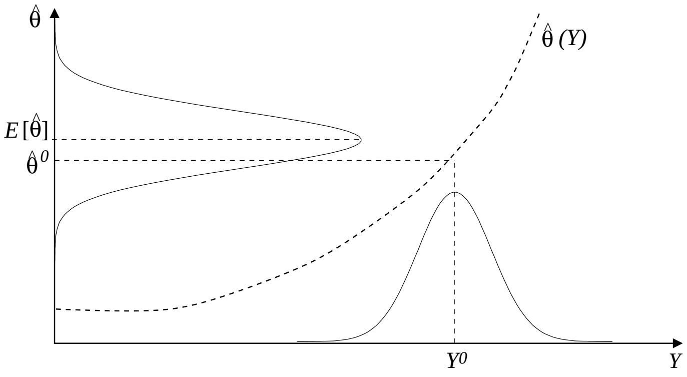


Figure 2.3: *MLE $\hat{\theta}$ as a function of the random variable Y*

Let us first consider the nonlinear φ-function based model described in section 1.2. We focus on the linear parameter $\theta = \lambda^2$, whereas δ has a fixed value.

The variance computation is based on the following assumptions :

- the log-likelihood is locally quadratic w.r.t. θ near $E[\hat{\theta}]$;
- it is also locally quadratic w.r.t. each component of the data Y near Y^0.

When the image Y varies around Y^0 because of the noise, the MLE varies around its mean value. The estimator $\hat{\theta}$ and Y are linked through the log-likelihood derivative, denoted $F(Y, \theta)$, by the equation :

$$F(Y, \hat{\theta}) = 0 \quad \text{with} \quad F(Y, \theta) = \frac{\partial \log P(Y \mid \theta)}{\partial \theta} \tag{2.23}$$

Then, let us take the Taylor expansion of F around the mean value $E[\hat{\theta}]$ and Y^0 :

$$F(Y, \hat{\theta}) \simeq F(Y^0, E[\hat{\theta}]) + (\hat{\theta} - E[\hat{\theta}]) \frac{\partial F}{\partial \theta}(Y^0, E[\hat{\theta}]) + \sum_{i,j} (Y_{ij} - Y^0_{ij}) \frac{\partial F}{\partial Y_{ij}}(Y^0, E[\hat{\theta}]) \tag{2.24}$$

To express the term $F(Y^0, E[\hat{\theta}])$ we take its Taylor expansion in θ (we assume that the mean value $E[\hat{\theta}]$ is close to $\hat{\theta}^0$) :

$$F(Y^0, E[\hat{\theta}]) \simeq F(Y^0, \hat{\theta}^0) + (E[\hat{\theta}] - \hat{\theta}^0) \frac{\partial F}{\partial \theta}(Y^0, \hat{\theta}^0) \tag{2.25}$$

These expressions are simplified by noting that $F(Y, \hat{\theta}) = F(Y^0, \hat{\theta}^0) = 0$. So if we define β and β^0 as follows :

$$\beta = \left. \frac{\partial^2 \log P(Y \mid \theta)}{\partial \theta^2} \right|_{Y = Y^0,\, \theta = E[\hat{\theta}]} \qquad \beta^0 = \left. \frac{\partial^2 \log P(Y \mid \theta)}{\partial \theta^2} \right|_{Y = Y^0,\, \theta = \hat{\theta}^0} \tag{2.26}$$

and the vector α as the gradient of F w.r.t. Y :

$$\alpha = \nabla_Y \frac{\partial \log P(Y \mid \theta)}{\partial \theta}\bigg|_{Y=Y^0,\, \theta=E[\hat{\theta}]} \tag{2.27}$$

then equation (2.24) may be rewritten as :

$$(\hat{\theta} - E[\hat{\theta}])\beta + (E[\hat{\theta}] - \hat{\theta}^0)\beta^0 + (Y - Y^0)^t \alpha \simeq 0 \tag{2.28}$$

It is now possible to express the variance of the estimator as a function of the Gaussian noise $N = Y - Y^0$. The variables N_i are independent, with variance σ^2, so we obtain :

$$\text{var}(\hat{\theta}) = E\left[(\hat{\theta} - E[\hat{\theta}])^2\right] \simeq \frac{E\left[\left(N^t\alpha + (\hat{\theta}^0 - E[\hat{\theta}])\beta^0\right)^2\right]}{\beta^2} \tag{2.29}$$

which is simplified and finally gives, denoting $\gamma = (\hat{\theta}^0 - E[\hat{\theta}])\beta^0$:

$$\text{var}(\hat{\theta}) \simeq \frac{\sigma^2 \|\alpha\|^2 + \gamma^2}{\beta^2} \tag{2.30}$$

The computation of the mean value of $\hat{\theta}$ is relatively time consuming since the estimation has to be averaged on a large number of input images Y, obtained by adding noise to the constant Y^0.

We can use a MCMC method to estimate β and β^0, by sampling from both prior and posterior densities, in the same way as the gradient of the log-likelihood :

$$\frac{\partial^2 \log P(Y \mid \theta)}{\partial \theta^2} = E_Y\left[\frac{\partial \Phi(X)}{\partial \theta}^2\right] - E_Y\left[\frac{\partial \Phi(X)}{\partial \theta}\right]^2 - E_X\left[\frac{\partial \Phi(X)}{\partial \theta}^2\right] + E_X\left[\frac{\partial \Phi(X)}{\partial \theta}\right]^2 \tag{2.31}$$

Expectations are estimated using empirical means as in equation (1.11). This expression is computed with $Y = Y^0$. For β we take $\theta = E[\hat{\theta}]$ and for β^0 we take $\theta = \hat{\theta}^0$.

It is possible to neglect the term γ^2 in equation (2.30), since it is small compared to the noise-dependent term ($E[\hat{\theta}] \simeq \hat{\theta}^0$), and to simplify the problem by taking $\hat{\theta}^0$ instead of the mean of $\hat{\theta}$, which only needs one estimation step. Then we have to compute :

$$\text{var}(\hat{\theta}) \simeq \frac{\sigma^2 \|\alpha^0\|^2}{(\beta^0)^2} \tag{2.32}$$

where α^0 is defined as in (2.27) but with $\theta = \hat{\theta}^0$.

The formalism presented here is general and can be used for any type of model. Furthermore, it will be extended to the inhomogeneous case, since the goal of this study is to compare the performance of the MLE for both homogeneous and adaptive models.

Nonlinear homogeneous models

Let us first consider nonlinear homogeneous models, to demonstrate the accuracy of the MLE in this case. See appendix B for the detailed calculations of the vector α :

$$\alpha = \frac{1}{\sigma^2} E_Y \left[\frac{\partial \Phi(X)}{\partial \theta} H X^r \right] \tag{2.33}$$

If we split the posterior samples X into a random part X^r and a mean $E_Y[X]$, it finally gives (see appendix B) :

$$\text{var}(\hat{\theta}) \simeq \frac{1}{(\sigma \beta^0)^2} \left\| E_Y \left[\frac{\partial \Phi(X)}{\partial \theta} H X^r \right] \right\|^2_{Y=Y^0,\, \theta=\hat{\theta}^0} \quad \text{with} \;\; X^r = X - E_Y[X] \tag{2.34}$$

The above approximate expression of the variance is computed the same way as β, using MCMC and empirical means.

This expression enables to evaluate the estimation error, when the original image is known. Even if it is not the case when we process real data, this is true when we study the performance of the MLE on a particular image, by simulating the observation Y from $\mathcal{X}$. Thus, the value of the variance obtained this way is related to a precise type of data (satellite images in our case).

Thus, we have applied this method to the image of fig. 1.1 with the model presented in section 1.2, to study the variations of the MLE $\hat{\lambda}$ for the linear parameter λ^2. The estimation is done by algorithm 1.3.1. To reduce the computing time, we prefer to evaluate β^0, i.e. the second derivative of the log-likelihood, numerically : we evaluate only the first derivative for two values of θ and take the difference to compute β^0.

Then, equation (2.34) gives var($\hat{\theta}$) $\simeq 10^{-5}$. This means a relative accuracy of about 1%. See figure 2.4 for an illustration ; the estimate $\hat{\theta}$ corresponds to an acceptable value of the SNR, and the visual quality is nearly optimal near $\hat{\theta}$. This demonstrates the good properties of the estimator for homogeneous nonlinear models.

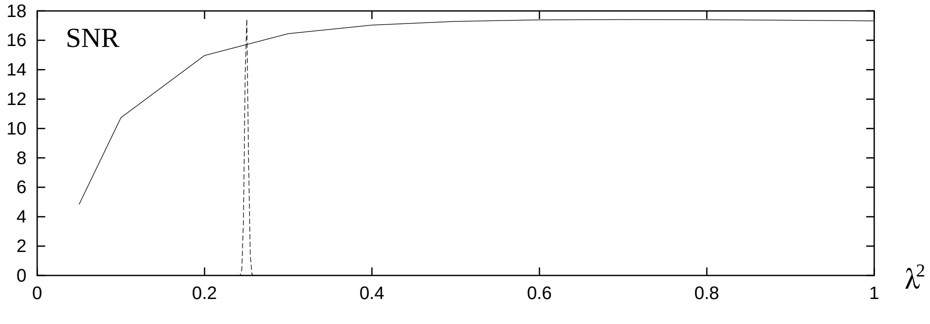


Figure 2.4: *Homogeneous nonlinear model : SNR of the reconstructed image of figure 1.1 for different values of $\theta = \lambda^2$; dashed : distribution of the MLE $\hat{\theta}$ (with $E[\hat{\theta}] \simeq 0.25$)*

The φ-models also include the Gaussian model, related to a quadratic φ-function. This case will be discussed in the next paragraph by using the general results of nonlinear models.

Homogeneous Gaussian models

In the homogeneous Gaussian case, all the expectations necessary to compute the MLE variance can be deterministically evaluated, whereas they only take into account Gaussian variables. The computations are made in the Fourier basis, which diagonalizes the derivative and convolution operators, as seen in section 1.3.2. Even if this model cannot be used for restoration because it does not preserve edges, this case will be useful later to get an approximation for inhomogeneous fields.

The expression of β is given by equation (B.2) in appendix B.

If we denote $c_{ij} = |\mathcal{F}[D_x]|^2_{ij} + |\mathcal{F}[D_y]|^2_{ij}$ and $w_{ij} = |\mathcal{F}[h]|^2_{ij}/2\sigma^2$, we have :

$$\beta = \frac{\partial^2 \log P(Y \mid b)}{\partial b^2} = \frac{1}{2} \sum_{i,j} \left(-\frac{1}{b^2} + \frac{c_{ij}^2}{(b\, c_{ij} + w_{ij})^2} + 2 \frac{c_{ij}^2 w_{ij} |\mathcal{F}[Y^0]_{ij}|^2}{\sigma^2 (b\, c_{ij} + w_{ij})^3} \right) \tag{2.35}$$

We refer again to appendix B for the final expression of the variance of $\hat{b}$, according to equation (2.34) :

$$\text{var}(\hat{b}) \simeq \frac{1}{\sigma^2 \beta^2} \sum_{i,j} \frac{c_{ij}^2 w_{ij}^2 |\mathcal{F}[Y^0]|^2_{ij}}{(b\, c_{ij} + w_{ij})^4} \tag{2.36}$$

We can evaluate this expression in $\hat{b}^0$ instead of $E[\hat{b}]$, it simplifies the problem without changing the result (because it varies slowly between these two values of b).

As we can see on figure 2.5, the homogeneous MLE exhibits good properties, because it provides a satisfying SNR and a near optimal visual quality when applied to image deconvolution, and its variance is small (about 10^{-11}), giving a relative accuracy of about 1% for $\hat{b}$. The robustness of this estimator essentially comes from the mean computed over the whole image. What happens if we replace this global computation by a local, space variant one?

Inhomogeneous Gaussian models

In the inhomogeneous case, equations (2.31) and (2.34) cannot be simplified, because the diagonalization in the frequency space is no longer possible, as the derivative terms are not block-circulant.

There are two possible ways to estimate the variance. The first one is the most general and should work in any case, but is intractable from a computational point of view. That is why we present a second one, which only works in a particular case but which is sufficient to demonstrate the wrong properties of the variance for adaptive models.

- *Stochastic method* : sample from the prior and posterior laws, with the adaptive sampling methods described in appendix C, to evaluate the expression (2.34). Although

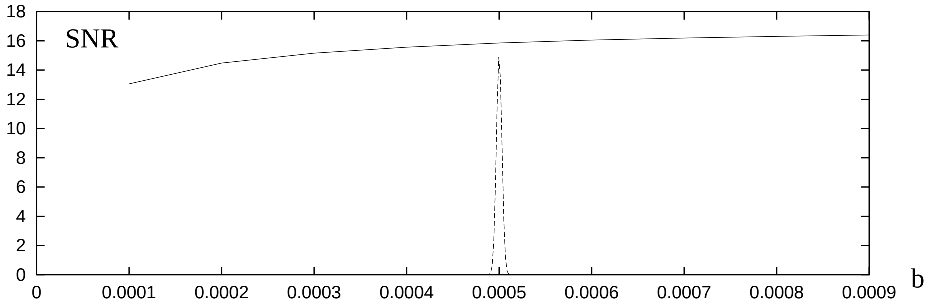


Figure 2.5: *Homogeneous Gaussian model : SNR of the reconstructed image of figure 1.1 for different values of* b *; dashed : distribution of the MLE* $\hat{b}$ *(with* $E[\hat{b}] \simeq 0.0005$*)*

this computation is time consuming, it can be applied on satellite images, whose optimal parameters are space varying when estimated in an inhomogeneous framework.

- *Homogeneous optimal parameters* : the stochastic method is intricate, because the parameter values are not equal. Therefore we take the particular case of a carefully designed image $\mathcal{X}$, which gives mean estimated parameters that are equal to a constant value b. Then it is possible to use simple expressions like the previous ones, since all computations are made for a constant parameter field.

Let us detail here the second method to evaluate the quality of the adaptive MLE. Space invariant regularization can be applied, for example, on images whose pixel differences have a space invariant magnitude. A checkerboard texture, at pixel level, has such a property. We can take $\mathcal{X}_{ij} = 0$ for $i+j$ odd and $\mathcal{X}_{ij} = M$ for $i+j$ even. The amplitude M can be chosen between 10 and 100, corresponding to values encountered in real satellite images.

To recover such a texture from an ideal blurred image (not noisy), regularizing parameters have to be equal for each pixel difference, as all pixel differences are equal in magnitude.

Equation (2.28) is rewritten by recalling that b is now a vector, and the Taylor expansion is taken w.r.t. all the components of b. Then β is a Hessian matrix ; we denote β^{rs}_{ijkl} the derivative w.r.t. $\theta_1 = b^r_{ij}$ and $\theta_2 = b^s_{kl}$ where r and s symbolize the directions x or y :

$$\beta^{rs}_{ijkl} = \frac{\partial^2 \log P(Y \mid \{b^x\}, \{b^y\})}{\partial b^r_{ij} \partial b^s_{kl}} \tag{2.37}$$

We also have to redefine α, which is now a matrix whose lines are given by :

$$\alpha^{x,y}_{ij} = \nabla_Y \frac{\partial \log P(Y \mid \{b^x\}, \{b^y\})}{\partial b^{x,y}_{ij}} \Bigg|_{Y=Y^0,\, \theta=E[\hat{\theta}]} \tag{2.38}$$

Then equation (2.28) becomes a system of $4N_xN_y$ equations :

$$(\hat{b} - E[\hat{b}])^t \beta + (E[\hat{b}] - \hat{b}^0)^t \beta^0 + (Y - Y^0)^t \alpha \simeq 0 \tag{2.39}$$

In appendix B, we show that this system is nearly diagonal. Furthermore, the diagonal elements of β are equal to a constant in this particular case, which we denote $\bar{\beta}$. In the same way, the elements of α are equal to a constant vector denoted $\bar{\alpha}$. Then we find an expression of the variance similar to (2.32) :

$$\text{var}(\hat{\theta}) \simeq \frac{\sigma^2 \|\bar{\alpha}^0\|^2}{(\bar{\beta}^0)^2} \tag{2.40}$$

The expectations necessary to compute $\bar{\beta}^0$ of equation (2.31) are estimations of the variance, covariance and 4-order moment of the Gaussian variables $(D_x X)_{kl}$ and $(D_y X)_{kl}$. These quantities are computed in appendix B.

$$\bar{\beta} = 2\Big[V_Y(V_Y + 2m) - V_X^2\Big] \tag{2.41}$$

where m is the magnitude of any pixel difference of the reconstructed image $\hat{X}$. For the chosen image $\mathcal{X}$ the expression of m^2 can be simplified, whereas the Fourier transform of $\mathcal{X}$ has only two nonzero coefficients at $(0,0)$ and $(\frac{N_x}{2}, \frac{N_y}{2})$. We have :

$$m^2 = \frac{M^2 k^2}{(4b + w_{N_x/2, N_y/2})^2} \tag{2.42}$$

V_X and V_Y are the prior and posterior expectations of $(D_x X^r)^2_{ij}$ or $(D_y X^r)^2_{ij}$, where X^r is the random part ($X^r = X - E_Y[X]$) :

$$V_X = \frac{1}{4\bar{b}}, \quad V_Y = \frac{1}{4} \sum_{i,j} \frac{c_{ij}}{\bar{b}\, c_{ij} + w_{ij}} \tag{2.43}$$

The variance is then given by :

$$\text{var}(\hat{b}^{x,y}_{kl}) \simeq \frac{m^2}{\bar{\beta}^2} \sum_{ij} \frac{c_{ij} w_{ij}}{(\bar{b}\, c_{ij} + w_{ij})^2} \tag{2.44}$$

To ensure there is no loss of information between $\mathcal{X}$ and Y, h has to provide a sufficiently sharp PSF in this case. Evaluations can also be done without convolution.

Let us consider the results for the denoising problem (i.e. H is equal to identity). This gives a variance around 10^{-6} for a mean value of $\hat{b}$ around 10^{-3}, with M near 10. For other values of M, the estimated parameters change according to M, and the relative fluctuation of b always remains of the same order as b. These results must be taken into account without forgetting that the computation is based on an approximation, i.e. $\hat{b}$ varies slowly around its mean value, because of the noise N.

How to interpret these results? It is not surprising that parameters adapt to the noise : indeed, the estimation takes into account the observed data. We have to know if the parameters locally adapt to the data by providing more regularization for higher noise, or if the parameters are corrupted by the noise and provide insufficient regularization.

To address this problem, let us consider the expressions of V_X and V_Y (2.43), and m^2 (2.42), which give the variances of the pixel differences for the prior and posterior law, when all inhomogeneous parameters are equal to b. An approximate method to estimate the model parameters consists of taking the corresponding $b_{ij}^{x,y}$ instead of b to evaluate the variances and the mean m at pixel (i,j) (see appendix C.2 for details).

Equation (2.18) can be rewritten in the following way :

$$\frac{\partial \log P(Y \mid \{b^x\},\{b^y\})}{\partial b_{ij}^{x,y}} \simeq V_X(b_{ij}^{x,y}) - V_Y(b_{ij}^{x,y}) - m^2(b_{ij}^{x,y}) \tag{2.45}$$

if the parameters are nearly constant.

We have plotted $V_X - V_Y$ versus m^2 in Fig. 2.6, the first term is independent from the data, and the other contains the data and also the noise. Indeed, if there is some noise in Y, the mean value m^2 is increased or decreased by Δm^2 for a given pixel difference. How does this variation Δm^2 affect the estimated value $\hat{b}_{ij}^{x,y}$? On Fig. 2.6, we have represented two shifted curves for m^2, the shift corresponding to the noise. The limit of m^2 for $b \to 0$ is finite, corresponding to deconvolution without regularization. But for $b \to 0$, we have $V_X - V_Y \to \infty$ because of the term $1/4b$. So m^2 is lower than $V_X - V_Y$ for $b < \hat{b}$, and it is also a decreasing function. In the presence of noise, this curve is slightly shifted.

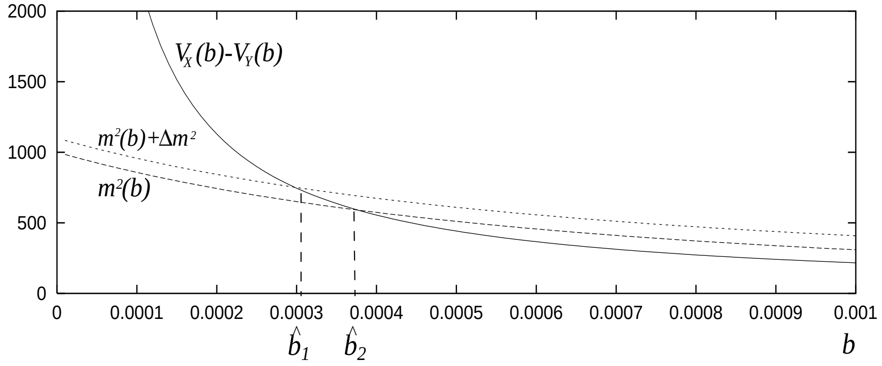


Figure 2.6: *Solid : $V_X(b) - V_Y(b)$, dashed : $m^2(b)$, the shift corresponds to the observation noise, the intersection gives the estimated value $\hat{b}$*

If m^2 is increased by the noise, the intersection of the curves is shifted to the left, so the log-likelihood is cancelled for a lower b value. It means that MLE provides less regularization in this case. So the difference Δm^2 induced by the noisy observation cannot be suppressed by the regularizing function, and all the local variations of b preserve the noise instead of filtering it.

As b has near 100% relative variation in the studied case, it is sure that for many pixel differences, which are high only because of the noise, the corresponding estimated parameters are too low, and provide insufficient regularization. The result is a noisy restored image.

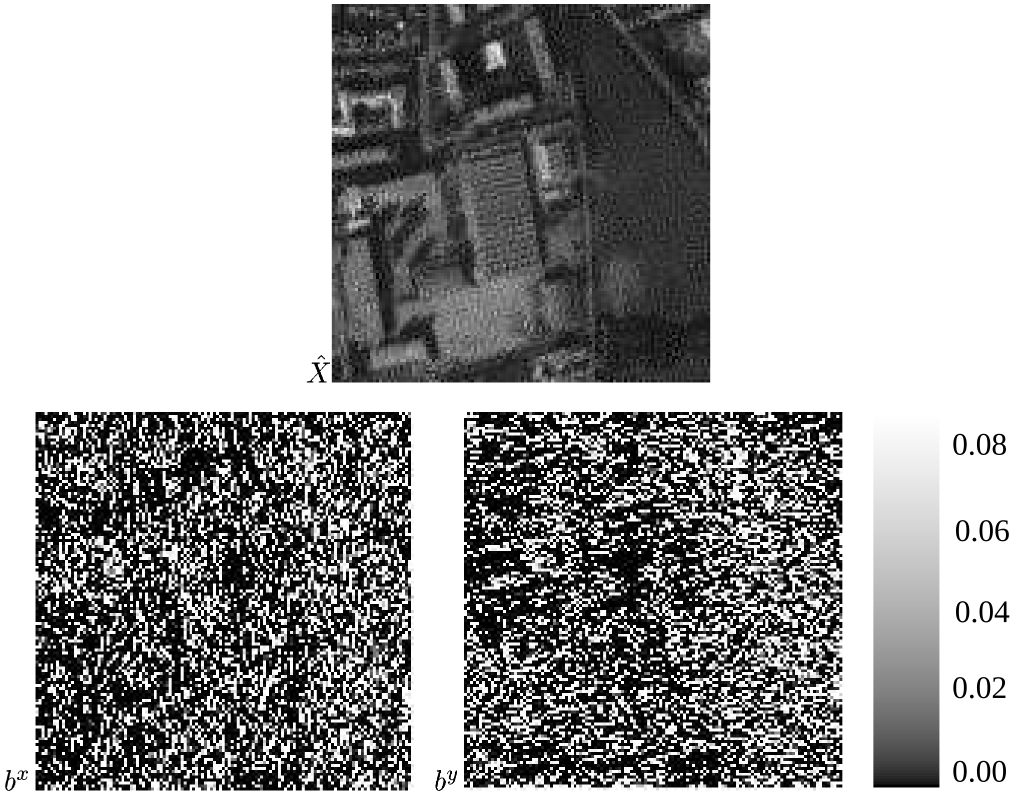


Figure 2.7: *Image of fig. 1.1 reconstructed with adaptive parameter fields* $\{b^x\}, \{b^y\}$ *estimated from blurred and noisy data* Y

As shown by the images presented in figure 2.7, even in the case of a constant ground truth, the adaptative MLE is too sensitive to the noise, and the space variant b values do not adapt to the data in the sense we expect, i.e. by avoiding noise amplification while preserving the details.

Furthermore, the convergence of the estimation algorithm is not sure, because of the high-dimensionality of the optimization problem. In practice, results may depend on the initialization. The initial parameter choice is constant : the value can be chosen by a fast homogeneous estimation method, or by hand. More generally, there is no proof of the unicity of the likelihood maximizer, and multiple solutions can be found, without giving satisfactory deconvolution results. Experiments have shown that results are always noisy, regardless of the initialization.

This estimator is also not suitable for the deconvolution problem with inhomogeneous parameters. Therefore, we have to find another method, robust to noise to optimize the model parameters.

Chapter 3

Approximation of the original image

3.1 Necessity of a good approximation

As seen in the previous chapter, the complete data MLE provides optimal regularization parameters within an image deconvolution framework. The main difficulty of implementing such a tool comes from the fact that we cannot access the ground truth, i.e. the original image $\mathcal{X}$. Therefore, in the following, we will try to compute an approximation of this image, whose properties enable a robust parameter estimation.

Many methods can be used to determine such an image from the data Y. But we have to recall that the proposed method to estimate the parameters (MLE on complete data) is quite sensitive to small variations in regions corresponding to a homogeneous ground truth. It is also sensitive to the quality of the small features. This excludes the use of classical deconvolution algorithms as Wiener filter [19] or Artur algorithm [8], which either keep a large amount of noise, or smooth the textures. It also excludes the simple denoising of Y, which does not remove the blur.

In fact, there are reasons to prefer using completely different methods to estimate the approximation of $\mathcal{X}$ and to achieve the restoration. The residual noise coming from the approximation must have a different spatial localization from the noise linked to the final deconvolution process.

For example, for a given pixel, a very low value of b (coming from an estimation error) means little regularization. If the nonregularized solution $H^{-1}Y$ (simple inversion of H) exhibits a peak of noise at this place, this peak is not removed because the value of b is too low. It is a problem only if the level of the peak is high, which means that the estimation error and the noise in $H^{-1}Y$ are correlated.

Therefore, the iterative algorithms which alternately restore and estimate from the restored image give noisy results in constant regions, because of the too high correlation between estimation and restoration processes (a high gradient g gives low b, which restores again a high value of g). To avoid this, a two-step method is proposed below.

We develop a *hybrid* method consisting of two separate steps :

- finding a good approximation and estimating the inhomogeneous parameters from it,
- computing the regularized solution with the estimated parameters within an IGMRF context.

3.2 Efficient image representations for denoising

The process used to estimate $\mathcal{X}$ has to preserve the textures, as seen in paragraph 2.4.2. Moreover, the noise must remain small in homogeneous parts. Therefore, the denoising method must be efficient and detail-preserving. To achieve this, some authors such as Donoho et al. [12, 13, 14], Mallat and Kalifa [25, 26, 27, 28, 32], have proposed to denoise the image after a deconvolution without regularization. The images are represented using a wavelet or wavelet packet basis, and the denoising process is done in this basis.

We have seen in paragraph 1.2 that the deconvolution problem is ill-posed. Thus, a simple inversion in the frequency space, consisting of dividing $\mathcal{F}[Y]$ by $\mathcal{F}[h]$, gives an unacceptably noisy solution. To denoise such a solution, an efficient image representation has to be chosen, in order to separate the signal as much as possible from the noise.

A compact representation consists of approximating a signal with a few parameters, which can be the coefficients of the decomposition in a given basis. This basis has to adapt to the structure of the data to be restored. To denoise a signal contaminated by white noise, it is decomposed in a basis which transforms the signal into a few high amplitude coefficients, with a small rest of low amplitude. Then, most of the noise is suppressed by thresholding the representation, i.e. by cancelling the smallest coefficients [12]. Donoho and Johnstone [14] have shown that such a thresholding estimator is optimal, if the signal representation is *sufficiently compact*.

To address the deconvolution problem, we have to filter a colored noise, therefore one needs to adapt the basis to the covariance properties of the noise. It means that this covariance should be *nearly diagonal* in this basis. The Fourier basis achieves such a diagonalization, but the energy of the signal no more concentrates, so the Fourier transform is not suitable for any thresholding method. Indeed, the Fourier transform does not approximate efficiently bounded variation signals as satellite images.

A good compromise is made if a *wavelet packet* basis [9] is used, since it nearly realizes the two essential conditions, i.e. the signal representation is sparse, the noise covariance operator is nearly diagonalized. (See figure 3.1 for an illustration : the signal and the noise are efficiently separated).

In [24], we have proposed to use complex wavelet packets. Complex wavelets have first been introduced by Kingsbury in [30]. They provide a better restoration than real wavelets, by taking into account 6 directions. We have implemented a complex wavelet *packet* algorithm, to ensure the near diagonalization of the noise covariance necessary for

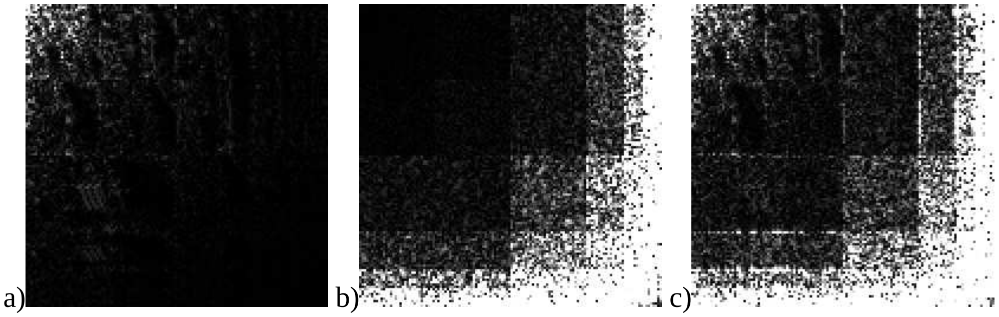


Figure 3.1: *Representations in a real wavelet packet basis : a) original image $\mathcal{X}$, b) deconvolved noise, c) deconvolved image by nonregularized inversion*

deconvolution. The noise variance and the optimal threshold in each subband are automatically estimated by the proposed algorithm. This method performs the inversion faster than real transforms and better reconstructs features of various orientations (see [24] for details).

So it provides a sufficiently good approximation of the ground truth (i.e. sharp textures and noiseless homogeneous areas) from which the adaptive parameter estimation can be made.

The full algorithm, which ables to automatically estimate the parameters and deconvolve the image, by using this hybrid approach, will be detailed in the next chapter.

Chapter 4

Deconvolution algorithm

4.1 Introduction

The regularizing model is an inhomogeneous Gaussian model, defined by equation (2.1). As seen in chapter 1, the regularized solution $\hat{X}$ is computed by a deterministic algorithm, minimizing the following functional :

$$\hat{X} = \arg\inf_{X} \left[\frac{\|Y - HX\|^2}{2\sigma^2} + \sum_{ij} \left\{ b^x_{ij} (D_x X)^2_{ij} + b^y_{ij} (D_y X)^2_{ij} \right\} \right] \tag{4.1}$$

The deconvolution algorithm presented here can be summarized as follows : first, we use a thresholding method in a wavelet basis to estimate the source image from its observation Y. The image obtained this way is not sufficiently sharp to solve the deconvolution problem. However, we use this estimate to determine the inhomogeneous parameters of the model and use them to get the final result.

The 3 essential steps of the algorithm are :

- automatically thresholding the image Y, deconvolved without regularization, in a complex wavelet packet basis (algorithm COWPATH defined in [24]) ;
- estimating the inhomogeneous parameters of the model (2.1) on the result of the previous step, by a complete data MLE (equation (2.10) of chapter 2) ;
- solving the equation (4.1) by an accelerated descent algorithm to compute the final solution $\hat{X}$.

Here, we focus on the last two steps, i.e. the practical implementation of the inhomogeneous parameter estimation and of the related adaptive deconvolution method, studying in particular the optimization and the robustness of each step.

4.2 Estimation of the optimal adaptive parameters

The study of the robustness of the adaptive complete data MLE started in section 2.4. It can be completed now, since we have determined which method we use to obtain an approximation of the original image $\mathcal{X}$. The algorithm COWPATH detailed in [24] provides an estimate of the unknown image which satisfies the desired properties, i.e. low noise in homogeneous areas and sharp textures.

The edges are visually not sharp enough, but are sufficient to provide an accurate parameter estimation. Indeed, we have shown that this estimator is robust to the blur on the edges, if the directions of the features are preserved.

4.2.1 Evaluation of the residual noise

On the other hand, the estimator seems to be quite sensitive to the noise in constant areas. Therefore we have to take into account the residual noise of the wavelet shrinkage step. Thus, we redefine the optimal weighting (2.10) to get a more robust estimator in these regions :

- simulating an image Y of a white Gaussian noise $N(0, \sigma^2)$;
- applying algorithm COWPATH [24] to deconvolve this image, the result is denoted X and corresponds to the residual noise ;
- computing the derivative $D_x X$ or $D_y X$, whose pixels are denoted Δg ;
- computing the histogram of this gradient image.

By studying the distribution of the gradients Δg of the residual noise, we finally find that they have bounded values, i.e. $|\Delta g| \leq \epsilon$. We can use some thresholding method to filter the gradients whose magnitude is lower than ϵ. This provides an efficient way to get a more robust MLE and it will be detailed in the next paragraph, since it is relatively simple.

But it is not the only method, and many other ones could be used. For example, we could compute the MLE in the incomplete data case (the data is a noisy version of the ground truth), taking into account the statistics of the residual noise on the gradients, instead of filtering the gradients and then applying the MLE on complete data.

4.2.2 A more robust optimal weighting

To ensure that the residual noise does not affect the estimated values of the parameters b, the derivatives g of the deconvolved images issued from algorithm COWPATH [24] are thresholded before applying equation (2.10). A hard thresholding is performed by setting to zero all gradients verifying $|g| < \epsilon$. We then modify equation (2.10) in this sense. We also set an upper bound for b ($b_{max} = 1$) to avoid computational difficulties in the final deconvolution :

$$\hat{b}_{ij}^{x,y} = \frac{1}{\min\left(4\left[\theta_\epsilon^h(D_{x,y}X_{ij})\right]^2, 1\right)} \tag{4.2}$$

where $\theta_\epsilon^h(g) = g$ if $|g| < \epsilon$, and 0 elsewhere.

Then, in homogeneous areas which contain only deconvolved noise, the regularization is maximum ($b = 1$) because g is null. This avoids reconstructing isolated noise peaks, since the residual noise often leads to insufficient regularization.

4.3 Improving the efficiency

4.3.1 Taking into account the directions

A significant improvement of the method presented in the previous sections consists of introducing the orientation of the gradients. The model used for parameter estimation only takes into account the horizontal and vertical directions. It is possible to introduce the diagonal directions u and v without increasing too much the computational complexity of the estimation step, by combining the horizontal and vertical pixel differences (related to the 4 nearest neighbours) with the diagonal gradients. This introduces two new parameter fields b^u and b^v, related to pixel differences along these directions.

The nearest neighbours model is then replaced with a 8 neigbour MRF model, in the restoration step only (the estimation is done with the previous 4 neigbours MRF). This model needs 4 parameters for each pixel, corresponding to 4 directions.

Then, for each pixel, we use the 4 gradients g^x, g^y, g^u, g^v :

$$g^x_{ij} = X_{i+1,j} - X_{ij} \quad g^y_{ij} = X_{i,j+1} - X_{ij} \quad g^u_{ij} = X_{i+1,j+1} - X_{ij} \quad g^v_{ij} = X_{i-1,j+1} - X_{ij} \tag{4.3}$$

to estimate the 4 associated parameters b^x, b^y, b^u, b^v using equation (2.10).

But they cannot be used directly in the new 8 neigbour model by simply minimizing the corresponding functional with the parameters estimated this way, because the estimator (2.10) is related to a 4 neigbour MRF.

To combine all these parameters, we use a technique inspired from the discretization of the divergence term $div(b\nabla X)$ presented in [35]. This term occurs, within a variational context, in the minimization of this functional :

$$\|Y - h \star X\|^2 + \int_\Omega b|\nabla X|^2 \, d\Omega \tag{4.4}$$

where b, X, Y and h are continuous functions. Indeed, the associated Euler-Lagrange equation is :

$$div(b\nabla X) = \frac{1}{2\sigma^2}\|Y - h \star X\|^2 \tag{4.5}$$

The model presented here is in fact a discretization of this functional, and taking 8 instead of 4 neighbours finally consists of taking a more accurate discretization.

The technique consists of using the direction α of the gradient of the approximate original image at pixel (i, j) :

$$\alpha \equiv \arccos\left(g^y_{ij} / \sqrt{(g^x_{ij})^2 + (g^y_{ij})^2}\right) [\pi/2] \tag{4.6}$$

(To obtain more robust results, it is preferable to compute the gradients in this expression by smoothing along the orthogonal direction.)

The value of α determines how to distribute the parameters over each one of the 4 directions, in the term $div(b\nabla X)$ which is used in the deconvolution algorithm :

$$div(b\nabla X) \propto X\star \left[(1-\frac{4\alpha}{\pi}) \begin{pmatrix} 0 & b^y_{i,j-1} & 0 \\ b^x_{i-1,j} & -s_{xy} & b^x_{i,j} \\ 0 & b^y_{i,j} & 0 \end{pmatrix} + \frac{2\alpha}{\pi} \begin{pmatrix} b^u_{i-1,j} & 0 & b^v_{i,j-1} \\ 0 & -s_{uv} & 0 \\ b^v_{i,j} & 0 & b^u_{i,j} \end{pmatrix} \right] \tag{4.7}$$

where $s_{xy} = b^x_{i,j} + b^x_{i-1,j} + b^y_{i,j} + b^y_{i,j-1}$ and $s_{uv} = b^u_{i,j} + b^u_{i-1,j} + b^v_{i,j} + b^v_{i,j-1}$.

For example, if $\alpha = 45^o$, we only take into account the diagonal parameters b^u and b^v. If there is a diagonally oriented texture, this approach enables to smooth only along the direction of the texture, while the standard approach produces both smoothing along horizontal and vertical directions, which generally can damage the texture.

4.3.2 Choosing the best initialization

Since the deconvolution method is iterative, the choice of the initial image determines the speed of convergence. A bad initialization often slows down the algorithm, even if the solution does not depend on the followed path. Thus, we initialize with an image close to the solution. This image is simply the approximation used for parameter estimation. Then, in the case of the SPOT 5 simulation, 30 to 40 iterations are sufficient.

To understand why this initialization seems to be the best one, let us recall that minimizing the functional corresponding to the adaptive regularization model is equivalent, in homogeneous areas, to solve the heat equation. It is equivalent to a diffusion process, which smoothes these areas by averaging many pixels iterately. Starting with a noisy image is clearly not a good initialization, because many iterations are needed to achieve the diffusion process over constant areas which can be very large and to obtain smooth areas.

If we use the approximation image obtained by a wavelet thresholding algorithm, these areas are smooth. So there is no need for additional smoothing. The only parts to process are the edges, which are not sufficiently sharp because of the insufficient spatial localization of the wavelets. Thus, the aim of the adaptive deconvolution step is to sharpen the edges, leaving other parts unchanged. Finally, we conclude that starting with the approximation image is much faster than starting with the observed image Y.

4.3.3 Optimization method

We can use a conjugate gradient algorithm to optimize equation (4.1). But this method can be improved by making some assumptions.

We have just seen that the aim of the adaptive deconvolution algorithm is to sharpen the edges. Edges correspond to high gradient values (i.e. low regularizing parameter values). It is possible to accelerate the recovery of the edges by lowering the corresponding regularizing parameters at the beginning of the algorithm and then by restoring them progressively to their estimated value. This ensures to obtain the right solution, related to the estimated parameter values. This technique can speed up the gradient descent step by a factor 2. This is important, given that the limiting factor of the hybrid algorithm (wavelet thresholding followed by adaptive estimation and deconvolution) is the last adaptive estimation step. Indeed, the complexity of this step is directly linked to the size of the blur kernel, since the convolutions are done in the image space, and not in the frequency space, due to the adaptive nature of the model.

4.4 The proposed deconvolution method

4.4.1 The hybrid algorithm "DEPA"

The proposed deconvolution method, called DEPA (Deconvolution with Estimation of Adaptive Parameters), consists of the following steps (see Fig. 4.1) :

ALGORITHM 4.4.1 (DEPA)

- *Deconvolution of Y with algorithm COWPATH [24]*
- *Computation of the gradients*
- *Estimation of the residual noise of the gradients :*
  1. *Simulation of a white Gaussian noise of variance σ^2*
  2. *Deconvolution of the noise with COWPATH,*
     with the same parameters as for the image Y
  3. *Computation of the residual gradients*
  4. *Estimation of the variance σ_g^2 of the residual gradients*
- *Thresholding of the gradients, using σ_g^2*
- *Estimation of the IGMRF parameters using the MLE cf. equ. (2.10) (these two steps are combined in equation (4.2))*
- *Optional: computation of the diagonal parameters (see section 4.3.1)*
- *Deconvolution by MAP estimation, by minimizing equ. (4.1)*
  (initialization by the result of COWPATH [24], and minimization of (4.1) by a conjugate gradient algorithm)

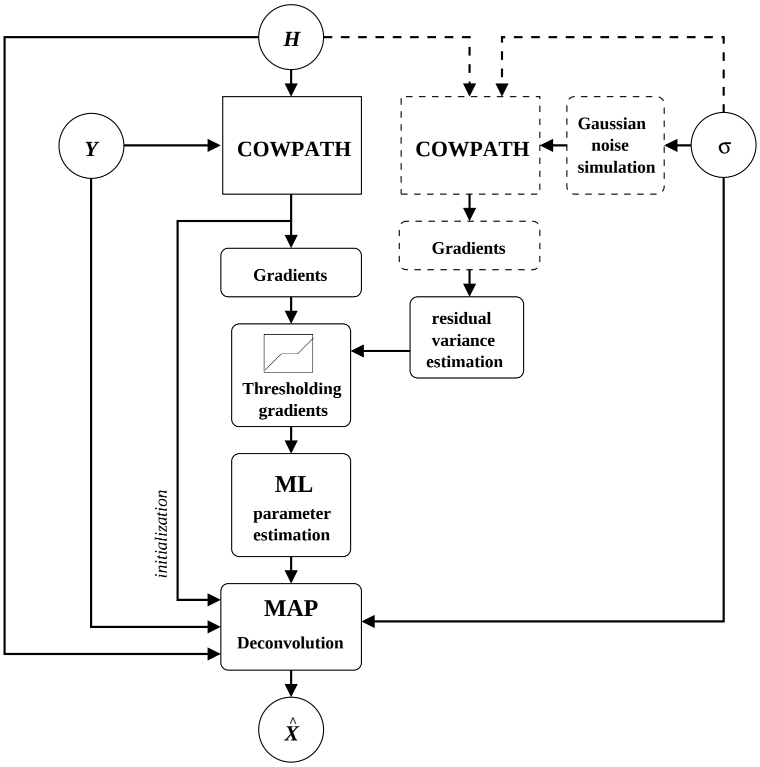


Figure 4.1: *The DEPA algorithm (Deconvolution with Estimation of Adaptive Parameters)*

4.4.2 Cost of the algorithm

Deconvolution with adaptive parameters

With given parameter fields b^x and b^y, the functional (4.1) is minimized by a conjugate gradient algorithm, which needs $20 + 8n$ op/pix (operations per pixel) for each iteration, where n is the size of the separable convolution kernel h (in the case of the 11×11 convolution kernel provided by the CNES, which is shown in Fig. 1.1, it represents about 110 op/pix). The classical algorithm needs about 40 iterations, which means 4400 op/pix for the complete minimization. An accelerated method can be used, as explained in section 4.3.3. Even if stopped after only 20 iterations, this technique allows to obtain satisfactory results. It corresponds to 2200 op/pix.

If we take into account the directions (see section 4.3.1), add 15 op/pix for each iteration. It means 5000 op/pix for the minimization, and 2500 op/pix for the accelerated version.

Adaptive parameter estimation

This part of the algorithm is the fastest one. It is possible to use a fast version of the COWPATH method described in [24], to obtain the approximation image needed for ML parameter estimation. It is essentially based on 3 DCT (Discrete Cosine Transform) and 3 CWPT (Complex Wavelet Packet Transform), and it requires about $15\log_2 N_x + 445$ op/pix (for a 512×512 image, it needs 580 op/pix). The estimation step, consisting in computing b^x and b^y by using equation (4.2), is negligible compared to the previous step.

Total cost

The total cost is $15\log_2 N_x + 445 + (20 + 8n)N_i$ op/pix, where N_i is the number of iterations in the adaptive deconvolution step, n is the size of the filter and N_x the size of the image. In the case of the image provided by the CNES (see Fig. 4.2 and Fig. 1.1), with $N_x = 512$ and $n = 11$, we have $N_i = 20$, which gives a total number of about 2780 op/pix.

If we take into account the directions, the total number is $15\log_2 N_x + 445 + (35 + 8n)N_i$ op/pix (about 3080 op/pix for the chosen image).

4.5 Satellite image deconvolution results

4.5.1 Evaluation of the restoration quality

We do not use the classical Signal to Noise Ratio (SNR) computation, because it usually consists of averaging signal and noise variances over the whole image. As human vision is not sensitive to the noise in the same manner in homogeneous areas and along the edges [20], we should define a quality criterion which penalizes the noise more in constant areas (where noise peaks clearly appear).

Let us define an Adaptive Signal to Noise Ratio (ASNR) where, for each pixel, the noise and the signal are divided by the magnitude of the local derivative of the image g. This derivative is approximated by averaged pixel differences w.r.t. rows and columns, ∇_x and ∇_y. Thus, we allow some noise to affect the edges if they are well-defined, i.e. if they correspond to high pixel differences.

$$\text{ASNR}(X) = 10\ \log \frac{\sum_{i,j} \frac{\mathcal{X}_{ij}^2}{1+g_{ij}^2}}{\sum_{i,j} \frac{(\mathcal{X}_{ij} - X_{ij})^2}{1+g_{ij}^2}} \quad \text{where} \quad g_{ij} = (\nabla_x \mathcal{X})_{ij}^2 + (\nabla_y \mathcal{X})_{ij}^2 \qquad (4.8)$$

4.5.2 Observed data and resulting images

- Fig. 4.2 shows the observed image, provided by the French Space Agency (CNES). It is a SPOT 5 simulation at 2.5m resolution, simulated by a convolution with the kernel h (see Fig. 1.1), and by adding a white noise, approximated by a Gaussian white noise with a standard deviation $\sigma \simeq 1.35$. The resulting ASNR is 21.2 dB.

- Fig. 4.3 shows the restored image, processed with the automatic algorithm 4.4.1. The resulting ASNR is 25.2 dB.

- Fig. 4.4 shows the original image, provided by the French Space Agency (CNES).

- Fig. 4.5 shows the error image, i.e. the difference between the restored and the original image, amplified by a factor 2.

- Fig. 4.6 and Fig. 4.6 show original, blurred and restored images, with the same convolution kernel and the same noise as in Fig. 4.2, for two areas of a satellite image of Cayenne (SPOT 3, resolution 10m, courtesy of Univ. of Marne-la-Vallée).

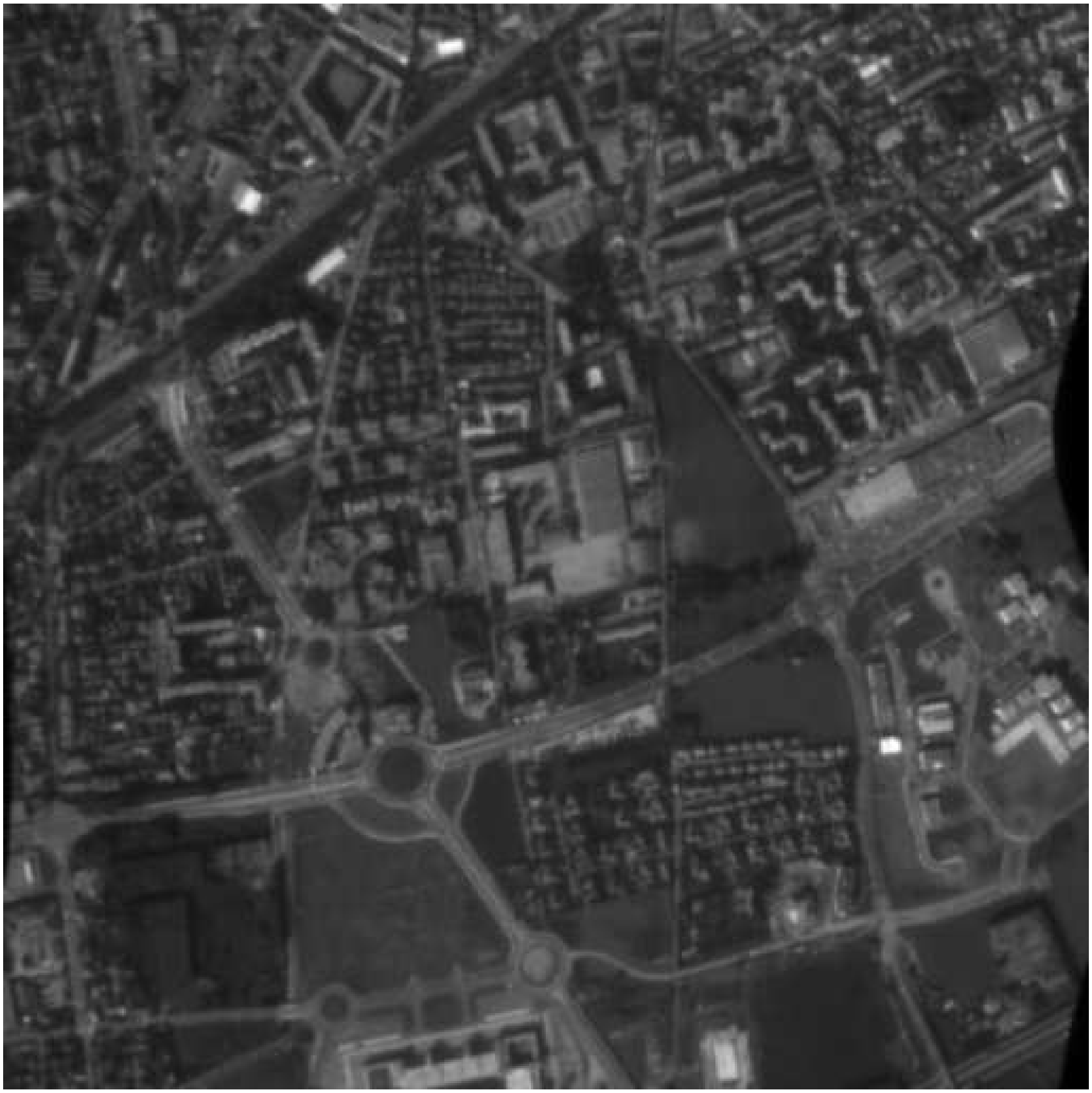


Figure 4.2: *Observed image of Nîmes (see Fig. 1.1 for* h*),* 512×512*, 256 grey levels ©CNES - ASNR=21.2 dB*

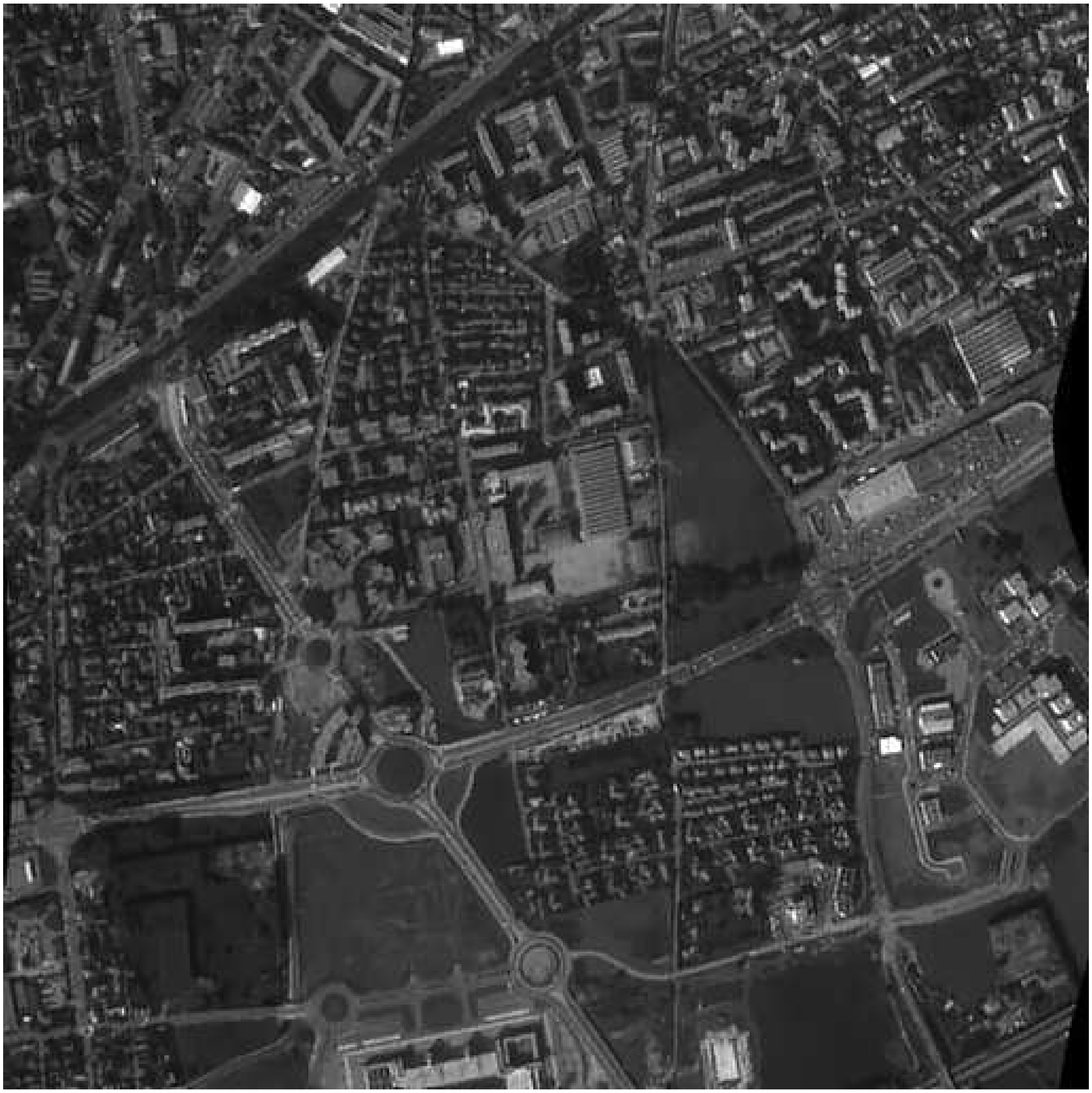


Figure 4.3: *Image of Fig. 4.2 deconvolved with algorithm 4.4.1 (DEPA) - ASNR=25.2 dB*

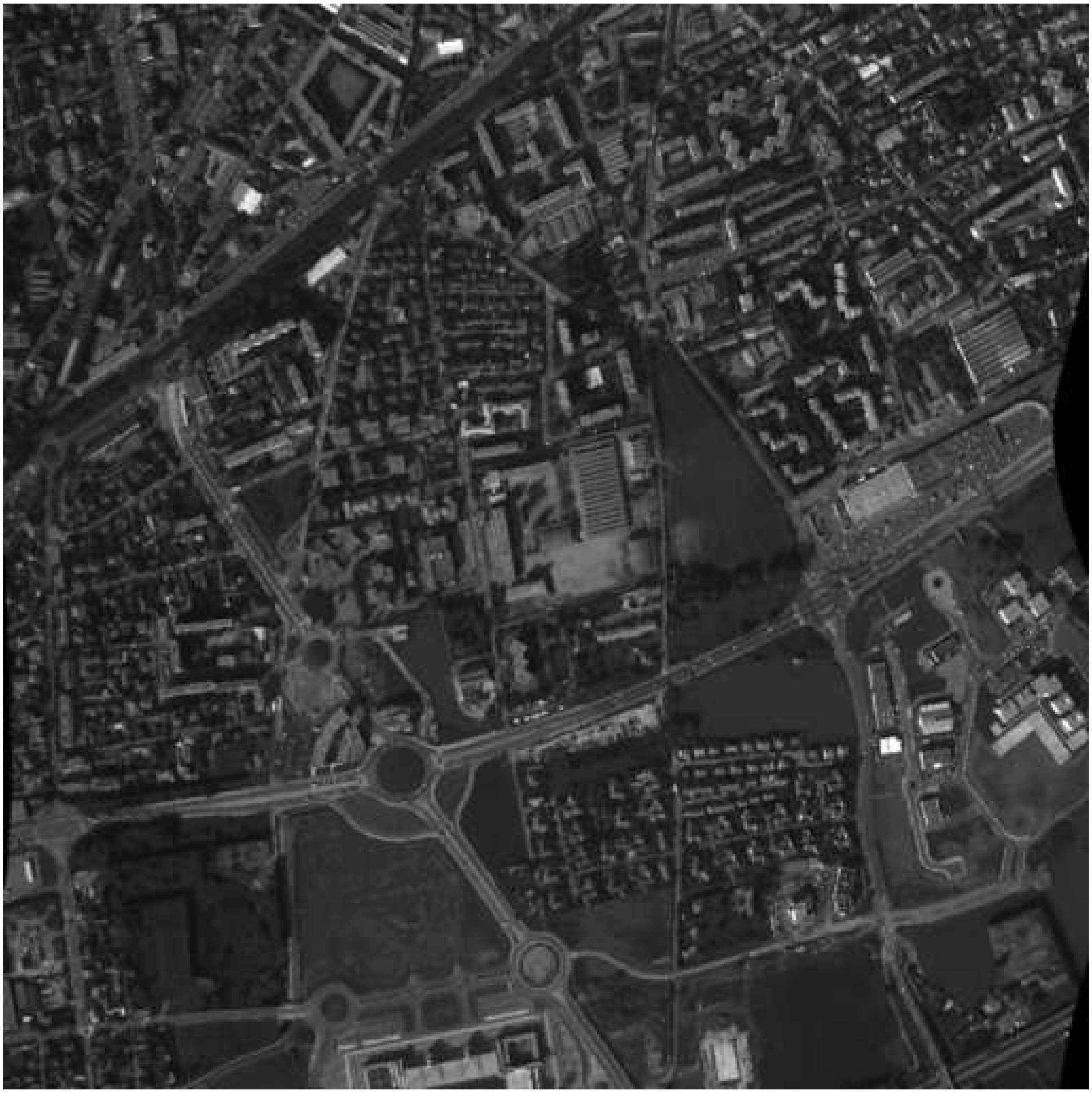


Figure 4.4: *Original image of Nîmes,* 512×512*, 256 grey levels ©CNES*

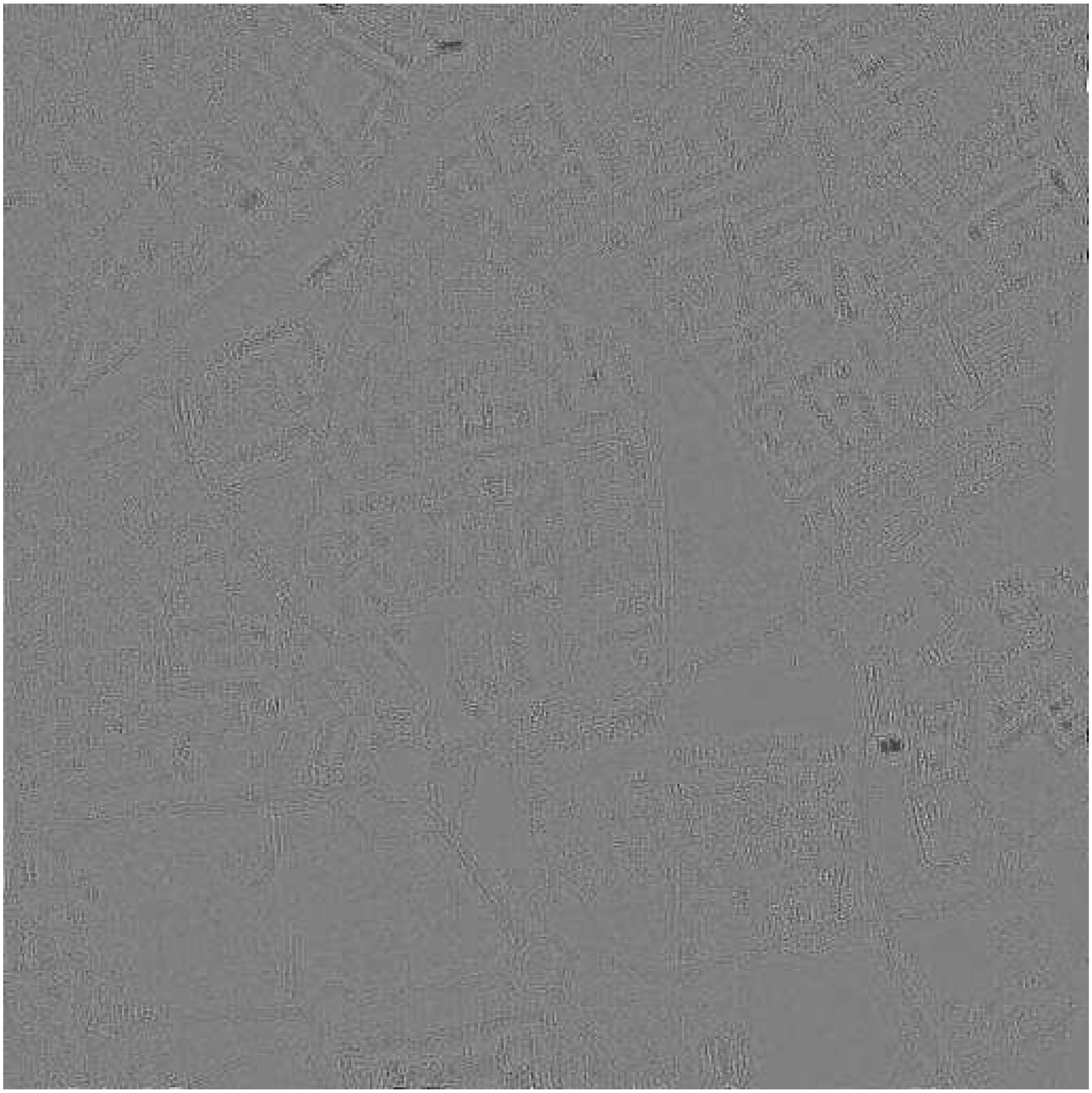


Figure 4.5: *Error image : difference between images of Fig. 4.3 and Fig. 4.4, multiplied by 2*

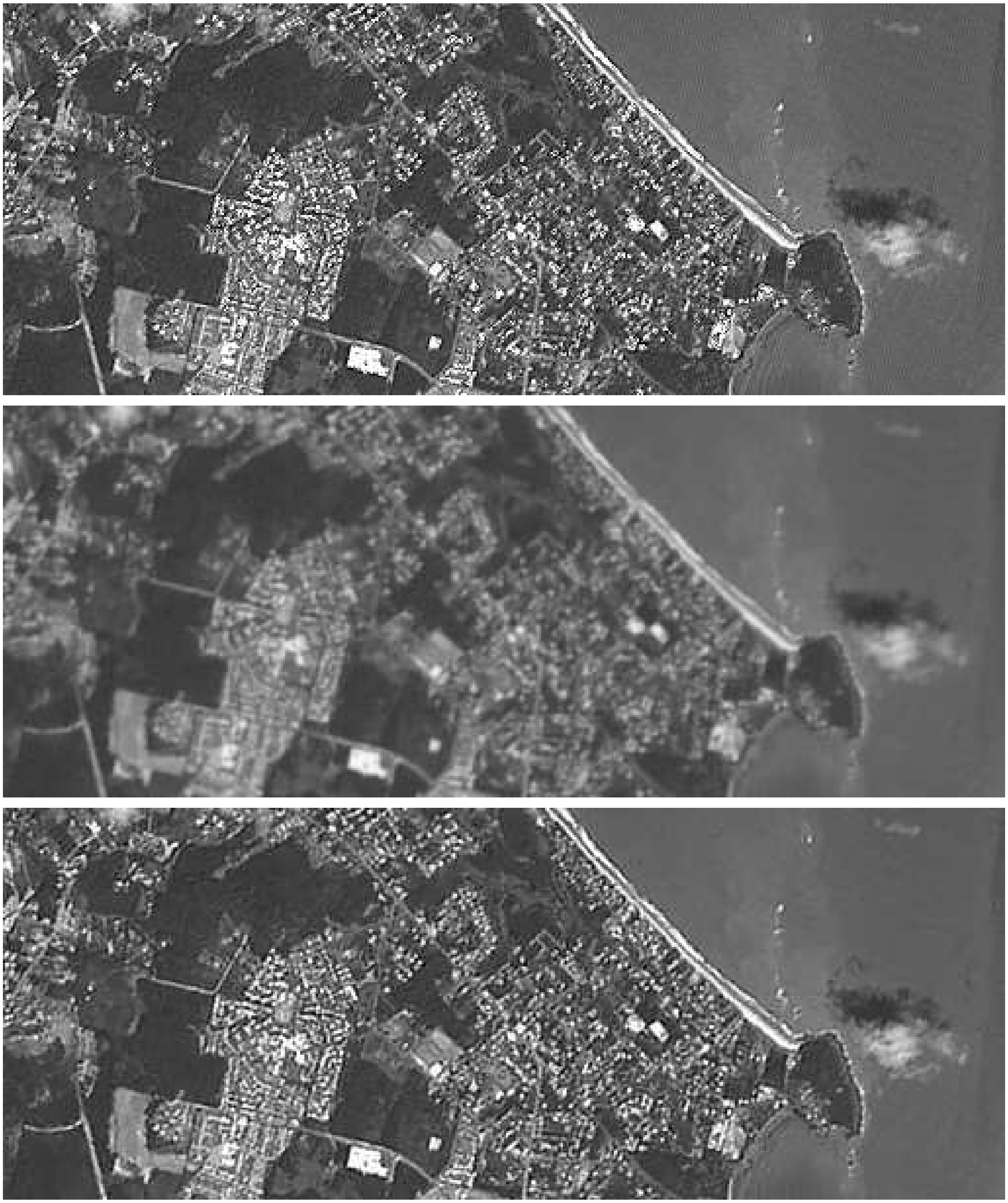


Figure 4.6: *512×200 area extracted from Cayenne (SPOT 3, resolution 10m, courtesy of Univ. of Marne-la-Vallée). Up : original, middle : blurred and noisy (same h and σ as in Fig. 4.2), bottom : deconvolved with algorithm 4.4.1 (DEPA)*

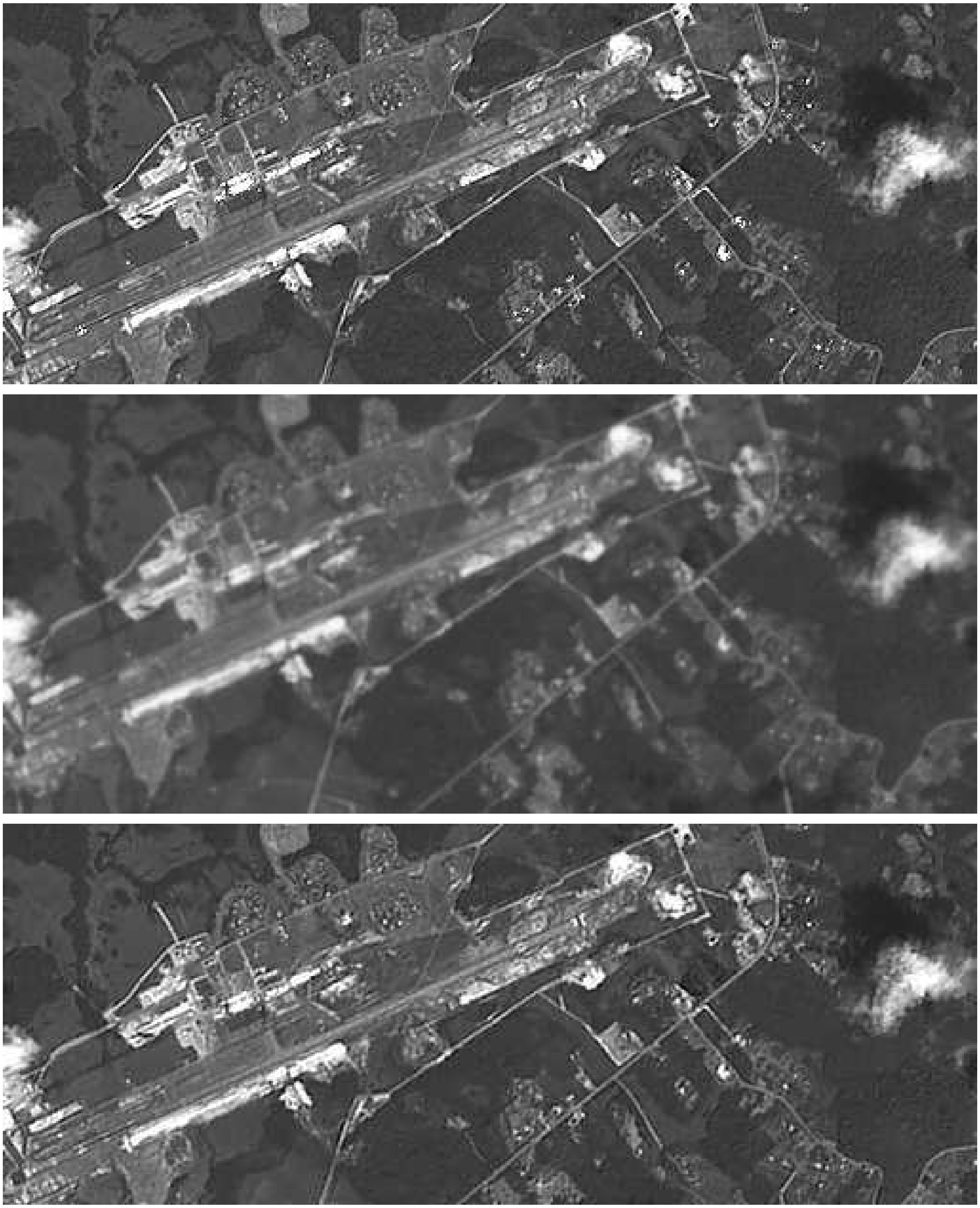


Figure 4.7: *512×200 area extracted from Cayenne (SPOT 3, resolution 10m, courtesy of Univ. of Marne-la-Vallée). Up : original, middle : blurred and noisy (same h and σ as in Fig. 4.2), bottom : deconvolved with algorithm 4.4.1 (DEPA)*

4.6 Comparison with other methods

4.6.1 Wiener filter and quadratic regularization

It is possible to use a regularization method involving a quadratic function [40], consisting of minimizing the following criterion to find the estimate $\hat{X}$:

$$\hat{X} = \arg\inf_X \ \|Y - HX\|^2/2\sigma^2 + b\left(\|D_x X\|^2 + \|D_y X\|^2\right) \tag{4.9}$$

where D_x and D_y represent derivative operators w.r.t. columns and lines, and b is the regularizing parameter. We estimate the optimal value of b with the method described in section 1.3.2. Using this method for the image of Fig. 4.2, we obtain $b \simeq 0.005$. The computation of $\hat{X}$ is easily achieved in the Fourier domain as follows :

$$\mathcal{F}[\hat{X}] = \frac{\mathcal{F}[h]^* \mathcal{F}[Y]}{|\mathcal{F}[h]|^2/2\sigma^2 + b\left(|\mathcal{F}[d_x]|^2 + |\mathcal{F}[d_y]|^2\right)} \tag{4.10}$$

where d_x and d_y are the generators of the block circulant operators D_x and D_y.

This provides the same results as the Wiener filter [19], which gives the same results. It is also equivalent to isotropic diffusion [36]. The edges are filtered as well as the noise, as seen on Fig. 4.8. Therefore it is impossible to obtain sharp details and noisefree homogeneous areas at the same time. Thus, the ASNR remains small (about 19.7 dB) because of the insufficient noise filtering in homogeneous areas.

4.6.2 The RHEA algorithm

The RHEA algorithm has been presented in [22, 23]. It consists of minimizing the following non-quadratic criterion :

$$\hat{X} = \arg\inf_X \ \|Y - HX\|^2/2\sigma^2 + \lambda^2 \sum_{i,j} \left[\varphi\left((D_x X)_{ij}/\delta\right) + \varphi\left((D_y X)_{ij}/\delta\right)\right] \tag{4.11}$$

where φ is a non-quadratic function, whose behaviour enables to preserve the edges while filtering the noise only along these edges and in constant areas. The parameters λ and δ are automatically estimated. The resulting image is represented in Fig. 4.9 and exhibits sharp edges, compared to the previous one. However, some noise remains in homogeneous regions and textures are strongly attenuated.

4.6.3 Wavelet packet thresholding

Using a real wavelet packet thresholding algorithm, such as the one proposed by Kalifa [25], enables to recover the textures and to obtain very smooth homogeneous areas (see Fig. 4.10). However, the edges are less sharp than with the DEPA algorithm.

Using a complex wavelet packet thresholding such as COWPATH [24] provides better results (see Fig. 4.11), but the sharpness of the edges can still be improved by using an adaptive parameter estimation on them, as it is the case in the DEPA algorithm.

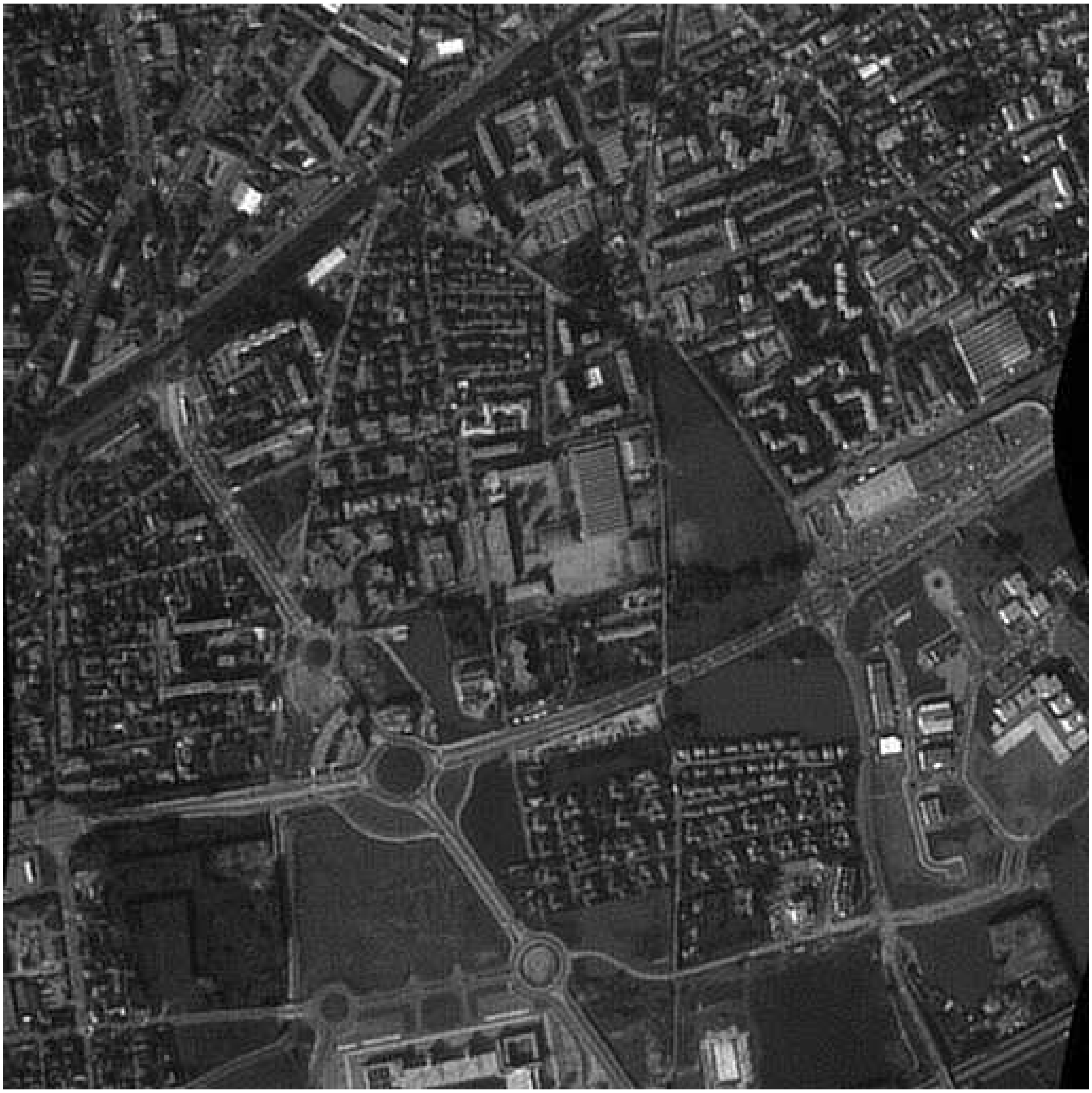


Figure 4.8: *Image of Fig. 4.2 deconvolved with quadratic regularization (Tikhonov) and optimal hyperparameter* $b = 0.0005$ *- ASNR=19.7 dB*

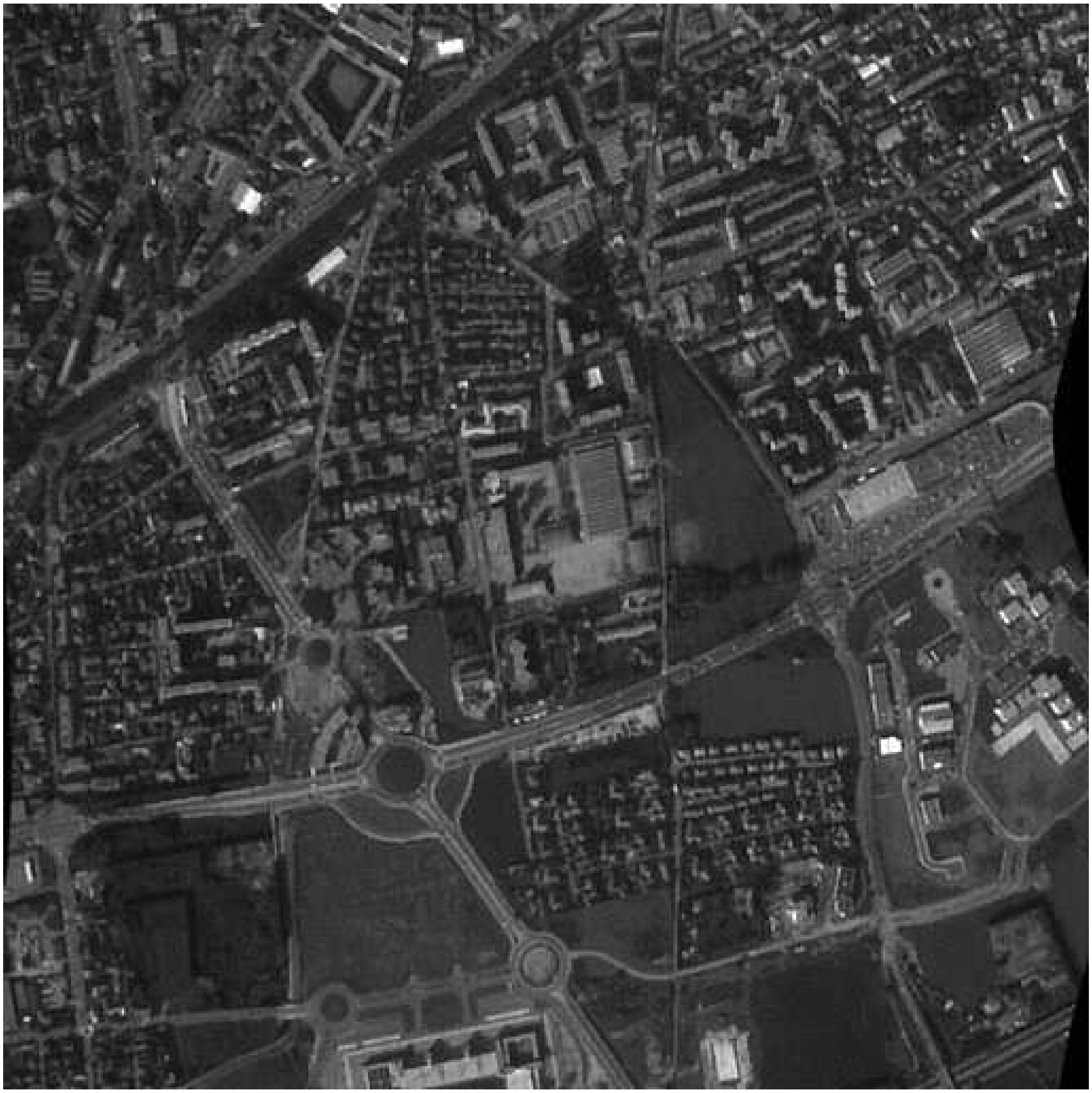


Figure 4.9: *Image of Fig. 4.2 deconvolved with RHEA algorithm - ASNR=24.5 dB*

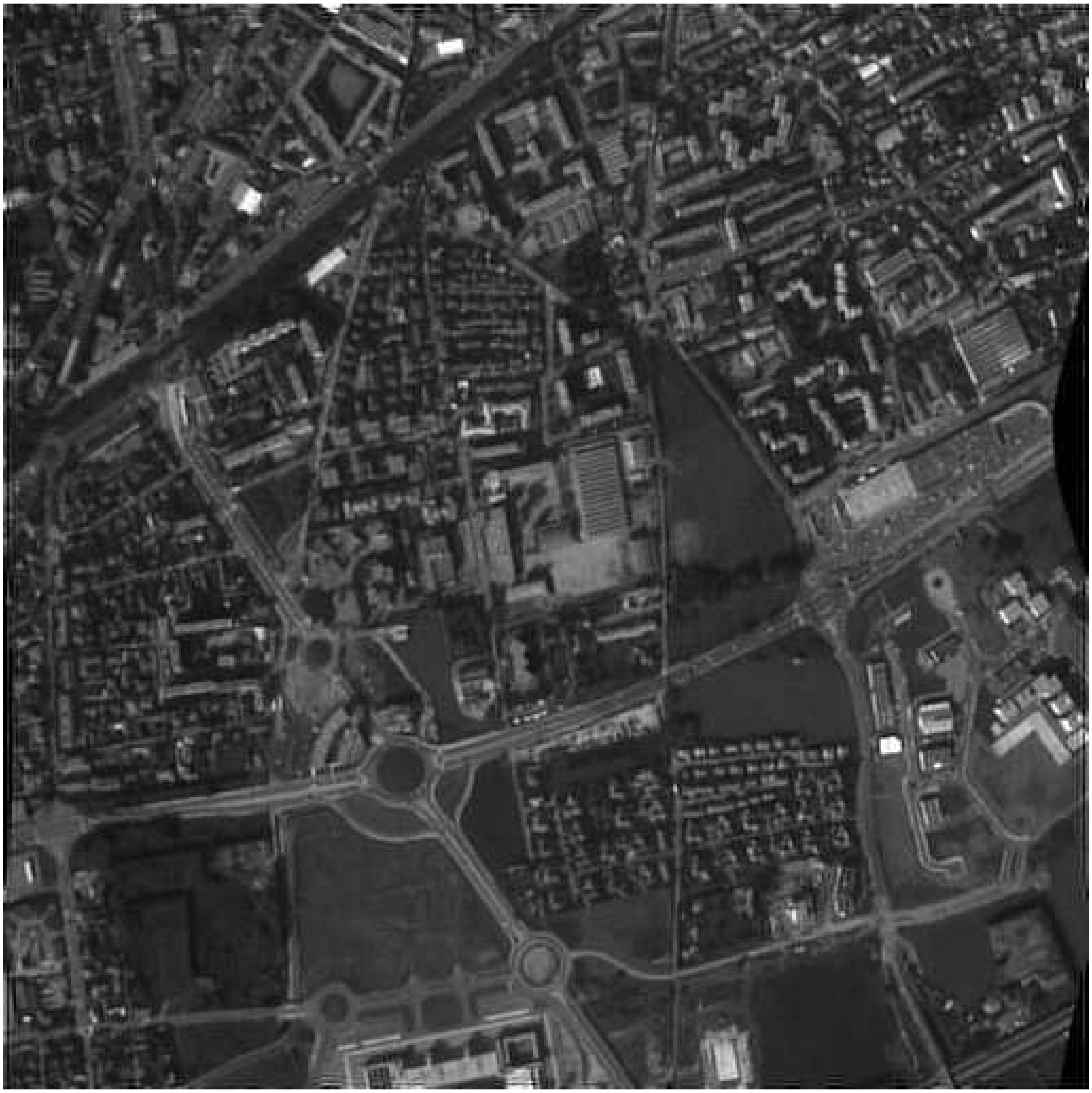


Figure 4.10: *Image of Fig. 4.2 deconvolved by a real wavelet packet thresholding algorithm [25] - ASNR=24.6 dB*

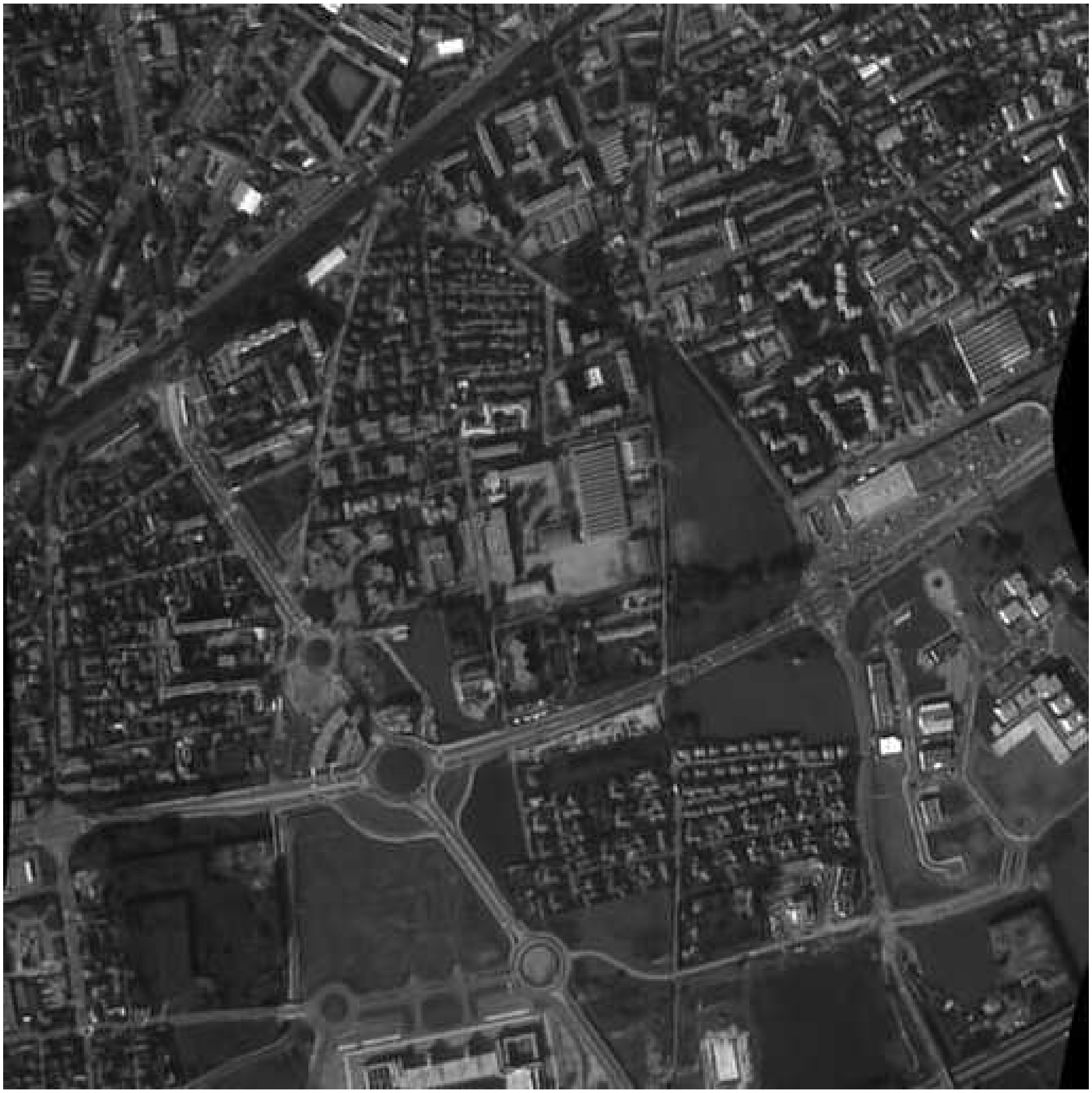
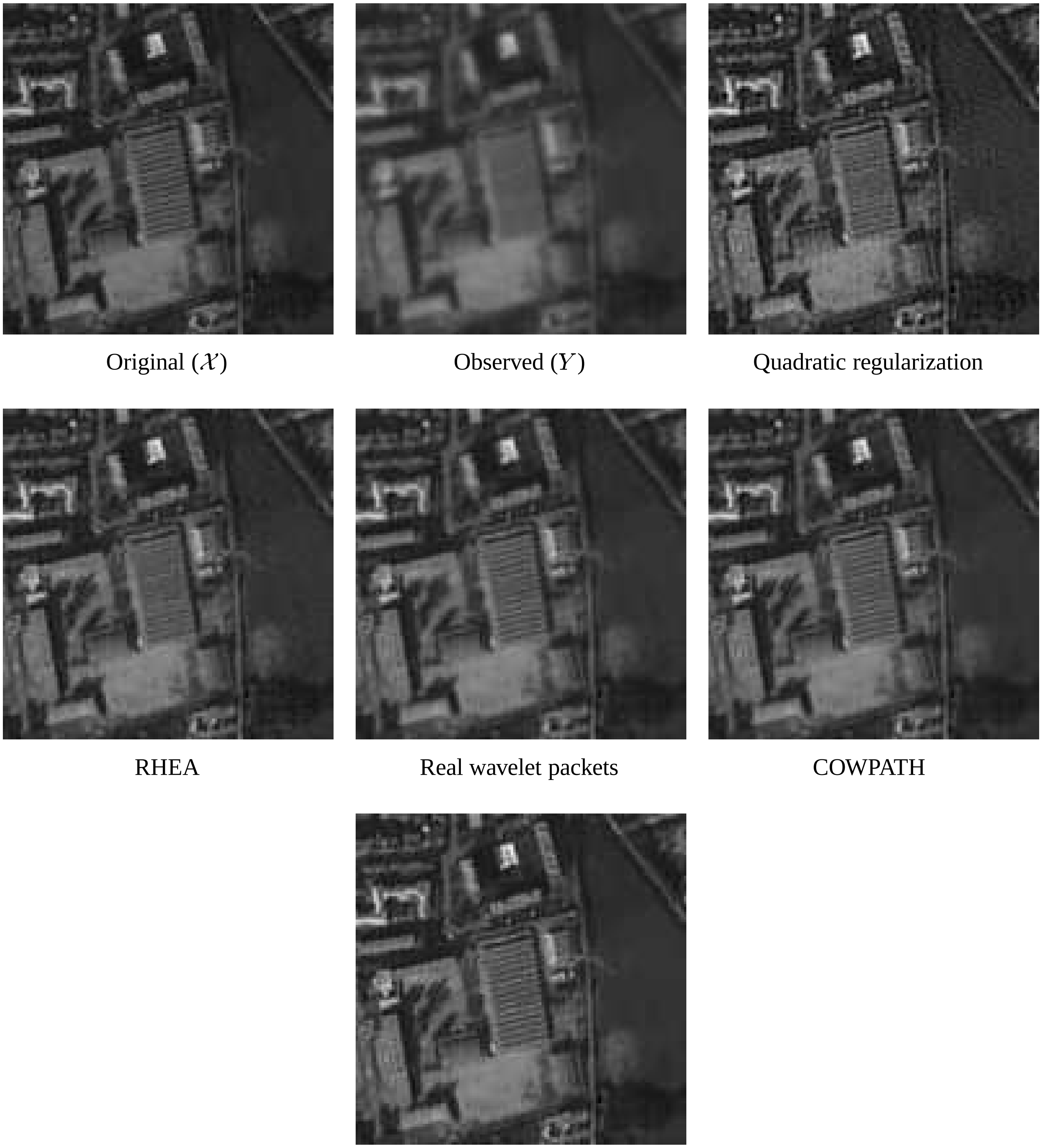


Figure 4.11: *Image of Fig. 4.2 deconvolved with COWPATH [24] - ASNR=25.4 dB*



Original ($\mathcal{X}$)

Observed (Y)

Quadratic regularization

RHEA

Real wavelet packets

COWPATH

DEPA

Figure 4.12: *Zoom on a 128×128 area extracted from Fig. 4.2: comparison of the different deconvolution methods (see previous figures for entire images)*

Chapter 5

Conclusion and future work

5.1 Concluding remarks

We have shown in this report that, in the case of an inhomogeneous regularizing model, the MLE on observed data, which is a usual approach for parameter estimation within a Bayesian framework, is not a robust estimator. We use the MLE on the complete data instead, supposing we know a good approximation of the original image. This image approximation could be obtained by a wavelet-based thresholding algorithm, like the one presented in [24].

This technique is robust, and gives very satisfying deconvolution results on high resolution satellite data. Furthermore, the resulting images exhibit much sharper details than images produced by concurrent deconvolution algorithms.

5.2 Furher developements

5.2.1 A more accurate modeling

The model described by equation (2.1) is the simplest inhomogeneous model. The results could be improved by using a more accurate modeling, for example by using a Generalized Gaussian MRF instead of a simply Gaussian field.

To better take into account the textures, which are characterized by longer distance interactions than edges, it should be preferable to define a model with higher order interations. The current model is limited to the 4 nearest neighbours, and the parameters are related to the first order derivatives. More complicated regularizing functionals, involving higher order derivatives, should be investigated. In spite of their complexity, they would probably enable to restore more regular textures and edges.

For example, the approach used in section 4.3.1 to take into account the diagonal directions can be developed, by directly studying the estimation of the parameters associated to the diagonal gradients.

5.2.2 Taking into account the properties of the reference image

The reference (or approximation image) is used to estimate the adaptive parameters by complete data MLE. Thus, the estimation results strongly depend on the quality of this approximation. We have shown that to get a robust estimate, it is necessary to threshold the residual noise which contaminates the gradients of this approximation image. It could be preferable to take into account the statistics of the residual noise of this image, to better estimate the adaptive parameters. It has to be checked if the estimated values of the parameters are more accurate in this case and if they provide better deconvolution results.

5.2.3 Towards an iterative hybrid algorithm

The method described in this report essentially consists of two steps - approximation by a wavelet-based method, and adaptive deconvolution. We wonder whether the output of the algorithm could be injected back into the first step, to obtain a better approximation, and then to provide better parameter estimates. In this case, it could be possible to build an iterative algorithm. The stability of such a method is not sure, and convergence studies have to be carried out.

Bibliography

[1] G. Aubert and L. Vese. A variational method in image recovery. *SIAM Journal of Num. Anal.*, 34(5):1948–1979, Oct. 1997.

[2] R.G. Aykroyd. Bayesian estimation for homogeneous and inhomogeneous Gaussian random fields. Technical Report STAT-96/17, Department of Statistics, University of Leeds, 1996.

[3] R.G. Aykroyd and P.J. Green. Global and local priors, and the location of lesions using gamma-camera imagery. *Phil. Trans. R. Soc. Lond. A*, 332:323–342, 1991.

[4] J. Besag. On the statistical analysis of dirty pictures. *Journal of Royal Statistic Society*, B(68):259–302, 1986.

[5] C. Bouman and K. Sauer. A generalized gaussian image model for edge-preserving map estimation. *IEEE Trans. on IP*, 2(3):296–310, July 1993.

[6] B. Chalmond. Image restoration using an estimated Markov model. *Signal Processing*, 15:115–129, 1988.

[7] P. Charbonnier. *Reconstruction d'image : régularisation avec prise en compte de discontinuités*. PhD dissertation, Laboratoire I3S, UPRES-A 6070 du CNRS, University of Nice-Sophia Antipolis, France, Sept. 1994.

[8] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic Edge-Preserving Regularization in Computed Imaging. *IEEE Trans. on IP*, 6(2):298–311, Feb. 1997.

[9] R. Coifman, Y. Meyer, and M. Wickerhauser. *Wavelet analysis and signal processing*, pages 153–178. John and Barlett B. Ruskai et al. Editors, 1992.

[10] G. Demoment. Image reconstruction and restoration : overview of common estimation structures and problems. *IEEE Trans. on ASSP*, 37(12):2024–2036, Dec. 1989.

[11] X. Descombes, R. Morris, J. Zerubia, and M. Berthod. Estimation of Markov random field prior parameters using Markov Chain Monte Carlo Maximum Likelihood. *IEEE Trans. on IP*, 8(7):954–963, July 1999.

[12] D. Donoho. Denoising by soft thresholding. *IEEE Trans on IT*, 41:613–627, 1995.

[13] D. Donoho and I. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 81:425–455, 1994.

[14] D. Donoho, I. Johnstone, G. Kerkyacharian, and D. Picard. Wavelet shrinkage: asymptopia. *J. Roy. Stat. Soc.*, B 57(2):301–369, 1995.

[15] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Trans. on PAMI*, 14(3):367–383, March 1992.

[16] D. Geman and C. Yang. Nonlinear Image Recovery with Half-Quadratic Regularization. *IEEE Trans. on IP*, 4(7):932–946, July 1995.

[17] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images. *IEEE Trans. on PAMI*, 6(6):721–741, Nov. 1984.

[18] J. Hadamard. *Lectures on Cauchy's Problem in Linear Partial Differential Equations*. Yale University Press, New Haven, 1923.

[19] C. W. Helstrom. Image restoration by the method of least squares. *J. Opt. Soc. Amer.*, 1967.

[20] D. Hubel. *Eye, brain, and vision*. Scientific american Library, 1988.

[21] J. Idier. Problèmes inverses en restauration de signaux et d'images. *Mémoire d'habilitation à diriger les recherches*, Juil. 2000.

[22] A. Jalobeanu, L. Blanc-Féraud, and J. Zerubia. Estimation d'hyperparamètres pour la restauration d'images satellitaires par une méthode MCMCML. *INRIA Research Report*, 3469, Aug. 1998.

[23] A. Jalobeanu, L. Blanc-Féraud, and J. Zerubia. Hyperparameter estimation for satellite image restoration by a MCMCML method. In Springer, editor, *LNCS - EMMCVPR*, York, July 1999.

[24] A. Jalobeanu, L. Blanc-Féraud, and J. Zerubia. Satellite image deconvolution using complex wavelet packets. *INRIA Research Report*, June 2000.

[25] J. Kalifa. *Restauration minimax et déconvolution dans une base d'ondelettes miroirs*. PhD thesis, École Polytechnique, May. 1999.

[26] J. Kalifa and S. Mallat. Restauration d'images par paquets d'ondelettes. In *GRETSI*, Grenoble, France, 1997.

[27] J. Kalifa and S. Mallat. Wavelet packet deconvolutions. Technical report, CMAP, Ecole Polytechnique, Palaiseau, France, 1998.

[28] J. Kalifa and S. Mallat. *Bayesian inference in wavelet based methods*, chapter Minimax restoration and deconvolution. Springer, 1999.

[29] S.M. Kay. *Fundamentals of statistical signal processing*. Prentice Hall, 1993.

[30] N. Kingsbury. The dual-tree complex wavelet transform: a new efficient tool for image restoration and enhancement. In *EUSIPCO*, pages 319–322, Rhodes, Greece, 1998.

[31] S. Lakshmanan and H. Derin. Simultaneous parameter estimation and segmentation of Gibbs random field using simulated annealing. *IEEE Trans. on PAMI*, 11(8):322–336, Aug. 1989.

[32] S. Mallat. *A Wavelet tour of signal processing*. Academic Press, 1998.

[33] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. of Chem. Physics*, 21:1087–1092, 1953.

[34] A.M. Mood, F.A. Graybill, and D.C. Boes. *Introduction to the theory of statistics*. Mc Graw Hill, 1974.

[35] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on PAMI*, 12(7):629–639, July 1990.

[36] Bart M. Ter Haar Romeny, editor. *Geometry-driven diffusion in computer vision*. Kluwer Academic Publishers, 1994.

[37] S.S. Saquib, C.A. Bouman, and K. Sauer. A non-homogeneous MRF model for multiresolution Bayesian estimation. In *IEEE ICIP*, 1996.

[38] S.S. Saquib, C.A. Bouman, and K. Sauer. ML parameter estimation for Markov random fields, with applications to Bayesian tomography. *IEEE Trans. on IP*, 7(7):1029–1044, July 1998.

[39] M. Sigelle. Simultaneous image restoration and hyperparameter estimation for incomplete data by a cumulant analysis. *INRIA Research Report*, 3249, Sept. 1997.

[40] A.N. Tikhonov. Regularization of incorrectly posed problems. *Sov. Math. Dokl.*, 4:1624–1627, 1963.

[41] S. Tramini, M. Antonini, M. Barlaud, and G. Aubert. Quantization noise removal for optimal transform decoding. In *IEEE ICIP*, Chicago, USA, Oct. 1998.

[42] L. Younes. Parametric inference for imperfectly observed Gibbsian fields. *Probability Theory and Related Fields*, 82:625–645, 1989.

Appendix A

Likelihood computation for a homogeneous GMRF

Prior distribution

The partition function of the prior distribution is :

$$Z_X = \int_\Omega e^{-\Phi(X)} dX = \int_\Omega e^{-\frac{1}{2}X^t \Sigma_X^{-1} X} dX = 2\kappa \sqrt{\frac{(2\pi)^{N_x N_y - 1}}{\det' \Sigma_X^{-1}}}$$

$$\text{with } \Sigma_X^{-1} = 2b\,(D_x^t D_x + D_y^t D_y)$$

Σ_X^{-1} is diagonalized by a Fourier transform, so we integrate over the transformed variables. $\det'$ is the product of all non-null eigenvalues ($N_x N_y - 1$). Only the first eigenvalue is null. It corresponds to the eigenvector defined by the sum of all pixels (zero frequency). This determines the choice of the state space Ω :

$$\Omega = \mathbb{R}^{N_x N_y} \cap \Big\{ -\kappa \le \sum_{i,j} x_{ij} \le \kappa \Big\}$$

This space is constructed by taking all the points located between the two hyperplanes defined by $\sum_{i,j} x_{ij} = \pm\kappa$. This avoids the prior distribution to be improper, and enables the above integral to be proportional to κ. The bound κ is taken sufficiently large to allow any satellite image to belong to Ω, and remains fixed.

$$\det{}' \Sigma_X^{-1} = \prod_{(i,j) \ne (0,0)} 2b\, c_{ij}$$

$$\text{where} \quad c_{ij} = |\mathcal{F}[D_x]|_{ij}^2 + |\mathcal{F}[D_y]|_{ij}^2$$

Posterior distribution

In the same manner, we can exactly compute Z_Y :

$$Z_Y = \int_\Omega e^{-U(X)} dX = \int_\Omega e^{-||Y-HX||^2/2\sigma^2 - b\,(||D_x X||^2 + ||D_y X||^2)} dX$$

We introduce $\hat{X}$, which is the minimizer of the quadratic form $U(X)$. This enables to put the constant terms out of the integral :

$$Z_Y = e^{-U(\hat{X})} \int_\Omega e^{-\frac{1}{2}(X-\hat{X})^t \Sigma_Y^{-1}(X-\hat{X})} dX \simeq e^{-U(\hat{X})} \sqrt{\frac{(2\pi)^{N_x N_y}}{\det \Sigma_Y^{-1}}}$$

$$\text{with} \quad \Sigma_Y^{-1} = H^t H/\sigma^2 + 2b\left(D_x^t D_x + D_y^t D_y\right)$$

The integration is done by diagonalizing the quadratic form in the Fourier space. All the transformed variables over $\mathbb{R}$, excepted for the first one, integrated over $[-\kappa, \kappa]$. Therefore, the approximation is accurate, because κ is sufficiently large.

Let us denote $w_{ij} = |\mathcal{F}[h]|^2_{ij}/2\sigma^2$. Then, we obtain :

$$\det \Sigma_Y^{-1} = \prod_{i,j} 2\left(bc_{ij} + w_{ij}\right)$$

Let us now compute $U(\hat{X})$.

$\hat{X}$ is the image reconstructed with a quadratic regularization :

$$\mathcal{F}[\hat{X}] = \frac{\mathcal{F}[h]^* \mathcal{F}[Y]}{2\sigma^2(b\,c + w)}$$

The energy U is :

$$U(\hat{X}) = \sum_{i,j} \Big(|\mathcal{F}[Y]_{ij} - \mathcal{F}[h]\mathcal{F}[\hat{X}]_{ij}|^2/2\sigma^2 + b\left(|\mathcal{F}[d_x]\mathcal{F}[\hat{X}]_{ij}|^2 + |\mathcal{F}[d_y]\mathcal{F}[\hat{X}]_{ij}|^2\right)\Big)$$

We finally obtain

$$U(\hat{X}) = \frac{1}{2\sigma^2}\left(||Y||^2 - \sum_{i,j} \frac{w_{ij}|\mathcal{F}[Y]_{ij}|^2}{b\,c_{ij} + w_{ij}} \right)$$

and the likelihood, proportional to Z_Y/Z_X, becomes :

$$P(Y \mid b) \propto e^{-U(\hat{X})} \sqrt{\prod_{i,j} \frac{b\,c_{ij}}{b\,c_{ij} + w_{ij}}}$$

which finally gives the log-likelihood expression :

$$\log P(Y \mid b) = \sum_{i,j} \left[\frac{1}{2} \log\left(\frac{b\,c_{ij}}{b\,c_{ij} + w_{ij}} \right) + \frac{w_{ij}|\mathcal{F}[Y]_{ij}|^2}{2\sigma^2(b\,c_{ij} + w_{ij})} \right] + cst. \tag{A.1}$$

The log-likelihood derivative w.r.t. the parameter b is :

$$\frac{\partial \log P(Y \mid b)}{\partial b} = \frac{1}{2} \sum_{i,j} \left(\frac{1}{b} - \frac{c_{ij}}{b\,c_{ij} + w_{ij}} - \frac{c_{ij} w_{ij} |\mathcal{F}[Y]_{ij}|^2}{\sigma^2(b\,c_{ij} + w_{ij})^2} \right) \tag{A.2}$$

Appendix B

MLE variance calculus

Homogeneous nonlinear model

The expression of the Likelihood $P(Y \,|\, \theta) \propto Z_Y/Z_X$ and equation (2.27) enable to calculate the vector α, remembering property (1.9) for the partition function derivative :

$$\alpha = \nabla_Y \left(\frac{\partial \log Z_Y}{\partial \theta} \right) = \nabla_Y \left(\frac{1}{Z_Y} \int_\Omega - \frac{\partial \Phi(X)}{\partial \theta} e^{-||Y-HX||^2/2\sigma^2 - \Phi(X)} dX \right)$$

The prior partition function is not used since it does not depend on Y.

$$\alpha = \frac{\nabla_Y Z_Y}{Z_Y^2} \int_\Omega \frac{\partial \Phi(X)}{\partial \theta} e^{-\cdots} dX - \frac{1}{Z_Y} \int_\Omega - \frac{Y-HX}{\sigma^2} \frac{\partial \Phi(X)}{\partial \theta} e^{-\cdots} dX$$

$$\text{and} \quad \frac{1}{Z_Y} \nabla_Y Z_Y = \int_\Omega - \frac{Y-HX}{\sigma^2} e^{-\cdots} dX$$

therefore :

$$\alpha = E_Y \left[\frac{Y-HX}{\sigma^2} \frac{\partial \Phi(X)}{\partial \theta} \right] - E_Y \left[\frac{Y-HX}{\sigma^2} \right] \, E_Y \left[\frac{\partial \Phi(X)}{\partial \theta} \right]$$

which gives the following expression for α :

$$\alpha = \frac{1}{\sigma^2} \left(E_Y \left[\frac{\partial \Phi(X)}{\partial \theta} \right] \, E_Y[HX] - E_Y \left[\frac{\partial \Phi(X)}{\partial \theta} \, HX \right] \right)$$

This expression is simplified by splitting it into a mean $E_Y[X]$ and a random part X^r :

$$X = E_Y[X] + X^r$$

Then we have :

$$\alpha = -\frac{1}{\sigma^2} E_Y \left[\frac{\partial \Phi(X)}{\partial \theta} \, HX^r \right]$$

The variance of the MLE is finally given by :

$$\mathrm{var}(\hat{\theta}) \simeq \sigma^2 \frac{\|\alpha\|^2}{\beta^2} = \frac{1}{\sigma^2} \frac{\left\| E_Y \left[\frac{\partial \Phi(X)}{\partial \theta} \, HX^r \right] \right\|^2}{\left(E_Y \left[\frac{\partial \Phi(X)}{\partial \theta}^2 \right] - E_Y \left[\frac{\partial \Phi(X)}{\partial \theta} \right]^2 - E_X \left[\frac{\partial \Phi(X)}{\partial \theta}^2 \right] + E_X \left[\frac{\partial \Phi(X)}{\partial \theta} \right]^2 \right)^2}$$

Homogeneous Gaussian model

In this case, we can replace $E_Y[X]$ by the maximizer $\hat{X}$, since the Gaussian law is symmetric around its mode $\hat{X}$.

$$\text{var}(\hat{b}) \simeq \frac{1}{\sigma^2\beta^2} \left\| E_Y \left[\frac{\partial \Phi(X)}{\partial b} \, HX^r \right] \right\|^2 \quad \text{where} \quad X^r = X - \hat{X} \qquad \text{(B.1)}$$

First step : compute β by using the expression of the log-likelihood derivative of equation (A.2).

$$\beta = \frac{\partial^2 \log P(Y \mid b)}{\partial b^2} = \frac{1}{2} \sum_{i,j} \left(-\frac{1}{b^2} + \frac{c_{ij}^2}{(b\, c_{ij} + w_{ij})^2} + 2 \frac{c_{ij}^2 w_{ij} |\mathcal{F}[Y]_{ij}|^2}{\sigma^2 (b\, c_{ij} + w_{ij})^3} \right) \qquad \text{(B.2)}$$

Second step : compute the L_2 norm of the expectation of equation (B.1). It can also be evaluated by computing sums in the frequency space, using the Parseval theorem. Thus, we compute the Fourier transform of the term inside the expectation, i.e. $\mathcal{F}\left[HX^r \frac{\partial\Phi(X)}{\partial b}\right]$. The derivative of the regularization term Φ w.r.t. the parameter b is $||D_x X||^2 + ||D_y X||^2$, and we again split X into a random part and a mean part, $X = X^r + \hat{X}$, then we obtain :

$$\mathcal{F}\left[HX^r \frac{\partial \Phi(X)}{\partial b}\right] = \mathcal{F}\left[HX^r \left(||D_x(X^r + \hat{X})||^2 + ||D_y(X^r + \hat{X})||^2 \right)\right]$$
$$= \mathcal{F}[h]\mathcal{F}[X^r] \left(||\mathcal{F}[D_x](\mathcal{F}[X^r] + \mathcal{F}[\hat{X}])||^2 + ||\mathcal{F}[D_y](\mathcal{F}[X^r] + \mathcal{F}[\hat{X}])||^2 \right)$$

The random part X^r is given by $\mathcal{F}[X^r] = \sqrt{W}R$, where R is a white Gaussian random vector, whose elements are independent, with variance $1/2$ and mean 0. The mean value $\hat{X}$, i.e. the restored image with parameter b, is given by $\mathcal{F}[\hat{X}] = W\mathcal{F}[h]^*\mathcal{F}[Y]/2\sigma^2$, where the vector W is defined by $W_{ij} = (b\, c_{ij} + w_{ij})^{-1}$. Then, we have :

$$\mathcal{F}[h]\sqrt{W}R \left(\|\mathcal{F}[D_x]\|^2 + \|\mathcal{F}[D_y]\|^2 \right) \left(\sqrt{W}R + W\mathcal{F}[h]^*\mathcal{F}[Y]/2\sigma^2 \right)$$

The expectation $E_Y[\,]$ of the expression (B.1), for the posterior law of X, is now given by the expectation $E[\,]$ for the Gaussian law of R. The (i,j) component is given by :

$$E_Y \left[\mathcal{F}[h]_{ij} \sqrt{W}_{ij} R_{ij} \sum_{k,l} \left(|\mathcal{F}[D_x]|_{kl}^2 + |\mathcal{F}[D_y]|_{kl}^2 \right) \left| \sqrt{W}_{kl} R_{kl} + \frac{W_{kl}\mathcal{F}[h]_{kl}^* \mathcal{F}[Y]_{kl}}{2\sigma^2} \right|^2 \right]$$

Because of the zero-mean of R, and because of the independence hypothesis, only the terms in $R_{ij}R_{kl}$ with $(i,j) = (k,l)$ remain. As $E[R_{ij}^2] = 1/2$, we obtain :

$$E_Y \left[\mathcal{F}\left[HX^r \frac{\partial \Phi}{\partial b} \right] \right]_{ij} = \frac{c_{ij}\mathcal{F}[h]_{ij} W_{ij}^2 (\mathcal{F}[h]_{ij}^* \mathcal{F}[Y]_{ij} + \mathcal{F}[h]_{ij}\mathcal{F}[Y]_{ij}^*)}{4\sigma^2}$$

This expression is simplified by using the symmetry properties of h and Y. We consider that both the observation Y and the kernel h are symmetric w.r.t. lines and columns (this is true for h of fig. 1.1, and Y can be symmetrized to become a $2N_x 2N_y$ image). Therefore we have $\mathcal{F}[h]^* \mathcal{F}[Y] = \mathcal{F}[h]\mathcal{F}[Y]^*$. If h and Y are not symmetric, we take a symmetric approximation of h, and Y is symmetrized w.r.t. rows and columns (here, we only focus on variance computation, not on deconvolution, and we can choose arbitrary images).

Using the same notations c and w, and replacing W by $b\,c + w$, we finally get the approximate expression for var$(\hat{b})$:

$$\text{var}(\hat{b}) \simeq \frac{1}{\sigma^2\beta^2} \left\| E_Y \left[\mathcal{F} \left[H X^r \frac{\partial \Phi}{\partial b} \right] \right] \right\|^2 = \frac{1}{\sigma^2\beta^2} \sum_{ij} \frac{c_{ij}^2 w_{ij}^2 |\mathcal{F}[Y]|_{ij}^2}{(b\,c_{ij} + w_{ij})^4}$$

Inhomogeneous Gaussian model

In the following, we consider the homogeneous approximation. It means that the expressions are evaluated when all the parameters are equal to a single value $\bar{b}$. This is obviously not realistic, but it is needed to make the estimation computationally tractable.

The variance of the inhomogeneous parameters $\{b^x\}$ and $\{b^y\}$ is given by computing the error terms $E[\hat{b}_{ij}^{x,y}] - \hat{b}_{ij}^{x,y}$, which requires solving the system :

$$(\hat{b} - E[\hat{b}])^t \beta + (E[\hat{b}] - \hat{b}^0)^t \beta^0 + (Y - Y^0)^t \alpha \simeq 0$$

where α is a matrix whose lines are defined by :

$$\alpha_{ij}^{x,y} = \nabla_Y \frac{\partial \log P(Y \mid \{b^x\}, \{b^y\})}{\partial b_{ij}^{x,y}} \Bigg|_{Y=Y^0,\, \theta = E[\hat{\theta}]}$$

To compute the Hessian β corresponding to the second derivatives of the log-likelihood we use a stochastic expression corresponding to equation (2.31), with second derivatives w.r.t. different parameters.

We denote β_{ijkl}^{rs} the derivative w.r.t. $\theta_1 = b_{ij}^r$ and $\theta_2 = b_{kl}^s$ where r and s symbolize the directions x or y :

$$\beta_{ijkl}^{rs} = \frac{\partial^2 \log P(Y \mid \{b^x\}, \{b^y\})}{\partial b_{ij}^r \partial b_{kl}^s}$$

We will show that this matrix is *nearly diagonal* when all the parameters are equal.

$$\beta_{ijkl}^{rs} = E_Y \left[\frac{\partial \Phi}{\partial \theta_1} \frac{\partial \Phi}{\partial \theta_2} \right] - E_Y \left[\frac{\partial \Phi}{\partial \theta_1} \right] E_Y \left[\frac{\partial \Phi}{\partial \theta_2} \right] - E_X \left[\frac{\partial \Phi}{\partial \theta_1} \frac{\partial \Phi}{\partial \theta_2} \right] + E_X \left[\frac{\partial \Phi}{\partial \theta_1} \right] E_X \left[\frac{\partial \Phi}{\partial \theta_2} \right]$$

Diagonal terms $(\theta_1 = \theta_2 = \theta)$

As we have $\partial\Phi/\partial\theta = g_\theta^2$ where g_θ is the pixel difference corresponding to the parameter $\theta = b_{ij}^{x,y}$, the diagonal terms are :

$$\beta_{ijij}^{rr} = E_Y[g_\theta^4] - E_Y[g_\theta^2]^2 - E_X[g_\theta^4] + E_X[g_\theta^2]^2$$

X is an inhomogeneous GMRF, but all the evaluations are made when *all the parameters are equal*. Indeed, we take the particular case of an image $\mathcal{X}$, which gives estimated parameters having all the same mean value $E[\hat{\theta}]$. Therefore all the gradients have the same variance, $E_X[g_\theta^2] = V_X$ and $E_Y[g_\theta^2] = V_Y$. Let m be the mean of g_θ and N_Y the Gaussian 0-mean variable random part of g_θ, corresponding to the posterior probability. Then we have :

$$E_Y[g_\theta^4] - E_Y[g_\theta^2]^2 = E_Y[(m+N_Y)^4] - E_Y[(m+N_Y)^2]^2 = m^4 + 6m^2 + 3V_Y^2 - (m^2 + V_Y)^2$$

We have also $E_X[g_\theta^4] - E_X[g_\theta^2]^2 = E_X[N_X^4] - E_X[N_X^2]^2 = 3V_X^2 - V_X^2$, and finally :

$$\beta_{ijij}^{rr} = 2[V_Y(V_Y + 2m) - V_X^2]$$

Non-diagonal terms $(\theta_1 \neq \theta_2)$

$$\beta_{ijkl}^{rs} = E_Y[g_{\theta_1}^2 g_{\theta_2}^2] - E_Y[g_{\theta_1}^2]E_Y[g_{\theta_2}^2] - E_X[g_{\theta_1}^2 g_{\theta_2}^2] + E_X[g_{\theta_1}^2]E_Y[g_{\theta_2}^2]$$

The posterior terms give, after simplification (using the Wick theorem which gives a simple decomposition of the 4th order moments in the case of Gaussian random variables) :

$$2E_Y[N_{Y1}N_{Y2}](E_Y[N_{Y1}N_{Y2}] + 2m^2) = 2C_{12}^Y(C_{12}^Y + 2m^2)$$

where C_{12}^Y is the posterior covariance of the gradients g_{θ_1} and g_{θ_2}, N_{Y1} and N_{Y2} are the random parts.

The prior terms are rewritten as $2E_X[N_{Y1}N_{Y2}]^2 = 2(C_{12}^X)^2$.

We obtain :

$$\beta_{ijkl}^{rs} = 2[C_{12}^Y(C_{12}^Y + 2m^2) - (C_{12}^X)^2]$$

which depends on the covariance of the GMRF pixel differences. This covariance is null for far away pixels, as there are only short distance interactions. We can consider the non-diagonal terms are negligible compared to the diagonal ones, as an approximation (this has been verified numerically).

The covariance of X^r (random part of X) can be easily calculated, as it is given by the covariance matrices Σ_X or Σ_Y (see previous appendix). In the Fourier basis, these matrices are diagonal and can be respectively written as $(b\,c)^{-1}$ and $(b\,c+w)^{-1} = W$. To obtain their expression in the image space, we recall that covariance matrices are block-circulant, and their respective generators are the inverse Fourier transforms of their diagonal expression. The covariances are also given by :

$$E_X[X^r_{ij}X^r_{kl}] = \frac{1}{2}(\Sigma_X)_{ij,kl} = \frac{1}{2}\mathcal{F}^{-1}\left[\frac{1}{b\,c}\right]_{i-k,j-l} \tag{B.3}$$

$$E_Y[X^r_{ij}X^r_{kl}] = \frac{1}{2}(\Sigma_Y)_{ij,kl} = \frac{1}{2}\mathcal{F}^{-1}\left[\frac{1}{b\,c+w}\right]_{i-k,j-l} = \frac{1}{2}\mathcal{F}^{-1}[W]_{i-k,j-l} \tag{B.4}$$

The covariance of the gradients g_{θ_1} and g_{θ_2} is calculated in the same manner, by modifying the diagonal terms to take into account the derivative operators w.r.t. lines or columns :

$$C^X_{12} = \frac{1}{2}\mathcal{F}^{-1}\left[\frac{\mathcal{F}[D_m]^*\mathcal{F}[D_n]}{\bar{b}\,c}\right]_{i-k,j-l}, \qquad C^Y_{12} = \frac{1}{2}\mathcal{F}^{-1}\left[\frac{\mathcal{F}[D_m]^*\mathcal{F}[D_n]}{\bar{b}\,c+w}\right]_{i-k,j-l}$$

We only consider the diagonal terms, which are all equal in the constant approximation case. Then the system is simply inverted and gives a solution which has the same expression as in the homogeneous case :

$$\text{var}(\hat{b}^{x,y}_{kl}) \simeq \frac{\sigma^2\|\bar{\alpha}\|^2}{\bar{\beta}^2} = \frac{1}{\sigma^2\bar{\beta}^2}\|E_Y[(D_{x,y}X)^2_{kl}HX^r]\|^2$$

where $\bar{\alpha}$ and $\bar{\beta}$ are respectively a vector and a constant (the elements of α are all equal to $\bar{\alpha}$ and the elements of β are all equal to $\bar{\beta}$).

Computation of $\left\|E_Y[(D_xX)^2_{kl}HX^r]\right\|^2$

Let us first compute the expectation $E_Y[(g^x)^2_{kl})HX^r]$.

The mean value of the gradients is denoted by m. We have :

$$g^x_{kl} = m + (X^r_{k+1,l} - X^r k, l)$$

We consider the expectation for pixel (p,q). Because of the zero-mean of the random parts and zero 3rd order moments, the remaining terms are :

$$E_Y\left[2m(X^r_{k+1,l} - X^r_{kl})(HX^r)_{pq}\right]$$

The convolution term can be expanded : $(HX^r)_{pq} = \sum_{ij} h_{p-i,q-j}X^r_{ij}$.

Using equation (B.4) we obtain :

$$E_Y[(g^x)^2_{kl})HX^r] = m\sum_{ij} h_{p-i,q-j}(\mathcal{F}^{-1}[W]_{i-k-1,j-l} - \mathcal{F}^{-1}[W]_{i-k,j-l})$$

We remark that $\sum_{ij} h_{p-i,q-j}\mathcal{F}^{-1}[W]_{i-k-\delta,j-l} = (h * \mathcal{F}^{-1}[W])_{p-\delta,q}$ where δ is equal to 1 or 0. Let us call Q the convolution of h and $\mathcal{F}^{-1}[W]$, then we have :

$$E_Y[\ldots] = m(Q_{p-1,q} - Q_{p,q}) = m\,(D^t_xQ)_{pq}$$

Finally the L_2 norm of the expectation vector is computed in the frequency space and is given by the following expression, simplified using symmetry properties :

$$\|E_Y[\ldots]\|^2 = \frac{m^2}{\sigma^2} \sum_{ij} \frac{|\mathcal{F}[h]_{ij} \mathcal{F}[D_x]^*_{ij}|^2}{(\bar{b}\, c_{ij} + w_{ij})^2} = m^2 \sum_{ij} \frac{c_{ij} w_{ij}}{(\bar{b}\, c_{ij} + w_{ij})^2}$$

The variance is then evaluated with $Y = Y^0$ and $b = E[\hat{b}]$ where $E[\hat{b}]$ is the mean value of the estimated parameter field. All the estimated parameters are equal, since we consider a particular case, for which the computation is possible. To simplify it is also possible to replace $E[\hat{b}]$ with $\hat{b}^0$, the results remaining practically the same.

Appendix C

Sampling from inhomogeneous Gaussian distributions

The Gibbs sampler [17] is used to explore both prior and posterior inhomogeneous Gaussian distributions. This sampler enables to construct iterately a chain of images $\{X^n\}$. After a sufficient number of iterations, the equilibrium distribution is reached.

One iteration of the Gibbs algorithm consists of :

- Choosing a pixel or site s (randomly or deterministically) ;
- Computing the local conditional probability of X_s knowing the neighbouring pixels V_s : $P(X_s \mid V_s)$;
- Sampling X_s from the law $P(X_s \mid V_s)$.

C.1 Prior IGMRF

C.1.1 Inhomogeneous Gibbs prior sampler

The prior law is given by :

$$P(X \mid \{b^x\}, \{b^y\}) \propto e^{-\sum_{ij}\left\{b^x_{ij}(D_xX)^2_{ij}+b^y_{ij}(D_yX)^2_{ij}\right\}}$$

Therefore, the local conditional probability of X_{ij} is Gaussian :

$$P(X_{ij} \mid X_{kl\neq ij}) \propto e^{-b^x_{ij}(X_{i+1,j}-X_{ij})^2-b^y_{ij}(X_{i,j+1}-X_{ij})^2-b^x_{i-1,j}(X_{i-1,j}-X_{ij})^2-b^y_{i,j-1}(X_{i,j-1}-X_{ij})^2}$$

ALGORITHM C.1.1 (PRIOR IGMRF GIBBS)

- *Initialization :* X^0 *=constant ;*
- *First sample the odd pixels (*$i+j$ *odd), then the even pixels.*
  For each pixel, the update consists of taking :

$$X^{n+1}_{ij} \sim \frac{b^x_{ij}X^n_{i+1,j}+b^y_{ij}X^n_{i,j+1}+b^x_{i-1,j}X^n_{i-1,j}+b^y_{i,j-1}X^n_{i,j-1}}{b^x_{ij}+b^y_{ij}+b^x_{i-1,j}+b^y_{i,j-1}} + N\left(0\,,\,\frac{1}{2(b^x_{ij}+b^y_{ij}+b^x_{i-1,j}+b^y_{i,j-1})}\right)$$

To estimate the parameters b, we need to compute the variances :

$$(V^X)^x_{ij} = E_X[(X_{i+1,j} - X_{ij})^2] \quad \text{and} \quad (V^X)^y_{ij} = E_X[(X_{i,j+1} - X_{ij})^2]$$

The variances are estimated by averaging the squared pixel differences over a large set of Gibbs samples. About 100 burn-in iterations are necessary to ensure reaching the prior distribution.

C.1.2 A deterministic approximation of prior IGMRF local variance

The variances V^X and V^Y corresponding to synthetic b^x and b^y fields has been represented on figure C.1. For a given pixel difference, only the corresponding b value seems to have an effect on the variance. Therefore we choose the following approximation, which consists of taking for each gradient $(D_{x,y}X)_{ij}$ the variance of the gradients of an homogeneous model, with $b = b^{x,y}_{ij}$:

$$(V^X)^{x,y}_{ij} \simeq \frac{1}{4b^{x,y}_{ij}} \tag{C.1}$$

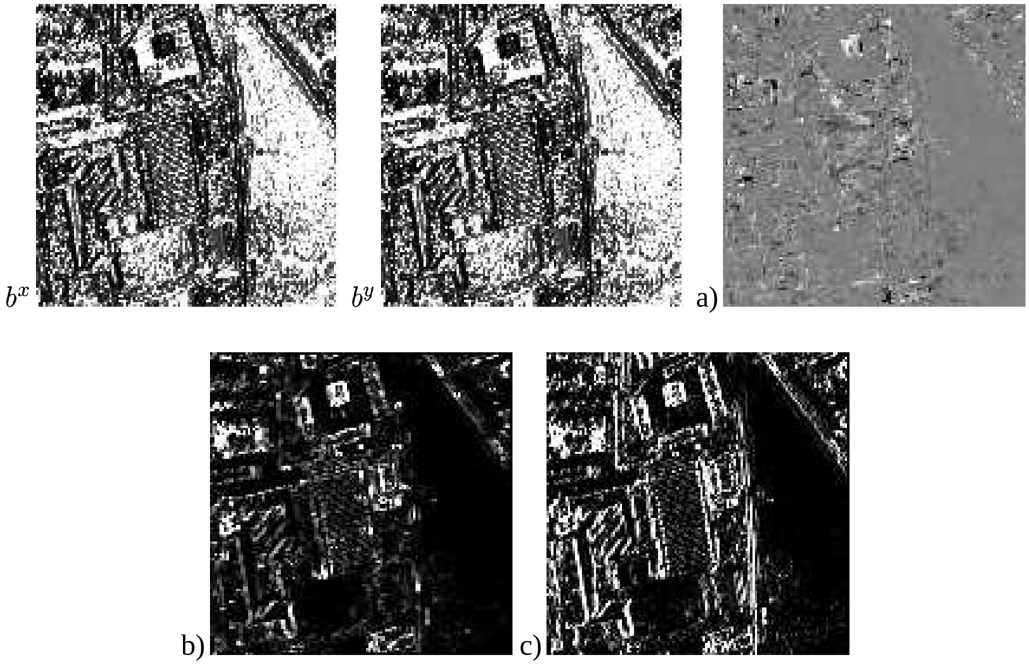


Figure C.1: *a) prior sample (contrast amplified by a factor 3), b) estimated variance $(V^X)^x$, c) approximate variance using (C.1)*

C.2 Posterior IGMRF

C.2.1 Inhomogeneous Gibbs posterior sampler

The posterior law is given by :

$$P(Y \mid X, \{b^x\}, \{b^y\}) \propto e^{-||Y-HX||^2/2\sigma^2 - \sum_{ij}\left\{b^x_{ij}(D_xX)^2_{ij}+b^y_{ij}(D_yX)^2_{ij}\right\}}$$

Therefore, the local conditional probability of X_{ij} is Gaussian :

$$\begin{aligned} P(X_{ij} \mid Y, X_{kl\neq ij}) \propto\ & e^{-X^t(H^tH)X/2\sigma^2+2(H^tY)_{ij}X_{ij}/2\sigma^2} \\ & \times e^{-b^x_{ij}(X_{i+1,j}-X_{ij})^2-b^y_{ij}(X_{i,j+1}-X_{ij})^2-b^x_{i-1,j}(X_{i-1,j}-X_{ij})^2-b^y_{i,j-1}(X_{i,j-1}-X_{ij})^2} \end{aligned}$$

The term $X^t(H^tH)X$ contains both terms in X^2_{ij} and X_{ij}. The quadratic term of $X^t(H^tH)X/2\sigma^2$ is $s = (H^tH)_{jj}/2\sigma^2 = \sum_{ij} h^2_{ij}/2\sigma^2$. Then we have :

ALGORITHM C.2.1 (POSTERIOR IGMRF GIBBS)

- *Initialization :* $X^0 = \hat{X}$ *(restored image with parameters* $\{b^x\}, \{b^y\}$*) ;*
- *Compute* $A^n = s\,X^n + (H^tY - H^tHX^n)/2\sigma^2$ *;*
- *Update simultaneously independent pixels on a grid G, by using a coding scheme like in Fig. C.2 :*

$$\begin{aligned} X^{n+1}_{ij} \sim\ & \frac{A^n_{ij}+b^x_{ij}X^n_{i+1,j}+b^y_{ij}X^n_{i,j+1}+b^x_{i-1,j}X^n_{i-1,j}+b^y_{i,j-1}X^n_{i,j-1}}{s+b^x_{ij}+b^y_{ij}+b^x_{i-1,j}+b^y_{i,j-1}} \\ & +N\left(0\,,\ \frac{1}{2(s+b^x_{ij}+b^y_{ij}+b^x_{i-1,j}+b^y_{i,j-1})}\right) \end{aligned}$$

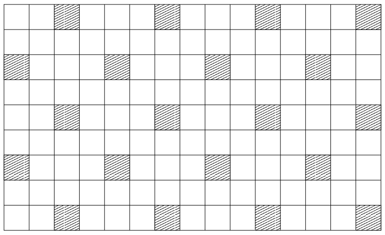


Figure C.2: *Grid of quasi-independent pixels of the posterior IGMRF corresponding to the kernel h defined in figure 1.1*

To optimize this algorithm, H^tHX^n is computed once and used to update N_xN_y/K pixels. For the convolution kernel of fig. 1.1, we have $K = 8$, and we use the coding scheme of fig. C.2 to update only independent pixels at once. This avoids recomputing H^tHX^n for each pixel, which is time consuming.

To estimate the parameters b, we need to compute the variances :

$$(V^Y)^x_{ij} + (m^2)^x_{ij} = E_Y[(X_{i+1,j} - X_{ij})^2] \quad \text{and} \quad (V^Y)^y_{ij} + (m^2)^y_{ij} = E_Y[(X_{i,j+1} - X_{ij})^2]$$

where V^Y is the variance corresponding to the fluctuations of the gradients of X and m^2 the mean of the pixel differences, i.e. the squared gradients of $\hat{X}$.

C.2.2 A deterministic approximation of posterior IGMRF local variance

Figure C.3 shows the variance with the same values of the b fields as in figure C.1. The deterministic part m^2 is usually much higher than V^Y. Even if V^Y, for a given gradient, is not only a function of the corresponding b, it can be approximated by the corresponding homogeneous variance as for the prior law :

$$(V^Y)^{x,y}_{ij} \simeq \frac{1}{4} \sum_{k,l} \frac{c_{kl}}{b^{x,y}_{ij}\, c_{kl} + w_{kl}} \tag{C.2}$$

which can be tabulated since w is constant for a given deconvolution problem.

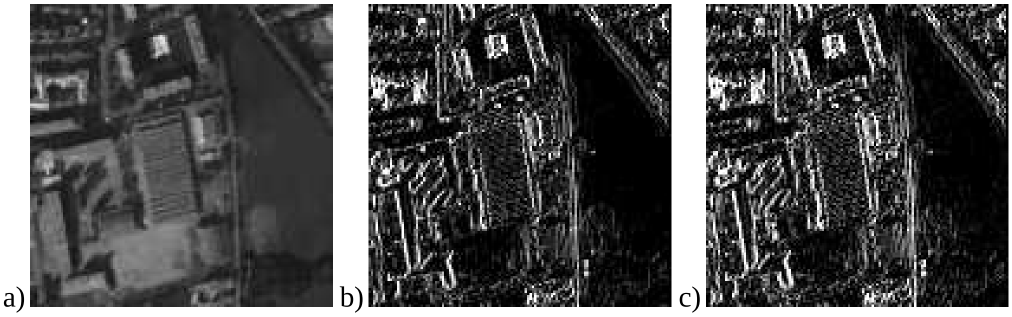


Figure C.3: *a) posterior sample computed with the paramaters of fig. C.1 and Y given in fig. 1.1, b) estimated variance $(V^Y)^x$, c) approximate variance using (C.2)*

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399