HAL
open science

Estimation d'hyperparamètres pour la restauration d'images satellitaires par une méthode MCMCML

André Jalobeanu, Laure Blanc-Féraud, Josiane Zerubia

▶ **To cite this version:**

André Jalobeanu, Laure Blanc-Féraud, Josiane Zerubia. Estimation d'hyperparamètres pour la restauration d'images satellitaires par une méthode MCMCML. RR-3469, INRIA. 1998. inria-00073221

HAL Id: inria-00073221

https://inria.hal.science/inria-00073221v1

Submitted on 24 May 2006

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

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

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Estimation d'hyperparamètres pour la restauration d'images satellitaires par une méthode « MCMCML »

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

N° 3469

Août 1998

THÈME 3

Rapport de recherche

ISSN 0249-6399

Estimation d'hyperparamètres pour la restauration d'images satellitaires par une méthode « MCMCML »

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

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

Rapport de recherche n° 3469 —Août 1998 —72 pages

Résumé : Le problème que nous abordons ici est la déconvolution d'images satellitaires, qui sont dégradées par l'optique et l'électronique utilisées pour leur acquisition. Les dégradations sont connues : les images sont convoluées par un opérateur H, et la variance du bruit N additif, blanc et gaussien, est connue.

Nous utilisons un modèle de régularisation introduisant une fonction de potentiel φ, qui interdit l'amplification du bruit lors de la restauration tout en préservant les discontinuités. Ce modèle admet deux hyperparamètres λ et δ. Nous nous intéressons ici à l'estimation des hyperparamètres optimaux afin d'effectuer la déconvolution de manière automatique.

Nous proposons pour cela d'utiliser l'estimateur du maximum de vraisemblance appliqué à l'image observée. Cet estimateur constitue le critère que nous allons optimiser. Pour évaluer ses dérivées, nous devons estimer des espérances calculées sur des échantillons, tenant compte des données observées et de l'a priori imposé. Cette probabilité faisant intervenir l'opérateur de convolution, il est très difficile d'utiliser un échantillonneur classique. Nous avons développé un algorithme de type Geman-Yang modifié, utilisant une variable auxiliaire, ainsi qu'une transformée en cosinus. Nous présentons à cette occasion un nouvel algorithme de déconvolution, rapide, qui est dérivé de cette méthode d'échantillonnage.

Nous proposons un algorithme « MCMCML » permettant d'effectuer simultanément l'estimation des hyperparamètres λ et δ et la restauration de l'image dégradée.

Une étude des échantillonneurs (y compris ceux de Gibbs et Metropolis), portant sur la vitesse de convergence et les difficultés de calcul liées à l'attache aux données, a également été réalisée.

Mots-clés : Régularisation, Phi-fonction, Hyperparamètres, Modèles variationnels, Champs de Markov, Estimation, Maximum de vraisemblance, Images satellitaires

Remerciements : Les auteurs souhaitent remercier le CNES pour l'image de Nîmes (SPOT 5), ainsi que Marc Sigelle et Xavier Descombes pour des discussions fructueuses.

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

Hyperparameter estimation for satellite image restoration by a MCMCML method

Abstract: This report deals with satellite image restoration. These images are corrupted by an optical blur and electronic noise, due to the physics of the sensors. The degradation model is known : blurring is modeled by convolution, with a linear operator H, and the noise is supposed to be additive, white and Gaussian, with a known variance.

The recovery problem is ill-posed and therefore must be regularized. We use a regularization model which introduces a φ function, which avoids noise amplification while preserving image discontinuities (ie. edges) of the restored image. This model exhibits two hyperparameters (λ and δ). Our goal is to estimate the optimal parameters in order to reconstruct images automatically.

Herein, we propose to use the Maximum Likelihood estimator, applied to the observed image. To optimize this criterion, we must estimate expectations by sampling (samples are extracted from a Markov chain) to evaluate its derivatives. These samples are images whose probability takes into account the convolution operator. Thus, it is very difficult to obtain them directly by using a standard sampler. We have developped a modified Geman-Yang algorithm, using an auxiliary variable and a cosine transform. We also present a new reconstruction method based on this sampling algorithm.

We detail the MCMCML algorithm which ables to simultaneously estimate λ and δ parameters, and to reconstruct the corrupted image.

An experimental study of samplers (including Gibbs and Metropolis algorithms), with respect to the rate of convergence and the difficulties of dependent data sampling, is also presented in this report.

Key-words: Regularization, Phi-function, Hyperparameters, Variational models, Markov random fields, 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), and Marc Sigelle and Xavier Descombes for interesting discussions.

Table des matières

1 Introduction : position du problème **7**
1.1 Régularisation . . . 7
1.1.1 Principe . . . 7
1.1.2 Préservation des discontinuités . . . 8
1.1.3 Différents modèles de régularisation . . . 8
1.2 Importance du choix des hyperparamètres . . . 9
1.3 Aspect probabiliste . . . 10
1.4 Théorème semi-quadratique . . . 12
1.5 Algorithmes de restauration . . . 12
1.5.1 L'algorithme Legend (déterministe) . . . 12
1.5.2 Autres techniques de restauration . . . 13

2 Estimation des hyperparamètres **15**
2.1 Choix de l'estimateur . . . 15
2.1.1 Introduction . . . 15
2.1.2 Maximum de vraisemblance sur Y (image observée) . . . 17
2.2 Calcul du gradient et du hessien . . . 18

3 Champs de Markov et échantillonnage **19**
3.1 Champs de Markov . . . 19
3.2 Echantillonnage du modèle a posteriori . . . 20
3.2.1 Echantillonneurs de Gibbs et Metropolis . . . 20
3.2.2 Echantillonneur de Geman-Yang modifié . . . 20
3.2.3 Résultats . . . 22
3.3 Echantillonnage du modèle a priori . . . 22
3.3.1 Dynamique de Metropolis . . . 23
3.3.2 Echantillonneur de Gibbs . . . 24
3.3.3 Echantillonneur de Geman-Yang modifié . . . 25
3.3.4 Résultats . . . 25
3.4 Application des échantillonneurs à la restauration . . . 26
3.4.1 Techniques de relaxation stochastique . . . 26
3.4.2 Un nouvel algorithme déterministe . . . 27

4 Difficultés de l'échantillonnage **31**
4.1 Convergence : études théoriques 31
4.2 Convergence : étude expérimentale 32
4.2.1 Comparaison des différents algorithmes 32
4.2.2 Influence des hyperparamètres et transition de phase 33
4.2.3 Sensibilité aux conditions initiales 34
4.2.4 Critère d'arrêt 36
4.3 Difficultés liées à la convolution 38
4.4 Conclusion 38

5 L'algorithme d'estimation et restauration simultanées **41**
5.1 « Markov Chain Monte Carlo Maximum Likelihood » 41
5.1.1 Description 41
5.1.2 Convergence 42
5.1.3 Précision de l'estimation 43
5.2 Résultats 44
5.2.1 Multiplicité des solutions 44
5.2.2 Critère localement quadratique 44
5.2.3 Influence de la précision de l'estimation 45
5.2.4 Influence de la zone d'extraction des imagettes 46
5.3 Comparaison avec le gradient stochastique 48

6 Conclusion et perspectives **49**
6.1 A priori sur les hyperparamètres 49
6.2 Une meilleure qualité de la restauration 49
6.3 Vers un modèle plus local 50

A Résultats **55**

B Comportement des fonctions φ **61**

C Théorème semi-quadratique **63**

D Diagonalisation de H et DCT **65**

E Calcul du gradient et du hessien **67**

F Espérances précalculées (modèle a priori) **69**

G Coût des algorithmes **71**

Notations

La dimension des images est N_x colonnes par N_y lignes.

X désigne un vecteur, formé par les pixels d'une image dans l'ordre lexicographique.
X_s est la valeur du site (ou pixel) s.
$X_{i,j}$ est la valeur du pixel situé à la colonne i et à la ligne j.
X^n est le n-ième vecteur de la suite (X^n).
Par abus de notation on écrit $P(X)$ au lieu de $P(x = X)$.

$\mathcal{X}$ désigne l'image non dégradée.
Y est l'observation (image dégradée).
H est l'opérateur de convolution (matrice de dimension $N_x^2 N_y^2$).

B^x, B^y et W sont des vecteurs de même dimension que X.

h est le générateur de H, c'est la réponse impulsionnelle de la fonction de flou.

D_x et D_y sont les opérateurs de dérivation sur les images, par rapport aux colonnes et aux lignes : $(D_x X)_{i,j} = X_{i+1,j} - X_{i,j}$ et $(D_y X)_{i,j} = X_{i,j+1} - X_{i,j}$.

d_x et d_y sont les générateurs de D_x et D_y.

h^4 est le générateur de H^4, H^4 étant une matrice de dimension $(2N_x)^2(2N_y)^2$ obtenue par extension de H (ajout de zéros).

d_x^4 et d_y^4 sont les générateurs de D_x^4 et D_y^4, opérateurs de dérivation sur les images symétrisées de dimension $2N_x \times 2N_y$.

$\mathcal{F}$ désigne la transformée de Fourier.

Ω est l'espace de configuration (ensemble des images de dimension $N_x \times N_y$ à pixels réels, soit $\mathbb{R}^{N_x N_y}$).

Chapitre 1

Introduction : position du problème

Si on appelle $\mathcal{X}$ l'image non dégradée, H l'opérateur de convolution et N le bruit additif, on observe l'image Y telle que $Y = H\,\mathcal{X} + N$. C'est l'image que nous devons restaurer.

- Le bruit est considéré comme blanc, gaussien, et stationnaire.
- la réponse impulsionnelle est symétrique en x et y, positive, conserve les valeurs moyennes et respecte la propriété de Shannon.

Les images que nous devons traiter dans un premier temps sont fournies par le CNES ; elles simulent des images satellitaires à 5 m de résolution (futur satellite SPOT 5). La variance du bruit, ainsi que l'opérateur H, sont fournis. H est, dans le cas du CNES, séparable et symétrique [1] (voir annexe A, fig. A.1).

Nous pouvons également appliquer les algorithmes de restauration à n'importe quel type d'images dont H et N respectent les propriétés ci-dessus.

1.1 Régularisation

1.1.1 Principe

Le problème de la restauration des images satellitaires est mal posé au sens de Hadamard (voir [HAD23]).

La régularisation [DEM89, CHL88] consiste à introduire une contrainte de régularité sur l'image que l'on obtient par déconvolution (plus ou moins lisse par exemple).

Le critère des moindres carrés consiste à chercher l'image X qui minimise la fonctionnelle $U_0(X) = ||Y - HX||^2$. La régularisation correspond à l'introduction d'un terme supplémentaire dans l'expression de U_0, qui augmente si X s'éloigne d'une image ayant

1. Lorsque l'on utilise un opérateur de convolution H ou de dérivation, on doit préciser les conditions aux bords de l'image. Elles sont *symétriques* dans notre cas : cela signifie que les gradients sont nuls sur les bords, et l'on extrapole l'image en effectuant une symétrie axiale par rapport aux bords.

les bonnes propriétés de régularité [CHA94, CHA97]. On cherche alors :

$$\hat{X} = \arg \inf_{X \in \Omega} U(X) \tag{1.1}$$

$$U(X) = \frac{1}{2\sigma^2}||Y - HX||^2 + \Phi(X) \tag{1.2}$$

$$\text{où} \quad \Phi(X) = \lambda^2 \sum_{ij} \left\{ \varphi\left(\frac{X_{i+1,j} - X_{i,j}}{\delta}\right) + \varphi\left(\frac{X_{i,j+1} - X_{i,j}}{\delta}\right) \right\}$$

λ et δ désignent ici les **hyperparamètres**, la qualité de l'image restaurée dépend sensiblement de leur choix (voir paragraphe 1.2). La fonction φ est symétrique, positive et croissante ; elle est appliquée aux gradients de l'image (différences entre pixels voisins) de façon à pénaliser les images très bruitées.

En prenant par exemple $\varphi(u) = u^2$, on effectue une régularisation au sens de Tikhonov [TIK63]. Le terme que l'on ajoute ainsi pénalise les forts gradients, aussi bien le bruit que les contours, en lissant dans toutes les directions de manière identique.

1.1.2 Préservation des discontinuités

Pour éviter un lissage isotrope, on choisit une fonction φ non quadratique, qui pénalise les faibles gradients (dus au bruit) tout en préservant les plus forts (ie. les contours). Si on se place sur un contour Γ (ie. ligne de niveau) de l'image, on voudrait effectuer alors un lissage selon le contour, en évitant de lisser dans la direction du gradient pour ne pas dégrader ce contour.

Le choix de la fonction de potentiel φ est déterminant, c'est elle qui fixe le « coût » de chaque valeur du gradient.

PROPRIÉTÉ 1.1.1 (FONCTION DE POTENTIEL φ [CHA94, CHA97])
Pour préserver les détails tout en lissant les zones homogènes, φ doit vérifi er les propriétés suivantes :
i) $\lim_{u \to 0^+} \frac{\varphi'(u)}{2u} = 1$
ii) $\lim_{u \to \infty} \frac{\varphi'(u)}{2u} = 0$
iii) $\frac{\varphi'(u)}{2u}$ *est continue et strictement décroissante sur* $[0, +\infty[$.

Cela signifie grossièrement que φ doit croître « moins vite » que u^2, pour permettre la préservation des forts gradients, tout en restant équivalente à u^2 en 0 (faibles gradients), effectuant une régularisation de Tikhonov pour les zones homogènes.

1.1.3 Différents modèles de régularisation

Plusieurs fonctions vérifient les conditions ci-dessus. Leur comportement est illustré dans l'annexe B.

Le comportement de φ est quadratique en-dessous d'un certain seuil. Ce comportement est différent selon la fonction (le passage du lissage à la préservation est brutal pour φ_{QT}, beaucoup moins pour φ_{HS}).

Lors de la restauration, on obtient des contours plus nets avec une fonction qui fait clairement la différence entre faibles et forts gradients (asymptote finie en l'infini), qu'avec des fonctions comme φ_{GR} ou φ_{HS} (asymptote linéaire en l'infini). Mais plus le comportement est différent, plus une faible variation des données peut avoir de fortes conséquences sur le résultat, ce qui n'est pas le but recherché lors de la régularisation. En outre, l'emploi de φ_{GR} et φ_{HS} entraîne l'unicité du minimum $U(X)$, car ces fonctions φ sont *strictement convexes* et H est de rang plein : cela permet d'utiliser des méthodes déterministes (cf. algorithme 3.4.2) pour la recherche de ce minimum.

Si les fonctions φ sont harmonisées (on effectue un changement de variable permettant d'avoir un même seuil quelle que soit φ) alors le résultat de la déconvolution, pour des hyperparamètres (λ, δ) fixés, ne doit pas trop dépendre du choix de φ.

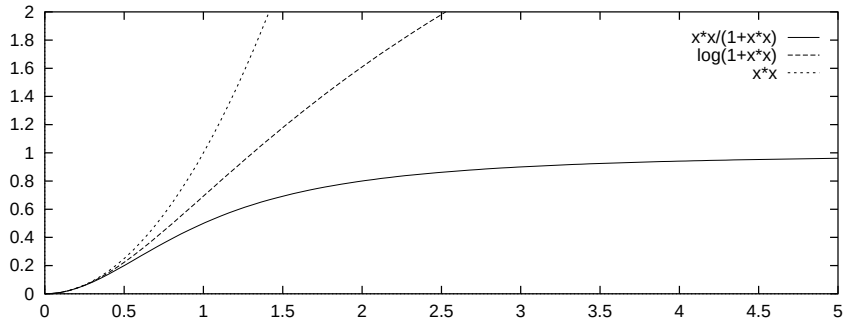
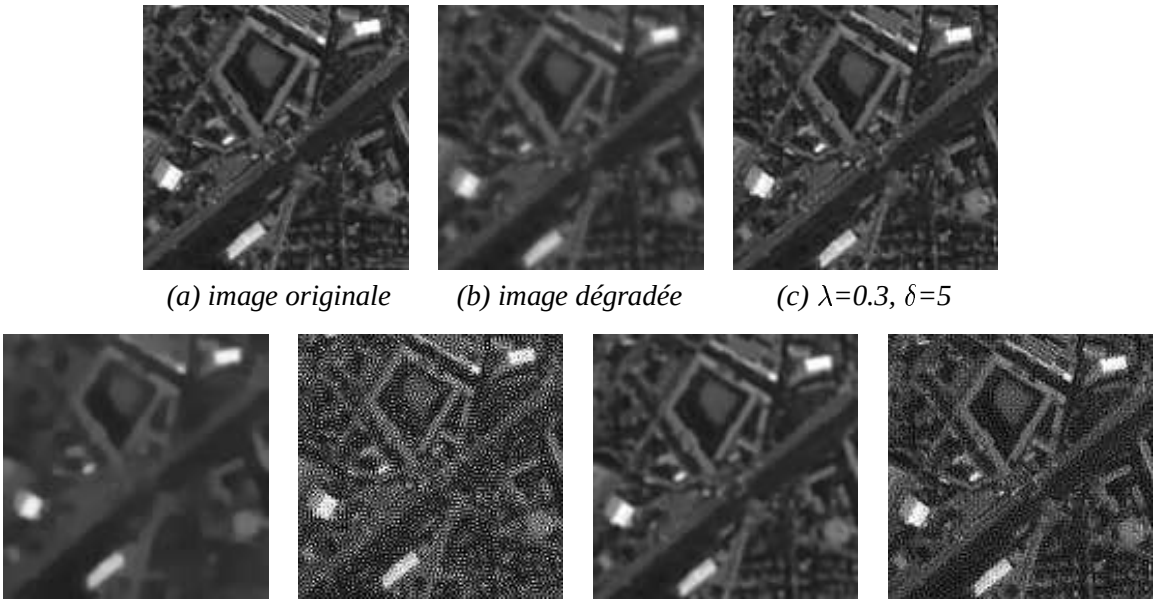


FIG. 1.1: *Quelques fonctions $\varphi(u)$ en fonction de u : φ_{GM}, φ_{HL}, quadratique.*

Fonction de potentiel	**Expression de $\varphi(\mathbf{u})$**	**Expression de $\varphi'(\mathbf{u})/\mathbf{2u}$**
φ_Q Quadratique (*convexe*)	u^2	1
φ_{QT} Quadratique tronquée (*non convexe*)	$\min(u^2, 1)$	1 si $u < 1$, 0 si $u \geq 1$
φ_{GM} Geman & McClure (*non convexe*)	$u^2/1 + u^2$	$1/(1+u^2)^2$
φ_{HL} Hebert & Leahy (*non convexe*)	$\log(1+u^2)$	$1/(1+u^2)$
φ_{GR} Green (*convexe*)	$2\ \log(\cosh(u))$	$tanh(u)/u$
φ_{HS} Hyper Surfaces (*convexe*)	$2\sqrt{1+u^2} - 2$	$1/\sqrt{1+u^2}$

1.2 Importance du choix des hyperparamètres

Comme on peut le constater sur la figure 1.2, la qualité de l'image restaurée dépend sensiblement du choix des hyperparamètres. Un λ trop grand entraîne une sur-régularisation :



(a) image originale *(b) image dégradée* *(c) λ=0.3, δ=5*

(d) λ=2, δ=1 *(e) λ=0.02, δ=5* *(f) λ=0.3, δ=0.5* *(g) λ=0.3, δ=50*

FIG. 1.2: *Résultat de la restauration en fonction des hyperparamètres avec φ_{GM}*

on obtient une image avec un très bon rapport signal-bruit, mais trop lisse (d), éventuellement encore plus floue que l'image que l'on cherche à corriger! Par contre, si λ est trop faible, l'image est nette mais noyée dans le bruit (e).

La signification de δ est différente : il est inversement proportionnel à un seuil en-dessous duquel on lisse les gradients, et un seuil trop grand entraîne la disparition des détails les plus fins(f). Un seuil trop petit entraîne, par contre, une diminution insuffisante du bruit (g).

L'image (c) semble plus correcte, mais ce n'est certainement pas la meilleure, les hyperparamètres ont été choisis manuellement. Cette manière de choisir λ et δ, peu rapide même sur de petites images, est incompatible avec le traitement de grandes images satellitaires, et nécessite une interaction avec l'utilisateur. Nous proposons d'effectuer une estimation *automatique* d'hyperparamètres, autrement dit, de développer un algorithme à la fois capable de minimiser U et de choisir le meilleur couple (λ, δ) correspondant au problème.

1.3 Aspect probabiliste

Si on se place dans un cadre bayésien, on peut s'intéresser à la probabilité d'avoir une image X à l'origine de l'observation Y, les hyperparamètres λ et δ étant fixés. On s'inté-

resse à :

$P(X \mid Y) = P(Y \mid X)\, P(X) \,/\, P(Y)$, avec $P(Y) =$ constante car Y est l'observation.

Maximum de vraisemblance

$P(Y \mid X)$ désigne ici la probabilité d'obtenir une image Y par dégradation de X. Comme on a $Y = HX + N$, X étant fixé, Y suit la distribution du bruit $N = Y - HX$, qui est indépendant de X. On a donc :

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

C'est la vraisemblance de X. K_σ est une constante de normalisation qui ne dépend pas de X. On remarque alors que la minimisation de $||Y - HX||^2$ équivaut à la maximisation de $P(Y \mid X)$: le critère des *moindres carrés* est équivalent au critère du *maximum de vraisemblance*, dans le cas d'un bruit additif gaussien.

Maximum a posteriori

$P(X)$ correspond à la connaissance a priori de l'image X que l'on veut obtenir. C'est ici que l'on introduit les contraintes sur les contours et les zones homogènes de l'image, autrement dit, la *régularisation*. Il faut se référer au chapitre sur les champs de Markov pour comprendre la forme gibbsienne de cette probabilité :

$$P(X) = \frac{1}{Z} e^{-\Phi(X,\lambda,\delta)} \tag{1.3}$$

qui fait intervenir Φ définie en (1.2), et qui signifie simplement qu'une image n'ayant pas les propriétés de régularité désirées est peu probable.

Finalement, on obtient l'expression suivante pour la probabilité a posteriori :

$$P(X \mid Y) = \frac{1}{K_\sigma} e^{-||Y-HX||^2/2\sigma^2} \frac{1}{Z} e^{-\Phi(X,\lambda,\delta)} = \frac{1}{Z_Y} e^{-||Y-HX||^2/2\sigma^2 - \Phi(X,\lambda,\delta)} \tag{1.4}$$

Minimiser l'énergie $U(X, \lambda, \delta)$ définie en (1.2), est équivalent à maximiser $P(X \mid Y) \propto e^{-U(X,\lambda,\delta)}$: la *déconvolution avec régularisation* équivaut à maximiser le critère du *maximum a posteriori*.

Les deux formulations déterministe et probabiliste sont équivalentes, et l'on peut traiter le problème des deux façons. Si $U(X)$ a un minimum unique (φ convexe), il vaut mieux faire appel à un algorithme déterministe (plus rapide, voir algorithme 3.4.2), alors que dans le cas contraire, seul un algorithme stochastique (tel le recuit simulé [LAA87], cf. paragraphe 3.4.1) parviendra à sortir des minima locaux.

1.4 Théorème semi-quadratique

La puissance de la régularisation par φ-modèle réside dans sa non-linéarité. Mais c'est aussi de là que proviennent les difficultés du calcul d'optimisation. Si φ est quadratique, la fonctionnelle à minimiser est quadratique, donc le minimum est unique et facile à calculer en une seule étape.

On va donc chercher à se ramener à un modèle quadratique, en utilisant une technique de variable auxiliaire. Cela consiste à remplacer $\varphi(u)$ par une nouvelle fonction de u et une variable auxiliaire b, en faisant appel au théorème semi-quadratique (C.0.1) (annexe C) [CHA94, GEM95] :

$$\varphi(u) \mapsto (u-b)^2 + \psi(b) \quad \text{avec } \varphi(u) = \inf_{b\in\mathbb{R}} \left[(u-b)^2 + \psi(b)\right]$$

On ne va plus minimiser $U(X)$ (ou maximiser $P(Y \mid X)$) par rapport à X (donc u, qui traduit les gradients), mais de manière *alternée*, par rapport à X (à b fixé) et à B_x et B_y (à X fixé). Les images B_x et B_y regroupent toutes les variables auxiliaires b, associées aux gradients de X selon les colonnes et les lignes.

On peut écrire pour chaque pixel et pour les gradients en x :

$$\varphi\left(\frac{g^x_{ij}}{\delta}\right) = \inf_{B^x_{ij}} \left[\left(\frac{g^x_{ij}}{\delta} - B^x_{ij}\right)^2 + \psi(B^x_{ij})\right], \quad \text{où } g^x_{ij} = X_{i+1,j} - X_{i,j}$$

$$\Phi(X) = \inf_{B^x,B^y} \Phi^*(X,B^x,B^y) = \inf_{B^x,B^y} \left[\lambda^2 \sum_{ij} \left\{\left(\frac{g^x_{ij}}{\delta} - B^x_{ij}\right)^2 + \psi(B^x_{ij}) + \left(\frac{g^y_{ij}}{\delta} - B^y_{ij}\right)^2 + \psi(B^y_{ij})\right\}\right]$$

Sachant que $\hat{X}$ (1.1) est l'image qui minimise $U(X)$ définie en (1.2) on cherche

$$\hat{X} = \arg\inf_X \left[\inf_{B^x,B^y} \left[\frac{1}{2\sigma^2}||Y - HX|| + \lambda^2 \sum_{ij} \left\{\left(\frac{g^x_{ij}}{\delta} - B^x_{ij}\right)^2 + \psi(B^x_{ij}) + \left(\frac{g^y_{ij}}{\delta} - B^y_{ij}\right)^2 + \psi(B^y_{ij})\right\}\right]\right]$$

ce qui conduit aux algorithmes que nous allons présenter dans la suite, qui fonctionnent tous suivant un schéma de minimisation alternée sur X et B^x, B^y.

1.5 Algorithmes de restauration

1.5.1 L'algorithme Legend (déterministe)

L'algorithme de restauration Legend, décrit dans [CHA94, CHA97], est déterministe : son rôle est de minimiser la *fonctionnelle* $U(X)$ de manière alternée sur X et B^x B^y.

ALGORITHME 1.5.1 (LEGEND [CHA97])
$X^0 = Y$ *; répéter (jusqu'à la convergence) :*

- $(\hat{B^x}, \hat{B^y}) = \arg\inf_{B^x, B^y} \Phi^*(X, B^x, B^y)$
  Ceci se fait en une seule étape, suivant le théorème semi-quadratique :
  $\Rightarrow \quad b_{ij}^{xy} = g_{ij}^{xy}/\delta - \frac{1}{2}\varphi'(g_{ij}^{xy}/\delta).$
- $\hat{X} = \arg\inf_X \left[||Y - HX||^2/2\sigma^2 + \Phi^*(X, B^x, B^y)\right]$
  Il s'agit de résoudre un système linéaire, on utilise pour cela une méthode de gradient conjugué, itérative, avec les équations normales

$$\left(\frac{1}{2\sigma^2} H^t\, H - \frac{\lambda^2}{\delta^2}\Delta\right) X = \frac{1}{2\sigma^2} H^t\, Y + \frac{\lambda^2}{\delta^2}\left(D_x^t B^x + D_y^t B^y\right)$$

  où Δ est le laplacien et $D_x\, D_y$ les opérateurs de dérivation sur les lignes et sur les colonnes.

1.5.2 Autres techniques de restauration

D'autres méthodes de restauration sont utilisables, notamment les méthodes *stochastiques* telles que le recuit simulé (cf. paragraphe 3.4.1), qui permettent d'échapper aux minima locaux. Moyennant quelques modifications, on peut en tirer des algorithmes déterministes (ICM par exemple), notamment celui que nous utilisons pour déconvoluer les images satellitaires (cf. algorithme 3.4.1), qui est équivalent à Legend dans le cas des fonctions φ étudiées.

Ces techniques sont présentées à la fin du chapitre sur les champs de Markov, car elles font appel à des échantillonneurs.

Chapitre 2

Estimation des hyperparamètres

Il s'agit ici de choisir le couple d'hyperparamètres (λ, δ) qui fournit la meilleure image X sachant que l'on observe Y, lorsque l'on déconvolue en minimisant $U(X, \lambda, \delta)$ (1.2), ou en maximisant $P(X, \lambda, \delta)$ par rapport à X.

Le problème qui se pose est le choix de l'estimateur qui va nous servir à sélectionner les meilleurs hyperparamètres. Il faut trouver un critère à optimiser, qui reflète le plus fidèlement possible le critère visuel utilisé par le photo-interprète lorsqu'il juge la qualité des images déconvoluées. Celles-ci doivent montrer un maximum de détails, avec le moins de bruit possible.

2.1 Choix de l'estimateur

2.1.1 Introduction

Nous allons présenter trois estimateurs, et expliquer pourquoi nous n'avons retenu que le troisième, bien qu'il soit plus difficile à implémenter.

Maximum de vraisemblance sur X (image restaurée)

Si l'on considère une réalisation X dans l'espace Ω, $P(X \mid \lambda, \delta)$ est la probabilité d'avoir cette image a priori (quelle que soit l'observation qu'on en fait), les hyperparamètres du modèle étant donnés.

Si X est par exemple une image satellitaire, et si on suppose que le modèle est bien adapté à ce type d'image, on peut chercher les meilleurs hyperparamètres a priori qui lui correspondent. L'estimateur du *maximum de vraisemblance* des hyperparamètres (λ, δ) est donc

$$(\hat{\lambda}, \hat{\delta}) = \arg\max_{\lambda,\delta} P(X \mid \lambda, \delta) = \arg\max_{\lambda,\delta} \frac{1}{Z} e^{-\Phi(X,\lambda,\delta)} \tag{2.1}$$

où Φ est l'énergie associée à X et au φ-modèle utilisé.

Les hyperparamètres estimés de cette façon ont un sens par rapport aux *données complètes*, c'est-à-dire non dégradées par H et par le bruit. Cela permet par exemple d'effectuer une *segmentation* de l'image dans de bonnes conditions, comme dans [DES96] ou [MOR96b]. Pour la segmentation on a en effet besoin de calculer l'énergie Φ, afin de décider si un pixel appartient ou non à une certaine classe ; le meilleur choix de cette fonction d'énergie correspond au meilleur $(\hat{\lambda}, \hat{\delta})$ avec l'estimateur (2.1).

Par contre, pour la restauration, le problème qui se pose est de filtrer le bruit en préservant les détails ; il paraît alors naturel que les hyperparamètres optimaux dépendent du type de dégradation (σ du bruit et taille de H) subie par l'image. Lorsque l'image n'est pas dégradée on a $\lambda = 0$ (régularisation inutile), alors que si l'image est très bruitée, on aura tendance à beaucoup régulariser, et à choisir un seuil des discontinuités assez grand (δ faible).

Sur quelle image X doit-on alors estimer λ et δ? Si on les estime sur l'image non dégradée (en supposant que l'on puisse la connaître), comme on le fait pour la segmentation, ils ne seront pas forcément adaptés au problème puisqu'ils ne dépendront pas de H et du bruit. Si on les estime sur Y, comme Y est plus « floue » que X, on risque d'effectuer une sur-régularisation justement dans le cas où le bruit est le plus faible, ce qui n'est pas le but recherché !

Cet estimateur ne semble donc pas convenir à notre problème, malgré sa facilité d'implémentation, par rapport à celui que nous avons retenu.

Approche de Lakshmanan et Derin

Lakshmanan et Derin [LAK89] proposent d'utiliser le critère suivant :

$$(\hat{X}, \hat{\lambda}, \hat{\delta}) = \arg\max_{X,\lambda,\delta} P(X, Y \mid \lambda, \delta) \tag{2.2}$$

qui fait intervenir la loi jointe de X et Y. Il est possible de chercher successivement $\hat{X}$ et $(\hat{\lambda}, \hat{\delta})$ en utilisant deux critères *sous-optimaux* :

$$\hat{X} = \arg\max_{X} P(X, Y \mid \hat{\lambda}, \hat{\delta}) \quad \text{et} \quad (\hat{\lambda}, \hat{\delta}) = \arg\max_{\lambda,\delta} P(\hat{X}, Y \mid \lambda, \delta)$$

Le premier correspond au maximum a posteriori pour $(\hat{\lambda}, \hat{\delta})$ donnés.
Le second peut s'écrire $P(Y \mid \hat{X}, \lambda, \delta)\, P(\hat{X} \mid \lambda, \delta)$, et comme l'attache aux données ne dépend pas des hyperparamètres, on a $P(Y \mid \hat{X}, \lambda, \delta) = P(Y \mid \hat{X})$. Il s'agit là du maximum de vraisemblance de (λ, δ) sur $\hat{X}$.

Le même type de critère a également été utilisé plus récemment par Champagnat, Goussard et Idier dans [CHM96].

Cela revient à appliquer l'estimateur précédent (2.1) à l'image $\hat{X}$ obtenue par restauration de Y avec les λ et δ courants. L'algorithme consiste à estimer successivement les hyperparamètres sous-optimaux, et à déconvoluer avec ces hyperparamètres, en répétant jusqu'à la convergence. On devrait alors tendre vers une estimée proche de $\mathcal{X}$ (image non

dégradée). Ce qui conduit à penser que l'on obtient alors un couple $(\hat{\lambda}, \hat{\delta})$ proche de celui estimé sur $\mathcal{X}$, donc qui ne dépend pas (ou très peu) du type de dégradation donnant $Y = H\,\mathcal{X} + N_\sigma$, et le problème qui se pose est le même que ci-dessus : les hyperparamètres obtenus ne paraissent pas adaptés à la déconvolution.

C'est pour cette raison que nous n'avons pas retenu non plus cet estimateur, malgré ses avantages concernant son implémentation. En effet, tout comme le précédent, il nécessite uniquement l'échantillonnage du modèle a priori, *sans attache aux données*. Comme nous le montrons dans l'annexe F, il est possible d'accélérer sensiblement ce calcul en utilisant des valeurs précalculées.

La prise en compte des données Y dans l'échantillonnage nous paraît indispensable à l'estimation correcte des hyperparamètres. C'est pourquoi nous avons choisi le critère détaillé dans le paragraphe suivant.

2.1.2 Maximum de vraisemblance sur Y (image observée)

L. Younes [YOU88, YOU89] propose d'utiliser le *maximum de vraisemblance sur Y*, c'est-à-dire de maximiser la probabilité de l'observée Y sachant les hyperparamètres (λ, δ) :

$$(\hat{\lambda}, \hat{\delta}) = \arg\max_{\lambda,\delta} P(Y \mid \lambda, \delta) \tag{2.3}$$

L'observation étant effectuée, cette vraisemblance ne dépend plus que des hyperparamètres, et la rendre maximale est une condition *nécessaire* pour l'obtention d'une image de « bonne » qualité. Pour calculer cette probabilité, on va chercher à se ramener aux distributions a priori (1.3) et a posteriori (1.4). On a :

$$P(Y \mid \lambda, \delta) = \sum_{X \in \Omega} P(Y, X \mid \lambda, \delta) \text{ avec } P(Y, X \mid \lambda, \delta) = P(Y \mid X, \lambda, \delta)\, P(X \mid \lambda, \delta) \text{ (Bayes)}$$

ce qui permet d'écrire que $P(Y \mid \lambda, \delta) = Z_Y \,/\, Z\, K_\sigma$ (voir annexe E), où Z et Z_Y sont respectivement les fonctions de partition correspondant aux distributions a priori et a posteriori (ie. les coefficients de normalisation) :

$$Z = \sum_{X \in \Omega} e^{-\Phi(X,\lambda,\delta)} \quad \text{et} \quad Z_Y = \sum_{X \in \Omega} e^{-||Y - HX||/2\sigma^2 - \Phi(X,\lambda,\delta)}$$

Toute la difficulté de l'estimation des hyperparamètres provient de ces fonctions, qui ne sont pas calculables directement en raison du trop grand cardinal de l'espace de configuration Ω (ensemble des images possibles de dimensions $N_x \times N_y$), même lorsqu'on se restreint à M niveaux de gris : on a en effet Card$(\Omega) = M^{N_x\, N_y}$. Sachant que les plus petites images (imagettes extraites des images satellitaires) sont de taille minimum 128×128, en 256 niveaux de gris, on a Card$(\Omega) = 256^{16384}$; nous allons optimiser P sans chercher à calculer explicitement Z et Z_Y.

Nous chercherons à minimiser $J(\lambda, \delta) = -\log P(Y \mid \lambda, \delta)$:

$$J(\lambda, \delta) = \log Z_Y(\lambda, \delta) - \log Z(\lambda, \delta) + \text{ const.} \tag{2.4}$$

la constante étant $\log K_\sigma$, qui ne dépend que de la statistique du bruit qui est connue.

2.2 Calcul du gradient et du hessien

Pour optimiser la log-vraisemblance sur Y, il faut évaluer ses dérivées jusqu'au second ordre (gradient et hessien) par rapport aux hyperparamètres. Il faut se référer à l'annexe E pour le calcul détaillé. A la base de ce calcul, si on note θ un des hyperparamètres, se trouve la propriété suivante :

PROPRIÉTÉ 2.2.1 (DÉRIVÉE DE LOG Z)
Si $P(X \mid \lambda, \delta) = Z^{-1} e^{-U(X \mid \lambda, \delta)}$ avec la fonction de partition $Z = \sum_{X \in \Omega} e^{-U(X \mid \lambda, \delta)}$, alors on a

$$\frac{\partial \log Z}{\partial \theta} = -E_{X \sim P(X \mid \lambda, \delta)} \left[\frac{\partial U(X \mid \lambda, \delta)}{\partial \theta} \right] \tag{2.5}$$

où $E[\,]$ désigne l'espérance de X de loi de probabilité $P(X \mid \lambda, \delta)$.

DÉMONSTRATION 2.2.1
$\partial \log Z / \partial \theta = Z^{-1}\, \partial Z / \partial \theta = \sum_{X \in \Omega} \partial U(X)/\partial \theta\, Z^{-1} e^{-U(X)}$ et comme $P(X) = Z^{-1} e^{-U(X)}$ on obtient $\sum_{X \in \Omega} P(X)\, \partial U(X)/\partial \theta$, ce qui est bien une espérance $E_{X \sim P(X)}[\partial U(X)/\partial \theta]$.

Tous les calculs de dérivées passent donc par l'évaluation de ces espérances, que l'on peut approcher par la moyenne empirique :

$$E_{X \sim P(X)} \left[\frac{\partial U(X)}{\partial \theta} \right] = \frac{1}{N} \sum_{X^k \sim P(X)} \frac{\partial U(X^k)}{\partial \theta} \tag{2.6}$$

si N est le nombre d'échantillons X tirés avec la loi de probabilité $P(X \mid \lambda, \delta)$. Comme nous pouvons le voir dans l'expression du gradient (E.1) (et du hessien (E.2)) de la -log vraisemblance

$$\nabla_\lambda = E_Y[\Phi_\lambda] - E[\Phi_\lambda] \quad \text{et} \quad \nabla_\delta = E_Y[\Phi_\delta] - E[\Phi_\delta]$$

il est nécessaire de prendre en compte deux types d'espérance :

- $E[\,]$ calculée avec $X \sim P(X) = Z^{-1} e^{-\Phi(X)}$, *sans attache aux données* (paragraphe 3.3)
- $E_Y[\,]$ calculée avec $X \sim P_Y(X) = Z_Y^{-1} e^{-||Y - HX||^2 - \Phi(X)}$, *avec attache aux données* (paragraphe 3.2)

Comment pouvons-nous construire des images X de distribution $P(X)$? Nous devons utiliser un *échantillonneur*, c'est-à-dire un algorithme qui va générer de façon itérative une suite d'images obéissant à la bonne loi de probabilité. Cette suite d'images constitue une *chaîne de Markov* ; cette notion sera développée au chapitre suivant.

Chapitre 3

Champs de Markov et échantillonnage

3.1 Champs de Markov

Soit $x = \{x_1, \dots x_m\}$ une famille de variables aléatoires sur l'ensemble S, ensemble fini des sites ($m = N_x N_y$). Chaque variable x_i prend une valeur X_i dans L, ensemble des valeurs possibles. On peut distinguer deux cas : on se restreint à $L = \{0, 1, \dots 255\}$ (images codées sur 8 bits) lorsque l'on utilise un échantillonneur de Gibbs ou Metropolis, mais l'on préfère $L = \mathbb{R}$ dans le cas général.

L'image X est un champ de Markov [AZE88] : la valeur d'un pixel X_i ne dépend que de ses voisins V_i. En effet, la régularisation fait intervenir les plus proches voisins (prise en compte des contours), et le noyau de convolution H a une taille petite devant la taille de l'image à traiter.

$$P(X_i \mid X_j,\ j \neq i) = P(X_i \mid V_i),\ \forall i \in S \tag{3.1}$$

THÉORÈME 3.1.1 (HAMMERSLEY-CLIFFORD [GEM84])
Ω est un champ de Markov sur S par rapport à un système de voisinage V_S si et seulement si il est un champ de Gibbs sur S par rapport à un système de voisinage V_S.

Grâce à ce théorème, il est possible d'écrire l'expression de la probabilité $P(X)$ en fonction d'une énergie $U(X)$; nous allons utiliser l'expression de $U(X)$ définie dans l'équation (1.2). Le paramètre λ et l'écart-type σ correspondent à la température ($T \leftrightarrow 2\sigma^2$ ou bien $T \leftrightarrow 1/\lambda^2$).

$$P_T(X) = \frac{1}{Z_T} e^{-U(X)/T} \quad \text{avec } Z_T = \sum_{X \in \Omega} e^{-U(X)/T}$$

Z_T est une constante appelée fonction de partition, par analogie avec la physique statistique, T est la température et $U(X)$ l'énergie du champ X.

Nous utilisons des *échantillonneurs* pour générer deux chaînes de Markov (X^i) possédant les propriétés d'ergodicité et d'invariance, l'une avec la distribution d'équilibre $\pi(X) = P(X \mid Y, \lambda, \delta)$ définie par l'équation (1.4), et l'autre avec $\pi(X) = P(X \mid \lambda, \delta)$ donnée par l'équation (1.3). Nous nous servons de ses éléments pour calculer les espérances (2.6), donc les gradients, pour maximiser la vraisemblance sur Y.

3.2 Echantillonnage du modèle a posteriori

Le modèle a posteriori fait intervenir notre connaissance de l'image (donc λ et δ), aussi bien que l'image observée Y et le noyau de convolution H ; il s'agit ici de fabriquer des échantillons ayant la distribution d'équilibre

$$\pi(X) = \frac{1}{Z} e^{-\frac{1}{2\sigma^2}||Y-HX||^2 - \lambda^2 \sum_{ij}\left[\varphi\left(\frac{X_{i+1,j}-X_{i,j}}{\delta}\right)+\varphi\left(\frac{X_{i,j+1}-X_{i,j}}{\delta}\right)\right]} \tag{3.2}$$

donc tenant compte de l'*attache aux données*.

3.2.1 Echantillonneurs de Gibbs et Metropolis

Etant donné la forme de π, le voisinage correspondant au champ de Markov n'est pas limité au premier ordre (4 voisins), à cause de l'opérateur H. Si H est de taille $(2p+1) \times (2p+1)$, on doit considérer le voisinage dont l'ordre correspond à tous les pixels $X_{k,l}$ dont dépend $X_{i,j}$: on a $k \in \{i-p \dots i+p\}$ et $l \in \{j-p \dots j+p\}$.

Le cœur des algorithmes de Metropolis et Gibbs est le calcul des variations d'énergie lorsque l'on modifie un pixel au hasard. Cette variation d'énergie est d'autant plus longue à calculer qu'elle fait intervenir un grand nombre de pixels, ici $(2p+1)^2$, soit 121 pour le noyau fourni par le CNES, et plus pour d'autres noyaux H plus étendus. Chaque itération de l'échantillonneur de Gibbs correspondrait (en temps de calcul) à autant de convolutions par H qu'il y a de niveaux de gris (256), sans que la séparabilité de H puisse être utilisée pour accélérer les calculs car on traite les pixels un par un, de manière séquentielle.

Il est indispensable d'utiliser un autre type d'échantillonneur, si on veut tenir compte de l'attache aux données.

3.2.2 Echantillonneur de Geman-Yang modifié

Le principe de cet échantillonneur, proposé par Geman et Yang dans [GEM95], réside dans la diagonalisation de l'opérateur de convolution (voir annexe D), ce qui équivaut à remplacer la convolution (présente dans l'attache aux données), lourde à calculer, par une multiplication pixel par pixel dans l'espace des fréquences. On utilise pour cela une transformée de Fourier (FFT).

Il s'agit ici de générer des X en utilisant une *variable auxiliaire*, avec le développement semi-quadratique (C.0.1). Au lieu de fabriquer X^{n+1} à partir de X^n comme avec Metropolis (3.3.1) ou Gibbs (3.3.2), on génère successivement X^{n+1} à partir de $B^{x\,n}, B^{y\,n}$ avec $P(X \mid B^x, B^y)$ et $B^{x\,n}, B^{y\,n}$ à partir de X^n avec $P(B^x, B^y \mid X)$.

$$P(X, B^x, B^y) \propto e^{-\frac{||Y-HX||^2}{2\sigma^2} - \lambda^2 \sum_{ij}\left\{\left(\frac{X_{i+1,j}-X_{i,j}}{\delta} - B^x_{ij}\right)^2 + \psi(B^x_{ij}) + \left(\frac{X_{i,j+1}-X_{i,j}}{\delta} - B^y_{ij}\right)^2 + \psi(B^y_{ij})\right\}}$$

L'intérêt de cette méthode est de rendre φ quadratique lorsque la variable auxiliaire b est fixée, ce qui signifie que la probabilité de X sachant B^x, B^y est *gaussienne* . La matrice de

covariance correspondante Σ est calculée à l'aide de H et des opérateurs de dérivation D_x D_y. Grâce à la décomposition semi-quadratique :

$$P(X \mid B^x, B^y) \propto e^{\left\{-X^t\left[\frac{1}{2\sigma^2}H^tH+\frac{\lambda^2}{\delta^2}(D_x^tD_x+D_y^tD_y)\right]X+2\left[\frac{1}{2\sigma^2}H^tY+\frac{\lambda^2}{\delta}(D_x^tB^x+D_y^tB^y)\right]^tX\right\}}$$

$$\text{on a} \quad \Sigma = \left(\frac{1}{2\sigma^2}H^tH + \frac{\lambda^2}{\delta^2}(D_x^tD_x + D_y^tD_y)\right)^{-1}$$

Σ est alors circulante par blocs, elle est donc *diagonalisée* par la FFT, ce qui permet un tirage de X à B fixé en une seule étape.

Le tirage de B^x, B^y à X fixé s'effectue de même en une seule passe, car les B_{ij}^x et B_{ij}^y sont indépendants deux à deux.

La modification que nous avons apportée à cette méthode consiste à employer une transformée en cosinus rapide au lieu d'une FFT, car notre noyau H et les conditions aux bords sont symétriques. En effet, l'utilisation de conditions aux bords périodiques (comme pour la FFT) entraîne l'apparition de discontinuités sur les bords de l'image, qui font naître des oscillations lors de la déconvolution. L'algorithme original travaillait avec des complexes ; nous utilisons des réels puisque nous traitons l'image Y symétrisée sur les deux axes, la DCT de Y équivaut alors à la FFT de l'image Y symétrisée.

ALGORITHME 3.2.1 (GEMAN-YANG MODIFIÉ, A POSTERIORI)
$X^0 = \hat{X}$ obtenu par restauration avec l'algorithme 3.4.1 (voir figure 3.1). On calcule d'abord la transformée de Fourier $\mathcal{F}[h^4]$ et DCT$[Y]$, ainsi que W défini ci-dessous. h^4 est la réponse impulsionnelle, génératrice de la matrice circulante H^4 correspondant à la convolution des images symétrisées (de dimension $2N_x \times 2N_y$). Les vecteurs d_x^4 et d_y^4 génèrent les opérateurs de dérivation D_x^4 et D_y^4 appliqués aux images symétrisées.

$$W = \left(\frac{1}{2\sigma^2}|\mathcal{F}[h^4]|^2 + \frac{\lambda^2}{\delta^2}\left(|\mathcal{F}[d_x^4]|^2 + |\mathcal{F}[d_y^4]|^2\right)\right)^{-1}$$

Passage de X^n à X^{n+1} :

- *Tirage de B^x, B^y en fonction des gradients de X^n, ce qui se fait **globalement**, car les b sont **indépendants** deux à deux. On utilise la loi :*

$$P_g(b) \propto \exp\left(\lambda^2\left[b(2g/\delta - b) - \psi(b)\right]\right)$$

- *On fabrique une image intermédiaire R, dont chaque pixel est une variable aléatoire **gaussienne**, de moyenne nulle et de variance $1/2$.*

- *Le tirage de X^{n+1} se fait également **en une seule étape**, dans le domaine **fréquentiel**, en fonction de B^x, B^y et R :*

$$X^{n+1} = DCT^{-1}\left[W\left(\frac{\lambda^2}{\delta}DCT\left[D_x^tB^x + D_y^tB^y\right] + \frac{1}{2\sigma^2}\mathcal{F}[h^4]DCT[Y]\right) + \sqrt{W}\,R\right]$$

où D_x^t et D_y^t sont les transposés des opérateurs de dérivation, utilisés dans le domaine ***spatial****.*

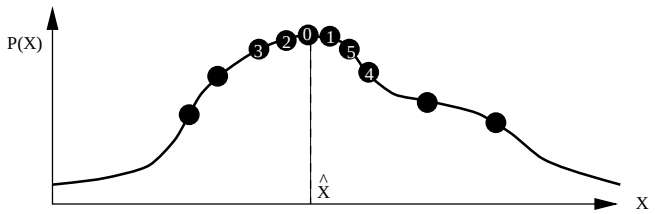


FIG. 3.1: *Justification de l'initialisation des échantillonneurs par l'image $\hat{X}$ qui maximise la probabilité $P(X)$*

3.2.3 Résultats

Nous ne présentons ici que les échantillons construits par la méthode développée ci-dessus (la seule réellement applicable).

Lorsque λ est petit, le terme de régularisation disparaît devant le terme d'attache aux données : on obtient alors des fluctuations *gaussiennes* (de variance σ^2) autour d'une moyenne qui n'est autre que l'image obtenue par déconvolution avec le critère des moindres carrés, particulièrement bruitée (les fluctuations liées à l'échantillonnage se perdent complètement dans le bruit amplifié lors de la restauration).

Lorsque λ est grand, c'est le contraire : le terme de régularisation prédomine et l'on se ramène à l'échantillonnage du modèle a priori, développé précédemment.

Pour les valeurs intermédiaires, on observe des fluctuations autour d'une moyenne, cette moyenne étant $\hat{X}$, image restaurée avec les hyperparamètres λ et δ. Ces fluctuations sont d'autant plus faibles que le rapport λ/δ est grand (voir figure 3.2).

3.3 Echantillonnage du modèle a priori

Le modèle a priori fait intervenir notre connaissance a priori de l'image, et fait donc appel aux hyperparamètres λ et δ ; il s'agit ici de fabriquer des images ayant la distribution d'équilibre

$$\pi(X) = \frac{1}{Z} e^{-\Phi(X,\lambda,\delta)} = \frac{1}{Z} e^{-\lambda^2 \sum_{ij} \left[\varphi\left(\frac{X_{i+1,j}-X_{i,j}}{\delta}\right)+\varphi\left(\frac{X_{i,j+1}-X_{i,j}}{\delta}\right)\right]} \tag{3.3}$$

donc *sans attache aux données*. Nous nous limitons ici au voisinage du premier ordre, car l'expression de l'énergie fait appel uniquement aux gradients $X_{i+1,j} - X_{i,j}$ et $X_{i,j+1} - X_{i,j}$. Etant donné les propriétés de régularité des espérances de $\sum \varphi$, $\sum \varphi_\delta$ et $\sum \varphi_{\delta\delta}$ (cf. gradient et hessien annexe E), il est possible de constituer un tableau de valeurs précalculées, avec

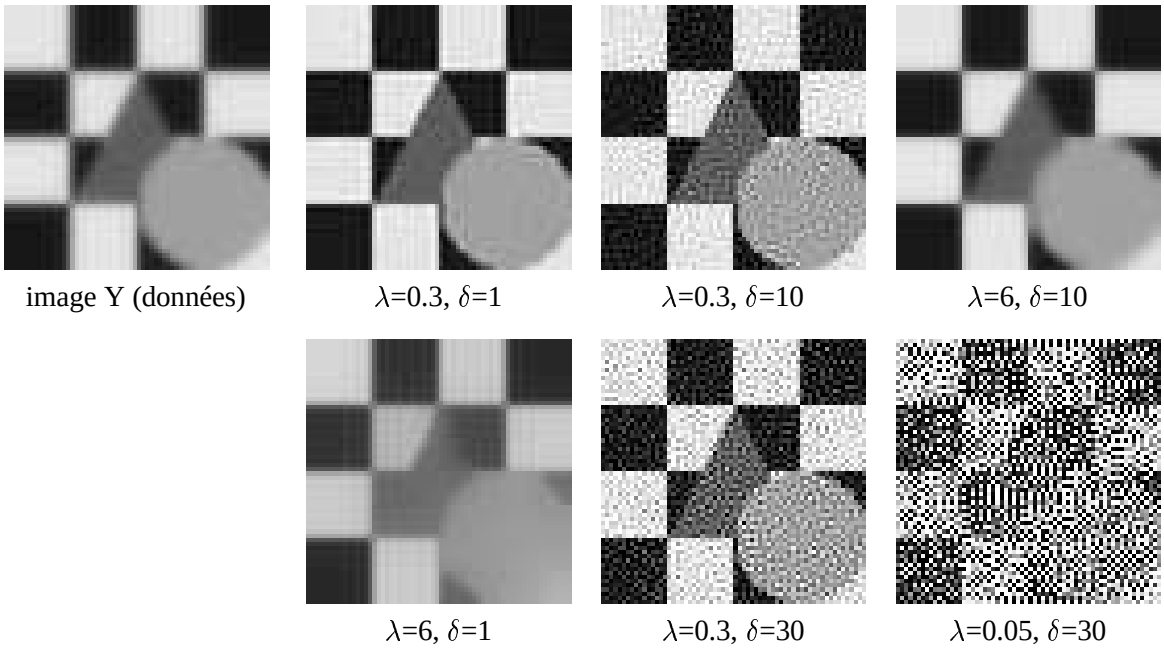


FIG. 3.2: *Echantillons a posteriori (64×64), 5 itérations (Geman-Yang modifié), avec* φ_{HS}

une discrétisation de λ et δ, et d'interpoler ce tableau afin d'obtenir les espérances pour toute valeur des hyperparamètres (voir annexe F). En effet, ces espérances ne dépendent pas des données observées.

3.3.1 Dynamique de Metropolis

L'algorithme de Metropolis [MET53] permet de générer une chaîne de Markov d'images (X^n) ayant la distribution de probabilité $P(X) \propto e^{-U(X)}$ où $U = \Phi$, énergie associée à X pour le modèle a priori. Il consiste à modifier un pixel choisi dans X et de conserver cette modification avec une certaine probabilité, qui dépend uniquement des 4 plus proches voisins, puisque X est un champ de Markov (4-connexe par construction de Φ).

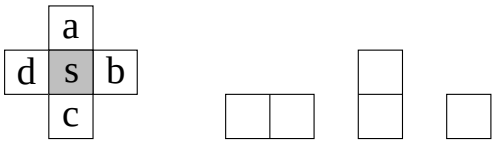


FIG. 3.3: *Voisinage du premier ordre et cliques correspondantes.*

ALGORITHME 3.3.1 (METROPOLIS)
X^0 =const. (cf. chapitre 4) ; passage de X^n à X^{n+1} en répétant autant de fois qu'il y a de pixels :

- *Choix d'un site s de valeur v (choix aléatoire ou par balayage déterministe)*
- *Tirage d'une nouvelle valeur r uniforme sur $L = \{0, 1 \ldots 255\}$*
- *Calcul de la variation d'énergie $\Delta U = U(X,\, X_s = r) - U(X,\, X_s = v)$. Elle est égale à la variation locale, facile à calculer :*
  $\Delta U = \lambda^2 \sum_{\sigma \in V_s} \varphi[(X_\sigma - r)/\delta] - \lambda^2 \sum_{\sigma \in V_s} \varphi[(X_\sigma - v)/\delta]$ avec $V_s = \{a, b, c, d\}$ (voir figure 3.3).
  - *si $\Delta U \leq 0$ accepter r : $X_s \leftarrow r$*
  - *si $\Delta U > 0$ accepter r avec la probabilité $e^{-\Delta U}$ (pour cela, on tire $\omega \in [0, 1]$ selon une loi uniforme et on accepte r si $\omega < e^{-\Delta U}$).*

On a alors $\lim_{n \to \infty} P_n(X^n) = P(X) = Z^{-1} e^{-\Phi(X)}$, au bout d'un grand nombre d'itérations (étudié au paragraphe 4.2.1). On garde les échantillons à partir du rang M, en éliminant la période transitoire pendant laquelle P n'a pas été atteinte.

3.3.2 Echantillonneur de Gibbs

On peut construire une nouvelle image X^{n+1} à partir d'une image X^n, avec la probabilité de transition $\pi(X^{n+1} \,|\, X^n) = P(s \,|\, V_s)$ en changeant la valeur du pixel s. V_s désigne ici les 4 plus proches voisins du site s.

L'échantillonneur de Gibbs [GEM84] permet, en partant d'une image X^0 quelconque, de générer, au bout d'un certain nombre d'itérations, des échantillons X^m qui obéissent à la loi de probabilité $P(X)$.

ALGORITHME 3.3.2 (GIBBS)
X^0 =const. ; passage de X^n à X^{n+1} en répétant autant de fois qu'il y a de pixels :

- *Choix d'un site s de valeur v (choix aléatoire ou par balayage déterministe)*
- *Tirage d'une nouvelle valeur r sur $L = \{0, 1 \ldots 255\}$, d'après la distribution conditionnelle locale :*
  $P(X_s = r \,|\, X_\sigma,\, \sigma \in V_s) = Z_s^{-1} \exp\left(-\lambda^2 \sum_{\sigma \in V_s} \varphi[(X_\sigma - r)/\delta]\right)$ avec $V_s = \{a, b, c, d\}$ (voir figure 3.3).
  En pratique, on construit la fonction de répartition $F(r)$:
  On tire $\omega \in [0, 1]$ selon une loi uniforme, en prenant le premier r tel que $F(r) \geq \omega$.
  Comme $r \in \{0, 1, \ldots 255\}$, on calcule $p_0, p_1 \ldots p_{255}$ avec $p_n = P(X_s = n \,|\, X_\sigma)$, puis $F(n) = \sum_{i=0}^{n} p_i$.

3.3.3 Echantillonneur de Geman-Yang modifié

Il faut se référer au paragraphe 3.2.2 et à [GEM95] pour comprendre l'origine de cet échantillonneur, que nous avons simpifié pour le cas sans attache aux données. Il suffit de considérer que Y n'intervient pas (donc H n'apparaît pas) et d'effectuer les calculs à la limite où $\lambda \to \infty$, le terme d'attache aux données disparaissant devant le terme de régularisation.

On obtient alors l'algorithme suivant, qui est bien plus rapide que les précédents car il effectue une mise à jour *globale* des sites en fonction de leur voisinage.

ALGORITHME 3.3.3 (GEMAN-YANG MODIFIÉ, A PRIORI)
$X^0 = const.$; on calcule d'abord la transformée de Fourier $\mathcal{F}[h^4]$ et DCT$[Y]$, ainsi que W_0 défi ni ci-dessous. h^4 est le vecteur générateur de la matrice circulante H^4 appliquée aux images symétrisées, de dimension $2N_x \times 2N_y$. Les vecteurs d_x^4 et d_y^4 génèrent les opérateurs de dérivation D_x^4 et D_y^4.

$$W_0 = \frac{\delta^2}{\lambda^2}\left(|\mathcal{F}[d_x^4]|^2 + |\mathcal{F}[d_y^4]|^2\right)^{-1}$$

Passage de X^n à X^{n+1} :

- *Tirage de B^x, B^y en fonction des gradients g de X^n, ce qui se fait **globalement**, car les b sont **indépendants** deux à deux. On utilise la loi :*

$$P_g(b) \propto \exp\left(\lambda^2\left[b(2g/\delta - b) - \psi(b)\right]\right).$$

- *Le tirage de X^{n+1} se fait également **en une seule étape**, dans le domaine fréquentiel :*

$$X^{n+1} = DCT^{-1}\left[\frac{\lambda^2}{\delta}W_0\, DCT\left[D_x^t B^x + D_y^t B^y\right] + \sqrt{W_0}\, R\right]$$

  *où D_x^t et D_y^t sont les transposés des opérateurs de dérivation, utilisés dans le domaine spatial. R une image dont chaque pixel est une variable aléatoire **gaussienne**, de moyenne nulle et de variance $1/2$.*

3.3.4 Résultats

L'aspect des échantillons dépend évidemment de la valeur de λ et δ (voir figure 3.4). Lorsque λ est petit, les images X sont équiprobables, les valeurs des pixels sont distribuées pratiquement de manière uniforme sur $\{0, 1, \ldots 255\}$, car si $\lambda \to 0$, $U(X) \to 0 \ \forall \delta \neq 0$.

Lorsque λ est grand, l'aspect varie en fonction du rapport λ/δ : si ce rapport est grand, tous les gradients non nuls sont pénalisés, car ils augmentent beaucoup l'énergie $U(X)$ et donc les images les plus probables seront des constantes, avec de faibles fluctuations (gradients de l'ordre de ± 1). Par contre, si le rapport λ/δ est plus petit, les fluctuations ont une plus grande amplitude (d'autant plus grande que ce rapport est faible).

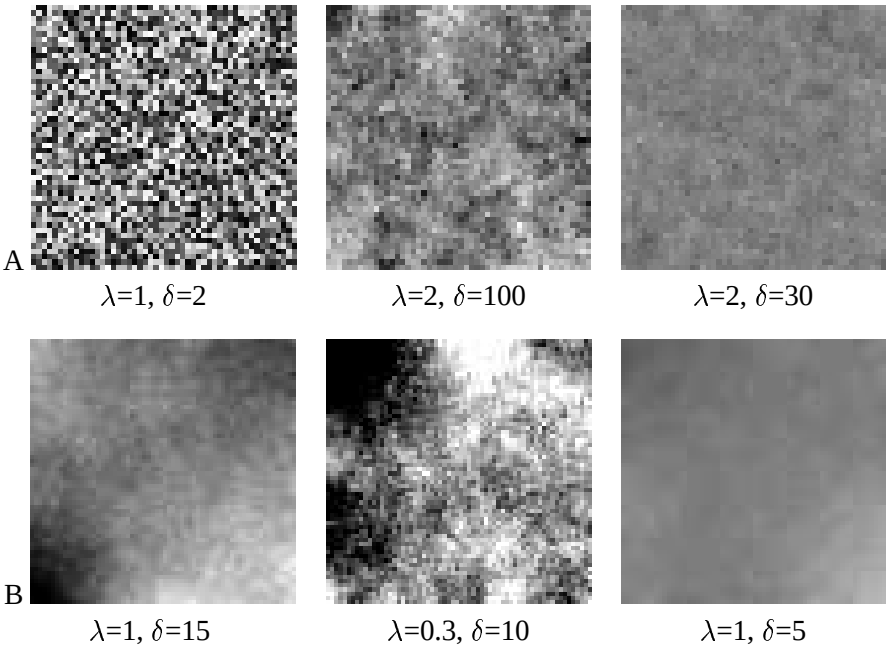


FIG. 3.4: *Exemples d'échantillons a priori : échantillonneurs de Gibbs (A) et Geman-Yang modifié (B)*

Finalement, les échantillons produits sont peu corrélés lorsque λ est faible. La corrélation devient importante au-delà d'une certaine valeur de λ (le seuil dépend peu de δ), le réseau de pixels « s'organise », devient structuré : les interactions entre voisins sont alors suffisantes pour générer des textures, faciles à distinguer. On dit que le modèle présente une *transition de phase* [GEO88], en passant d'une phase désordonnée à une phase plus structurée avec des interactions à longue portée. La brutalité de la transition de phase varie avec le modèle : lorsque le seuil entre la partie quadratique et la partie non-quadratique de φ est net, la transition de phase est très brutale (on a, pour δ faible, une discontinuité de l'énergie des échantillons). Cette transition s'estompe lorsque δ augmente (on se retrouve alors dans la partie quadratique de φ).

3.4 Application des échantillonneurs à la restauration

3.4.1 Techniques de relaxation stochastique

On peut effectuer la restauration par des algorithmes *stochastiques*, qui maximisent la *probabilité* $P(X\,|\,Y) \propto e^{-U(X)}$. Lorsque $U(X)$ n'admet pas un minimum unique, ces techniques permettent d'approcher le minimum global, alors que les méthodes de descente comme 1.5.1 seront bloquées par n'importe quel minimum local.

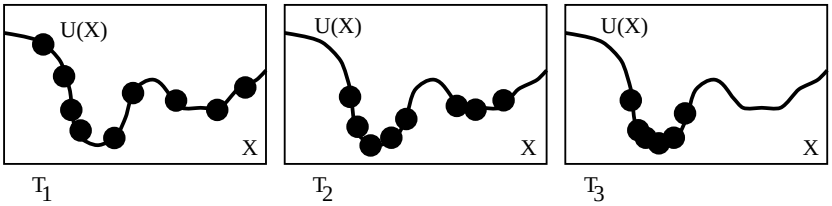


FIG. 3.5: *Recuit simulé : échantillonnage à températures décroissantes ($T_1 > T_2 > T_3$)*

Le recuit simulé [LAA87], en introduisant la température T dans l'expression de P, permet d'étaler la distribution de X (à la limite où $T \to \infty$, tous les X sont équiprobables) et de sortir ainsi des minima locaux. Les X sont alors distribués autour de $\hat{X}$ (cf. équation 1.1), avec une variance qui augmente avec T.

On fabrique, grâce à un échantillonneur basé sur la probabilité a posteriori (paragraphe 3.2), une suite d'images (X^n) avec des températures (T_n) *décroissantes*. Au bout de 100 à 1000 itérations, la suite converge vers le minimum *global* si $T_n \geq k/\log(n)$ (voir fig. 3.5).

3.4.2 Un nouvel algorithme déterministe

On peut également utiliser des méthodes de relaxation *déterministes* ; leur intérêt est de converger beaucoup plus rapidement, mais vers un minimum *local*.

L'un d'entre eux, l'ICM, introduit par Besag [BES74], équivaut à employer la technique de recuit ci-dessus en prenant une température $T \simeq 0$, au sens où tous les changements sont acceptés. On utilise pour cela un échantillonneur de Gibbs, et pour chaque pixel on choisit la valeur qui minimise la probabilité conditionnelle locale. Lorsque l'on échantillonne à faible température, la distribution de X se concentre autour d'une image qui maximise localement la probabilité, et l'on génère alors une suite d'images qui converge vers cet optimum. Nous proposons d'utiliser un échantillonneur de type Geman-Yang au lieu de l'algorithme de Gibbs.

Comme nous le verrons au chapitre suivant, une des difficultés de l'échantillonnage provient de la non-linéarité du modèle et de l'attache aux données. L'algorithme de Geman et Yang [GEM95] permet, comme on l'a vu, de générer des X en utilisant une variable auxiliaire. Le problème est traité dans l'espace des fréquences, ce qui permet de remplacer la convolution par une multiplication.

Nous avons repris le principe de cet échantillonneur en effectuant les deux modifications suivantes :

- $\mathbf{T \simeq 0}$: on ne cherche plus à échantillonner $P(X \mid Y)$ mais à calculer (de manière itérative) l'image $\hat{X}$ qui maximise cette probabilité. L'algorithme devient alors déterministe, on garde uniquement le X moyen en éliminant les fluctuations.

- Utilisation d'une **transformée en cosinus (DCT)** au lieu d'une transformée de Fourier (FFT). On respecte ainsi les conditions au bord de Neuman (dérivées nulles). Ceci

équivaut à symétriser l'image Y sur les deux axes, et à traiter une image 4 fois plus grande, mais avec un temps de calcul correspondant à l'image d'origine.

La convergence est généralement obtenue en 3 à 10 itérations, comme on peut le voir sur la figure 3.6. Ce nombre varie selon le type d'image, la dégradation qu'elle a subie et la valeur de λ et δ. Pour pouvoir remplacer les FFT par des DCT, il est nécessaire que le noyau H soit symétrique (c'est le cas pour les images du CNES, voir annexe A).

Cet algorithme est une version fréquentielle de Legend présenté en 1.5.1.

ALGORITHME 3.4.1 (RESTAURATION ICM-DCT)
$X^0 = Y$; on calcule d'abord la transformée de Fourier $\mathcal{F}[h^4]$ et DCT$[Y]$, ainsi que W défi ni ci-dessous. h^4 est le vecteur générateur de la matrice circulante H^4 appliquée aux images symétrisées, de dimension $2N_x \times 2N_y$. Les vecteurs d_x^4 et d_y^4 génèrent les opérateurs de dérivation D_x^4 et D_y^4.

$$W = \left(\frac{\lambda^2}{\delta^2} \left(|\mathcal{F}[d_x^4]|^2 + |\mathcal{F}[d_y^4]|^2 \right) + \frac{1}{2\sigma^2} |\mathcal{F}[h^4]|^2 \right)^{-1}$$

Répéter (jusqu'à la convergence) :

- $(\hat{B^x}, \hat{B^y}) = \arg\inf_{B^x, B^y} \Phi^*(X, B^x, B^y)$ *(Φ^* est défi nie en 1.4)*
  Ceci se fait en une seule étape, suivant le théorème semi-quadratique :
  $\Rightarrow \quad b_{ij}^{xy} = g_{ij}^{xy}/\delta - \frac{1}{2}\varphi'(g_{ij}^{xy}/\delta).$

- $\hat{X} = \arg\inf_X \left[||Y - HX||^2/2\sigma^2 + \Phi^*(X, B^x, B^y) \right]$
  Ce qui se fait également ***en une seule étape****, dans le domaine* ***fréquentiel*** *:*

$$\hat{X} = DCT^{-1} \left[W \left(\frac{\lambda^2}{\delta} DCT \left[D_x^t B^x + D_y^t B^y \right] + \frac{1}{2\sigma^2} \mathcal{F}[h^4] DCT[Y] \right) \right]$$

  où D_x^t et D_y^t sont les transposés des opérateurs de dérivation, utilisés dans le domaine ***spatial****.*

Pour l'image du CNES (512×512) on doit effectuer environ 140 opérations par pixel et par itération (voir annexe G). Si les hyperparamètres sont fixés, et si l'on traite une série d'images ayant les mêmes hyperparamètres optimaux, il est possible d'effectuer la restauration à bord du satellite, car l'algorithme est rapide. De plus, la DCT qu'il utilise est particulièrement bien adaptée à certains algorithmes de compression, pouvant être utilisés en aval.

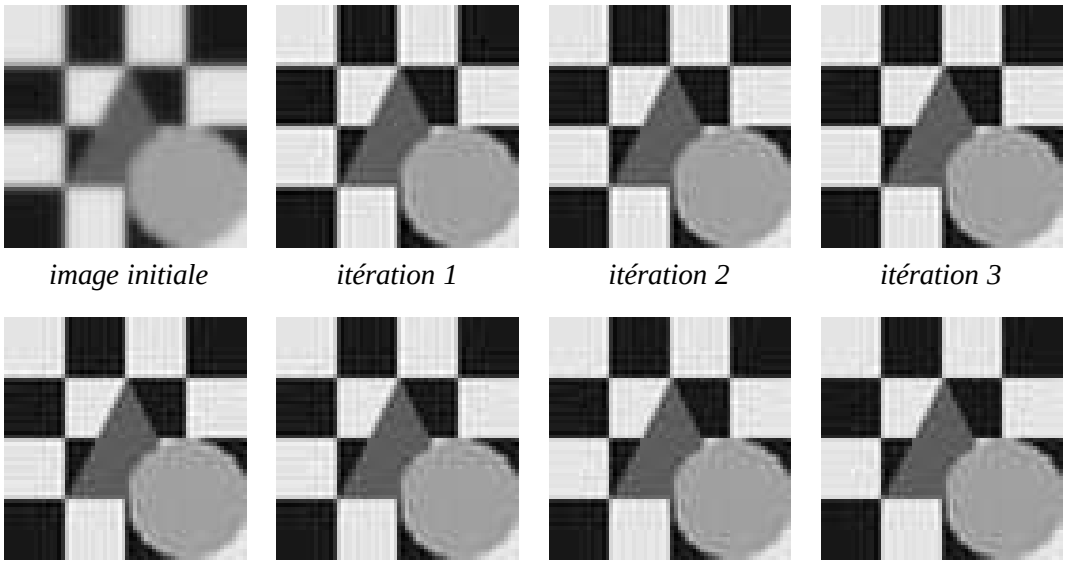


image initiale *itération 1* *itération 2* *itération 3*

itération 4 *itération 5* *itération 6* *itération 7*

FIG. 3.6: *Résultat de la restauration par l'algorithme 3.4.1 (avec* $\lambda = 0.2$, $\delta = 2.5$*)*

Chapitre 4

Difficultés de l'échantillonnage

4.1 Convergence : études théoriques

On peut trouver dans [ORU96, TAN96, ROB96] des indications concernant l'étude théorique de la convergence des chaînes de Markov. Nous retiendrons essentiellement que la possibilité de passer d'un état X^i à tout autre état X^j (par la matrice de transition), est une *condition suffisante* de convergence. Ainsi, les algorithmes de Gibbs et Metropolis, par construction, permettent à la chaîne (X^n) qu'ils génèrent de converger vers la distribution stationnaire $\pi(X)$.

Ce qui nous intéresse particulièrement, c'est la *vitesse de convergence*, c'est-à-dire le nombre d'itérations à effectuer pour que la probabilité p_n au temps n converge vers la distribution invariante π. Cette vitesse, dans le cas où l'espace des états Ω est fini (c'est le cas lorsque l'on utilise l'algorithme de Gibbs ou de Metropolis), dépend de la deuxième plus grande valeur propre de T, T étant la matrice de transition permettant de passer de X^n à X^{n+1} :

$$P_{n+1}(X) = \sum_{\{X'\}} P_n(X')\, T_n(X', X) \tag{4.1}$$

En pratique, les valeurs propres ne sont pas calculables en raison de la dimension de l'espace Ω, d'où la nécessité des *études pratiques* de la vitesse de convergence.

Notre algorithme de type Geman-Yang modifié fonctionne dans un espace Ω de dimension infinie, chaque pixel étant réel, contrairement aux autres échantillonneurs. On ne peut donc plus parler de matrice de transition entre les différents états. Les transitions ne s'effectuent pas directement de X^i à X^j, mais de X à B et de B à X. Une condition suffisante de convergence est, comme on l'a vu précédemment, la possibilité de passer d'un état à l'autre (ici par l'intermédiaire d'une image auxiliaire B).

On cherche ici à échantillonner la loi jointe $P(X, B)$, en utilisant les transitions $P(X \mid B)$ et $P(B \mid X)$. Cette méthode est correcte, car la mesure $P(X, B)$ est invariante par rapport à ces deux transitions. La réversibilité est en effet vérifiée :

$$\begin{aligned} P(X,B).P(X' \mid B) &= P(X',B).P(X \mid B) \quad \forall X, X' \in \Omega \\ P(B,X).P(B' \mid X) &= P(B',X).P(B \mid X) \quad \forall B, B' \in \Omega \end{aligned}$$

En ce qui concerne notre algorithme d'estimation, C. Geyer montre dans [GEY92] une certaine forme de convergence (qu'il appelle hypoconvergence) de la -log vraisemblance calculée par une méthode de Monte Carlo, vers la -log vraisemblance exacte.

4.2 Convergence : étude expérimentale

4.2.1 Comparaison des différents algorithmes

Voici un aperçu des performances comparées des différents algorithmes, lorsqu'il s'agit d'échantillonner le modèle a priori (pour l'attache aux données la comparaison n'est pas possible car une seule méthode est applicable). Il faut noter que seule la comparaison des vitesses de calcul est possible, entre l'algorithme de Geman-Yang et les autres, car les échantillons générés sont différents. En effet, l'espace de configuration n'est pas le même (les images sont réelles dans le cas Geman-Yang, alors que les pixels ne prennent que 256 valeurs possibles pour Gibbs et Metropolis, pour des raisons pratiques).

- **Echantillonneur de Metropolis**
  Il faut au minimum $5 \times M$ itérations, M étant le nombre de niveaux de gris (255 dans notre cas) dans les cas les plus favorables. Une itération consiste à balayer tous les sites une fois, et à les perturber en acceptant la perturbation avec une certaine probabilité. Le grand nombre d'itérations vient du fait que le taux d'acceptation est assez faible, il faut visiter chaque site un grand nombre de fois avant que l'image obéisse à la bonne loi de probabilité.

- **Echantillonneur de Gibbs**
  Il faut ici au moins 5 itérations dans les cas favorables. Une itération de l'échantillonneur de Gibbs consiste là aussi à balayer les sites, mais *le taux d'acceptation est de 100%*, contrairement à Metropolis. Pour chaque pixel on tire une valeur avec une *probabilité conditionnelle* dépendant des voisins, ce qui implique le calcul d'une fonction de répartition, soit M valeurs de P, pour chaque pixel. Les itérations sont M fois plus longues, mais plus efficaces ; l'efficacité l'emporte sur le nombre de calculs, ce qui rend cet échantillonneur environ 2 fois plus rapide que le précédent.

- **Echantillonneur de Geman-Yang modifié**
  En général 3 à 5 itérations suffisent à obtenir de bons échantillons ; si l'on tient compte du nombre d'opérations (cf. annexe G) à effectuer, cet algorithme est le plus rapide. La vitesse de convergence s'explique par le traitement plus *global* du problème (une partie est traitée dans l'espace des fréquences par une DCT), ce qui permet d'établir plus rapidement l'interaction à longue portée correspondant à certaines valeurs d'hyperparamètres (alors qu'il faut attendre la propagation des influences entre voisins sur toute l'image d'une itération à l'autre, pour Gibbs et Metropolis).

Des versions plus élaborées de ces échantillonneurs ont été proposées par différents auteurs, la plupart sont une combinaison des algorithmes classiques, comme Gibbs et Me-

tropolis [LIU96], mais les performances ne sont pas meilleures dans notre cas. Il faut citer également l'algorithme de Swendsen-Wang, décrit dans [SWE87] et appliqué par Higdon au modèle a priori dans [HIG97], mais il a été montré dans [GAU97] que cette méthode n'améliore pas notablement la vitesse de convergence, sauf dans le cas simple d'un modèle d'Ising ou de Potts [MOR96a].

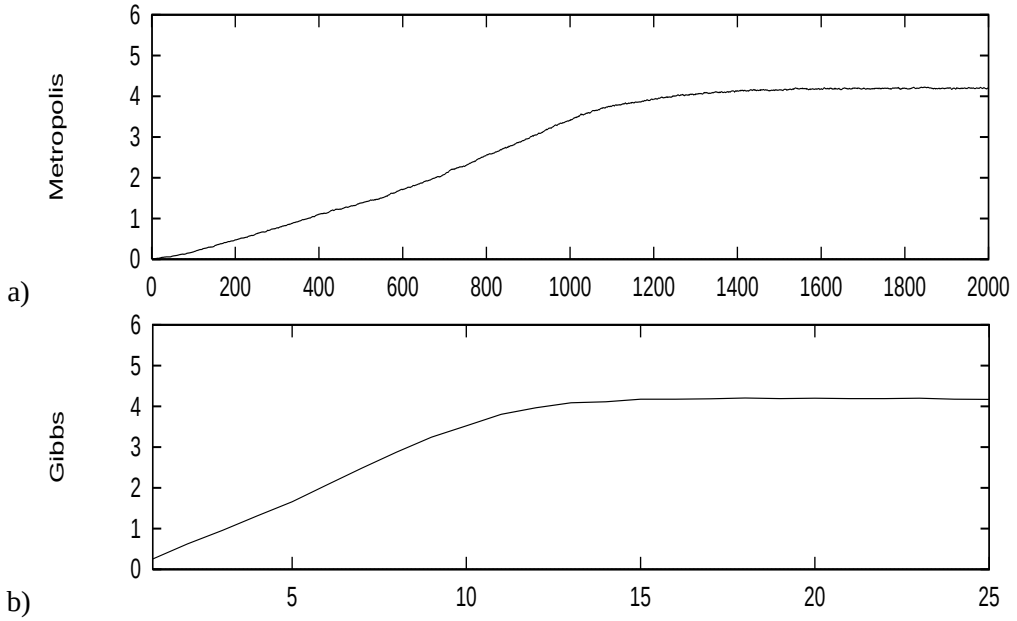


FIG. 4.1: *Vitesse de convergence des échantillionneurs de Metropolis et Gibbs : énergie en fonction du nombre d'itérations (modèle a priori, φ_{GM}) avec $\lambda = 1.5$, $\delta = 1$.*

4.2.2 Influence des hyperparamètres et transition de phase

Comme on peut le voir sur la figure 4.2, le choix des hyperparamètres peut influencer la vitesse de convergence. La chaîne de Markov atteint l'équilibre en 2 ou 3 itérations en général, mais il est parfois nécessaire d'effectuer 5 à 10 itérations de plus.

On initialise en effet par $\hat{X}$ qui est le maximum de P, la distribution est plus ou moins concentrée autour de ce mode, selon l'écart-type du bruit σ et le rapport λ/δ. Si ce rapport est faible ou si σ est grand, la distribution « s'étale » et il paraît normal que l'on atteigne moins rapidement l'équilibre. Dans le cas contraire, la situation est proche de l'ICM (échantillonnage à température nulle en théorie, voir 3.4.2), et l'on s'écarte très peu de $\hat{X}$.

Lorsque l'on échantillonne le modèle a priori, on peut observer une *transition de phase* (fig. 4.3), selon la netteté du seuil associé au modèle, seuil en-dessous duquel celui-ci est quadratique et filtre le bruit, et au-delà duquel les contours sont préservés. Les φ non convexes, comme φ_{GM} ou φ_{QT}, présentent une forte transition de phase. Cela signifie que

pour un δ donné (généralement inférieur à 5) les échantillons n'ont pas du tout le même aspect d'un côté et de l'autre de la transition. Lorsque l'on augmente λ, la somme des φ (énergie divisée par λ^2, qui ne dépend donc pas explicitement de λ) décroit brusquement lorsque λ dépasse λ_c (de façon discontinue si $\delta \to 0$), et présente deux plateaux pour $\lambda < \lambda_c$ et $\lambda > \lambda_c$.

Dans la zone de transition, l'échantillonneur a du mal à se décider s'il va tendre vers une image du type constant ($\lambda > \lambda_c$) ou très bruité ($\lambda < \lambda_c$). Quelle que soit l'initialisation de l'algorithme, *la convergence est assez lente autour de* λ_c, ce qui peut poser un problème (voir fig. 4.4). Mais la taille de cette zone est assez réduite, et la discontinuité s'estompe lorsque δ augmente, car les gradients g_i/δ se retrouvent dans la zone quadratique de φ et le modèle quadratique ne présente aucune discontinuité de l'énergie (la distribution est gaussienne dans ce cas).

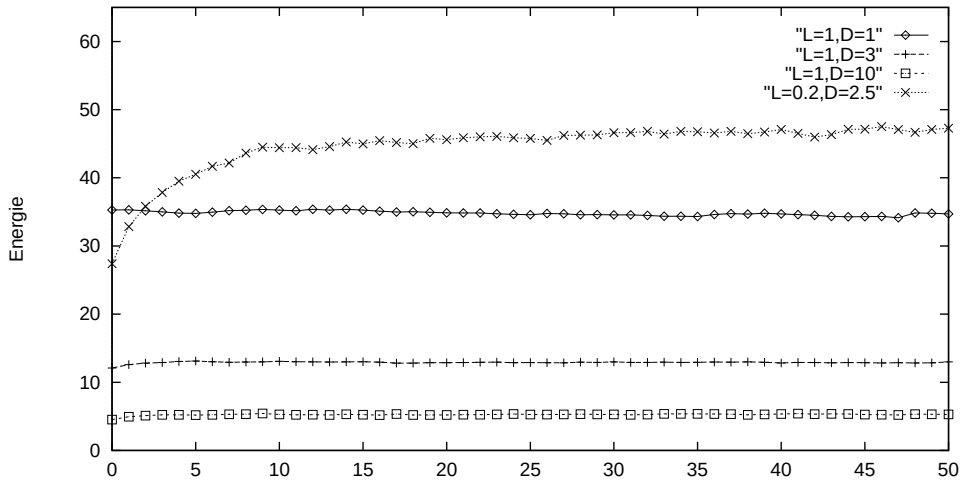
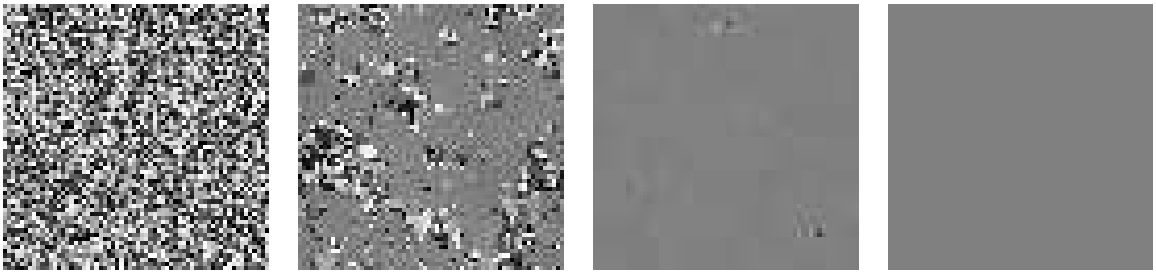


FIG. 4.2: *Energie des échantillons a posteriori (algorithme de Geman-Yang modifié), en fonction du nombre d'itérations (φ_{HS}) ($L = \lambda, D = \delta$)*

4.2.3 Sensibilité aux conditions initiales

Lorsque l'on cherche à échantillonner le modèle a priori, trois principaux types d'image peuvent être utilisés pour initialiser les algorithmes :

- Image aléatoire, dont chaque pixel a une valeur uniformément distribuée dans l'ensemble des labels $L = \{0, 1 \ldots 255\}$.
- Image constante, chaque pixel ayant une valeur fixée, en général 128 (pour éviter tout dépassement de dynamique)



$\lambda = 0.5$ $\lambda = 1.5$ $\lambda = 1.8$ $\lambda = 3.0$

Echantillons a priori autour de la transition de phase pour $\delta = 5$

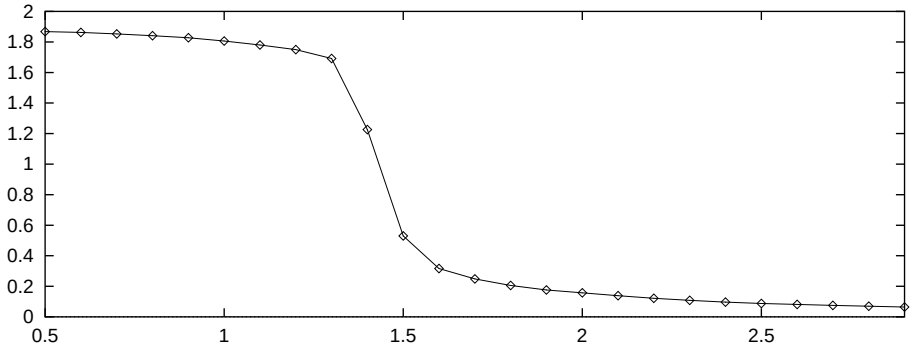


FIG. 4.3: *Energie des échantillons a priori divisée par* λ^2 *(échantillonneur de Gibbs), en fonction de* λ*, pour* $\delta = 5$ *(avec* φ_{GM} *non convexe) :*$\lambda_c \simeq 1.4$

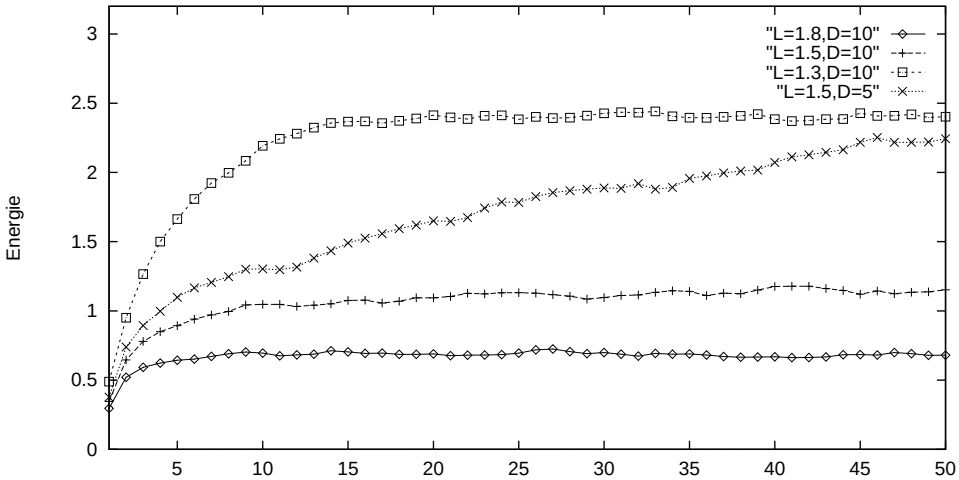


FIG. 4.4: *Energie des échantillons du modèle a priori* φ_{GM} *(échantillonneur de Gibbs), en fonction du nombre d'itérations, près de la transition de phase (*$L = \lambda, D = \delta$*)*

– Echantillon calculé pour des valeurs λ', δ' proches de λ, δ

Comme on peut le voir sur la figure 4.5, avec l'échantillonneur de Gibbs, la convergence est très lente en général si les conditions initiales sont aléatoires. Selon la valeur des hyperparamètres, les réalisations les plus probables sont plus ou moins « bruitées », avec des gradients plus ou moins marqués. Lorsque λ/δ est faible, elles sont pratiquement aléatoires, c'est le seul cas où l'algorithme peut être initialisé de façon aléatoire, et il converge alors en quelques itérations. Par contre, dès que λ devient supérieur à δ, ce type d'initialisation entraîne l'apparition de zones plus ou moins homogènes séparées par de forts gradients, et la taille de ces zones augmente avec le nombre d'itérations. Le modèle tend à créer des interactions à longue portée, les contours sont instables, et les zones finissent par fusionner. Au bout d'un très grand nombre d'itérations (qui dépend de la taille de l'échantillon) il ne reste plus qu'une seule zone assez homogène (gradients faibles).

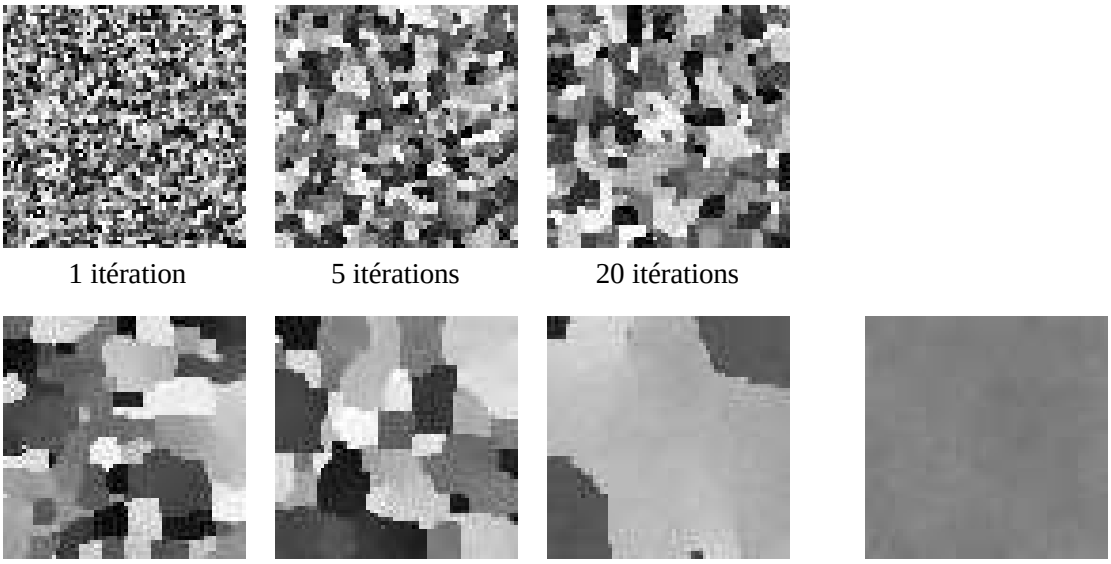
Ce type d'image est en revanche obtenu en quelques itérations seulement (5 ou 10 en-dehors de la zone de transition) lorsque l'on part d'une image constante. Ce qui démontre l'importance du choix des conditions initiales, malgré les preuves théoriques assurant la convergence quelle que soit l'image de départ. Sur la figure représentant l'énergie, on voit que l'équilibre est atteint très rapidement dans ce cas, alors que l'énergie décroît progressivement et lentement dans l'autre cas, on peut alors espérer une convergence très lente, au bout de 1000 ou 10000 itérations...

Lorsque l'on initialise par un échantillon déjà calculé pour des valeurs proches des hyperparamètres, la convergence est accélérée de façon sensible. La convergence est d'autant plus rapide que les valeurs λ, δ sont proches de celles qui correspondent à l'échantillon de départ. Cette technique est utilisée dans l'algorithme d'estimation présenté au chapitre suivant.

Un cas intéressant est l'initialisation par une image présentant de forts gradients uniquement sur un axe, par exemple des bandes verticales ou horizontales. Pour λ/δ supérieur à 1, on constate que ce type d'image constitue un état *métastable* : il sera pratiquement impossible d'en sortir, mais on n'a pas atteint l'équilibre global, qui est une image quasiment constante. Ceci s'explique, dans Metropolis ou Gibbs, en considérant les 4 voisins d'un pixel : sur une frontière de bande (passage par exemple de 100 à 200) 3 pixels sur 4 ont la même valeur, et la probabilité de tirer cette valeur sera très proche de 1 vu le rapport λ/δ. On va alors conserver cette valeur et il sera très difficile de déplacer la frontière des zones homogènes de cette manière, pixel par pixel.

4.2.4 Critère d'arrêt

On peut trouver dans [BRO97] toute une série de critères de convergence appliqués aux algorithmes de type MCMC. Certains sont basés sur la norme L^2 entre la distribution courante, à l'itération n, et la distribution d'équilibre π que l'on cherche à atteindre. On montre alors que cette distance peut être estimée à l'aide de plusieurs chaînes de Markov parallèles (indépendantes), initialisées différemment, en utilisant le noyau de transition (4.1)



1 itération | 5 itérations | 20 itérations

50 itérations | 100 itérations | 500 itérations | ∞ ou init. constante

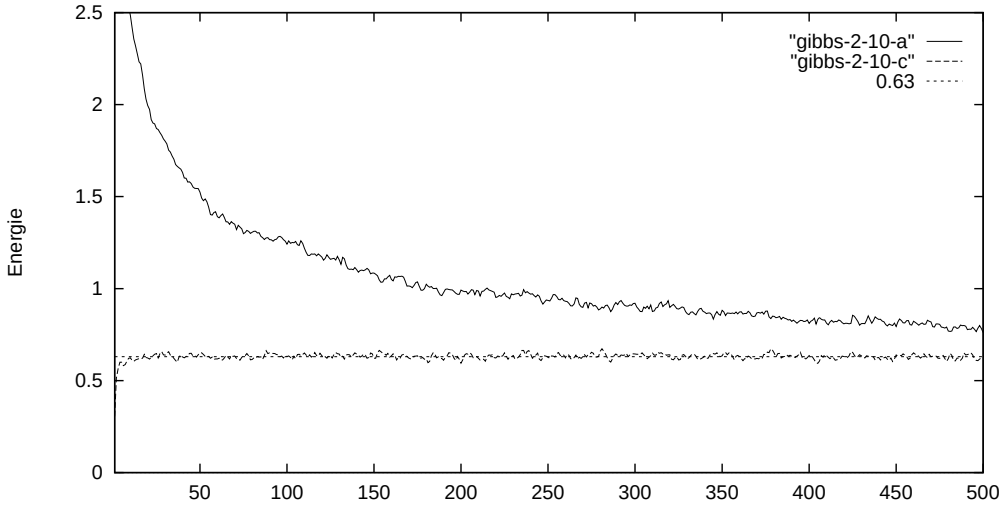


FIG. 4.5: *Echantillons a priori, obtenus avec $\lambda = 2$ et $\delta = 10$ et φ_{GM} (algorithme de Gibbs) avec une initialisation aléatoire (au-dessus), et leur énergie Φ, en fonction du nombre d'itérations. Cette énergie est comparée à celle obtenue avec une initialisation constante (en-dessous), on a $\Phi \to 0.63$.*

permettant de passer d'un échantillon à un autre (comme on le fait dans les algorithmes de Gibbs ou Metropolis).

D'autres cherchent à obtenir une borne supérieure de la norme L^1 entre des probabilités de transition de plusieurs chaînes définies de la même façon. Il est également possible d'utiliser les taux d'acceptation et de réjection lorsqu'on utilise un échantillonneur de Metropolis.

La méthode que nous utilisons consiste à tracer l'énergie des échantillons (X^n) (ou une de ses dérivées par rapport à λ ou δ) en fonction du nombre d'itérations, et de voir à partir de combien d'itérations cette énergie se « stabilise » ; en effet durant la phase d'initialisation, tant que l'équilibre n'est pas atteint, cette quantité croît sensiblement si l'on part d'une image constante (à énergie minimale), ou décroît si l'on part, au contraire, d'une image très bruitée. Il est difficile d'établir un critère d'arrêt fiable, à cause des fluctuations de la quantité que l'on mesure sur les échantillons. Il est préférable de calculer de nombreuses chaînes en parallèle (n) pour effectuer des moyennes et réduire (d'un facteur $\sqrt{n}$) les fluctuations. Mais ceci ne peut constituer qu'une étude *préalable* du nombre d'itérations nécessaire à la convergence ; pas question de procéder ainsi durant l'estimation des paramètres, car cela allonge considérablement le temps de calcul.

4.3 Difficultés liées à la convolution

Comme on l'a vu au paragraphe 3.2, lors de l'échantillonnage avec attache aux données, une des principales difficultés lorsqu'on utilise un algorithme de Gibbs ou Metropolis est la lenteur de calcul liée à l'ordre élevé du voisinage dont dépendent les pixels (ordre qui augmente avec la taille du noyau H).

Une autre difficulté dans l'emploi de ces échantillonneurs réside dans la convergence, qui n'est plus assurée, notamment lorsque les conditions initiales ne sont pas judicieusement choisies. Les pixels sont d'autant plus difficiles à perturber que leur voisinage est grand, et le moindre maximum local de la probabilité suffirait à bloquer l'échantillonneur (cf. cas des bandes verticales ou horizontales déjà mentionné).

4.4 Conclusion

Les conditions initiales doivent être les plus proches possibles de $\hat{X}$ qui maximise la loi de probabilité que l'on échantillonne, soit une constante pour le modèle a priori et l'image restaurée avec les hyperparamètres (λ, δ) pour le modèle avec attache aux données (voir fig. 3.1).

Le nombre d'itérations à effectuer dans chaque cas variant en fonction des images de données utilisées, et des hyperparamètres, il est nécessaire d'effectuer une *étude préalable* de la convergence, en étudiant par exemple le comportement des moments (espérance de l'énergie Φ, etc.) en fonction des itérations. Aucun critère d'arrêt n'est réellement fiable (en particulier lorsqu'il y a une transition de phase) et le tracé de diagrammes de convergence,

en quelques points-clés de l'espace des hyperparamètres, doit permettre de vérifier que les échantillons fabriqués obéissent effectivement à la loi de probabilité choisie.

Le choix de l'échantillonneur ne laisse guère de doutes : on préfèrera celui de Geman-Yang modifié (algorithme 3.2.1) en raison de sa rapidité et de sa manière globale de traiter le problème, plus apte à sortir des situations difficiles.

Chapitre 5

L'algorithme d'estimation et restauration simultanées

5.1 « Markov Chain Monte Carlo Maximum Likelihood »

5.1.1 Description

L'algorithme que nous utilisons pour estimer les meilleurs hyperparamètres est basé sur une méthode de descente. Il s'agit en effet de minimiser la -log vraisemblance par rapport à λ, δ :

$$(\hat{\lambda}, \hat{\delta}) = \arg\min_{\lambda,\delta} [-\log P(Y \mid \lambda, \delta)]$$

Cette méthode est, dans le cas le plus général, une descente de gradient à pas constant. On peut préférer des méthodes plus élaborées comme Newton-Raphson, qui utilise le hessien. Mais le critère n'étant pas convexe en fonction de δ, ce type de descente ne peut être appliqué que suivant λ. Les conditions de convergence de l'algorithme seront discutées au paragraphe 5.1.2.

ALGORITHME 5.1.1 (MCMCML)

- ***Initialisation*** *:*
  Deux choix possibles pour les hyperparamètres de départ λ_0, δ_0 :
  - *Rapport λ/δ correspondant au meilleur filtre de Wiener (correspondant à φ quadratique) ; δ choisi de manière à pénaliser les gradients dus au bruit, et à préserver les contours.*
  - *λ_0, δ_0 sont le résultat de l'estimation effectuée sur une autre image, prise dans les mêmes conditions, et si possible de même type géographique (ville, champs, montagne...).*
- ***Calcul de*** *$\hat{X}$:*
  L'image $\hat{X}$ est obtenue à partir de Y par restauration ICM-DCT (cf. algorithme 3.4.1) avec les hyperparamètres (λ_n, δ_n).
- ***Calcul des espérances*** *$E[\,]$* ***et*** *$E_Y[\,]$* ***avec (***λ_n, δ_n***) :***
  On utilise deux échantillonneurs de type Geman et Yang modifiés (cf. algorithmes

3.2.1 et 3.3.3) pour estimer les espérances (2.6) avec et sans attache aux données. Dans le cas de l'attache aux données, $\hat{X}$ sert à initialiser l'algorithme.

- ***Passage de (λ_n, δ_n) à ($\lambda_{n+1}, \delta_{n+1}$) :***
  Si ∇_λ et ∇_δ (cf. annexe E) sont les dérivées de la -log vraisemblance par rapport à λ et δ, on a :
  $$\begin{pmatrix} \lambda_{n+1} \\ \delta_{n+1} \end{pmatrix} = \begin{pmatrix} \lambda_n \\ \delta_n \end{pmatrix} - \alpha \begin{pmatrix} \nabla_\lambda \\ \nabla_\delta \end{pmatrix}$$
  où α est un pas fixé (ou qui peut dépendre de λ et δ, pour faciliter la convergence).

- ***Critère d'arrêt :***
  On considère que l'on a convergé si la norme du gradient devient inférieure à un certain seuil : $\nabla_\lambda^2 + \nabla_\delta^2 < \epsilon$.

L'algorithme s'arrête lorsque la norme du gradient passe sous le seuil ϵ. Il peut alors être bloqué par tout *minimum local*. En effet, la non-convexité du critère à optimiser, liée à la façon dont δ intervient dans la fonction φ, rend possible l'existence de minima locaux, et l'on comprend alors que le choix des hyperparamètres de départ détermine la convergence vers certains de ces minima. On va chercher à privilégier les couples $(\hat{\lambda}, \hat{\delta})$ pour lesquels δ, présent dans $\varphi(g/\delta)$ où g est un gradient de l'intensité, permet de profiter de la forme de φ, quadratique pour les faibles gradients (liés au bruit) et non-quadratique pour préserver les plus forts (ie. les contours).

Il est souhaitable de forcer δ à rester proche de $\tilde{\delta}$, lié au seuil du bruit : si $\varphi(u)$ est quadratique pour $u < u_c$, on choisit $\tilde{\delta}$ tel que $g_b/\tilde{\delta} < u_c$, g_b étant les gradients dus au bruit, majorés par 5σ environ (car le bruit est gaussien de variance σ). On prend alors $\tilde{\delta} = 5\sigma/u_c$. Dans la pratique, on préfère ainsi fixer $\delta \simeq 1.5\sigma$ (fonction φ_{HS}), et chercher $\hat{\lambda}$ qui optimise le critère. En effet, comme le montrent les résultats de l'estimation (cf. fig. 5.1), il n'y a pas unicité de $(\hat{\lambda}, \hat{\delta})$. On doit choisir un des couples optimaux, et l'on constate que la meilleure déconvolution est obtenue en fixant $\delta = \tilde{\delta}$.

Mais lorsque l'on doit traiter une grande série d'images satellitaires, il convient mieux de prendre comme hyperparamètres de départ ceux qui correspondent à une image *déjà traitée* qui représente le même type de terrain, et avec le même type de dégradation (ie. même H et même N).

5.1.2 Convergence

La convergence de l'algorithme d'estimation, près d'un minimum local, est gouvernée par la forme de la fonctionnelle à minimiser : selon le gradient et le hessien, celle-ci est plus ou moins proche d'une forme quadratique. Si le hessien est constant, le critère est localement quadratique. Mais, dans la pratique, ce n'est jamais le cas suivant δ.

Selon le pas α, l'algorithme peut converger ou diverger. En divisant α par le hessien on peut, comme dans la méthode de Newton, converger rapidement vers le minimum local, si le critère est localement quadratique. Mais dès que la courbure s'inverse (c'est parfois le

cas dans la pratique) ce choix du pas α peut entraîner une instabilité (inversion du sens de la descente).

Il semble que le meilleur choix du pas ne fasse pas intervenir le hessien (cf. annexe E, équation E.2), car il est trop difficile à estimer (il faut en effet calculer la covariance de l'énergie des échantillons, et un grand nombre d'échantillons est nécessaire). Un bon choix des pas est donné par $\alpha_\lambda = \alpha_0/h_{\lambda\lambda}$ et $\alpha_\delta = \alpha_0/h_{\delta\delta}$, α_0 étant choisi de manière empirique, en cherchant à accélérer la convergence tout en limitant les risques d'instabilité (il est compris entre 0 et 1) ; $h_{\lambda\lambda}$ et $h_{\delta\delta}$ sont ici les *dérivées secondes* du critère, calculées en dérivant les *gradients* (cf. annexe E, équation E.1) de façon numérique :

$$h_{\lambda\lambda} = [\nabla_\lambda(\lambda + \Delta\lambda, \delta) - \nabla_\lambda(\lambda, \delta)]/\Delta\lambda \quad \text{et} \quad h_{\delta\delta} = [\nabla_\delta(\delta + \Delta\delta) - \nabla_\delta(\lambda, \delta)]/\Delta\delta$$

Cette méthode de descente, inspirée de l'algorithme de Newton-Raphson, est particulièrement efficace lorsque le critère à optimiser est localement quadratique. Cela semble être le cas lorsque l'on est proche d'un optimum (voir paragraphe 5.2).

5.1.3 Précision de l'estimation

La précision d'estimation des hyperparamètres dépend

- du seuil d'arrêt de l'algorithme ϵ ;
- de la précision de l'estimation des espérances e_C, qui dépend du nombre d'échantillons utilisés pour calculer les moyennes, et de la taille des échantillons.

Le critère d'arrêt faisant intervenir la norme de la dérivée par rapport aux hyperparamètres, un ϵ trop grand entraîne l'arrêt alors que l'on se trouve trop « loin » du couple $(\hat{\lambda}, \hat{\delta})$ le plus proche, qui annule cette norme. Mais un ϵ trop petit peut empêcher l'arrêt à cause des fluctuations des dérivées ∇_λ et ∇_δ, alors que l'on est « près » d'un minimum local. Le choix du seuil d'arrêt se fait de manière empirique, en utilisant aussi bien des images réelles que synthétiques, avec toujours le même noyau de convolution H et le même bruit N.

Les images échantillonnées avec les lois a priori et a posteriori présentent des fluctuations, qui se répercutent sur les cumulants calculés sur ces images (comme l'énergie Φ par exemple). Si e_C^1 est l'écart-type (supposé gaussien) de l'erreur associée à la quantité $C(X)$ ($C = \Phi$ ou ses dérivées) sur l'échantillon X, les espérances étant estimées par une moyenne empirique sur n échantillons, e_C^n est l'erreur commise sur l'estimation de l'espérance de $C(X)$ on a alors $e_C^n = e_C^1/\sqrt{n}$. La quantité e_C^1 est caractéristique de l'échantillonneur, et dépend de la forme des échantillons (plus ils sont « bruités », plus les fluctuations sont grandes).

Si on considère e_c^p l'écart-type des fluctuations par pixel, alors comme on effectue une moyenne à la fois sur n échantillons et $N_x \times N_y$ pixels, l'erreur sur l'espérance de $C(X)$ est $e_C = e_C^p/\sqrt{n\,N_x\,N_y}$.

5.2 Résultats

Les résultats de la restauration (avec les hyperparamètres estimés) de l'image fournie par le CNES sont détaillés dans l'annexe A. L'estimation des hyperparamètres a été effectuée sur une imagette extraite de l'image à traiter (voir fig. A.5).

5.2.1 Multiplicité des solutions

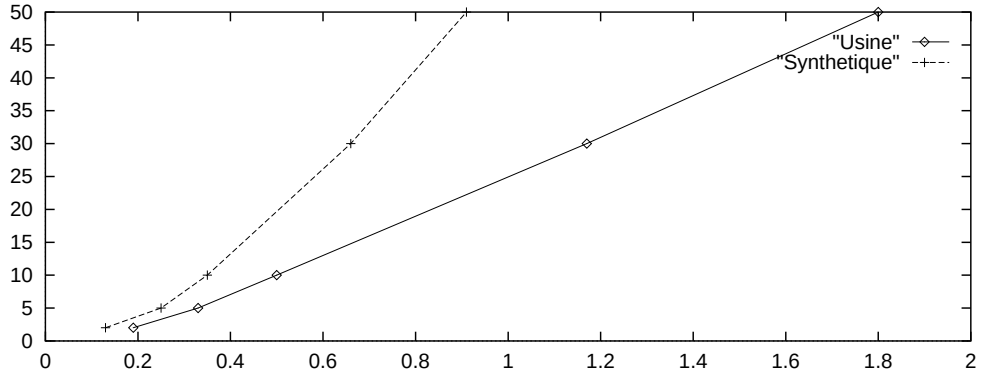


FIG. 5.1: *Ensemble des hyperparamètres (λ, δ) pour lesquels le gradient de la vraisemblance s'annule (image synthétique et usine, voir fig. A.6)*

Comme on peut voir sur la figure 5.1, il existe un ensemble d'hyperparamètres qui annulent le gradient du critère ; ce sont des minima locaux qui bloquent l'algorithme d'estimation. Ces minima semblent uniformément répartis sur une courbe, qui devient une droite lorsque δ est grand.

La présence d'une asymptote pour $\delta \to \infty$ peut s'expliquer par la forme de la fonction φ utilisée. Quelle que soit cette fonction, $\varphi(u)$ est quadratique lorsque $u \to 0$. Comme $u = g/\delta$ (g est une différence de pixels voisins), le modèle devient pratiquement quadratique lorsque $\delta \to \infty$, et l'on a alors $\lambda^2\varphi(g/\delta) \simeq \lambda^2/\delta^2\,\varphi(g)$: le modèle dépend uniquement du rapport λ/δ et c'est ce rapport-là que l'on estime, ce qui correspond à une droite $\lambda/\delta = k_\infty$.

La fonction HyperSurfaces φ_{HS} devient linéaire lorsque $u \to \infty$ (voir fig. B.1), c'est-à-dire lorsque $\delta \to 0$. On doit logiquement s'attendre à estimer encore une fois le rapport λ/δ dans ce cas, mais pour un modèle $\varphi(u) \simeq |2u|$, la courbe des minima locaux doit donc ressembler à un segment $\lambda/\delta = k_0$.

5.2.2 Critère localement quadratique

Sur la figure 5.2, où l'on a tracé les gradients autour d'un minimum local (pour $\lambda = 0.19, \delta = 2$), on constate que le critère est *localement quadratique en fonction de λ*, car sa dérivée par rapport à λ est linéaire. Cela justifie l'emploi d'une méthode de descente de type Newton sur λ.

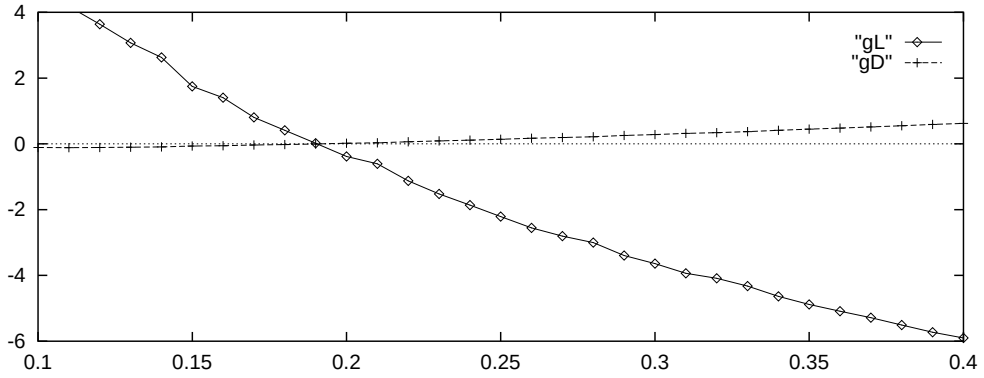


FIG. 5.2: *Gradients du critère ∇_λ et ∇_δ (gL et gD) en fonction de λ, pour $\delta = 2$ (imagette usine extraite de Nîmes, voir fig. A.5)*

Dans la pratique, on préfère ainsi effectuer l'estimation de λ à δ fixé, ce qui ne nécessite que le calcul de ∇_λ. Cette méthode converge rapidement : en partant de $\lambda = \delta/10$ avec $\delta = 1.5\sigma$, l'optimum $\hat{\lambda}$ est atteint en 4 à 8 itérations, avec un pourcentage d'erreur sur $\hat{\lambda}$ inférieur à 5%.

5.2.3 Influence de la précision de l'estimation

% d'erreur	-30%	-20%	-10%	0%	+10%	+20%	+30%
λ	0.13	0.15	0.17	**0.19**	0.21	0.23	0.25
SNR (dB)	22.13	22.51	22.64	**22.70**	22.65	22.57	22.48

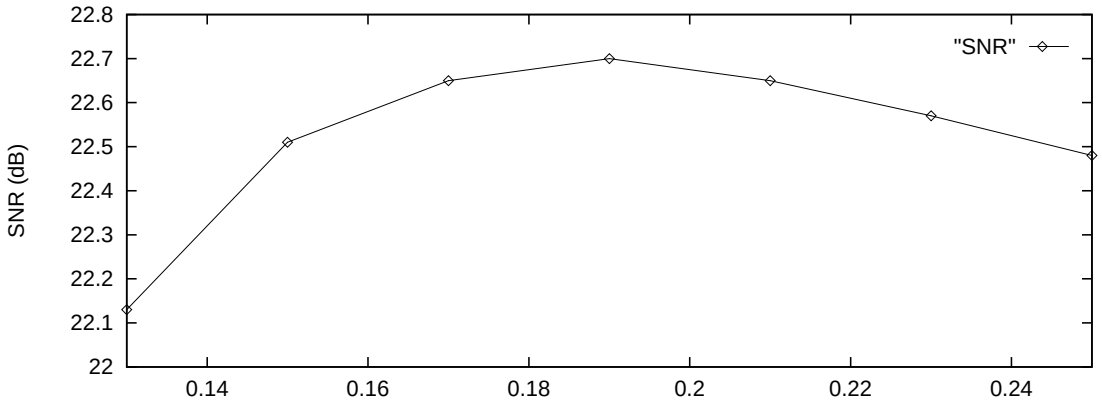


FIG. 5.3: *Rapport signal/bruit de l'image restaurée avec différents λ autour de $\hat{\lambda} = 0.19$, avec $\delta = 2$ (imagette usine extraite de Nîmes, voir fig. A.5)*

L'influence de la précision de l'estimation de λ (à δ fixé) sur la restauration est illustrée sur la figure 5.3 : on peut tolérer jusqu'à 10% d'erreur sur les hyperparamètres sans observer une dégradation notable de la qualité de l'image restaurée.

Si l'on s'intéresse à la sensibilité de la restauration au choix de δ (avec λ optimal), on constate que tous les minima locaux sont équivalents pour le rapport signal/bruit *global*, quel que soit δ. Ce rapport reste voisin de 22.7 dB pour l'image de la figure A.5. Par contre, le rapport signal/bruit *local* (calculé par exemple sur une zone homogène, ou sur une zone striée) dépend beaucoup du choix de δ. La contrainte $\delta = \tilde{\delta}$ permet d'imposer une restauration optimale des zones homogènes (avec un minimum de bruit), tout en préservant les discontinuités.

5.2.4 Influence de la zone d'extraction des imagettes

Lorsque l'on cherche à déconvoluer *rapidement* une image satellitaire de grande taille (512×512 minimum), il est indispensable d'effectuer l'estimation des hyperparamètres sur une imagette de petite taille qui en est extraite (64×64 ou 128×128 par exemple). En effet, le modèle de régularisation utilise *un seul* couple (λ, δ) sur la totalité de l'image, et ce couple devrait pouvoir convenir à n'importe quelle imagette quelle que soit la zone d'où elle est extraite.

Nous montrons sur la figure 5.4 que le résultat de l'estimation dépend des imagettes, et les meilleurs hyperparamètres pour restaurer l'image dans son ensemble sont moins adaptés au problème que ceux qui sont estimés *localement* sur des zones 64×64.

Les zones homogènes ont en effet besoin d'une régularisation bien plus poussée (jusqu'à $\lambda = 0.4$) que les zones très contrastées contenant de nombreux contours (on a alors $\lambda = 0.17$). Le choix du couple (λ, δ) convenant à toute l'image ne peut alors être fait correctement que si l'imagette choisie est représentative de l'ensemble de la scène. Pour cette raison, même si l'algorithme d'estimation en lui-même est automatique, l'extraction de l'imagette ne doit pas être effectuée arbitrairement par le logiciel. L'intervention de l'utilisateur est nécessaire pour cette opération, mais elle est aisée, et ne demande que très peu de temps (contrairement à l'estimation manuelle des hyperparamètres).

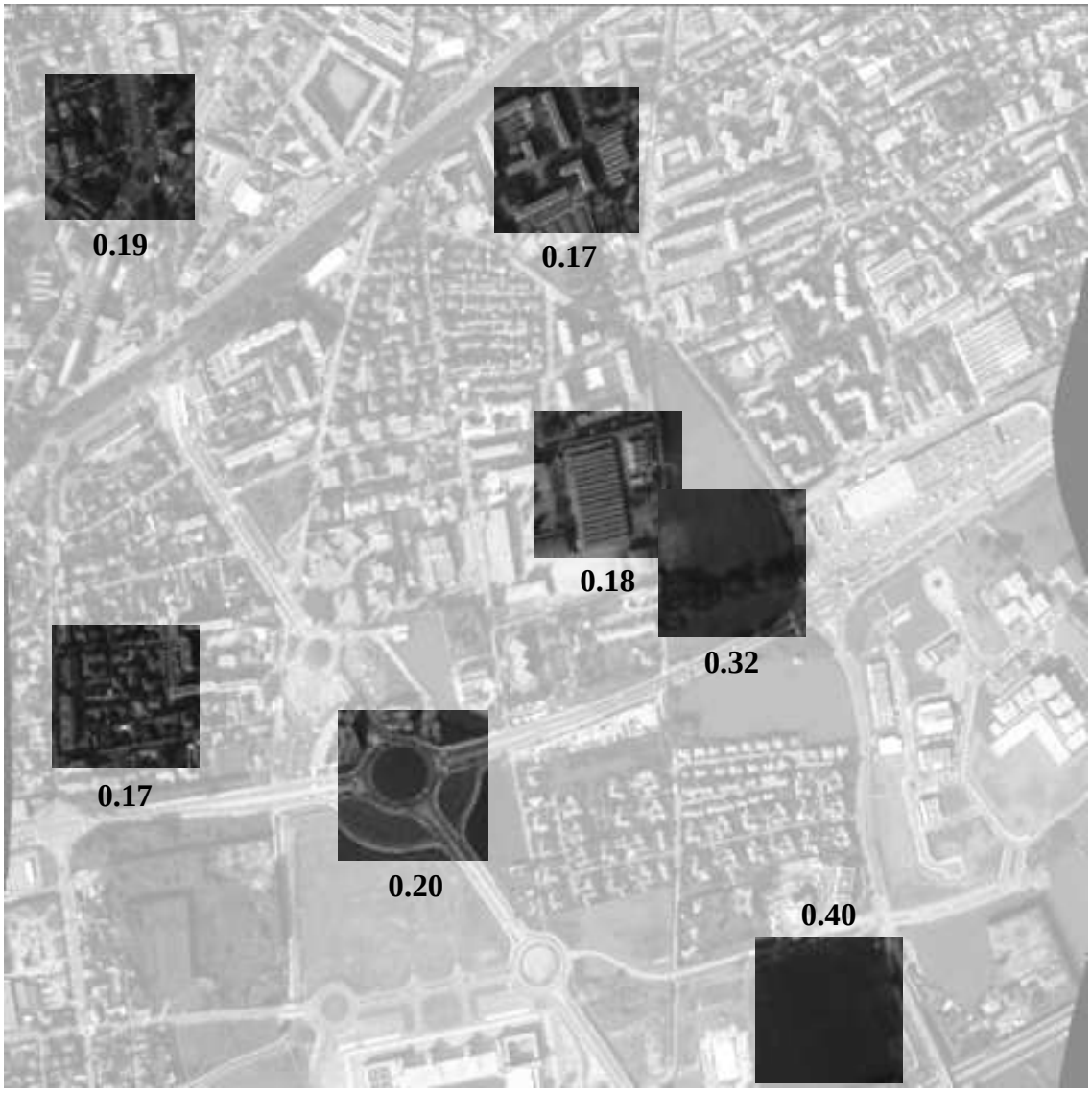


FIG. 5.4: *Influence de la zone d'extraction de l'imagette (64×64) sur l'estimation de* λ*, avec* $\delta = 2$

5.3 Comparaison avec le gradient stochastique

Il existe un autre algorithme permettant d'effectuer l'estimation d'hyperparamètres et la restauration simultanées : c'est le gradient stochastique généralisé, proposé par Younes [YOU88, YOU89].

On construit deux chaînes de Markov parallèles avec des échantillons (X_1^n) et (X_2^n), correspondant aux lois a priori ($P \propto e^{-\Phi(X)}$) et a posteriori ($P_Y(X) \propto e^{-U(X)}$), en conservant tous les échantillons, sans chercher à atteindre les distributions stationnaires. On utilise alors l'itération suivante pour passer de θ_n à θ_{n+1}, si θ est l'un des hyperparamètres :

$$\theta_{n+1} = \theta_n + \frac{1}{(n+1)A}\left(\frac{\partial \Phi}{\partial \theta}(X_1^{n+1}, \theta_n) - \frac{\partial U}{\partial \theta}(X_2^{n+1}, \theta_n)\right)$$

Le point commun avec l'estimateur du maximum de vraisemblance est l'échantillonnage avec les deux lois P_Y et P (avec et sans attache aux données) ; on doit donc s'attendre au même type de résultat (pas d'unicité des hyperparamètres optimaux).

La différence essentielle par rapport à notre méthode « MCMCML » est qu'il n'est pas nécessaire d'atteindre l'équilibre des chaînes de Markov à chaque itération, et d'effectuer des moyennes : *un seul échantillon* avec et sans attache aux données suffit pour calculer l'incrément de θ. Le coefficient A (similaire à α dans notre cas) doit être déterminé de manière à ce que la méthode puisse converger.

Une version de cet algorithme a été utilisée dans [ZER98] et [KHO98], où l'itération permettant de passer de θ_n à θ_{n+1} est :

$$\theta_{n+1} = \theta_n + \frac{1}{(n+1)A}\left(\frac{\partial \Phi}{\partial \theta}(X_1^{n+1}, \theta_n) - \frac{\partial U}{\partial \theta}(\hat{X}, \theta_n)\right)$$

les échantillons a posteriori (X_2^n) du gradient stochastique généralisé sont remplacés par l'image $\hat{X}$ restaurée avec les hyperparamètres θ_n, ce qui permet d'éviter d'échantillonner avec la loi a posteriori. Cela revient à appliquer la méthode de Lakshmanan et Derin en données complètes (voir paragraphe 2.1.1).

Chapitre 6

Conclusion et perspectives

6.1 A priori sur les hyperparamètres

Il est nécessaire d'introduire une *connaissance a priori* sur les hyperparamètres si l'on veut trouver une solution unique (minimum global de la -log vraisemblance) lors de leur estimation. On peut construire une loi de probabilité $P(\lambda, \delta)$ qui tient compte du fait que δ ne peut être trop grand (on se ramènerait alors à un φ-modèle quadratique) ni trop petit, dans les deux cas, on perdrait la forme non-linéaire et non-quadratique du modèle, ce qui en fait toute la puissance. D'autre part, on peut interdire à λ d'être trop faible (régularisation insuffisante) ou trop élevé (sur-régularisation, donc perte des détails).

Un choix possible de $P(\lambda, \delta)$ serait basé sur une grande série d'hyperparamètres estimés sur un grand nombre d'images, et la probabilité traduirait alors la fréquence d'apparition de certains couples (λ, δ).

6.2 Une meilleure qualité de la restauration

On obtiendrait probablement de meilleures images restaurées en prenant en compte les *dérivées d'ordre supérieur*, lorsque l'on calcule les gradients sur les images (différences entre pixels voisins). Geman et Yang [GEM95] utilisent un modèle de régularisation au second ordre, ce qui permet de tenir compte d'un ordre 2 dans le voisinage, et de restaurer les images de manière plus efficace.

L'introduction de dérivées d'ordre supérieur entraîne l'augmentation du nombre d'hyperparamètres ; au second ordre on doit introduire au minimum un λ_2 et un δ_2 supplémentaire, qui interviendront sur la somme des φ-fonctions appliquées aux gradients d'ordre 2. De ce fait, l'estimation est plus difficile, car le nombre de degrés de liberté augmente (on a alors au minimum 4 hyperparamètres à estimer). Cependant, les méthodes de descente restent applicables, dans la mesure où il est toujours possible d'estimer les dérivées du critère par rapport aux différents hyperparamètres, par des espérances calculées sur des échantillons.

6.3 Vers un modèle plus local

Les hyperparamètres peuvent être estimés sur une imagette, extraite d'une grande image, et ils traduisent alors la meilleure façon de restaurer cette imagette. A l'intérieur d'une même grande image, il n'est pas rare d'obtenir des hyperparamètres optimaux différents, selon l'endroit d'où l'on extrait la sous-image. Ceci montre que l'approche par un φ-modèle est trop *globale*, car on cherche à restaurer la grande image avec un *unique* couple optimal $(\hat{\lambda}, \hat{\delta})$, alors que *localement*, sur de petites sous-images, d'autres couples conviennent mieux. Il serait alors judicieux de diviser l'image en morceaux pour éviter, par exemple, que la présence de forts gradients (villes) influence la restauration des zones homogènes (champs, étendues désertes).

Un modèle plus local serait certainement plus adapté à la restauration : si λ et δ dépendaient de chaque site, ils traduiraient mieux la présence de détails ou de zones homogènes. Cela permettrait de traiter les grandes images de façon optimale et en une seule passe, alors que pour un couple $(\hat{\lambda}, \hat{\delta})$ global, il est préférable de les diviser en zones de forme particulière, et d'estimer les hyperparamètres puis de restaurer zone par zone, pour obtenir les meilleurs résultats.

Bibliographie

[AZE88] **R. Azencott** : *Image analysis and Markov fields*, in Int. Conf. on Ind. and Appl. Math. SIAM Paris, SIAM Philadelphia, 1988.

[BES74] **J. Besag** : *Spatial Interaction and Statistical Analysis of Lattice Systems*, A. Roy. Stat. Soc. Series B, Vol 36, pp 721-741, 1974.

[BRO97] **S. P. Brooks, G. O. Roberts** : *Assessing Convergence of Markov Chain Monte Carlo Algorithms*, Univ. of Cambridge, may 1997.

[CHA94] **P. Charbonnier** : *Reconstruction d'image : régularisation avec prise en compte des discontinuités*, Thèse de Doctorat, Univ. Nice-Sophia Antipolis, Sept. 1994.

[CHA97] **P. Charbonnier, L. Blanc-Féraud, G. Aubert, M. Barlaud** : *Deterministic edge-preserving regularization in computed imaging*, IEEE Trans. Image Proc., Vol 6, No 2, pp 298-311, Feb. 1997.

[CHL88] **B. Chalmond** : *Image Restoration using an Estimated Markov Model*, Signal Processing, 15, pp 115-129, 1988.

[CHM96] **F. Champagnat, Y. Goussard, J. Idier** : *Unsupervised deconvolution of sparse spike train using stochastic approximation*, IEEE Trans. on Image Processing, Vol 44, No 12, Dec. 1996.

[CHN96] **K. Chan, A. Gray** : *Robustness of automated data choices of smoothing parameter in image regularization*, Statistics and Computing, Vol 6, Chapman & Hall, 1996.

[DEM89] **G. Demoment**, *Image Reconstruction and Restoration : Overview of Common Estimation Structures and Problems*, IEEE Trans. on ASSP, Vol 37, No 12, pp 2024-2036, Dec. 1989.

[DES96] **X. Descombes, R. Morris, J. Zerubia, M. Berthod** : *Estimation of Markov random field prior parameters using Markov Chain Monte Carlo Maximum Likelihood*, Rapport de Recherche INRIA No 3015, Oct. 1996, et IEEE Trans. on Image Proc. à paraître en 1998.

[GAU97] **I. Gaudron** : *Rate of convergence of the Swendsen-Wang dynamics in image segmentation problems : a theoretical and experimental study*, ESAIM : Probability and Statistics, Soc. de Math. Appli. et Ind., Vol 1, pp 259-284, 1997.

[GEM84] **S. Geman, D. Geman** : *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Trans. Patt. Anal. Machine intell., Vol 6, No 6, pp 721-741, Nov. 1984.

[GEM90] **D. Geman** : *Random fields and inverse problems in imaging*, Springer-Verlag, Berlin, 1990.

[GEM92] **D. Geman, G. Reynolds** : *Constrained restoration and recovery of discontinuities*, IEEE Trans. Patt. Anal. Machine intell., Vol 16, No 3, pp 367-383, Mar. 1992.

[GEM95] **D. Geman, C. Yang** : *Nonlinear image recovery with half-quadratic regularization and FFTs*, IEEE Trans. Image Proc., Vol 4, No 7, pp 932-946, Jul. 1995.

[GEO88] **H.-O. Georgii** : *Gibbs Measures and Phase Transitions*, Gruyter - Studies in Mathematics, Vol 9, 1988.

[GEY92] **C. Geyer, E. A. Thompson** : *Constrained Monte Carlo Maximum Likelihood for dependent data*, J. R. Statist. Soc. B, Vol 54, No 3, pp 657-699, 1992.

[GEY93] **C. J. Geyer** : *Markov Chain Monte Carlo Maximum Likelihood*, School of Statistics Univ. of Minnesota Minneapolis, MN 55455, 1993.

[GEY94] **C. Geyer** : *On the convergence of Monte Carlo Maximum Likelihood calculations*, J. R. Statist. Soc. B Vol 56, No 1, pp 261-274, Nov. 1994.

[GIL96] **W. R. Gilks, S. Richardson, D. J. Spiegelhalter** : *Markov Chain Monte Carlo in practice*, Chapman & Hall, 1996.

[GRE95] **P. E. Greenwood, I. W. McKeague, W. Wefelmeyer** : *Outperforming the Gibbs Sampler Empirical Estimator of Nearest Neighbour Random Fields*, Univ. of British Columbia, Canada, 1995.

[HAD23] **J. Hadamard** : *Lectures on Cauchy's Problem in Linear Partial Differential Equations*, Yale University Press, New Haven, 1923.

[HAS70] **W. Hastings** : *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, No 57, pp 97-109, 1970.

[HIG97] **D. M. Higdon** : *Auxiliary variable methods for Markov Chain Monte Carlo with applications*, submitted to JASA Theory and Methods, revised version, Sept. 1997.

[HUR97] **M. Hurn** : *Difficulties in the use of auxiliary variables in Markov Chain Monte Carlo methods*, Statistics and Computing, Vol 7, pp 35-44, 1997.

[KHO98] **M. Khoumri, L. Blanc-Féraud, J. Zerubia** : *Unsupervised deconvolution of satellite images*, IEEE Int. Conference on Image Processing, Chicago, USA, 4-7 oct. 1998.

[LAA87] **P. J. M. van Laarhoven, E. H. L. Aarts** : *Simulated Annealing : theory and applications*, D. Reidel, 1987.

[LAK89] **S. Lakshmanan, H. Derin** : *Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing*, IEEE Trans. Patt. Anal. Machine intell., Vol 11, No 8, pp 799-813, Aug. 1989.

[LIU96] **J. S. Liu** : *Metropolized Gibbs sampler : an improvement*, Department of Statistics, Stanford University, 1996.

[MEN94] **K. L. Mengersen, R. L. Tweedie** : *Rates of convergence of the Hastings and Metropolis algorithms*, Queensland Univ. of Tech. and Colorado State Univ., 1994.

[MET53] **N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller** : *Equation of state calculations by fast computing machines*, J. of Chem. Physics, Vol 21, pp 1087-1092, 1953.

[MOR96a] **R. Morris, X. Descombes, J. Zerubia** : *An Analysis of Some Models Used in Image Segmentation*, Rapport de Recherche INRIA No 3016, Oct. 1996.

[MOR96b] **R. Morris, X. Descombes, J. Zerubia** : *Fully Bayesian image segmentation - an engineering perspective*, Rapport de Recherche INRIA No 3017, Oct. 1996.

[Num93] **W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery** : *Numerical Recipes in C : The Art of Scientific Computing*, 2nd Edition, Cambridge University Press, 1993.

[ORU96] **J. J. K. O Ruanaidh, W. J. Fitzgerald** : *Numerical Bayesian methods applied to signal processing*, Statistics and Computing, Springer-Verlag, 1996.

[ROB96] **C. Robert** : *Méthodes de Monte Carlo par chaînes de Markov*, Economica, Paris, 1996.

[SIG97] **M. Sigelle** : *Simultaneous image restoration and hyperparameter estimation for incomplete data by a cumulant analysis*, Rapport de Recherche INRIA No 3249, Sept. 1997.

[SWE87] **R. H. Swendsen, J.-S. Wang** : *Nonuniversal critical dynamics in Monte Carlo simulations*, Physical Review Letters, Vol 58, pp 86-88, 1987.

[TAN96] **M. A. Tanner** : *Tools for statistical inference*, Springer Series in Statistics, Springer-Verlag, 1996.

[TIK63] **A. N. Tikhonov** : *Regularization of incorrectly posed problems*, Sov. Math. Dokl., Vol 4, pp 1624-1627, 1963.

[YOU88] **L. Younes** : *Estimation and annealing for Gibbsian fields*, Ann. Inst. Poincaré, Vol 24, No 2, pp 269-294, 1988.

[YOU89] **L. Younes** : *Parametric inference for imperfectly observed Gibbsian fields*, Prob. Th. Fields, No 82, pp 625-645, Springer-Verlag, 1989.

[YOU96] **L. Younes** : *Synchronous random fields and image restoration*, rapport CMLA, Ecole Normale Supérieure de Cachan, 1996.

[ZER98] **J. Zerubia, L. Blanc-Féraud** : *Hyperparameter estimation of a variational model using a stochastic gradient method*, Bayesian Inference for Inverse Problems, part of SPIE's Int. Symposium on Optical Science, Engineering and Instrumentation, Vol 3459, San Diego, USA, 19-24 jul. 1998.

Annexe A

Résultats

Le noyau de convolution fourni par le CNES est de taille 11×11, il est séparable selon les lignes et les colonnes, et symétrique. On note ici $\mathcal{X}$ l'image originale et Y l'image dégradée.

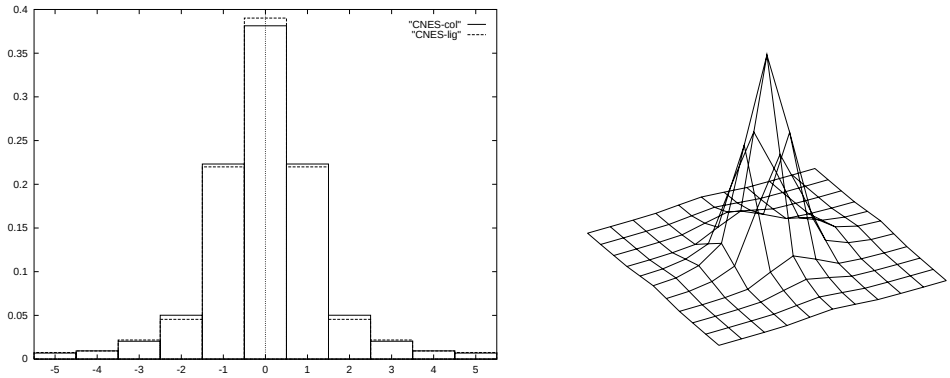


FIG. A.1: *Noyau de convolution selon lignes et colonnes, et représentation 3D*

L'écart-type du bruit est calculé de la façon suivante : $\sigma = \sqrt{A + Bn + Cn^2}$ avec $A = 1.299$, $B = 0.00676$, $C = 1.89\,10^{-5}$, n étant la valeur du pixel considéré (entre 0 et 255). Pour l'image fournie on trouve $\sigma \simeq 1.35$, et on considère que le bruit est gaussien stationnaire (dépend peu de n).

Cela correspond à un rapport signal/bruit de 16.1 dB.

Le rapport signal/bruit (en dB) est donné par l'expression :

$$\text{SNR} = 10 \log \left[\frac{\sum (\mathcal{X}_{ij})^2}{\sum (Y_{ij} - \mathcal{X}_{ij})^2} \right]$$

FIG. A.2: *Nîmes, image dégradée Y (H : voir fig. A.1 et bruit σ, SNR $= 16.1$ dB),* 512×512, *256 niveaux de gris - ©CNES*

FIG. A.3: *Nîmes, image restaurée avec* $\lambda = 0.49, \delta = 10$ *(SNR* $= 21.54$ *dB)*

FIG. A.4: *Nîmes, image originale* $\mathcal{X}$, 512×512, *256 niveaux de gris - ©CNES*

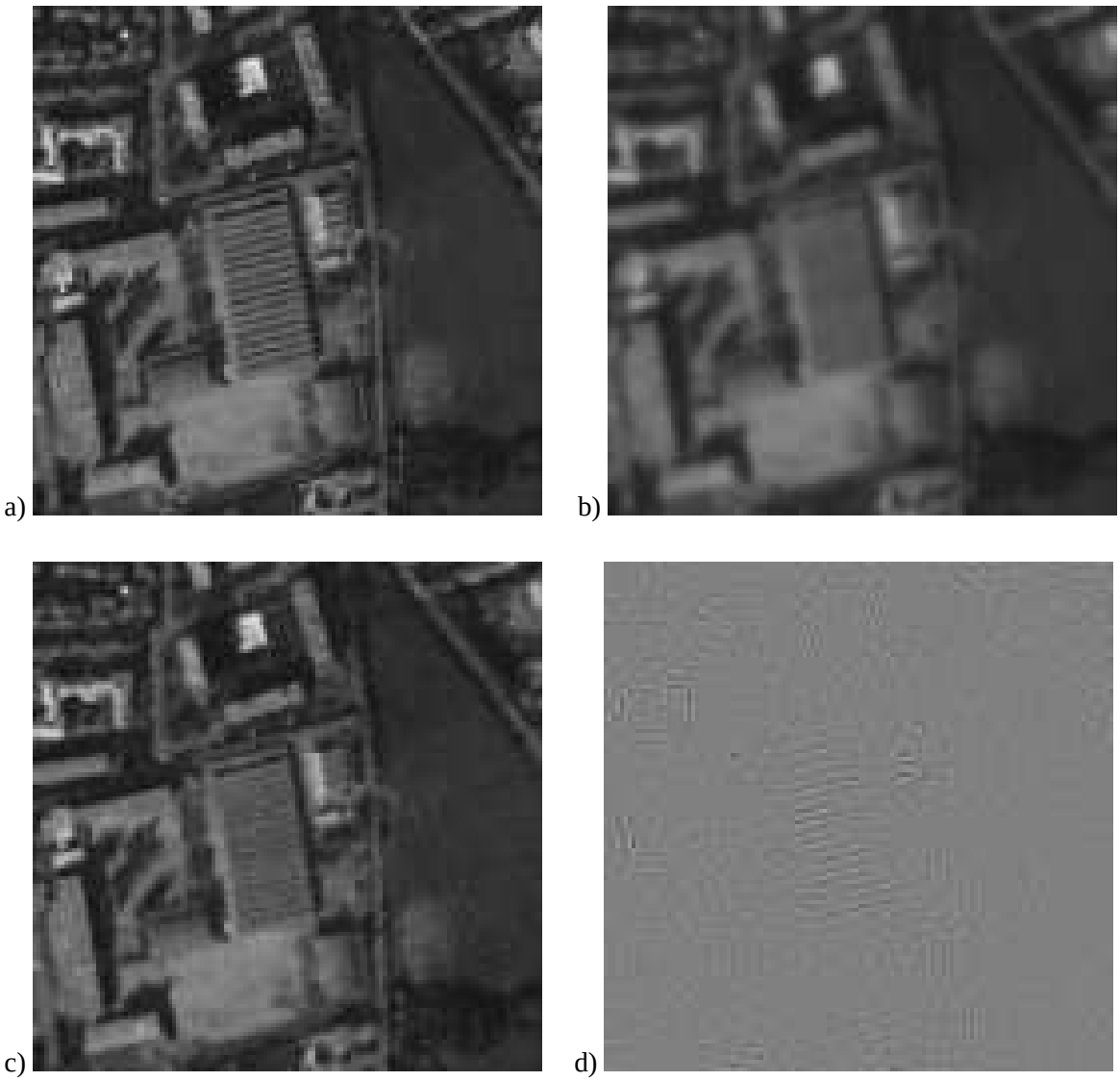


FIG. A.5: *Image* 128×128 *extraite de Nîmes ; a) originale, b) dégradée (SNR=* 17.6 *dB), c) restaurée avec* $\lambda = 0.49, \delta = 10$*, d) erreur (SNR=* 22.82 *dB)*

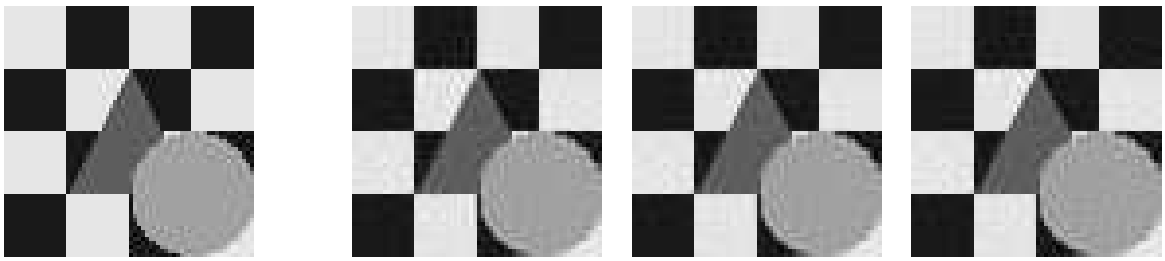


image originale | λ=0.13, δ=2 SNR=27.5 dB | λ=0.25, δ=5 SNR=28.9 dB | λ=0.35, δ=10 SNR=28.4 dB

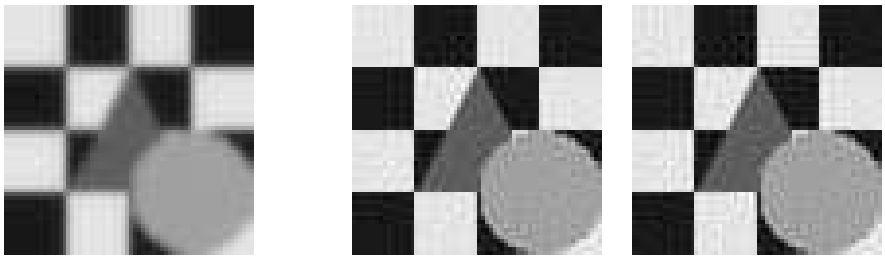


image dégradée SNR=15.9 dB | λ=0.66, δ=30 SNR=26.3 dB | λ=0.91, δ=50 SNR=25.0 dB

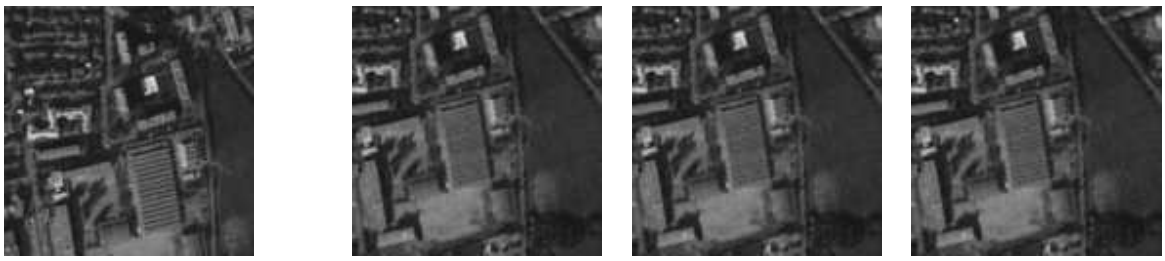


image originale | λ=0.19, δ=2 SNR=22.70 dB | λ=0.33, δ=5 SNR=22.72 dB | λ=0.49, δ=10 SNR=22.82 dB

image dégradée SNR=16.7 dB | λ=1.2, δ=30 SNR=22.79 dB | λ=1.8, δ=50 SNR=22.73 dB

FIG. A.6: *Image synthétique et extrait de Nîmes (annexe A) : comparaison des résultats de la restauration avec une série d'hyperparamètres optimaux (fonction φ_{HS})*

Annexe B

Comportement des fonctions φ

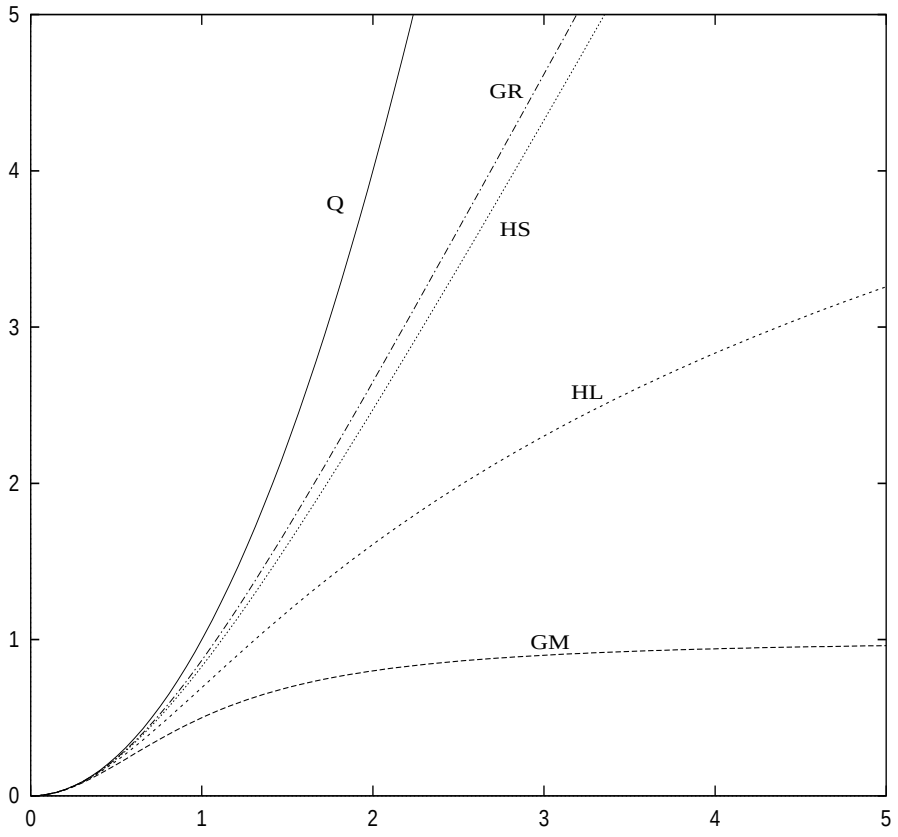


FIG. B.1: *Fonctions de potentiel* $\varphi_{GM}, \varphi_{HS}, \varphi_{HL}, \varphi_{GR}$*, comparées à la fonction quadratique* $\varphi_Q(u) = u^2$

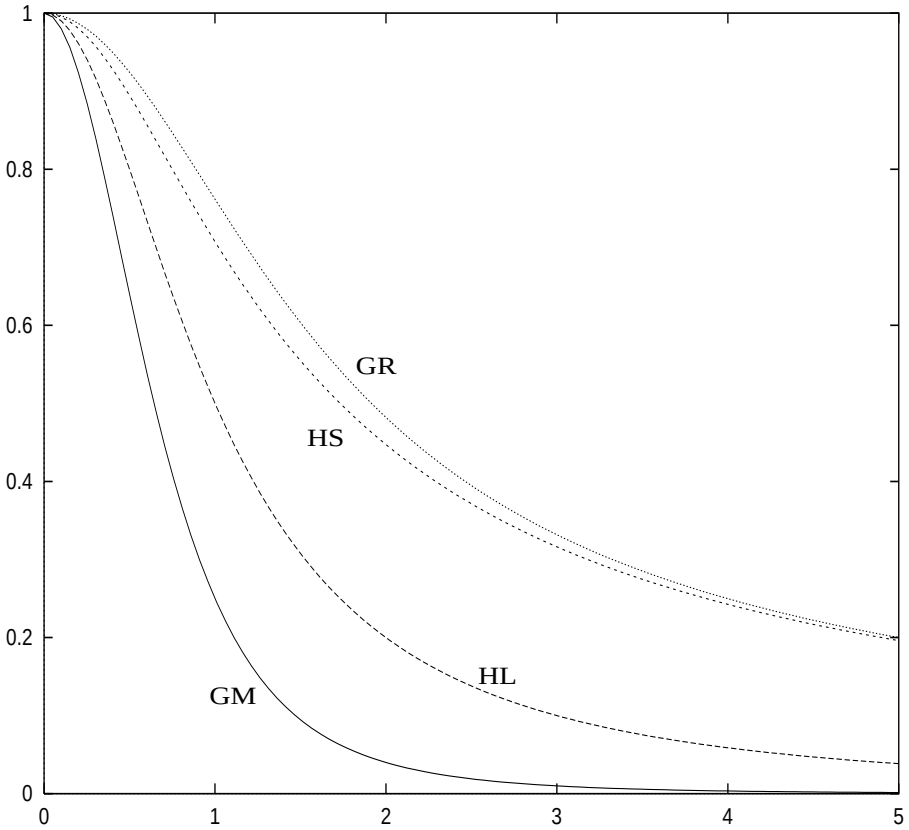


FIG. B.2: *Fonctions de pondération* $\varphi'(u)/2u$ *(GM, HS, HL, GR)*

Annexe C

Théorème semi-quadratique

THÉORÈME C.0.1 (SEMI-QUADRATIQUE)
Soit une fonction $\varphi : [0, +\infty[\to [0, +\infty[$ *de classe* $\mathcal{C}^1$ *telle que* $f(u) = u^2 - \varphi(u)$ *soit strictement convexe sur* $]0, +\infty[$.

Alors :

- *Il existe une fonction* $\psi : [0, +\infty[\to [0, +\infty[$ *telle que :*

$$\varphi(u) = \inf_{b \in \mathbb{R}} \left[(b-u)^2 + \psi(b) \right]$$

- *Si* $\lim_{u \to 0^+} \frac{\varphi'(u)}{2u}$ *existe, alors pour tout* $u \geq 0$ *fixé, la valeur de* b_u *pour laquelle l'inf est atteint :*

$$\inf_{b \in \mathbb{R}} \left[(b-u)^2 + \psi(b) \right] = (b_u - u)^2 + \psi(b_u)$$

  est unique et donnée par $b_u = u - \frac{1}{2}\varphi'(u)$.

La démonstration de ce théorème est basée sur les paires de Legendre ; elle est détaillée dans [CHA94].

Les conditions d'application de ce théorème sont vérifiées, puisque nous utilisons des fonctions φ telles que

$$i) \quad \lim_{u \to 0^+} \frac{\varphi'(u)}{2u} = 1 \quad \text{et} \quad ii) \quad \frac{\varphi'(u)}{2u} \text{ continue et strictement décroissante sur } [0, +\infty[$$

$$iii) \lim_{u \to \infty} \frac{\varphi'(u)}{2u} = 0.$$

Si $ii)$ est vérifiée, alors $\frac{\varphi''(u)}{2} < \frac{\varphi'(u)}{2u}$. Si de plus $i)$ est vérifiée, alors on a $\frac{\varphi''(u)}{2} < \frac{\varphi'(u)}{2u} \leq 1$. Ce qui entraîne que $f''(u) = 1 - \frac{\varphi''(u)}{2} > 0$, $f(u)$ est alors strictement convexe sur $[0, +\infty[$.

Soit b la valeur associée à u pour laquelle l'inf est atteint. On calcule les variations de u et $b : u \to u + \delta u$ et $b \to b + \delta b$. Si on appelle g la fonction $g(b, u) = (b - u)^2$:

$$\varphi(u + \delta u) = g(u + \delta u, b + \delta b) + \psi(b + \delta b) = \varphi(u) + \delta b \left(\frac{\partial g}{\partial b} + \frac{\partial \psi}{\partial b} \right) + \frac{\partial g}{\partial u} \delta u$$

Or $\left(\frac{\partial g}{\partial b} + \frac{\partial \psi}{\partial b} \right) = 0$ car b réalise l'inf. On obtient finalement $\frac{d\varphi}{du} = \frac{\partial g}{\partial u}$, ce qui donne l'expression de b_u qui minimise $(b - u)^2 + \psi(b)$.

Annexe D

Diagonalisation de H et DCT

Matrices circulantes par blocs

Une matrice M ($N \times N$) est dite circulante par blocs si elle est générée par une matrice m ($I \times J$), avec $N = IJ$:

$$M = \begin{pmatrix} C(0) & C(I-1) & \dots & C(1) \\ C(1) & C(0) & \dots & C(2) \\ \vdots & \vdots & \vdots & \vdots \\ C(I-1) & C(I-2) & \dots & C(0) \end{pmatrix}$$

où $C(i)$ est la matrice circulante ($J \times J$) dont la première ligne est $(m_{i,0}, m_{i,J-1}, \dots m_{i,1})$, la ligne j est obtenue par décalage cyclique à droite de la ligne précédente.

La *transformée de Fourier* discrète bidimensionnelle permet de *diagonaliser* les matrices circulantes par blocs. La matrice $\mathcal{W}$ ($N \times N$), constituée de I^2 blocs de dimension $J \times J$, est définie par
$\mathcal{W}_{x,y}(k,l) = N^{-1/2} \exp(2\pi i(xk/I + yl/J))$ (pour le bloc x, y)
et on a $\mathcal{F} = \mathcal{W}^{-1} = \mathcal{W}^*$ où l'opérateur $\mathcal{F}$ désigne la transformée de Fourier bidimensionnelle. On peut alors écrire $M = \mathcal{W}\mathcal{G}\mathcal{W}^{-1}$, où $\mathcal{G}$ est une matrice diagonale, dont les éléments diagonaux sont donnés par $\sqrt{N}\,\mathcal{W}^{-1}m$.

Cette propriété permet de diagonaliser l'opérateur de convolution H, circulant par blocs si on considère que les conditions aux bords de l'image sont *périodiques*. H est généré par h et les éléments de la matrice diagonale sont $\sqrt{N}\,\mathcal{W}^{-1}h$. La convolution (multiplication par H) est alors remplacée par une multiplication par les éléments diagonaux ($I \times J$ éléments au lieu de $N \times N$).

Ceci justifie l'algorithme 3.2.1 proposé par Geman & Yang dans [GEM95] pour le tirage de X à B fixé : la loi est gaussienne, et la matrice de covariance Σ est diagonalisée de cette manière car elle est construite avec des matrices circulantes par blocs (H, et les gradients sur les colonnes et les lignes D_x D_y).

Utilisation d'une transformée en cosinus

La propriété ci-dessus n'est valable que pour des matrices circulantes, qui utilisent un décalage cyclique, équivalent à des conditions aux bords *périodiques* (image repliée sur un tore). Mais nos conditions aux bords sont *symétriques*. On peut se ramener au cas périodique en symétrisant les images que l'on traite, suivant les deux axes (images de taille 4 fois plus grande). Si l'on périodise ces images, la symétrie sur les bords est respectée. On applique alors la propriété de diagonalisation en utilisant une FFT, sur les images ainsi symétrisées. Or la FFT d'une image symétrique est donnée de façon plus efficace par la transformée en cosinus (on évite ainsi de faire les calculs correspondant aux images 4 fois plus grandes).

En voici la démonstration, pour des vecteurs de dimension N pixels :

$$\mathrm{DCT}[f]_k = \sum_{j=0}^{N-1} f_j \cos\left(\frac{\pi k(j+\frac{1}{2})}{N}\right)$$

$$\mathrm{DCT}^{-1}[f]_k = \frac{f_0}{N} + \frac{2}{N}\sum_{k=1}^{N-1} f_k \cos\left(\frac{\pi k(j+\frac{1}{2})}{N}\right)$$

donnent les transformées en cosinus directe (DCT) et inverse (DCT^{-1}) de f. On montre qu'à un déphasage près, la DCT est la transformée de Fourier (FFT) du même signal, symétrisé, de dimension $2N$:

$$\mathrm{DCT}_N[f]_k = e^{-ik\pi/2N}\mathrm{FFT}_{2N}[f]_k.$$

On en déduit la propriété suivante permettant de calculer $f = g * h^2$ sur des vecteurs symétriques de dimension $2N$:

$$\mathrm{DCT}_N[f] = \mathrm{DCT}_N[g].\mathrm{FFT}_{2N}[h^2]$$

où h^2 est l'extension de h à la dimension $2N$, et DCT_N désigne la transformée en cosinus appliquée à la moitié du vecteur, de dimension N.

Une condition essentielle pour que cette propriété soit valable est la symétrie de h^2, sans laquelle le produit de convolution ne peut être symétrique.

La démonstration reste la même pour les transformées bidimensionnelles. L'extension h^4 du noyau de convolution h est supposée symétrique (c'est le cas du noyau fourni par le CNES). Il en est de même pour les générateurs des opérateurs $D_x^{4\,t}D_x^4$ et $D_y^{4\,t}D_y^4$, ce qui permet d'écrire l'itération de l'algorithme 3.2.1 à l'aide de transformées en cosinus, en utilisant la propriété ci-dessus.

Annexe E

Calcul du gradient et du hessien

La vraisemblance de Y (image observée) $P(Y \mid \lambda, \delta)$ est donnée par la formule de Bayes :

$$P(Y \mid \lambda, \delta) = \sum_{X \in \Omega} P(Y, X \mid \lambda, \delta) = \sum_{X \in \Omega} P(Y \mid X, \lambda, \delta) P(X \mid \lambda, \delta)$$

$$P(Y \mid \lambda, \delta) = \sum_{X \in \Omega} \frac{1}{K_\sigma} e^{-||Y - HX||^2/2\sigma^2} \frac{1}{Z} e^{-\Phi(X,\lambda,\delta)} = \frac{Z_Y}{Z\, K_\sigma}$$

avec $Z_Y = \sum_{X \in \Omega} e^{-||Y-HX||^2/2\sigma^2 - \Phi(X,\lambda,\delta)}$, $Z = \sum_{X \in \Omega} e^{-\Phi(X,\lambda,\delta)}$ et $K_\sigma = \sum_{X \in \Omega} e^{-||Y-HX||^2/2\sigma^2}$.

On s'intéresse uniquement aux dérivées par rapport à λ et δ de $-\log P(Y \mid \lambda, \delta)$. Si θ est un de ces deux hyperparamètres, alors :

$$\frac{\partial}{\partial \theta} - \log P(Y \,/\, X, \lambda, \delta) = \frac{\partial \log Z_Y}{\partial \theta} - \frac{\partial \log Z}{\partial \theta} - \frac{\partial \log K_\sigma}{\partial \theta}$$

Comme K_σ ne dépend pas des hyperparamètres, on a

$$\frac{\partial}{\partial \theta} - \log P(Y \,/\, X, \lambda, \delta) = \frac{1}{Z_Y} \frac{\partial}{\partial \theta} \sum_{X \in \Omega} e^{-||Y-HX||^2/2\sigma^2 - \Phi(X,\lambda,\delta)} - \frac{1}{Z} \frac{\partial}{\partial \theta} \sum_{X \in \Omega} e^{-\Phi(X,\lambda,\delta)}$$

$$\text{avec} \quad \Phi(X, \lambda, \delta) = \lambda^2 \sum_{i,j} \left[\varphi(\frac{X_{i+1,j} - Xi,j}{\delta}) + \varphi(\frac{X_{i,j+1} - Xi,j}{\delta}) \right]$$

$$\begin{aligned} \frac{\partial}{\partial \theta} - \log P(Y \,/\, X, \lambda, \delta) &= \sum_{X \in \Omega} \left(-\frac{\partial \Phi}{\partial \theta} \right) \frac{1}{Z_Y} e^{-||Y-HX||^2/2\sigma^2 - \Phi} - \sum_{X \in \Omega} \left(-\frac{\partial \Phi}{\partial \theta} \right) \frac{1}{Z} e^{-\Phi} \\ &= E\left[\frac{\partial \Phi}{\partial \theta} \right] - E_Y \left[\frac{\partial \Phi}{\partial \theta} \right] \end{aligned}$$

où $E_Y[\,]$ et $E[\,]$ désignent respectivement les espérances calculées sur des échantillons X avec et sans attache aux données.

Pour les dérivées secondes, on obtient les résultats suivants :

$$\frac{\partial^2}{\partial\theta_1\partial\theta_2}-\log P(Y\,/\,X,\lambda,\delta)=\frac{\partial}{\partial\theta_1}\left[\sum_{X\in\Omega}\left(-\frac{\partial\Phi}{\partial\theta_2}\right)\frac{1}{Z_Y}e^{-||Y-HX||^2/2\sigma^2-\Phi}-\sum_{X\in\Omega}\left(-\frac{\partial\Phi}{\partial\theta_2}\right)\frac{1}{Z}e^{-\Phi}\right]$$

Pour le premier terme on obtient :

$$\sum_{X\in\Omega}\left[-\frac{\partial^2\Phi}{\partial\theta_1\partial\theta_2}-\frac{\partial\Phi}{\partial\theta_2}\left(-\frac{1}{Z_Y}\frac{\partial Z_Y}{\partial\theta_1}\right)+\frac{\partial\Phi}{\partial\theta_1}\frac{\partial\Phi}{\partial\theta_2}\right]\left(\frac{1}{Z_Y}e^{-||Y-HX||^2/2\sigma^2-\Phi}\right)$$

comme $\frac{1}{Z_Y}\frac{\partial Z_Y}{\partial\theta_1}=-E_Y\left[\frac{\partial\Phi}{\partial\theta_1}\right]$ alors on obtient finalement :

$$\frac{\partial^2}{\partial\theta_1\partial\theta_2}-\log P(Y\,/\,X,\lambda,\delta)=-E_Y\left[\frac{\partial^2\Phi}{\partial\theta_1\partial\theta_2}\right]-E_Y\left[\frac{\partial\Phi}{\partial\theta_1}\right]E_Y\left[\frac{\partial\Phi}{\partial\theta_2}\right]+E_Y\left[\frac{\partial\Phi}{\partial\theta_1}\frac{\partial\Phi}{\partial\theta_2}\right]$$

Si l'on désigne les dérivées premières par $\Phi_\theta\ \varphi_\theta$ et les dérivées secondes par $\Phi_{\theta1\theta2}\ \varphi_{\theta1\theta2}$:

$$\textbf{gradient}=\begin{pmatrix}E\left[\Phi_\lambda\right]-E_Y\left[\Phi_\lambda\right]\\ E\left[\Phi_\delta\right]-E_Y\left[\Phi_\delta\right]\end{pmatrix}=\begin{pmatrix}2\lambda\left(E[\sum\varphi]-E_Y[\sum\varphi]\right)\\ \lambda^2\left(E[\sum\varphi_\delta]-E_Y[\sum\varphi_\delta]\right)\end{pmatrix}\qquad\text{(E.1)}$$

$$\textbf{hessien}=\begin{pmatrix}A & B\\ B & C\end{pmatrix}$$

avec :

$$A=-E_Y[\Phi_{\lambda\lambda}]-E_Y[\Phi_\lambda]^2+E_Y[\Phi_\lambda^2]+E[\Phi_{\lambda\lambda}]+E[\Phi_\lambda]^2-E[\Phi_\lambda^2]$$
$$B=-E_Y[\Phi_{\lambda\delta}]-E_Y[\Phi_\lambda]E_Y[\Phi_\delta]+E_Y[\Phi_\lambda\Phi_\delta]+E[\Phi_{\lambda\delta}]+E[\Phi_\lambda]E[\Phi_\delta]-E[\Phi_\lambda\Phi_\delta]$$
$$C=-E_Y[\Phi_{\delta\delta}]-E_Y[\Phi_\delta]^2+E_Y[\Phi_\delta^2]+E[\Phi_{\delta\delta}]+E[\Phi_\delta]^2-E[\Phi_\delta^2]$$

Si on exprime A, B, C à l'aide de φ :

$$\begin{aligned}A=&\ -2(E_Y[\sum\varphi]-E[\sum\varphi])+4\lambda^2(-E_Y[\sum\varphi]^2+E[\sum\varphi]^2+E_Y[(\sum\varphi)^2]-E[(\sum\varphi)^2])\\ B=&\ 2\lambda^3(-E_Y[\sum\varphi]E_Y[\sum\varphi_\delta]+E[\sum\varphi]E[\sum\varphi_\delta]+E_Y[(\sum\varphi)(\sum\varphi_\delta)]-E[(\sum\varphi)(\sum\varphi_\delta)])\\ &-2\lambda(E_Y[\sum\varphi_\delta]-E[\sum\varphi_\delta])\\ C=&\ -\lambda^2(E_Y[\sum\varphi_{\delta\delta}]-E[\sum\varphi_{\delta\delta}])+\lambda^2(-E_Y[\sum\varphi_\delta]^2+E[\sum\varphi_\delta]^2+E_Y[(\sum\varphi_\delta)^2]-E[(\sum\varphi_\delta)^2])\end{aligned}\qquad\text{(E.2)}$$

Annexe F

Espérances précalculées (modèle a priori)

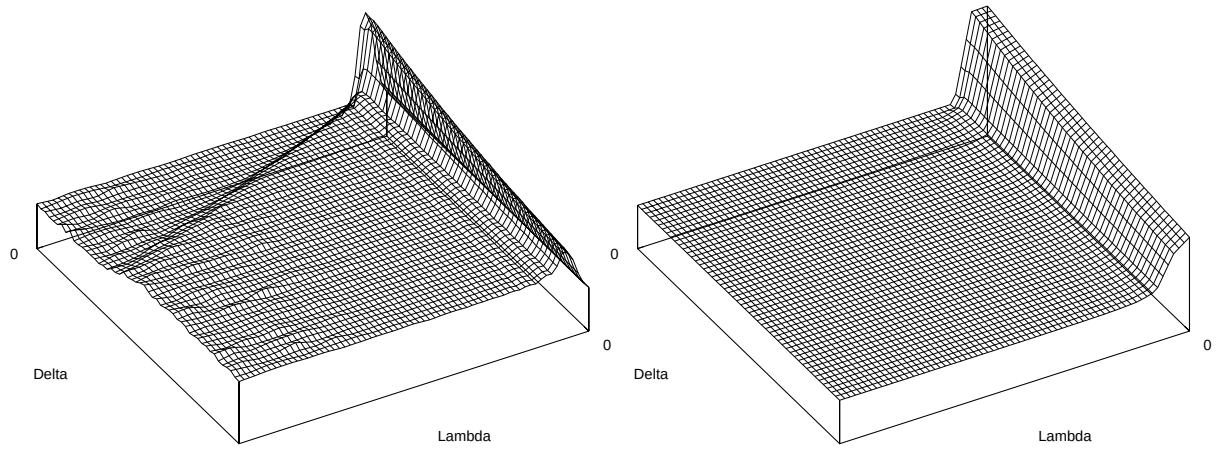


FIG. F.1: *Espérance de l'énergie* $\Phi = \lambda^2 \sum \varphi$ *et de* $\sum \varphi$ *des échantillons a priori (modèle* φ_{GM}*) en fonction de* λ *et* δ*, pour* $\lambda \in [0, 20]$ *et* $\delta \in [0, 50]$

Les bonnes propriétés de régularité[1] de l'espérance de l'énergie permettent, pour les échantillons du modèle a priori, de calculer cette espérance une fois pour toutes, car elle ne dépend que des hyperparamètres (puisqu'il n'y a pas d'attache aux données). Il en est de même pour les quantités qui font intervenir les différentes dérivées de φ par rapport à δ (non représentées).

Il est possible de diminuer le bruit relatif présent sur ces surfaces, en effectuant des moyennes sur de nombreux échantillons, en combinant des élements appartenant à la même chaîne de Markov, avec des éléments d'autres chaînes (ergodicité).

1. La régularité jusqu'à l'ordre 2 est constatée, en pratique, sur tout l'espace des hyperparamètres, hormis la zone de transition de phase (de faibles dimensions), où la dérivée de l'énergie est discontinue.

Pour de grands δ il faut se ramener au modèle quadratique, ce qui permet de déduire les espérances des quantités calculées pour le δ_{max} (par exemple 100) car elles ne dépendent plus alors que du rapport λ/δ.

Le pas d'échantillonnage en λ et δ doit être choisi de manière à préserver la transition de phase (pour les modèles qui en ont une, $\lambda \simeq 1.5$ pour $\delta < 20$ avec φ_{GM}). Le nombre d'itérations utilisé pour produire les échantillons doit être suffisant (minimum 20 pour la zone de transition, avec Gibbs, et il est alors nécessaire de contrôler la convergence).

Annexe G

Coût des algorithmes

- **Restauration ICM-DCT** (algo. 3.4.1)
  La DCT est calculée grâce à une FFT. La FFT à 1 dimension et n pixels demande $5\,n\log_2 n/2$ opérations, à 2 dimensions on effectue $2n$ FFT à 1 dimension, soit $5\,n^2\log_2 n$ opérations en tout, soit $5\log_2 n$ par pixel. Il faut compter environ 15 opérations supplémentaires pour la DCT. 2 DCT sont nécessaires à chaque itération, ainsi que 10 à 20 opérations pour les calculs effectués sur B_xB_y (dépend de la fonction φ utilisée).

  Tout ceci donne environ $50+10\log_2 n$ opérations pour chaque itération, auxquelles il faut ajouter l'initalisation (FFT de h et Y, etc.) soit environ $10+10\log_2 n$ opérations.

  Exemple : image 512×512 → 100 op.pixel^{-1} + 140 op.pixel^{-1}iter^{-1}

- **Echantillonneur Geman-Yang modifié a priori** (algo. 3.3.3)
  Le coût du calcul de chaque échantillon a priori avec cette méthode est un peu plus long que pour chaque itération de l'algorithme ci-dessus. Les B_xB_y et X sont aléatoires, et on utilise de préférence des nombres aléatoires précalculés (en effectuant de temps en temps un décalage aléatoire) pour ne pas allonger le temps de calcul.

  On effectue une approximation de la loi de b pour éviter de calculer la fonction de répartition. On doit ainsi ajouter pour chaque pixel l'équivalent de 10 opérations au maximum.

  L'absence d'attache aux données permet de se passer du calcul des FFT de H et Y.

  Les échantillons sont calculés sur une imagette (extraite de l'image à traiter), car le modèle suppose que les hyperparamètres ne dépendent pas de la zone d'où l'on tire cette imagette. Ceci permet un gain de temps dans l'estimation.

  Exemple : image 128×128 → 130 op.pixel^{-1}échantillon^{-1}

- **Echantillonneur Geman-Yang modifié a posteriori** (algo. 3.2.1)
  Cet algorithme ne diffère du précédent que par la présence de H et Y. Il faut ajouter une étape d'initialisation, car l'échantillon de départ est l'image restaurée par l'algorithme ICM-DCT avec les paramètres de départ.

Le nombre d'opérations nécessaire correspond donc à la somme des coûts des deux premiers algorithmes.

Exemple : image $128 \times 128 \rightarrow 80\,\text{op.pixel}^{-1}$ $+\ 120\,\text{op.pixel}^{-1}\text{iter}^{-1}$
$+\ 130\,\text{op.pixel}^{-1}\text{échantillon}^{-1}$

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