INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Étude de la restitution des paramètres instrumentaux en imagerie satellitaire

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

N° 3957

Juin 2000

THÈME 3

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--3957--FR

Étude de la restitution des paramètres instrumentaux en imagerie satellitaire

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

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

Rapport de recherche n° 3957 — Juin 2000 — 69 pages

Résumé : Le but de cette étude est l'estimation des paramètres du bruit et de la fonction de flou en imagerie satellitaire. En effet, ces images sont dégradées par le système optique, et par un bruit additif lié au capteur. Les paramètres instrumentaux, connus lors du lancement du satellite, peuvent évoluer au cours du temps. Il est alors nécessaire de pouvoir les estimer à partir des images observées, afin de pouvoir corriger ces images, par déconvolution, dans les meilleures conditions.

Le noyau de convolution est paramétré par une fonction traduisant la physique du système imageur étudié. Il s'agit d'estimer les paramètres du noyau, ainsi que la variance du bruit, qui est supposé blanc et gaussien.

Pour la déconvolution à paramètres fixés, nous utilisons une approche variationnelle, qui consiste à minimiser une fonctionnelle traduisant l'attache aux données et la régularisation de l'image cherchée, interdisant l'amplification du bruit tout en préservant les contours.

La méthode proposée repose essentiellement sur deux étapes. Le bruit est estimé en utilisant un filtre passe-bande au moyen d'une transformée en cosinus. Ensuite, l'estimation conjointe du paramètre de régularisation et des paramètres du noyau est effectuée par Maximum de Vraisemblance (MV), en utilisant une méthode de Monte Carlo par Chaînes de Markov (MCMC).

Nous présentons également dans ce rapport un état de l'art des méthodes de déconvolution aveugle, ainsi qu'une étude sur l'estimation du noyau de convolution lorsqu'il n'est pas paramétré.

Mots-clés : Dévonvolution aveugle, Champs de Markov, Modèles variationnels, Estimation, Maximum de Vraisemblance, Paramètres, Images satellitaires

Remerciements : Les auteurs souhaitent remercier le CNES pour l'image de Nîmes (simulation SPOT 5).

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

Study of the estimation of instrumental parameters in satellite imaging

Abstract: The purpose of this study is the estimation of the parameters of the noise and the blur function in remote sensing. Indeed, satellite images are corrupted by the optical system and by an additive noise due to the sensor. The instrumental parameters, known at the lauch of the satellite, can evolve with time. Therefore, it is necessary to estimate them from the observed images, to enable the deconvolution of these images in the best conditions.

The convolution kernel is parametrized by a function which describes the physics of the imaging system. We have to estimate the parameters of the kernel as well as the variance of the noise supposed to be white and Gaussian.

For the deconvolution with fixed parameters, we use a variational approach which consists of minimizing a functional involving the data and the regularization of the solution, avoiding the amplification of the noise while preserving edges.

The proposed method essentially consists of two steps. The noise is estimated using a bandpass filter using a Cosine transform. Then, the joint estimation of the regularizing parameter and the kernel parameters is achieved by computing the Maximum Likelihood (ML), using a Markov Chain Monte Carlo (MCMC) method.

We also present in this report the state of the art of blind deconvolution methods and a study of the estimation of the convolution kernel when it is not parametrized.

Key-words: Blind deconvolution, Markov Random Fields, Variational models, Estimation, Maximum Likelihood, Parameters, Satellite images

Acknowledgements: The authors would like to thank the French Space Agency (CNES) for providing the image of Nîmes (SPOT 5 simulation).

Table des matières

Chapitre 1

Notations, hypothèses

La dimension des images est N_x colonnes par N_y lignes.

X désigne un vecteur, formé par les valeurs d'intensité aux pixels de l'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. On a $X(i\Delta x,j\Delta y) = X(i,j)$.
X^n est le n-ième vecteur de la suite (X^n).
Par abus de notation, on écrit $P(X)$ au lieu de $P(x = X)$ où x est une variable aléatoire multidimensionnelle.

Y est l'observation (image dégradée).
H est l'opérateur de convolution (matrice de dimension $N_x^2 N_y^2$).

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

La réponse impulsionnelle h est positive ($h_{ij} \geq 0$), conserve les valeurs moyennes ($\sum_{ij} h_{ij} = 1$), et respecte la propriété de Shannon.

Dans le cas où h est paramétré, il s'écrit $h_{ij} = f(i,j,k)$ où i et j sont des indices, et k est un vecteur de paramètres. La fonction f est construite de manière à respecter les conditions énoncées ci-dessus.

N représente le bruit. Il est considéré comme additif, gaussien, et dont la variance dépend de l'intensité. Sa variance au pixel (i,j) est donnée par $a + b\,Y_{ij}^0$ où Y^0 est l'image floue non bruitée.
La variance moyenne est σ^2.

L'équation d'observation est $Y = h \star \mathcal{X} + N$ qui s'écrit aussi $Y = H\,\mathcal{X} + N$.

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

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 $2N_x \times 2N_y$.

Ω_X est l'espace de configuration des images (ensemble des images de dimension $N_x \times N_y$ à pixels réels tel que : $\Omega_X = \mathbb{R}^{N_x N_y} \cap \{ -\kappa \leq \sum_{i,j} X_{ij} \leq \kappa \}$).

Ω_h est l'espace de configuration des noyaux h, défini en fonction des contraintes sur h :
$\Omega_h = \mathbb{R}^{+N_x N_y} \cap \{ \sum_{i,j} h_{ij} = 1 \}$.

$\mathcal{F}$ désigne la transformée de Fourier, continue ou discrète.

Chapitre 2

État de l'art

2.1 Introduction

Le problème posé est celui de la déconvolution d'images satellitaires en présence de bruit, lorsque la fonction de transfert de l'appareil optique n'est pas connue. Ce problème est appelé, dans la littérature scientifique, le problème de la déconvolution aveugle. Nous disposons d'une image observée (image dégradée) notée Y, dont on suppose qu'elle vérifie $Y = h \star \mathcal{X} + N$ où h est la réponse impulsionnelle de l'opérateur de flou, $\mathcal{X}$ est l'image réelle non dégradée, et N représente le bruit. Il est supposé blanc, additif, gaussien, stationnaire et indépendant des données, et de variance σ^2. Cette équation s'écrit aussi $Y = H\,\mathcal{X} + N$, où H est la matrice de dimension $N_x^2 N_y^2$, générée à partir de h.
Seules les données Y sont connues. Le problème est de calculer une approximation X de l'image réelle $\mathcal{X}$, ne connaissant que la structure de la dégradation. Il nous faut donc calculer en même temps X, h et σ^2.

Le problème de la déconvolution connaissant la dégradation (h et σ^2) est un problème mal posé qu'il faut régulariser. De manière générale, un modèle est choisi sur X. Il sert de contrainte sur X lors de la déconvolution. La technique de déconvolution que l'on utilise (voir paragraphe 3.1.2 pour les détails et les équations) consiste à minimiser un critère composé d'un terme des moindres carrés issu de l'équation d'observation $\frac{\|Y - h \star X\|^2}{2\sigma^2}$ et d'un ou plusieurs termes de contrainte sur X. Nous utilisons un modèle sur X pour obtenir une régularisation par morceaux, c'est-à-dire une régularisation qui préserve les discontinuités de X afin de reconstruire des images ayant des contours non lissés. Nous utilisons le modèle des φ-fonctions introduit dans l'approche stochastique par utilisation de champs de Markov [9, 20, 21, 22, 23, 25, 27, 29, 44] ou dans l'approche variationnelle [1, 6, 18, 51, 58, 69].

Si le problème de la déconvolution (connaissant h et σ^2) est mal posé, celui de la déconvolution aveugle l'est encore plus car nous augmentons le nombre d'inconnues sans augmenter le nombre de données. A priori, il est difficile de discerner à partir d'une image floue, quelle est la part du flou de celle du signal. Il semble que plusieurs solutions puissent répondre à ce problème, comme par exemple la solution triviale $X = Y$ et h un dirac d'amplitude un. Afin de diminuer l'ensemble des solutions possibles et de régulariser au mieux ce problème, des contraintes doivent être introduites non seulement sur X comme cela est fait en déconvolution classique mais aussi sur h. La plupart des méthodes proposées dans la littérature sur le sujet utilisent des contraintes fortes sur l'image et sur la fonction de flou, comme par exemple des contraintes de positivité (des valeurs négatives ne peuvent correspondre à des signaux physiques). D'autre part, on remarque que des informations a priori sont souvent introduites afin de lever l'indétermination liée à la déconvolution aveugle [28].

Nous présentons brièvement dans le chapitre suivant les grandes catégories de méthodes de déconvolution aveugle.

2.2 Synthèse des méthodes

Un certain nombre de méthodes ont été proposées en déconvolution aveugle. Une grande partie des travaux se situe dans le domaine de l'astrophysique, pour la déconvolution d'images astronomiques. Certaines contraintes introduites sont spécifiques à cette application et ne pourront donc pas correspondre à notre problème (comme la contrainte de support sur X par exemple qui suppose toute image constante sur le bord). Nous présentons néanmoins ces travaux dans la mesure où la méthode considérée peut s'appliquer à notre problème, en changeant ou en retirant simplement cette contrainte. Par contre, nous n'aborderons pas du tout le problème connexe, mais différent, de la déconvolution aveugle à partir de plusieurs images du même objet avec des fonctions de flou différentes [26, 66]. Les problèmes posés sont alors fondamentalement différents, ne serait-ce que par le volume d'inconnues par rapport aux observations.

Nous considérons les travaux en déconvolution aveugle à partir d'une seule observation. Divers classements peuvent être dégagés parmi toutes les méthodes, selon que l'on met en avant les spécificités d'une application, le type de bruit considéré sur l'observation, la paramétrisation de la fonction de flou, les méthodes d'estimation utilisées... Nous avons choisi un regroupement qui s'inspire de celui proposé dans [38, 39].

2.2.1 Identification indépendante de la restauration

Ce paragraphe correspond aux méthodes où l'on s'intéresse à l'identification de la fonction de flou, de manière indépendante à l'estimation de l'image. Ces travaux correspondent à des applications où l'on a de très fortes contraintes sur l'objet ou sur la fonction de flou. Ce sont des applications comme le contrôle non destructif en imagerie X où l'on a une connaissance de certains contours de l'objet imagé X [15]. On considère aussi les applications où la fonction de flou modélise un bougé de la camera ou une défocalisation [11, 16, 45]. Ces fonctions de flou ont des zéros dans l'espace de Fourier. En modélisant un bougé par la direction et la longueur, il est complètement défini par ses zéros dans l'espace de Fourier. Il s'agit donc d'estimer les zéros de l'image observée et de les attribuer à la fonction de flou afin de la reconstruire. Cette technique est un peu sensible au bruit dans la mesure où le bruit fausse l'estimation de l'emplacement des zéros dans l'espace de Fourier. Evidemment, pour les fonctions de flou que nous considèrerons ces techniques ne sont pas adaptées.

2.2.2 Séparation des plages de zéros (*zero sheet*)

Ces méthodes sont basées sur le théorème de Lane et Bates de 1987 [43], qui stipule que le problème consistant à trouver h et X à partir de $Y = h \star X$ a une solution unique à condition qu'un certain nombre d'hypothèses soient vérifiées, comme le fait que la dimension de travail soit ≥ 2, que X et h soient irréductibles (on ne peut trouver X_1 et X_2 tels que $X = X_1 \star X_2$, idem pour h), que l'observation soit non bruitée... Ces hypothèses ne sont pas vérifiées en pratique et les algorithmes issus de cette approche sont très sensibles au bruit. Ils sont, d'autre part, de complexité N^4 si N est le nombre de pixels de l'image. Ce théorème est toutefois intéressant pour la compréhension du problème de la déconvolution, analysé à travers les zéros de la transformée en Z de Y, dans un espace de dimension supérieur à deux.

2.2.3 Modélisation ARMA et identification

L'image X est modélisée par un modèle AutoRégressif (AR). Le système formé de l'équation du modèle AR et de l'équation d'observation $Y = h \star X + N$. Ceci peut se récrire en une nouvelle équation formant un modèle ARMA, la partie moyenne mobile MA (*Moving Average*) correspondant à l'équation d'observation. En utilisant les techniques d'identification des modèles ARMA, les paramètres du modèle et ceux de la dégradation sont estimés en même temps. Ces paramètres sont estimés au sens du maximum de vraisemblance en utilisant un algorithme EM par exemple, [7, 40, 41] ou par validation croisée généralisée [55]. L'inconvénient de ces méthodes est la simplicité du modèle sur l'image X dans le sens où le modèle est linéaire et stationnaire, ce qui ne correspond pas du tout à des hypothèses réalistes pour une image. X peut être considéré linéaire et stationnaire par morceaux, mais cette hypothèse ne peut s'étendre au niveau global sans entraver la restitution des contours de l'image.

2.2.4 Paramétrisation de la fonction de flou

Une autre catégorie de techniques consiste à donner une forme imposée à la fonction de flou, cette forme étant paramétrée par un nombre restreint de paramètres. C'est le moyen le plus simple d'imposer un très grand nombre de contraintes sur h simultanément, comme la positivité, le fait que h n'ait pas de gain ($\sum_i h_i = 1$), que les coefficients soient décroissants dans toutes les directions depuis l'origine, que le filtre coupe les hautes fréquences...

Ce petit nombre de paramètres peut alors être estimé par les techniques d'estimation de paramètres classiques. Le cadre de l'estimation est celui des données incomplètes puisque X est inconnu. On peut considérer l'estimation des paramètres de h par maximum de vraisemblance $P(Y \mid \theta)$, où θ représente le vecteur contenant tous les paramètres à estimer : $\theta = (\lambda, k_1, \ldots, k_m, \sigma)$. Dans le cas des données incomplètes, cette vraisemblance n'est pas unimodale. Les méthodes par descente de gradient ne donneront qu'un minimum local. Ce sera le cas du gradient stochastique généralisé développé par L. Younes [71], qui est une extension de la méthode du gradient stochastique en données complètes.

De plus, le calcul de cette probabilité est fait par inférence sur X, puisque X est inconnu (voir chapitre suivant), ce qui conduit au calcul d'espérance mathématique (et donc d'échantillonnage) selon les lois a priori et a posteriori sur X pour l'évaluation du gradient de $P(Y \mid \theta)$ par rapport à θ. L'approximation de ces espérances peut être envisagée par construction de chaînes de Markov, comme il est fait dans la méthode du gradient stochastique généralisé [71] ou dans [29]. Une difficulté est le calcul d'échantillons suivant la loi a posteriori de X, car le terme d'attache aux données induit, au travers de l'opérateur H, un voisinage de dépendance entre les pixels très étendu, rendant impossible l'utilisation d'algorithmes classiques d'échantillonnage tels que l'échantillonneur de Gibbs ou la dynamique de Metropolis. Cependant, même si l'on sait développer des échantillonneurs pour la loi a posteriori, par passage dans le plan de Fourier [29] , il n'est pas certain que le nombre d'échantillons nécessaires à l'estimation de ces espérances ne soit prohibitif dans le cas de l'estimation des paramètres de la fonction de flou.

Il est possible de se ramener à des méthodes ne demandant qu'un échantillonnage de la loi a priori par des techniques d'optimisation alternées en X et h. Un certain nombre d'algorithmes itératifs pour l'estimation de paramètres en données incomplètes ont été proposés dans la littérature, dont le plus populaire est sans doute l'algorithme E.M. ("Expectation-Maximization") introduit dans [19] et utilisé dans [13, 14, 36, 72]. Une amélioration de l'algorithme EM a été proposée par Celeux et Diebolt sous forme de EM stochastique (dit SEM) [12]. Un autre type d'algorithme consiste à maximiser la vraisemblance généralisée $P(Y{,}X \mid \theta)$ par une méthode sous-optimale de minimisation alternée sur les paramètres θ et l'image X. C'est ce que proposent Lakshmanan et Derin [42] avec le recuit simulé adaptatif ou bien Pieczynski [53] pour l'algorithme de l'estimation conditionnelle itérative, dit ICE ("Iterated Conditional Estimation").

Une telle technique itérative a été développée pour la déconvolution aveugle dans [60]. Les auteurs proposent une paramétrisation de la fonction de flou, basée sur un modèle physique et en estiment les paramètres et l'image par un algorithme EM. L'estimation de X s'effectue par l'algorithme de Richardson-Lucy [46, 56], très populaire en astrophysique. C'est un algorithme de gradient de type multiplicatif, *i.e.* un algorithme dont le but est de résoudre l'équation $Y = HX$ lorsque les données sont dégradées par un bruit de Poisson et non plus gaussien. C'est le cas de données à faible flux comme le comptage de photons, que l'on retrouve en médecine nucléaire par exemple ou en astrophysique. On rappelle que dans le cas additif, l'itération m de l'algorithme de gradient est

$$X^{m+1} = X^m + aH^*(Y - HX^m) \tag{2.1}$$

et dans le cas multiplicatif

$$X^{m+1} = X^m.H^*(Y/(HX^m)) \tag{2.2}$$

où les caractères '.' et '/' symbolisent des opérations entre vecteurs effectuées point à point. L'algorithme multiplicatif est à convergence lente, c'est-à-dire que s'il n'y a pas de bruit sur les observations, il faut beaucoup d'itérations pour avoir une bonne précision sur les données. Lorsqu'il y a du bruit, l'algorithme doit être arrêté avant que le bruit ne soit amplifié, mais la plage des itérations admissibles est grande. L'itération d'arrêt est fixée de manière arbitraire. Ceci ne peut être fait avec un algorithme d'inversion à convergence plus rapide comme le gradient conjugué par exemple car le bruit est amplifié trop rapidement. C'est la raison principale pour laquelle cet algorithme a été longtemps utilisé en médecine (l'algorithme est déduit de l'algorithme EM [19]) ou en astrophysique. La régularisation est obtenue par arrêt de l'algorithme itératif mais les résultats sont peu sensibles à l'instant d'arrêt.

La deuxième raison de son succès est qu'étant multiplicatif, les signes sont conservés, autrement dit si l'estimée initiale est positive, la solution le sera aussi et partout où il y a des zéros, ils seront conservés. C'est pourquoi on retrouve souvent l'algorithme de Richardson-Lucy pour la déconvolution d'images astronomiques. Dans [60], l'algorithme de Richardson-Lucy est utilisé au cours de l'algorithme EM pour l'estimation conjointe de X et des paramètres définissant h.

2.2.5 Fonction de flou estimée en tout point

Dans le paragraphe précédent, après reparamétrisation, le nombre d'inconnues définissant la fonction de flou est faible. Elles sont donc estimées en tant que paramètres déterministes d'une loi de distribution. Ici au contraire, sans reparamétrisation, le nombre de coefficients définissant h est relativement élevé. La suite de ces coefficients forme la réponse impulsionnelle h.

Du point de vue de l'estimation, ces coefficients sont alors considérés comme aléatoires, avec une certaine distribution de probabilité. Cette distribution de probabilité contiendra l'information correspondant aux contraintes sur h. Ces paramètres seront estimés selon les estimateurs statisques classiques, de la même manière que le signal X. C'est dans cette catégorie de méthodes que l'on trouve la plupart des méthodes les plus générales pour la déconvolution aveugle.

Le principe en est simple : le terme d'attache aux données $\frac{\|Y-HX\|^2}{2\sigma^2}$ sera minimisé en H et en X. Des contraintes sont introduites au cours de la minimisation. Elles peuvent l'être de manière artificielle comme dans Gerchberg et Saxton [24] et l'on ne sait plus ce que l'on minimise, ni si l'algorithme va converger. Elles le seront en général de manière plus propre, soit par projection, soit par pénalité, soit par reparamétrisation, ou encore par minimisation sous contrainte. Nous serons amenés à minimiser un critère du type

$$\frac{\|Y - HX\|^2}{2\sigma^2} + U(X) + U(H)$$

où $U(X)$ et $U(H)$ comprennent les termes de contrainte sur X et H.

L'interprétation bayesienne est de considérer la maximisation de la loi jointe a posteriori $P(X,h \mid Y) \propto P(Y \mid X,h)P(h)P(X)$. Les contraintes seront exprimées dans les loi a priori $P(X)$ et $P(h)$. Ces critères seront minimisés alternativement en X à h fixé et en h à X fixé. Cette procédure est courante et se retrouve sous le terme d'algorithme IBD ("Iterative Blind Deconvolution").

La contrainte de positivité sera satisfaite naturellement si l'algorithme utilisé est celui de Richardson-Lucy. Sinon, cette contrainte peut être imposée par l'introduction d'une pénalité sur les valeurs négatives, sous la forme d'un terme $\sum_s \varphi^+(X_s)$ où $\varphi^+(t) = t$ si $t < 0$ et $\varphi^+(t) = 0$ sinon. Cette technique est utilisée dans [35] par exemple. Dans ce travail, les auteurs imposent un certain nombre de contraintes sur X et h dans le domaine spatial et fréquentiel sous forme de pénalité comme pour la positivité, et le critère est dérivé en X et en h. Les équations dérivées sont annulées alternativement en X à h fixé et en h à X fixé, selon la procédure IBD. Bien sûr ce type de procédure est sensible à l'estimée initiale et peut être instable.

Ce type de procédure se retrouve dans [67, 68] où le critère considéré est

$$\frac{\|Y - HX\|^2}{2\sigma^2} + \lambda \sum_s \varphi(|\nabla X|_s) + \mu \sum_s \varphi(|\nabla H|_s)$$

Les variables X et h sont régularisées de manière anisotrope par des φ-fonctions. Le critère est minimisé en effectuant quelques itérations de gradient conjugué sur X puis sur H itérativement. Les contraintes sur H ($h_i > 0$ et $\sum_i h_i = 1$) sont introduites par projection.

Ce même principe est repris dans [5] et [64], mais l'algorithme minimise

$$\frac{\|Y - HX\|^2}{2\sigma^2}$$

alternativement en X et h par l'algorithme de Richardson-Lucy. Les contraintes de positivité sont nécessairement vérifiées et la contrainte de normalisation de h est obtenue par projection à chaque itération sur h. La régularisation est obtenue par arrêt de l'algorithme de Richardson-Lucy sur X et h.

Dans Ayers et Dainty [2], les auteurs proposent un algorithme itératif IBD basé sur des reconstructions alternées de X et de h par algorithme de Wiener, avec introduction de contraintes sur X et h (type support, positivité, fréquence de coupure...), par projection dans l'espace des contraintes en spatial et en fréquentiel. L'algorithme utilise donc des transformées de Fourier directes et inverses pour la projection des contraintes dans l'espace de Fourier.

Dans [62], les contraintes sont introduites de manière plus strictes : X_s est reparamétré par X_s^2 pour assurer la contrainte de positivité et h_i est reparamétré par $\frac{h_i^2}{\sum_i h_i^2}$ pour assurer en même temps la positivité et la contrainte de normalisation. D'autres contraintes sont utilisées (contrainte de support sur h en spatial et en fréquentiel, ainsi qu'une contrainte d'aberration de phase) et le critère final est minimisé par deux gradients conjugués itératifs, et constitue encore un algorithme IBD.

Un autre algorithme itératif, différent des algorithmes IBD, est celui proposés par Kundur [38, 39]. Il consiste à trouver un filtre FIR u^k, qui varie au cours des itérations k. Ce filtre est appliqué sur l'image floue Y. Les coefficients du filtre sont optimisés de manière à ce que la sortie $X^k = Y * u^k$ vérifie un certain nombre de contraintes, par minimisation d'un critère. Par exemple, u^{k+1} sera obtenu par minimisation de

$$\sum_{s \in D_{sup}} {X_s^k}^2 \left[\frac{1 - sgn(X_s^k)}{2} \right] + \sum_{s \in \bar{D}_{sup}} [X_s^k - L_B]^2 + \gamma \left[\sum_s u_s - 1 \right]^2$$

où $X^k = Y * u^k$, D_{sup} est le support de l'objet imagé et $\bar{D}_{sup}$ est le complémentaire. Dans ce critère, des contraintes de support et de positivité sont introduites. Il est possible d'introduire aussi des contraintes de lissage linéaire sur X^k. Si la convergence semble plus rapide et plus sûre que celle des algorithmes IBD, cet algorithme a cependant un gros inconvénient. En effet, cette approche par filtrage s'apparente à un filtre inverse, certes optimisé pour que la solution vérifie les contraintes, mais l'amplification du bruit n'est pas prise en compte explicitement. Les auteurs proposent d'éviter l'amplification du bruit par arrêt de l'algorithme, dont la convergence est plus rapide et plus sûre que celles des algorithmes IBD.

Citons encore, pour terminer sur ce type d'approche, les travaux de McCallum [47] où le critère $\frac{\|Y - HX\|^2}{2\sigma^2}$ est minimisé en h et X par un algorithme de recuit simulé, dans lequel h et X sont imposées positives et à support connu.

Chapitre 3

Méthode proposée : étude théorique

3.1 Introduction

3.1.1 Principe

Nous rappelons que l'équation d'observation, modélisant la formation de l'image observée Y à partir d'une image originale $\mathcal{X}$, s'écrit :

$$Y = h \star \mathcal{X} + N \tag{3.1}$$

où h est le noyau de convolution et N le bruit. L'image réelle $\mathcal{X}$, le noyau h et les paramètres du bruit sont inconnus (la loi du bruit est donnée).

Dans ce chapitre, nous proposons une méthode automatique permettant à la fois d'estimer les paramètres instrumentaux, comme le bruit et le noyau de convolution, et de déconvoluer l'image observée.

La déconvolution étant un problème inverse mal posé, il est régularisé par ajout d'un terme qui pénalise les solutions trop bruitées. Ce problème inverse, dans notre approche, concerne tout aussi bien la restauration de l'image X que la restitution du noyau de convolution h.

Ainsi, il est possible de construire un algorithme complet, qui ne nécessite qu'une image observée, et très peu de connaissance a priori sur cette image et la manière dont elle a été dégradée. En effet, les images satellitaires sont modélisées par des zones homogènes séparées par des contours francs, admettant également des détails assez fins (au niveau du pixel). Nous utilisons pour cela une pénalisation basée sur des φ-fonctions (modèle non linéaire).

D'autre part, le noyau de convolution est supposé positif et conserve les moyennes, ce qui est physiquement réaliste dans la mesure où le flou provient du système optique et des mouvements du système imageur.

Pour le respect de ces contraites, ainsi que pour prendre en compte la physique du capteur, il est apparu nécessaire de paramétrer le noyau de la manière suivante, par une fonction f :

$$h_{ij} = f(i,j,k) \tag{3.2}$$

où k est le vecteur de m paramètres, $k = (k_1,...,k_m)$.

Une étude menée dans le cadre d'un contrat CNES [30] a montré que si l'on cherche à estimer le noyau en tout point, il faut alors le régulariser de la même façon que l'image. Il est alors non seulement il est difficile (voire impossible?) d'estimer les paramètres de régularisation associés, mais la forme du noyau est étroitement liée au modèle imposé. La méthode d'estimation est, en plus, particulièrement coûteuse en temps de calcul.

Si l'on impose directement la forme du noyau sous forme d'une fonction (3.2), le noyau obtenu dépend très étroitement des paramètres k mais il n'y a pas besoin de le régulariser en calculant, par exemple, un maximum a posteriori. De plus, les contraintes de positivité et de normalisation peuvent être imposées directement dans la fonction f. Ce type d'approche a été utilisé par exemple dans [45] ou [60].

Il nous semble plus judicieux de paramétrer le noyau par une fonction connue, relative à la physique du problème étudié (par exemple une gaussienne, un bougé linéaire...), que de chercher à l'estimer en tout point.

Enfin, le bruit est supposé blanc et gaussien. Comme il est essentiellement d'origine thermique (l'observation de la Terre s'effectue à fort flux), l'hypothèse est ici réaliste. Il est stationnaire, mais sa variance dépend de l'intensité. Ses paramètres sont estimés au préalable à partir du spectre de Y.

Nous supposons, lors de la reconstruction et de l'estimation du noyau de convolution, que la variance du bruit, notée σ^2, est constante, pour des raisons pratiques (moindre complexité calculatoire). Cette variance est obtenue en moyennant les variances calculées lors de l'estimation des paramètres du bruit.

Pour estimer les hyperparamètres du modèle de régularisation, ainsi que les paramètres du noyau, il est possible d'utiliser l'estimateur du Maximum de Vraisemblance (MV) calculé sur l'image observée. Cet estimateur a de bonnes propriétés, dans la mesure où il est efficace et non biaisé [37, 49], lorsque l'on dispose d'un grand nombre de données. L'optimisation de la vraisemblance est effectuée, dans ce cas, par une technique stochastique basée sur des méthodes de Monte Carlo par chaînes de Markov (MCMC) [57]. Son calcul implique l'évaluation d'espérances, nécessitant d'échantillonner une suite d'images (X_n) par rapport à certaines lois de probabilité qui entrent en jeu dans cet estimateur.

L'image est déconvoluée en utilisant une approche basée sur celle que nous avons développée dans [29], dans laquelle le paramètre du modèle de régularisation sur l'image, ainsi que les paramètres du noyau, sont estimés en même temps que l'image.

3.1.2 L'approche bayésienne

Maximum a posteriori

$P(Y \mid X,k,\sigma)$ désigne ici la vraisemblance de X, k et σ, i.e. la probabilité d'obtenir une image Y par dégradation de X, sachant que tous les paramètres instrumentaux sont connus. D'après l'équation d'observation (3.1), X étant fixé, Y suit la distribution du bruit (supposé gaussien de variance σ^2), qui est indépendant de X. Comme $N = Y - h \star X$, on a donc :

$$P(Y \mid X,k,\sigma) = P(N \mid X) = K_\sigma^{-1} e^{-\|N\|^2/2\sigma^2} = K_\sigma^{-1} e^{-\|Y-h\star X\|^2/2\sigma^2} \tag{3.3}$$

K_σ est une constante de normalisation qui ne dépend ni de X ni de h. On remarque alors que la minimisation de $\|Y - h \star X\|^2$ par rapport à X équivaut à la maximisation de $P(Y \mid X,k,\sigma)$: le critère des *moindres carrés* est équivalent au critère du *maximum de vraisemblance*, dans le cas d'un bruit additif gaussien, ce qui est bien connu. Si l'on cherche à maximiser cette vraisemblance pour restaurer l'image, on tient compte alors uniquement de l'adéquation aux données et la solution obtenue est très bruitée (en effet, le problème est mal posé au sens d'Hadamard), même si elle permet précisément de retrouver Y en appliquant l'équation d'observation.

Pour estimer les paramètres du noyau de convolution, on cherche à maximiser la probabilité d'avoir X à l'origine de l'observation Y, sachant Y et les paramètres σ, λ et k :

$$(\hat{X})_{MAP} \equiv \arg\max_{X,h} P(X \mid Y,\lambda,k,\sigma) \tag{3.4}$$

Expression des différentes lois de probabilité

$\star$ ***Régularisation de l'image***

$P(X \mid \lambda)$ correspond à la connaissance a priori de l'image X que l'on veut obtenir. C'est ici que l'on introduit les contraintes de lissage avec préservation des contours sur l'image, autrement dit, la *régularisation*. Cela signifie simplement qu'une image n'ayant pas les propriétés de régularité désirées est considérée comme peu probable.

L'image X est un champ de Markov [4] : 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 petite taille devant la taille de l'image à traiter.

$$P(X_i \mid X_j, j \neq i) = P(X_i \mid V_i), \forall (i,j) \in S^2 \tag{3.5}$$

THÉORÈME 3.1.1 (HAMMERSLEY-CLIFFORD [23])
X 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 ce même 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 $\Phi(X)$.

$$P(X \mid \lambda) = \frac{1}{Z_\lambda} e^{-\lambda \Phi(X)} \quad \text{avec } Z_\lambda = \int_{X \in \Omega_X} e^{-\lambda \Phi(X)} \tag{3.6}$$

où $\Phi(X)$ s'écrit en fonction des gradients de l'image X :

$$\Phi(X) = \sum_{ij} \left\{ \varphi\left(\frac{(D_x X)_{ij}}{\delta} \right) + \varphi\left(\frac{(D_y X)_{ij}}{\delta} \right) \right\} \tag{3.7}$$

où D_x et D_y représentent les opérateurs de dérivation suivant les colonnes et les lignes.

Ω_X est l'espace des configurations de X (voir chapitre 1). Z_λ est une constante appelée fonction de partition, par analogie avec la physique statistique ($1/\lambda$ est la température et $\Phi(X)$ l'énergie du champ X). Cette énergie est définie à partir de l'ensemble C des cliques associées au système de voisinages V_s. Une clique $c \in C$ est soit un point seul, soit un ensemble de points tels que : $\forall s,t \in c \quad s \neq t \Longrightarrow s \in V_t$. A chaque type de voisinage correspond un ensemble de cliques. Nous utilisons ici la différence entre deux pixels adjacents, c'est-à-dire que les cliques utilisées sont constituées uniquement de deux pixels voisins. φ est une fonction potentiel à définir. Par exemple, φ peut être une des fonctions habituellement utilisées dans les modèles variationnels [17, 18]. La fonction φ est symétrique, positive et croissante ; elle est appliquée aux gradients de l'image de façon à pénaliser les images très bruitées.

En prenant par exemple $\varphi(u) = u^2$, on effectue une régularisation au sens de Tikhonov [63]. 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.

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

PROPRIÉTÉ 3.1.1 (FONCTION POTENTIEL φ [17, 18])
Pour préserver les détails tout en lissant les zones homogènes, φ doit vérifier 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. Les principales fonctions utilisées sont illustrées sur la figure 3.1, leurs expressions sont données par le tableau 3.2.

Habituellement, on utilise un facteur d'échelle δ pour adapter la forme de la fonction φ à l'amplitude des contours présents dans l'image (voir équation (3.7)). Ce paramètre est fixé, car son estimation pose des problèmes d'unicité comme il a été montré dans [29], et aussi parce qu'il ne constitue pas réellement un paramètre de régularisation mais juste un moyen de déformer le modèle.

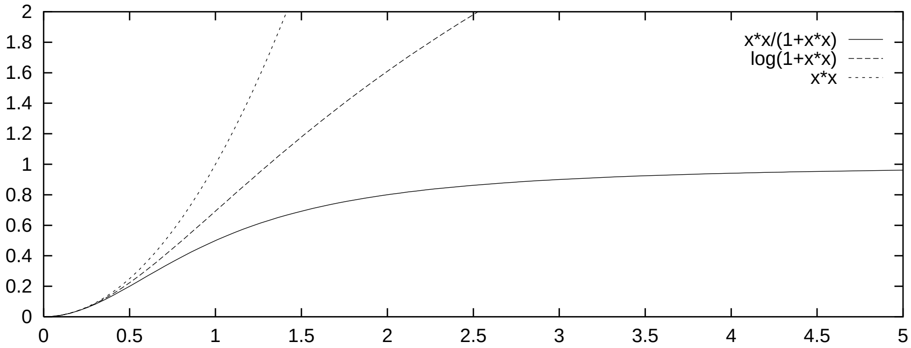

FIG. 3.1 – *Quelques fonctions* $\varphi(u)$ *en fonction de* u *:* φ_{GM}, φ_{HL}, *quadratique.*

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

FIG. 3.2 – *Tableau des fonctions* $\varphi(u)$

$\star$ ***Expression de la loi a posteriori***

En utilisant le théorème de Bayes, on développe la loi (3.4) :

$$P(X \mid Y,\lambda,k,\sigma) = \frac{P(Y \mid X,\lambda,k,\sigma)P(X \mid \lambda,k,\sigma)}{P(Y \mid \lambda,k,\sigma)} \tag{3.8}$$

Si l'on cherche à la maximiser par rapport à X, le terme $P(Y \mid \lambda,k,\sigma)$ est fixé.

Finalement, on obtient l'expression suivante pour la loi a posteriori, en utilisant les expressions de la vraisemblance (3.3) et de la loi a priori (3.7) :

$$P(X \mid Y,\lambda,k,\sigma) \propto \frac{1}{Z_{Y,\lambda,\mu,\sigma}} e^{-\|Y-h\star X\|^2/2\sigma^2-\lambda\Phi(X)} \tag{3.9}$$

avec la fonction de partition a posteriori définie par :

$$Z_{Y,\lambda,\mu,\sigma} = \int_{X\in\Omega_X} e^{-\|Y-h\star X\|^2/2\sigma^2-\lambda\Phi(X)}\, dX \tag{3.10}$$

Ainsi, on voit que minimiser l'énergie $U(X,h)$ définie par :

$$U(X) = \|Y - h \star X\|^2/2\sigma^2 + \lambda\Phi(X) \tag{3.11}$$

est équivalent à maximiser la loi a posteriori, car elle est proportionnelle à $e^{-U(X)}$.

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, cette approche étant d'ailleurs assez répandue dans la littérature [1, 8, 18, 69]. Dans le cas contraire, seul un algorithme stochastique [23] (tel le recuit simulé [65]), parviendra à sortir des minima locaux.

Nous utilisons ici une approche déterministe, basée sur des travaux que nous avons précédemment effectués [29, 32].

Estimation des paramètres par maximum de vraisemblance

L. Younes [70, 71] propose d'utiliser le *maximum de vraisemblance sur Y*, c'est-à-dire de maximiser la probabilité de l'image observée Y sachant les paramètres (λ,k,σ).

$$(\hat\lambda,\hat k,\hat\sigma) = \arg\max_{\lambda,k,\sigma} P(Y \mid \lambda,k,\sigma) \tag{3.12}$$

L'observation étant effectuée, cette vraisemblance ne dépend plus que des paramètres. On a :

$$P(Y \mid \lambda,k,\sigma) = \int_{X\in\Omega_X} P(Y,X \mid \lambda,k,\sigma)\, dX \tag{3.13}$$

D'après le théorème de Bayes, on a :

$$P(Y \mid \lambda,k,\sigma) = \int_{X\in\Omega_X} P(Y \mid X,\lambda,k,\sigma)P(X \mid \lambda)\,dX \tag{3.14}$$

Pour calculer cette probabilité, on utilise les expressions de la distribution a priori (3.6) et a posteriori (3.9). On obtient alors la relation :

$$P(Y \mid \lambda,k,\sigma) = \frac{Z_{Y,\lambda,k,\sigma}}{Z_\lambda Z_\sigma} \tag{3.15}$$

où Z_λ, Z_σ et $Z_{Y,\lambda,k,\sigma}$ sont respectivement les fonctions de partition correspondant à la loi a priori de X, à la vraisemblance de X, et à la loi a posteriori.

Toute la difficulté de l'estimation des hyperparamètres provient de ces fonctions de partition, qui ne sont pas calculables directement.

Nous chercherons donc à minimiser $L(\lambda,k,\sigma) = -\log P(Y \mid \lambda,k,\sigma)$:

$$L(\lambda,k,\sigma) = -\log Z_{Y,\lambda,k,\sigma} + \log Z_\lambda + \log Z_\sigma \tag{3.16}$$

par rapport à chacun des paramètres, par une méthode de descente. En effet, si l'on calcule les dérivées de L par rapport à λ, k et σ, il est possible d'utiliser une méthode de type Newton-Raphson ou de type gradient conjugué pour effectuer l'optimisation.

On peut écrire $\partial L(\hat{\theta})/\partial\theta_i = 0$ pour chaque θ_i si θ représente le vecteur des paramètres, i.e. $\theta = (\lambda,k_1,...,k_m,\sigma)$. L'estimateur du maximum de vraisemblance est alors noté $\hat{\theta}$. Les paramètres optimaux vérifient donc les équations suivantes :

$$\frac{\partial \log Z_{Y,\hat{\lambda},\hat{k},\hat{\sigma}}}{\partial\lambda} = \frac{\partial \log Z_{\hat{\lambda}}}{\partial\lambda} \tag{3.17}$$

$$\forall i,\quad \frac{\partial \log Z_{Y,\hat{\lambda},\hat{k},\hat{\sigma}}}{\partial k_i} = 0 \tag{3.18}$$

$$\frac{\partial \log Z_{Y,\hat{\lambda},\hat{k},\hat{\sigma}}}{\partial\sigma} = \frac{\partial \log Z_{\hat{\sigma}}}{\partial\sigma} \tag{3.19}$$

$\star$ ***Calcul des dérivées de la log-vraisemblance***

Si l'on note θ le vecteur de tous les paramètres, θ_i étant un des paramètres, on utilise la propriété suivante :

PROPRIÉTÉ 3.1.2 (DÉRIVÉE DE LOG Z)
Si $P(X \mid \theta) = Z^{-1}e^{-u(X,\theta)}$, avec la fonction de partition $Z = \int_{X\in\Omega_X} e^{-u(X,\theta)}$, alors on a

$$\frac{\partial \log Z}{\partial\theta_i} = -E_{(X)\sim P(X \mid \theta)}\left[\frac{\partial u(X,\theta)}{\partial\theta_i}\right] \tag{3.20}$$

où $E[\,]$ désigne l'espérance sur X de loi $P(X \mid \theta)$.

DÉMONSTRATION 3.1.1
$\partial \log Z / \partial\theta_i = Z^{-1}\, \partial Z/\partial\theta_i = \int_{X\in\Omega_X} \partial u(X,\theta)/\partial\theta_i\, Z^{-1} e^{-u(X,\theta)}$
et comme $P(X) = Z^{-1}e^{-u(X,\theta)}$ *on obtient* $\int_{X\in\Omega_X} P(X)\,\partial u(X,\theta)/\partial\theta_i$*, ce qui est bien l'espérance* $E_{(X)\sim P(X)}[\partial u(X,\theta)/\partial\theta_i]$.

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

$$E_{(X)\sim P(X)}\left[\frac{\partial E(X)}{\partial\theta_i}\right] \simeq \frac{1}{n}\sum_{(X^k)\sim P(X)} \frac{\partial E(X^k)}{\partial\theta_i} \tag{3.21}$$

si n est le nombre d'échantillons X tirés avec la loi de probabilité $P(X \mid \theta_i)$.

On note $E_{Y,\lambda,k,\sigma}[\,]$, $E_\lambda[\,]$ et $E_\sigma[\,]$ les espérances respectivement par rapport aux lois (3.9), (3.6) et (3.3).

Les équations (3.20) et (3.21), écrites pour une loi quelconque notée $P(X)$, peuvent être facilement réécrites avec la loi a priori $P(X \mid \theta)$ ou la loi a posteriori $P(X \mid Y,\theta)$. a posteriori $P(X)$, peuvent être facilement adaptées à la loi $P(X)$.

On a alors à résoudre le système stochastique suivant :

$$E_{Y,\hat\lambda,\hat k,\hat\sigma}[\Phi(X)] = E_{\hat\lambda}[\Phi(X)] \tag{3.22}$$

$$E_{Y,\hat\lambda,\hat k,\hat\sigma}\left[\frac{\partial U(X)}{\partial k}\right] = 0 \tag{3.23}$$

L'équation (3.19) donne quant à elle une expression directe de σ^2 en fonction de l'espérance a posteriori :

$$E_{Y,\hat\lambda,\hat k,\hat\sigma}\left[\|Y - h\star X\|^2\right] = \sigma^3 \frac{\partial \log Z_{\hat\sigma}}{\partial\sigma} \tag{3.24}$$

or la fonction de partition Z_σ est calculable (elle correspond à une loi gaussienne), on a $Z_\sigma \propto (\sigma\sqrt{2\pi})^{N_xN_y}$. On trouve alors :

$$\hat\sigma^2 = \frac{1}{N_xN_y} E_{Y,\hat\lambda,\hat k,\hat\sigma}\left[\|Y - h\star X\|^2\right] \tag{3.25}$$

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

- $E_\lambda[\,]$ calculée avec la loi a priori de l'image $X \sim e^{-\lambda\Phi(X)}$, *sans attache aux données* (équation (3.6)) ;
- $E_{Y,\lambda,k,\sigma}[\,]$ calculée avec la loi a posteriori $X \sim e^{-\|Y-h\star X\|^2/2\sigma^2 - \lambda\Phi(X)}$, *avec attache aux données* (équation (3.9)) ;

Comment pouvons-nous construire des images X de distribution donnée $P(X)$? 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*, d'où l'appellation méthode MCMC [57].

3.2 Échantillonnage

Nous allons chercher à résoudre les équations stochastiques (3.22) et (3.23) afin d'estimer les paramètres k du noyau h et le paramètre de régularisation λ, ainsi que l'équation (3.25) pour estimer la variance du bruit σ^2.

Ces équations faisant intervenir des espérances, nous avons vu que leur calcul nécessite d'échantillonner les différentes lois de probabilité : la loi a posteriori $P(X \mid Y,\lambda,k,\sigma)$ et la loi a priori $P(X \mid \lambda)$. Nous allons expliquer, dans cette partie, quelles sont les difficultés rencontrées lors de l'échantillonnage, quel est le principe des échantillonneurs, et nous détaillerons finalement les algorithmes d'échantillonnage que nous avons développés.

3.2.1 Difficultés de l'échantillonnage

Convergence des échantillonneurs

Soit $\pi(X)$ la loi à échantillonner. Un échantillonneur est un algorithme itératif, permettant de créer une chaîne de Markov (X^n), constituant une suite d'images dont la distribution de probabilité converge vers la distribution stationnaire $\pi(X)$. T est la matrice de transition permettant de passer d'un échantillon X^n au suivant X^{n+1} avec la probabilité de transition :

$$P_{n+1}(X) = \int_{\Omega_X} P_n(X')\, T_n(X',X)\, dX' \tag{3.26}$$

On peut trouver dans [57, 59, 61] 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 T), est une *condition suffisante* de convergence.

Loi a posteriori

L'échantillonnage soulève ici une difficulté, quelle que soit la manière dont h est estimé : pour la loi a posteriori, il faut tenir compte de l'*attache aux données*, donc de la convolution ce qui interdit, pour des raisons pratiques, l'utilisation d'échantillonneurs classiques comme Gibbs [23] ou Metropolis [48].

Attache aux données

Le voisinage correspondant au champ de Markov n'est pas limité au premier ordre (4 voisins), à cause de l'opérateur de convolution par 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 liés à la dynamique de Metropolis et à l'échantillonneur de Gibbs est le calcul de l'énergie U lorsque l'on modifie un pixel au hasard. Dans le premier cas, on cherche une variation d'énergie locale, et dans le second cas, on construit une fonction de répartition en calculant les probabilités conditionnelles locales. Cette énergie est d'autant plus longue à calculer qu'elle fait intervenir un grand nombre de pixels, ici $(2p+1)^2$. 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 au minimum).

Il est indispensable d'utiliser un autre type d'échantillonneur, si l'on veut tenir compte de l'attache aux données. C'est pourquoi nous avons introduit dans [29] une version modifiée de l'algorithme de Geman & Yang [21]. Son principe réside dans la diagonalisation de l'opérateur de convolution, ce qui équivaut à remplacer la convolution, lourde à calculer, par une multiplication pixel par pixel dans l'espace des fréquences. On utilise pour cela une transformée de Fourier (FFT) dans la méthode originale. Nous avons utilisé une transformée en Cosinus (DCT) permettant de respecter les conditions aux bords symétriques de l'image et d'éviter ainsi les artefacts qui apparaissent avec la FFT.

3.2.2 Introduction de variables auxiliaires

Nous avons proposé dans [29, 31] un algorithme d'échantillonnage de X dans l'espace fréquentiel en introduisant une variable auxiliaire. Cela permet de *linéariser* la dérivée du terme de régularisation (3.7) contenant la fonction φ.

La puissance de la régularisation par φ-modèle réside dans le fait qu'elle n'est pas quadratique. Mais c'est aussi de là que proviennent les difficultés du calcul d'optimisation et de l'échantillonnage. Si φ est quadratique, la fonctionnelle à minimiser est quadratique, donc le minimum est unique et facile à calculer en une seule étape, par résolution d'une équation linéaire.

On va donc chercher à se ramener à un modèle quadratique. Cela consiste à remplacer $\varphi(u)$ par une nouvelle fonction de u et d'une variable auxiliaire b, en faisant appel au théorème semi-quadratique [1, 17, 21, 28] :

$$\varphi(u) \mapsto (u-b)^2 + \psi(b) \quad \text{avec } \varphi(u) = \inf_{b \in \mathbb{R}} \left[(u-b)^2 + \psi(b) \right] \tag{3.27}$$

La valeur de b qui minimise ce développement est donnée par :

$$b_{inf} = u - \frac{1}{2}\varphi'(u) \tag{3.28}$$

La fonction ψ est définie par la relation suivante :

$$\psi(b) = \sup_{u \in \mathbb{R}} \left[\varphi(u) - (u-b)^2 \right] \tag{3.29}$$

Ce développement semi-quadratique a été introduit à l'origine par Geman & Yang [21] pour permettre d'optimiser les fonctionnelles de type $U(X)$ (cf. équ. (3.11)). La minimisation de l'énergie U est effectuée de manière alternée par rapport aux variables principales et auxiliaires, le théorème ci-dessus assurant que cela correspond bien à une minimisation utilisant la fonction φ.

Ce type de développement a été utilisé dans un contexte probabiliste, et il est à la base de l'algorithme d'échantillonnage de Geman & Yang [21]. Étant donné la fonction Φ définie par (3.7), les variables auxiliaires b sont les composantes des images auxiliaires B^x et B^y, correspondant respectivement aux gradients selon les colonnes et les lignes. On génère ainsi 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 $P(B^x, B^y \mid X)$. On définit ainsi un processus augmenté (X, B^x, B^y), dont la convergence sera étudiée au paragraphe suivant.

La forme quadratique en X obtenue lorsque B^x et B^y sont fixés permet d'avoir une distribution conditionnelle gaussienne et de diagonaliser la matrice de covariance par une FFT. Les formes quadratiques et les matrices de covariance correspondantes seront détaillées dans les paragraphes 3.2.7 et 3.2.8. L'échantillonnage consiste à effectuer le tirage dans la base de Fourier (base des vecteurs propres de la matrice de covariance) où les composantes sont indépendantes. Tous les pixels sont alors tirés en parallèle selon la loi normale $N(0,1)$, puis multipliés par la valeur propre correspondante. L'échantillon est finalement obtenu en calculant la FFT inverse.

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

On remplace donc Φ par le critère augmenté $\Phi^{\#}$ défini par :

$$\Phi^{\#}(X, B^x, B^y) = \left\| \frac{D_x X}{\delta} - B^x \right\|^2 + \left\| \frac{D_y X}{\delta} - B^y \right\|^2 + \sum_{i,j} \left(\psi(B^x_{ij}) + \psi(B^y_{ij}) \right) \quad (3.30)$$

Nous montrerons dans le paragraphe suivant que l'échantillonnage ainsi effectué ne correspond pas tout à fait aux lois (3.6) et (3.9) définies avec la fonction φ, mais à une loi où intervient une fonction $\tilde{\varphi}$, voisine de φ (même comportement, quadratique en 0 et linéaire en $+\infty$).

3.2.3 Convergence des processus augmentés

Marginale d'une loi jointe

On considère ici l'échantillonnage d'une loi $P(X)$, à l'aide d'une variable auxiliaire X'. X et X' sont des vecteurs, on considère alors le processus augmenté (X, X'). Au lieu d'échantillonner directement la loi $P(X)$ on préfère, pour des raisons pratiques, échantillonner la loi jointe $P(X, X')$, et utiliser ensuite uniquement les X sans tenir compte des X' qui ont été générés.

Cela revient alors à considérer la loi $P(X)$ comme loi marginale de la loi jointe $P(X, X')$, en écrivant $P(X) = \int_{\Omega_{X'}} P(X, X') dX'$. L'espace de configuration de la variable auxiliaire $\Omega_{X'}$ est ici $\mathbb{R}^{N_x N_y}$. Cela impose donc une condition sur le choix de la distribution jointe, puisque sa marginale doit nous permettre de retrouver exactement la loi à échantillonner $P(X)$.

Considérons un opérateur linéaire A, un paramètre ν et une loi définie par :

$$P_0(X) \propto e^{-\nu \sum_{i,j} \varphi((AX)_{ij})} \tag{3.31}$$

On est tenté, lorsqu'on cherche à échantillonner une telle loi, d'utiliser le développement semi-quadratique pour définir la variable auxiliaire et, de la même façon, l'expression de la loi jointe. Soit φ une fonction non quadratique, qui est l'une des fonctions du tableau 3.2 utilisée pour la régularisation de l'image X. Il convient alors de partir d'un développement donné :

$$\varphi^{\#}(u,b) = (u-b)^2 + \psi(b) \tag{3.32}$$

où la fonction ψ peut ou non correspondre au développement semi-quadratique donné par (3.27) et (3.29), et de définir la loi jointe de la façon suivante :

$$P(X,X') \propto e^{-\nu \sum_{i,j} \varphi^{\#}\left((AX)_{ij},X'_{ij}\right)} = e^{-\nu \sum_{i,j}\left[\left((AX)_{ij}-X'_{ij}\right)^2+\psi\left(X'_{ij}\right)\right]} \tag{3.33}$$

Soit $\tilde{\varphi}^{\nu}$ la fonction intervenant dans la marginale de cette loi, où $^{\nu}$ est un indice montrant la dépendance de cette fonction par rapport au paramètre ν.

$$P(X) = \int_{\Omega_{X'}} P(X,X')dX' \propto \prod_{i,j} e^{-\nu\tilde{\varphi}^{\nu}((AX)_{ij})} \tag{3.34}$$

D'après l'expression (3.33), les X'_{ij} sont indépendants, alors on a :

$$P(X) \propto \prod_{i,j} \int_{-\infty}^{+\infty} e^{-\nu\left[\left((AX)_{ij}-X'_{ij}\right)^2+\psi\left(X'_{ij}\right)\right]} dX'_{ij} \tag{3.35}$$

ce qui permet de calculer $\tilde{\varphi}^{\nu}$ (on pose $u=(AX)_{ij}$ et $b = X'_{ij}$) :

$$e^{-\nu\tilde{\varphi}^{\nu}(u)} = \int_{-\infty}^{+\infty} e^{-\nu\left[(u-b)^2+\psi(b)\right]} db \tag{3.36}$$

La fonction $\tilde{\varphi}^{\nu}$ ainsi définie dépend de la valeur du paramètre ν, et n'a aucune raison d'être égale à φ définie par (3.27).

Exemple : la fonction « hypersurfaces »

On s'intéresse à la fonction φ, que nous prenons ici égale à la fonction « hypersurfaces » (HS) : $\varphi(u) = 2\sqrt{1+u^2}-2$, illustrée sur la figure 3.1, et définie aussi dans le tableau 3.2. Il est possible de définir une fonction ψ^{ν} pour chaque valeur de ν, qui est, à une constante près, une approximation analytique de la fonction ψ donnée par l'équation (3.29), correspondant au développement semi-quadratique (3.27).

$$\psi^\nu(b) = |b| + \sqrt{1+b^2} - 1 + \epsilon^\nu \quad \text{avec } \epsilon^\nu = \int_{-\infty}^{+\infty} e^{-\nu(b^2+|b|+\sqrt{1+b^2}-1)} db \qquad (3.37)$$

Les fonctions $\tilde{\varphi}^\nu$ correspondantes sont données par intégration numérique de (3.36). La constante ϵ^ν permet de s'assurer que $\tilde{\varphi}^\nu(0) = 0$. Ces fonctions sont illustrées sur la figure 3.3. On constate que lorsque ν est grand, ces fonctions tendent vers une fonction-limite qui n'est autre que la fonction φ.

En utilisant la loi jointe (3.33), on génère donc des échantillons relatifs à la loi (3.31) avec $\varphi = \tilde{\varphi}^\nu$, qui est une approximation de la fonction φ. Pour de grandes valeurs du paramètre de régularisation ν (que l'on identifie ici à λ), on se rapproche de la fonction φ. Dans la pratique, la valeur de ce paramètre n'est pas forcément grand devant 1 et il faut alors tenir compte de la forme exacte des fonctions $\tilde{\varphi}^\nu$ lors de la déconvolution. Ceci peut être réalisé en calculant au préalable $\tilde{\varphi}^\nu(u)$ pour des valeurs discrètes de ν et u et en effectuant ensuite une interpolation, car u et ν sont des réels.

Mais ce n'est pas le seul moyen d'effectuer la restauration et l'estimation par rapport à la même fonction φ. Il existe une méthode plus simple, rapide et qui donne de très bons résultats. Elle est décrite ci-dessous.

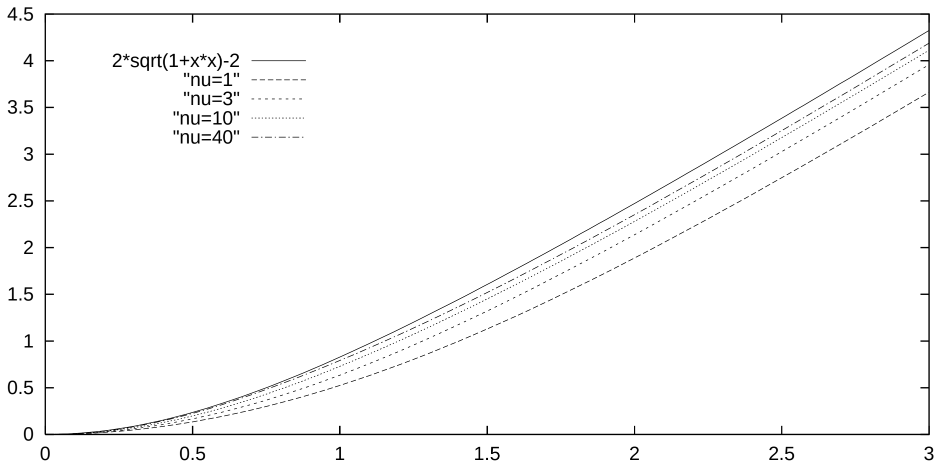

FIG. 3.3 – *Fonctions $\tilde{\varphi}^\nu(u)$ pour différentes valeurs de ν (en pointillés), et fonction limite $\varphi(u)$ (trait plein).*

Approximation par un mélange gaussien

Nous pouvons calculer une très bonne approximation de la distribution $p(u) \propto e^{-\lambda\varphi(u)}$, au moyen d'un *mélange gaussien*. Il s'agit d'exprimer $u \sim p(u)$ comme le mélange pondéré de plusieurs variables gaussiennes, de même variance, et de moyenne différente. Pour une fonction φ telle que la fonction HS, cette approximation est très satisfaisante (voir simulation figure 3.5). Ce type d'approximation est connu dans le domaine des réseaux de neurones : toute fonction continue, bornée et intégrable, peut être décomposée en une somme pondérée de noyaux gaussiens [52].

Nous pourrions utiliser la méthode d'optimisation de Levenberg-Marquardt [50] afin d'ajuster un mélange gaussien sur la fonction $p(u)$, mais nous préférons développer une méthode plus simple, dans laquelle les gaussiennes ont toutes la même variance (correspondant à la fonction $\varphi(u) = u^2$). En effet, nous voulons exprimer $p(u)$ comme un produit de convolution d'une gaussienne $e^{-\lambda u^2}$ et d'une autre fonction.

En notant $\Delta = \sqrt{2/\lambda}$, l'approximation en $2n$ composantes gaussiennes que nous proposons est la suivante :

$$e^{-\lambda\varphi(u)} \simeq \sum_{k=0}^{n} a_k \left(e^{-\lambda(u-k\Delta)^2} + e^{-\lambda(u+k\Delta)^2} \right) \quad \text{avec} \quad \forall i, a_i > 0 \tag{3.38}$$

Le calcul des coefficients a_k est détaillé dans le paragraphe 3.2.6. L'équation (3.38) s'écrit aussi :

$$e^{-\lambda\varphi(u)} \simeq e^{-\lambda u^2} \star \sum_{k=-n}^{n} a_k \left[\boldsymbol{\delta}(u - k\Delta) + \boldsymbol{\delta}(u + k\Delta) \right] \tag{3.39}$$

où $\boldsymbol{\delta}$ est la distribution de Dirac. Nous pouvons alors écrire explicitement la convolution :

$$e^{-\lambda\varphi(u)} \simeq \int_{-\infty}^{+\infty} e^{-\lambda(u-b)^2} \left[\sum_{k=-n}^{n} a_k \, \boldsymbol{\delta}(b - k\Delta) \right] db \quad \text{avec} \quad a_{-k} = a_k \tag{3.40}$$

On pose :

$$d(b) = \sum_{k=-n}^{n} a_k \, \boldsymbol{\delta}(b - k\Delta) \tag{3.41}$$

On obtient alors l'expression suivante :

$$e^{-\lambda\varphi(u)} \simeq \int_{-\infty}^{+\infty} e^{-\lambda(u-b)^2} d(b) \, db \tag{3.42}$$

ce qui revient à écrire $p(u)$ comme la marginale de la loi jointe $p(u,b)$ définie de la manière suivante :

$$p(u,b) \propto e^{-\lambda(u-b)^2} d(b) \tag{3.43}$$

Ainsi, nous pouvons introduire une loi augmentée différente par rapport au développement semi-quadratique vu précédemment (voir équation (3.36)), et qui a l'avantage de se

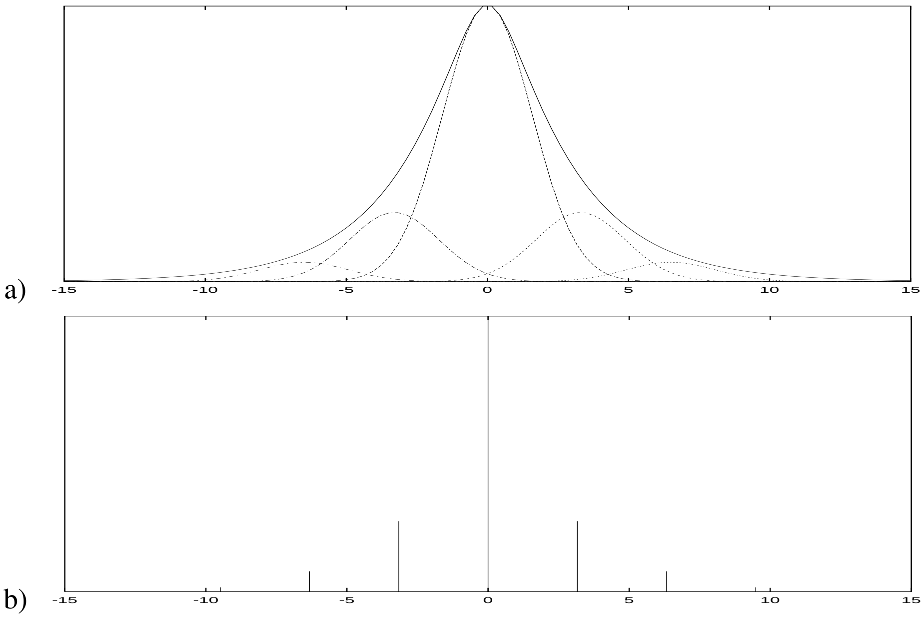

FIG. 3.4 – *Distribution* $P(x) \propto e^{-\lambda\varphi(x)}$ *avec* $\lambda = 0.2$ *: a) P(x) (trait plein) et mélange gaussien correspondant (pointillés), b) Distribution* $d(b)$ *de la variable auxiliaire discrète* b

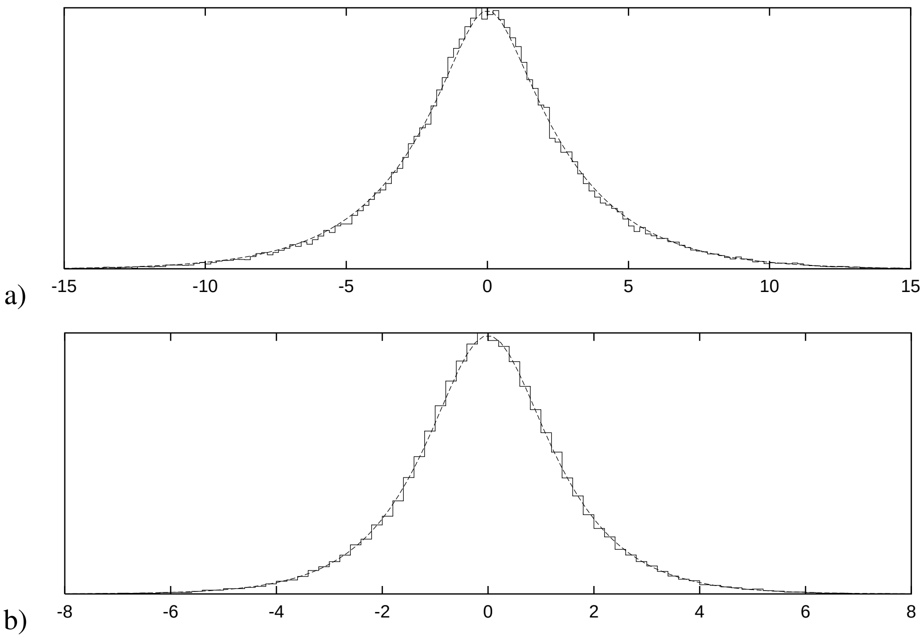

FIG. 3.5 – *Distribution* $P(x) \propto e^{-\lambda\varphi(x)}$ *(pointillés) et simulation par mélange gaussien (trait continu) - a)* $\lambda = 0.2$*, b)* $\lambda = 0.5$

ramener à une loi marginale très proche de $p(u) \propto e^{-\lambda\varphi(u)}$. La particularité de cette loi augmentée est que la loi conditionnelle de la variable auxiliaire b est discrète. Elle est donc très simple à échantillonner, d'autant plus que l'ordre du développement (n) est généralement inférieur à 10. (Voir figure 3.4 pour une illustration du mélange gaussien et de la variable associée b.)

Comme nous allons le voir dans le paragraphe suivant, le tirage s'effectue de manière alternée selon les lois conditionnelles $p(u \mid b)$ et $p(b \mid u)$, ainsi on obtient des couples (u,b) obéissant à la loi jointe. Si l'on regarde alors la distribution de u l'on s'aperçoit qu'elle correspond bien à la distribution cherchée (voir figure 3.5).

Dans le cas d'une image, nous définissons de la même manière la distribution augmentée a priori :

$$P(X,B^x,B^y \mid \lambda,k,\sigma) \propto e^{-\lambda\left(\|D_x X/\delta - B^x\|^2 + \|D_y X/\delta - B^y\|^2\right)} \prod_{i,j} d(B^x_{ij}) d(B^y_{ij}) \tag{3.44}$$

et la distribution augmentée a posteriori :

$$P(X,B^x,B^y \mid Y,\lambda,k,\sigma) \propto \tag{3.45}$$
$$e^{-\|Y - h\star X\|^2/2\sigma^2 - \lambda\left(\|D_x X/\delta - B^x\|^2 + \|D_y X/\delta - B^y\|^2\right)} \prod_{i,j} d(B^x_{ij}) d(B^y_{ij})$$

Pour l'échantillonnage de X sachant B^x et B^y, la méthode reste la même que dans [29]. D'autre part, lors du tirage de B^x et B^y sachant X, les éléments de B^x et B^y restent indépendants deux à deux. L'algorithme permettant le tirage des variables auxiliaires est détaillé au paragraphe 3.2.6.

On peut remarquer ici que malgré la forme des distributions ci-dessus, les lois marginales de B^x et B^y ne permettent pas d'avoir, comme les lois conditionnelles, des éléments indépendants deux à deux. Pour échantillonner les lois ci-dessus, il reste indispensable d'effectuer un tirage alterné, comme décrit ci-dessous.

Convergence de l'échantillonnage alterné

Entre deux échantillons successifs, les transitions ne s'effectuent pas directement de X^i à X^j, comme dans un échantillonneur classique, mais de X à X' et de X' à X. Une condition suffisante de convergence est, comme on l'a vu précédemment, la possibilité de passer d'un état à l'autre (par l'intermédiaire d'une image auxiliaire X').

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

$$P(X,X').P(\bar{X} \mid X') = P(\bar{X},X').P(X \mid X') \quad \forall (X,\bar{X}) \in \Omega_X^2 \tag{3.46}$$
$$P(X',X).P(\bar{X}' \mid X) = P(\bar{X}',X).P(X' \mid X) \quad \forall (X',\bar{X}') \in \Omega_X^2 \tag{3.47}$$

Il est donc possible de générer (X,X') selon la loi jointe $P(X,X')$ par un échantillonnage alterné, en considérant uniquement les probabilités conditionnelles d'une variable par rapport à l'autre. Cela permet finalement d'échantillonner une loi de type $P(X \mid Y,\lambda,k,\sigma)$ (loi a posteriori (3.9)) en prenant la marginale de la loi $P(X,B^x,B^y \mid Y,\lambda,k,\sigma)$ qui est plus simple à échantillonner que la loi originale. Cela permet ainsi de s'affranchir du problème de la non-linéarité soulevé par la fonction φ.

Nous introduisons donc finalement deux variables auxiliaires B^x et B^y. Ces variables correspondent à la linéarisation de φ comme nous l'avons vu précédemment. Il faut donc considérer la loi jointe a posteriori de X, B^x et B^y, résultant de l'expression (3.9) et du développement semi-quadratique (3.30) :

$$P(X,B^x,B^y \mid Y,\lambda,k,\sigma) \propto e^{-\|Y-h\star X\|^2/2\sigma^2+\lambda\Phi^{\#}(X,B^x,B^y)} \tag{3.48}$$

Pour la loi a posteriori, on alterne les étapes suivantes :

- $X \sim P(X \mid B^x,B^y,Y,\lambda,k,\sigma)$
- $(B^x,B^y) \sim P(B^x,B^y \mid X,Y,\lambda,k,\sigma)$ (les éléments de B^x et B^y sont indépendants)

La technique reste la même pour l'échantillonnage de la loi a priori.

Nous déterminerons, au chapitre suivant, le nombre d'itérations nécessaires à l'initialisation de l'échantillonneur (période de chauffage permettant d'atteindre la distribution d'équilibre désirée). Dans le cas de l'échantillonnage alterné de X, B^x et B^y qui a été étudié dans [29, 31], 5 itérations suffisent.

On peut trouver dans [10] 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 (3.26) permettant de passer d'un échantillon à un autre (comme on le fait dans l'échantillonneur de Gibbs ou la dynamique de Metropolis).

D'autres auteurs 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 une fonction scalaire (par exemple l'énergie de régularisation Φ) des échantillons en fonction du nombre d'itérations, et de voir à partir de combien d'itérations cette énergie se « stabilise ». Elle est justifiée par le fait que lors de l'estimation des paramètres, nous devons calculer l'espérance $E[\Phi]$, et il est nécessaire d'attendre la stabilisation de la chaîne afin de les calculer avec la précision souhaitée.

Durant la phase d'initialisation, tant que l'équilibre n'est pas atteint, l'énergie 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. Pour réduire d'un facteur $\sqrt{n}$ les fluctuations, il est possible de calculer n chaînes en parallèle et effectuer des moyennes. Mais ceci n'est qu'une étude *préalable*, permettant de fixer une fois pour toutes le nombre d'itérations de chauffage nécessaires à la convergence. Pas question de procéder ainsi durant l'estimation des paramètres, car cela allonge considérablement le temps de calcul.

3.2.4 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 $[0,255]$;
– image constante, chaque pixel ayant une valeur fixée, en général 128 (pour éviter tout dépassement de dynamique) ;
– échantillon calculé pour des valeurs de paramètres proches de celles qui nous intéressent.

La convergence est très lente, en général, si les conditions initiales sont aléatoires [29]. Selon la valeur des paramètres, pour la loi conditionnelle de X, 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 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. L'équilibre est atteint très rapidement dans ce cas, alors que l'énergie décroît progressivement et lentement dans l'autre cas (convergence très lente en 10000 itérations minimum!).

3.2.5 Symétries et optimisation

Dans la pratique, pour accélérer les calculs des transformées de Fourier (FFT) nécessaires à l'échantillonnage, nous profitons des différentes propriétés de symétrie relatives aux images à traiter. Nous construisons des images X et Y de taille $N_x \times N_y$, à partir d'une zone de taille $\frac{1}{2}N_x \times \frac{1}{2}N_y$ extraite d'une image satellitaire, qui vérifient la propriété de symétrie suivante, permettant ainsi de respecter les conditions aux bords de Neuman :

$$0 \leq i < N_x, 0 \leq j < N_y \Rightarrow X_{i,j} = X_{N_x-i-1,j} = X_{i,N_y-j-1} = X_{N_x-i-1,N_y-j-1} \quad (3.49)$$

Nous avons montré dans [29] que la FFT d'une image qui respecte cette propriété est donnée, à un déphasage près, par sa transformée en cosinus (DCT). Il suffit de calculer cette DCT sur le premier quadrant de l'image, de taille $\frac{1}{2}N_x \times \frac{1}{2}N_y$, les autres quadrants étant déduits par symétrie. En notant F la FFT, on a en effet pour $0 \leq u < N_x$ et $0 \leq v < N_y$:

$$\begin{aligned} |F_{u,v}| = |F_{N_x-u,v}| = |F_{u,N_y-v}| = |F_{N_x-u,N_y-v}| = \mathrm{DCT}[X]_{u,v} \\ \arg(F_{uv}) = \pi u/N_x + \pi v/N_y \qquad \text{et} \quad F_{u,N_y/2} = F_{N_x/2,v} = 0 \end{aligned} \tag{3.50}$$

D'autre part, h est supposé vérifier une autre propriété : h est symétrique par rapport aux lignes et aux colonnes

$$0 \leq i < N_x, 0 \leq j < N_y \Rightarrow h_{i,j} = h_{N_x-i,j} = h_{i,N_y-j} = h_{N_x-i,N_y-j} \tag{3.51}$$

La FFT est alors réelle, ce qui simplifie le traitement.

Enfin, les vecteurs B^x et B^y, correspondant à des dérivées d'images vérifiant la propriété (3.49), sont antisymétriques :

$$0 \leq i < N_x, 0 \leq j < N_y \Rightarrow B^x_{i,j} = -B^x_{N_x-i-2,j} \text{ et } B^y_{i,j} = -B^y_{i,N_y-j-2} \tag{3.52}$$

Nous en tiendrons compte lors de leur échantillonnage.

3.2.6 Échantillonnage des variables auxiliaires

Les méthodes qui sont présentées dans ce paragraphe conviennent à la fois pour les lois a posteriori (3.9) et a priori (3.6).

Échantillonnage de B^x et B^y

Si l'on se réfère à la loi jointe a posteriori de X, B^x et B^y (3.45) ou à la loi jointe a priori (3.44), l'expression de la probabilité conditionnelle de B^x et B^y est la suivante :

$$P(B^x, B^y \mid X) \propto e^{-\lambda\left(\|D_xX/\delta - B^x\|^2 + \|D_yX/\delta - B^y\|^2\right)} \prod_{i,j} d(B^x_{ij}) d(B^y_{ij}) \tag{3.53}$$

Les variables B^x et B^y, ainsi que toutes leurs composantes, sont indépendantes, comme on le voit dans l'expression suivante :

$$P(B^x, B^y \mid X) \propto \prod_{i,j} e^{-\lambda\left([D_xX]_{ij}/\delta - B^x_{ij}\right)^2} d(B^x_{ij}) e^{-\lambda\left([D_yX]_{ij}/\delta - B^y_{ij}\right)^2} d(B^y_{ij}) \tag{3.54}$$

Chaque composante b suit donc une loi différente notée $P_{u,\lambda}(b)$, déterminée par les paramètres $(D_xX)_{ij}/\delta$, $(D_yX)_{ij}/\delta$ et λ, selon la direction des gradients. D'après l'équation (3.41), on a :

$$P_{u,\lambda}(b) \propto e^{-\lambda(u-b)^2} d(b) = \sum_{k=-n}^{n} a_k e^{-\lambda(u-b)^2} \boldsymbol{\delta}(b - k\Delta) \tag{3.55}$$

$$P_{u,\lambda}(b) = \sum_{k=-n}^{n} p_k \boldsymbol{\delta}(b - k\Delta) \quad \text{où} \quad p_k = \frac{a_k e^{-\lambda(u-k\Delta)^2}}{\sum_{j=-n}^{n} a_j e^{-\lambda(u-j\Delta)^2}} \tag{3.56}$$

Il s'agit donc d'échantillonner une loi discrète, consistant à tirer une valeur dans l'ensemble $\{-n\Delta,...,n\Delta\}$ avec les probabilités respectives $\{p_{-n},...,p_n\}$. Pour obtenir les p_i, il faut donc calculer les coefficients de la décomposition a_i, les pondérer par $e^{-\lambda(u-b)^2}$, et ensuite normaliser.

Le calcul des coefficients a_i est effectué de manière itérative. Nous tenons compte de la parité $a_{-k} = a_k$.

Le premier coefficient a_0 correspond à une fonction φ quadratique. On a donc $a_0 = 1$. Ensuite on calcule la valeur en Δ du résidu $e^{-\lambda\varphi(u)} - e^{-\lambda u^2}$ (différence entre la distribution originale et son approximation), ce qui donne a_1. On itère ensuite, et on obtient la récurrence suivante, qui consiste à évaluer en $k\Delta$ le résidu à l'ordre k :

$$a_{k+1} = e^{\lambda\varphi(k\Delta)} - \sum_{j=-k}^{k} a_j e^{-\lambda(k\Delta - j\Delta)^2} \tag{3.57}$$

On peut simplifier en remplaçant Δ par $\sqrt{2/\lambda}$. On a alors :

$$a_0 = 1, \quad a_{k+1} = e^{\lambda\varphi(k\Delta)} - \sum_{j=-k}^{k} a_j e^{-2\lambda(k-j)^2} \quad \text{et} \quad a_{-k} = a_k \tag{3.58}$$

Ces coefficients sont générés avant l'échantillonnage, pour une valeur donnée de λ. Durant l'échantillonnage, pour chaque élément de B^x et B^y, on effectue une pondération tenant compte du gradient local u, et ensuite on effectue le tirage.

On simule B^x_{ij} et B^y_{ij} suivant $\tilde{P}_{u,\lambda}$:

$$(B^x)^{n+1}_{ij} \sim P_{u,\lambda}(b) \propto \sum_{k=-n}^{n} a_k e^{-\lambda(u-b)^2} \boldsymbol{\delta}(b - k\Delta) \quad \text{avec} \quad u = (D_x X)_{ij}/\delta \tag{3.59}$$

$$(B^y)^{n+1}_{ij} \sim P_{u,\lambda}(b) \propto \sum_{k=-n}^{n} a_k e^{-\lambda(u-b)^2} \boldsymbol{\delta}(b - k\Delta) \quad \text{avec} \quad u = (D_y X)_{ij}/\delta \tag{3.60}$$

Nous devons cependant respecter l'antisymétrie (3.52). Cela signifie que seul le premier quadrant de B^x ou B^y est échantillonné ; nous utilisons alors la propriété (3.52) pour déduire les autres variables.

3.2.7 Échantillonnage de la loi a posteriori

Nous utilisons ici les résultats que nous avons décrits dans [29, 31] permettant d'échantillonner la loi a posteriori de X sachant les autres variables, par une méthode introduisant les deux variables auxiliaires B^x et B^y.

Nous nous intéressons ici à la loi conditionnelle de X :

$$P(X \mid Y,B^x,B^y,h) \propto e^{-\|Y-h\star X\|^2/2\sigma^2-\lambda\left[\|D_xX/\delta-B^x\|^2+\|D_yX/\delta-B^y\|^2\right]} \tag{3.61}$$

qui est une loi gaussienne, et nous notons Σ^X_{post} sa matrice de covariance et X^0 sa moyenne, qui la définissent entièrement. Nous avons alors :

$$P(X \mid Y,B^x,B^y,h) \propto e^{-\frac{1}{2}(X-X^0)^t\left(\Sigma^X_{post}\right)^{-1}(X-X^0)} \tag{3.62}$$

où Σ^X_{post} et X^0 sont calculés en utilisant (3.30) :

$$(\Sigma^X_{post})^{-1} = H^tH/\sigma^2 + 2\frac{\lambda}{\delta^2}\left(D_x^tD_x + D_y^tD_y\right) \tag{3.63}$$

$$X^0 = \Sigma^X_{post}\left[H^tY/\sigma^2 + 2\frac{\lambda}{\delta}\left(D_x^tB^x + D_y^tB^y\right)\right] \tag{3.64}$$

Nous profitons ici du fait que les opérateurs H, D_x et D_y intervenant dans l'expression (3.63) sont circulants par blocs, cela permet d'effectuer une diagonalisation par FFT, comme nous l'avons vu pour l'échantillonnage de h.

La modification que nous avons apportée à la méthode présentée dans [21] par Geman et Yang consiste à employer une transformée en cosinus rapide au lieu d'une FFT. 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. C'est pourquoi nous imposons des conditions aux bords *symétriques*. Pour cela, les images traitées, de taille $N_x \times N_y$, doivent être symétriques conformément à l'équation (3.49). On utilise alors une DCT au lieu de la FFT, comme nous l'avons vu précédemment. L'échantillonnage est réalisé uniquement sur le premier quadrant de X, et ensuite le résultat est symétrisé. Ceci est utilisé aussi bien pour X et Y que pour $D_x^tB^x + D_y^tB^y$, puisque cette expression obéit à la même symétrie (3.49).

D'autre part, l'algorithme original travaillait avec des valeurs complexes ; nous utilisons des valeurs réelles étant donné les symétries (3.49) et (3.51).

Nous notons W^X_{post} la FFT de Σ^X_{post} :

$$W^X_{post} = \left(|\mathcal{F}[h]|^2/\sigma^2 + 2\frac{\lambda}{\delta^2}(|\mathcal{F}[d_x]|^2 + |\mathcal{F}[d_y]|^2)\right)^{-1} \tag{3.65}$$

La FFT de la moyenne X^0 s'écrit :

$$\mathcal{F}[X^0] = W^X_{post}\left(\mathcal{F}[h]^*\mathcal{F}[Y]/\sigma^2 + 2\frac{\lambda}{\delta}\left(\mathcal{F}[D_x^tB^x + D_y^tB_y]\right)\right) \tag{3.66}$$

Si l'on note R l'image dont les pixels sont des variables normales $N(0,1)$, alors X est donné par :

$$X = X^0 + \mathcal{F}^{-1}\left[\sqrt{W^X_{post}}\,R\right] \tag{3.67}$$

La formule permettant d'obtenir X^{n+1} à partir de h^n et $(B^x)^n$,$(B^y)^n$ est finalement :

$$X^{n+1} = \mathcal{F}^{-1}\left[W^X_{post}\left(\mathcal{F}[h^n]^*\mathcal{F}[Y]/\sigma^2 + 2\frac{\lambda}{\delta}\left(\mathcal{F}\left[D^t_x(B^x)^n + D^t_y(B_y)^n\right]\right)\right) + \sqrt{W^X_{post}}\,R\right] \tag{3.68}$$

Ainsi, tous les pixels du premier quadrant de R étant indépendants, ils sont tirés en parallèle, et le calcul d'un échantillon X sachant h, B^x et B^y est rapide, puisqu'il ne nécessite qu'une FFT (taille $N_x \times N_y$) et deux DCT (taille $\frac{1}{2}N_x \times \frac{1}{2}N_y$). Dans le cas où h est paramétré, sa FFT est calculée au préalable (avant l'échantillonnage) et il ne faut plus compter alors que deux DCT par échantillon.

Algorithme d'échantillonnage a posteriori

La loi à échantillonner est la loi a posteriori $P(X \mid Y,\lambda,k,\sigma)$.

ALGORITHME 3.2.1 (ÉCHANTILLONNAGE X A POSTERIORI)
L'image Y est observée, et les paramètres λ, k et σ sont fixés.

- *Déconvolution de Y par calcul du MAP pour obtenir $\hat{X}$ (paragraphe 3.4, algorithme 3.4.1)*
- *Initialisation de X par la moyenne de la distribution de X : $X^0 = \hat{X}$*
- *Répéter M_i fois (initialisation de la chaîne X,B^x,B^y) puis M fois (nombre d'échantillons) :*
  1. *Tirage de $(B^x)^{n+1}$ et $(B^y)^{n+1}$ sachant X^n en utilisant (3.59) et (3.60)*
  2. *Tirage de X^{n+1} sachant $(B^x)^{n+1}$, $(B^y)^{n+1}$ et h^n, en utilisant (3.68) et (3.65)*

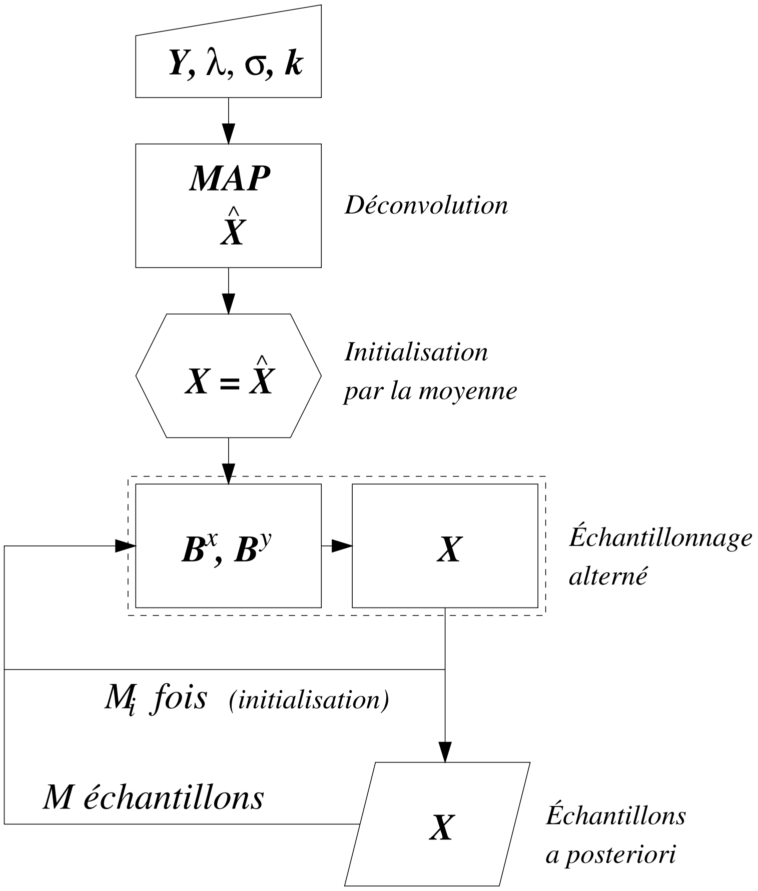

FIG. 3.6 – *Algorithme d'échantillonnage de la loi a posteriori* $P(X \mid Y,\lambda,k,\sigma)$

3.2.8 Échantillonnage de la loi a priori

Il est possible d'utiliser ici la même méthode que pour la distribution a posteriori, c'est-à-dire de diagonaliser $\Phi^{\#}(X,B^x,B^y)$, le développement semi-quadratique de $\Phi(X)$, et d'utiliser une méthode alternée de type Geman et Yang [21].

La loi a priori de X sachant B^x et B^y s'écrit alors :

$$P(X \mid B^x,B^y) \propto e^{-\lambda\left[\|D_x X/\delta - B^x\|^2 + \|D_y X/\delta - B^y\|^2\right]} \tag{3.69}$$

qui est une loi gaussienne, et nous notons Σ^X_{pri} sa matrice de covariance et X^0 sa moyenne. Nous avons alors :

$$P(X \mid B^x,B^y) \propto e^{-\frac{1}{2}(X-X^0)^t\left(\Sigma^X_{pri}\right)^{-1}(X-X^0)} \tag{3.70}$$

où Σ^X_{pri} et X^0 sont calculées en utilisant (3.30) :

$$(\Sigma^X_{pri})^{-1} = 2\frac{\lambda}{\delta^2}\left(D_x^t D_x + D_y^t D_y\right) \tag{3.71}$$

$$X^0 = \Sigma^X_{pri} 2\frac{\lambda}{\delta}\left(D_x^t B^x + D_y^t B^y\right) \tag{3.72}$$

Nous effectuons une diagonalisation par une FFT, comme nous l'avons vu précédemment. Nous notons W^X_{pri} la FFT de Σ^X_{pri} :

$$W^X_{pri} = \left(2\frac{\lambda}{\delta^2}(|\mathcal{F}[d_x]|^2 + |\mathcal{F}[d_y]|^2)\right)^{-1} \tag{3.73}$$

La FFT de la moyenne X^0 s'écrit :

$$\mathcal{F}[X^0] = 2W^X_{pri}\frac{\lambda}{\delta}\left(\mathcal{F}[D_x^t B^x + D_y^t B_y]\right) \tag{3.74}$$

Si l'on note R l'image dont les pixels sont des variables normales $N(0,1)$, alors X est donné par :

$$X = X^0 + \mathcal{F}^{-1}\left[\sqrt{W^X_{pri}}\,R\right] \tag{3.75}$$

La formule permettant d'obtenir X^{n+1} à partir de $(B^x)^n,(B^y)^n$ est finalement :

$$X^{n+1} = \mathcal{F}^{-1}\left[2W^X_{pri}\frac{\lambda}{\delta}\left(\mathcal{F}\left[D_x^t(B^x)^n + D_y^t(B_y)^n\right]\right) + \sqrt{W^X_{pri}}\,R\right] \tag{3.76}$$

Comme nous l'avons vu pour la distribution a posteriori, l'échantillonnage n'est effectué que sur le premier quadrant (en utilisant des DCT), le reste étant déduit par symétrie.

Ainsi, tous les pixels du premier quadrant de R étant indépendants, ils sont tirés en parallèle, et le calcul d'un échantillon X sachant B^x et B^y est rapide, puisqu'il ne nécessite que deux DCT (de taille $\frac{1}{2}N_x \times \frac{1}{2}N_y$), W^h_{pri} étant calculée une fois au début.

Algorithme d'échantillonnage a priori

Pour échantillonner X selon $P(X \mid \lambda)$, nous utilisons la loi jointe $P(X,B^x,B^y \mid \lambda)$.

ALGORITHME 3.2.2 (ÉCHANTILLONNAGE DE X A PRIORI)
Le paramètre λ est fixé.

- *Initialisation de X par la moyenne de la distribution : $X^0 = 0$.*
- *Calculer W_{pri}^X en utilisant (3.73)*
- *Répéter M_i fois au début (initialisation de la chaîne), puis M fois (nombre d'échantillons) :*
  1. *Tirage de $(B^x)^{n+1}$ et $(B^y)^{n+1}$ sachant X^n en utilisant (3.59) et (3.60)*
  2. *Tirage de X^{n+1} sachant $(B^x)^{n+1}$, $(B^y)^{n+1}$ et h^n, en utilisant (3.76) et W_{pri}^X*

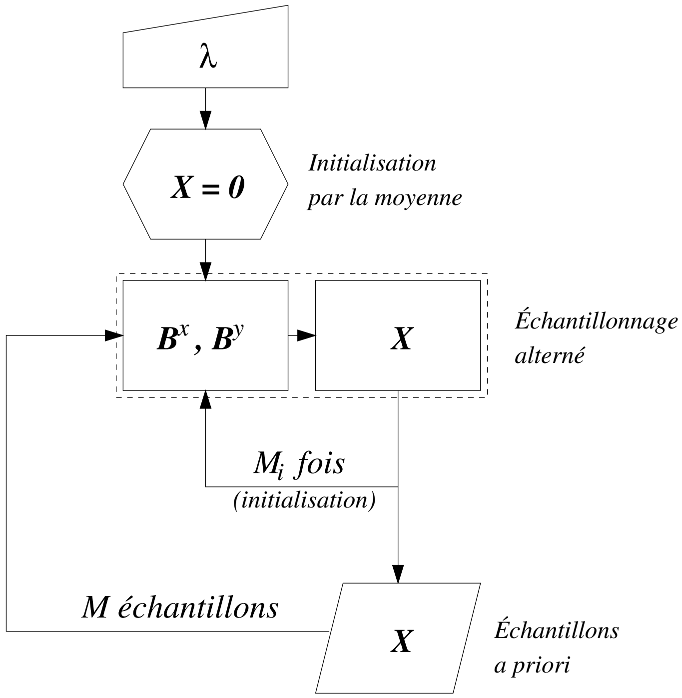

FIG. 3.7 – *Algorithme d'échantillonnage de la loi a priori $P(X \mid \lambda)$*

3.3 Estimation des paramètres

Soit θ le vecteur des paramètres tel que $\theta = (\lambda,k_1,...k_m,\sigma)$. Nous rappelons que la valeur optimale, au sens du maximum de vraisemblance, est la solution de l'équation stochastique suivante :

$$E_{Y,\hat{\theta}}[f(X)] = E_{\hat{\theta}}[f(X)] \tag{3.77}$$

ce qui correspond aux équations (3.22), (3.23) et (3.24), $f(X)$ étant une fonction des échantillons X, obtenus par les algorithmes des paragraphes précédents.

Ces espérances sont estimées par la moyenne empirique suivante, où M est le nombre d'échantillons X tirés avec la loi $P(X)$:

$$E_{(X)\sim P(X)}[f(X)] \simeq \frac{1}{M} \sum_{(X^k)\sim P(X)} f(X^k) \tag{3.78}$$

Descente de gradient

Pour maximiser la vraisemblance $P(Y \mid \theta)$, une méthode classique consiste à minimiser $L(\theta)$ définie par l'équation (3.16) en utilisant un algorithme de descente de type Newton-Raphson [29, 31]. Nous notons ∇_{θ_i} la dérivée $\partial L/\partial\theta_i$, et G_{θ_i} la dérivée seconde $\partial^2 L/\partial{\theta_i}^2$ (diagonale du Hessien). On a alors :

$$\nabla_{\theta_i} = E_{Y,\theta}\left[\frac{\partial U(X,h)}{\partial\theta_i}\right] - E_{\theta_i}\left[\frac{\partial U(X,h)}{\partial\theta_i}\right] \tag{3.79}$$

La dérivée seconde est, quant à elle, calculée numériquement :

$$G_{\theta_i} \simeq \frac{\nabla_{\theta_i+\Delta\theta_i} - \nabla_{\theta_i}}{\Delta\theta_i} \tag{3.80}$$

L'itération permettant de passer de θ^n à θ^{n+1} est donc :

$$\theta_i^{n+1} = \theta_i^n - a\,\frac{\nabla_{\theta_i^n}}{G_{\theta_i^n}} \simeq \theta_i^n - a\,\Delta\theta_i\,\frac{\nabla_{\theta_i^n}}{\nabla_{\theta_i^n+\Delta\theta_i} - \nabla_{\theta_i^n}} \tag{3.81}$$

où a est un facteur compris entre 0 et 1, permettant de stabiliser l'algorithme.

On note $A(X,h) = \|Y - h \star X\|^2$.

L'équation (3.79) s'écrit, pour chaque paramètre :

$$\nabla_\lambda = E_{Y,\lambda,k,\sigma}[\Phi(X)] - E_\lambda[\Phi(X)] \tag{3.82}$$

$$\nabla_k = \frac{1}{\sigma^2} E_{Y,\lambda,k,\sigma}[(\frac{\partial h}{\partial k} \star X)^t (Y - h \star X)] \tag{3.83}$$

$$\nabla_\sigma = \frac{1}{\sigma^3} E_{Y,\lambda,k,\sigma}[A(X,h)] - \frac{N_x N_y}{\sigma} \tag{3.84}$$

En pratique, pour chaque valeur des paramètres, deux chaînes de Markov de M éléments chacune sont générées, suivant la loi a posteriori (3.9) et la loi a priori (3.6) : $\{X^k_{post}\}$ et $\{X^k_{pri}\}$, en utilisant les algorithmes 3.2.1 et 3.2.2. Les gradients de L par rapport aux paramètres sont estimés de façon suivante dans le cas où h n'est pas paramétré :

$$\nabla_\lambda \simeq \frac{1}{M}\left[\sum_{k=1}^{M}\Phi(X^k_{post}) - \sum_k \Phi(X^k_{pri})\right] \tag{3.85}$$

$$\nabla_k \simeq \frac{1}{M}\frac{1}{\sigma^2}\left[\sum_{k=1}^{M}\left(\frac{\partial h}{\partial k}\star X^k_{post}\right)^t (Y - h\star X^k_{post})\right] \tag{3.86}$$

$$\nabla_\sigma \simeq -\left[\frac{1}{\sigma^3}\frac{1}{M}\sum_{k=1}^{M} Q(h^k_{post})\right] + \frac{N_x N_y}{\sigma} \tag{3.87}$$

Algorithme de type EM

Pour accélérer l'algorithme d'optimisation de la log-vraisemblance, il est possible d'estimer les paramètres de régularisation en utilisant une technique différente, basée sur les méthodes de type EM (Expectation Maximization). Le principe de l'EM consiste à maximiser l'espérance de la probabilité jointe de X et Y, prise selon la loi a posteriori, :

$$\hat{\theta} = \arg\max_\theta E_{Y,\theta}[\log P(X,Y \mid \theta)] \tag{3.88}$$

En pratique, la recherche de paramètres optimaux selon ce critère est très difficile, c'est pourquoi nous utilisons une forme itérative, plus simple à mettre en œuvre, et pour laquelle des résultats théoriques existent [3, 54].

Si l'on suppose que θ^n est connu à l'itération n, cela consiste à chercher θ^{n+1} tel que :

$$\hat{\theta}^{n+1} = \arg\max_\theta E_{Y,\theta^n}[\log P(X,Y \mid \theta)] \tag{3.89}$$

En dérivant cette expression par rapport à chaque θ_i, on obtient finalement :

$$E_{\theta^{n+1}}\left[\frac{\partial U(X)}{\partial \theta_i}\right] = E_{Y,\theta^n}\left[\frac{\partial U(X)}{\partial \theta_i}\right] \tag{3.90}$$

ce qui s'écrit, pour chaque paramètre :

$$E_{\hat{\lambda}^{n+1}}[\Phi(X)] = E_{Y,\hat{\lambda}^n,\hat{k}^n,\hat{\sigma}^n}[\Phi(X)] \tag{3.91}$$

$$\frac{N_x N_y}{\sigma^{n+1}} = \frac{1}{(\sigma^n)^3} E_{Y,\hat{\lambda}^n,\hat{k}^n,\hat{\sigma}^n}[A(X)] \tag{3.92}$$

Si l'on se réfère aux équations (3.22) et (3.23), la principale différence entre l'emploi d'une méthode de type EM et une descente de gradient de type Newton réside dans la manière de mettre à jour les paramètres. Dans le cas de l'EM, il s'agit d'attribuer une nouvelle valeur aux paramètres qui correspond à maximiser la vraisemblance en données complètes, alors que l'autre méthode consiste à descendre d'un pas dans la direction du gradient, ce qui est en général moins rapide, et la stabilité est moindre.

En ce qui concerne le vecteur de paramètres k, son estimation n'est pas adaptée à un algorithme de type EM, étant donné que k n'intervient que dans la distribution a posteriori. L'estimation sera donc effectuée par une méthode de descente de gradient.

On distingue donc deux étapes dans une méthode de type EM : le calcul de l'espérance a posteriori avec θ^n, puis la maximisation, c'est-à-dire la recherche des nouvelles valeurs des paramètres θ_i^{n+1} telles que l'espérance a priori soit égale à l'espérance a posteriori, qui est une constante calculée lors de l'étape précédente.

Supposons que l'on connaisse, comme pour σ, l'expression de l'espérance a priori en fonction de λ, que l'on notera $\mathcal{E}_\lambda(\lambda)$. L'étape de maximisation devient alors très rapide, en utilisant la fonctions réciproque $\mathcal{E}_\lambda^{-1}$:

$$\lambda^{n+1} = \mathcal{E}_\lambda^{-1}\left(E_{Y,\hat{\lambda}^n,\hat{\mu}^n,\hat{\sigma}^n}[\Phi(X)]\right) \tag{3.93}$$

$$\sigma^{n+1} = \frac{N_x N_y}{(\sigma^n)^3}\left(E_{Y,\hat{\lambda}^n,\hat{\mu}^n,\hat{\sigma}^n}[A(X,h)]\right)^{-1} \tag{3.94}$$

Il est possible de calculer numériquement (en utilisant les échantillonneurs décrits précédemment), pour un ensemble de valeurs discrètes de λ, l'espérance $\mathcal{E}_\lambda(\lambda)$. Ensuite, la réciproque est déterminée puis stockée dans un tableau, qui pourra être utilisé pour optimiser l'estimation, sans qu'il soit nécessaire de recalculer les espérances par une méthode MCMC.

La fonction $\mathcal{E}_\lambda(\lambda)$ et sa réciproque sont illustrées par les figures 3.8 et 3.9. Pour estimer le paramètre λ, nous utiliserons la variante de la méthode EM présentée précédemment, avec l'équation (3.93). Nous pouvons en effet déterminer expérimentalement une approximation analytique de la fonction réciproque $\mathcal{E}_\lambda^{-1}$:

$$\mathcal{E}_\lambda^{-1}(E_\lambda) \simeq 0.6\, E_\lambda^{-0.91} \tag{3.95}$$

3.3.1 Problèmes liés à l'estimation de σ

L'estimation de σ par maximum de vraisemblance s'est avérée impossible, dans la mesure où la valeur de σ qui maximise la vraisemblance est $\sigma \simeq 0$. Des tests ont été effectués avec une fonction φ quadratique, ce qui a permis d'obtenir une expression déterministe du gradient de la log-vraisemblance.

D'autre part, dans ce cas particulier, il est également possible d'évaluer directement la vraisemblance en effectuant des sommes dans le domaine fréquentiel, puisque l'on a une forme quadratique qui est diagonalisée par la transformée de Fourier (FFT). Nous avons pu vérifier que la vraisemblance augmente lorsque σ diminue, quelle que soit la valeur de λ

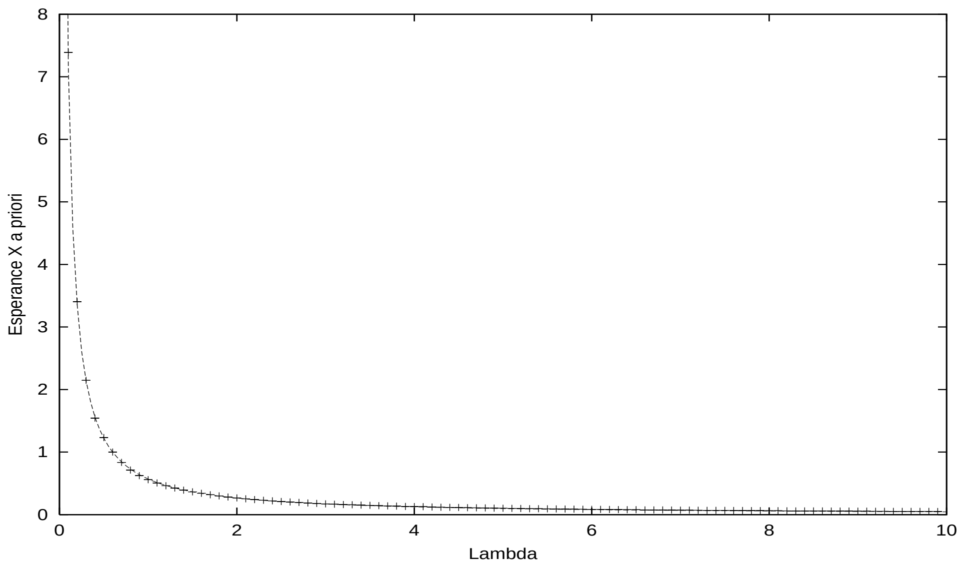

FIG. 3.8 – *Espérance $\mathcal{E}_\lambda(\lambda)$ pour la loi a priori $P(X \mid \lambda)$ en fonction de λ*

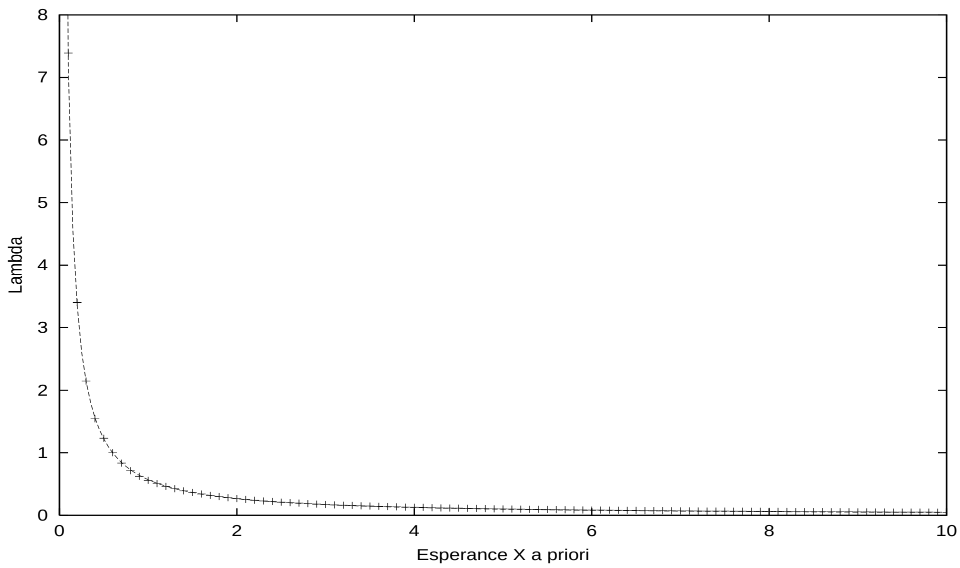

FIG. 3.9 – *Réciproque de l'espérance $\mathcal{E}_\lambda^{-1}(E)$ pour la loi a priori $P(X \mid \lambda)$ en fonction de l'espérance E*

ou de k. Cela est vrai aussi bien pour de faibles valeurs de σ^0 que pour de plus grandes valeurs (10 à 20), où l'on note σ^0 la vraie valeur de σ (bruit ajouté à l'image floue lors de la simulation).

Lorsque l'image originale et le noyau sont connus, l'estimateur du maximum de vraisemblance donne de bons résultats pour l'estimation du bruit. Mais dans le cas où ni le noyau, ni l'image ne sont connus, cette méthode ne fonctionne plus, et nous devons utiliser une autre technique afin de déterminer les paramètres du bruit.

3.3.2 Estimation du bruit : méthode proposée

Principe de la méthode

Le modèle de bruit additif, blanc et gaussien relatif aux simulations SPOT 5 est :

$$\text{var}(N_{ij}) = a + b\, Y^0_{ij} \tag{3.96}$$

où Y^0 désigne l'image floue non bruitée, telle que $Y^0 = h \star \mathcal{X}$.

Nous supposerons dans la suite que pour estimer b, on considère que $Y^0 \simeq Y$. La variance du bruit en chaque pixel dépend alors de la valeur, en ce pixel, de l'image observée Y. En effet, le bruit est suffisamment faible pour que l'on puisse faire ce genre d'approximation :

$$\text{var}(N_{ij}) \simeq a + b\, Y_{ij} \tag{3.97}$$

Lors de la déconvolution, il n'a été constaté aucune amélioration sensible des résultats lorsque l'on prend en compte le fait que le bruit dépende de l'intensité. C'est pourquoi nous utilisons, dans nos algorithmes, uniquement l'écart-type σ équivalent à un bruit blanc gaussien de variance constante, qui est lié à a et b par la relation :

$$\sigma^2 \simeq a + b \frac{1}{N_x N_y} \sum_{i,j} Y_{ij} \tag{3.98}$$

ce qui équivaut à prendre la variance moyenne sur toute l'image observée Y.

Nous proposons ici une méthode générale permettant d'estimer les paramètres a et b à partir de l'image dégradée Y. L'écart-type σ peut être retrouvé soit à partir de ces paramètres en utilisant la relation ci-dessus, soit directement à partir du spectre de Y.

Nous supposons, dans la suite de ce rapport, que l'image Y est floue, et donc que les plus hautes fréquences de son spectre contiennent uniquement du bruit. D'autre part, nous supposons également que les conditions aux bords sont symétriques (conditions de Neuman), ce qui permet d'employer une DCT au lieu d'une FFT. En effet, l'utilisation d'une FFT conduit généralement à l'apparition d'artefacts dans le spectre, ceci étant dû aux conditions aux bords symétriques, et non périodiques, qui sont appliquées lors de la convolution de l'image originale par le noyau h lors de la simulation de Y. L'emploi d'une DCT nous a permis de réduire sensiblement ces artefacts, et donc d'obtenir une estimation des paramètres du bruit plus précise.

Le principe de la méthode proposée est d'utiliser un filtre passe-bande pour isoler le bruit qui se trouve dans les plus hautes fréquences de la DCT. Les basses fréquences, au-dessous de ν_0, sont éliminées, car elles sont susceptibles de contenir du signal en plus du bruit. D'autre part, le bruit inter-colonne présent sur les simulations SPOT 5 se traduit par l'apparition de pics dans les plus hautes fréquences, ce bruit n'est pas gaussien et ne correspond pas au modèle (3.97), nous ne chercherons donc pas à l'estimer. Ce bruit structuré est donc filtré de la même façon que les basses fréquences. En résumé, sont mises à zéro toutes les fréquences en-dehors de la bande $[\nu_0,\nu_1]$ (cf. figure 3.10).

Le choix de ν_0 et ν_1 est déterminé par la nature du bruit présent dans les images, ainsi que par l'importance du flou. Pour être certain que la bande sélectionnée ne contient que du bruit, nous prenons $\nu_0 > \frac{1}{2}$. D'autre part, s'il existe du bruit structuré en lignes ou en colonnes, il se traduit par des pics dans les fréquences spatiales les plus élevées. Ce bruit n'est pas estimé puisqu'il ne correspond pas aux hypothèses de travail, il est donc filtré et l'on choisit $\nu_1 < \frac{\sqrt{2}}{2}$.

Soit $n_{\Delta\nu}$ le nombre de pixels de la DCT dont la fréquence ν vérifie $\nu_0 \leq \nu \leq \nu_1$. On obtient alors, pour la variance moyenne :

$$\sigma^2 \simeq \frac{1}{n_{\Delta\nu}} \sum_{\nu_0 \leq \nu_{ij} \leq \nu_1} \mathrm{DCT}[Y]_{ij}^2 \tag{3.99}$$

où ν_{ij} est la fréquence correspondant au pixel de coordonnées (i,j) dans l'espace de la DCT.

Le bruit, une fois isolé dans le domaine fréquentiel, est transformé par DCT inverse pour fournir, dans l'espace image, un bruit coloré permettant de remonter aux paramètres a et b. Ce bruit est noté N^f. Bien qu'il ne soit plus blanc, après avoir subi un filtre passe-bande, ses caractéristiques spatiales sont suffisantes pour permettre l'estimation du paramètre b. Nous effectuons une approximation qui consiste à dire que la variance de N^f est proportionnelle à la variance de N, le bruit original non filtré. Le coefficient de proportionnalité κ est égal à la fraction des pixels conservée lors du filtrage, $\kappa = \frac{n_{\Delta\nu}}{N_x N_y}$. Le bruit filtré est donc divisé par κ avant l'estimation des variances.

Il faut ensuite estimer la variance du bruit filtré au pixel (i,j) en fonction de la valeur Y_{ij} correspondante. Pour obtenir une meilleure précision des quantités calculées, nous séparons les valeurs des pixels en plusieurs classes notées C_k, et pour chaque classe nous calculons la variance v_k sur l'image Y, et la moyenne m_k sur le bruit filtré N^f. Cela permet de tracer la variance moyenne de chaque classe en fonction de la moyenne de cette classe, les couples (moyenne,variance) formant une droite dont les paramètres (estimés par régression linéaire) sont égaux à a et b.

Le nombre de classes doit être suffisamment élevé pour permettre de retrouver b avec précision, et les classes doivent être suffisamment peuplées afin que les variances soient estimées avec une bonne précision. En prenant 16 classes de 16 valeurs chacune (C_0 regroupe

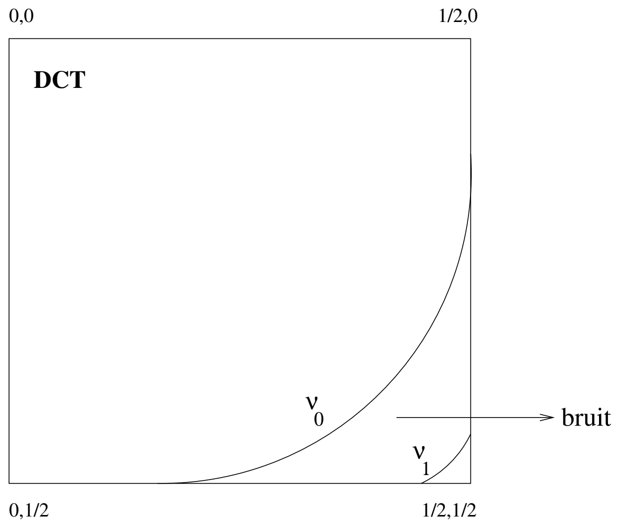

FIG. 3.10 – *Principe de la méthode d'extraction du bruit (filtre passe-bande dans la DCT)*

les pixels de valeurs entre 0 et 15, etc.), on vérifie ces deux conditions (pour des images codées sur 8 bits, ce qui est le cas des simulations SPOT 5 étudiées).

On a donc :

$$v_k \simeq a + b\, m_k \tag{3.100}$$

Si l'on note n_k le nombre de pixels de la classe C_k, les moyennes et variances peuvent être estimées de la manière suivante :

$$m_k = \frac{1}{n_k} \sum_{Y_{ij} \in C_k} Y_{ij} \quad \text{et} \quad v_k = \frac{1}{n_k - 1} \sum_{Y_{ij} \in C_k} (N^f_{ij})^2 \tag{3.101}$$

Les données ainsi obtenues, $\{m_k\}$ et $\{v_k\}$, sont pondérées par $\sqrt{n_k}$, l'inverse de l'erreur relative sur la variance pour la classe k, lors de la régression linéaire pour obtenir les paramètres optimaux a et b. Sur la figure 4.4 sont représentées les variances en fonction des moyennes, les barres d'erreur correspondant à l'erreur relative commise lors du calcul des variances, proportionnelle à $1/\sqrt{n_k}$. Les classes très peu peuplées ne permettent pas d'estimer précisément une variance. Ce sont les classes les plus peuplées (barres d'erreur les plus petites) qui déterminent la droite optimale $m_k = a + b\, v_k$.

Cette méthode a été testée sur des images floues auxquelles nous avons ajouté un bruit simulé avec différentes valeurs de a et b, et les résultats obtenus sont très satisfaisants.

ALGORITHME 3.3.1 (ESTIMATION DU BRUIT (σ,a,b))

- *DCT de l'image observée Y*
- *Filtre passe-bande conservant $\nu \in [\nu_0,\nu_1]$*
- *DCT inverse pour obtenir N^f*
- *Normalisation : division de N^f par $\kappa = \frac{n_{\Delta\nu}}{N_x N_y}$*
- *Les pixels de N^f sont regroupés en différentes classes C_k en fonction de Y (les classent forment une partition de l'espace des valeurs des pixels $\{0,1...255\}$)*
- *Calcul du nombre de pixels n_k de chaque classe*
- *Calcul des moyennes $\{m_k\}$ et des variances $\{v_k\}$ de chaque classe en utilisant les équations (3.101)*
- *Régression linéaire pondérée sur $\{m_k\}$ et $\{v_k\}$ (en utilisant (3.100))*

Il est possible d'effectuer une régression linéaire en utilisant tous les pixels : cela consiste à trouver la relation linéaire entre la valeur de chaque pixel de Y et le carré de la valeur du bruit N_f associée à ce pixel. Néanmoins, la méthode précédente est plus précise, dans la mesure où elle permet d'éliminer les classes les moins peuplées, pour lesquelles il peut y avoir un écart notable par rapport à la loi linéaire que l'on cherche à ajuster (écart lié au manque de précision sur la variance).

Autres méthodes possibles

Il est possible d'estimer les paramètres a et b par maximum de vraisemblance, à partir du bruit filtré N^f et de l'image observée Y. On suppose pour cela que N^f constitue une bonne approximation du bruit non filtré N.

On écrit alors :

$$(\hat{a},\hat{b})_{MV} = \arg\max_{a,b} P(N \mid a,b) \tag{3.102}$$

avec la loi non stationnaire pour le bruit N :

$$P(N \mid a,b) = \frac{1}{Z_{ab}} \prod_{i,j} e^{-N_{ij}^2/2(a+b\,Y_{ij})} \tag{3.103}$$

On considère, en effet, que les pixels de N sont indépendants (cela fait partie des hypothèses). La fonction de partition Z_{ab} s'écrit alors :

$$Z_{ab} = \prod_{i,j} \sqrt{2\pi(a + b\,Y_{ij})} \tag{3.104}$$

Il est alors possible de calculer les dérivées de $-\log P(N \mid a,b)$ par rapport à a et b afin de maximiser la vraisemblance par une méthode de gradient. Nous ne donnons pas ici les détails de cette technique, car cette méthode n'apporte pas d'amélioration significative des résultats par rapport à la méthode approchée du paragraphe précédent. D'autre part, la technique précédente est plus rapide dans la mesure où elle n'est pas itérative.

3.3.3 Estimation des paramètres du noyau

Pour l'estimation de k, il est possible d'utiliser le maximum de vraisemblance, comme l'ont suggéré certains auteurs [60] :

$$\hat{k} = \arg\max_k P(Y \mid \lambda,k,\sigma) \tag{3.105}$$

ce qui s'écrit aussi :

$$E_{Y,\lambda,\hat{k},\sigma}\left[\frac{1}{\sigma^2}\left(\frac{\partial h}{\partial k}\star X\right)^t (Y - h\star X)\right] = 0 \tag{3.106}$$

Principe de la méthode d'estimation

L'écart-type du bruit σ a été déterminé au préalable. Il reste donc à estimer λ (régularisation de X) et k (paramètres du noyau).

Nous rappelons ici l'expression approchée qui nous permet, à partir d'une chaîne (X^n_{post}) d'images tirées avec la loi a posteriori $P(Y \mid \lambda,k,\sigma)$, d'estimer la dérivée de la -log vraisemblance :

$$\nabla_k \simeq \frac{1}{M}\frac{1}{\sigma^2}\left[\sum_{n=1}^{M}\left(\frac{\partial h}{\partial k}\star X^n_{post}\right)^t (Y - h\star X^n_{post})\right] \tag{3.107}$$

Cette équation est à la base d'une méthode de descente de type Newton-Raphson. La dérivée seconde, nécessaire à cet algorithme, est calculée numériquement à partir de ∇_k.

Nous utilisons donc l'algorithme suivant, permettant d'estimer conjointement λ et k :

ALGORITHME 3.3.2 (ESTIMATION DE λ ET k)
σ est estimé par l'algorithme 3.3.1.
λ^0 et k^0 sont initialisés avec des valeurs réalistes, selon le type d'image utilisé.
Répéter jusqu'à la convergence :

- *Mise à jour de λ (cf. figure 3.9 pour $\mathcal{E}_\lambda^{-1}$) :*

$$\lambda^{n+1} = \mathcal{E}_\lambda^{-1}\left(E_{Y,\lambda^n,k^n,\sigma}[\Phi(X)]\right)$$

- *Mise à jour de k :*

$$k^{n+1} = k^n - \alpha\nabla_k(k)\left(\frac{\nabla_k(k+\Delta k) - \nabla_k(k)}{\Delta k}\right)^{-1}$$

où $\Delta k > 0$ et $0 < \alpha < 1$ (coefficient facilitant la convergence)

Résultats obtenus

Nous avons simulé des images floues et bruitées en utilisant des noyaux paramétrés de différentes formes et différentes valeurs de l'écart-type du bruit. Nous avons ensuite estimé les paramètres du noyau ainsi que le paramètre de régularisation λ, par maximum de vraisemblance. Cela nous a permis de vérifier que l'estimateur proposé permet de retrouver, avec une bonne précision, les paramètres du noyau h.

Des tests ont été effectués avec un noyau gaussien :

$$h_{ij} \propto e^{-i^2/2k^2 - j^2/2k^2} \tag{3.108}$$

L'écart-type du bruit est choisi entre 0.5 et 2. Une zone de taille 128×128, extraite de l'image originale de Nîmes, a été utilisée pour les tests. Les résultats sont donnés sur la figure 3.11.

Ces résultats montrent que le paramètre k est estimé correctement, malgré la présence du bruit. D'ailleurs, les cas choisis sont particulièrement difficiles, et les dégradations subies par les images satellitaires sont moindres dans le cas réel. Les valeurs de k estimées par maximum de vraisemblance permettent une restauration correcte de l'image même dans ces cas difficiles.

Nous présenterons les résultats sur l'image Nîmes (simulation SPOT 5) au chapitre suivant, d'autres tests sur d'autres simulations ont été effectués dans le cadre d'un contrat CNES [30].

	σ=0.5	σ=1	σ=2
k=1	1.0	1.1	1.1
k=2	1.9	1.8	1.7
k=3	2.8	2.6	2.4

FIG. 3.11 – *Valeurs du paramètre k estimées sur une image dégradée par différentes fonctions de flou (définies par l'équation (3.108)) et différentes valeurs de σ*

3.4 Déconvolution

3.4.1 Régularisation avec préservation des contours

Nous supposons, dans ce paragraphe, que les paramètres λ, k et σ sont fixés.

Nous rappelons que pour chercher l'image X liée à l'image observée Y par l'équation (3.1), nous devons minimiser le critère suivant par rapport à X :

$$U(X) = \|Y - h \star X\|^2/2\sigma^2 + \lambda\Phi(X) \tag{3.109}$$

Le choix d'une fonction φ non quadratique, telle que la fonction $\varphi(u) = 2\sqrt{1+u^2} - 2$, permet de filtrer le bruit tout en préservant les contours. La fonction intervenant dans la régularisation du noyau est quadratique, interdisant justement les discontinuités, puisque ce noyau est supposé lisse.

Nous utilisons l'algorithme défini dans [29, 31], qui est rapide, et dont la complexité ne dépend pas de la taille de h puisqu'il travaille dans le domaine fréquentiel.

On remplace Φ par le critère augmenté $\Phi^{\#}$, pour se ramener à une fonctionnelle quadratique par rapport à X :

$$\Phi^{\#}(X,B^x,B^y) = \left\| \frac{D_x X}{\delta} - B^x \right\|^2 + \left\| \frac{D_y X}{\delta} - B^y \right\|^2 + \sum_{i,j} \left(\psi(B^x_{ij}) + \psi(B^y_{ij}) \right) \quad (3.110)$$

en utilisant le théorème semi-quadratique (3.27). Cela définit l'énergie augmentée à minimiser par rapport à X et aux variables auxiliaires B^x et B^y :

$$U^{\#}(X,B^x,B^y) = \|Y - h \star X\|^2/2\sigma^2 + \lambda \Phi^{\#}(X,B^x,B^y) \quad (3.111)$$

Nous minimisons U de manière alternée par rapport à X, B^x et B^y.

Lorsque les variables auxiliaires B^x et B^y sont fixées, on a une forme quadratique en X :

$$(X - X^0)^t \, M_X \, (X - X^0) \quad (3.112)$$

avec M_X et X^0 donnés par :

$$M_X = H^t H/\sigma^2 + 2\frac{\lambda}{\delta^2} \left(D_x^t D_x + D_y^t D_y \right) \quad (3.113)$$

$$X^0 = M_X^{-1} \left[H^t Y/\sigma^2 + 2\frac{\lambda}{\delta} \left(D_x^t B^x + D_y^t B^y \right) \right] \quad (3.114)$$

Nous utilisons une diagonalisation par une FFT, calculée au moyen d'une DCT [29], comme pour l'échantillonnage a posteriori de X détaillé dans le paragraphe 3.2.7.

Nous notons W^X la FFT de M_X :

$$W^X = \left(|\mathcal{F}[h]|^2/\sigma^2 + 2\frac{\lambda}{\delta^2} (|\mathcal{F}[d_x]|^2 + |\mathcal{F}[d_y]|^2) \right)^{-1} \quad (3.115)$$

La FFT de la moyenne X^0 s'écrit :

$$\mathcal{F}[X^0] = W^X \left(\mathcal{F}[h]^* \mathcal{F}[Y]/\sigma^2 + 2\frac{\lambda}{\delta} \left(\mathcal{F}[D_x^t B^x + D_y^t B_y] \right) \right) \quad (3.116)$$

La formule permettant d'obtenir X^{n+1} à partir de h^n et $(B^x)^n$,$(B^y)^n$ est finalement :

$$X^{n+1} = \mathcal{F}^{-1} \left[W^X \left(\mathcal{F}[h^n]^* \mathcal{F}[Y]/\sigma^2 + 2\frac{\lambda}{\delta} \left(\mathcal{F} \left[D_x^t (B^x)^n + D_y^t (B_y)^n \right] \right) \right) \right] \quad (3.117)$$

Cet algorithme est rapide, puisqu'il ne nécessite qu'une FFT (taille $N_x \times N_y$) et deux DCT (taille $\frac{1}{2}N_x \times \frac{1}{2}N_y$).

Lorsque X est fixé, la minimisation par rapport aux variables auxiliaires est effectuée en utilisant le théorème semi-quadratique (3.28), avec la fonction de pondération :

$$\varphi_p(k) = k - \frac{1}{2}\varphi'(k) = k - k/\sqrt{1+k^2} \tag{3.118}$$

On obtient alors $(B^x)^{n+1}$ et $(B^y)^{n+1}$ en fonction de X^{n+1} :

$$(B^x)^{n+1}_{ij} = \varphi_p([D_x X^{n+1}]_{ij}/\delta) \tag{3.119}$$

$$(B^y)^{n+1}_{ij} = \varphi_p([D_y X^{n+1}]_{ij}/\delta) \tag{3.120}$$

3.4.2 Algorithme proposé

ALGORITHME 3.4.1 (DÉCONVOLUTION X À PARAMÈTRES FIXÉS)
L'image Y est observée, et les paramètres λ, k et σ sont fixés.

- *Initialisation de X : $X^0 = Y$*
- *Répéter M fois, jusqu'à la convergence :*
  1. *Minimisation par rapport à $(B^x)^{n+1}$ et $(B^y)^{n+1}$ à X^n fixé en utilisant (3.119) et (3.120)*
  2. *Minimisation par rapport à X^{n+1} à $(B^x)^{n+1}$, $(B^y)^{n+1}$ et h^n fixés en utilisant (3.117) et (3.115)*

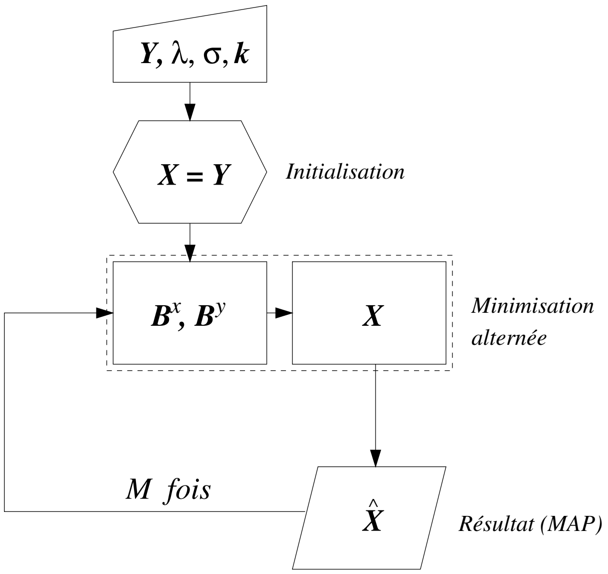

FIG. 3.12 – *Algorithme de déconvolution par minimisation de $U(X)$*

3.5 Algorithme complet : déconvolution et estimation

3.5.1 Algorithme proposé

L'estimation de l'image X est effectuée en calculant le Maximum A Posteriori (MAP), par l'algorithme 3.4.1. Le noyau de convolution est paramétré par une fonction $f(i,j,k)$ correspondant à la physique du système imageur. Les paramètres du noyau ainsi que le paramètre de régularisation de l'image sont estimés par maximum de vraisemblance. Les paramètres a et b du bruit, ainsi que l'écart-type moyen σ sont estimés par l'algorithme 3.3.1.

On suppose que la fonction $\mathcal{E}_\lambda^{-1}$ a été calculée au préalable et tabulée, en utilisant l'échantillonneur a priori 3.2.2.

L'initialisation des paramètres λ et k est arbitraire ; on peut profiter ici d'éventuelles connaissances a priori sur le type d'image utilisé et la taille du noyau h. Lorsque l'on traite une série d'images, l'estimation sur la première image peut servir à initialiser les valeurs des paramètres pour toutes les autres, si celles-ci ont des caractéristiques semblables.

ALGORITHME 3.5.1 (DÉCONVOLUTION AVEUGLE)
L'image Y est observée.
Les paramètres λ et k sont initialisés ($\lambda \sim 0.1$, $k \sim 1$).
Répéter M fois, jusqu'à la convergence :

- *Estimation de σ, a et b par l'algorithme 3.3.1*
- *Estimation de X^{n+1} par déconvolution, en fonction de λ^n, k^n et σ, en maximisant $P(X \mid Y,\lambda^n,k^n,\sigma)$ (algorithme 3.4.1)*
- *Échantillonnage a posteriori sur X (algorithme 3.2.1) en fonction de λ^n, k^n et σ, pour calculer les espérances E_Φ et E_k, relatives aux gradients par rapport à λ et k, par moyenne empirique (équation (3.78))*
- *Calcul de λ^{n+1} en fonction de E_Φ et $\mathcal{E}_\lambda^{-1}$ en utilisant l'équation (3.93)*
- *Calcul de k^{n+1} en fonction de E_k et k^n en utilisant l'expression du gradient (3.86) (voir algorithme 3.3.2)*

Cet algorithme de déconvolution aveugle effectue en fait de façon simultanée l'estimation des paramètres et la déconvolution, puisque l'échantillonneur a posteriori est initialisé avec le résultat de la déconvolution pour les valeurs courantes des paramètres. Néanmoins, nous l'utilisons sur une petite zone de l'image (en général 64×64 ou 128×128) afin de réduire le temps de calcul, et en profitant du fait que le noyau h est invariant par translation. En effet, le but de cet algorithme est essentiellement l'estimation de h et de (σ,a,b). Une fois que ces paramètres instrumentaux ont été estimés, il est possible de mettre en œuvre des techniques de déconvolution plus rapides [34, 33] afin de restaurer l'image dégradée dans son intégralité.

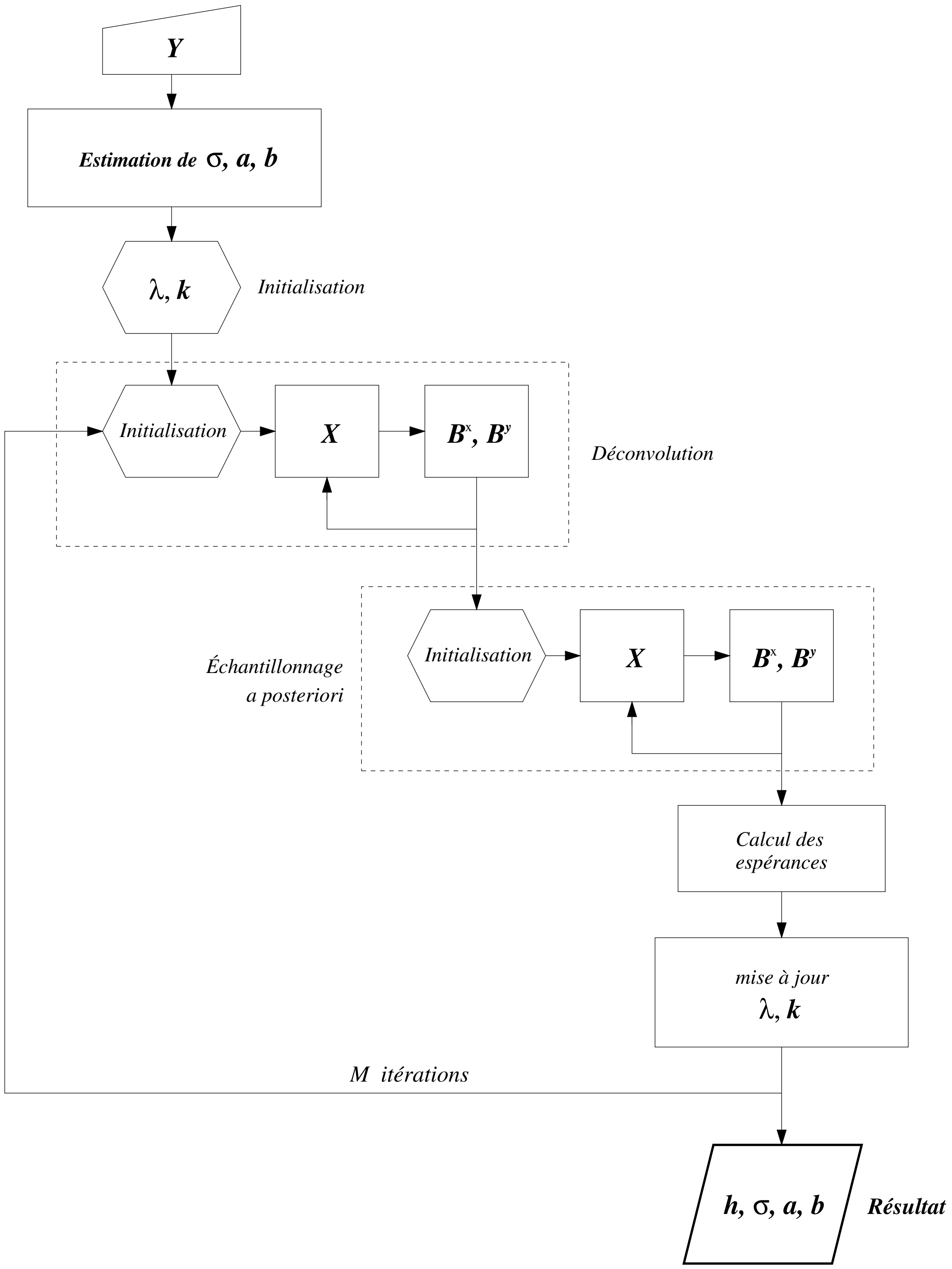

FIG. 3.13 – *Algorithme de déconvolution aveugle : estimation du bruit* (σ,a,b)*, puis estimation conjointe du noyau de convolution* h *paramétré, de l'image* X *et du paramètre de régularisation* λ

3.5.2 Coût et complexité

Estimation du bruit

L'estimation des paramètres du bruit par l'algorithme 3.3.1 nécessite deux DCT, appliquées à l'image entière afin d'obtenir une précision maximale sur a et b. Le coût lié aux autres opérations est négligeable par rapport aux DCT.

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 chaque pixel, lors du calcul de la DCT par une FFT.

Cela représente donc $10\log_2 n + 30$ opérations par pixel pour l'estimation du bruit (par exemple 110 opérations par pixel pour l'image de Nîmes 512×512).

Estimation du noyau de convolution

Nous considérons ensuite l'algorithme 3.5.1, pour l'estimation du noyau de convolution.

La complexité de cette technique est essentiellement déterminée par la FFT et la DCT. Ces transformées sont effectuées aussi bien dans l'échantillonnage a posteriori que dans l'étape de déconvolution à paramètres fixés.

Pour des imagettes de taille 128×128, le calcul des espérances a posteriori nécessite environ 20 échantillons (on suppose que les espérances a priori ont déjà été calculées). Il faut ajouter à cela les M_i=15 itérations nécessaires pour l'initialisation, donc il faut compter environ M=35 itérations au total pour l'échantillonneur a posteriori. Le calcul de l'espérance de Φ n'est pas coûteux en lui-même, en revanche le calcul de l'espérance correspondant au gradient ∇_k (paramètres du noyau) nécessite deux DCT (des multiplications sont effectuées aussi bien dans l'espace image que dans l'espace fréquentiel).

Les M itérations sont nécessaires pour le tirage alterné X, B^x et B^y. Le tirage conditionnel de X sachant les variables auxiliaires nécessite deux DCT. Le temps nécessaire au tirage de B^x et B^y peut être négligé.

Pour le calcul des espérances, il faut compter environ 30 DCT pour les itérations d'initialisation, puis 80 DCT pour les itérations comprenant le calcul des espérances, donc 110 DCT au total, et ceci lors de chaque étape de descente de gradient. Mais si l'on utilise des imagettes de taille 128×128, ce qui est largement suffisant pour l'estimation des paramètres de h, cela représente moins de 4 secondes sur un Sun Ultra 1 à 166 MHz.

Il faut ajouter à cela l'étape de déconvolution, dont le temps de calcul est plus court que pour l'estimation, puisque 15 itérations (ce qui représente 30 DCT) suffisent pour obtenir la convergence. Cela nécessite seulement quelques secondes de calcul sur la même machine, si l'on se réfère à des imagettes de même taille.

Chaque itération de base de l'algorithme 3.5.1 (déconvolution, espérances et mise à jour des paramètres) nécessite finalement le calcul de 140 DCT.

Comme 20 itérations (descente de gradient) suffisent, finalement le coût total de l'estimation du noyau h est de 2800 DCT, ou $14000 \log_2 n + 42000$ opérations par pixel. En temps de calcul, cela signifie 80 secondes sur un Sun Ultra 1 à 166 MHz.

3.5.3 Optimisations possibles

Le calcul préalable d'espérance a priori permet de gagner du temps au cours de l'estimation. Nous en avons tenu compte dans les calculs ci-dessus. Ainsi seules les espérances a posteriori sont à estimer pour chaque image, puisqu'elles dépendent de l'image à traiter.

Nous avons testé des techniques d'approximation qui permettraient de diviser le temps de calcul par 35 (moins de 3 secondes pour une imagette 128×128). Les détails de ces techniques sont confidentiels. Même si la qualité des images ainsi restaurées n'est pas aussi satisfaisante, nous avons pu vérifier que les paramètres du noyau h ainsi estimés sont corrects. Ces résultats devront être confirmés par des tests complémentaires sur un grand nombre d'images.

Chapitre 4

Résultats

Nous avons estimé les paramètres instrumentaux sur l'image de Nîmes (simulation SPOT 5 fournie par le CNES), de taille 512×512. L'estimation du bruit a été effectuée sur l'image entière, mais nous avons utilisé une imagette de taille 128×128 pour l'estimation des paramètres du noyau.

Le noyau h théorique, ayant servi pour la dégradation, est donné sur la figure 4.1.

L'écart-type théorique, dans l'approximation où le bruit est de variance constante, est $\sigma \simeq 1.35$. Les paramètres du bruit variable sont $a = 1.299$ et $b = 0.00676$.

Avec l'algorithme 3.3.1 nous avons obtenu $a \simeq 1.30$, $b \simeq 0.0070$ et $\sigma \simeq 1.32$. La relation entre la variance du bruit et la valeur des pixels de l'image observée est illustrée par la figure 4.4.

Le noyau est paramétré dans l'espace fréquentiel par une exponentielle. Nous n'avons pas modélisé ici le flou lié au déplacement du satellite.

Dans l'espace image, cela revient à utiliser la fonction suivante :

$$h_{ij} \propto \frac{1}{1 + i^2/2k_1^2} \frac{1}{1 + j^2/2k_2^2} \tag{4.1}$$

L'algorithme complet de déconvolution aveugle 3.5.1 fournit les valeurs $k_1 = k_2 \simeq 1.1$, le noyau obtenu est représenté sur la figure 4.3.

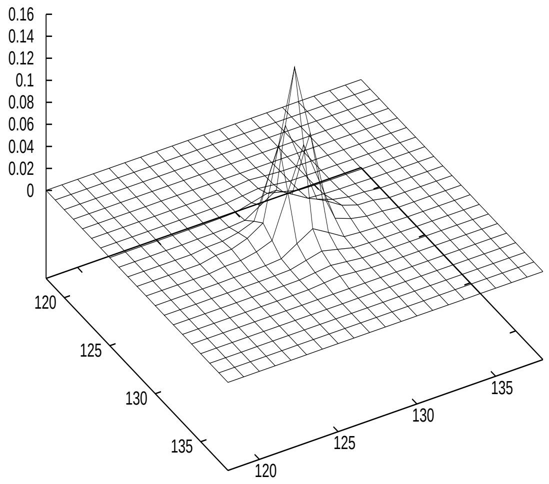

FIG. 4.1 – *Noyau h correspondant à l'image de Nîmes à 2.5m*

FIG. 4.2 – *Image de Nîmes* 512×512, *floue et bruitée (résolution 2.5m)*

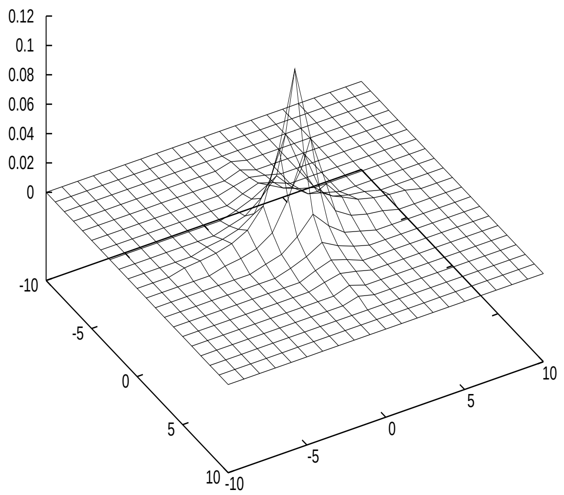

FIG. 4.3 – *Noyau h estimé sur les imagettes de la figure 4.2, avec un paramétrage donné par l'équation (4.1) et $k_1 = k_2 \simeq 1.1$*

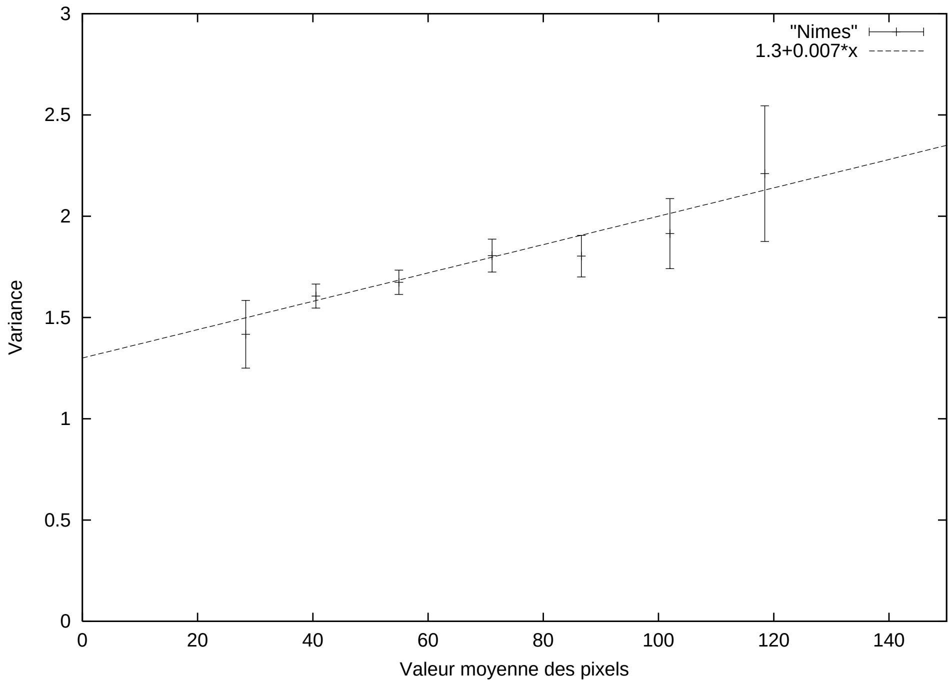

FIG. 4.4 – *Estimation des paramètres a et b du bruit blanc gaussien variable par l'algorithme 3.3.1*

Chapitre 5

Conclusion

5.1 Synthèse des résultats

Le but de cette étude est la restitution des paramètres instrumentaux en imagerie satellitaire (tests sur des simulations SPOT 5).

Pour cela, le noyau de convolution est paramétré, par une fonction traduisant la physique du système imageur étudié. Il s'agit d'estimer les paramètres de ce noyau, ainsi que ceux du bruit additif lié au capteur.

La méthode proposée repose essentiellement sur deux étapes :

- Estimation des paramètres (a,b,σ) du bruit blanc gaussien variable, par filtrage passe-bande de la DCT de l'image observée Y, et par calcul de la variance du bruit après classification des pixels dans l'espace image.
- Estimation conjointe des paramètres k (noyau de convolution) et du paramètre λ (régularisation de l'image) par maximum de vraisemblance, en utilisant une méthode MCMC. La vraisemblance est optimisée par une méthode de type Newton ; les gradients sont estimés par échantillonnage des lois a priori et a posteriori relatives à l'image X.

Nous avions déjà proposé dans [29] une méthode d'estimation de paramètre pour le modèle sur X (estimation de λ). Les problèmes d'échantillonnage en X selon les loi a priori et a posteriori étaient donc déjà résolus.

Cependant, le passage au cas où h est aussi inconnu, a posé plusieurs difficultés nouvelles, car l'on a d'abord cherché à l'estimer en tout point [30]. Nous avons traité le problème de l'échantillonnage du noyau h, sous contraintes de positivité et de normalisation, permettant d'optimiser la vraisemblance des paramètres par une méthode MCMC. Cela a été rendu possible en utilisant une diagonalisation par DCT, comme dans le cas où le noyau est paramétré. D'autre part, il a été nécessaire d'échantillonner les lois jointes de X et de h. Toutes ces difficultés ont été levées [30].

Cependant, il n'a pas été possible d'estimer h en tout point, car il n'existe pas de solution au problème d'estimation du paramètre μ de régularisation du noyau. D'autre part, nous avons également constaté que le modèle de régularisation sur h impose, en fait, la forme du noyau, et cela équivaut, en pratique, à paramétrer le noyau [30].

La méthode qui a été finalement retenue consiste à paramétrer le noyau de convolution, en fonction des données dont on dispose concernant la physique du problème étudié. Les résultats obtenus sont très satisfaisants, compte tenu du fait que l'unique a priori imposé à l'ensemble du problème de déconvolution aveugle se réduit au choix de la fonction h paramétrée. En effet, nous n'avons pas imposé de contrainte particulière à l'image inconnue X, si ce n'est le modèle a priori, mais le paramètre de ce modèle a été estimé de manière automatique.

5.2 Perspectives

5.2.1 Estimation de la réponse impulsionnelle

Pour permettre de valider et d'améliorer la méthode utilisée pour l'estimation des paramètres du noyau de convolution, des tests doivent être effectués sur différentes images (simulations SPOT 5 et Pléiades), avec différents niveaux de bruit, chacune des images correspondant soit à un modèle de réponse impulsionnelle différent, soit à différentes valeurs des paramètres du même modèle.

La connaissance de la physique du capteur étudié pourra être avantageusement utilisée, afin d'améliorer la précision du noyau estimé. Ainsi, les fonctions correspondant à un flou de défocalisation, de bougé, etc. pourront être utilisées dans le modèle ; les paramètres estimés auront une signification physique et leur utilisation sera plus aisée (par exemple pour corriger, à distance, les défauts de réglage du système optique).

5.2.2 Prise en compte d'une grille quinconce

La méthode que nous avons développée fonctionne avec une grille d'échantillonnage carrée.

Lorsque les images sont échantillonnées sur une grille quinconce, ce qui sera le cas pour le futur satellite SPOT 5, il est possible de se ramener à une grille carrée par sous-échantillonnage, mais la précision d'estimation est moindre par rapport au cas où l'échantillonnage quinconce est directement incorporé dans l'algorithme d'estimation des paramètres instrumentaux.

D'autre part, la prise en compte de la grille quinconce permettrait également d'exprimer la fonction de flou directement dans le repère du capteur, ce qui simplifierait la modélisation. Cela permettrait notamment d'améliorer l'estimation du bougé lié au déplacement du satellite pendant la prise de vue.

Bibliographie

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

[2] G.R. Ayers and J.C. Dainty. Iterative blind deconvolution method and its applications. *Optics Letters*, 13(7):547–549, July 1988.

[3] L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occuring in the statistical analysis of probabilistic functions of markov chains. *Ann. Maths. Stats.*, 41:164–171, 1972.

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

[5] D.S.C. Biggs and M. Andrews. Iterative blind deconvolution of extended objects. In *IEEE Proceedings of the 3rd International Conference of Image Processing*, volume II, pages 454–457, Santa Barbara, Oct. 1997.

[6] A. Blake and A.Zisserman. *Visual reconstruction*. The MIT Press Series in artificial intelligence, 1987.

[7] L. Blanc-Féraud, M. Barlaud, and P. Mathieu. Blind restoration of noisy blurred images using Maximum Likelihood methods with constraints. *Optical Engineering*, 30(4):431–437, April 1991.

[8] L. Blanc-Féraud, P. Charbonnier, G. Aubert, and M. Barlaud. Nonlinear image processing : Modeling and fast algorithm for regularization with edge detection. In *IEEE Proc. of ICIP*, pages 474–477, Washington, USA, Oct. 1995.

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

[10] S.P. Brooks and G.O. Roberts. Assessing convergence of Markov Chain Monte Carlo algorithms. *Statistics and Computing*, 8:319–335, 1998.

[11] M. Cannon. Blind deconvolution of spatially invariant image blurs with phase. *IEEE Trans. on ASSP*, 24(1):58–63, Feb. 1976.

[12] G. Celeux and J. Diebolt. A random imputation principle : the stochastic E.M. algorithm. *INRIA Research Report*, 901, 1988.

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

[14] B. Chalmond. An iterative Gibbsian technique for reconstruction of m-ary images. *Pattern Recognition*, 22(6):747–761, 1989.

[15] B. Chalmond. PSF estimation for image deblurring. *Graphical Models and Image Processing*, 53(4):364–372, July 1991.

[16] M.M. Chang, A.M. Tekalp, and A.T. Erdem. Blur identification using the bispectrum. *IEEE Trans. on SP*, 39(10):2323–2325, Oct. 1991.

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

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

[19] A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from incomplete data via the EM algorithm. *Journal of Roy. Stat. Soc. B*, 39:1–38, 1977.

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

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

[22] S. Geman and D.E. Mc Clure. Bayesian image analysis : an application to single photon emission tomography. In Amer. Statist. Assoc., editor, *Proc. Statist. Comput. Sect.*, pages 12–18, Washington DC, 1985.

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

[24] R. Gerchberg and W. Saxton. A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik*, 35:237–246, 1972.

[25] P. J. Green. Bayesian reconstruction from emission tomography data using a modified EM algorithm. *IEEE Trans. on MI*, 9(1):84–93, March 1990.

[26] G. Harikumar and Y. Bresler. Exact image deconvolution from multiple FIR blurs. *IEEE Trans. on IP*, 8(6), June 1999.

[27] T. Hebert and R. Leahy. A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors. *IEEE Trans. on MI*, 8(2):194–202, June 1989.

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

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

[30] A. Jalobeanu, L. Blanc-Féraud, and J. Zerubia. *Etude de la restitution des paramètres instrumentaux*. Rapport de contrat CNES, Toulouse, Février 1999.

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

[32] A. Jalobeanu, L. Blanc-Féraud, and J. Zerubia. Restauration automatique d'images satellitaires par une méthode MCMC. In *GRETSI*, Vannes, Sept. 1999.

[33] A. Jalobeanu, L. Blanc-Féraud, and J. Zerubia. Estimation of adaptive parameters for satellite image deconvolution. *INRIA Research Report*, June 2000.

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

[35] S.M. Jefferies and J.C. Christou. Restoration of astronomical images by iterative blind deconvolution. *The Astrophysical Journal*, 415:862–874, Oct. 1993.

[36] A.K. Katsaggelos and K.T. Lay. Maximum Likelihood blur identification and image restoration using the EM algorithm. *IEEE Trans. on IP*, 39:729–733, 1991.

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

[38] D. Kundur and D. Hatzinakos. Blind image deconvolution. *IEEE Signal Processing Magazine*, 13(3):43–64, May 1996.

[39] D. Kundur and D. Hatzinakos. Blind image deconvolution revisited. *IEEE Signal Processing Magazine*, 13(6):61–63, Nov. 1996.

[40] R.L. Lagendijk, J. Biemond, and D.E. Boekee. Identification and restoration of noisy blurred images using the expectation-maximization algorithm. *IEEE Trans. on ASSP*, 38(7), July 1990.

[41] R.L. Lagendijk, A.M. Tekalp, and J. Biemond. Maximum Likelihood image and blur identification: a unifying approach. *Optical Engineering*, 29(5):422–435, May 1990.

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

[43] R.G. Lane and R.H.T. Bates. Automatic multidimensional deconvolution. *Journal of Optics Soc. Am. A*, 4(1), 1987.

[44] K. Lange. Convergence of EM image reconstruction algorithm with Gibbs smoothing. *IEEE Trans. on MI*, MI-9(4):439–446, Dec. 1990.

[45] Q. Li and Y. Yoshida. Parameter-estimation and restoration for motion blurred images. *IEICE*, E80A(8):1430–1437, Aug. 1997.

[46] L. B. Lucy. An iterative technique for the rectification of observed images. *The Astronomical Journal*, 79(6):745–754, June 1974.

[47] B.C. McCallum. Blind deconvolution by simulated annealing. *Optics Communications*, 75(2):101–105, Feb. 1990.

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

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

[50] J.J. More. The levenberg-marquardt algorithm: Implementation and theory. New York, l978. Springer-Verlag.

[51] N. Nordström. Biased anisotropic diffusion : a unified regularization and diffusion approach to edge detection. *Image and Vision Computing*, 8(4):318–327, 1990.

[52] J. Park and I.W. Sandberg. Universal approximation using radial-basis-function networks. volume 3, pages 246–257, 1991.

[53] W. Pieczynski. Champs de Markov cachés et estimation conditionnelle itérative. *Revue Traitement du Signal*, 11(2):141–153, 1994.

[54] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, 1984.

[55] S.J. Reeves and R. M. Mersereau. Blur identification by the method of generalized cross-validation. *IEEE Trans. on IP*, 1(3):301–311, July 1992.

[56] W. H. Richardson. Bayesian-based method of image restoration. *journal of optical Society of America*, 62(1):55–59, Jan. 1972.

[57] C. Robert. *Méthodes de Monte Carlo par chaînes de Markov*. Statistique mathématique et probabilité. Economica, 1996.

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

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

[60] T.J. Schulz and S.C. Cain. Simultaneous phase retrieval and deblurring for the Hubble space telescope. In R.J. Hanisch and R.L. White, editors, *The restoration of HST images and Spectra II*, pages 206–211. Space telescope science institute, 1994.

[61] M. A. Tanner. *Tools for statistical inference*. Springer series in Statistics. Springer, 1996.

[62] E. Thiebaut and J-M. Conan. Strict a priori constraints for maximum likelihood blind deconvolution. *Journal Opt. Soc. Am. A*, 12(3):485–492, March 1995.

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

[64] F. Tsumuraya, N. Miura, and N. Baba. Iterative blind deconvolution method using Lucy's algorithm. *Astronomy and Astrophysics*, 282:699–708, 1994.

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

[66] K. Wirawan, P. Duhamel, and H. Maître. Robust MRE methods for blind multichannel image restoration. In *GRETSI*, Vannes, France, 1999.

[67] Y-L You and M. Kaveh. A regularization approach to blind restoration of images degraded by shift-variant blurs. In *IEEE ICIP*, pages 2607–2610, Washington, USA, 1995.

[68] Y.-L. You and M. Kaveh. Blind image restoration by anisotropic regularization. *IEEE Trans. on IP*, 8(3):396–407, March 1999.

[69] Y-L. You, W. Xu, A. Tannenbaum, and M. Kaveh. Behavioral analysis of anisotropic diffusion in image processing. *IEEE Trans. on IP*, 5(11):1539–1553, Nov. 1996.

[70] L. Younes. Estimation and annealing for Gibbsian fields. *Annales de l'Institut Poincaré*, 24:269–294, 1988.

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

[72] J. Zhang. The mean field theory in EM procedures for blind Markov random fields image restoration. *IEEE Trans. on IP*, 2(1), Jan. 1993.

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