[14] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller et E. Teller, "Equation of State Calculations by Fast Computing Machines, *J. of Chem. Physics*, Vol. 21, pp 1087–1092, 1953.

[15] M. Sigelle et R. Ronfard, "Relaxation d'Images de Classification par Champs de Markov et Modèles de la Physique Statistique", *Proc. RFIA*, Lyon, France, 1991.

[16] J. Zerubia et R. Chellappa, "Mean field approximation using compound Gauss - Markov random field for edge detection and image restoration", *Proc. ICASSP*, Albuquerque, USA, Avr. 1990.

[17] J. Zerubia et F. Ployette, "Détection de contours et restauration d'image par des algorithmes déterministes de relaxation. Mise en œuvre sur la machine à connexions CM2", *Rapport de recherche INRIA N° 1291*, Oct. 1990.

[18] "Connection Machine Model CM-2 Technical Summary, Version 5.1", *Thinking Machines Corporation*, Cambridge, Massachusetts, 1989.

[19] "C* Programming Guide, Version 6.0", *Thinking Machines Corporation*, Cambridge, Massachusetts, 1991.

Bibliographie

[1] R. Azencott, "Image Analysis and Markov Fields", *Proc. ICIAM*, Paris, France, 1987.

[2] R. Azencott, "Markov Fields and Image Analysis", *Proc. AFCET*, Antibes, France, 1987.

[3] M. Berthod, "Definition of a Consistent Labeling as a Global Extremum", *Proc. ICPM*, Munnich, Allemagne, 1982.

[4] J. Besag, "On the Statistical Analysis of Dirty Pictures", *Jl Roy. Statist. Soc. B.*, 1986.

[5] A. Blake et A. Zisserman, "Visual reconstruction", *MIT Press, Cambridge - MA*, 1987.

[6] P. Chou et C. Brown, "The Theory and Practice of Bayesian Image Labeling", *International Journal of Computer Vision*, Vol. 4, pp 185–210, 1990.

[7] O. Faugeras et M. Berthod, "Scene labeling: an optimization approach", *Pattern Recognition*, Vol. 12, pp 339–347, 1980.

[8] D. Geiger et F. Girosi, "Parallel and deterministic algorithms for MRFs : surface reconstruction and integration", *Proc. ECCV*, Antibes, Avr. 1990.

[9] S. Geman et D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images", *IEEE Trans. Pattern Analysis, Machine Intel.*, Vol. PAMI-6, pp 721–741, 1984.

[10] C. Graffigne et J. Zerubia, "Rapport sur la Segmentation: Approches Statistiques", à paraître dans le rapport segmentation du GdR TdSI, Mai 1991.

[11] R. Hummel et S. Zucker,"On the foundations of relaxation labeling processes", *IEEE Trans. Pattern Analysis, Machine Intel.*, Vol. PAMI-5, pp 267–287, 1983.

[12] F. Jeng et J. Woods, "Compound Gauss - Markov Random fields for Image Estimation", *IEEE Trans. Acoust., Speach and Signal Proc.*, Vol. ASSP-39, pp 638–697, 1991.

[13] P. van Laarhoven et E. Aarts, "Simulated Annealing: Theory and Applications" *D. Reidel Publishing Company*, Dordrecht, Holland, 1987.

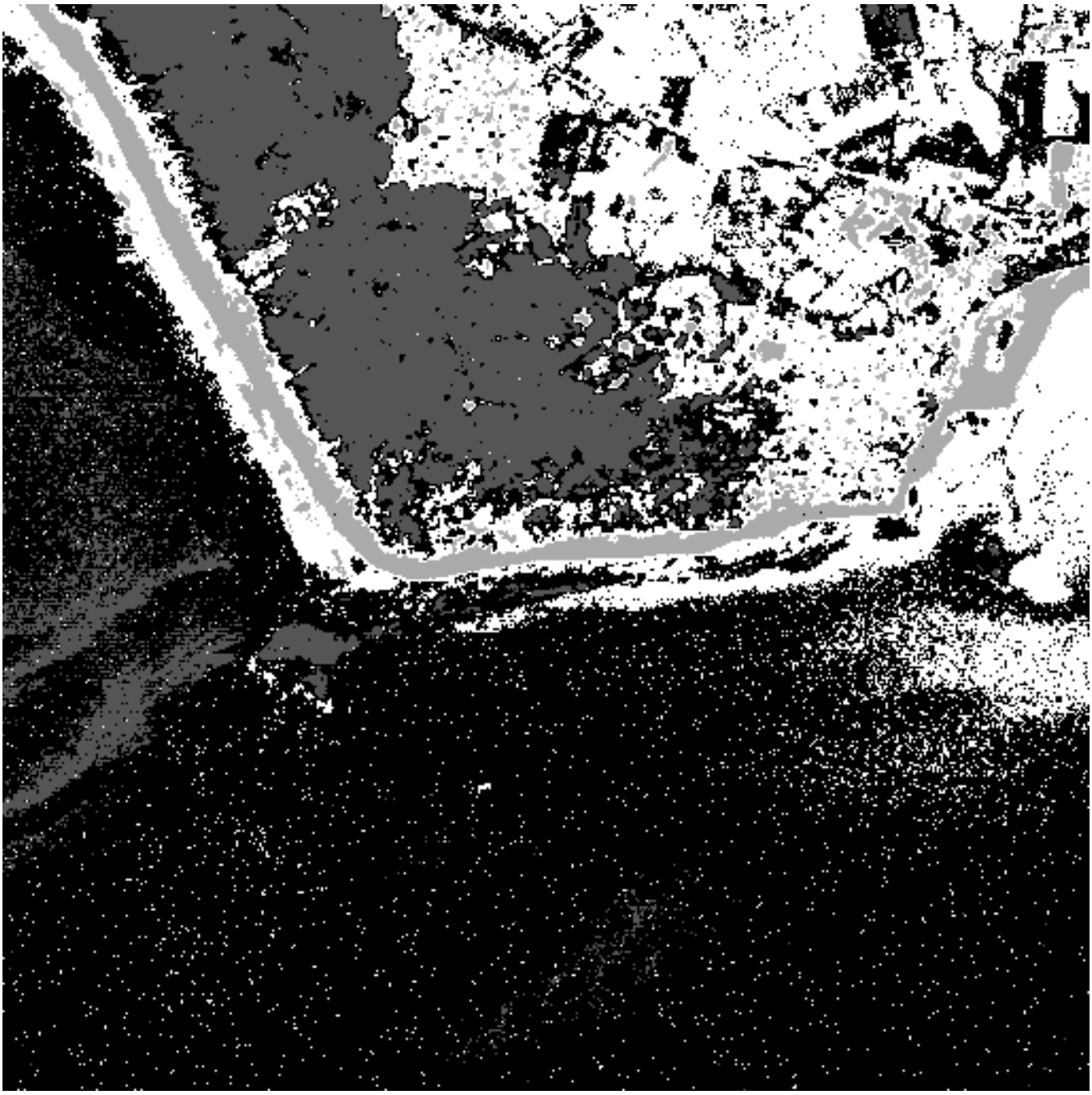


Figure 24 : Metropolis, 200 itérations

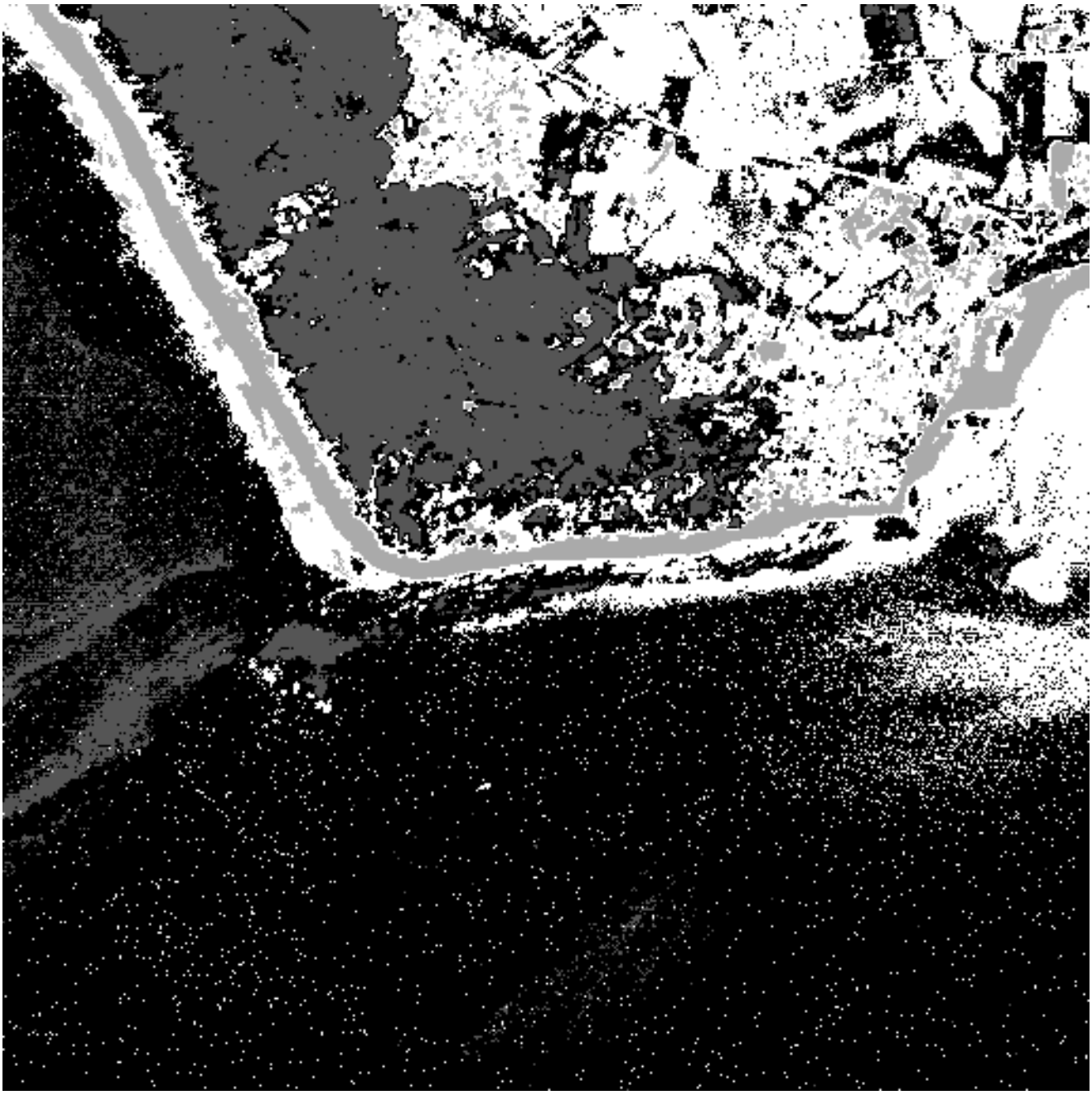


Figure 23 : Echantillonneur de Gibbs, 200 itérations

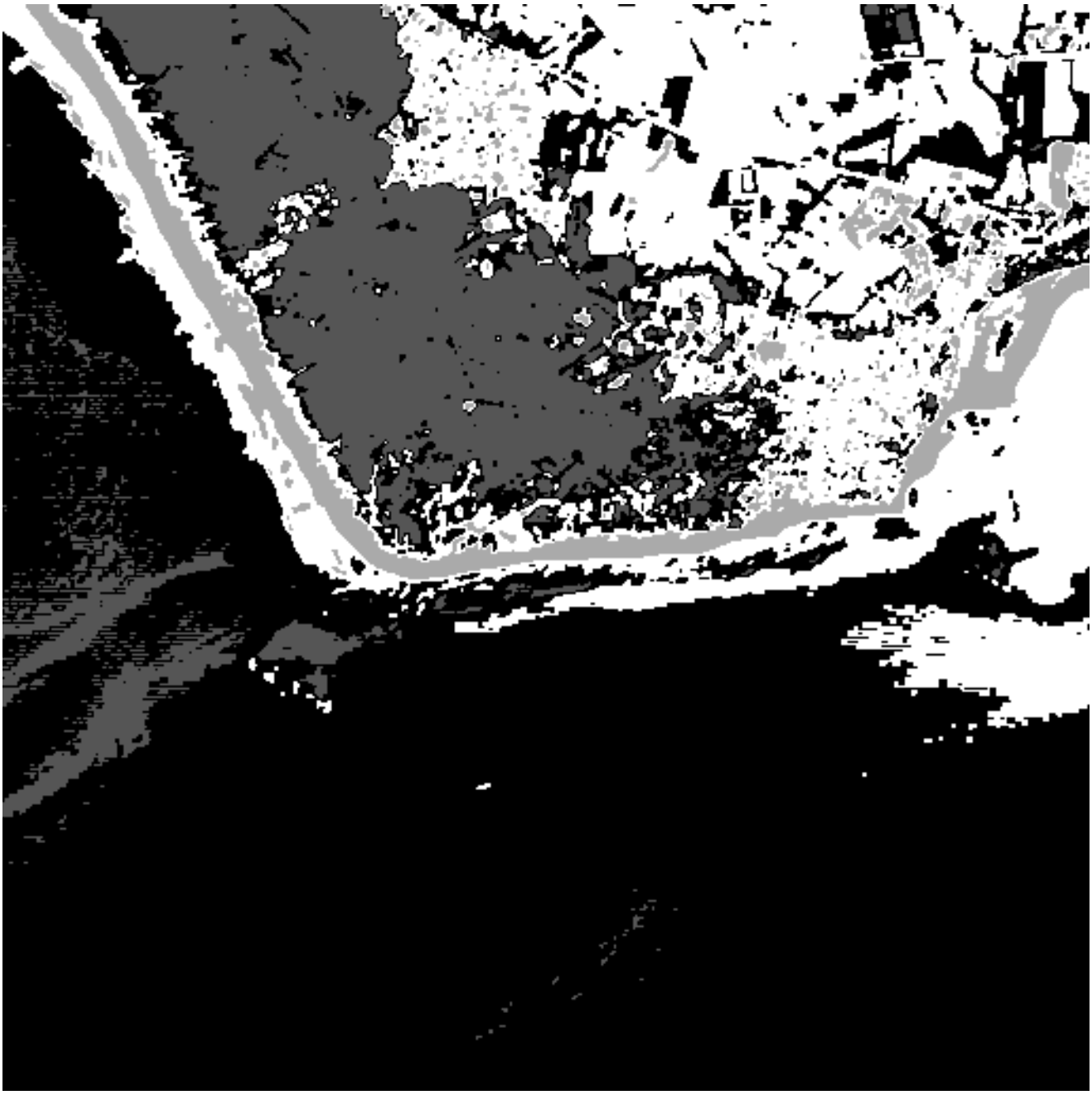


Figure 22 : ICM, 6 itérations

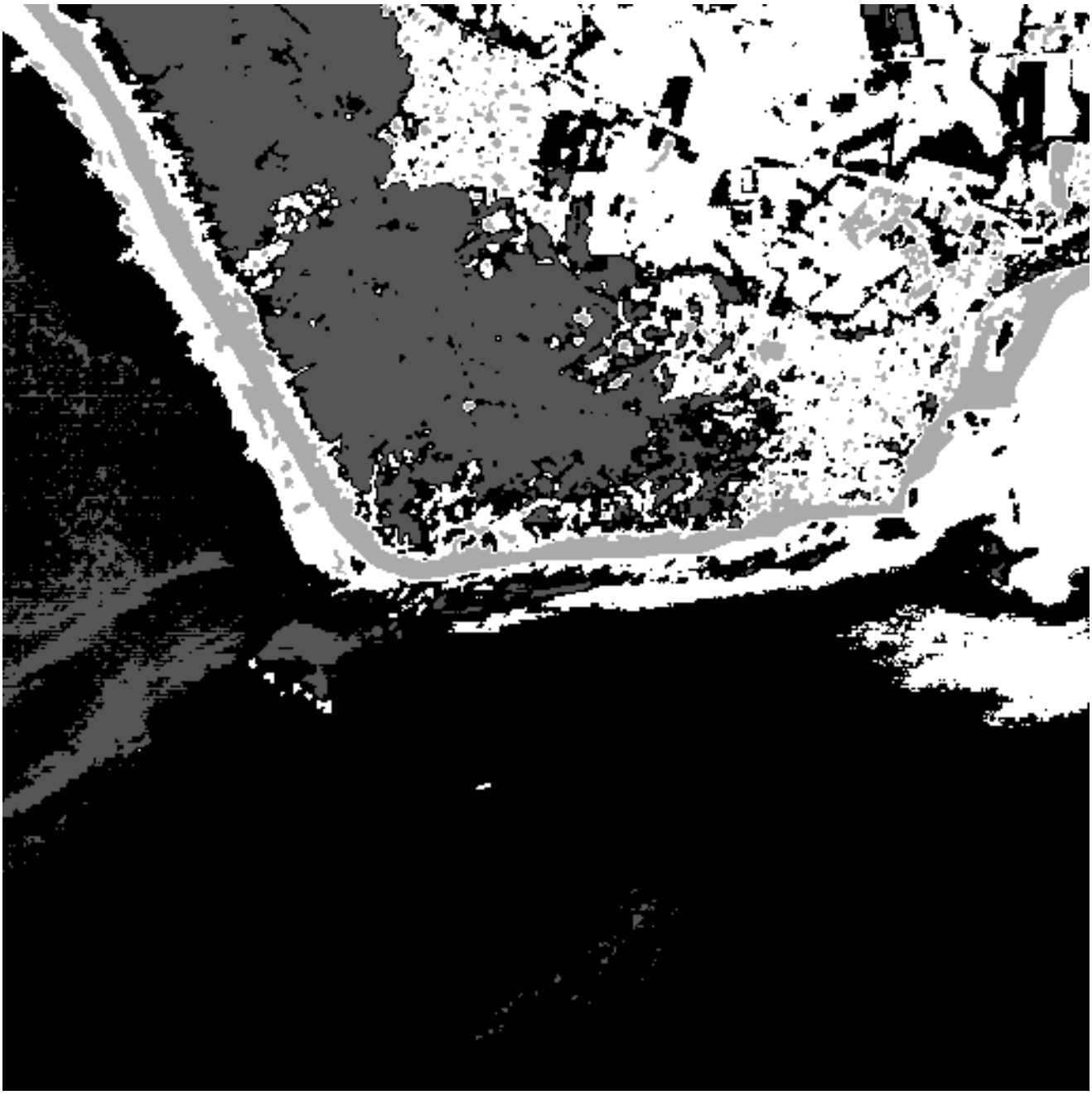
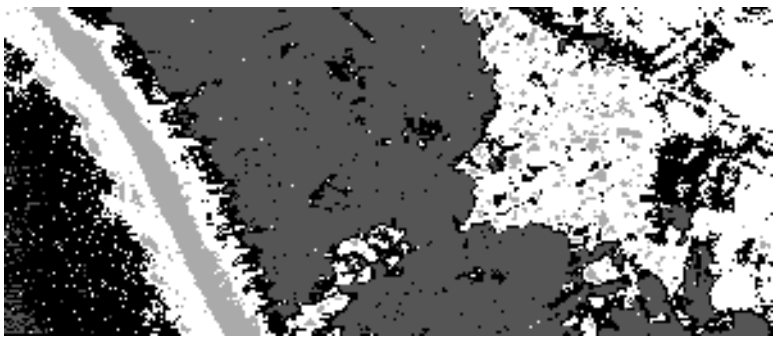


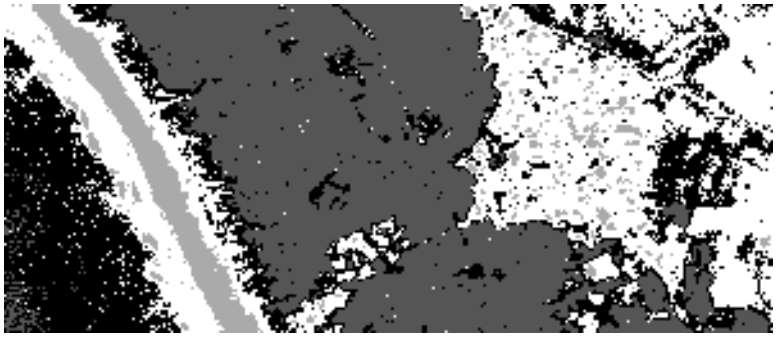
Figure 21 : Méthode proposée, 200 itérations

Figure 20 : Image originale SPOT 512×512

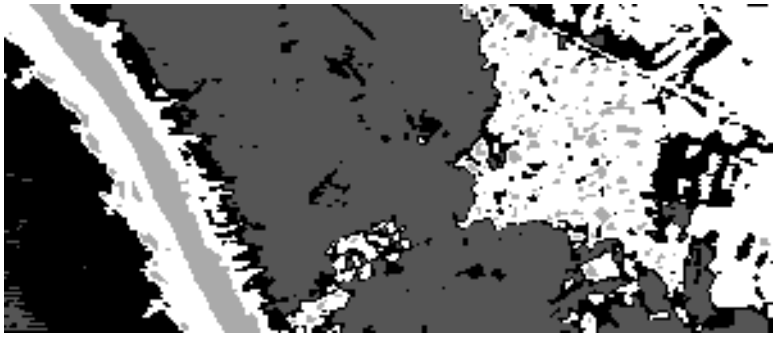
C Résultats obtenus sur une image SPOT 512×512



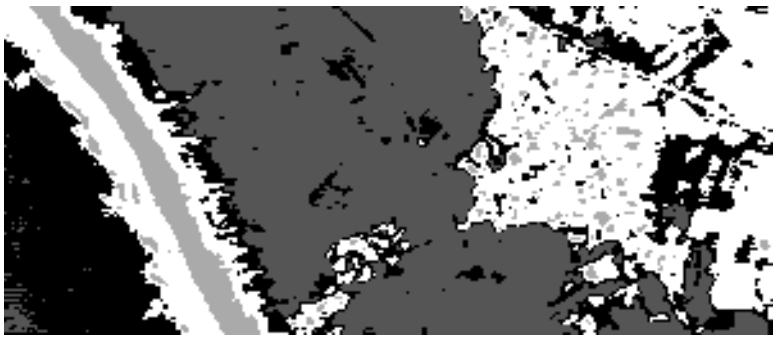
Metropolis



Echantillonneur de Gibbs



ICM



Méthode proposée

Figure 19 : Résultats obtenus sur une image SPOT 300×128

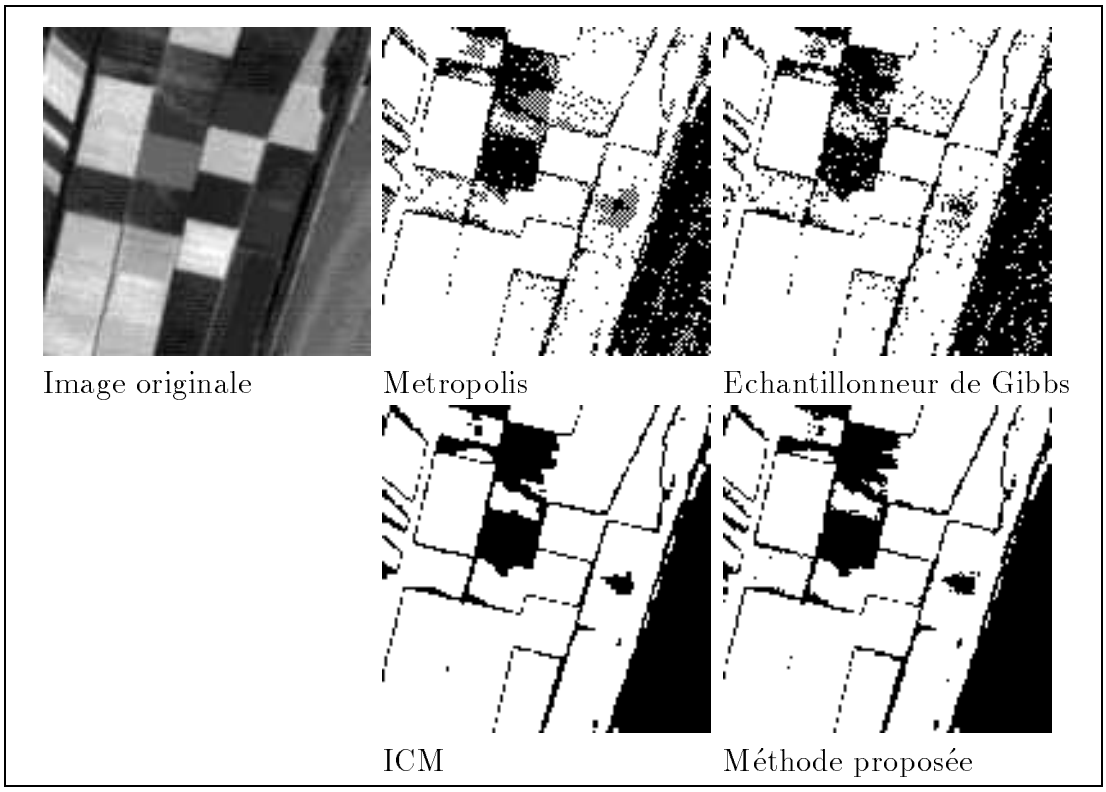


Figure 17 : Résultats obtenus sur une image SPOT 128×128

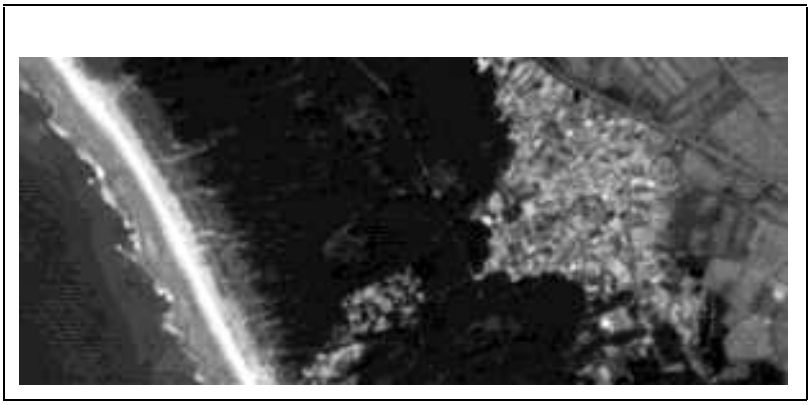


Figure 18 : Image originale SPOT 300×128

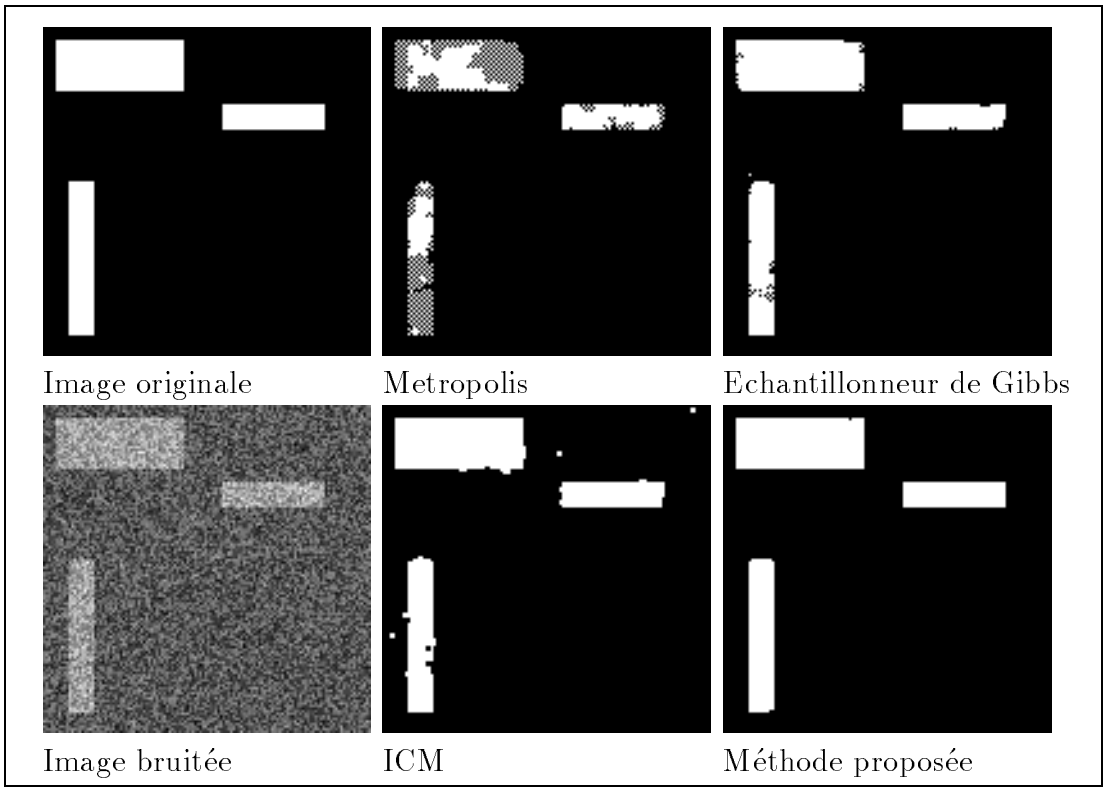


Figure 16 : Résultats obtenus sur une image synthétique 128×128

B Images

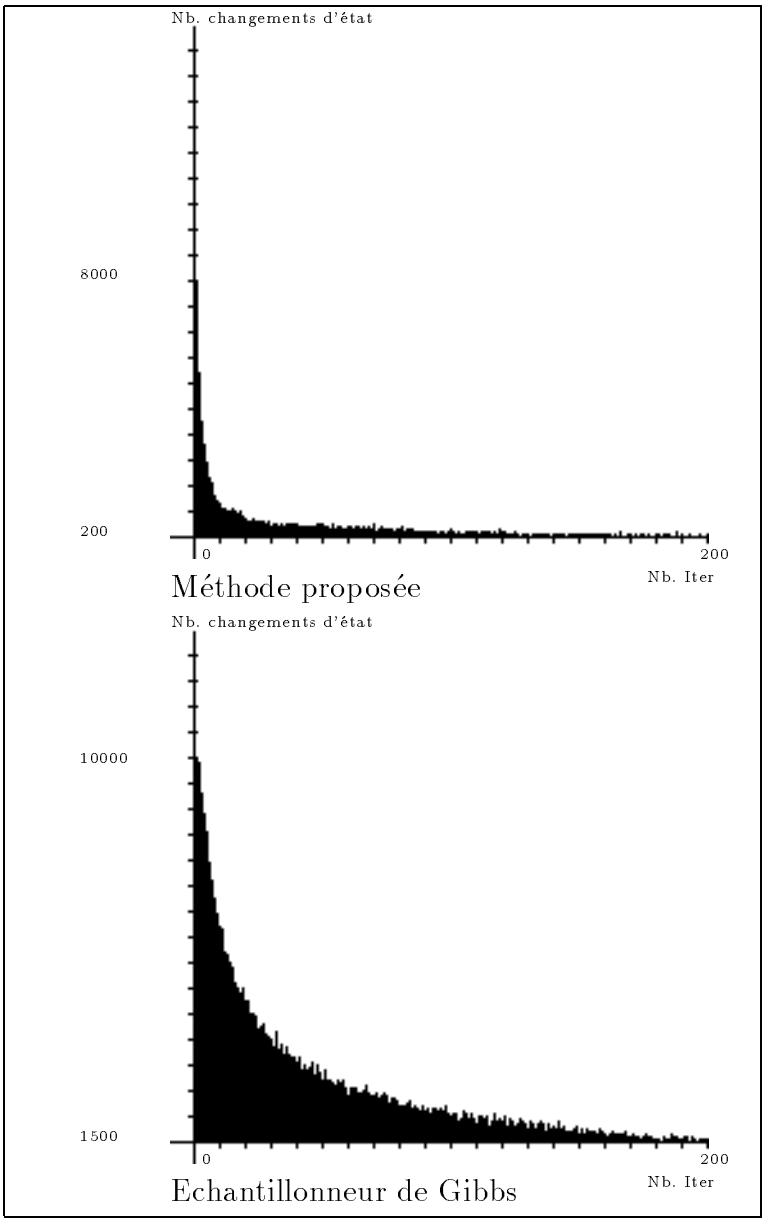


Figure 15 : Le nombre de sites en fonction du nombre d'itérations où la nouvelle énergie est plus grande que la précédente (image SPOT 300×128)

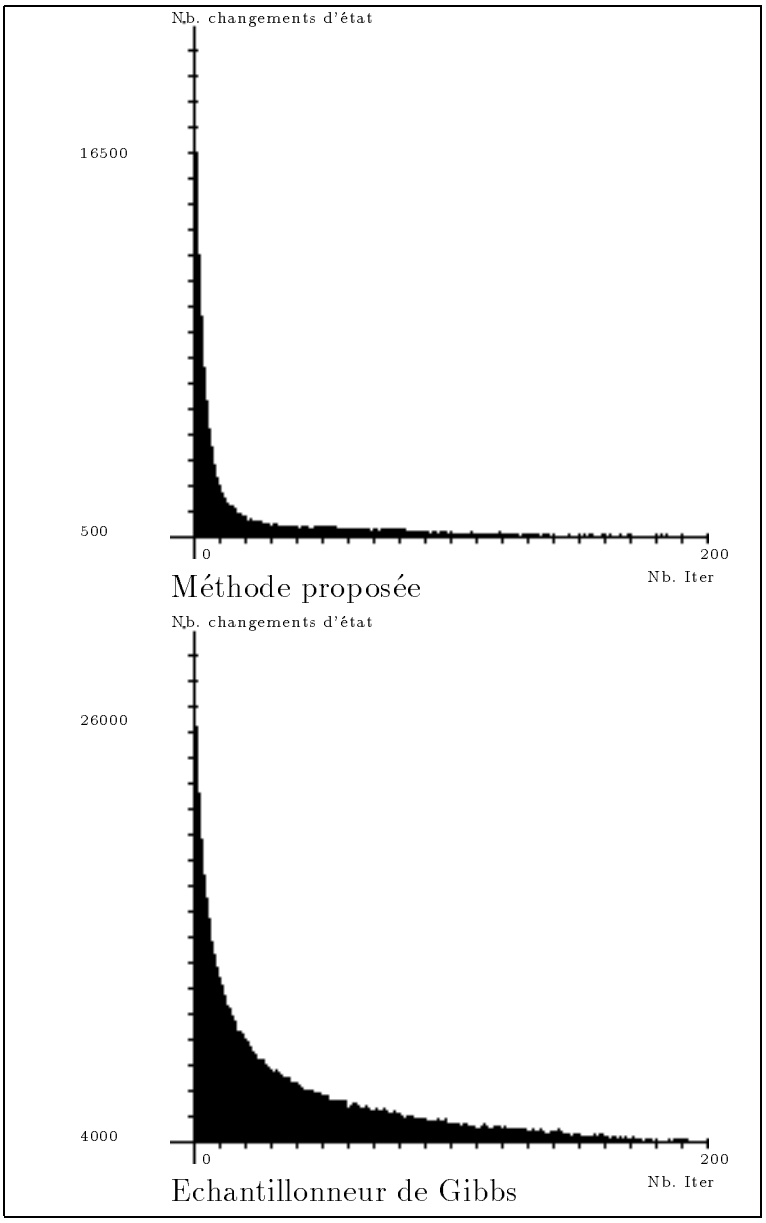


Figure 14 : Le nombre de changements d'état en fonction du nombre d'itérations (image SPOT 300×128)

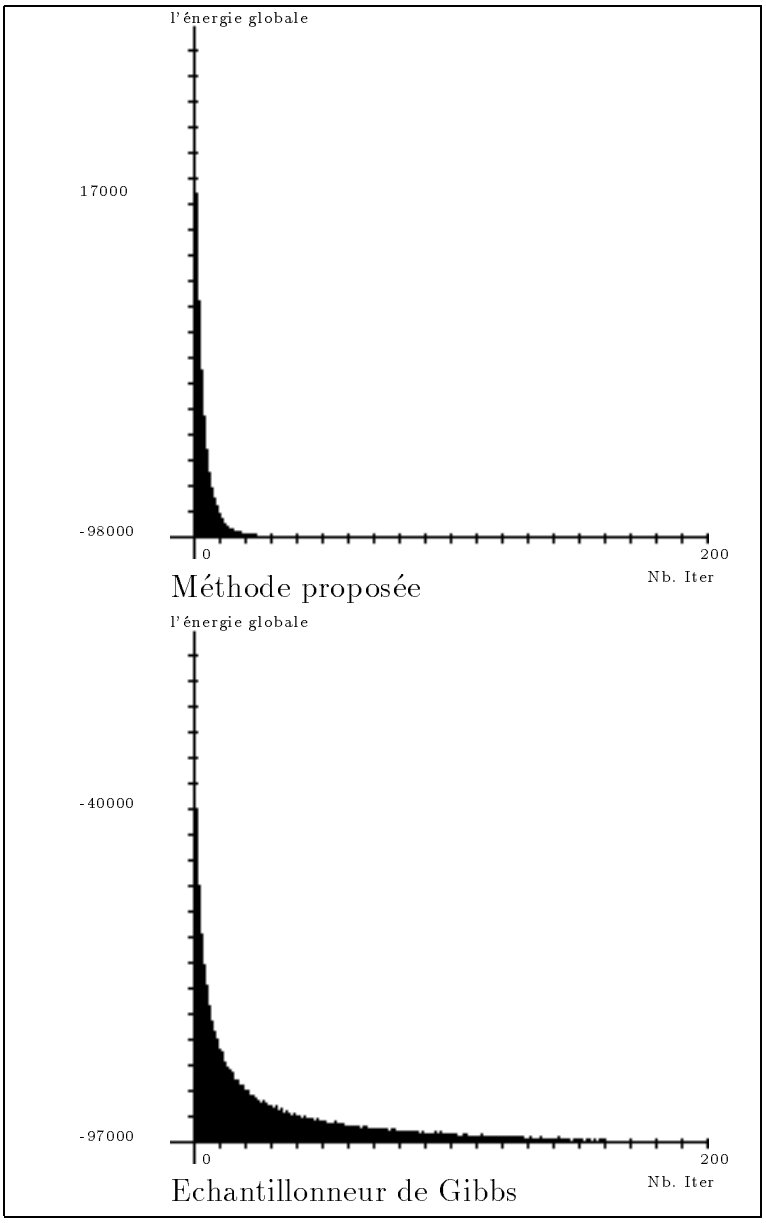


Figure 13 : L'énergie globale en fonction du nombre d'itérations (image SPOT 300×128)

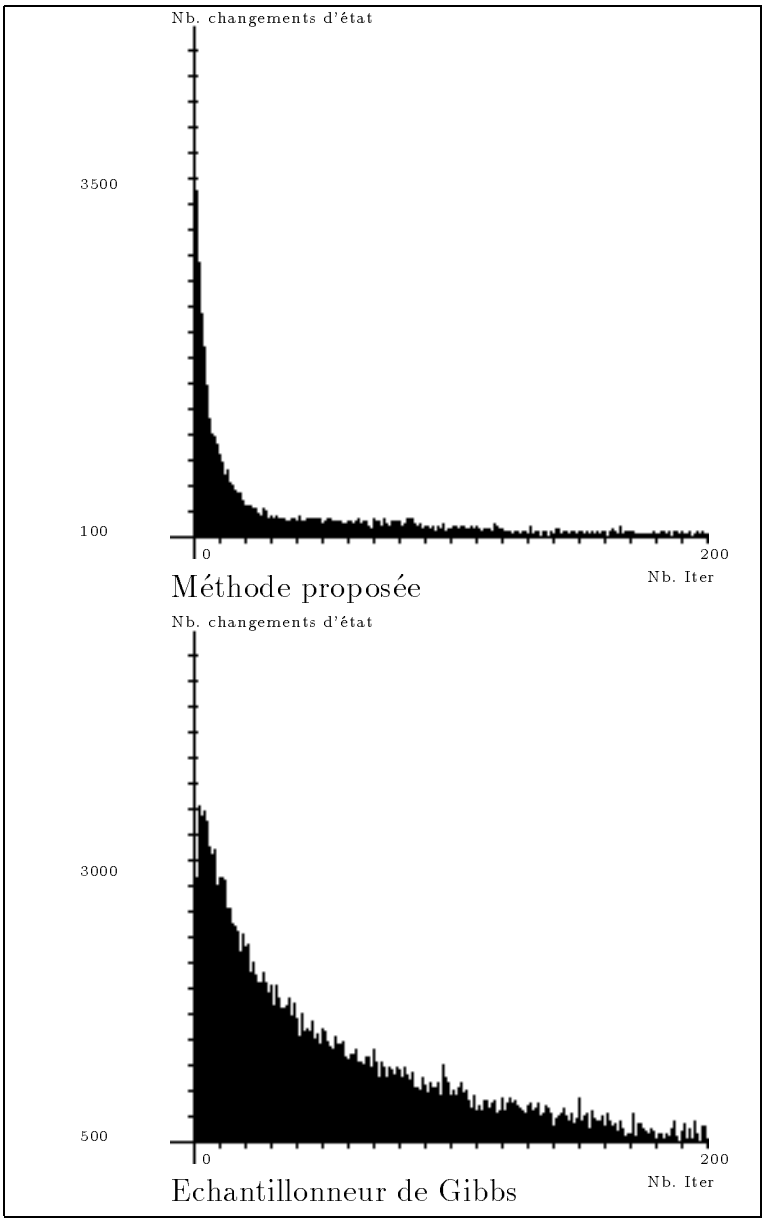


Figure 12 : Le nombre de sites en fonction du nombre d'itérations où la nouvelle énergie est plus grande que la précédente (image SPOT 128×128)

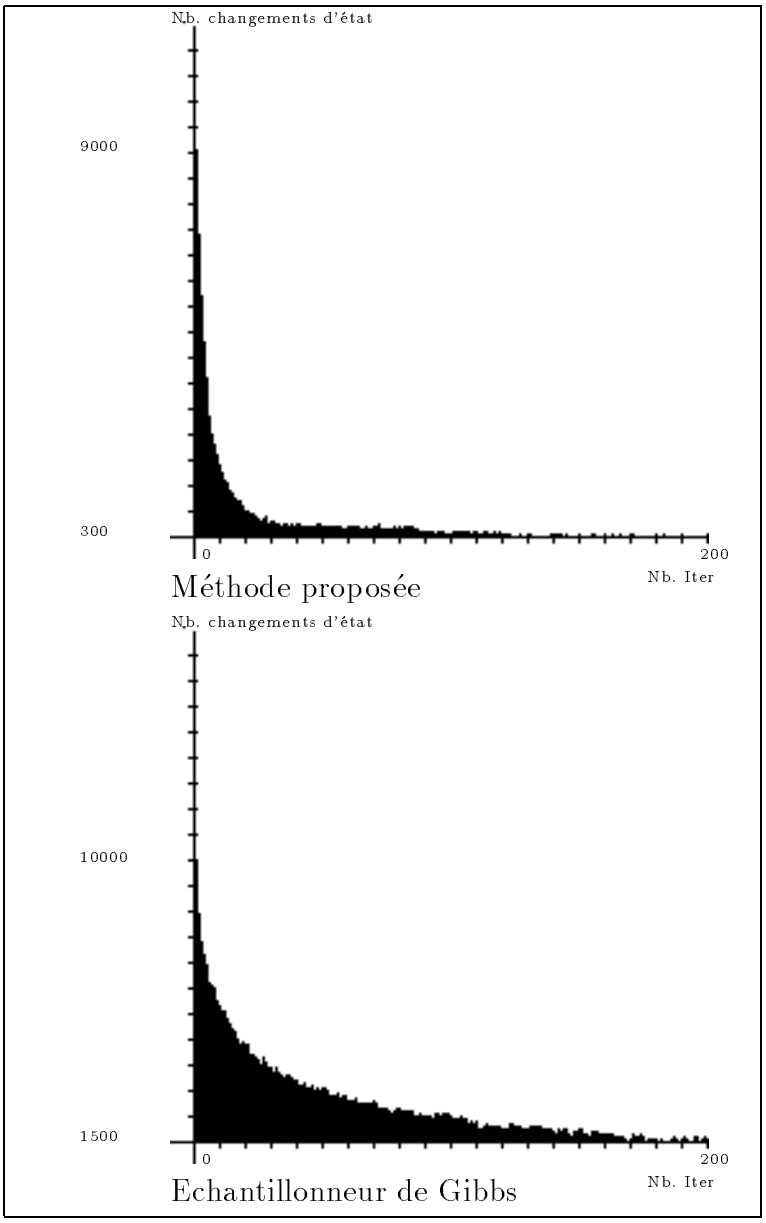


Figure 11 : Le nombre de changements d'état en fonction du nombre d'itérations (image SPOT 128×128)

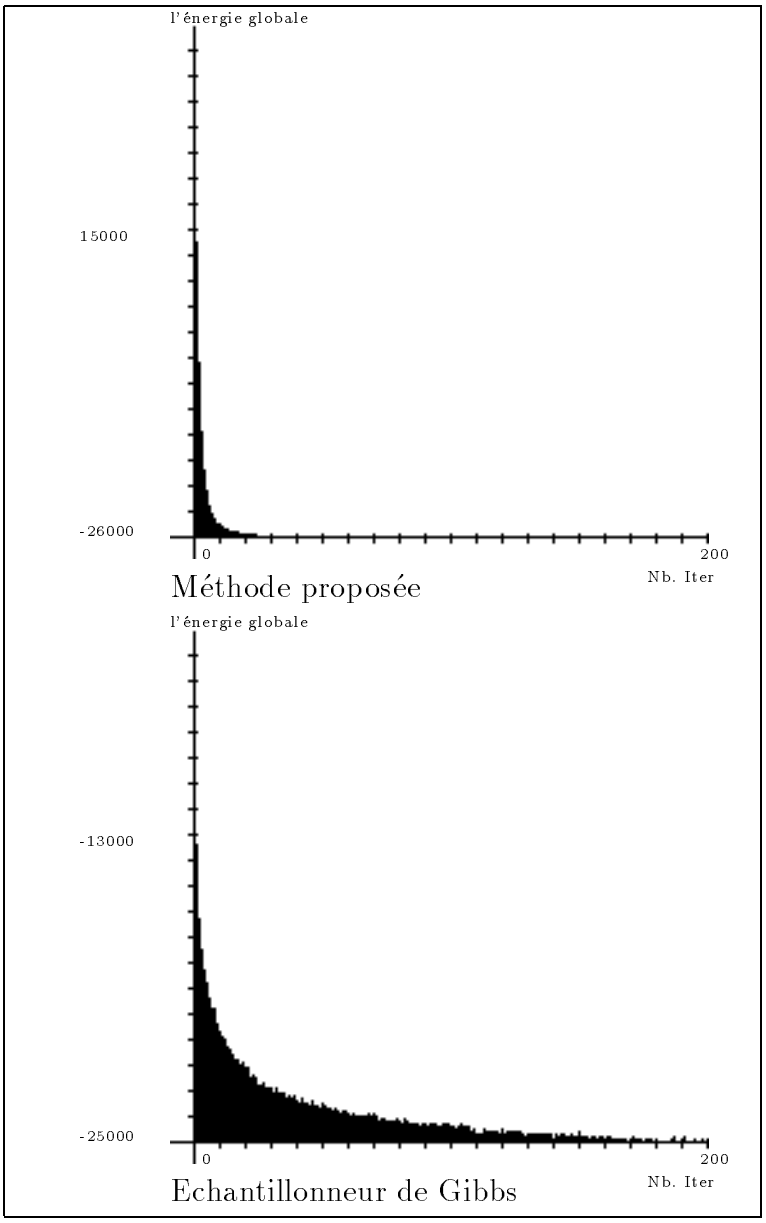


Figure 10 : L'énergie globale en fonction du nombre d'itérations (image SPOT 128×128)

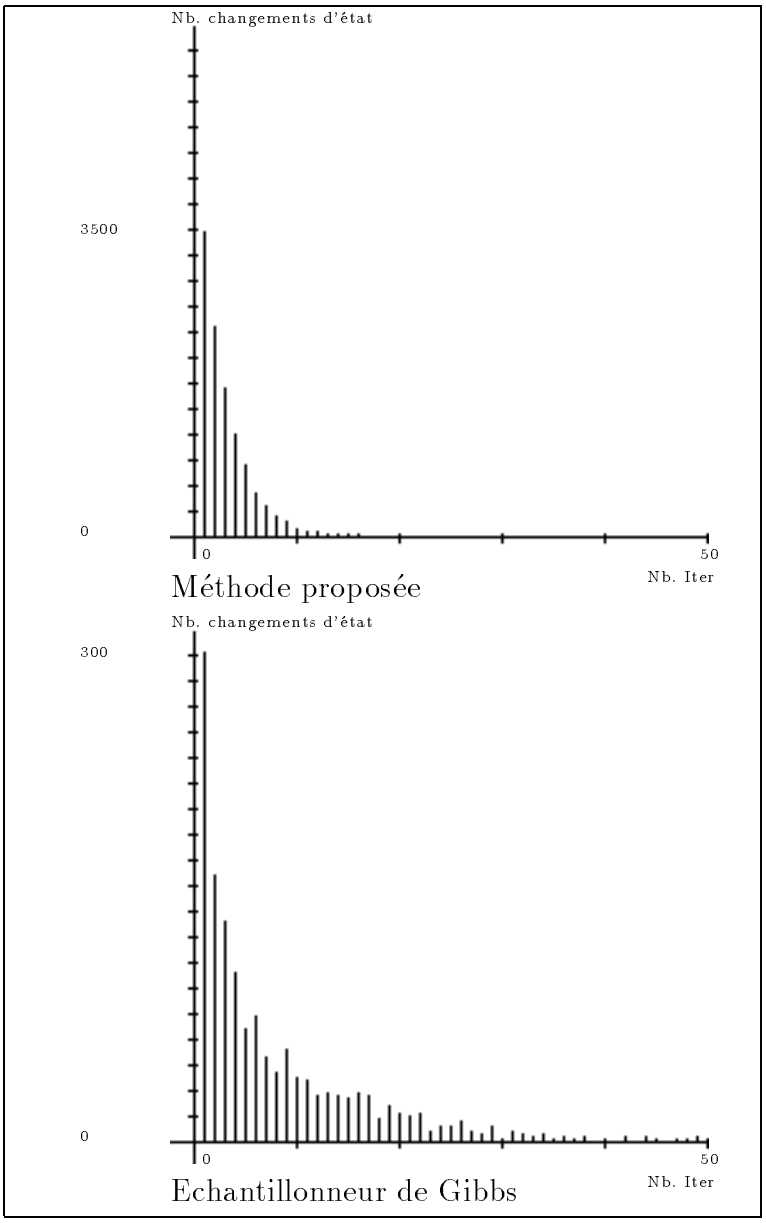


Figure 9 : Le nombre de sites en fonction du nombre d'itérations où la nouvelle énergie est plus grande que la précédente (image synthétique 128×128)

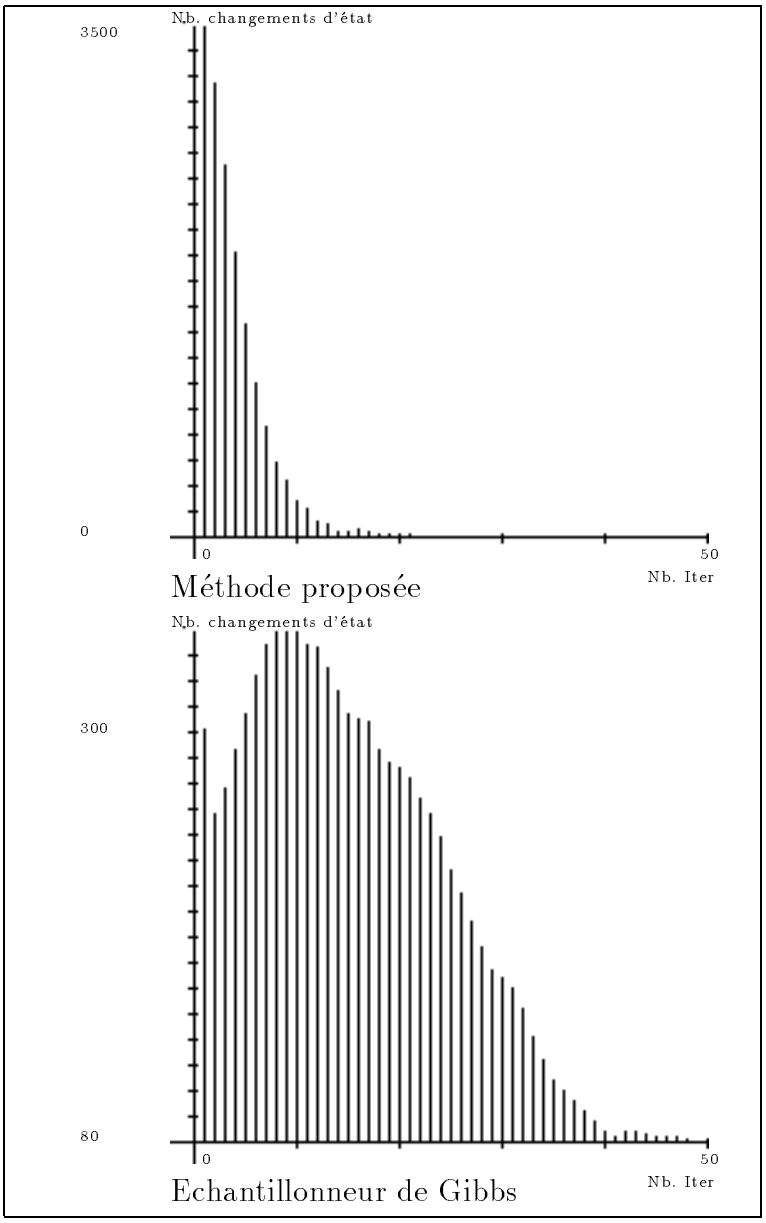


Figure 8 : Le nombre de changements d'état en fonction du nombre d'itérations (image synthétique 128×128)

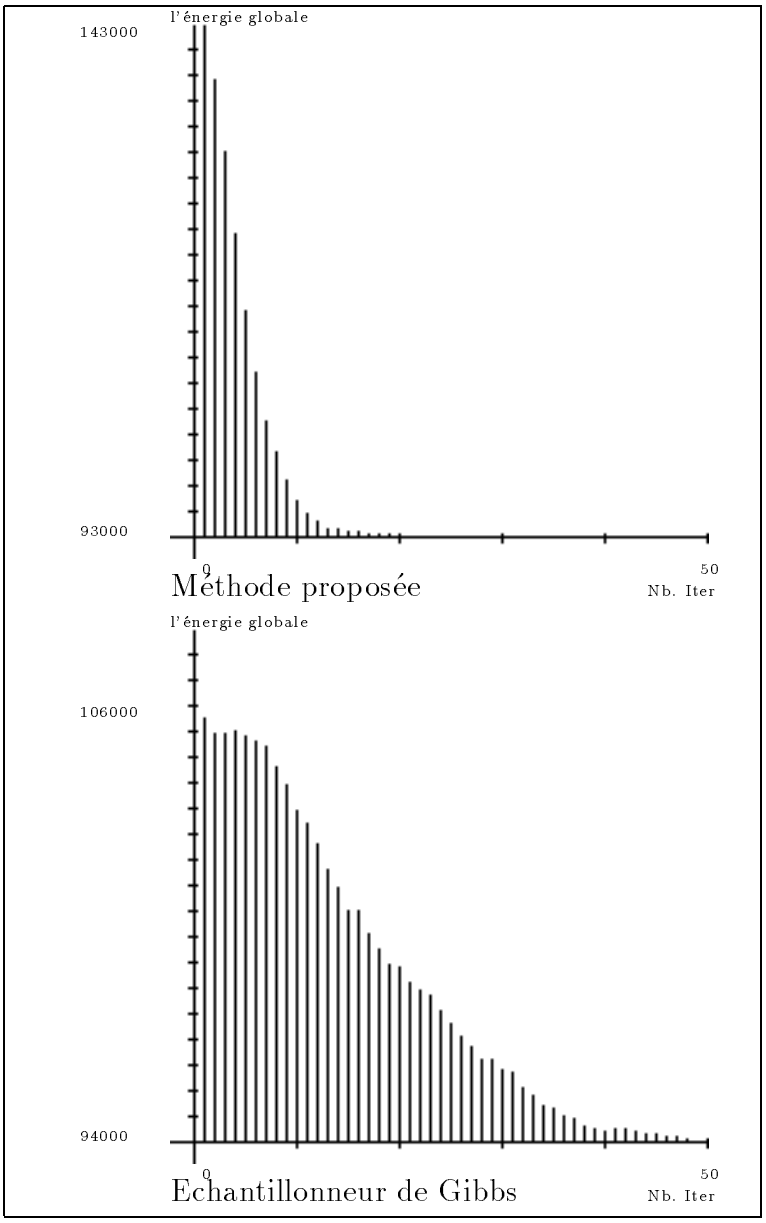


Figure 7 : L'énergie globale en fonction du nombre d'itérations (image synthétique 128×128)

A.2 Figures

En conséquence de (24), pour $\forall \lambda, \kappa \in \Omega$ tels que $\mathcal{E}(\lambda) < \mathcal{E}(\kappa)$, on a:

$$\exp\left(-\frac{\mathcal{E}(\kappa) - \mathcal{E}(\lambda)}{T(k)}\right) \leq \exp\left(-\frac{\Delta\mathcal{E}_{min}}{T(k)}\right)$$

et en considerant (23), on obtient:

$$\lim_{k \to \infty} \exp\left(-\frac{\Delta\mathcal{E}_{min}}{T(k)}\right) = 0 \quad .$$

Alors il existe un seuil T_α dépendant de α tel que si $T(k) < T_\alpha$ alors:

$$\alpha \geq \exp\left(-\frac{\Delta\mathcal{E}_{min}}{T(k)}\right) \geq \exp\left(-\frac{\mathcal{E}(\kappa) - \mathcal{E}(\lambda)}{T(k)}\right) \quad (\forall \lambda, \kappa \in \Omega : \mathcal{E}(\lambda) < \mathcal{E}(\kappa)) \tag{26}$$

C'est à dire que l'on accepte uniquement les états de descente d'énergie si $T(k) < T_\alpha$. □

Il y a deux conséquences au théorème précédent:

1. $$\alpha = \exp\left(-\frac{\Delta\mathcal{E}_{min}}{T_\alpha}\right)$$

   ou bien

   $$T_\alpha = \exp\left(-\frac{\Delta\mathcal{E}_{min}}{\log \alpha}\right)$$

   C'est qui fait la liaison entre α et la température.

2. Si

   $$T(k) = \frac{K}{\log(1+k)}$$

   alors $T(k) < T_\alpha$ si et seulement si $k > K_\alpha$ où K_α est un seuil déterminé par

   $$K_\alpha = \exp\left(-\frac{K \log \alpha}{\Delta\mathcal{E}_{min}}\right) - 1$$

   C'est à dire, après K_α itérations, l'algorithme va accepter les états de descente d'énergie.

A Résultats de convergence

A.1 Preuve de la convergence

Dans ce paragraphe, nous donnons les résultats théoriques de la convergence du nouvel algorithme proposé au paragraphe 1.4.

Soit Ω l'ensemble de tous les états possibles et $\mathcal{E}(\omega)$ ($\omega \in \Omega$) l'énergie globale du ω. Considerons la règle suivante d'accepter la nouvelle configuration κ si l'état courant est λ:

$$\lambda \longrightarrow \kappa \iff \begin{cases} \mathcal{E}(\lambda) \geq \mathcal{E}(\kappa) & \text{(descente d'énergie)} \\ \mathcal{E}(\lambda) < \mathcal{E}(\kappa) \text{ et } \alpha < \exp\left(-\frac{\mathcal{E}(\kappa)-\mathcal{E}(\lambda)}{T(k)}\right) & \text{(incrément d'énergie)} \end{cases} \tag{22}$$

où $\alpha \in (0,1)$ est constant et k est le nombre d'itérations.

Pour la convergence vers un minimum local, il suffit de prouver qu'après un certain nombre d'itérations, seuls les états de descente d'énergie sont acceptés.

Théorème 4 *Pour $\alpha \in (0,1)$ quelconque, il existe un seuil de température T_α dépendant de α tel que si $T(k) < T_\alpha$ alors l'algorithme accepte uniquement les états de descente d'énergie globale. Donc si*

$$\lim_{k\to\infty} T(k) = 0 \tag{23}$$

alors les états ω^k produits par l'algorithme convergent vers un minimum local $\hat{\omega}_{loc}$:

$$\lim_{k\to\infty} \omega^k = \hat{\omega}_{loc}$$

Preuve:
Soit

$$\Delta\mathcal{E}_{min} = \min_{\lambda,\kappa\in\Omega:\mathcal{E}(\lambda)\neq\mathcal{E}(\kappa)} \mid \mathcal{E}(\lambda) - \mathcal{E}(\kappa) \mid \tag{24}$$

et $\lambda, \kappa \in \Omega$ des états quelconques tels que $\mathcal{E}(\lambda) < \mathcal{E}(\kappa)$. En supposant que $\omega^{k-1} = \lambda$,

$$\omega^k = \kappa \iff \alpha < \exp\left(-\frac{\mathcal{E}(\kappa) - \mathcal{E}(\lambda)}{T(k)}\right) \tag{25}$$

Annexes

4 Conclusion

Dans la première partie de ce rapport, nous avons présenté un modèle Markovien pour la classification d'image. Le principal avantage de ce modèle est d'exprimer la vraisemblance de l'observation et l'influence du contexte en une seule expression qui correspond à une fonction énergie. En minimisant cette fonction non-convexe, on obtient l'étiguetage le plus probable.

Cette minimisation est fait à l'aide des méthodes de relaxation. Nous avons mis en œuvre trois méthodes connues de relaxation (l'algorithme de Metropolis, l'échantillonneur de Gibbs et l'ICM) et nous avons proposé, dans la deuxième partie, une méthode pseudo-stochastique. Cet algorithme est plus rapide que les méthodes stochastiques, mais seule la convergence vers un minimum local est prouvée. Par contre, il est moins rapide que l'ICM, mais il donne de bons résultats à partir de n'importe quelle configuration initiale (à condition de bien choisir le seuil α!) ce qui n'est pas le cas avec l'ICM.

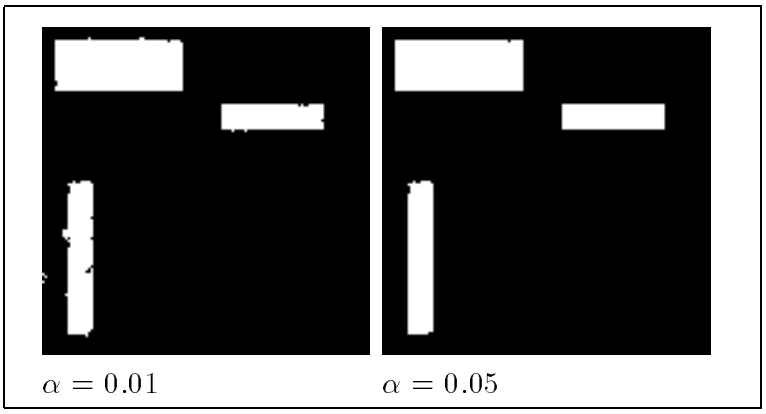


Figure 6 : Résultats à $T(1) \approx 16$ $(K = 5)$

- Il faut choisir un valeur faible (proche de 0), si l'image de départ est très bruitée ou la température initiale est élevée.
- Il faut utiliser une valeur plus grande, si l'image de départ est de bonne qualité (voir figure 5) ou la température initiale est faible (voir figure 6).

Il faut remarquer que α dépend plus fortement de la qualité d'image que de la température initiale.

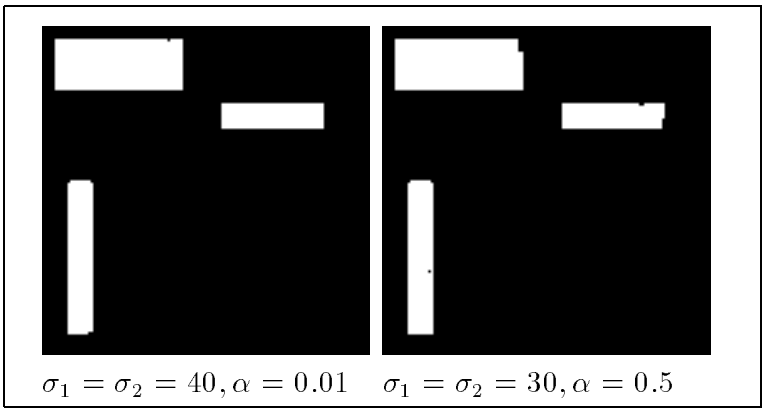


Figure 5 : Résultats sur deux images synthétiques

- Le nombre de changements d'état de départ et final (*CD et CF*).
- Le nombre de sites de départ et final, où l'énergie locale du nouvel état est plus grande que la précédente (*GD et GF*).

En ce qui concerne la rapidité de convergence, les figures 7–9 montrent les résultats de l'échantillonneur de Gibbs et de la méthode proposée sur l'image synthétique, les figures 10–12 sur l'image SPOT 128×128 et les figures 13–15 sur l'image SPOT 300×128 (voir annexe A). On peut remarquer, en fonction du nombre d'itérations, le changement des énergies globales (figures 7, 10, 13), le nombre de changements d'états (figures 8, 11, 14) et le nombre de sites où l'énergie locale du nouvel état est plus grande que la précédente (figures 9, 12, 15).

Sur la figure 4, nous présentons les résultats de l'échantillonneur de Gibbs et de l'algorithme proposé avec un critère d'arrêt 0.1% (c'est à dire l'algorithme s'arrête, si le nombre de modification d'états est plus petit que 0.1% du nombre de sites). Pour l'échantillonneur de Gibbs, l'arrêt s'est accompli après *95* itérations et pour la méthode proposée après *24* itérations.

3.4 Le choix du seuil α

La difficulté essentielle de la méthode proposée est de bien choisir le seuil α décrit au paragraphe 2.4. Il n'existe pas de formule explicite pour α, mais nous avons trouvé quelques liaisons entre α, la qualité d'image de départ et la température initiale:

	Nb. Iter	$\mathcal{E}_{glob}$ dép.	$\mathcal{E}_{glob}$ fin.	CD	CF	GD	GF
Metrop.	200	-22617	-24670	14684	2588	4989	878
Gibbs	200	-13298	-24949	9956	1705	2823	812
ICM	4	-25934	-25938	162	0	0	0
MMD	200	14355	-25848	8266	349	3348	166

Table 6 : La convergence sur l'image SPOT 128×128

	Nb. Iter	$\mathcal{E}_{glob}$ dép.	$\mathcal{E}_{glob}$ fin.	CD	CF	GD	GF
Metrop.	200	20021	-95690	24083	11064	6116	675
Gibbs	200	-40488	-96455	25854	4050	9648	1984
ICM	5	-98762	-98768	464	0	0	0
MMD	200	16470	-98567	16111	493	7702	235

Table 7 : La convergence sur l'image SPOT 300×128

Echantillonneur de Gibbs Méthode proposée

Figure 4 : Résultats obtenus avec un critère d'arrêt 0.1%

	Nb. Iter	$\mathcal{E}_{glob}$ dép.	$\mathcal{E}_{glob}$ fin.	CD	CF	GD	GF
Metrop.	50	105109	100542	294	738	293	2
Gibbs	50	105740	94411	297	88	296	6
ICM	5	96791	94000	2187	0	0	0
MMD	50	142462	93795	3462	2	3462	2

Table 5 : La convergence sur l'image synthétique 128×128

- Le temps par itération (temps CM + temps "Front End").

On peut constater que l'ICM est le plus rapide. C'est n'est pas surprenant puisque c'est un algorithme déterministe et son critère d'arrêt se produit en un nombre généralement faible d'itérations. De même, il faut remarquer qu'en temps par itération, l'algorithme proposé est le plus rapide. C'est une conséquence de la différence entre ces algorithmes, plus exactement, entre les méthodes de selection du nouvel état:

- Pour l'ICM, il faut calculer l'énergie locale de tous les états possibles en un site. Alors le coût de calcul est $O(L)$, où L est le nombre d'états.
- Pour l'échantillonneur de Gibbs, au delà du calcul des énergies locales, il faut fabriquer des nombres aléatoires. Alors le coût de calcul est $O(L)+\tau$, où τ est le coût à payer pour fabriquer les nombres aléatoires.
- Pour l'algorithme de Metropolis, le coût de calcul ne dépend pas de L mais il faut fabriquer deux fois des nombres aléatoires. Alors le coût de calcul est 2τ.
- Pour la méthode proposée, il y a seulement une fabriquation de nombres aléatoires et le coût de calcul ne dépend pas de L, donc cette fois le coût de calcul est τ.

3.3.3 Comparison de la convergence

En ce qui concerne la convergence, les tables 5, 6, 7 montrent les résultats obtenus par les quatre algorithmes. Pour chacune des images, nous donnons:

- Le nombre d'itérations.
- L'énergie globale de départ et finale avec $T = 1$.

	VPR	Nb. Iter	Tps. Total	Tps. par It.
Metrop.	2	50	212.1 sec.	4.2 sec.
Gibbs	2	50	346.1 sec.	6.9 sec.
ICM	2	5	15.7 sec.	3.1 sec.
MMD	2	50	111.3 sec.	2.2 sec.

Table 2 : Les temps sur l'image synthétique 128×128

	VPR	Nb. Iter	Tps. Total	Tps. par It.
Metrop.	2	200	816.7 sec.	4.1 sec.
Gibbs	2	200	1353.4 sec.	6.8 sec.
ICM	2	4	12.8 sec.	3.2 sec.
MMD	2	200	422.7 sec.	2.1 sec.

Table 3 : Les temps sur l'image SPOT 128×128

3.3.2 Comparison des temps de calcul

Nous comparons les algorithmes mis en œuvre sur la machine à connexions. Les tables 2, 3, 4 montrent les résultats obtenus sur les trois images se trouvant dans l'annexe B. Pour chacune de cettes images, nous donnons:

- Le nombre de processeurs virtuels par processeur réel ou le nombre de pixels par processeur (*VPR*).
- Le temps total (temps CM + temps "Front End").
- Le nombre d'itérations des algorithmes.

	VPR	Nb. Iter	Tps. Total	Tps. par It.
Metrop.	3	200	835.9 sec.	4.2 sec
Gibbs	3	200	2298.4 sec	11.5 sec
ICM	3	5	26.2 sec.	5.2 sec.
MMD	3	200	445.7sec.	2.2 sec.

Table 4 : Les temps sur l'image SPOT 300×128

$$\begin{aligned}\mathcal{E}_{glob}(\omega) &= \sum_{s=(i,j)\in\mathcal{S}} \left(\log\sqrt{2\pi}\,\sigma_{\omega_s} + \frac{(f_s-\mu_{\omega_s})^2}{2\sigma_{\omega_s}^2} \right) + \\ &\sum_{s=(i,j)\in\mathcal{S}:1\le i<N} E_{\{(i,j),(i+1,j)\}}(\omega_{(i,j)},\omega_{(i+1,j)}) + \\ &\sum_{s=(i,j)\in\mathcal{S}:1\le j<M} E_{\{(i,j),(i,j+1)\}}(\omega_{(i,j)},\omega_{(i,j+1)}) \end{aligned} \tag{20}$$

$\mathcal{E}_s(\omega)$ représente l'énergie locale de l'état ω en s:

$$\mathcal{E}_s(\omega) = \log\sqrt{2\pi}\,\sigma_{\omega_s} + \frac{(f_s-\mu_{\omega_s})^2}{2\sigma_{\omega_s}^2} + \sum_{r\in\mathcal{V}_s} E_{\{s,r\}}(\omega_s,\omega_r) \tag{21}$$

3.3 Comparison des algorithmes

3.3.1 Choix des paramètres

Pour les méthodes stochastiques et pour la méthode proposée, nous avons utilisé la loi de décroissance de la température suivante:

$$T(k) = \frac{K}{\log(1+k)} \quad (k = 1, 2, \ldots)$$

où k est le numéro de l'itération et K est une constante déterminée par la température initiale choisi. Après avoir testé différentes valeurs, nous avons choisi $K = 3 \Leftrightarrow T(1) \approx 4.3$ qui donne de bons résultats sur toutes les images (voir annexe B).

Pour l'ICM, nous avons utilisé un critère d'arrêt simple, qui consiste à compter le nombre de modification d'états lors d'une itération et à s'arrêter si ce nombre est nul. Pour les autres méthodes, nous avons utilisé un nombre déterminé d'itérations, notamment 200 pour les images SPOT et 50 pour l'image synthétique. Pour l'échantillonneur de Gibbs et la méthode proposée, nous avons utilisé aussi un critère d'arrêt similaire à celui de l'ICM, qui consiste à s'arrêter si le nombre de modification d'états est faible (0.1% du nombre de sites).

Pour la nouvelle méthode, nous avons choisi $\alpha = 0.5$ dans le cas des images SPOT et $\alpha = 0.01$ dans le cas de l'image synthétique. Ces valeurs ont été déterminées empiriquement et donnent les meilleurs résultats sur ces images.

Alors la fonction à minimiser est la suivante:

$$\hat{\omega} = \min_{\omega \in \Omega} \left(\sum_{s \in \mathcal{S}} \left(\log \sqrt{2\pi}\, \sigma_{\omega_s} + \frac{(f_s - \mu_{\omega_s})^2}{2\sigma_{\omega_s}^2} \right) + \sum_{C \in \mathcal{C}} E_C(\omega) \right)$$

La valeur moyenne et l'écart-type des classes ont été déterminés à partir d'un ensemble d'apprentissage (apprentissage supervisé).

Nous avons testé les algorithmes sur trois images (pour la valeur moyenne et l'écart-type, voir la table 1):

- image synthétique 128×128 avec deux classes (voir annexe B),
- image satellite SPOT 128×128 avec deux classes et
- image satellite SPOT 300×128 avec quatre classes.

Les fonctions d'énergies potentielles ont été déterminées de la façon suivante:

- Pour l'image synthétique, nous avons utilisé les fonctions suivantes:

$$\begin{aligned} \mathcal{E}_{\{s\}}(\omega) &= 0 \\ \mathcal{E}_{\{s,r\}}(\omega_s, \omega_r) &= \begin{cases} 0.011757 = -\log P(\omega_s = \lambda \text{et} \omega_r = \lambda) & \text{si } \omega_s = \omega_r \\ 4.449193 = -\log P(\omega_s = \lambda \text{et} \omega_r = \kappa) & \text{si } \omega_s \neq \omega_r \end{cases} \\ & \quad (\lambda \neq \kappa \ ; \ \ \lambda, \kappa \in \{0, 1\}) \end{aligned}$$

- Pour les images réelles, après avoir testé plusieurs fonctions, nous avons utilisé les fonctions suivantes:

$$\begin{aligned} \mathcal{E}_{\{s\}}(\omega) &= 0 \\ \mathcal{E}_{\{s,r\}}(\omega_s, \omega_r) &= \begin{cases} -0.1 & \text{si } \omega_s = \omega_r \\ 0.1 & \text{si } \omega_s \neq \omega_r \end{cases} \end{aligned}$$

Les énergies globale et locale sont définis par:

Image	μ_1	σ_1	μ_2	σ_2	μ_3	σ_3	μ_4	σ_4
synthétique	65	40	190	40	—	—	—	—
SPOT 128×128	0.224	0.0005	0.241	0.0072	—	—	—	—
SPOT 300×128	0.167	0.0006	0.120	0.0001	0.375	0.0022	0.228	0.0024

Table 1 : Les valeurs moyennes et les écart-types des classes

3 Résultats expérimentaux

3.1 Mise en œuvre sur la CM-2

Tous les algorithmes décrits dans la partie précédente ont été implémentés sur la machine à connexions CM-2 [18]. C'est une machine *SIMD* ("Single Instruction Multiple Data") ayant de *8K* à *64K* processeurs (les algorithmes ont été réalisés sur *8K*). Chaque processeur est un processeur *1* bit disposant de *32K* octets de mémoire locale. L'utilisateur a une possibilité de définir dynamiquement une géométrie particulière pour l'ensemble des processeurs adaptée à son application. Cette géométrie décrit l'organisation logique des processeurs: le nombre de dimensions de la structure et le nombre de processeurs par dimension. Grâce au mécanisme des processeurs virtuels, il peut s'affranchir de la limite imposée par le nombre de processeurs physiques.

Cette mise en œuvre a été réalisée en *C** [19] qui est une extension de *ANSI C* pour permettre la programmation parallèle. Le type de communication inter-processeurs que l'on utilise le plus souvent en traitement d'image bas niveau est la communication locale appelée *NEWS* ("North, Est, West, South"). Nous l'avons utilisée aussi pour réaliser la communication entre les pixels voisins.

3.2 Le modèle réalisé

Nous avons réalisé un cas particulier du modèle décrit dans le paragraphe 1.4. Soit:

$$\mathcal{V} = \{\mathcal{V}_{(i,j)} \mid (i,j) \in \mathcal{S}\},\ \mathcal{V}_{(i,j)} = \{(k,l) \in \mathcal{S} \mid (k-i)^2 + (l-j)^2 = 1\} \quad (19)$$

C'est à dire un voisinage de premier ordre (voir figure 1), et

$$\mid \vec{f_s} \mid = 1 \quad , f_s = \text{valeur de l'intensité du pixel s.}$$

Les désavantages sont:

- La difficulté de bien choisir le seuil α.
- C'est moins rapide que l'ICM et donne presque le même résultat.

Dans la partie suivante, nous présentons quelques résultats expérimentaux et nous comparons la nouvelle méthode avec les trois autres.

3. En chaque site s où $\eta_s \neq \omega_s^k$, on calcule l'énergie locale $\mathcal{E}_s^{\eta'}$ à l'aide de la formule (16) en supposant que

$$\eta'_r = \begin{cases} \eta_s & \text{si } r = s \\ \omega_r^k & \text{si } r \neq s \end{cases} . \tag{18}$$

Il y a donc certains sites où on ne fait rien. Soit $\Delta\mathcal{E}_s = \mathcal{E}_s^{\eta'} - \mathcal{E}_s^{\omega^k}$. La règle d'accepter le nouvel état en s est la suivante:

$$\omega_s^{k+1} = \begin{cases} \eta_s & \text{si } \Delta\mathcal{E}_s \leq 0 \text{ ou } \Delta\mathcal{E}_s > 0 \text{ et } \alpha \leq \exp\left(-\frac{\Delta\mathcal{E}_s}{T}\right) \\ \omega_s^k & \text{sinon} \end{cases}$$

où α est un *seuil fixé* au début de l'algorithme entre 0 et 1. Celui-ci reste constant pendant toutes les itérations.

4. Faire décroitre la température: $k = k + 1$, $T = T(k)$ (k est le numéro de l'itération).

5. Poursuivre l'étape 2. pour un nombre déterminé d'itérations ou jusqu'à la réalisation d'un critère d'arrêt.

Au delà des difficultés se trouvant dans l'algorithme de Metropolis et l'échantillonneur de Gibbs, il y a un nouveau problème: la détermination du seuil α. Il n'y a aucune formule explicite pour α. En règle générale, dans le cas de classification d'image, il est conseillé d'utiliser une valeur faible (proche de 0) si l'image de départ est très bruitée. Mais dans le cas contraire, une valeur plus grande est suffisante ($\alpha \approx 0.5$).

Les avantages de la méthode sont les suivants:

- En temp de calcul, c'est moins coûteux que les méthodes stochastiques.
- La convergence est plus rapide que pour les méthodes stochastiques. C'est à dire, il suffit d'utiliser un faible nombre d'iterations pour obtenir le même résultat, mais il faut noter que seul la convergence vers un minimum local est prouvée en annexe A.
- Elle donne de bons résultats à partir de n'importe quelle configuration initiale. Par contre, l'ICM donne de bons résultats à condition de bien choisir les données initiales.

2. Pour assurer la convergence, on divise le réseau $\mathcal{S}$ en deux sous-réseaux $\mathcal{S}_1$ et $\mathcal{S}_2$ tels que pour $s \in \mathcal{S}_i$ $(i = 1, 2)$ quelconque, on aie (voir figure 3):

$$\begin{aligned} \mathcal{S}_1 &= \{s = (i,j) \mid i \equiv j \pmod 2\} \\ \mathcal{S}_2 &= \mathcal{S} - \mathcal{S}_1 \, . \end{aligned}$$

3. Pour chaque site $s \in \mathcal{S}_1$ (respectivement $s \in \mathcal{S}_2$), on choisit l'état le plus probable, c'est à dire:

$$\omega_s^{k+1} = \lambda \iff \lambda = \max_{\iota \in \Lambda} P_s^\iota \quad \text{où}$$

$$P_s^\iota = \exp\left(-\frac{\mathcal{E}_s^\iota}{T}\right)$$

$\mathcal{E}_s^\iota$ est l'énergie locale en s correspondant à $\iota \in \Lambda$.

4. Poursuivre l'étape 3. pour un nombre déterminé d'itérations ou jusqu'à la réalisation d'un critère d'arrêt.

Cet algorithme converge toujours vers un minimum local.

2.4 La méthode proposée ("Modified Metropolis Dynamics")

Le nouvel algorithme proposé est basé sur le schèma de Metropolis mais c'est une méthode pseudo-stochastique dans le sens où:

- Le choix du nouvel état est fait au hasard avec une distribution uniforme.
- La règle d'accepter le nouvel état est déterministe.

C'est dans ce sens qu'il s'agit d'une méthode pseudo-stochastique. L'algorithme parallèle est le suivant:

1. Tirer aléatoirement une configuration initiale ω^0, $k = 1$ et $T = T(1)$.
2. On tire au hasard (avec une distribution uniforme) un état global η tel que

$$\forall s \in \mathcal{S} \; : \eta_s \in \Lambda \, .$$

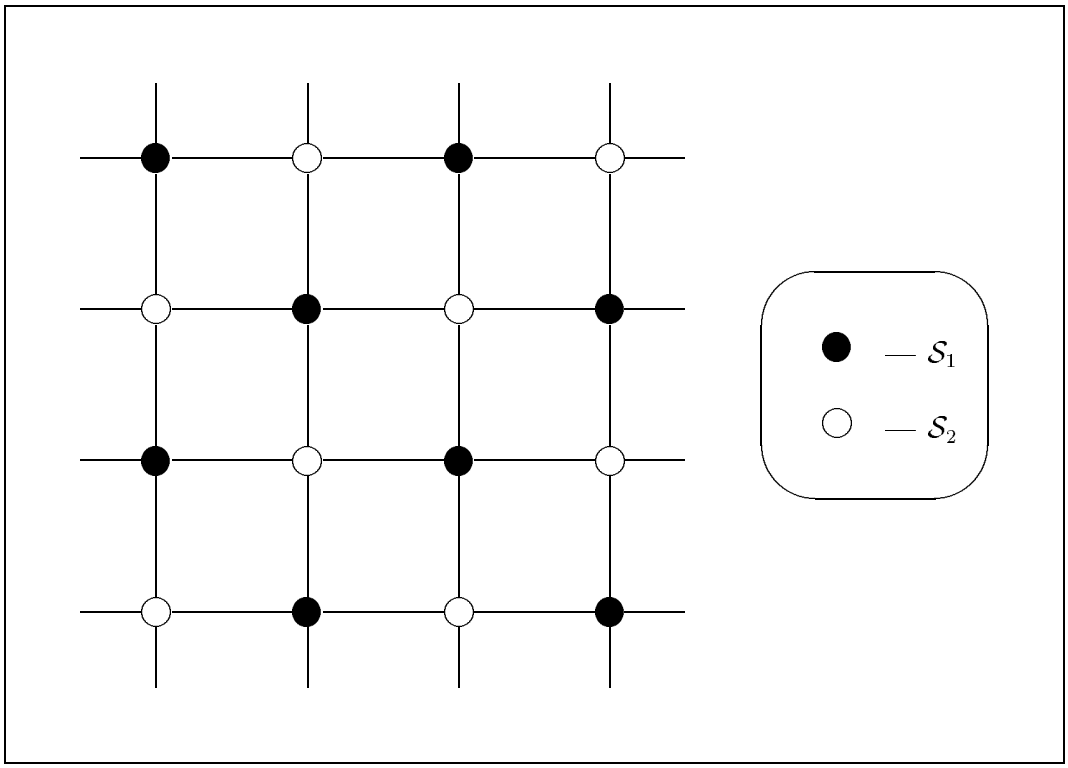


Figure 3 : Les deux sous-réseaux de $\mathcal{S}$ dans le cas de voisinage d'ordre un

3. Faire décroitre la température: $k = k + 1$, $T = T(k)$ (k est le numéro de l'itération).

4. Poursuivre l'étape 2. pour un nombre déterminé d'itérations ou jusqu'à la réalisation d'un critère d'arrêt.

L'algorithme converge asymptotiquement vers le minimum global d'énergie avec des conditions similaires à celles requises pour la convergence de la méthode de Metropolis.

2.3 L'ICM ("Iterated Conditional Mode")

C'est un algorithme déterministe proposé par Besag [4]. Il n'y a aucun tirage aléatoire, on ne réalise qu'une descente dans un puit d'énergie déterminé par la configuration initiale et le mode de balayage des sites. Une version parallèle de cet algorithme est la suivante:

1. Choisir une configuration initiale ω^0, aussi proche que possible d'une configuration optimale; $k = 1$, $T = 1$.

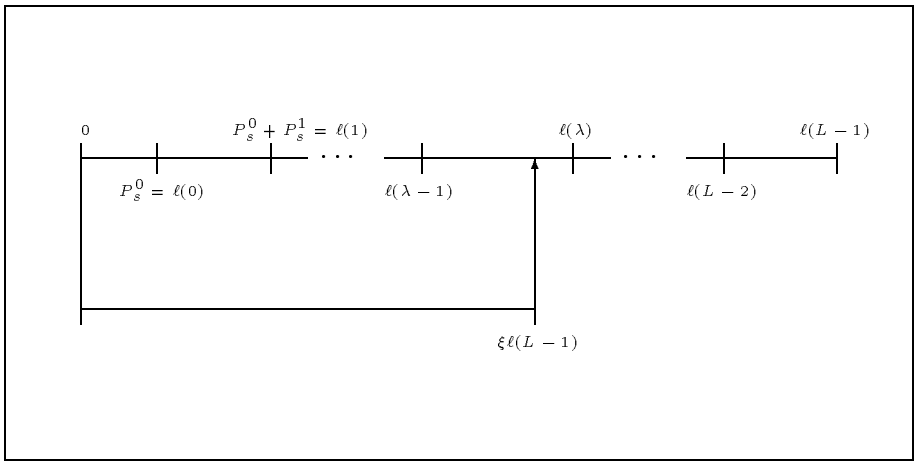


Figure 2 : La sélection du nouvel état

L'échantillonneur de Gibbs introduit par D. Geman et S. Geman [9] est connu comme la meilleure méthode stochastique. Elle permet d'effectuer des simulations d'un champ de Markov.

Considérons la loi de probabilité $P(\omega \mid F)$ définit par (14), le but est d'échantillonner cette loi et de trouver l'estimateur MAP $\hat{\omega}$ de ω. Une variante parallèle de cet algorithme est la suivante:

1. Tirer aléatoirement une configuration initiale ω^0, $k = 1$ et $T = T(1)$.

2. En chaque site s, on calcule l'énergie locale $\mathcal{E}_s^\lambda$ $(\forall \lambda \in \Lambda)$ à l'aide de la formule (16). Ensuite un tirage aléatoire permet de sélectionner l'étiquette, qui sera attribuée à ce site:

$$\omega_s^{k+1} = \lambda \iff \underbrace{\sum_{\iota=0}^{\lambda-1} P_s^\iota}_{=\ell(\lambda-1)} < \underbrace{\xi \sum_{\iota=0}^{L-1} P_s^\iota}_{=\xi\ell(L-1)} \leq \underbrace{\sum_{\iota=0}^{\lambda} P_s^\iota}_{=\ell(\lambda)}$$

où

$$P_s^\lambda = \exp\left(-\frac{\mathcal{E}_s^\lambda}{T}\right) \quad (\lambda \in \Lambda)$$

et $\xi \in [0,1)$ est une variable aléatoire de distribution uniforme (voir figure 2).

3. En chaque site s, on calcule l'énergie locale $\mathcal{E}_s^{\eta'}$ à l'aide de la formule (16) en supposant que

$$\eta'_r = \begin{cases} \eta_s & \text{si } r = s \\ \omega_r^k & \text{si } r \neq s \end{cases} . \tag{17}$$

Soit $\Delta\mathcal{E}_s = \mathcal{E}_s^{\eta'} - \mathcal{E}_s^{\omega^k}$. La règle d'accepter le nouvel état en s est la suivante:

$$\omega_s^{k+1} = \begin{cases} \eta_s & \text{si } \Delta\mathcal{E}_s \leq 0 \text{ ou } \Delta\mathcal{E}_s > 0 \text{ et } \xi \leq \exp\left(-\frac{\Delta\mathcal{E}_s}{T}\right) \\ \omega_s^k & \text{si non} \end{cases}$$

où $\xi \in [0,1)$ est une variable aléatoire de distribution uniforme. Il y a donc une probabilité non nulle de continuer avec un état plus coûteux que l'état courant.

4. Faire décroitre la température: $k = k+1$, $T = T(k)$ (k est le numéro de l'itération).

5. Poursuivre l'étape 2. pour un nombre déterminé d'itérations ou jusqu'à la réalisation d'un critère d'arrêt.

Il est prouvé que si le nombre d'itérations tend vers l'infini, si la température tend vers zéro et si la décroissance de la température de l'ordre $O(-\log(k))$ alors le processus construit converge en loi vers $P(\omega \mid F)$ (cf [13] pour plus de détails).

On peut noter que lorsque la température est élevée, l'algorithme peut accepter des accroissements importants de l'énergie, mais lorsque la température décroit les accroissements deviennent de plus en plus rares et le système se stabilise.

Dans la pratique, la difficulté essentielle est de choisir un critère d'arrêt, une température initiale et une loi de décroissance de la température. On peut trouver plusieurs exemples dans [13].

2.2 L'échantillonneur de Gibbs

2 Les algorithmes de relaxation

De nombreux algorithmes de minimisation de fonctions non-convexes ont déjà été proposés pour résoudre des problèmes de vision bas-niveau. Il existe plusieurs méthodes de relaxation. Les techniques stochastiques, telles que le recuit simulé [13], convergent asymptotiquement vers le minimum global mais sont très coûteuses en temps de calcul. Les méthodes déterministes sont sous-optimales, mais donnent de bons résultats et sont plus rapides que les méthodes stochastiques.

Dans cette partie nous présentons un nouvel algorithme qui est une variante de la plus ancienne méthode, l'algorithme de Metropolis [14,13]. La méthode proposée est pseudo-stochastique dans le sens, où à chaque itération on tire au hasard un nouvel état global (c'est la partie "stochastique"), mais la décision d'accepter cet état est déterministe.

En comparant le nouvel algorithme avec les anciens (l'échantillonneur de Gibbs [9,12], l'algorithme de Metropolis [14,13] et l'ICM [4,12]) nous trouvons que la nouvelle méthode est plus rapide que les algorithmes stochastiques et elle donne de bons résultats. Par contre, elle est moins rapide que l'ICM. Mais il est bien connu que l'ICM donne de bons résultats à condition de bien choisir les données initiales ce qui n'est pas le cas avec la méthode proposée qui est indépendante des conditions initiales.

2.1 L'algorithme de Metropolis

L'algorithme de Metropolis a été la première méthode pour résoudre un problème d'optimisation combinatoire [14,13] . L'idée est venu de simuler le recuit d'un liquide. Metropolis a proposé une méthode Monte Carlo qui réalise un générateur de séquences d'états d'un liquide. Une variante parallèle de cet algorithme se décompose de la manière suivante:

1. Tirer aléatoirement une configuration initiale ω^0, $k = 1$ et $T = T(1)$.
2. On tire au hasard (avec une distribution uniforme) un état global η tel que

$$\forall s \in \mathcal{S} \ : \eta_s \in \Lambda - \{\omega_s^k\} \ .$$

Remarquons que le dernier terme de (16) peut avoir moins de membres, si s est un site de près du bord de réseaux.

Après cela, il reste à retrouver l'estimateur MAP $\hat{\omega}$. Cela se fait à l'aide des algorithmes de relaxation qui sont présentés au chapitre suivant.

où $Z = \sum_{\omega \in \Omega} P(\mathcal{F} \mid \omega) P(\omega)$ est constant, d'où:

$$\hat{\omega} = \max_{\omega \in \Omega} P(\mathcal{F} \mid \omega) P(\omega) \tag{11}$$

Le premier terme exprime ici la vraisemblance de l'observation et le deuxième les contraintes appliquées aux étiquettes. On suppose que le premier terme est Gaussien et le deuxième Markovien:

$$P(\mathcal{F} \mid \omega) = \prod_{s \in \mathcal{S}} P(\vec{f_s} \mid \omega_s) = \prod_{s \in \mathcal{S}} \frac{1}{\sqrt{(2\pi)^n \mid \Sigma_{\omega_s} \mid}} \exp\left(-\frac{1}{2}[\vec{f_s} - \vec{\mu}_{\omega_s}]^T \Sigma_{\omega_s}^{-1} [\vec{f_s} - \vec{\mu}_{\omega_s}]\right) \tag{12}$$

où $\vec{\mu}_\iota$ ($\iota \in \Lambda$) est la moyenne et Σ_ι ($\iota \in \Lambda$) est la matrice de variance de la classe ι,

$$P(\omega) = \exp\left(-\frac{U(\omega)}{T}\right) \;=\; \exp\left(-\frac{1}{T} \sum_{C \in \mathcal{C}} E_C(\omega)\right) \tag{13}$$

En supposant que $T = 1$, on a:

$$\begin{aligned}
\hat{\omega} &= \max_{\omega \in \Omega} \left(\log P(\mathcal{F} \mid \omega) + \log P(\omega)\right) \\
&= \max_{\omega \in \Omega} \left(\sum_{s \in \mathcal{S}} - \left(\log \sqrt{(2\pi)^n \mid \Sigma_{\omega_s} \mid} + \frac{1}{2}[\vec{f_s} - \vec{\mu}_{\omega_s}]^T \Sigma_{\omega_s}^{-1} [\vec{f_s} - \vec{\mu}_{\omega_s}]\right) - \sum_{C \in \mathcal{C}} E_C(\omega)\right) \\
&= \min_{\omega \in \Omega} \left(\sum_{s \in \mathcal{S}} \left(\log \sqrt{(2\pi)^n \mid \Sigma_{\omega_s} \mid} + \frac{1}{2}[\vec{f_s} - \vec{\mu}_{\omega_s}]^T \Sigma_{\omega_s}^{-1} [\vec{f_s} - \vec{\mu}_{\omega_s}]\right) + \sum_{C \in \mathcal{C}} E_C(\omega)\right)
\end{aligned} \tag{14}$$

A l'aide de la formule précédente, on peut calculer les énergies globale et locale de la façon suivante:

$$\mathcal{E}_{glob}(\omega) = \sum_{s \in \mathcal{S}} \left(\log \sqrt{(2\pi)^n \mid \Sigma_{\omega_s} \mid} + \frac{1}{2}[\vec{f_s} - \vec{\mu}_{\omega_s}]^T \Sigma_{\omega_s}^{-1} [\vec{f_s} - \vec{\mu}_{\omega_s}]\right) + \sum_{C \in \mathcal{C}} E_C(\omega) \tag{15}$$

$\mathcal{E}_s(\omega)$ note l'énergie locale de l'état ω en s:

$$\mathcal{E}_s(\omega) = \log \sqrt{(2\pi)^n \mid \Sigma_{\omega_s} \mid} + \frac{1}{2}[\vec{f_s} - \vec{\mu}_{\omega_s}]^T \Sigma_{\omega_s}^{-1} [\vec{f_s} - \vec{\mu}_{\omega_s}] + \sum_{C_s} E_{C_s}(\omega) \tag{16}$$

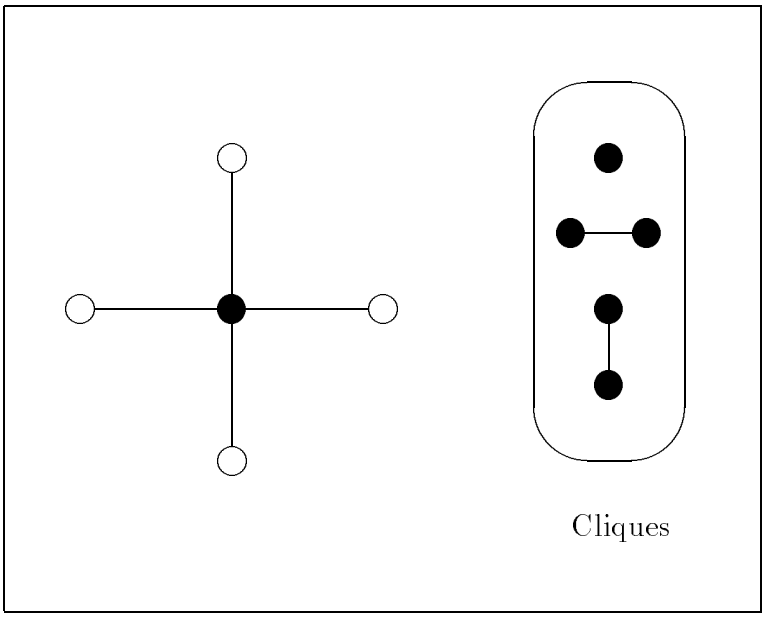


Figure 1 : Voisinage du premier ordre et les cliques correspondantes

$$\mathcal{S} = \{s = (i,j) \mid 1 \leq i \leq N,\, 1 \leq j \leq M\}\ .$$

Chaque $\vec{f_s} \in \mathcal{F}$ est un vecteur correspondant aux attributs observés du pixel s (par exemple le niveau de gris, la texture, ...) tel que

$$\forall s \in \mathcal{S} :\mid \vec{f_s} \mid = n$$

où n est le nombre des propriétés considerées.

Considérons un système arbitraire de voisinage $\mathcal{V}$ de $\mathcal{S}$. L'exemple le plus simple est le voisinage du premier ordre (voir figure 1). On cherche l'état $\hat{\omega} \in \Omega$ qui maximise la probabilité a posteriori de ω, autrement dit un estimateur MAP de ω:

$$\hat{\omega} = \max_{\omega \in \Omega} P(\omega \mid \mathcal{F})$$

En appliquant le théorème de Bayes (9), on obtient:

$$\hat{\omega} = \max_{\omega \in \Omega} \frac{1}{Z} P(\mathcal{F} \mid \omega) P(\omega)$$

Si $\Sigma_{\vec{x}}$ est une matrice diagonale, les composantes de $\vec{x}$ sont statistiquement indépendantes puisque $p(\vec{x})$ peut être factorisée en un produit de n densités de probabilité scalaires. C'est à dire en produit de densités de probabilité marginales:

$$\vec{\mu}_{\vec{x}} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} \qquad \text{et} \qquad \Sigma_{\vec{x}} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ 0 & \cdots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_n^2 \end{bmatrix}$$

$$\Longrightarrow \quad p(\vec{x}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\,\sigma_i} \exp\left(\frac{(x_i-\mu_i)^2}{2\sigma_i^2}\right) \tag{7}$$

1.3 Théorème de Bayes

Théorème 2 *Soit $\{A_1, A_2, \ldots, A_n\}$ des événements exlusifs et exhaustifs tels que $P(A_i) \neq 0$ $(i = 1, 2, \ldots, n)$. Alors pour tout événement B, on a:*

$$P(B) = \sum_{i=1}^{n} P(B \mid A_i) P(A_i) \tag{8}$$

Théorème 3 (Bayes) *Si $\{A_1, A_2, \ldots, A_n\}$ sont des événements exclusifs et exhaustifs tels que $P(A_i) \neq 0$ $(i = 1, 2, \ldots, n)$ alors pour tout événement B tel que $P(B) \neq 0$, on a:*

$$P(A_i \mid B) = \frac{P(B \mid A_i) P(A_i)}{\underbrace{\sum_{i=1}^{n} P(B \mid A_i) P(A_i)}_{=P(B) \text{ d'après (8)}}} \tag{9}$$

En supposant que $P(B)$ soit constant dans (9), on obtient:

$$P(A_i \mid B) \propto P(B \mid A_i) P(A_i)\,. \tag{10}$$

$P(A_i \mid B)$ représente ici la probabilité a posteriori, $P(B \mid A_i)$ la probabilité a priori et $P(A_i)$ les contraintes appliquées à A_i.

1.4 Le modèle Markovien

Soit $\mathcal{F} = \{\vec{f}_s\}_{s \in \mathcal{S}}$ un ensemble d'observations sur une image définie sur un réseaux fini $N \times M$ où

1.2 La distribution Gaussienne multidimensionelle

Considerons un vecteur aléatoire $\vec{x}$:

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad .$$

Soit $p(\vec{x})$ la densité de probabilité de ce vecteur. *La valeur moyenne* ou *l'espérance mathématique* du vecteur $\vec{x}$ est définie par

$$\vec{\mu}_{\vec{x}} = E(\vec{x}) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_n} \vec{x} p(\vec{x}) \quad .$$

De même, on définit *la matrice de variance* $\Sigma_{\vec{x}}$ du vecteur $\vec{x}$ par

$$\begin{aligned} \Sigma_{\vec{x}} &= E([\vec{x} - \vec{\mu}_{\vec{x}}]^T [\vec{x} - \vec{\mu}_{\vec{x}}]) \\ &= \sum_{x_1} \sum_{x_2} \cdots \sum_{x_n} [\vec{x} - \vec{\mu}_{\vec{x}}]^T [\vec{x} - \vec{\mu}_{\vec{x}}] p(\vec{x}) \quad . \end{aligned}$$

Lorsque cette matrice est diagonale, les composantes aléatoires x_i du vecteur $\vec{x}$ sont dites *non corrélées*. On peut montrer que dans ce cas là, la densité de probabilité est factorisable:

$$p(\vec{x}) = p(x_1) p(x_2) \cdots p(x_n) \quad ,$$

car les composantes x_i sont indépendantes.

Définition 4 *Un vecteur aléatoire $\vec{x}$ est dit gaussien s'il possède une densité de probabilité à la forme suivante:*

$$p(\vec{x}) = \frac{1}{\sqrt{(2\pi)^n \mid \Sigma_{\vec{x}} \mid}} \exp\left(-\frac{1}{2}[\vec{x} - \vec{\mu}_{\vec{x}}]^T \Sigma_{\vec{x}}^{-1} [\vec{x} - \vec{\mu}_{\vec{x}}]\right) \tag{6}$$

où

- $\vec{\mu}_{\vec{x}}$ *est la valeur moyenne du* $\vec{x}$
- $\Sigma_{\vec{x}}$ *est la matrice de variance du* $\vec{x}$
- $\mid \Sigma_{\vec{x}} \mid$ *est la déterminant de* $\Sigma_{\vec{x}}$

2.

$$P(X_s = \omega_s \mid X_r = \omega_r,\, r \neq s) = P(X_s = \omega_s \mid X_r = \omega_r,\, r \in \mathcal{V}_s) \qquad (2)$$

(2) signifie que la connaissance d'un voisinage du site s est suffisante pour calculer la probabilité en chaque site s.

Définition 3 *Une* **distribution de Gibbs** *est une mesure de probabilité π sur Ω à la forme suivante:*

$$\pi(\omega) = \frac{1}{Z} \exp\left(-\frac{1}{T} U(\omega)\right) \qquad (3)$$

où

- *T est la température, c'est une constante.*
- *Z est la fonction de partition. C'est une constante de normalisation définie par*

$$Z = \sum_{\omega \in \Omega} \exp\left(-\frac{U(\omega)}{T}\right) . \qquad (4)$$

- *$U(\omega)$ est une fonction énergie définie par*

$$U(\omega) = \sum_{C \in \mathcal{C}} E_C(\omega) \qquad (5)$$

où $\mathcal{C}$ est l'ensemble de cliques correspondant au voisinage $\mathcal{V}$ et $E_C(\omega)$ est une fonction potentielle permettant d'exprimer les contraintes appliquées à ω.

Le théorème de Hammersley-Clifford nous permet de travailler simplement avec les champs de Markov, notamment de spécifier les potentiels au lieu des probabilités.

Théorème 1 (Hammersley - Clifford) *Le champ aléatoire $\mathcal{X}$ défini sur $\mathcal{S}$ est un champ de Markov par rapport au système de voisinage $\mathcal{V}$ si et seulement si sa distribution $\pi(\omega) = P(\mathcal{X} = \omega)$ est une distribution de Gibbs par rapport au $\mathcal{V}$.*

1 Classification d'image

Beaucoup de problèmes peuvent être formulés en termes d'assignation d'étiquettes à des objets. On a besoin de méthodes pour choisir le meilleur étiguetage. Les algorithmes dits "relaxation labeling" sont une classe de méthodes possible. Ces algorithmes contiennent deux étapes majeures:

1. Préclassification par pixel (le classificateur Bayésien est le plus souvent utilisé)

2. Nettoyage, autrement dit, trouver un étiguetage consistent par rapport aux contraintes appliquées aux étiquettes [11,3,7,15].

Nous proposons une méthode basée sur les champs de Markov. L'idée de base est la suivante: Au lieu d'exécuter les deux étapes séparément, on définit un estimateur MAP ("Maximum A Posteriori") qui contient deux termes:

- un terme de vraisemblance de l'observation (qui correspond à l'étape 1.)
- un terme d'influence du contexte (qui correspond à l'étape 2.).

Cette maximisation est souvent réalisée par un algorithme de recuit simulé.

1.1 Champs de Markov et distribution de Gibbs

Soit $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ un ensemble de sites et $\mathcal{V} = \{\mathcal{V}_s, s \in \mathcal{S}\}$ une famille arbitraire de voisinages locaux de $\mathcal{S}$.

Définition 1 *Un sous-ensemble $C \subseteq \mathcal{S}$ est une* **clique**, *si deux points quelconques de C sont des voisins par rapport au système de voisinage $\mathcal{V}$.*

Soit $\Lambda = \{0, 1, 2, \ldots, L-1\}$ l'espace d'état, $\mathcal{X} = \{X_s \mid s \in \mathcal{S}\}$ une famille des variables aléatoires telle que $\forall s \in \mathcal{S} : X_s \in \Lambda$ et

$$\Omega = \{\omega = (\omega_{s1}, \omega_{s2}, \ldots, \omega_{sN}) \mid \omega_{si} \in \Lambda, 1 \leq i \leq N\} = \Lambda^{\mathcal{S}}$$

l'ensemble de tous les états possibles.

Définition 2 *$\mathcal{X}$ est un* **champ de Markov** *par rapport à $\mathcal{V}$, si pour $\forall \omega \in \Omega$ et $\forall s \in \mathcal{S}$ on a:*

1.

$$P(\mathcal{X} = \omega) > 0 \tag{1}$$

Introduction

L'utilisation des champs de Markov en vision par ordinateur s'est développée depuis quelques années [1,2,9]. Les principales applications de ce type de modélisation sont la détection de contours [5,8,16,17], la restauration d'image [5,9,16,17], et la reconstruction de surface [5,8].

Le principal avantage des champs de Markov est de fournir un modèle mathématique simple permettant de prendre en compte les intéractions locales au niveau du pixel par les énergies potentielles. Il existe plusieurs méthodes stochastiques et déterministes pour minimiser une énergie non-convexe:

- l'algoritme de Metropolis [14,13]
- l'échantilloneur de Gibbs [9,12]
- GNC ("Graduated Non Convexity") [5,17]
- ICM ("Iterated Conditional Mode") [4,12,10]
- HCF ("Highest Confidence First") [6]

  ⋮

Un inconvénient majeur de tous ces algorithmes est le coût de calcul.

Le but de ce stage est de réaliser un modèle Markovien pour la classification d'image et de comparer les résultats obtenus avec différents algorithmes de relaxation.

Dans la première partie de ce rapport, nous présentons le modèle Markovien développé pour la classification d'image. Dans la deuxième partie, nous proposons un nouvel algorithme de relaxation et nous le comparons avec les anciennes méthodes (l'algorithme de Metropolis, l'échantillonneur de Gibbs et l'ICM ont été réalisés).

Table des tableaux

1 Les valeurs moyennes et les écart-types des classes 20
2 Les temps sur l'image synthétique 128×128 23
3 Les temps sur l'image SPOT 128×128 23
4 Les temps sur l'image SPOT 300×128 23
5 La convergence sur l'image synthétique 128×128 24
6 La convergence sur l'image SPOT 128×128 25
7 La convergence sur l'image SPOT 300×128 25

Table des figures

1 Voisinage du premier ordre et les cliques correspondantes 10
2 La sélection du nouvel état 15
3 Les deux sous-réseaux de $\mathcal{S}$ dans le cas de voisinage d'ordre un . 16
4 Résultats obtenus avec un critère d'arrêt 0.1% 25
5 Résultats sur deux images synthétiques 26
6 Résultats à $T(1) \approx 16$ $(K = 5)$ 27
7 L'énergie globale en fonction du nombre d'itérations (image synthétique 128×128) 33
8 Le nombre de changements d'état en fonction du nombre d'itérations (image synthétique 128×128) 34
9 Le nombre de sites en fonction du nombre d'itérations où la nouvelle énergie est plus grande que la précédente (image synthétique 128×128) 35
10 L'énergie globale en fonction du nombre d'itérations (image SPOT 128×128) 36
11 Le nombre de changements d'état en fonction du nombre d'itérations (image SPOT 128×128) 37
12 Le nombre de sites en fonction du nombre d'itérations où la nouvelle énergie est plus grande que la précédente (image SPOT 128×128) 38
13 L'énergie globale en fonction du nombre d'itérations (image SPOT 300×128) 39
14 Le nombre de changements d'état en fonction du nombre d'itérations (image SPOT 300×128) 40
15 Le nombre de sites en fonction du nombre d'itérations où la nouvelle énergie est plus grande que la précédente (image SPOT 300×128) 41
16 Résultats obtenus sur une image synthétique 128×128 43
17 Résultats obtenus sur une image SPOT 128×128 44
18 Image originale SPOT 300×128 44
19 Résultats obtenus sur une image SPOT 300×128 45
20 Image originale SPOT 512×512 47
21 Méthode proposée, 200 itérations 48
22 ICM, 6 itérations 49
23 Echantillonneur de Gibbs, 200 itérations 50
24 Metropolis, 200 itérations 51

Table des matières

1 Classification d'image **6**
1.1 Champs de Markov et distribution de Gibbs 6
1.2 La distribution Gaussienne multidimensionelle 8
1.3 Théorème de Bayes 9
1.4 Le modèle Markovien 9

2 Les algorithmes de relaxation **13**
2.1 L'algorithme de Metropolis 13
2.2 L'échantillonneur de Gibbs 14
2.3 L'ICM ("Iterated Conditional Mode") 16
2.4 La méthode proposée ("Modified Metropolis Dynamics") 17

3 Résultats expérimentaux **20**
3.1 Mise en œuvre sur la CM-2 20
3.2 Le modèle réalisé 20
3.3 Comparison des algorithmes 22
3.3.1 Choix des paramètres 22
3.3.2 Comparison des temps de calcul 23
3.3.3 Comparison de la convergence 24
3.4 Le choix du seuil α 26

4 Conclusion **28**

A Résultats de convergence **30**
A.1 Preuve de la convergence 30
A.2 Figures 32

B Images **42**

C Résultats obtenus sur une image SPOT 512×512 **46**

Modélisation Markovienne en Classification d'Image. Mise en œuvre d'algorithmes de relaxation

KATO Zoltan

2 septembre 1991

Rapport de DEA

INRIA - Sophia Antipolis
2004 route des Lucioles
06560 Valbonne Cedex
France

Responsables de stage: Josiane Zerubia et Marc Berthod