N^o d'ordre:

THÈSE

présentée à

L'UNIVERSITÉ DE NICE SOPHIA ANTIPOLIS

pour obtenir le titre de

DOCTEUR EN SCIENCES

Spécialité

SCIENCES DE L'INGÉNIEUR

par

Zoltan KATO

Sujet de la thèse:

Modélisations markoviennes multirésolutions en vision par ordinateur. Application à la segmentation d'images SPOT

Soutenue le 20 Décembre 1994 devant la commission d'examen composée de:

M	**Michel**	**BARLAUD**	Président
Mme	**Christine**	**GRAFFIGNE**	Rapporteurs
M	**Fabrice**	**HEITZ**	
M	**Marc**	**BERTHOD**	Examinateurs
M	**Marc**	**SIGELLE**	
Mme	**Josiane**	**ZERUBIA**	

J'adresse mes remerciements:

à Michel BARLAUD pour m'avoir fait l'honneur de présider le jury;
à Christine GRAFFIGNE et Fabrice HEITZ pour avoir participer à ce jury en tant que rapporteurs. Je leur suis reconnaissant de leurs remarques sur le contenu de cette thèse;
à Marc SIGELLE pour avoir participer à ce jury;
à Marc BERTHOD pour m'avoir proposé ce sujet de recherche et m'avoir toujour soutenu;
à Josiane ZERUBIA pour m'avoir guidé tout au long de ces trois années;
à Meir GRINASTY pour les discussions fructueuses sur le sujet d'estimation des paramètres et la méchanique statistique;
à Jean KHEDARI pour avoir consacré du temps à la lecture de la première version du manuscrit français.

Version française abrégée

Table des matières

Figures

Chapitre 2. **27**

Tableaux

Introduction

La vision par ordinateur se réfère aux algorithmes variés pour restaurer ou interpréter les images digitales. On peut distinguer deux niveaux de traitement d'images: Le but de la vision haut niveau est d'extraire les attributs symboliques (par exemple la reconnaissance des lettres écrites à la main) et le but de la vision bas niveau (ou vision pré-attentive) est d'extraire des attributs nécessaires pour la vision haut niveau (par exemple l'extraction des contours). Le premier étape du traitement d'images est le traitement bas niveau [77, 31].

Dans cette thèse, nous nous intéressons à une approche statistique de la vision pré-attentive. Dans une image réelle, les pixels voisines ont un niveau de gris similaire. Dans un cadre probabilistique, une telle régularité est bien modélisée par les champs de Markov. D'un autre côté, le comportement local des champs markoviens permet de mettre en œuvre des algorithmes massivement parallèles pour résoudre les problèmes d'optimisation combinatorial associés à une telle modélisation. Nous discuterons aussi les méthodes d'estimation des paramètres, un problème très important dans les applications réelles.

Dans le Chapitre 2, nous proposons un nouveau modèle markovien hiérarchique et dans le Chapitre 4, nous présentons une méthode d'estimation des paramètres de ce modèle. Dans le Chapitre 3, nous proposons un nouveau recuit multi-température et nous prouverons la convergence vers un minimum global. Nous proposons aussi une méthode de relaxation déterministe avec une étude détaillée de la convergence. Tous les modèles et algorithmes présentés dans cette thèse ont été mis en œuvre sur une machine parallèle CM200. Des tests comparatifs seront présentés à la fin de chaque chapitre.

Le traitement d'images

L'une des premières applications du traitement d'images était la transmission des images entre New York et London par un câble sous-marin [31]. Les images ont été codées pour la transmission et décodées après la réception. Le problème initial était d'améliorer la qualité des images transmises.

Avec la construction des ordinateurs de $3^{\text{ième}}$ génération dans les années 60, l'utilisation du traitement d'images s'est étendue. D'un autre côté, le traitement d'images a été souvent l'application qui a inspiré la conception des ordinateurs massivement parallèles.

Le but de la vision par ordinateur est de traiter les images pour la perception autonome d'ordinateur. Un système de vision contient une ou plusieurs caméra et des algorithmes pour interpréter les images sensées. Le terme *image* (ou plus précisement *image monochrome*) se réfère à une fonction bidimensionelle dont la valeur dans un point est proportionnelle au *niveau de gris* [31]. Une *image digitale* est une image discrétisée en coordonnées et intensité. En générale, elle est représentée par une matrice à deux dimensions, les éléments de la matrice sont les pixels.

On peut distinguer deux niveau de traitement: *La vision bas niveau* traite une grande quantité de pixels et les transforme en attributs qui peuvent être directement utilisés dans *la vision haut niveau*. Dans cette thèse, nous nous intéressons à la vision pré-attentive, en particulier à la modélisation probabiliste par des champs markoviens.

La vision pré-attentive et les champs de Markov

Le but de la vision pré-attentive se réfère aux tâches suivantes [1]: compression d'images, restauration d'images [29, 40, 84, 86, 83], détection des contours [81, 84, 86, 83], segmentation [53, 44, 26, 20, 80, 21, 35], détection du mouvement [38], flôt optique, etc... La plupart de ces tâches peuvent se formaliser dans un cadre général appelé l'étiquetage d'images où à chaque pixel, on veut associer une étiquette appartenant à un ensemble fini. La signification des étiquettes dépend du problème à résoudre. Pour la restauration d'images, elles signifient les niveaux de gris; pour la détection de contour, elles signifient la présence ou la direction des éléments de contour; pour la segmentation, elles signifient les classes; etc... Le problème est de choisir une étiquette optimale pour un pixel. L'étiquetage par relaxation [41] est une méthode classique, non-probabiliste qui permet de résoudre ce problème.

Notre approche est probabiliste: pour chaque pixel, nous voulons choisir l'étiquette la plus probable. Dans ce but, nous avons besoin de définir une mesure de probabilité sur

l'ensemble des étiquetages possibles. Dans les images réelles, les pixels voisins ont une intensité similaire; les contours sont lisses et souvent droits. Dans un cadre probabiliste, des telles régularités sont bien exprimées par les champs markoviens (MRF). D'autre part, le théorème de *Hammersley-Clifford* [9, 68] permet de définir les champs de Markov par des fonctions de potentiel. Dans le problème d'étiquetage, cette modélisation nous ramène à l'estimation bayesienne suivante: nous cherchons l'estimation MAP du champs des étiquettes par la minimisation de la fonction d'énergie non-convexe.

Malheureusement, c'est un problème très dur au point de vue calcul. Par exemple, si nous considérons une image de taille 16×16 avec deux étiquettes possibles, nous obtenons une espace de configuration de 2^{256} éléments. Il est donc impossible de calculer toutes les valeurs possibles de la fonction d'énergie. D'un autre côté, l'utilisation des méthodes classiques n'est pas possible à cause de la non-convexité de la fonction d'énergie. Dans les années 80, un algorithme de type Monte-Carlo, appelé recuit simulé, a été proposé par Černy [17] et Kirkpatrick *et al.* [56] pour résoudre ce problème d'optimisation. Cependant, les premiers résultats mathématiques [29, 34] ont montré que l'utilisation correcte du recuit simulé exige une loi de température très lente, donc beaucoup de temps calcul. Pour éviter cet inconvénient, deux solutions ont été proposé: L'une est la parallélisation des algorithmes de relaxation [3]. L'autre est d'utiliser des algorithmes déterministes qui sont sous-optimaux mais convergent avec un nombre faible d'itérations [8, 53].

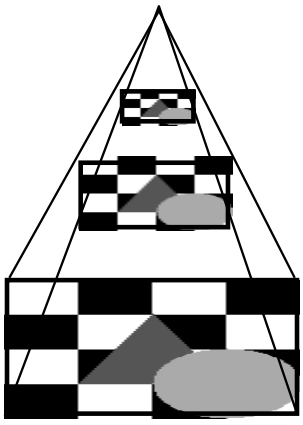

Figure 1: *Représentation pyramidale.*

Les modèles multigrilles (ou pyramidaux) [12, 62, 72, 45] peuvent aussi améliorer la vitesse de convergence et la qualité du résultat final des méthodes itératives. Les méthodes multigrilles sont utilisées depuis longtemps en analyse numérique. et depuis les années 70 [45] en traitement d'images. Nous nous intéressons ici aux méthodes appliqués à la modélisation markovienne d'images. Nous utilisons le mot *pyramidale* pour désigner les modèles *multigrilles* et *hiérarchiques*. Le but des approches pyramidales est de représenter les images à différentes résolutions (cf. Figure 1).

Si les couches dans la pyramide ne sont pas connectées, le modèle est *multigrille*. Dans ce cas, l'algorithme d'optimisation n'est peut être parallèle que sur les couches et séquentiel entre les niveaux. Une question importante est comment définir les cliques sur les niveaux plus grossiers. Plusieurs solutions ont été proposées [59] comme la méthode de renormalisation de Gidas [30, 69], les modèles multiéchelles de Perez *et al.* [38, 37, 73], ou le modèle de Bouman [11, 13].

Lorsque une communication entre les couches existe, le modèle est appelé *hiérar-*

chique [49, 48, 47]. L'algorithme d'optimisation peut être parallèle sur toute la pyramide mais le modèle markovien devient beaucoup plus compliqué et demande donc plus de temps calcul que les méthodes classiques.

La segmentation d'images

Dans cette thèse, les modèles et les algorithmes proposés sont testés sur des problèmes de segmentation d'images.

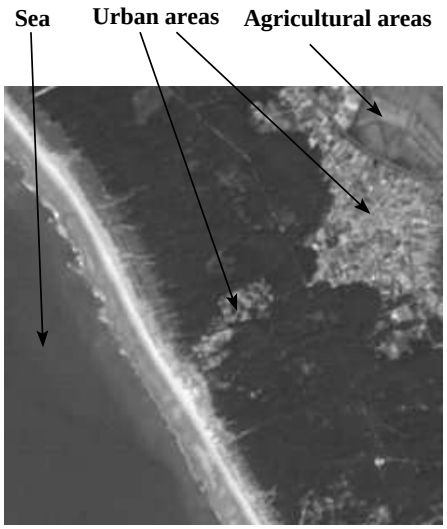

Figure 2: *Une image SPOT.*

Tout d'abord, nous définissons la notion de *classification* et celle de *segmentation*. Pratiquement, elles se réfèrent à la même tâche. Pourtant, nous préférons utiliser le mot *segmentation* car la *classification* est un terme plus général. La classification [23] suppose un extracteur d'attributs qui permet de différencier les éléments d'images (par exemple la mer et les agglomérations sur Figure 2). Le but de la classification est de partager *l'espace des attributs* en régions telles que chaque région correspond à une classe. Par exemple, nous avons une image satellite (Figure 2) et nous voulons localiser les agglomérations. En utilisant uniquement les valeurs de niveau de gris, nous ne pouvons pas différencier les agglomérations et les zones agricultures car elles sont dans le même spectre. Le seul attribut qui fait la différence est la *texture*. L'espace des attributs contient donc la texture *et* les valeurs de niveau de gris. Un espace plus complexe pourait être construit en utilisant des canaux différents (XS1, XS2, XS3 pour les images SPOT).

La segmentation peut être considérée comme un cas spécial de la classification où l'espace des attributs ne contient que les valeurs de niveau de gris. Le but de la segmentation est de partitioner une image en régions homogènes. Deux méthodes possibles: la détection des bords des régions ou la détection directe des régions sans les contours (*segmentation en régions*). Ici, nous nous intéressons à la deuxième approche. Les conditions de la segmentation en régions sont [31]:

- La segmentation doit être complète (c'est à dire, chaque pixel doit être dans une classe).

- Les pixels appartenant au même région doivent être connectés.
- Les régions doit être disjointes.

Des exemples classiques se trouvent dans [77, 31] (*region growing* ou *split and merge*).

Nous remarquons que le deuxième point dans la liste précédante signifie que les pixels voisins doivent être dans la même région. Cette contrainte est bien exprimée par les champs de Markov. De plus, nous attribuons une étiquette à chaque pixel et définissons un MRF sur ces étiquettes tel que les potentiels de clique favorisent les étiquettes similaires dans les pixels voisins. Cependant, avec ce modèle, nous obtiendrons une seule région. Il nous faut un autre terme qui fait la liaison entre les régions et les observations. Le modèle le plus naturel est de considérer chaque classe comme une distribution gaussienne. De cette façon, les régions sont caracterisées par la moyenne et la variance de la distribution normale correspondante. D'un autre côté, on peut introduire ces distributions dans le modèle markovien comme la fonction potentielle des cliques d'ordre un. Ce modèle est capable de segmenter correctement les images de niveau de gris.

Nous remarquons que dans le cadre markovien, on peut détecter les régions et les contours en même temps par l'introduction de *processus de ligne* [29, 84]. Cependant, nous n'avons pas utilisé ce modèle car notre but a été de construire un modèle universel facilement utilisable et d'étudier les implantations multiéchelles et hiérarchiques.

Sommaire par chapitres

Au Chapitre 1, nous traitons les fondaments des champs de Markov. Nous présentons aussi la théorie de la décision bayesienne et définissons les notions de base: probabilité, variables aléatoires, distribution, densité, convergence des variables aléatoires, distribution gaussienne, processus stochastique, etc. . .

Au Chapitre 2, nous présentons les modèles markoviens dans un cadre général, appelé étiquetage d'images. Nous présentons des méthodes multi-grilles et nous proposons un nouveau modèle markovien hiérarchique.

Au Chapitre 3, nous étudions les algorithmes d'optimisation combinatoire. Puisque les modèles markoviens exigent la minimisation d'une fonction non-convexe, le resultat final dépend fortement de l'algorithme d'optimisation utilisé. Nous présentons quelques

méthodes de relaxation stochastiques et déterministiques ainsi que des techniques de parallélisation. Nous proposons un algorithme de recuit multi-température dont la convergence a été démontrée [47] par la généralisation du théorème de Geman et Geman [29]. Nous proposons également un algorithme déterministe appelé Dynamique de Metropolis modifiée qui est un bon compromis entre la qualité et le temps de l'exécution. L'étude mathématique de l'algorithme a été établie sous forme de théorème.

Au Chapitre 4, nous proposons quelques méthode d'estimation des paramètres en particulier pour le modèle hiérarchique et nous appliquons les algorithmes aux problèmes de segmentation monogrille et hiérarchique non-supervisé.

Dans ce chapitre:

1.

Fondements

Dans ce chapitre, nous discutons les principaux fondements des champs de Markov d'un point de vu mathématique et physique. La théorie des champs markoviens a été inspirée par le méchanique statistique (modèle d'Ising). Dans le traitement d'images, nous utilisons les mêmes termes: énergie, potentiel, température... Bien sûr, le sens des mots est différent de celui utilisé en mécanique statistique.

Nous nous intéressons aussi à la théorie de la décision au sens bayesien qui est la base de l'estimation du Maximum A Posteriori (MAP) largement utilisé dans l'étiquetage d'images. Nous définissons quelques notions comme la probabilité, les variables aléatoires, la distribution, la densité, les processus stochastiques, la convergence des variables aléatoires, etc... La distribution gaussienne est présentée en détails car c'est la distribution la plus souvent utilisée dans la traitement d'images.

Le contenu de ce chapitre est fondé sur le livre de Papoulis sur la théorie de probabilité [70].

1.1 Probabilité et variables aléatoires

Notons l'événement certain par **I** (celui qui se produit dans chaque essai). Considérant deux événements A et B, $A \cup B$ note l'événement où tous les deux se produisent. A et B sont mutuellement exclusifs s'ils ne peuvent pas se produire au même temps. Nous définissons la probabilité comme une mesure:

Définition 1.1.1 (Probabilité) *La probabilité d'un événement A est le nombre $P(A)$ satisfaisant les axiomes suivantes:*

(i) $P(A)$ *est positif:* $P(A) \geq 0$

(ii) La probabilité d'événement certain est 1: $P(\mathbf{I}) = 1$

(iii) Si A et B sont mutuellement exclusifs , alors $P(A \cup B) = P(A) + P(B)$

Définition 1.1.2 (Variable aléatoire) *Une variable aléatoire X est une fonction dont le domaine est l'espace* **I**, *qui assigne le nombre $X(\xi)$ à chaque événement élémentaire $\xi \in \mathbf{I}$ tel que:*

(i) L'ensemble $\{X \leq x\}$ est un événement pour tous les x.

(ii) La probabilité des événements $\{X = +\infty\}$ et $\{X = -\infty\}$ est égale à zéro.

Maintenant, nous définissons quelques fonctions utiles pour caractériser les variables aléatoires.

Définition 1.1.3 (Distribution) *Étant donné une variable aléatoire X, la fonction*

$$F_X(x) = P\{X \leq x\} \tag{1.1}$$

est appelée la fonction de répartition *(ou distribution) de X pour tous les $x \in (-\infty, \infty)$.*

Définition 1.1.4 (Densité) *Étant donné une variable aléatoire X, la dérivé de sa distribution $F_X(x)$:*

$$f(x) = \frac{dF(x)}{dx} \tag{1.2}$$

est la fonction de densité *de X.*

Les paramètres les plus importants d'une variable aléatoire sont l'espérance mathématique (ou la valeur moyenne) et la variance.

Définition 1.1.5 (L'espérance mathématique) *L'espérance mathématique d'une variable aléatoire X est l'intégral*

$$E\{X\} = \int_{-\infty}^{\infty} xf(x)dx \tag{1.3}$$

où $f(x)$ est la fonction de densité de X.

Définition 1.1.6 (Variance) *La variance d'une variable aléatoire dont la moyenne est μ est donnée par:*

$$\sigma^2 = E\{(X-\mu)^2\} = \int_{-\infty}^{\infty} (x-\mu)^2 f(x)dx \tag{1.4}$$

σ est appelée l'écart type.

On peut spécifier les statistiques d'une variable aléatoire en utilisant ses moments:

Définition 1.1.7 (Moments) *Les moments m_k d'une variable aléatoire X sont définis par:*

$$m_k = E\{X^k\} = \int_{-\infty}^{\infty} x^k f(x)dx$$

Nous considérons deux variables aléatoires et définissons la distribution jointe et la densité jointe. Nous remarquons que ces définitions peuvent être généralisées à plusieurs variables aléatoires.

Définition 1.1.8 (Distribution jointe) *La distribution jointe des variables aléatoires X and Y est définie par*

$$F_{XY}(x,y) = P\{X \leq x, Y \leq y\}.$$

Les distributions $F_X(x)$ et $F_Y(y)$ sont appelées les marginales.

Définition 1.1.9 (Densité jointe) *Supposons que $F_{XY}(x,y)$ soit différentiable jusqu'à l'ordre deux. La densité jointe de X et Y est donnée par*

$$f_{XY}(x,y) = \frac{\partial^2 F_{XY}(x,y)}{\partial x \partial y} \tag{1.5}$$

Définition 1.1.10 (Covariance) *La covariance des deux variables aléatoires X et Y est définie par*

$$cov_{XY} = E\{(X - \mu_X)(Y - \mu_Y)\} \tag{1.6}$$

et le rapport

$$r = \frac{E\{(X - \mu_X)(Y - \mu_Y)\}}{\sqrt{E\{(X - \mu_X)^2\}E\{(Y - \mu_Y)^2\}}} = \frac{cov_{XY}}{\sigma_X \sigma_Y} \tag{1.7}$$

est appelé le coefficient de corrélation.

Dans les paragraphes suivantes, nous discutons la théorie de la probabilité bayesienne. L'idée générale dans cette théorie est que toutes les probabilités sont conditionnelles. Cependant, pour simplifier les notations, nous allons utiliser $P(A)$ au lieu de $P(A \mid .)$.

Définition 1.1.11 (Probabilité conditionnelle) *Étant donné un événement C dont la probabilité est positive, la probabilité conditionnelle de A sachant C est définie par:*

$$P(A|C) = \frac{P(A \cap C)}{P(C)} \tag{1.8}$$

Il y a deux règles pour manipuler les probabilités: la *règle de multiplication* et la *règle d'addition*:

$$\text{La règle de multiplication:} \quad P(A, B|C) = P(A|C)P(B|A, C) \tag{1.9}$$

$$\text{La règle d'addition:} \quad P(A \cup B|C) = P(A|C) + P(B|C) - P(A, B|C) \tag{1.10}$$

La règle de l'addition a un rôle important dans le théorème suivant:

Théorème 1.1.1 (Probabilité totale) *Étant donné n événements $A_1, \ldots, A_n$ mutuellement exclusifs dont la somme est l'événement certain:*

$$A_i \cap A_j = \emptyset \quad \forall i \neq j, \quad i = 1, \ldots n$$

$$\bigcup_{i=1}^{n} A_i = \mathbf{I}$$

L'équation suivante est satisfaite pour n'importe quel événement B:

$$P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i) \tag{1.11}$$

Le théorème le plus important dans la théorie de probabilité bayesienne est le suivant:

Théorème 1.1.2 (Bayes)

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

La probabilité $P(A|B)$ est appelée *la probabilité a posteriori* et $P(A)$ est *la probabilité a priori* de A.

Définition 1.1.12 (Distribution conditionnelle) *Étant donné un événement C dont la probabilité est positive. La distribution conditionnelle d'une variable aléatoire X est donnée par*

$$F_X(x|C) = P\{X \leq x|C\} = \frac{P\{X \leq x, C\}}{P(C)} \tag{1.12}$$

Définition 1.1.13 (Independence conditionnelle) *X_1 est conditionnellement indépendent de X_2 sachant X_3 si*

$$f(x_1, x_2|x_3) = f(x_1|x_3)f(x_2|x_3) \tag{1.13}$$

1.2 La distribution gaussienne

Définition 1.2.1 (Distribution normale) *Une variable aléatoire a une distribution normale si sa fonction de densité est une gaussienne (voir Figure 1.1)*

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{1.14}$$

où μ est la moyenne (Définition 1.1.5) et σ est l'écart type (Définition 1.1.6).

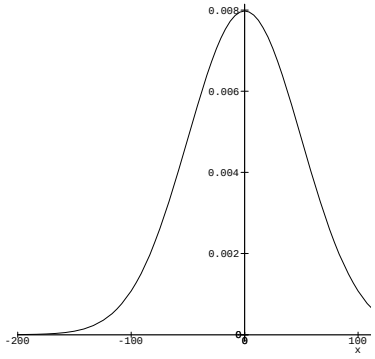

Figure 1.1: *Fonction de densité d'une variable aléatoire normale.*

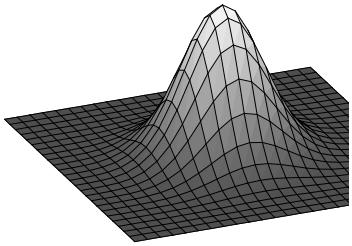

Figure 1.2: *Densité jointe des deux variables aléatoires normales*

Définition 1.2.2 (Distribution jointe normale) *Deux variables aléatoires X et Y ont une distribution jointe normale si leur fonction de densité est donnée par (voir Figure 1.2)*

$$f(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-r^2}} \exp\left(-\frac{\left(\frac{(x-\mu_X)^2}{\sigma_X^2} - \frac{2r(x-\mu_X)(y-\mu_Y)}{\sigma_X\sigma_Y} + \frac{(y-\mu_Y)^2}{\sigma_Y^2}\right)}{2(1-r^2)}\right). \quad (1.15)$$

où μ_X, μ_Y sont les moyennes et σ_X, σ_Y sont les écart types de X et Y respectivement. r est le coefficient de corrélation (cf. Définition 1.1.10).

On peut montrer que si deux variables aléatoires ont une distribution jointe normale alors elles sont aussi marginalement normales. L'inverse n'est vrai que si elles sont conditionnellement indépendentes. Pour la dimension n, la distribution jointe est donnée par:

$$f(x_1,\ldots,x_n) = f(\vec{x}) = \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} \exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu})^T\Sigma^{-1}(\vec{x}-\vec{\mu})\right) \quad (1.16)$$

où

$$\Sigma = \begin{pmatrix} \sigma_{11} & \ldots & \sigma_{1n} \\ \ldots & \ldots & \ldots \\ \sigma_{n1} & \ldots & \sigma_{nn} \end{pmatrix}, \quad \mu = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix} \quad (1.17)$$

Si les variables aléatoires X_i sont non-corrélées alors leur matrice de covariance Σ est une matrice diagonale, et leur fonction de densité peut être factorisée:

$$f(x_1, x_2, x_3) = f(x_1, x_2)f(x_3) \tag{1.18}$$

Pour l'estimation des paramètres (voir Chapitre 4), il sera très utile d'étudier la fonction de densité normale d'un point de vue géométrique. Le lieu des points du plan XY, tel que $f(x, y)$ est constant, est donné par l'équation suivante:

$$\frac{(x-\mu_X)^2}{\sigma_X^2} - \frac{2r(x-\mu_X)(y-\mu_Y)}{\sigma_X\sigma_Y} + \frac{(y-\mu_Y)^2}{\sigma_Y^2} = C \tag{1.19}$$

qui défini un ellipse de centre (μ_X, μ_Y) (voir Figure 1.3).

Dans les paragraphes suivants, nous discutons les formules récursives pour calculer les moments des variables aléatoires normales. Ces formules seront utilisées dans les algorithmes d'estimation des paramètres (cf. Chapitre 4). Pour une variable aléatoire normale avec une moyenne nulle, les formules suivantes donnent les moments:

$$E\{X^n\} = \begin{cases} 1 \cdot 3 \cdots (n-1)\sigma^n & \text{pour } n \text{ pair} \\ 0 & \text{pour } n \text{ impair} \end{cases} \tag{1.20}$$

$$E\{|X|^n\} = \begin{cases} 1 \cdot 3 \cdots (n-1)\sigma^n & \text{pour } n = 2k \\ \sqrt{\frac{2}{\pi}} 2^k k! \sigma^{2k+1} & \text{pour } n \text{ impair} \end{cases} \tag{1.21}$$

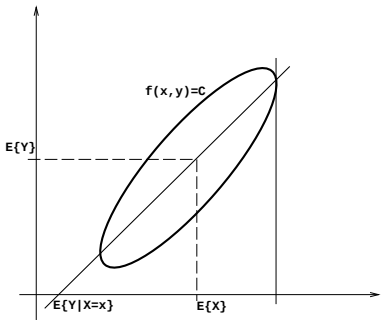

Figure 1.3: *Ensemble des points où $f(x, y)$ est constante.*

Dans le cas général, nous avons une formule qui est la fonction de la valeur moyenne μ et la variance σ^2:

$$m_k = E\{X^n\} = \frac{k(k-1)}{2} \int_0^{\sigma^2} m_{k-2} d\sigma^2 + \mu^k \text{ avec } m_0 = 1 \text{ et } m_1 = \mu \tag{1.22}$$

Pour les moments centraux η_k, nous pouvons obtenir une formule similaire:

$$\eta_k = E\{(x-\mu)^k\} = \frac{k(k-1)}{2} \int_0^{\sigma^2} \eta_{k-2} d\sigma^2 \text{ avec } \eta_0 = 1 \text{ et } \eta_1 = 0 \tag{1.23}$$

Les *moments joints* de deux variables aléatoires normales de covariance ς (voir Définition 1.1.10) sont donnés par:

$$E\{X^k Y^l\} = kl \int_0^{\varsigma} E\{X^{k-1} Y^{l-1}\} d\varsigma + E\{X^k\} E\{Y^l\} \tag{1.24}$$

1.2.1 Bruit blanc

Un modèle de bruit utile dans le traitement d'images est *le bruit blanc* [42]. La séquence $\{X_1, X_2, \ldots\}$ est blanche si elle est une séquence de Markov:

$$P(X_k \mid X_l, l < k) = P(X_k). \tag{1.25}$$

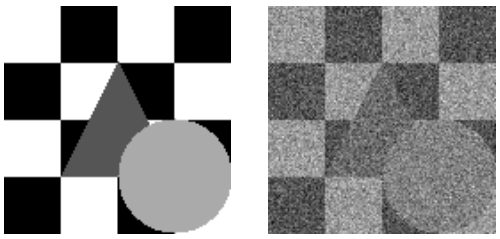

Figure 1.4: *Image bruitée (3dB).*

Si nous supposons que les X_k sont des variables aléatoires normales, la séquence $\{X_1, X_2, \ldots\}$ est appelée *bruit blanc gaussien*. En pratique, nous utilisons ce modèle. Étant donné que les X_k sont indépendentes, la matrice de covariance est diagonale et positive semidéfinie. En général, le bruit est caractérisé par le rapport *Signal/Bruit* (S/B) qui est mesuré en dB par l'équation suivante:

$$\text{S/B en } dB \ = 10 \lg \left(\frac{\sigma^2_{image}}{\sigma^2} \right), \tag{1.26}$$

où σ_{image} est la variance de l'image. Dans la Figure 1.4, nous montrons une image bruitée par $3dB$ de bruit blanc gaussien.

1.3 Convergence et la loi des grands nombres

Dans ce chapitre, nous donnons les définitions variées de la convergence des variables aléatoires.

Définition 1.3.1 (Convergence avec probabilité 1) *La séquence X_n converge vers X avec une probabilité de 1 si l'ensemble des événements ξ tel que*

$$\lim_{n \to \infty} X_n(\xi) = X(\xi) \tag{1.27}$$

a une probabilité égal à 1. On peut donc écrire

$$P\{X_n \to X\} = 1 \text{ pour } n \to \infty \tag{1.28}$$

Définition 1.3.2 (Convergence dans le sens des moindre carrés) *La séquence X_n converge vers X dans le sens des moindre carrés si*

$$\lim_{n \to \infty} E\{|X_n - X|^2\} = 0 \tag{1.29}$$

Définition 1.3.3 (Convergence en Probabilité) *Considérons la probabilité de* $|X_n - X| > \epsilon$ *pour un nombre* $\epsilon > 0$*:* $P\{|X_n - X| > \epsilon\}$*. Si elle converge vers zéro pour chaque* ϵ*,*

$$\forall \epsilon > 0 : \quad \lim_{n\to\infty} P\{|X_n - X| > \epsilon\} = 0 \tag{1.30}$$

alors la séquence X_n *converge vers* X *en probabilité.*

Définition 1.3.4 (Convergence en distribution) *Soit* $F_n(x)$ *et* $F(x)$ *la distribution (ou la fonction de répartition) de deux variables aléatoires* X_n *et* X*, respectivement. Si*

$$\lim_{n\to\infty} F_n(x) = F(x) \tag{1.31}$$

pour chaque point x *tel que* $F(x)$ *est continue, alors* X_n *converge vers* X *en distribution.*

Un théorème important en statistique est la loi des grands nombres:

Théorème 1.3.1 (Loi des grands nombres) *Si la probabilité d'un événement* A *est* p *dans un essai et l'essai est répété* n *foi, alors pour* $\epsilon > 0$ *quelconque,*

$$\lim_{n\to\infty} P\{\mid \frac{k}{n} - p \mid \leq \epsilon\} = 1 \tag{1.32}$$

où k *égal au nombre des succès de* A

1.4 La théorie de la décision

La théorie de la décision est une approche pour étudier les problèmes des mathématiques statistiques [24] qui est fortement liée à la théorie des jeux.

Définition 1.4.1 (Jeu) *Un Jeu de deux joueurs à somme zéro est composé des éléments suivants:*

(i) Θ *– L'ensemble des états possibles.*

(ii) $\mathcal{A}$ *– L'ensemble des actions possibles.*

(iii) $L(\vartheta, a)$ *– Une fonction de perte défini sur* $\Theta \times \mathcal{A}$*.*

Un jeu composé de ces éléments est noté par $(\Theta, \mathcal{A}, L)$*.*

Définition 1.4.2 (Fonction de risque) *La valeur moyenne de $L(\vartheta, d(X))$, quand ϑ est l'état vrai, est appelée la fonction de risque:*

$$R(\vartheta, d) = E\{L(\vartheta, d(X))|\vartheta\} = \int L(\vartheta, d(X))dP(x|\vartheta) \tag{1.33}$$

Définition 1.4.3 (Règle de décision) *La fonction $d(x) : \mathcal{X} \longrightarrow \mathcal{A}$ est une règle de décision si la fonction de risque est finie pour tous les $\vartheta \in \Theta$.*

Après les notations générales, nous nous intéressons à la décision bayesienne:

Définition 1.4.4 (Risque de Bayes) *Le risque de Bayes de la règle de décision δ par rapport à la distribution a priori P est donnée par*

$$r(P, \delta) = E\{R(Y, \delta)\}, \tag{1.34}$$

où Y est une variable aléatoire sur Θ avec une distribution P.

Définition 1.4.5 (Règle de décision bayesienne) *Étant donné une distribution a priori P, δ_0 est une règle de décision bayesienne par rapport à P si*

$$r(P, \delta_0) = \inf_{\delta} r(P, \delta). \tag{1.35}$$

La valeur $r(P, \delta_0)$ est appelée le risque bayesien minimal.

1.5 Processus stochastiques et chaînes de Markov

Tout d'abord, nous définissons les processus stochastiques et ensuite nous étudions les chaînes de Markov qui seront utilisées dans le Chapitre 3 pour la démonstration de la convergence des algorithmes de relaxation.

Définition 1.5.1 (Processus stochastique) *A chaque événement $\xi \in \mathbf{I}$, nous attribuons une fonction de temps $X(t, \xi)$. La famille de ces fonctions est appelée un processus stochastique.*

L'autocorrélation d'un processus stochastique $X(t)$ est le moment joint des variables aléatoires $X(t_1)$ et $X(t_2)$:

$$R(t_1, t_2) = E\{X(t_1)X(t_2)\} \tag{1.36}$$

L'auto-covariance est la covariance de $X(t_1)$ et $X(t_2)$:

$$C(t_1, t_2) = E\{(X(t_1) - \mu(t_1))(X(t_2) - \mu(t_2))\} \tag{1.37}$$

En combinant les deux équations précédentes, nous obtenons:

$$C(t_1, t_2) = R(t_1, t_2) - \mu(t_1)\mu(t_2) \tag{1.38}$$

Définition 1.5.2 (Processus strictement stationnaire) *Un processus est strictement stationnaire si les processus $X(t)$ et $X(t+\epsilon)$ ont les mêmes statistiques pour n'importe quel ϵ.*

Définition 1.5.3 (Processus faiblement stationaire) *Un processus est faiblement stationaire si sa valeur moyenne est constante et son autocorrelation ne dépend que de $r = t_1 - t_2$:*

$$E\{X(t)\} = \mu, \quad E\{X(t+r)X(t)\} = R(r) \tag{1.39}$$

1.5.1 Chaînes de Markov

Définition 1.5.4 (Processus markovien) *Un processus stochastique $X(t)$ est un processus markovien si pour chaque n et pour chaque $t_1 < t_2 < \cdots < t_n$, on a*

$$P\{X(t_n) \leq x_n | X(t_{n-1}), \ldots, X(t_1)\} = P\{X(t_n) \leq x_n | X(t_{n-1})\} \tag{1.40}$$

Définition 1.5.5 (Chaîne de Markov) *Soit $\{X_i\} = X_1, X_2, \ldots, X_n, \ldots$ une séquence des variables aléatoires avec les valeurs possibles $a_1, \ldots, a_N$. Si la propriété markovienne est satisfaite:*

$$P\{X_n = a_{i_n} | X_{n-1} = a_{i_{n-1}}, \ldots, X_1 = a_{i_1}\} = P\{X_n = a_{i_n} | X_{n-1} = a_{i_{n-1}}\} \tag{1.41}$$

alors $\{X_i\}$ est une chaîne de Markov.

Les probabilités conditionnelles et non-conditionnelles sont notées par

$$p_i(n) = P\{X_n = a_i\} \tag{1.42}$$

$$P_{ij}(n, s) = P\{X_n = a_i | X_s = a_j\} \tag{1.43}$$

Les probabilités conditionnelles $P_{ij}(n,s)$ sont aussi appelées les *probabilitiés de transition*. Les équations suivantes montrent les propriétés évidentes:

$$p_i(n) = \sum_{j=1}^{N} P_{ij}(n,s)p_j(s) \tag{1.44}$$

$$\sum_{i=1}^{N} p_i(n) = 1 \tag{1.45}$$

$$\sum_{i=1}^{N} P_{ij}(n,s) = 1 \tag{1.46}$$

Les équations de Chapman-Kolmogorov sont données par:

$$P_{ij}(n,s) = \sum_{k=1}^{N} P_{ik}(n,r)P_{kj}(r,s) \tag{1.47}$$

Les probabilités de transition $P_{ij}(n,s)$ peuvent se mettre sous une forme matricielle et les probabilités non-conditionnelles $p_i(n)$ sous une forme vectorielle:

$$P(n,s) = \begin{pmatrix} P_{11}(n,s) & P_{12}(n,s) & \dots \\ P_{21}(n,s) & P_{22}(n,s) & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}, \quad p_i(n) = \begin{pmatrix} p_1(n) \\ p_2(n) \\ \vdots \end{pmatrix} \tag{1.48}$$

Définition 1.5.6 (Chaînes homogènes) *Si les probabilités conditionnelles $P_{ij}(n, n+1)$ ne dépendent pas du paramètre n, alors la chaîne est homogène et les probabilités de transition $P_{ij}(n, n+1)$ sont notées par P_{ij}.*

Définition 1.5.7 (Distribution stationnarie) *Si les probabilités non-conditionnelles p_k^n ne dépendent pas de n, c'est à dire*

$$\forall n: \ \ p_k^n = p_k \tag{1.49}$$

alors la distribution p_k est stationnaire.

Définition 1.5.8 (Chaîne irreducible) *Une chaîne de Markov est irréductible si pour toutes les paires des états (a_j, a_k), on peut accéder a_k à partir de a_j avec une probabilité positive pour un nombre fini des transitions:*

$$\forall a_j, a_k \ \ \exists n: \quad P_{jk}^n > 0. \tag{1.50}$$

Définition 1.5.9 (Apériodicité) *Une chaîne de Markov est apériodique si pour tous les états a_i, le plus grand diviseur commun des entiers $n \geq 1$, tel que $P_{ii}^n > 0$, egal à 1.*

Définition 1.5.10 (Ergodicité faible) *Une chaîne de Markov inhomogène est faiblement ergodique si pour tous les $m \geq 1$:*

$$\lim_{n\to\infty}(P_{ik}(m,n) - P_{jk}(m,n)) = 0 \tag{1.51}$$

Définition 1.5.11 (Ergodicité forte) *Une chaîne de Markov inhomogène est fortement ergodique s'il existe un vecteur π, satisfaisant:*

$$\sum_i \pi_i = 1, \quad \forall i : \pi_i > 0, \tag{1.52}$$

tel que pour tous les $m \geq 1$:

$$\lim_{n\to\infty} P_{ij}(m,n) = \pi_j. \tag{1.53}$$

1.6 Champs de Markov

L'utilisation des champs de Markov est devenue populaire depuis la publication des résultats de Geman et Geman [29] en 1984. Nous discutons ici les fondements de la théorie des champs markoviens (MRF) [55, 79, 68, 22]. Nous définissons les MRF de la façon la plus générale sur les graphes. Soit $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ un graphe où $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ est l'ensemble des sommets (ou des sites) et $\mathcal{E}$ est l'ensemble des arêtes.

Définition 1.6.1 (Voisins) *Deux points s_i et s_j sont voisins s'il existe une arête $e_{ij} \in \mathcal{E}$ entre eux. l'ensemble des points qui sont voisins d'un site s (c'est à dire le voisinage de s) est noté par $\mathcal{V}_s$.*

Définition 1.6.2 (Système de voisinage) *$\mathcal{V} = \{\mathcal{V}_s \mid s \in \mathcal{S}\}$ est un système de voisinage pour $\mathcal{G}$ si*

(i) $s \notin \mathcal{V}_s$

(ii) $s \in \mathcal{V}_r \Leftrightarrow r \in \mathcal{V}_s$

À chaque site du graphe, nous attribuons un étiquette λ appartenant à un ensemble fini des étiquettes Λ. Un tel étiquetage est appelé une configuration ω qui a une certaine probabilité $P(\omega)$. La restriction de ω à un sous-ensemble $T \subset \mathcal{S}$ est notée par ω_T et $\omega_s \in \Lambda$ désigne l'étiquette attribuée au site s. Aux paragraphes suivants, nous nous interessons aux mesures de probabilié assignés à l'ensemble de toutes les configurations possibles Ω.

Définition 1.6.3 (Champ de Markov) *$\mathcal{X}$ est un champ de Markov (MRF) par rapport à $\mathcal{V}$ si*

(i) pour tous les $\omega \in \Omega$: $P(\mathcal{X} = \omega) > 0$,

(ii) pour tous les $s \in \mathcal{S}$ et $\omega \in \Omega$:
$P(X_s = \omega_s \mid X_r = \omega_r, r \neq s) = P(X_s = \omega_s \mid X_r = \omega_r, r \in \mathcal{V}_s)$.

Définition 1.6.4 (Clique) *Un sous-ensemble $C \subseteq \mathcal{S}$ est un clique si chaque paire de sites dans C est voisine. $\mathcal{C}$ note l'ensemble des cliques et* $\deg(\mathcal{C}) = \max_{C \in \mathcal{C}} \mid C \mid$.

En utilisant la définition précédente, nous pouvons définir une *mesure de Gibbs* sur Ω. Soit V une *fonction potentielle* qui assigne un nombre $V_C(\omega)$ à chaque sous-configuration ω_C. La fonction d'énergie sur Ω est définie par:

$$U(\omega) = -\sum_{T} V_T(\omega). \tag{1.54}$$

Définition 1.6.5 (Distribution de Gibbs) *La distribution de Gibbs est une mesure de probabilité π sur Ω avec la représentation suivante:*

$$\pi(\omega) = \frac{1}{Z} \exp\left(-U(\omega)\right), \tag{1.55}$$

où Z est la constante de normalisation:

$$Z = \sum_{\omega} \exp\left(-U(\omega)\right),$$

Le théorème suivant fait la liaison entre les champs de Markov et la distribution de Gibbs [9, 68].

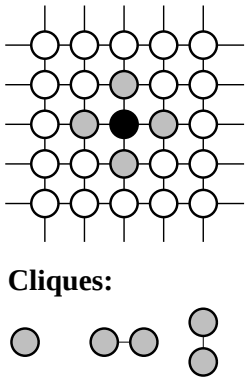

Figure 1.5: *Système de voisinage d'ordre un.*

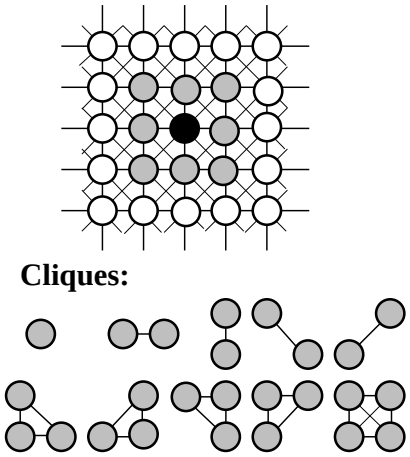

Figure 1.6: *Système de voisinage d'ordre deux.*

Théorème 1.6.1 (Hammersley-Clifford) *$\mathcal{X}$ est un champ de Markov par rapport au système de voisinage $\mathcal{V}$ si et seulement si $\pi(\omega) = P(\mathcal{X} = \omega)$ est une distribution de Gibbs, c'est à dire*

$$\pi(\omega) = \frac{1}{Z} \exp\left(- \sum_{C \in \mathcal{C}} V_C(\omega) \right) \tag{1.56}$$

1.6.1 Schémas spatiaux

Les schémas spatiaux sont les plus souvent utilisés en traitement d'images. Dans ce cas, nous considérons $\mathcal{S}$ comme une grille $\mathcal{L}$ telle que $\forall s \in \mathcal{S} : s = (i, j)$ et nous définissons les systèmes de vosisinage homogènes d'ordre n:

$$\mathcal{V}^n = \{\mathcal{V}^n_{(i,j)} : (i, j) \in \mathcal{L}\}, \tag{1.57}$$

$$\mathcal{V}^n_{(i,j)} = \{(k, l) \in \mathcal{L} : (k - i)^2 + (l - j)^2 \leq n\}. \tag{1.58}$$

Il est clair que $\mathcal{V}^0 \equiv \mathcal{S}$ et pour tous les $n \geq 0 : \mathcal{V}^n \subset \mathcal{V}^{n+1}$. La Figure 1.5 montre un système de voisinage d'ordre un ($n = 1$). Les cliques sont $\{(i, j)\}, \{(i, j), (i, j + 1)\}, \{(i, j), (i + 1, j)\}$. En pratique, les systèmes de voisinages d'ordre superieur à deux

ne sont pas utilisés car leur fonction d'énergie est trop compliquée et ils nécessitent un temps de calcul élevé.

Dans ce chapitre:

2.

Modèles markoviens d'images

La vision pré-attentive se réfère aux tâches de traitement des images digitales, traitant directement de larges quantités des pixels. Le but d'un tel traitement est de transformer les données en attributs significatifs (contours, texture, régions, etc. . .).

Les algorithmes utilisés sont souvent destinés à une seule application et parfois mis au point dans un environment spécifique. Un cadre général dans la vision pré-attentive est l'étiquetage d'images où nous voulons associer une étiquette à chaque pixel. Le sens de cet étiquette dépend du problème traité. Pour la restauration d'images, il représente les niveaux de gris; pour la détection de contour, il représente la présence ou la direction d'un élément de contour; pour la segmentation d'images, il représente les classes ou régions; etc. . . Le problème de base est comment choisir une étiquette pour chaque pixel. Notre approche est probabiliste: nous attribuons l'étiquette la plus probable à chaque pixel. Pour cela, nous avons besoin d'une mesure de probabilité sur l'ensemble des étiquetages possibles. Dans les images réelles, les pixels voisins ont souvent une intensité similaire. Dans un cadre probabiliste, cette régularité est bien exprimée par les champs de Markov. Une autre raison pour utiliser les modèles markoviens est *le théorème de Hammersley-Clifford* qui permet de définir les champs markoviens par les énergies potentielles. Dans le problème d'étiquetage, nous cherchons l'estimateur Maximum A Posteriori (MAP) du champ des étiquettes.

Malheureusement, trouver un tel étiquetage est une tâche très difficile. L'utilisation des modèles multi-grilles, proposée par les auteurs, rend le problème de minimisation plus facile. Nous proposons un nouveau type de modèle multi-grille que nous appelons modèle markovien hiérarchique. ce modèle permet de travailler avec des cliques contenant des sites éloignés pour un coût raisonable.

2.1 Un Modèle markovien général d'images

Nous présentons ici la formulation mathématique générale d'un modèle markovien d'images. Soit $\mathcal{R} = \{r_1, r_2, \ldots, r_M\}$ l'ensemble des sites et $\mathcal{F} = \{F_r : r \in \mathcal{R}\}$ l'ensemble des données (ou observations) sur ces sites. L'ensemble de toutes les observations possibles $f = (f_{r_1}, f_{r_2}, \ldots, f_{r_M})$ est noté par Φ. Nous avons un autre ensemble des site $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$, chacun de ces sites peut prendre une étiquette de $\Lambda = \{0, 1, \ldots, L-1\}$. L'espace des configurations Ω est l'ensemble de tous les étiquetages possibles $\omega = (\omega_{s_1}, \ldots, \omega_{s_N}), \omega_s \in \Lambda$. Les deux ensembles $\mathcal{R}$ et $\mathcal{S}$ ne sont pas nécessairement disjoints (par exemple le modèle de restauration de Geman [29] qui contient un processus de ligne). Notre but est de modéliser les étiquettes et les observations avec un champs aléatoire joint $(\mathcal{X}, \mathcal{F}) \in \Omega \times \Phi$. Le champ $\mathcal{X} = \{X_s\}_{s \in \mathcal{S}}$ est appelé *le champ des étiquettes* et $\mathcal{F} = \{F_r\}_{r \in \mathcal{R}}$ est appelé *le champ des observations.*

2.1.1 L'estimation bayesienne

Tout d'abord, nous construisons un estimateur bayesien de *champ des étiquettes.* Nous pouvons exprimer la probabilité jointe ainsi que la probabilité conditionnelle par les distributions a priori et a posteriori:

$$P_{\mathcal{X},\mathcal{F}}(\omega, f) = P_{\mathcal{F}|\mathcal{X}}(f \mid \omega) P_{\mathcal{X}}(\omega) \tag{2.1}$$

$$P_{\mathcal{X}|\mathcal{F}}(\omega \mid f) = \frac{P_{\mathcal{X},\mathcal{F}}(\omega, f)}{P_{\mathcal{F}}(f)} = \frac{P_{\mathcal{F}|\mathcal{X}}(f \mid \omega) P_{\mathcal{X}}(\omega)}{P_{\mathcal{F}}(f)} \tag{2.2}$$

$P(f)$ est constant car la réalisation du champ des observations est connue:

$$P_{\mathcal{X}|\mathcal{F}}(\omega \mid f) \propto P_{\mathcal{F}|\mathcal{X}}(f \mid \omega) P_{\mathcal{X}}(\omega) \tag{2.3}$$

L'estimateur est donné par la fonction de décision δ suivante: (voir Paragraphe 1.4):

$$\delta : \Phi \longrightarrow \Omega \tag{2.4}$$

$$f \mapsto \delta(f) = \hat{\omega} \tag{2.5}$$

Le *risque de Bayes* est donné par

$$r(P_X, \delta) = E\{R(\omega, \delta(f))\} \tag{2.6}$$

où $R(\omega, \delta(f))$ est la fonction de coût. En utilisant la Définition 1.4.5, l'estimateur doit avoir un risque de Bayes minimal:

$$\hat{\omega} = \arg\min_{\omega' \in \Omega} \int_{\omega \in \Omega} R(\omega, \omega') P_{\mathcal{X}|\mathcal{F}}(\omega \mid f) d\omega \tag{2.7}$$

Dans les paragraphes suivants, nous présentons les trois estimateurs bayesiens les plus connus [63].

2.1.1.1 Maximum A Posteriori (MAP)

L'estimateur MAP est le plus souvent utilisé en traitement d'images. Sa fonction de coût est définie par:

$$R(\omega, \omega') = 1 - \Delta_{\omega'}(\omega), \tag{2.8}$$

où $\Delta_{\omega'}(\omega)$ est la masse de Dirac en ω'. Il est claire que cette fonction donne le même coût pour chaque configuration différente de ω'. En utilisant l'Équation (2.7) et l'Équation (2.8), l'estimateur MAP du champ des étiquettes est donné par

$$\hat{\omega}^{MAP} = \arg\max_{\omega \in \Omega} P_{\mathcal{X}|\mathcal{F}}(\omega|f). \tag{2.9}$$

Cet estimateur, pour une observation donnée, fournit les modes de la distribution a posteriori. Cependant, l'Équation (2.9) pose un problème d'optimisation combinatoire et par conséquence, exige l'utilisation d'algorithmes spécifiques tel que le recuit simulé (voir Chapitre 3).

2.1.1.2 Modes a posteriori marginales (MPM)

La fonction de coût de l'estimateur MPM est définie par:

$$R(\omega, \omega') = \sum_{s \in \mathcal{S}} (1 - \Delta_{\omega'_s}(\omega_s)). \tag{2.10}$$

Nous remarquons que cette fonction est reliée au nombre des sites $s \in \mathcal{S}$ où $\omega_s \neq \omega'_s$. La solution de l' Équation (2.7) est donnée par:

$$\forall s \in \mathcal{S} : \hat{\omega}_s^{MPM} = \arg\max_{\omega_s \in \Lambda} P_{X_s|\mathcal{F}}(\omega_s \mid f), \tag{2.11}$$

qui permet de déterminer la configuration qui maximise à chaque site la distribution marginale a posteriori $P_{X_s|\mathcal{F}}(. \mid f)$.

2.1.1.3 Champ moyen (MF)

La fonction de coût est donnée par:

$$R(\omega, \omega') = \sum_{s \in \mathcal{S}} (\omega_s - \omega'_s)^2. \tag{2.12}$$

En utilisant l'Équation (2.7) et l'Équation (2.12), on a:

$$\forall s \in \mathcal{S} : \hat{\omega}_s^{MF} = \int_{\omega \in \Omega} \omega_s P_{\mathcal{X}|\mathcal{F}}(\omega \mid f) d\omega, \tag{2.13}$$

qui est la valeur moyenne conditionnelle de $\mathcal{X}$ sachant $\mathcal{F} = f$, c'est à dire *le champ moyen* de $\mathcal{X}$.

2.1.1.4 La distribution a priori

Supposons que $\mathcal{X}$ est un MRF avec le système de voisinage $\mathcal{V}' = \{\mathcal{V}'_s : s \in \mathcal{S}\}$ dont la distribution est définie par:

$$P(\mathcal{X} = \omega) = \frac{1}{Z} \exp\left(-U'(\omega)\right), \tag{2.14}$$

$$U'(\omega) = \sum_{C \in \mathcal{C}'} V'_C(\omega) \tag{2.15}$$

où $U'(\omega)$ est la fonction d'énergie (voir Paragraphe 1.6). Cette représentation utilise pour la définition de la probabilité a priori la distribution de Gibbs dont l'avantage est que l'on peut travailler avec les énergies potentielles sur les cliques au lieu de l'énergie globale.

2.1.1.5 Modèle d'images dégradé et la distribution a posteriori

Les observations sont reliées au processus des étiquettes par le modèle de dégradation qui modélise la relation entre le champ des étiquettes $\mathcal{X}$ et le processus des observations $\mathcal{F}$. La plupart des problèmes peuvent se formaliser par la fonction suivante [71]:

$$\mathcal{F} = \Psi(H(\mathcal{X}), N), \tag{2.16}$$

Au niveau des pixels:

$$\forall r \in \mathcal{R} : F_r = \Psi(H_r(X_{\psi(r)}), N_r) \tag{2.17}$$

où $\Psi(a, b)$ est une fonction inversible en a. H_r est une fonction locale définie sur un petit sous-ensemble $\psi(r)$ de $\mathcal{S}$ tel que $\psi(r) \in \mathcal{S}, \mid \psi(r) \mid \ll \mid \mathcal{S} \mid$ et $\psi^{-1}(s) = \{r \in \mathcal{R} \mid s \in \psi(r)\}$. N est une composante aléatoire (par exemple un bruit blanc gaussien). Si on suppose que la distribution de N est donnée par:

$$P_N(.) = \prod_{r \in \mathcal{R}} P_{N_r}(.) \tag{2.18}$$

alors, nous obtenons:

$$P_{\mathcal{F}|\mathcal{X}}(f \mid \omega) = \prod_{r \in \mathcal{R}} P_{N_r}(\Psi^{-1}(H_r(\omega_{\psi(r)}), f_r)). \tag{2.19}$$

En supposant que $P_{N_r}(.) > 0$ en chaque site $r \in \mathcal{R}$, la distribution conditionnelle du champ des observations $\mathcal{F}$ sachant $\mathcal{X}$, est définie par:

$$P_{\mathcal{F}|\mathcal{X}}(f \mid \omega) = \exp\left(\sum_{r \in \mathcal{R}} -\ln(P_{N_r}(\Psi^{-1}(H_r(\omega_{\psi(r)}), f_r)))\right), \tag{2.20}$$

En combinant l'équation précédente avec l'Équation (2.3) et l'Équation (2.14), la distribution a posteriori se met sous la forme suivante:

$$P_{\mathcal{X}|\mathcal{F}}(\omega \mid f) \propto \frac{1}{Z} \exp\left(\sum_{r\in\mathcal{R}} -\ln(P_{N_r}(\Psi^{-1}(H_r(\omega_{\psi(r)}), f_r))) + \sum_{C\in\mathcal{C}'} V'_C(\omega)\right) \tag{2.21}$$

On remarque que la distribution a posteriori est aussi une distribution de Gibbs avec le système de voisinage $\mathcal{V}$ le plus petit qui contient tous les cliques dans $\mathcal{C}'$ et les ensembles $\{\psi(r), r \in \mathcal{R}\}$:

$$\forall s \in \mathcal{S} : \mathcal{V}_s = \left(\bigcup_{r\in\psi^{-1}(s)} \psi(r) \setminus \{s\}\right) \cup \mathcal{V}'_s \tag{2.22}$$

Notons la fonction d'énergie correspondante par $U(\omega, f)$:

$$\begin{aligned} U(\omega, f) &= \sum_{r\in\mathcal{R}} -\ln(P_{N_r}(\Psi^{-1}(H_r(\omega_{\psi(r)}), f_r))) + \sum_{C\in\mathcal{C}'} V'_C(\omega) \\ &= \sum_{r\in\mathcal{R}} V_r(\omega_{\psi(r)}), f_r) + \sum_{C\in\mathcal{C}'} V'_C(\omega) \end{aligned} \tag{2.23}$$

Nous définissons maintenant $V_r(\omega_{\psi(r)}), f_r)$ d'une manière précise [71]:

$$V_r(\omega_{\psi(r)}), f_r) = V_r(\omega_{\psi(r)}) + \sum_{s\in\psi(r)} V_{s,r}(\omega_s, f_r). \tag{2.24}$$

Donc, elle peut se mettre sous la forme

$$\sum_{r\in\mathcal{R}} V_r(\omega_{\psi(r)}), f_r) = \sum_{r\in\mathcal{R}} V_r(\omega_{\psi(r)}) + \sum_{r\in\mathcal{R}} \sum_{s\in\psi(r)} V_{s,r}(\omega_s, f_r) \tag{2.25}$$

$$= \sum_{r\in\mathcal{R}} V_r(\omega_{\psi(r)}) + \sum_{s\in\mathcal{S}} \underbrace{\sum_{r\in\psi^{-1}(r)} V_{s,r}(\omega_s, f_r)}_{V_s(\omega_s, f_{\psi^{-1}(s)})} \tag{2.26}$$

$$\tag{2.27}$$

Finalement, nous avons la fonction d'énergie suivante:

$$U(\omega, f) = \sum_{s\in\mathcal{S}} V_s(\omega_s, f_{\psi^{-1}(s)}) + \sum_{C\in\mathcal{C}} V_C(\omega) \tag{2.28}$$

$$= U_1(\omega_s, f_{\psi^{-1}(s)}) + U_2(\omega). \tag{2.29}$$

où les énergies potentielles $V_C(\omega)$ des cliques sont définies par:

$$V_C(\omega) = \begin{cases} V'_C(\omega) & \text{si } C \in \mathcal{C}' \text{ et } C \notin \{\psi(r), r \in \mathcal{R}\} \\ V_r(\omega_{\psi(r)}) & \text{si } C = \psi(r) \text{ et } \psi(r) \notin \mathcal{C}' \\ V'_C(\omega) + V_r(\omega_{\psi(r)}) & \text{si } C = \psi(r) \text{ et } \psi(r) \in \mathcal{C}' \end{cases} \tag{2.30}$$

Si l'on suppose que l'image observée $\mathcal{F}$ en s ne dépend que du pixel s, l'Équation (2.28) se simplifie: $\psi(r)$ se réduit à s et le système de voisinage de la distribution a posteriori est équivalent à celui de la distribution a priori.

2.2 Un modèle de segmentation d'images

Dans cette partie, nous présentons le modèle de segmentation utilisé dans nos expériences. Le modèle est très simple parce qu'il n'utilise que les niveaux de gris. Notre but a été de mettre au point un modèle universel et d'étudier la mise en œuvre multi-échelle et hiérarchique de ce modèle.

Nous cherchons l'étiquetage $\hat{\omega}$ qui maximise la probabilité a posteriori $P(\omega \mid \mathcal{F})$, c'est à dire l'estimateur MAP du *champ des étiquettes* (cf. Paragraphe 2.1.1).

$$P(\omega \mid \mathcal{F}) = \frac{1}{P(\mathcal{F})} P(\mathcal{F} \mid \omega) P(\omega). \tag{2.31}$$

$P(\mathcal{F})$ ne dépend pas de l'étiquetage ω et nous supposons que

$$P(\mathcal{F} \mid \omega) = \prod_{s \in \mathcal{S}} P(f_s \mid \omega_s). \tag{2.32}$$

L'étiquetage recherché sera donné par

$$\hat{\omega} = \arg\max_{\omega \in \Omega} \prod_{s \in \mathcal{S}} P(f_s \mid \omega_s) \prod_{C \in \mathcal{C}} \exp(-V_C(\omega_C)) \ . \tag{2.33}$$

Cette équation montre que la probabilité a posteriori définit également un champ markovien. En supposant que $P(f_s \mid \omega_s)$ est gaussien, que la classe $\lambda \in \Lambda = \{0, 1, \ldots, L-1\}$ est représenté par sa valeur moyenne μ_λ et sa variance σ_λ, la fonction d'énergie devient (cf. Équation (2.29)):

$$U_1(\omega, \mathcal{F}) = \sum_{s \in \mathcal{S}} \left(\ln(\sqrt{2\pi}\sigma_{\omega_s}) + \frac{(f_s - \mu_{\omega_s})^2}{2\sigma_{\omega_s}^2} \right) \tag{2.34}$$

$$\text{et } U_2(\omega) = \sum_{C \in \mathcal{C}} V_2(\omega_C) \tag{2.35}$$

$$\text{où } V_2(\omega_C) = V_{\{s,r\}}(\omega_s, \omega_r) = \begin{cases} -\beta & \text{si } \omega_s = \omega_r \\ +\beta & \text{si } \omega_s \neq \omega_r \end{cases} \tag{2.36}$$

où β est l'hyperparamètre qui contrôl l'homogénité des régions. Nous avons $2L + 1$ paramètres notés par le vecteurΘ:

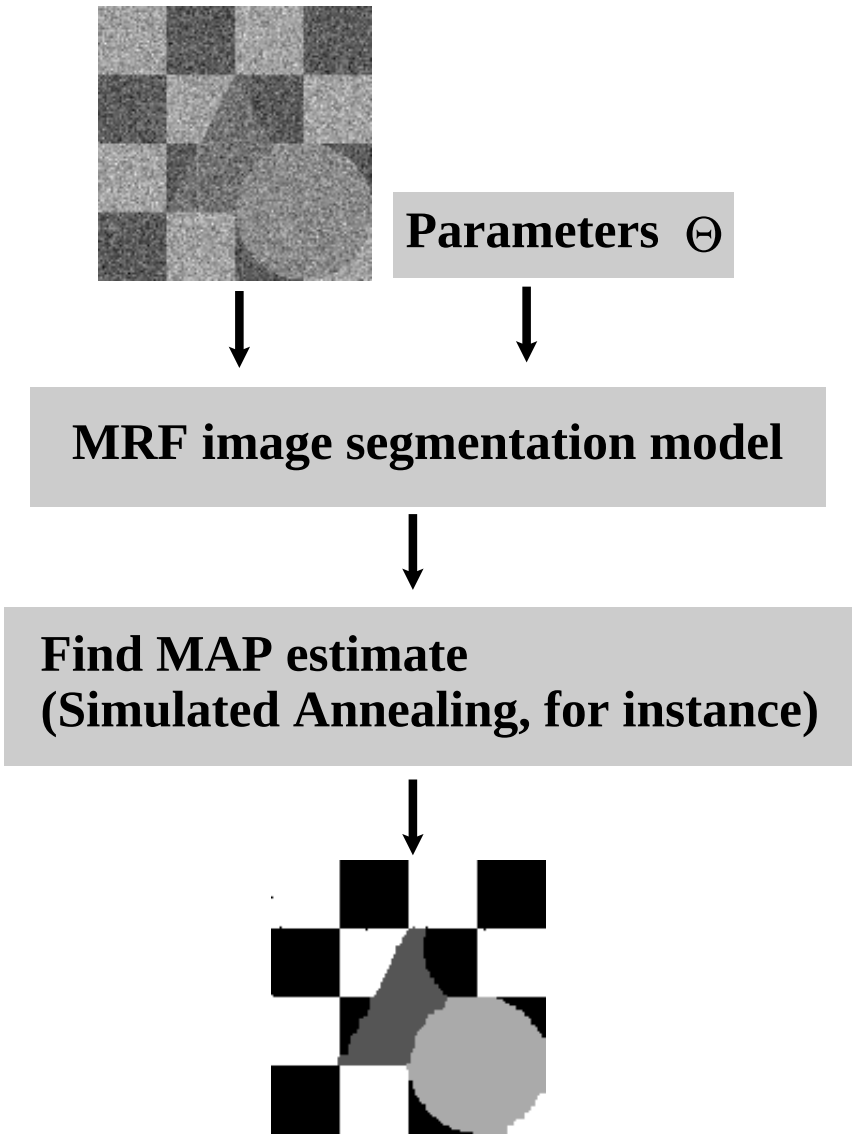

Figure 2.1: *Processus de segmentation supervisée.*

$$\Theta = \begin{pmatrix} \vartheta_0 \\ \vartheta_1 \\ \vdots \\ \vartheta_{2L} \end{pmatrix} \equiv \begin{pmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_{L-1} \\ \sigma_0 \\ \vdots \\ \sigma_{L-1} \\ \beta \end{pmatrix} \tag{2.37}$$

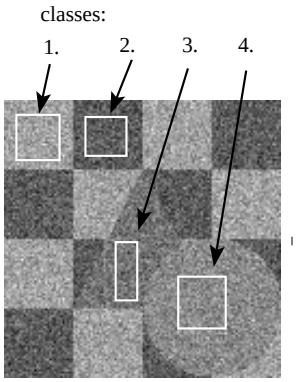

Figure 2.2: *Ensembles d'apprentissages sur une image synthétique.*

Lorsque les paramètres sont connus, le processus de segmentation est appelé *supervisé*. Dans le cas contraire, le processus est appelé *non-supervisé* La segmentation non-supervisée sera discutée dans le Chapitre 4.

Pour la segmentation supervisée, nous avons un ensemble d'apprentissage donné (petites sous-images), chacun représente une classe (voir Figure 2.2). En utilisant la loi des grands nombres (voir Paragraphe 1.3), les statistiques des classes (la moyenne et la variance) seront estimées par *la moyenne et la variance empiriques*:

$$\forall \lambda \in \Lambda : \qquad \mu_\lambda = \frac{1}{\mid S_\lambda \mid} \sum_{s \in S_\lambda} f_s, \tag{2.38}$$

$$\sigma_\lambda^2 = \frac{1}{\mid S_\lambda \mid} \sum_{s \in S_\lambda} (f_s - \mu_\lambda)^2, \tag{2.39}$$

où S_λ est l'ensemble des pixels appartenant à l'ensemble d'apprentissage de la classe λ. Le paramètre β est initialisé d'une façon empirique. Dans la Figure 2.1, nous présentons un exemple de processus de segmentation supervisée.

2.3 Un modèle markovien multi-échelle

Ce modèle a été proposé par Perez *et al.* [38, 37] pour la détection du mouvement.

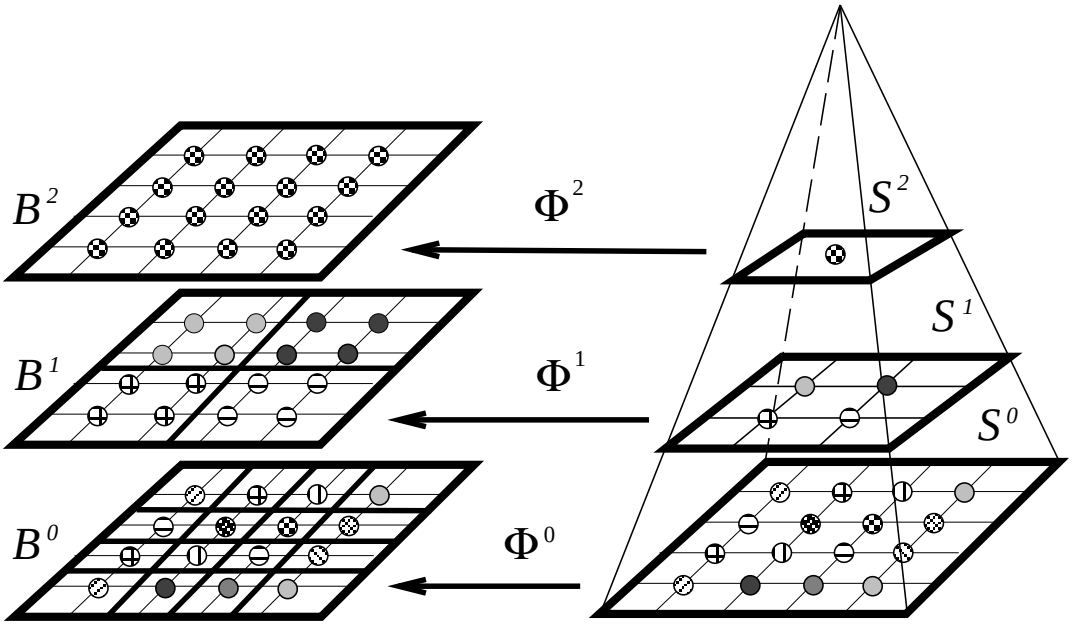

Figure 2.3: *L'isomorphisme Φ^i entre $\mathcal{B}^i$ et $\mathcal{S}^i$.*

2.3.1 Description générale

Supposons que $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ est une grille $W \times H$, telle que

$$\mathcal{S} \equiv \mathcal{L} = \{(i,j) : 1 \leq i \leq W \text{ et } 1 \leq j \leq H\}, \tag{2.40}$$

où $W = w^n$, $H = h^m$. Soit $\mathcal{V}$ un système de voisinage sur ces sites et $\mathcal{X}$ un champ de Markov sur $\mathcal{V}$ avec la fonction d'énergie U et les énergies potentielles $\{V_C\}_{C \in \mathcal{C}}$. Le processus suivant génère un modèle multi-échelle:

1. Soit $\mathcal{B}^0 \equiv \mathcal{S}$ et $\Omega_0 \equiv \Omega$.

2. Pour tous les $1 \leq i \leq M$ $(M = \inf(n,m))$, $\mathcal{S}$ est divisé en blocs de taille $w^i \times h^i$. Ces blocs forment une échelle $\mathcal{B}^i = \{b_1^i, \ldots, b_{N_i}^i\}$ $(N_i = N/(wh)^i)$.

Les étiquettes assignées aux sites d'un bloc sont les mêmes. L'étiquette commune du bloc b_k^i est notée par $\omega_k^i \in \Lambda$. Cette contrainte engendre un espace de configuration Ω_i qui est un sous-ensemble de l'espace originale Ω. Il est clair que pour chaque $0 \leq i \leq M$: $\Omega_i \subset \Omega_{i-1} \subset \cdots \subset \Omega_0 \equiv \Omega$.

Considérons maintenant le système de voisinage à une échelle i. Il est claire que b_k^i et b_l^i sont voisins si et seulement si il existe deux voisins $s \in \mathcal{S}$ et $r \in \mathcal{S}$ tel que $s \in b_k^i$ et $r \in b_l^i$. Nous avons donc les mêmes cliques que dans $\mathcal{C}$. Les cliques sont définies de la manière suivante. Soit $d = \deg(\mathcal{C})$. Pour $1 \leq j \leq d$, l'ensemble des blocs C_j^i contenant j sites sur l'échelle i est une clique de l'ordre j si il existe une clique $C \in \mathcal{C}$ (c'est à dire une clique sur l'échelle la plus fine) telle que:

$$\begin{aligned} &1. \quad C \subseteq \bigcup_{b_k^i \in C_j^i} b_k^i \\ &2. \quad \forall b_k^i \in C_j^i : C \cap b_k^i \neq \emptyset. \end{aligned}$$

L'ensemble des cliques sur l'échelle i est noté par $\mathcal{C}^i$ ($\mathcal{C}^0 \equiv \mathcal{C}$). L'ensemble de toutes les cliques, qui satisfait 1 et 2 pour C_j^i donné, est noté par $\mathcal{D}_{C_j^i} \subseteq \mathcal{C}$.

Partageons l'ensemble originale $\mathcal{C}$ en les ensembles disjoints suivantes: Pour chaque $1 \leq j \leq d$, soit $\mathcal{A}_j^i$ l'ensemble des cliques $C \in \mathcal{C}$ pour lesquels il existe une clique C_j^i (c'est à dire une clique d'ordre j sur l'échelle i) qui satisfait 1 et 2. En considérant la définition de $\mathcal{D}_{C_j^i}$ et $\mathcal{A}_j^i$, nous obtenons:

$$\mathcal{A}_j^i = \bigcup_{C_j^i \in \mathcal{C}^i} \mathcal{D}_{C_j^i}. \tag{2.41}$$

En utilisant cette décomposition, la fonction d'énergie U peut être écrite comme:

$$\begin{aligned} U(\omega) = \sum_{C \in \mathcal{C}} V_C(\omega) &= \sum_{C \in \mathcal{A}_1^i} V_C(\omega) + \cdots + \sum_{C \in \mathcal{A}_d^i} V_C(\omega) && (2.42) \\ &= \sum_{C_1^i \in \mathcal{C}^i} \sum_{C \in \mathcal{D}_{C_1^i}} V_C(\omega) + \cdots + \sum_{C_d^i \in \mathcal{C}^i} \sum_{C \in \mathcal{D}_{C_d^i}} V_C(\omega) && (2.43) \end{aligned}$$

L'avantage principale de cette décomposition est que l'on peut dériver les énergies potentielles sur des échelles grossières par un calcule simpl à partir de celles sur l'échelle la plus fine. Si nous notons les potentiels d'une clique C_j^i d'ordre j à l'échelle i par $V_{C_j^i}^{\mathcal{B}^i}$, nous obtenons à l'échelle $\mathcal{B}^i$ les potentiels suivantes:

$$V_{C_j^i}^{\mathcal{B}^i}(\omega) = \sum_{C \in \mathcal{D}_{C_j^i}} V_C(\omega) \tag{2.44}$$

Pour simplifier notre modèle, nous allons associer un site unique à chaque bloc. Ces sites ont l'étiquette commune du bloc correspondant et forment une grille grossière $\mathcal{S}^i$ qui est isomorphe à l'échelle $\mathcal{B}^i$ correspondant. L'espace des configurations $\Xi_i =$

$\{\xi_s^i : s \in \mathcal{S}^i, \xi_s^i \in \Lambda\}$ sur les échelles grossières est isomorphe à Ω_i. Il est clair, que $\Xi_0 \equiv \Omega_0 \equiv \Omega$. L'isomorphisme Φ^i de $\mathcal{S}^i$ à $\mathcal{B}^i$ n'est qu'une projection du champ des étiquettes grossières sur la grille la plus fine $\mathcal{S}^0 \equiv \mathcal{S}$:

$$\begin{aligned} \Phi^i \quad : \quad & \Xi_i \longrightarrow \Omega_i \\ & \xi^i \longmapsto \omega = \Phi^i(\xi^i). \end{aligned} \tag{2.45}$$

Φ^i garde le même système de voisinage sur $\mathcal{S}^i$ que sur $\mathcal{B}^i$ et les cliques sur $\mathcal{S}^i$ héritent les potentiels des cliques définis sur $\mathcal{B}^i$. Ces grilles forment une pyramide où le niveau i contient la grille $\mathcal{S}^i$. L'énergie du niveau i ($i = 0, \ldots, M$) est donnée par:

$$U^i(\xi^i) = \sum_{C^i \in \mathcal{C}^i} V_{C^i}^i(\xi^i) \quad i = 0, \ldots, M \tag{2.46}$$

$$\text{où } V_{C^i}^i(\xi^i) = V_{C^i}^{\mathcal{B}^i}(\Phi^i(\xi^i)). \tag{2.47}$$

L'algorithme multi-échelle va résoudre le problème de minimisation en utilisant une stratégie de descente dans la pyramide (voir Figure 2.5). Dans un premier temps, la couche la plus élevée de la pyramide est résolue ensuite, le niveau inférieur est initialisé par le résultat obtenu:

Algorithme 2.3.1 (Algorithme multi-échelle)

① *Soit $\mathcal{B}^0 \equiv \mathcal{S}$, $\Omega_0 \equiv \Omega$ et partageons $\mathcal{S}$ en blocs de la taille $w^i \times h^i$ ($1 \leq i \leq M$). Assigner un site unique à chaque bloc, ces sites forment une grille grossière.*

② *Calculer les potentiels sur les grilles grossières en utilisant l'Équation (2.47).*

③ *Soit $i = M$. Trouver le minimum global $\hat{\xi}^M$ de U^i dans l'Équation (2.46).*

④ *Initialiser le couche $i-1$ par la projection de $\hat{\xi}^i$ sur $\mathcal{S}^{i-1}$: $\xi^{i-1} = (\Phi^{i-1})^{-1} \circ \Phi^i(\hat{\xi}^i)$, et trouver le minimum $\hat{\xi}^{i-1}$ of U^{i-1}.*

⑤ *Arrêter si $i = 1$, sinon retourner à Étape ④ avec $i = i - 1$.*

Les avantages de cet algorithme sont clairs: chaque $\hat{\xi}^i$ donne un estimateur plus ou moins bon du résultat final. De l'autre côté, les niveaux plus hauts sont plus simples à résoudre car l'espace de configuration a moins d'éléments. Cet algorithme est particulièrement bien adapté à des algorithmes déterministes de relaxation qui sont plus sensibles à l'initialisation.

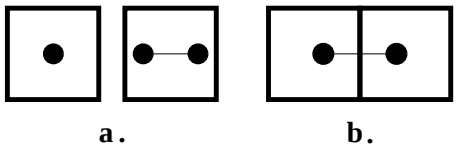

Figure 2.4: *Les deux sous-ensembles de $\mathcal{C}$ dans le cas d'un système de voisinage d'ordre 1. a: $\mathcal{C}_k^i$; b: $\mathcal{C}_{k,l}^i$.*

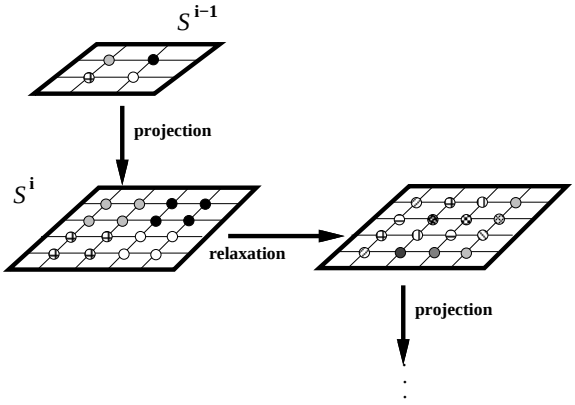

Figure 2.5: *Algorithme de relaxation multi-échelle.*

2.3.2 Un cas spécial

Dans ce paragraphe, nous examinons le modèle multi-échelle présenté précédemment. Supposons que le champ markovien soit défini sur un système de voisinage d'ordre 1 (voir Figure 1.5). La fonction d'énergie sera donnée par:

$$U(\omega, \mathcal{F}) = U_1(\omega, \mathcal{F}) + U_2(\omega). \tag{2.48}$$

U_1 (resp. U_2) désigne l'énergie des cliques d'ordre 1 (resp. d'ordre deux). La notation $U_1(\omega, \mathcal{F})$ signifie que les potentiels d'ordre un dépendent de l'étiquetage ainsi que de l'observation.

Pour établir le modèle multi-échelle correspondant, nous utilisons le processus décrit au Paragraphe 2.3.1. Soit $\mathcal{B}^i = \{b_1^i, \ldots, b_{N_i}^i\}$ l'ensemble des blocs et soit Ω_i l'espace des configurations sur l'échelle i ($\Omega_i \subset \Omega_{i-1} \subset \cdots \subset \Omega_0 = \Omega$). L'étiquette associée au bloc b_k^i est notée par ω_k^i. Nous pouvons définir le même système de voisinage sur $\mathcal{B}^i$ que sur $\mathcal{S}$:

$$b_k^i \text{ et } b_l^i \text{ sont voisins} \Longleftrightarrow \begin{cases} b_k^i \equiv b_l^i \text{ ou} \\ \exists C \in \mathcal{C} \mid C \cap b_k^i \neq \emptyset \text{ et } C \cap b_l^i \neq \emptyset \end{cases} \tag{2.49}$$

Maintenant, partageons l'ensemble original $\mathcal{C}$ en deux sous-ensembles disjoints $\{\mathcal{C}_k^i\}$ et $\{\mathcal{C}_{k,l}^i\}$:

1. Les cliques appartenant au bloc b_k^i(voir Figure 2.4/a.):

$$\mathcal{C}_k^i = \{C \in \mathcal{C} \mid C \subset b_k^i\} \tag{2.50}$$

2. Les cliques qui se trouvent entre deux blocs voisins $\{b_k^i, b_l^i\}$ (voir Figure 2.4/b.):

$$\mathcal{C}_{k,l}^i = \{C \in \mathcal{C} \mid C \subset (b_k^i \cup b_l^i) \text{ et } C \cap b_k^i \neq \emptyset \text{ et } C \cap b_l^i \neq \emptyset\} \tag{2.51}$$

La fonction d'énergie peut être décomposée de la façon suivante (voir Équation (2.48)):

$$\begin{aligned} U_1(\omega, \mathcal{F}) &= \sum_{s \in \mathcal{S}} V_1(\omega_s, f_s) \\ &= \sum_{b_k^i \in \mathcal{B}^i} \underbrace{\sum_{s \in b_k^i} V_1(\omega_s, f_s)}_{V_1^{\mathcal{B}^i}(\omega_k^i, \mathcal{F})} = \sum_{b_k^i \in \mathcal{B}^i} V_1^{\mathcal{B}^i}(\omega_k^i, \mathcal{F}) \qquad (2.52) \\ \text{et } U_2(\omega) &= \sum_{C \in \mathcal{C}} V_2(\omega_c) \\ &= \sum_{b_k^i \in \mathcal{B}^i} \underbrace{\sum_{C \in \mathcal{C}_k^i} V_2(\omega_c)}_{V_k^{\mathcal{B}^i}(\omega_k^i)} + \sum_{\{b_k, b_l\} voisins} \underbrace{\sum_{C \in \mathcal{C}_{k,l}^i} V_2(\omega_c)}_{V_{k,l}^{\mathcal{B}^i}(\omega_k^i, \omega_l^i)} \\ &= \sum_{b_k^i \in \mathcal{B}^i} V_k^{\mathcal{B}^i}(\omega_k^i) + \sum_{\{b_k, b_l\} voisins} V_{k,l}^{\mathcal{B}^i}(\omega_k^i, \omega_l^i) \qquad (2.53) \end{aligned}$$

Nous pouvons définir maintenant la pyramide (cf. Figure 2.3) où le niveau i contient la grille grossière $\mathcal{S}^i$ qui est isomorphe à l'échelle $\mathcal{B}^i$. L'espace de configuration réduit des grilles grossières est noté par $\Xi_i = \Lambda^{N_i}$.

Le modèle sur les grilles $\mathcal{S}^i$ $(i = 0, \ldots, M)$ défini un ensemble des modèles multi-échelles dont la fonction d'énergie est dérivée des Équations (2.52) et (2.53):

$$\begin{aligned} U^i(\xi^i, \mathcal{F}) &= U_1^i(\xi^i, \mathcal{F}) + U_2^i(\xi^i) \qquad (2.54) \\ &= U_1(\Phi^i(\xi_i), \mathcal{F}) + U_2(\Phi^i(\xi_i)) \quad i = 0, \ldots, M \\ \text{avec } U_1^i(\xi^i, \mathcal{F}) &= \sum_{k \in \mathcal{S}^i} (V_1^{\mathcal{B}^i}(\omega_k^i, \mathcal{F}) + V_k^{\mathcal{B}^i}(\omega_k^i)) = \sum_{k \in \mathcal{S}^i} V_1^i(\xi_k^i, \mathcal{F}) \qquad (2.55) \\ \text{et } U_2^i(\xi^i) &= \sum_{\{k,l\} voisins} V_{k,l}^{\mathcal{B}^i}(\omega_k^i, \omega_l^i) = \sum_{C^i \in \mathcal{C}^i} V_2^i(\xi_C^i) \qquad (2.56) \end{aligned}$$

où C^i est une clique d'ordre deux correspondant à la définition dans Équation (2.49) et $\mathcal{C}^i$ est l'ensemble des cliques sur la grille $\mathcal{S}^i$.

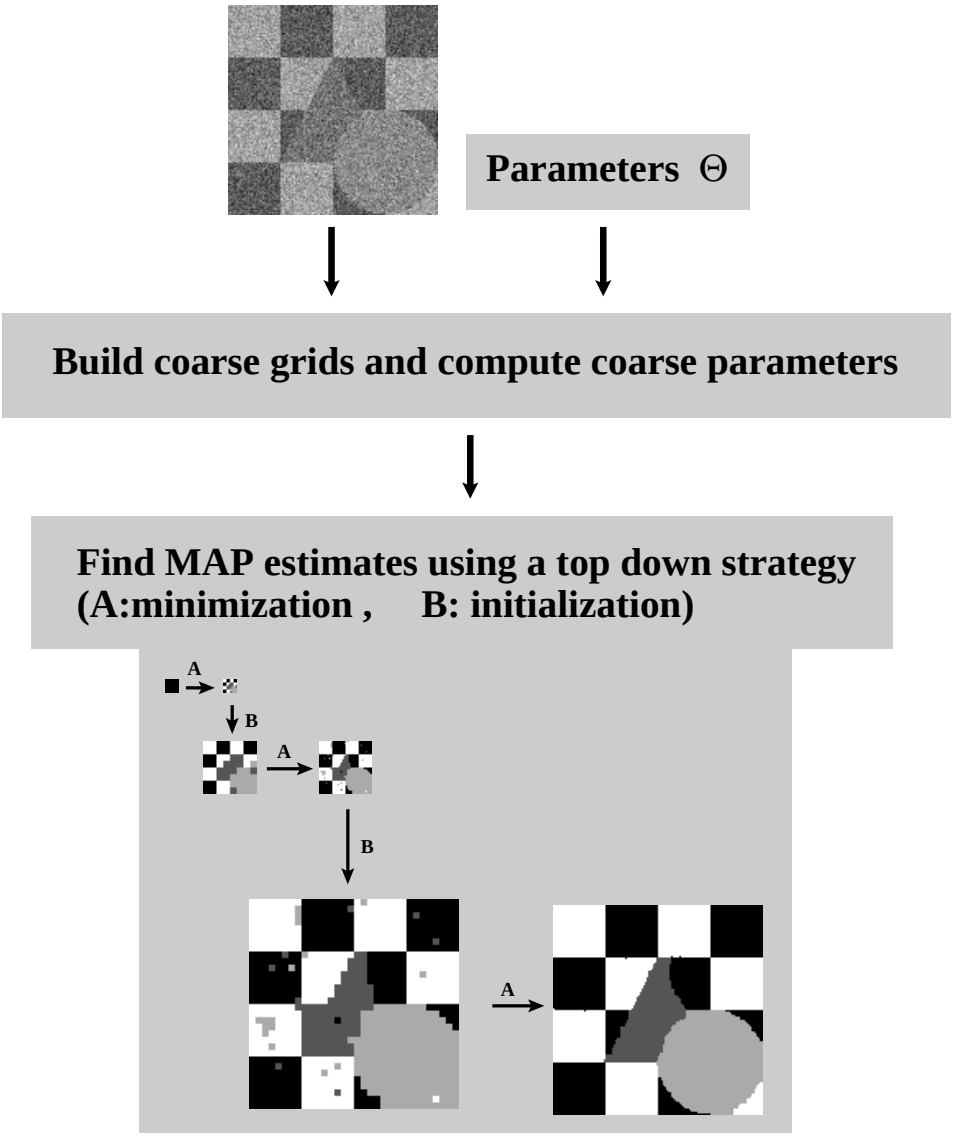

Figure 2.6: *Algorithme multi-échelle de segmentation d'images supervisée.*

2.3.3 Application à la segmentation d'images

Nous pouvons facilement adapter les équations obtenus dans le paragraphe précédent au modèle de segmentation présenté au Paragraphe 2.2. Pour simplifier les calculs, nous supposons que la taille d'un bloc est $n \times n$ (c'est à dire $w = h = n$). Nous obtenons donc [48, 49, 46]:

$$\begin{aligned}
U_1^i(\xi^i, \mathcal{F}) &= \sum_{s^i \in \mathcal{S}^i} V_1^i(\xi_{s^i}^i, \mathcal{F}) \\
\text{où } V_1^i(\xi_{s^i}^i, \mathcal{F}) &= \sum_{s \in b_{s^i}^i} V_1(\omega_s, f_s) + \sum_{C \in \mathcal{C}_{s^i}^i} V_2(\omega_C) \\
&= \sum_{s \in b_{s^i}^i} \left(\log(\sqrt{2\pi}\sigma_{\omega_s}) + \frac{(f_s - \mu_{\omega_s})^2}{2\sigma_{\omega_s}^2} \right) - p^i\beta \qquad (2.57) \\
\text{et } U_2^i(\xi^i) &= \sum_{C^i = \{r^i, s^i\} \in \mathcal{C}^i} V_2^i(\xi_{C^i}^i) \\
\text{où } V_2^i(\xi_{C^i}^i) &= \sum_{\{r,s\} \in \mathcal{D}_{C^i}} V_2(\omega_r, \omega_s) = \begin{cases} -q^i\beta & \text{si } \omega_r = \omega_s \\ +q^i\beta & \text{si } \omega_r \neq \omega_s \end{cases} \qquad (2.58)
\end{aligned}$$

Les valeurs p^i et q^i dépendent de la taille choisie des blocs et du système de voisinage. p^i est le nombre des cliques appartenant au même bloc sur l'échelle $\mathcal{B}^i$ et q^i est le nombre des cliques qui se trouvent entre deux blocs voisins sur l'échelle $\mathcal{B}^i$. Si nous considérons des blocs de la taille $n \times n$ et un système de voisinage d'ordre 1, on obtient alors:

$$\begin{aligned}
p^i &= 2n^i(n^i - 1) \qquad (2.59) \\
q^i &= n^i \qquad (2.60)
\end{aligned}$$

Dans la Figure 2.6, nous présentons un algorithme multi-échelle de segmentation *supervisée*. Comme dans le cas mono-grille, nous avons deux entrées: l'image observée et les paramètres $\Theta \equiv \Theta^0$ définis dans l'Équation (2.37). Ensuit nous établissons la pyramide et calculons les paramètres $\Theta^i (i = 1, \ldots, M)$ sur les grilles grossières. De cette façon, nous obtenons $M + 1$ fonctions d'énergie définies par l'Équation (2.57) et l'Équation (2.58). En utilisant une stratégie *descendante* dans la pyramide, nous minimisons les fonctions d'énergie et prenons l'étiquetage finale sur le niveau le plus fin comme le résultat final de la segmentation.

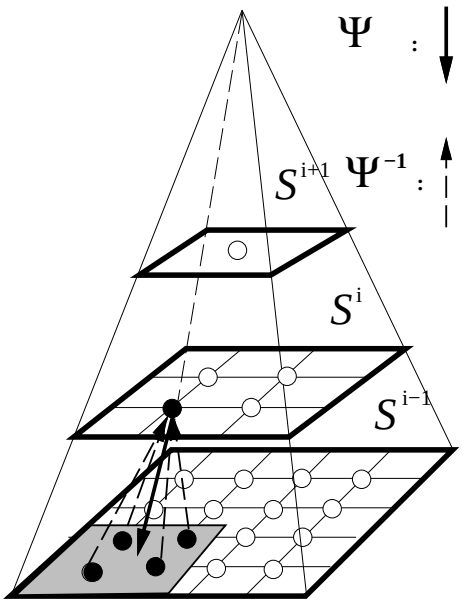

Figure 2.7: *La fonctions* Ψ *et* Ψ^{-1}

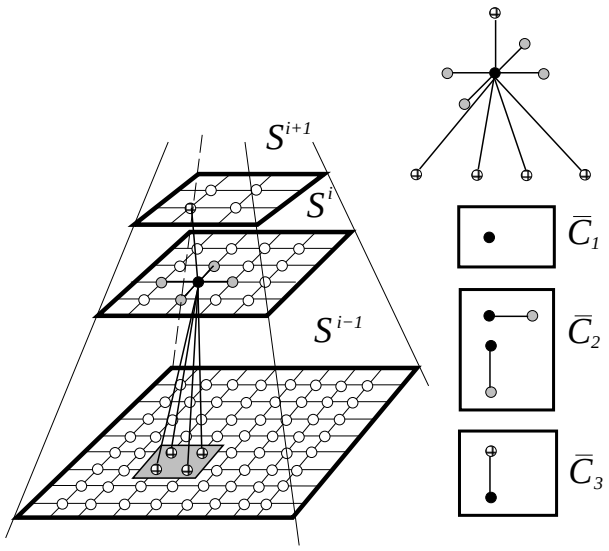

Figure 2.8: *Le système de voisinage* $\bar{\mathcal{V}}$ *et les cliques* $\bar{\mathcal{C}}_1$, $\bar{\mathcal{C}}_2$ *et* $\bar{\mathcal{C}}_3$.

2.4 Le modèle hiérarchique

Dans ce paragraphe, nous proposons un nouveau modèle hiérarchique [47, 50, 48, 49, 46]. L'idée de base est de fournir une meilleure communication entre les différentes niveaux de la pyramide que l'initialisation. Notre approche est d'introduire des interactions nouvelles entre les grilles voisines. Elle permet également la parallélisation des algorithmes de relaxation sur toute la pyramide.

2.4.1 Description générale

Nous considérons la pyramide décrite dans le paragraphe précédent. Notons par $\bar{\mathcal{S}} = \{\bar{s}_1, \ldots, \bar{s}_{\bar{N}}\}$ les sites de cette pyramide:

$$\begin{aligned} \bar{\mathcal{S}} &= \bigcup_{i=0}^{M} \mathcal{S}^i \\ \bar{N} &= \sum_{i=0}^{M} N_i. \end{aligned} \tag{2.61}$$

$\bar{\Omega}$ désigne l'espace des configurations de la pyramide:

$$\begin{aligned} \bar{\Omega} &= \Xi^0 \times \Xi^1 \times \cdots \times \Xi^M \\ &= \{\bar{\omega} \mid \bar{\omega} = (\xi^0, \xi^1, \ldots, \xi^M)\} \end{aligned} \tag{2.62}$$

Définissons la fonction suivante Ψ entre deux niveaux voisins qui associe à un site donné le bloc correspondant au-desous (c'est à dire ses descendants). Ψ^{-1} associe l'ancêtre à un site donné (voir Figure 2.7):

$$\begin{aligned} \Psi: \quad & \mathcal{S}^i \longrightarrow \mathcal{S}^{i-1} \\ \Psi(\bar{s}) &= \{\bar{r} \mid \bar{s} \in \mathcal{S}^i \Rightarrow \bar{r} \in \mathcal{S}^{i-1} \text{ et } b^{i-1}_{\bar{r}} \subset b^i_{\bar{s}}\} \end{aligned} \tag{2.63}$$

Nous pouvons maintenant définir sur ces sites le système de voisinage suivant (voir Figure 2.8):

$$\bar{\mathcal{V}} = (\bigcup_{i=0}^{M} \mathcal{V}_i) \cup \{\Psi^{-1}(\bar{s}) \cup \Psi(\bar{s}) \mid \bar{s} \in \bar{\mathcal{S}}\} \tag{2.64}$$

où $\mathcal{V}_i$ est le système de voisinage au niveau i. Les cliques sont alors:

$$\bar{\mathcal{C}} = (\bigcup_{i=0}^{M} \mathcal{C}^i) \cup \mathcal{C}^* \tag{2.65}$$

où $\mathcal{C}^*$ désigne les nouvelles cliques entre deux niveaux voisins. Le degré des nouvelles cliques dépend de la taille des blocs: chaque site communique avec son ancêtre et ses descendants. On a donc:

$$\deg(\mathcal{C}^*) = \max_{C^* \in \mathcal{C}^*} \mid C^* \mid = wh + 2 \tag{2.66}$$

$$\text{et } \deg(\bar{\mathcal{C}}) = \deg(\mathcal{C}) + \deg(\mathcal{C}^*) - 1. \tag{2.67}$$

Soit $\bar{\mathcal{X}}$ un champ de Markov sur $\bar{\mathcal{V}}$ avec une fonction d'énergie $\bar{U}$ et des potentiels $\{\bar{V}_{\bar{C}}\}_{\bar{C} \in \bar{\mathcal{C}}}$. La fonction d'énergie est donnée par:

$$\begin{aligned} \bar{U}(\bar{\omega}) &= \sum_{\bar{C} \in \bar{\mathcal{C}}} \bar{V}_{\bar{C}}(\bar{\omega}) \\ &= \sum_{i=0}^{M} \sum_{\bar{C} \in \mathcal{C}^i} V^i_{\bar{C}}(\bar{\omega}) + \sum_{\bar{C} \in \mathcal{C}^*} \bar{V}_{\bar{C}}(\bar{\omega}) \\ &= \sum_{i=0}^{M} \sum_{C^i \in \mathcal{C}^i} V^i_{C^i}(\xi^i) + \sum_{C^* \in \mathcal{C}^*} \bar{V}_{C^*}(\bar{\omega}) \\ &= \sum_{i=0}^{M} U^i(\xi^i) + U^*(\bar{\omega}) \end{aligned} \tag{2.68}$$

Nous remarquons que la fonction d'énergie se compose de deux termes. Le premier correspond à la somme des fonctions d'énergie des grilles définies au chapitre précédente et le second ($U^*(\bar{\omega})$) correspond à l'énergie des cliques inter-niveaux.

Étant donné que la fonction d'énergie est définie sur toute la pyramide, l'estimateur MAP est obtenu par la minimisation de cette fonction d'énergie exprimée par l'Équation (2.68). Les algorithmes utilisés sont essentiellement les mêmes que dans le cas mono-grille. Cependant nous pouvons mettre en œuvre un algorithme parallèle spécial grâce au structure pyramidale en définissant un nouveau type de recuit proposé dans Chapitre 3. Le résultat final est obtenu au plus bas niveau.

2.4.2 Un cas spécial

Dans ce paragraphe, nous étudions le modèle dans le cas d'un système de voisinage d'ordre un. Nous ne considérons ici que les cliques d'ordre un et deux. Nous pouvons regrouper les cliques en trois sous-ensembles disjoints $\bar{\mathcal{C}}_1, \bar{\mathcal{C}}_2, \bar{\mathcal{C}}_3$ correspondant aux cliques d'ordre un, aux cliques d'ordre deux au même niveau et aux cliques inter-niveaux d'ordre deux (voir Figure 2.8). En utilisant cette décomposition, nous pouvons définir la fonction d'énergie suivante:

$$\bar{U}(\bar{\omega}, \mathcal{F}) = \bar{U}_1(\bar{\omega}, \mathcal{F}) + \bar{U}_2(\bar{\omega}) \tag{2.69}$$

$$\begin{aligned} \bar{U}_1(\bar{\omega}, \mathcal{F}) &= \sum_{\bar{s} \in \bar{\mathcal{S}}} \bar{V}_1(\bar{\omega}_{\bar{s}}, \mathcal{F}) \\ &= \sum_{i=0}^{M} \sum_{s^i \in \mathcal{S}^i} V_1^i(\xi^i_{s^i}, \mathcal{F}) = \sum_{i=0}^{M} U_1^i(\xi^i, \mathcal{F}) \end{aligned} \tag{2.70}$$

$$\begin{aligned} \bar{U}_2(\bar{\omega}) &= \sum_{C \in \bar{\mathcal{C}}_2} \bar{V}_2(\bar{\omega}_C) + \sum_{C \in \bar{\mathcal{C}}_3} \bar{V}_2(\bar{\omega}_C) \\ &= \sum_{i=0}^{M} \sum_{C \in \mathcal{C}^i} V_2^i(\xi^i_C) + \sum_{C \in \bar{\mathcal{C}}_3} \bar{V}_2(\bar{\omega}_C) \\ &= \sum_{i=0}^{M} U_2^i(\xi^i) + \sum_{C \in \bar{\mathcal{C}}_3} \bar{V}_2(\bar{\omega}_C) \end{aligned} \tag{2.71}$$

2.4.3 Complexité

Ici, nous examinons la complexité de l'optimisation du modèle hiérarchique en fonction de la mémoire nécessaire (ou le nombre des processeurs dans l'implantation parallèle) et des communications nécessaires par rapport au modèle mono-grille.

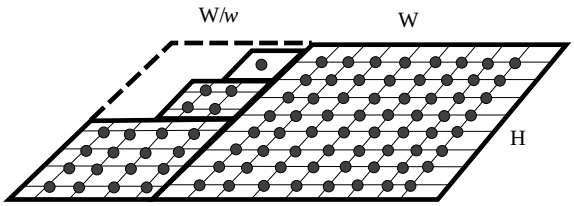

Figure 2.9: *Complexité de mémoire du modèle hiérarchique.*

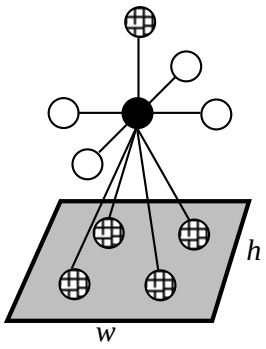

Figure 2.10: *Communication du modèle hiérarchique.*

Mémoire/processeur Supposons que la taille de l'image traitée soit $W \times H$. En suivant le processus décrit dans le Paragraphe 2.4.1, nous établirons une pyramide contenant $M+1$ niveaux. Sans perte de généralité, nous supposons que $W/w \leq H/h$, où $w \times h$ est la taille des blocs (w et h sont supérieurs à deux). Le modèle hiérarchique demande au maximum $(1+1/w)WH$ processeurs (cf. Équation (2.72)), car tous les niveaux doivent être stockés au même temps. La mémoire (ou le nombre des processeurs) nécessaire pour stocker les niveaux (voir Figure 2.9) est donnée par:

$$WH + \frac{WH}{wh} + \frac{WH}{(wh)^2} + \cdots + \frac{WH}{(wh)^M} = WH \sum_{i=0}^{M} \frac{1}{(wh)^i} < \left(1 + \frac{1}{w}\right) WH \qquad (2.72)$$

Communication En tenant compte que des cliques d'ordre un et deux, (le plus souvent utilisé en pratique, voir Figure 2.10), il est clair que nous avons $(wh+1)$ plus de communications par processeur que dans le cas mono-grille. Chaque site communique avec son ancêtre et ses descendants (il y en a wh).

Nous pouvons donc constater que le modèle proposé demande plus de processeurs et plus de communication que le modèle classique. Malgré cet inconvénient, nous verons plus loins (Paragraphe 2.5), que les expériments montrent que les résultats obtenus par le modèle hiérarchique sont meilleurs que ceux obtenus par les modèles classiques.

2.4.4 Application à la segmentation d'images

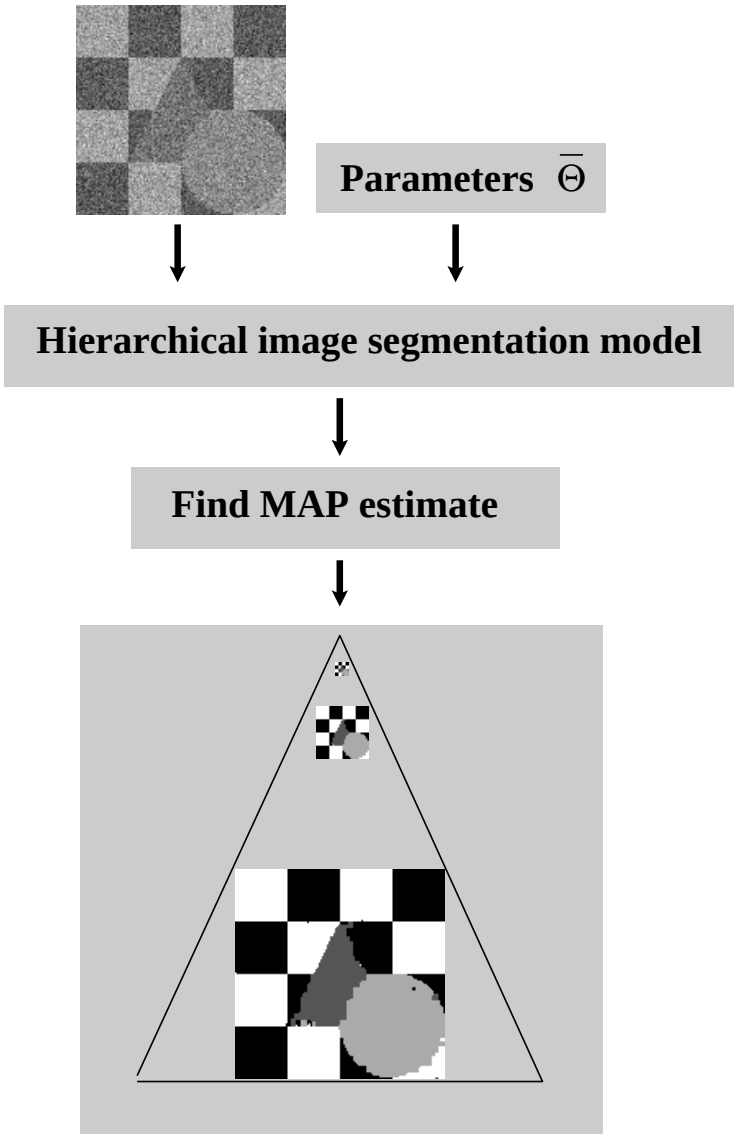

Figure 2.11: *Algorithme de segmentation hiérarchique supervisée.*

En adaptant le modèle présenté au Paragraphe 2.2, nous pouvons définir la version hiérarchique en utilisant l'Équation (2.70) et l'Équation (2.71) [48, 49, 47]:

$$\bar{U}_1(\bar{\omega}, \mathcal{F}) = \sum_{i=0}^{M} \sum_{s^i \in \mathcal{S}^i} V_1^i(\xi^i, \mathcal{F}) \tag{2.73}$$

$$\text{et } \bar{U}_2(\bar{\omega}) = \sum_{i=0}^{M} \sum_{C^i \in \mathcal{C}^i} V_2^i(\xi^i_{C^i}) + \sum_{C \in \bar{\mathcal{C}}_3} \bar{V}_2(\bar{\omega}_c) \tag{2.74}$$

$$\text{où } \bar{V}_2(\bar{\omega}_c) = \bar{V}_{\{\bar{s},\bar{r}\}}(\bar{\omega}_{\bar{s}}, \bar{\omega}_{\bar{r}}) = \begin{cases} -\gamma & \text{si } \bar{\omega}_{\bar{s}} = \bar{\omega}_{\bar{r}} \\ +\gamma & \text{si } \bar{\omega}_{\bar{s}} \neq \bar{\omega}_{\bar{r}} \end{cases} \tag{2.75}$$

où V_1^i et V_2^i sont définis par l'Équation (2.57) et l'Équation (2.58). Dans ces formules, le nouveau paramètre γ favorise les classes similaires entre un site, son ancêtre et ses descendants. Nous avons donc les paramètres suivantes (voir aussi l'Équation (2.37)):

$$\bar{\Theta} = \begin{pmatrix} \bar{\vartheta}_0 \\ \bar{\vartheta}_1 \\ \vdots \\ \bar{\vartheta}_{2L+1} \end{pmatrix} \equiv \begin{pmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_{L-1} \\ \sigma_0 \\ \vdots \\ \sigma_{L-1} \\ \beta \\ \gamma \end{pmatrix} \tag{2.76}$$

Dans la Figure 2.11, nous présentons un algorithme de segmentation hiérarchique supervisé. Nous avons deux entrées: l'image observée et les paramètres $\bar{\Theta}$. Les fonctions d'énergie sont définies par l'Équation (2.73)–Équation (2.75). Pour trouver le minimum de cette fonction, nous utilisons essentiellement les mêmes algorithmes que dans le cas mono-grille (Iterated Conditional Mode dans Figure 2.11). Le résultat est l'étiquetage obtenu au plus bas niveau de la pyramide.

2.5 Résultats expérimentaux

Nous comparons l'échantilloneur de Gibbs [29] et l'ICM [8, 43] en utilisant chaque modèle pour chaque algorithme. Les algorithmes ont été mis en œuvre sur la machine à connexions CM200 [39] avec $8K$ processeurs. Dans les tableaux (voir Annexe 2.B),

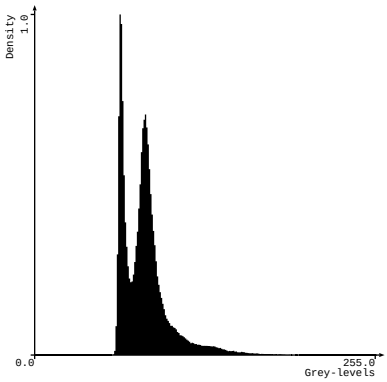

Figure 2.12: *Histogramme de l'image "assalmer" avec 6 classes.*

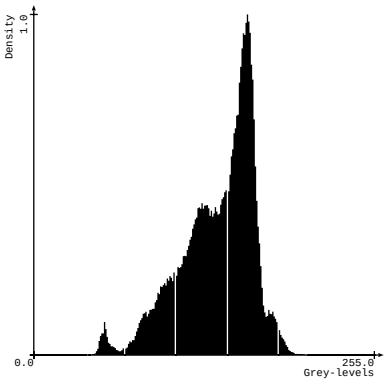

Figure 2.13: *Histogramme de l'image "holland" avec 10 classes.*

nous donnons pour chaque modèle et chaque algorithme le nombre de niveaux de la pyramide (pour le modèle mono-grille, c'est toujours un), le VPR [39], la température initiale (pour le modèle hiérarchique, nous utilisons des températures différentes avec le recuit multi-température), le nombre d'itérations, le temps d'exécution, l'erreur de classification (= le nombre des pixels mal-classés), le paramètre β (voir Équations (2.36), (2.57), (2.58)) et γ (voir Équation (2.75)).

2.5.1 Comparaison des modèles

Dans un premier temps, nous avons comparé les modèles sur les images synthétiques de la taille 128×128. La première image est une image de damier (voir Figure 2.14, Table 2.3) avec 2 classes et S/B (voir Équation (1.26) pour la définition) égal à $-5dB$. On remarque que le modèle multi-échelle donne de meilleurs résultats que le modèle mono-grille, en particulier avec l'algorithme ICM. La structure rectangulaire de l'image est bien adaptée au modèle multi-échelle qui est composé de blocs rectangulaires. Le modèle hiérarchique donne les meilleurs résultats avec les deux algorithmes, mais le temps d'exécution est beaucoup plus grand car, dans ce cas, nous ne pouvons pas utiliser la communication rapide de type "NEWS", et le VPR est plus grand que pour les autres modèles.

Dans la deuxième image, se trouvent des formes géométriques différentes (un cercle et un triangle, voir Figure 2.15, Table 2.4). Dans ce cas, nous avons étudié la sensitivité géométrique des modèles. L'échantilloneur de Gibbs donne pratiquement le même résultat dans tous les cas. Pourtant, l'ICM est plus sensible aux conditions initiales. Le

modèle multi-échelle donne un meilleur résultat que le modèle mono-grille mais le résultat n'est pas assez fin dans le triangle et le cercle car l'ICM n'est pas capable de corriger les erreurs d'initialisation dans ces régions. Pour le modèle hiérarchique, la communication inter-niveaux a permis d'obtenir un résultat aussi bon qu'avec l'échantilloneur de Gibbs.

La troisième image est un échiquier avec 16 classes (voir Figure 2.16 et Table 2.5). Pour l'ICM, il y a une amélioration considérable pour les modèles pyramidaux mais pour l'échantilloneur de Gibbs, nous n'observons qu'une amélioration faible.

Dans les Figures 2.17, 2.18 et 2.19, nous montrons quelques images réelles de taille 256×256: une image SPOT avec 4 classes (voir Figure 2.17, Table 2.6), une image de couloir avec 4 classes (voir Figure 2.18) et une image médicale avec 3 classes (voir Figure 2.19).

L'image suivante est une image SPOT de la taille 512×512 (voir Figure 2.20) avec les vérités terrains (voir Figure 2.21). Dans le tableau suivant (Tableau 2.1), nous donnons la moyenne (μ) et la variance (σ^2) pour chaque classe (6 classes).

classe	1	2	3	4	5	6
μ	65.3	81.3	75.4	98.5	82.5	129.0
σ^2	6.4	12.7	14.9	16.8	9.46	183.2

Tableau 2.1: *Paramètres de l'image "assalmer".*

Nous pouvons constater que les classes 2 et 5 ont des paramètres similaires, il est donc difficile de les distinguer. La Figure 2.12 montre l'histogramme de l'image originale. Nous pouvons facilement distinguer deux sommets (aux environs 64, 80, et 120) mais les autres classes sont confondues. La Figure 2.22 (resp. Figure 2.23) montre les résultats obtenus avec l'ICM (resp. l'échantilloneur de Gibbs). Pour les résultats, nous donnons une carte dessinée par un expert (la vérité terrain, voir Figure 2.21). Les classes $1-6$ correspondent aux régions $B_{3c}, B_{3b}, B_{3d}, a_2, hc$ et 92_a sur la carte. Pour le modèle hiérarchique, nous pouvons remarquer une amélioration par rapport aux autres modèles. Dans le Tableau 2.7, nous donnons les paramètres et le temps d'exécution pour chaque modèle et chaque algorithme.

Finalement, nous présentons une autre image SPOT avec 10 classes (Figure 2.24 – Figure 2.27, Tableau 2.8). La vérité terrain est superposée sur chaque image[1]. Dans le

[1]Les régions sont dessinées par un expert. Malheureusement, elles sont décalées de quelques pixels, ce dont il fait tenir compte pour l'évaluation des résultats.

Tableau 2.2, nous donnons la moyenne (μ) et la variance (σ^2) pour chaque classe. La Figure 2.13 montre l'histogramme de l'image originale. Les résultats sont sensiblement meilleurs pour le modèle hiérarchique que pour les autres modèles.

classe	1	2	3	4	5	6	7	8	9	10
μ	54.61	73.57	159.96	122.84	129.90	146.65	82.56	100.57	93.85	182.34
σ^2	93.10	4.10	31.31	8.90	37.42	15.83	35.58	308.86	93.71	73.18

Tableau 2.2: *Les paramètres de l'image "holland".*

Annexe

2.A Images

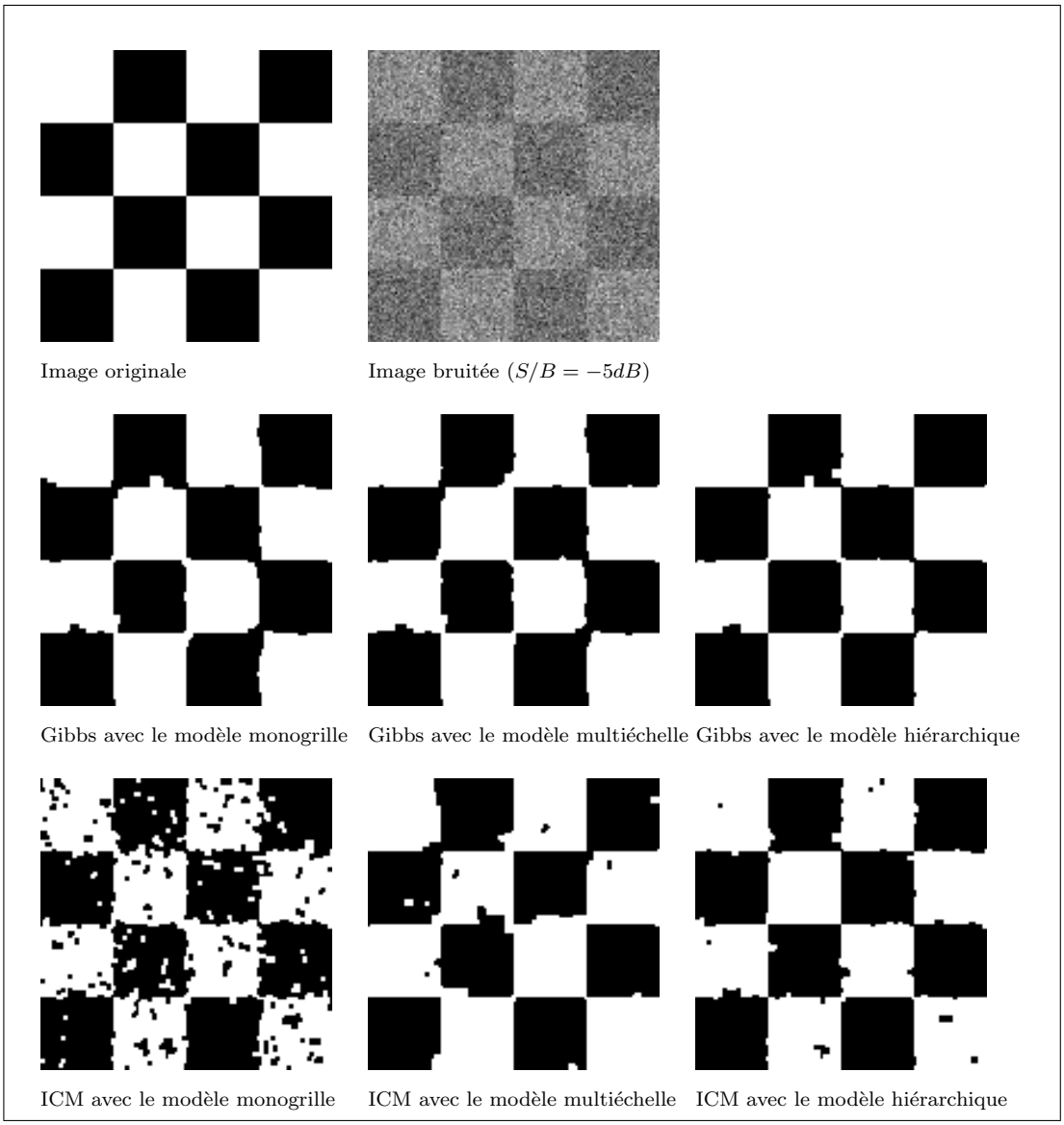

Figure 2.14: *Résultats sur l'image "checkerboard" avec 2 classes.*

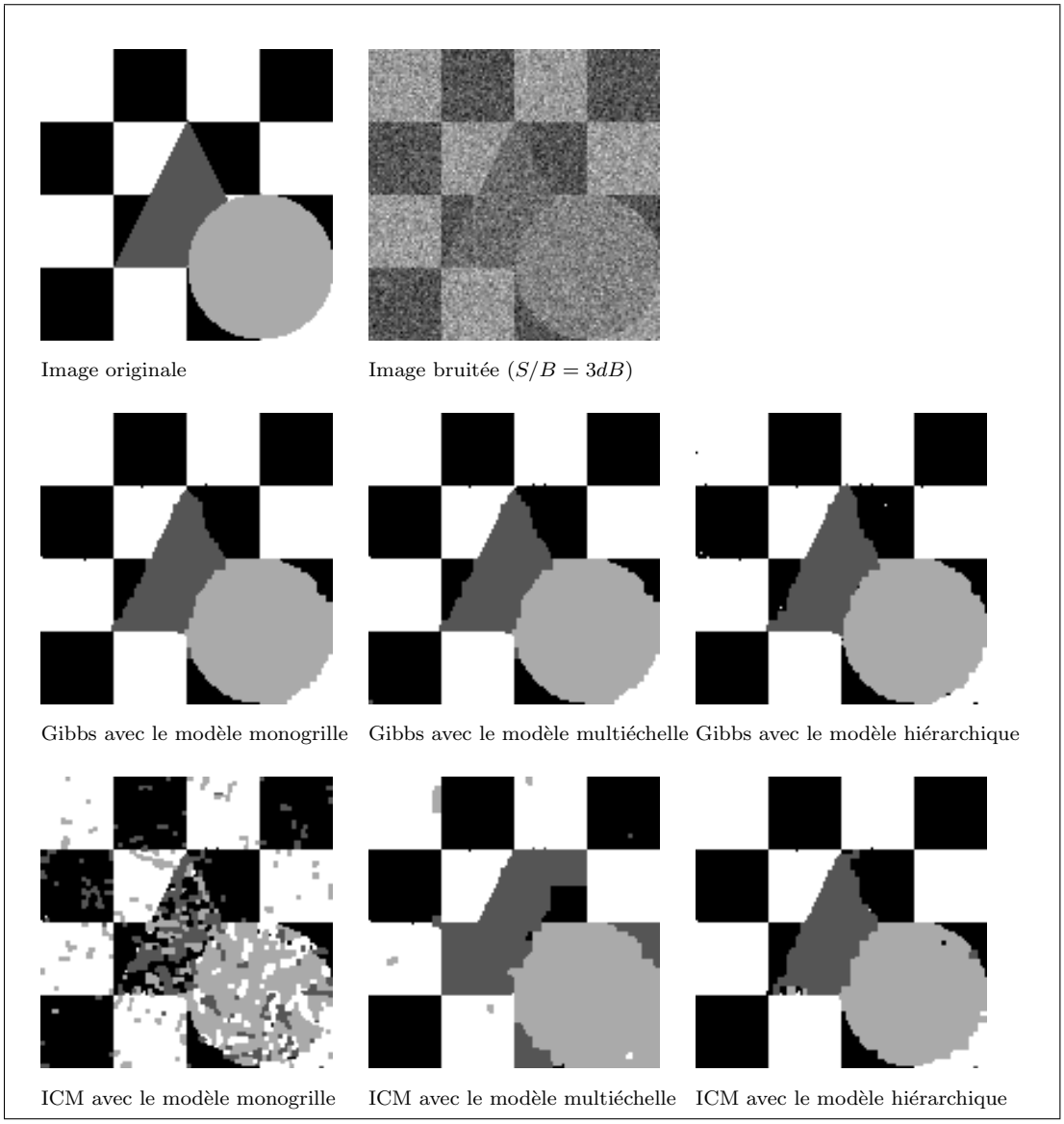

Figure 2.15: *Résultats sur l'image "triangle" avec 4 classes.*

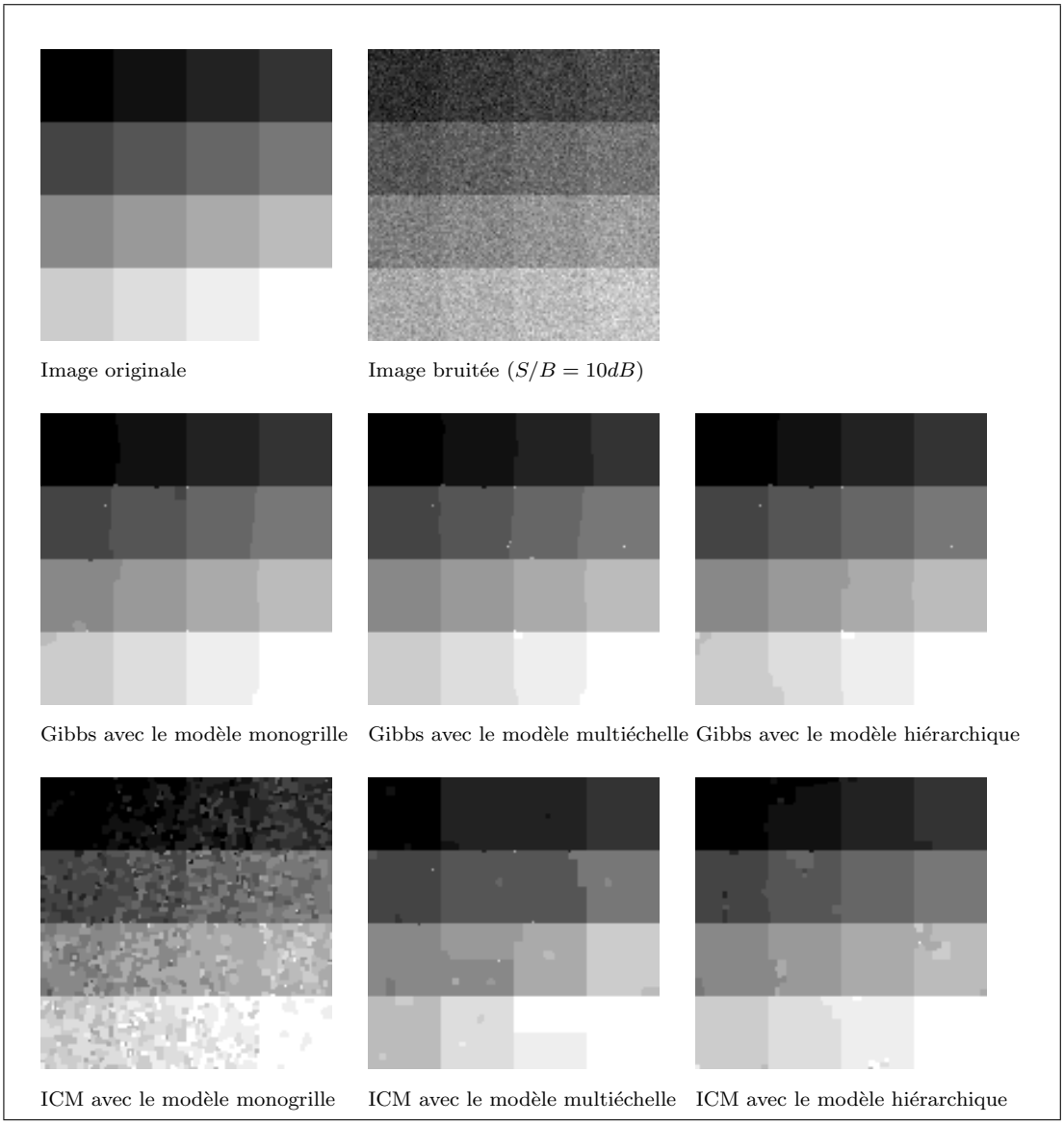

Figure 2.16: *Résultats sur l'image "grey-scale" avec 16 classes.*

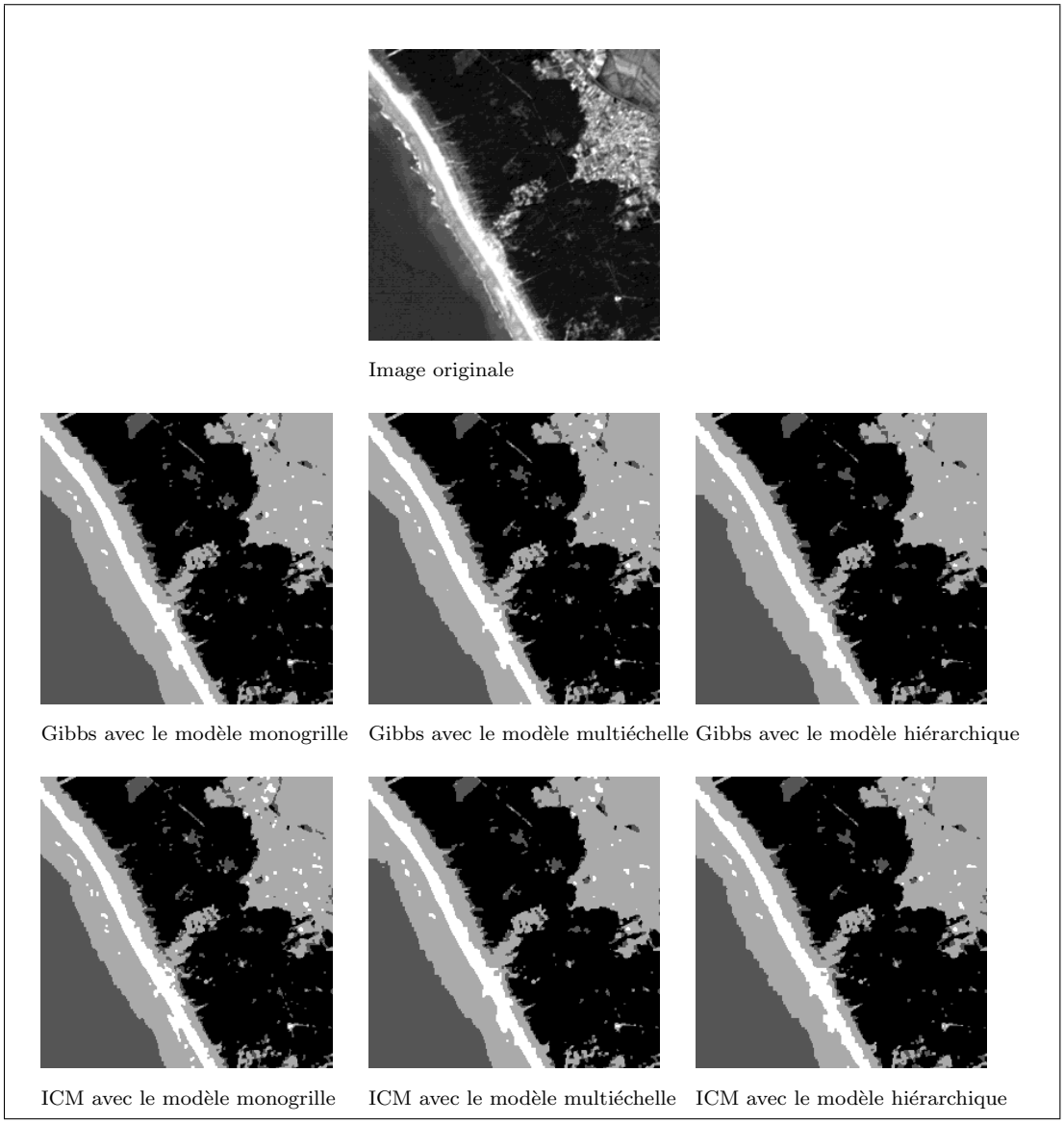

Figure 2.17: *Résultats sur l'image "SPOT" avec 4 classes.*

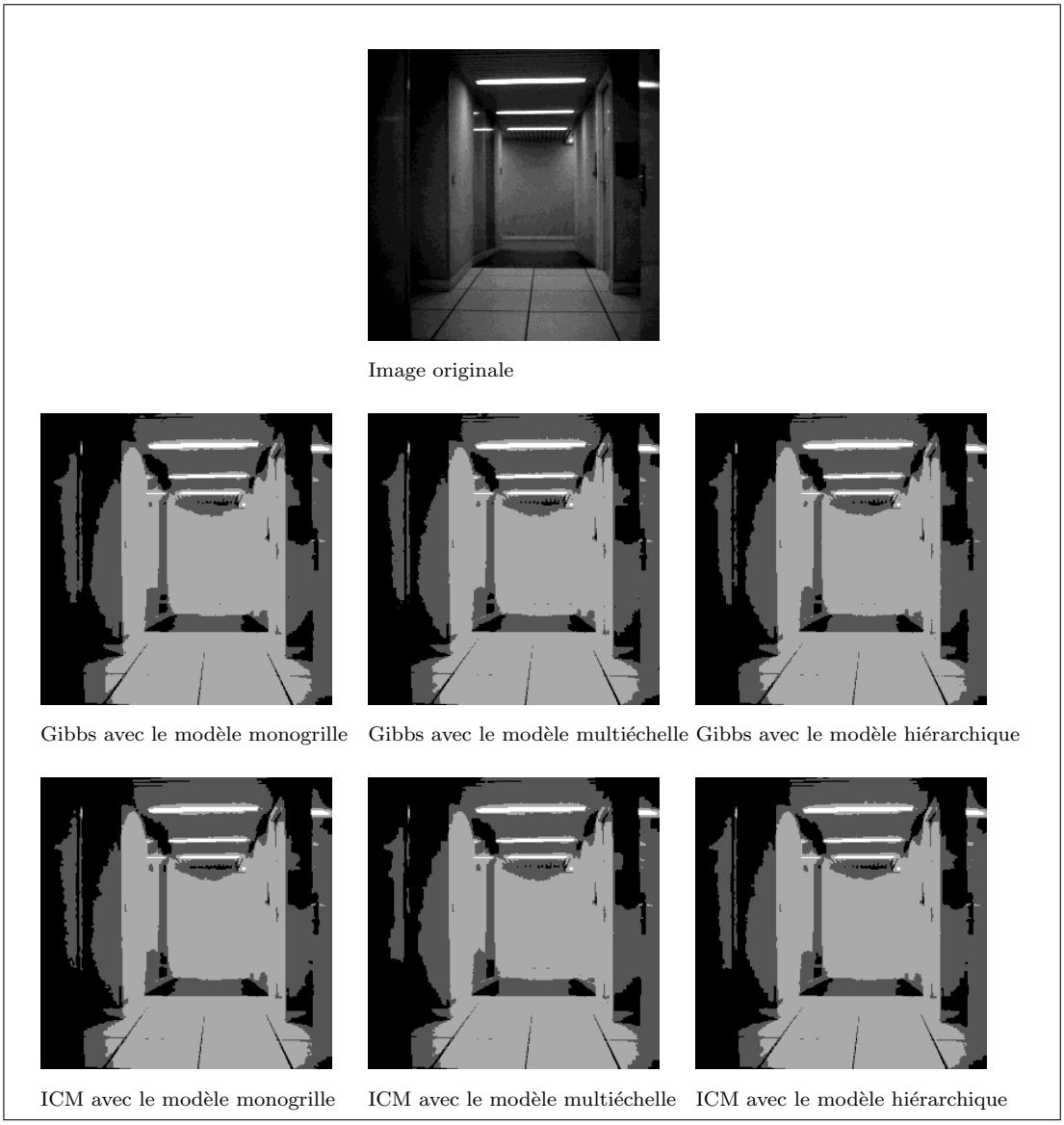

Figure 2.18: *Résultats sur l'image "couloir" avec 4 classes.*

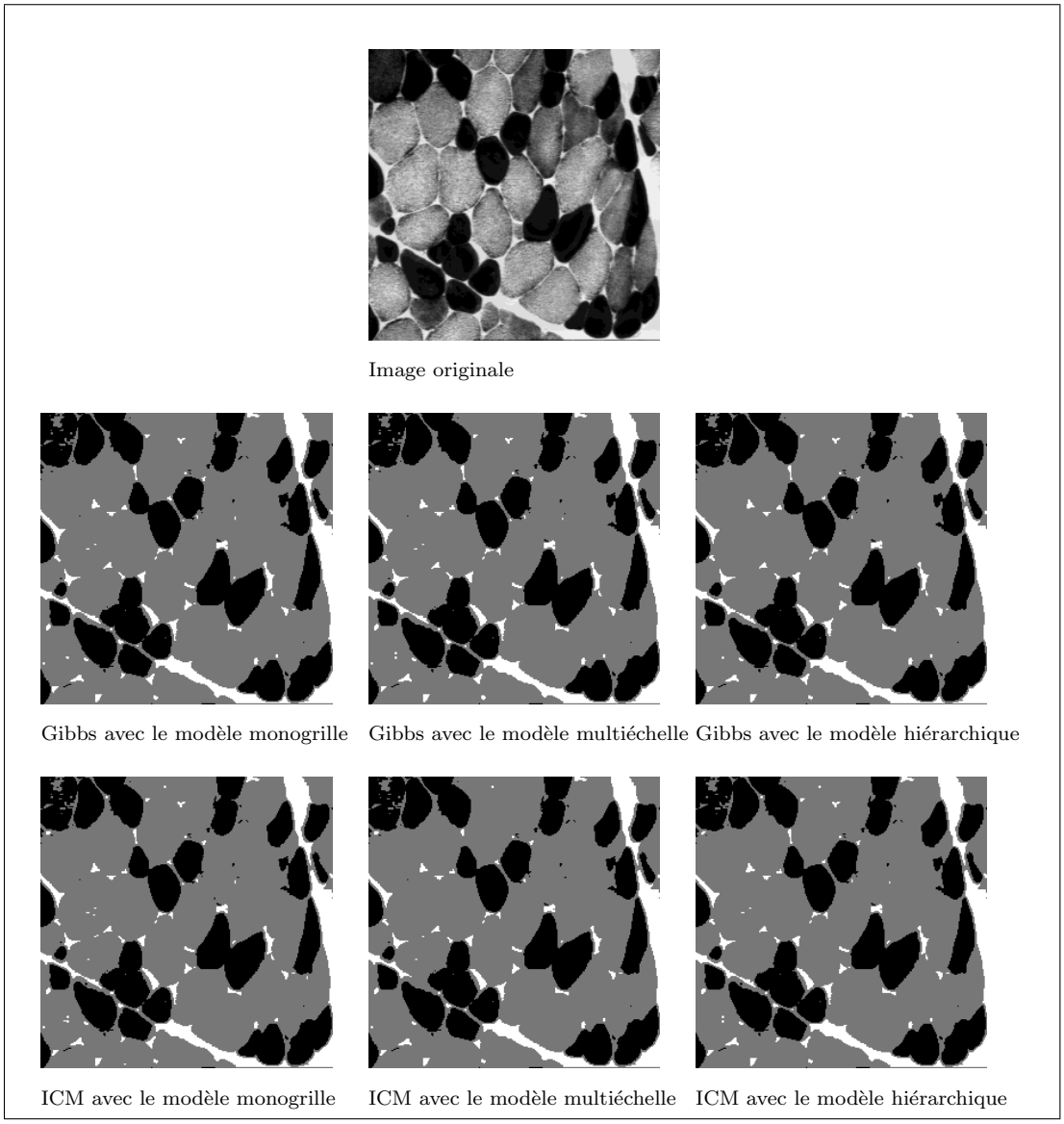

Figure 2.19: *Résultats sur l'image "muscle" avec 3 classes.*

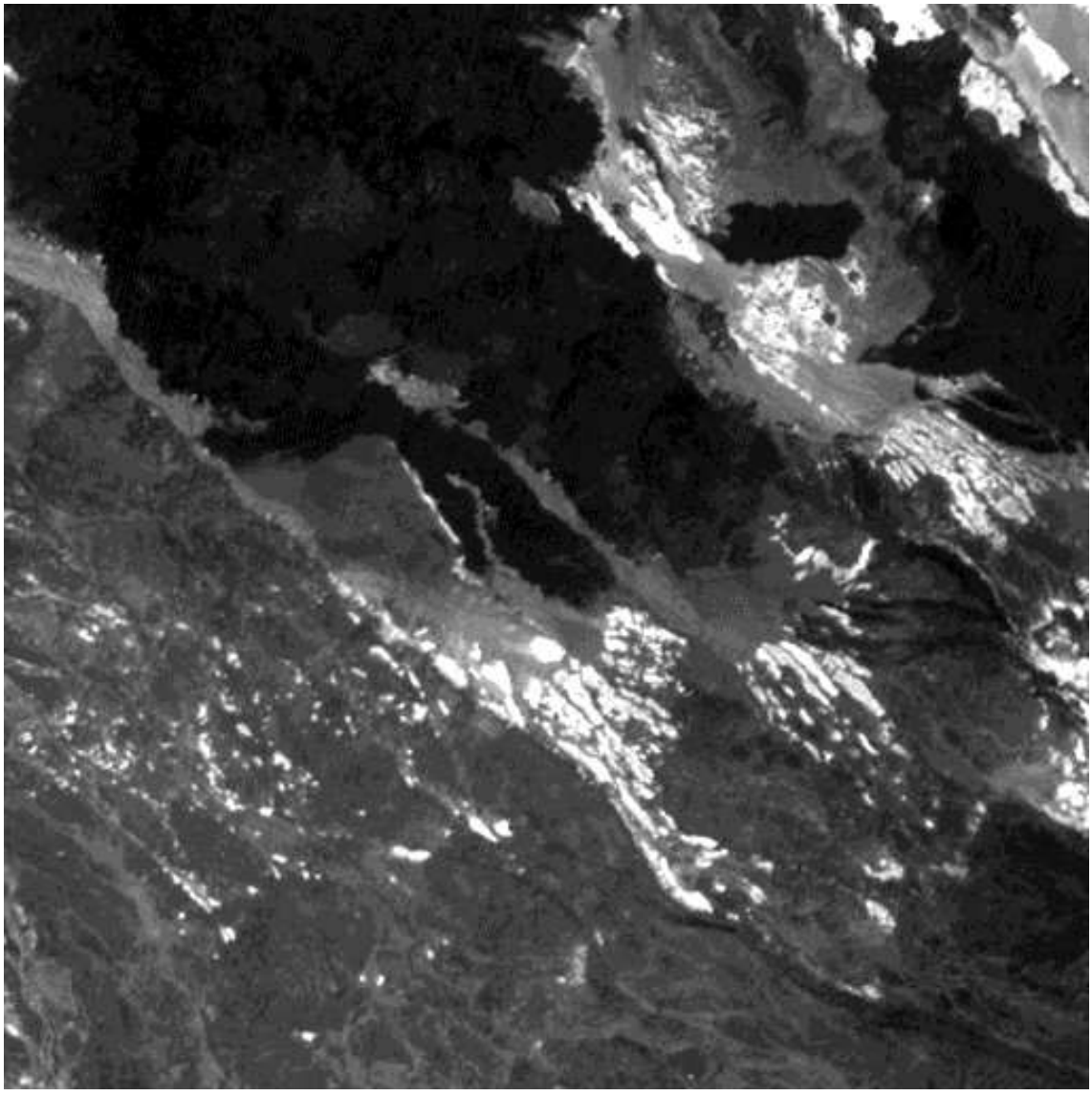

Figure 2.20: *Image originale "assalmer" avec 6 classes.*

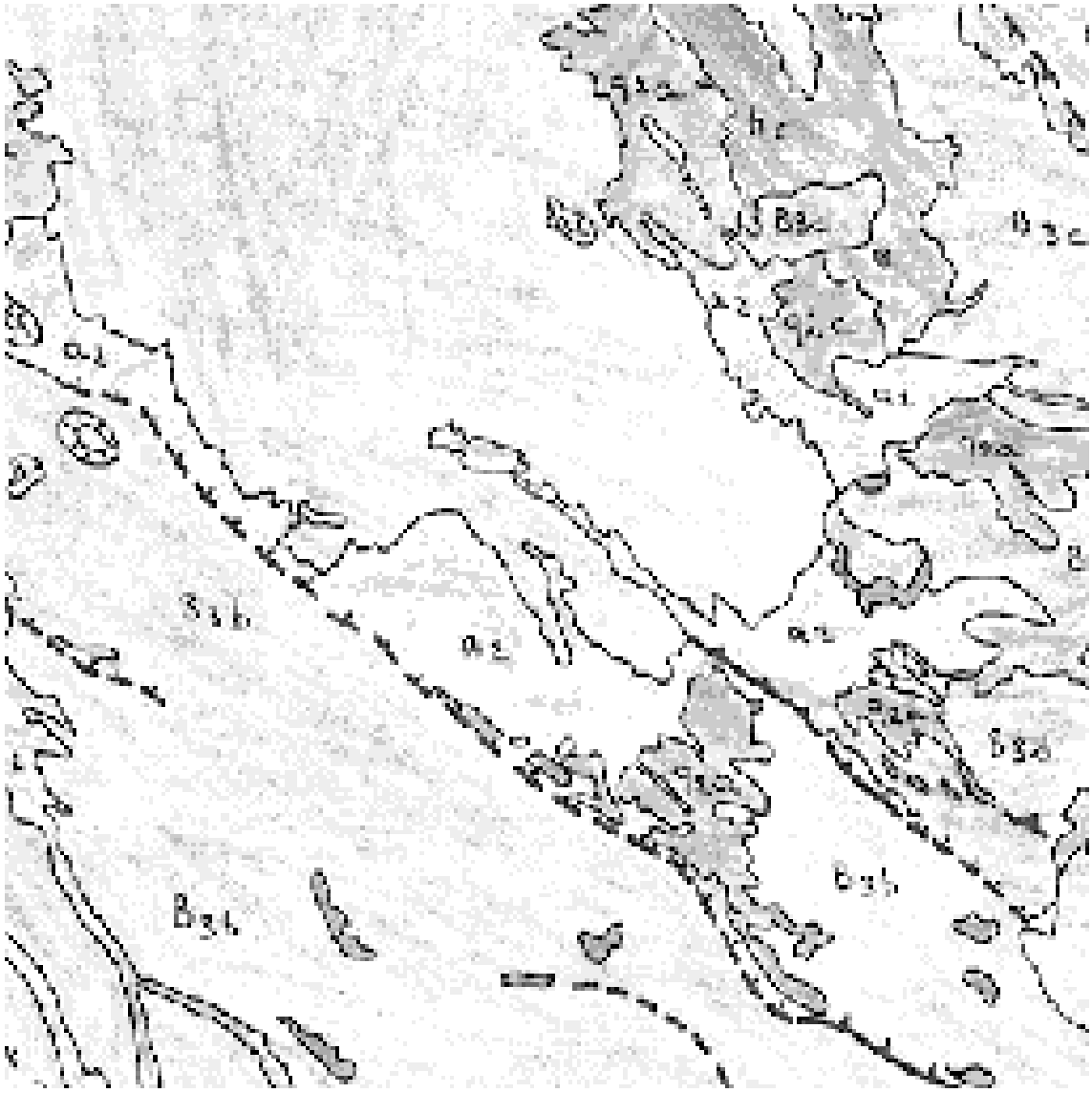

Figure 2.21: *Vérité terrain.*

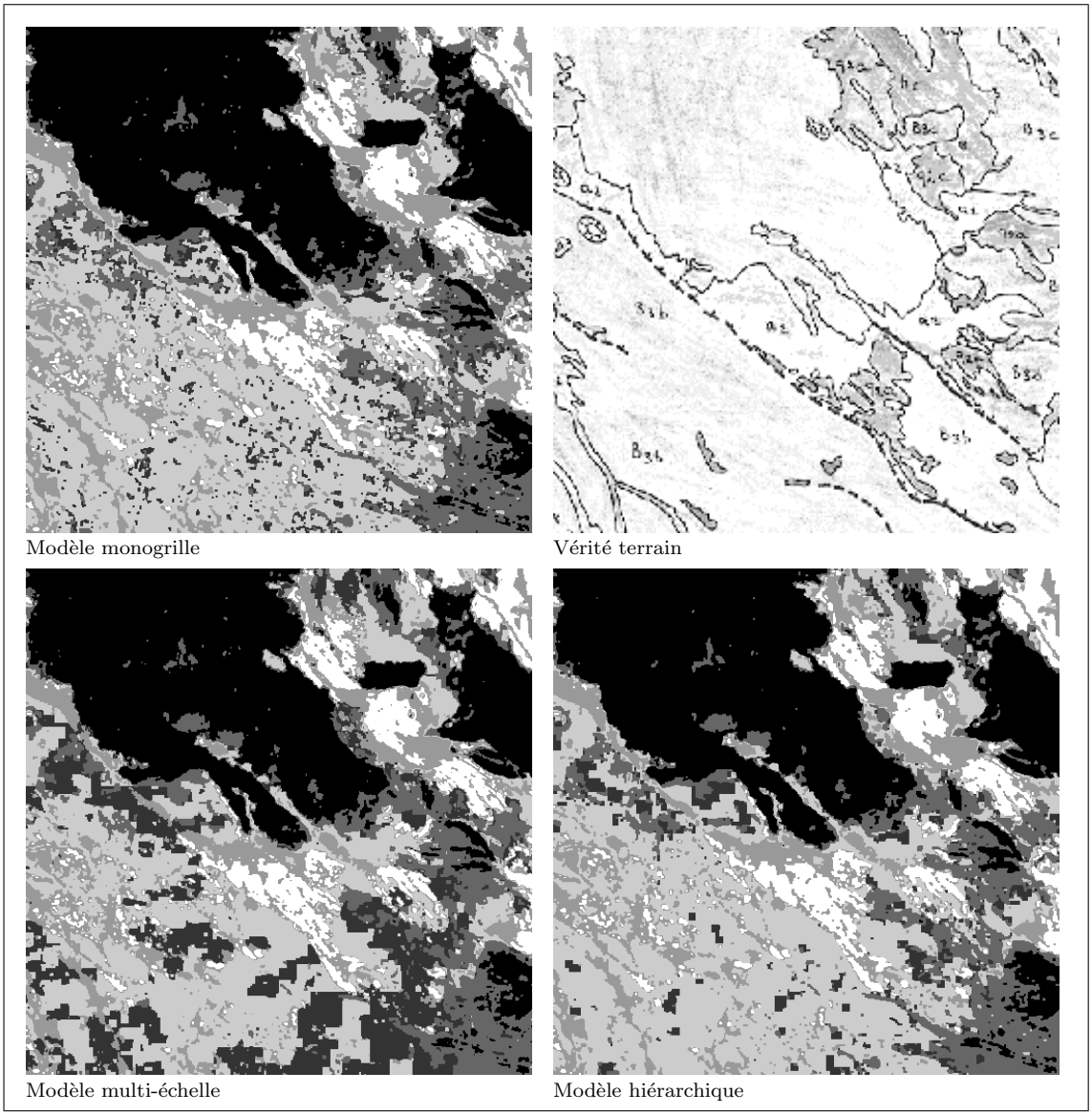

Figure 2.22: *Résultats de l'ICM.*

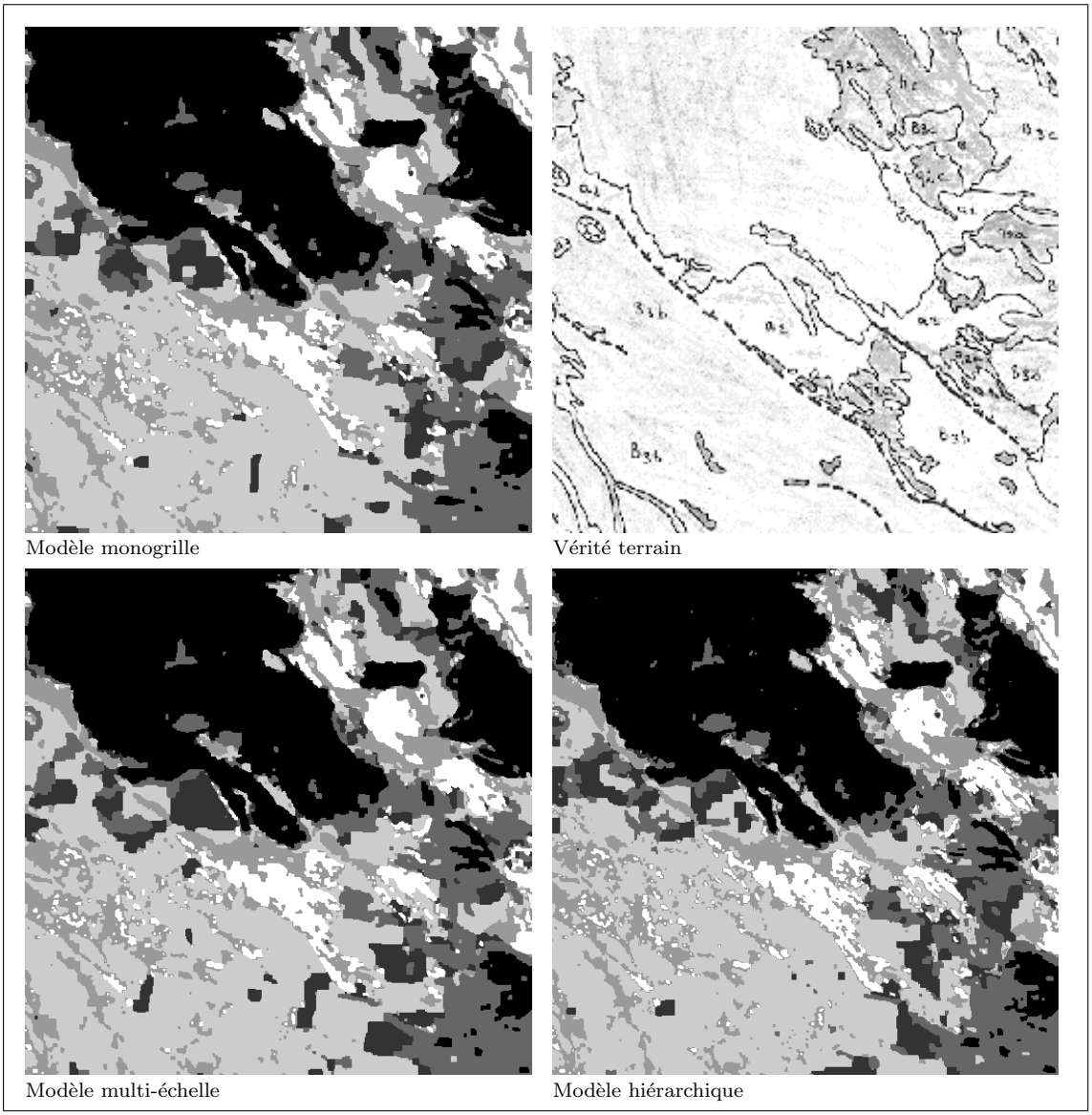

Figure 2.23: *Résultats de l'Echantilloneur de Gibbs.*

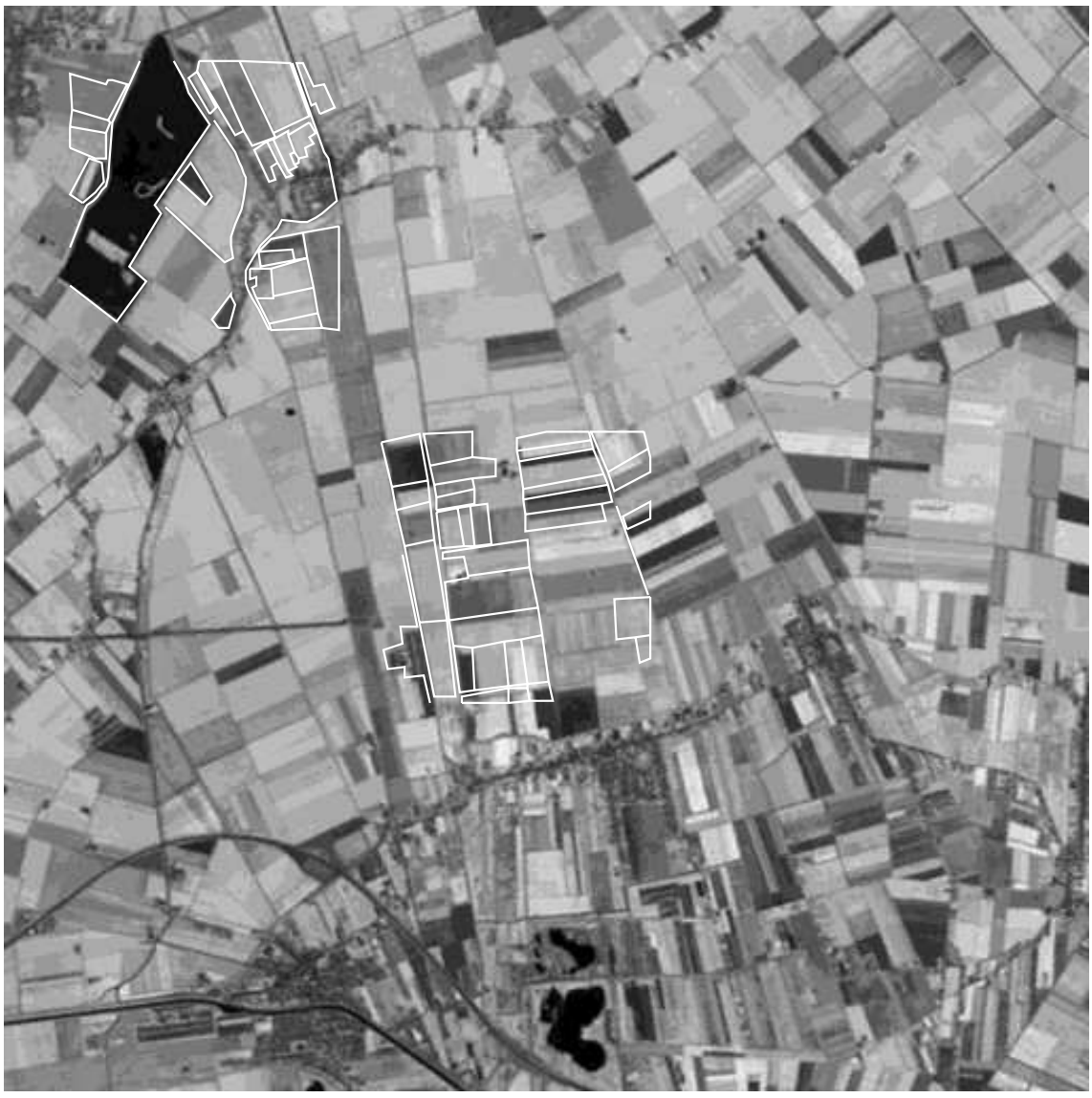

Figure 2.24: *Image originale "holland".*

Figure 2.25: *Résultat de la segmentation monogrille avec 10 classes (ICM).*

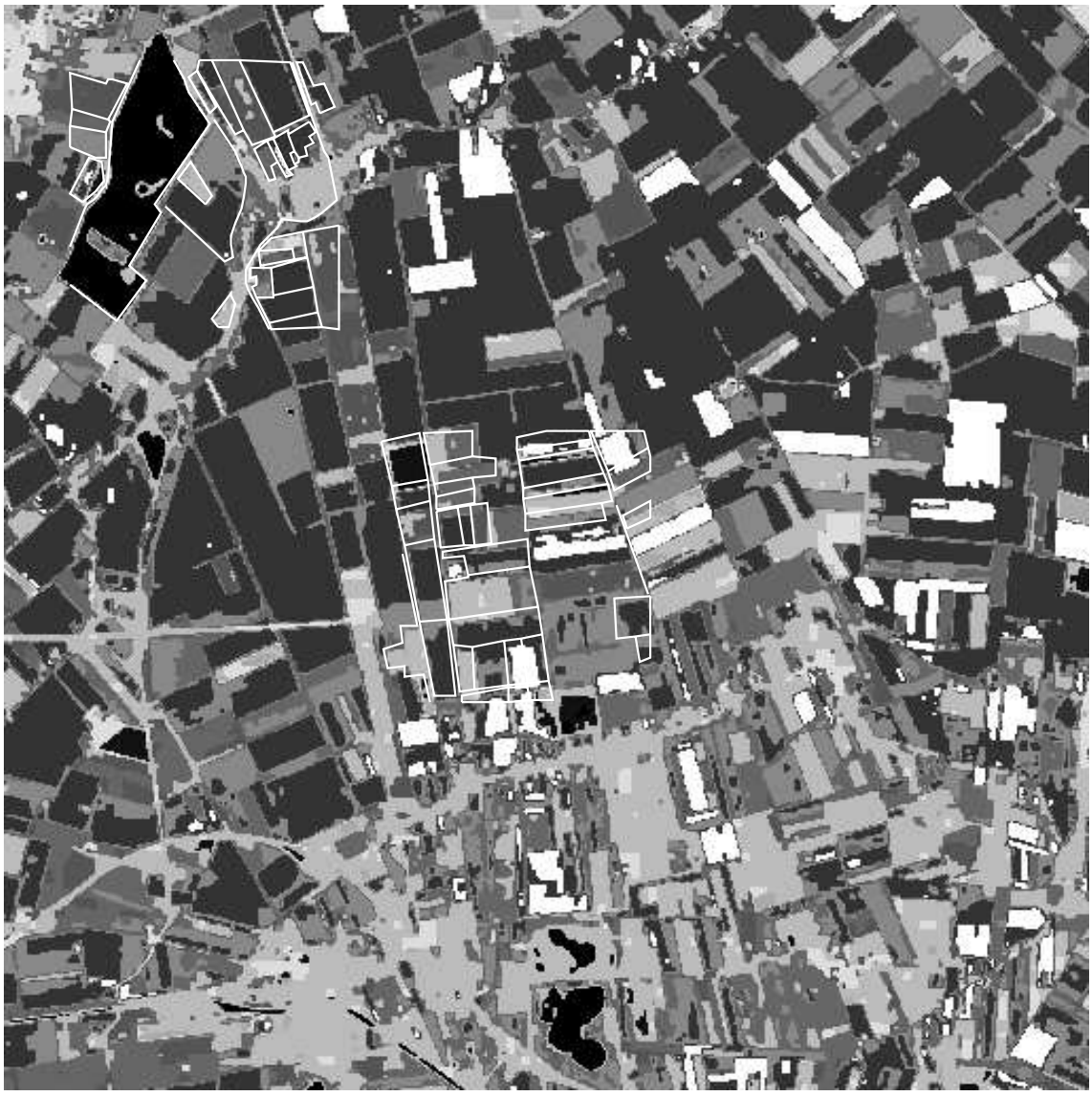

Figure 2.26: *Résultat de la segmentation multiéchelle avec* 10 *classes (ICM).*

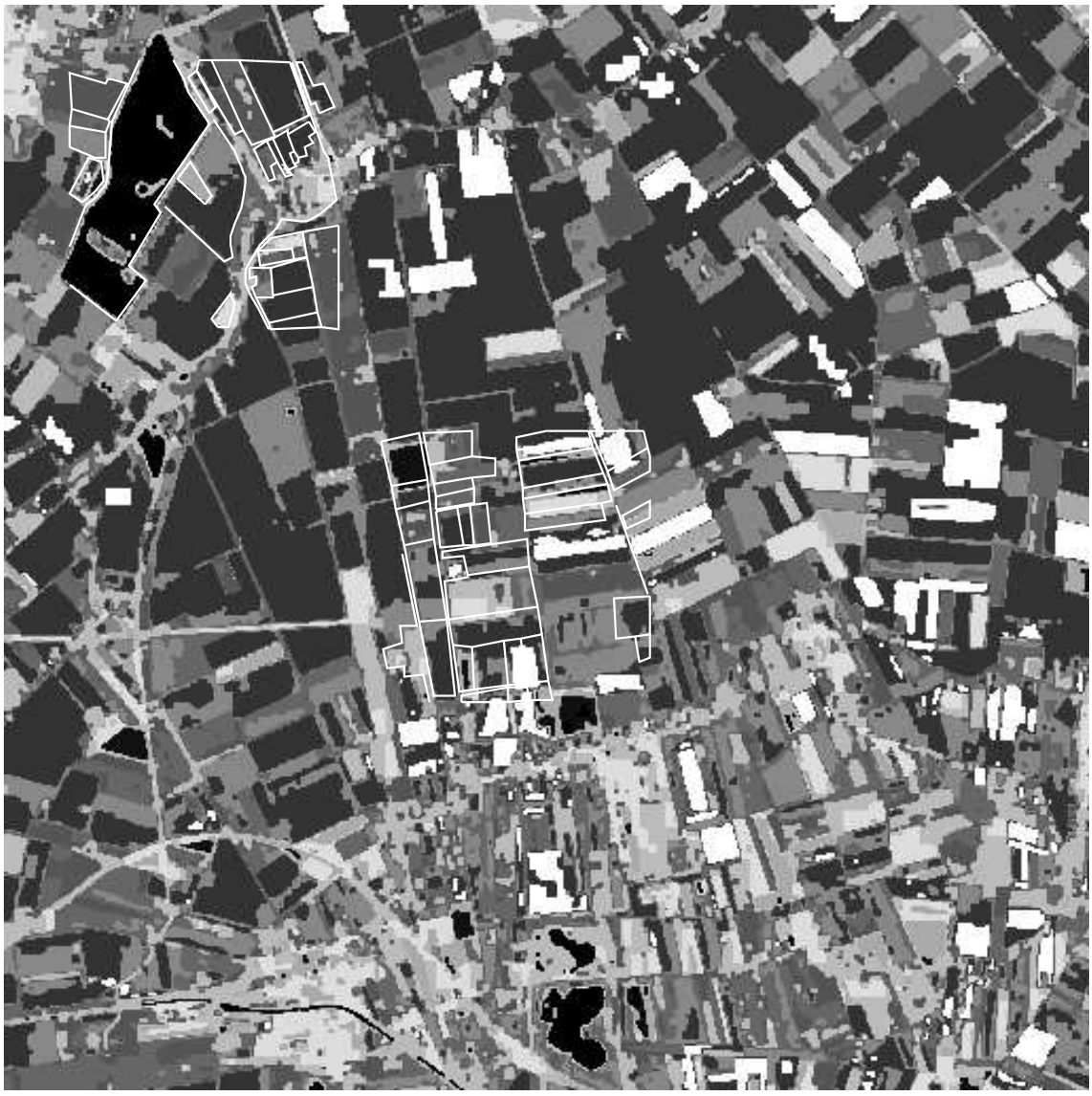

Figure 2.27: *Résultat de la segmentation hiérarchique avec* 10 *classes (ICM).*

2.B Tableaux

monogrille	niveau	VPR	T_0	iter.	temps total	temps/iter.	erreur	β	γ
Gibbs	1	2	4	62	1.91 sec.	0.03 sec.	260 (1.59%)	0.9	—
ICM	1	2	1	8	0.077 sec.	0.009 sec.	1547 (9.44%)	0.9	—
multiéchelle	—	—	—	—	—	—	—	—	—
Gibbs	4	1,2	4	136	3.25sec.	0.02 sec.	236 (1.44%)	0.7	—
ICM	4	1,2	1	18	0.14 sec.	0.008 sec.	465 (2.83%)	0.7	—
hiérarchique	—	—	—	—	—	—	—	—	—
Gibbs	4	4	4,3,2,1	23	50.1 sec.	2.18 sec.	115 (0.7%)	0.7	0.3
ICM	4	4	1	11	16.6 sec.	1.5 sec.	300 (1.83%)	0.7	0.3

Tableau 2.3: *Résultats sur l'image "checkerboard" (*128×128*) avec* 2 *classes.*

monogrille	niveau	VPR	T_0	iter.	temps total	temps/iter.	erreur	β	γ
Gibbs	1	2	4	68	3.01 sec.	0.04 sec.	183 (1.12%)	1.0	—
ICM	1	2	1	9	0.15 sec.	0.02 sec.	2948 (17.99%)	1.0	—
multiéchelle	—	—	—	—	—	—	—	—	—
Gibbs	4	1,2	4	101	3.85sec.	0.04 sec.	176 (1.07%)	1.0	—
ICM	4	1,2	1	17	0.22 sec.	0.01 sec.	1657 (10.11%)	0.9	—
hiérarchique	—	—	—	—	—	—	—	—	—
Gibbs	4	4	4,3,2,1	41	141.97 sec.	3.46 sec.	191 (1.16%)	0.7	0.1
ICM	4	4	1	11	30.17 sec.	2.74 sec.	293 (1.78%)	0.8	0.5

Tableau 2.4: *Résultats sur l'image "triangle" (*128×128*) avec* 4 *classes.*

monogrille	niveau	VPR	T_0	iter.	temps total	temps/iter.	erreur	β	γ
Gibbs	1	2	4	201	22.96 sec.	0.12 sec.	340 (2.08%)	1.0	—
ICM	1	2	1	10	0.55 sec.	0.05 sec.	8721 (53.22%)	1.0	—
multiéchelle	—	—	—	—	—	—	—	—	—
Gibbs	4	1,2	4	337	32.97sec.	0.1 sec.	331 (2.02%)	1.0	—
ICM	4	1,2	1	15	0.68 sec.	0.05 sec.	5198 (31.73%)	1.0	—
hiérarchique	—	—	—	—	—	—	—	—	—
Gibbs	4	4	4,3,2,1	107	1169.46 sec.	10.93 sec.	316 (1.93%)	1.0	0.2
ICM	4	4	1	16	162.87 sec.	10.18 sec.	795 (4.85%)	1.0	0.5

Tableau 2.5: *Résultats sur l'image "grey-scale" (*128×128*) avec* 16 *classes.*

monogrille	niveau	VPR	T_0	iter.	temps total	temps/iter.	β	γ
Gibbs	1	8	4	64	9.06 sec.	0.14 sec.	2.0	—
ICM	1	8	1	7	0.33 sec.	0.047 sec.	2.0	—
multiéchelle	—	—	—	—	—	—	—	—
Gibbs	4	1-8	4	106	10.22 sec.	0.09 sec.	2.0	—
ICM	4	1-8	1	37	1.14 sec.	0.03 sec.	1.0	—
hiérarchique	—	—	—	—	—	—	—	—
Gibbs	4	16	4,3,2,1	29	353.54 sec.	12.19 sec.	2.0	0.4
ICM	4	16	1	6	58.59 sec.	9.76 sec.	0.5	0.6

Tableau 2.6: *Résultats sur l'image "SPOT" (*256×256*) avec* 4 *classes.*

monogrille	niveau	VPR	T_0	iter.	temps total	temps/iter.	β	γ
Gibbs	1	32	4	234	163.18 sec.	0.69 sec.	1.5	—
ICM	1	32	1	8	2.03 sec.	0.25 sec.	1.5	—
multiéchelle	—	—	—	—	—	—	—	—
Gibbs	5	1-32	4	580	180.17 sec.	0.31 sec.	1.5	—
ICM	5	1-32	1	36	5.15 sec.	0.14 sec.	0.3	—
hiérarchique	—	—	—	—	—	—	—	—
Gibbs	5	64	4,3,2,1	154	9629.33 sec.	62.53 sec.	0.7	0.1
ICM	5	64	1	16	915.99 sec.	57.25 sec.	1.0	0.2

Tableau 2.7: *Résultats sur l'image "assalmer" (*512×512*) avec* 6 *classes.*

ICM	niveau	VPR	iter.	temps total	temps/iter.	β	γ
monogrille	1	32	9	3.56 sec.	0.39 sec.	0.5	—
multiéchelle	5	1-32	47	10.14 sec.	0.22 sec.	0.5	—
hiérarchique	5	64	21	1900.83 sec.	90.52 sec.	0.8	0.4

Tableau 2.8: *Résultats sur l'image "holland" (*512×512*) avec* 10 *classes.*

Dans ce chapitre:

3.

Optimisation

Les méthodes bayesiennes associées avec la modélisation markovienne donnent une fonction d'énergie non-convexe qui doit être minimisée pour trouver l'estimateur de champ des étiquettes. Malheureusement, c'est un problème très dur, appelé *optimisation combinatoire*. Par exemple, si nous considérons une image de taille 16×16 avec deux étiquettes possibles, nous obtenons un espace de configurations de 2^{256} éléments. Il est donc impossible de calculer toutes les valeurs possibles de la fonction d'énergie. D'un autre côté, l'utilisation de méthodes classiques n'est pas possible à cause de la non-convexité de la fonction d'énergie.

L'idée de la solution vient de la physique statistique: En 1953, Metropolis *et al.* [67] ont proposé une simulation Monte-Carlo pour trouver les états d'équilibre des systèmes thermodinamiques. Dans les années 80, Černy [17] et Kirkpatrick *et al.* [56] ont montré l'analogie entre la minimisation d'une fonction non-convexe et l'état d'équilibre des systèmes thermodynamiques.

Ils ont substitué la fonction d'énergie du solide à la fonction de coût à minimiser et exécuté l'algorithme de Metropolis en utilisant une séquence de température décroissant lentement. Ils ont appelé ce nouvel algorithme *recuit simulé* [57, 78, 27].

La recherche dans ce domaine est devenue intensive et a aboutis à des contributions variées dont la plus importante est probablement *l'échantillonneur de Gibbs* proposé par Geman et Geman [29]. Bien que les algorithmes de recuit simulé donnent un optimum global, ils exigent beaucoup de calcul. Pour éviter cet inconvénient, deux solutions ont été proposées: la parallélisation d'algorithmes de type recuit [3], et l'utilisation d'algorithmes *déterministes*, qui sont sous-optimaux mais convergent avec un nombre faible d'itérations [8, 53].

3.1 Recuit simulé

Soit $\omega, \eta, \ldots$ les configurations d'un problème d'optimisation combinatoirel (ils correspondent aux états d'un solide) et soit $U(\omega)$ le coût (ou énergie) de la configuration ω. Les éléments d'une configurations sont indexés par $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ et l'espace d'états commun est noté par $\Lambda = \{0, 1, \ldots, L-1\}$. L'ensemble de toutes les configurations possibles est noté par Ω. Puisque $\forall s \in \mathcal{S}: \omega_s \in \Lambda$, on a $\Omega = \Lambda^N$.

Algorithme 3.1.1 (Recuit simulé)

① *Soit $k = 0$, initialiser ω aléatoirement et choisir une température initiale $T = T_0$ assez grande.*

② *Construire une perturbation η à partir de la configuration courante ω telle que η est différente de ω en un seul élément.*

③ **(Critère de Metropolis)** *Calculer $\Delta U = U(\eta) - U(\omega)$ et accepter η si $\Delta U < 0$ ou avec une probabilité de $\exp(-\Delta U/T)$ si $\Delta U \geq 0$ (analogie avec la thermodynamique):*

$$\omega = \begin{cases} \eta & \text{si } \Delta U \leq 0, \\ \eta & \text{si } \Delta U > 0 \text{ et } \xi < \exp(-\Delta U/T), \\ \omega & \text{sinon} \end{cases} \tag{3.1}$$

où ξ est un nombre aléatoire uniforme dans $[0, 1)$.

④ *Continuer avec Étape ② jusqu'à l'obtention de l'équilibre.*

⑤ *Décroitre la température: $T = T_{k+1}$ et continuer avec Étape ② en utilisant $k = k + 1$ jusqu'à la congélation du système.*

Cet algorithme est connu sous le nom de *recuit homogène*, car il est décrit par une séquence des chaînes de Markov homogènes. Si la température décroît après chaque transition, l'algorithme est décrit par un chaîne de Markov inhomogène et il est appelé *recuit inhomogène*. Ce type de recuit est le plus souvent utilisé en pratique. Nous pouvons obtenir un tel algorithme si nous supprimons l'Étape ④ de l'Algorithme 3.1.1.

3.1.1 Modèle mathématique

Le modèle mathématique du recuit simulé a été étudié en détaille par Aarts and van Laarhoven [57]. Nous présentons ici ce modèle:

Le recuit simulé génère une séquence de configurations qui constituent un chaîne de Markov avec une probabilité de transition $P_{\omega,\eta}(k-1,k)$. De plus, soit $X(k)$ l'état atteint après la $k^{\text{ième}}$ transition. La probabilité de cet événement est donnée par:

$$P(X(k)=\omega)=\sum_{\zeta} P(X(k-1)=\zeta)P_{\zeta,\omega}(k-1,k) \quad k=1,2,\ldots \tag{3.2}$$

Si la probabilité de transition $P_{\omega,\eta}(k-1,k)$ ne dépend pas de k, la chaîne correspondante est homogène, sinon elle est inhomogène. Les probabilités de transition dépendent aussi de la température T. Donc, si T est constant, la chaîne est homogène et la matrice de transition $P=P(T)$ peut être écrite comme:

$$P_{\omega,\eta}(T)=\begin{cases} G_{\omega,\eta}(T)A_{\omega,\eta}(T) & \forall \eta \neq \omega \\ 1-\sum_{\zeta} G_{\omega,\zeta}(T)A_{\omega,\zeta}(T) & \eta=\omega \end{cases} \tag{3.3}$$

où $G_{\omega,\eta}(T)$ est la *probabilité de générer* η à partir de ω et $A_{\omega,\eta}(T)$ est la *probabilité d'acceptation* de la configuration η. Il est clair que $P(T)$ est une matrice stochastique (voir Équation (3.3)):

$$\forall \omega : \sum_{\zeta} P_{\omega,\zeta}(T)=1 \tag{3.4}$$

Dans l'Algorithme 3.1.1, $G_{\omega,\eta}(T)$ est une distribution uniforme sur les configurations η qui sont diffèrents de ω en un seul composent. $A(T)$ est donné par le critère de Metropolis:

$$A_{\omega,\eta}(T)=\min(1,\exp(-(U(\eta)-U(\omega))/T)) \tag{3.5}$$

où $U(\omega)$ est la fonction d'énergie.

3.1.1.1 Loi de température

Il est connu [57, 29, 34] que le recuit simulé converge avec une probabilité de un vers une optimum global si la loi de température T_k est moins rapide que $C/\ln(k)$ pour une certain constante C qui est indépendante de k.

En pratique, la loi théorique est approchée par une loi exponentielle car elle est trop lente. A cause de cet approximation, la convergence vers un optimum global n'est plus garantie.

Température initiale La température initiale T_0 doit être choisie telle que toutes les transitions peuvent être acceptées avec une probabilité non nulle. Il est très difficile de trouver une telle valeur initiale car elle est reliée au minimum et maximum d'énergie [29]. En pratique, on choisi une T_0 relativement faible pour assurer une convergence rapide. Dans [29] par exemple, $T_0=4$ a été suggéré et nous avons utilisé cette valeur dans les tests expérimentaux présentés.

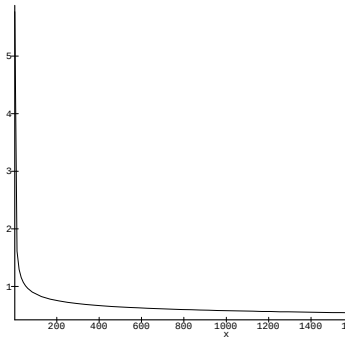

Figure 3.1: *Loi de température logarithmique (*$4/\ln(k)$*).*

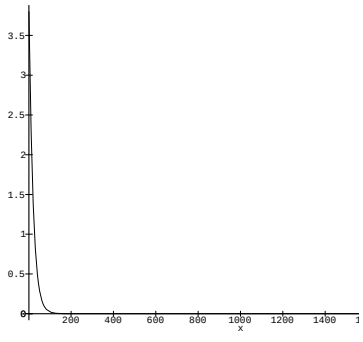

Figure 3.2: *Loi de température exponentielle (*$0.95^k \cdot 4$*).*

Température finale Évidemment, $lim_{k\to\infty} T_k = 0$ peut être seulement approximée par un nombre fini de valeurs T_k. Donc, nous avons besoin d'un critère d'arrêt. Le critère le plus souvent utilisé en pratique est d'arrêter l'exécution après un certain nombre d'itérations, ou bien si ΔU est inférieure à un certain seuil.

Loi de température Le point le plus important est la règle de décroissance de la température. Les lois logarithmiques (cf. Figure 3.1) sont en général trop lent. Pour cette raison, nous utilisons plutôt des lois exponentielles (cf. Figure 3.2):

$$T_{k+1} = c \cdot T_k, \qquad k = 0, 1, 2, \ldots \tag{3.6}$$

où $c < 1$ est un constant proche de 1. Cette règle populaire a été proposée par [56] (nous avons utilisé la même loi dans nos expériences).

3.1.1.2 Echantillonneur de Gibbs

Une matrice d'acceptation plus élaborée a été proposée par Geman et Geman [29]. Cette règle jointe à un recuit inhomogène est devenue l'algorithme le plus populaire appelé *Echantillonneur de Gibbs*:

Algorithme 3.1.2 (Echantillonneur de Gibbs)

① *Soit $k = 0$, choisir une configuration initiale ω quelconque et soit $T = T_0$ une température initiale assez élevée.*

② *Pour chaque configuration différent de la configuration courante ω par un élément au minimum (l'ensemble de telles configurations est noté par $\mathcal{N}_\omega$), calculer l'énergie $U(\eta)$ ($\eta \in \mathcal{N}_\omega$).*

③ **(Echantillonneur de Gibbs)** *Un échantillonage de $\mathcal{N}_\omega$ est effectué tel que η est accepté avec une probabilité de:*

$$\frac{\exp(-U(\eta))}{\sum_{\zeta \in \mathcal{N}_\omega} \exp(-U(\zeta))} \tag{3.7}$$

④ *Décroitre la température: $T = T_{k+1}$ et continuer avec l'Étape ② en utilisant $k = k + 1$ jusqu'à la congélation du système.*

Remarquons que la matrice de génération est donnée par $G_{\omega,\eta} = 1$ si $\eta \in \mathcal{N}_\omega$, 0 sinon.

3.2 Recuit multi-température

Nous proposons ici une nouvelle méthode de recuit que nous appellons le *recuit multi-température* [47, 50, 85]:

Soit $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ l'ensemble des sites, $\mathcal{V}$ un système de voisinage avec les cliques $\mathcal{C}$ et soit X un champ de Markov avec une fonction d'énergie U. Nous définissons un schéma de recuit où la température T dépend des itérations k et des cliques C. $\oslash$ désigne l'opérateur suivant:

$$P(X = \omega) = \pi_{T(k,C)}(\omega) = \frac{\exp(-U(\omega) \oslash T(k,C))}{Z} \tag{3.8}$$

$$\text{où } U(\omega) \oslash T(k,C) = \sum_{C \in \mathcal{C}} \frac{V_C(\omega)}{T(k,C)} \, . \tag{3.9}$$

Supposons que les sites soient mis à jour dans l'ordre $\{n_1, n_2, \ldots\} \subset \mathcal{S}$. Le processus stochastique correspondant est noté par $\{X(k), k = 0, 1, 2, \ldots\}$, où $X(0)$ est la configuration initiale. $X(k)$ est une chaîne de Markov dont la matrice de transition est:

$$P_{\omega,\eta}(k-1,k) = \begin{cases} G_{\omega,\eta}(T(k,C))A_{\omega,\eta}(T(k,C)) & \forall \eta \neq \omega \\ 1 - \sum_{\zeta \neq \omega} G_{\omega,\zeta}(T(k,C))A_{\omega,\zeta}(T(k,C)) & \eta = \omega \end{cases} \tag{3.10}$$

En considérant l'échantillonneur de Gibbs, la matrice de génération $G_{\omega,\eta}(T(k,C))$ et la matrice d'acceptation $A_{\omega,\eta}(T(k,C))$ sont:

$$G_{\omega,\eta}(T(k,C)) = G_{\omega,\eta}(k) = \begin{cases} 1, & \text{si } \eta = \omega|_{\omega_{n_k}=\lambda} \forall \lambda \in \Lambda \\ 0, & \text{sinon} \end{cases} \tag{3.11}$$

$$A_{\omega,\eta}(T(k,C)) = \pi_{T(k,C)}(X_{n_k} = \omega_{n_k} \mid X_s = \omega_s, s \neq n_k) \tag{3.12}$$

Notons que l'acceptation est gouvernée par les caractéristiques *locales*. $\pi_{T(k,C)}(X_{n_k} = \omega_{n_k} \mid X_s = \omega_s, s \neq n_k)$ est un peu différent de $\pi_{T(k,C)}(\omega)$ défini dans l'Équation (3.8):

$$\pi_{T(k,C)}(X_s = \omega_s \mid X_r = \omega_r, s \neq r) = \frac{1}{Z_s} \exp\left(- \sum_{C \in \mathcal{C}: s \in C} \frac{V_C(\omega)}{T(k,C)} \right) \tag{3.13}$$

$$\text{avec } Z_s = \sum_{\lambda \in \Lambda} \exp\left(- \sum_{C \in \mathcal{C}: s \in C} \frac{V_C(\omega|_{\omega_s=\lambda})}{T(k,C)} \right) \tag{3.14}$$

La matrice de transition en k est donc:

$$P_{\omega,\eta}(k) = \begin{cases} \pi_{T(k,C)}(X_{n_k} = \eta_{n_k} \mid X_s = \eta_s, s \neq n_k), & \text{si } \eta = \omega|_{\omega_{n_k}=\lambda} \forall \lambda \in \Lambda \\ 0, & \text{sinon} \end{cases} \tag{3.15}$$

Soit Ω_{opt} l'ensemble des configurations globalement optimales:

$$\Omega_{opt} = \{\omega \in \Omega : U(\omega) = \min_{\eta \in \Omega} U(\eta)\} \tag{3.16}$$

Soit π_0 la distribution uniforme sur Ω_{opt}, définissons:

$$U^{sup} = \max_{\omega \in \Omega} U(\omega), \tag{3.17}$$

$$U^{inf} = \min_{\omega \in \Omega} U(\omega), \tag{3.18}$$

$$\text{et } \Delta = U^{sup} - U^{inf}. \tag{3.19}$$

Examinons la décomposition de $U(\omega) \oslash T(k,C)$ définie dans l'Équation (3.9). Soit $\omega' \in \Omega_{opt}$ une configuration globalement optimale, qui nous donne $U(\omega') - U^{inf} = 0$. Dans le cas du recuit classique, la division par une température constante ne change pas cette relation (évidemment, $\forall k$: $(U(\omega') - U^{inf})/T_k$ reste 0). Mais il n'est pas nécessairement vrai que $(U(\omega') - U^{inf}) \oslash T(k,C)$ est aussi 0 car si nous choisissons des températures suffisement faibles pour les cliques où ω'_C est *localement* sous-optimale (c.a.d. en renforçant les cliques sous-optimales) et des températures suffisement grandes pour les cliques où ω'_C est *localement* optimale (c.a.d. en affaiblissant les cliques optimales), nous obtenons $(U(\omega') - U^{inf}) \oslash T(k,C) > 0$, qui veut dire que ω' n'est plus optimale *globalement*.

Nous devons donc imposer des conditions nouvelles sur la loi de température pour assurer la convergence vers une configuration optimale. Tout d'abord, examinons la décomposition de $U(\omega) - U(\eta)$ sur les cliques pour ω et η quelconques, $\omega \neq \eta$:

$$U(\omega) - U(\eta) = \sum_{C \in \mathcal{C}} (V_C(\omega) - V_C(\eta)). \tag{3.20}$$

En effet, il y a des membres positifs et négatifs dans cette décomposition, nous pouvons donc écrire:

$$\sum_{C \in \mathcal{C}} (V_C(\omega) - V_C(\eta))$$

$$= \underbrace{\sum_{C \in \mathcal{C}:(V_C(\omega) - V_C(\eta)) < 0} (V_C(\omega) - V_C(\eta))}_{\Sigma^-(\omega,\eta)} + \underbrace{\sum_{C \in \mathcal{C}:(V_C(\omega) - V_C(\eta)) \geq 0} (V_C(\omega) - V_C(\eta))}_{\Sigma^+(\omega,\eta)}. \tag{3.21}$$

Maintenant, examinons Δ défini dans l'Équation (3.19). Si nous voulons décomposer Δ, Nous devons choisir une configuration ω' qui a une énergie maximale (c.a.d. $U(\omega') = U^{sup}$) et une autre configuration ω'' qui a une énergie minimale (c.a.d. $U(\omega'') = U^{inf}$). Évidemment, il existe plusieurs décompositions possibles qui dépendent du nombre des configurations optimales ($\mid \Omega_{opt} \mid$) et du nombre des configurations avec une énergie maximale ($\mid \Omega_{sup} \mid$). La décomposition de Δ a donc, pour une paire (ω', ω'') donnée, la forme suivante:

$$\Delta = \Sigma^-(\omega', \omega'') + \Sigma^+(\omega', \omega'') \tag{3.22}$$

De plus, définissons Σ^+_Δ:

$$\Sigma^+_\Delta = \min_{\substack{\omega' \in \Omega_{sup} \\ \omega'' \in \Omega_{opt}}} \Sigma^+(\omega', \omega''). \tag{3.23}$$

Évidemment, $\Delta \leq \Sigma^+_\Delta$.

Théorème 3.2.1 (Recuit multi-température) *Supposons qu'il existe un entier $\kappa \geq N$ tel que pour chaque $k = 0, 1, 2, \ldots$, $\mathcal{S} \subseteq \{n_{k+1}, n_{k+2}, \ldots, n_{k+\kappa}\}$. Pour tous les $C \in \mathcal{C}$, soit $T(k, C)$ une séquence décroissante des températures en k telle que:*

(i) $\forall C: \lim_{k\to\infty} T(k, C) = 0$.
Notons respectivement par T_k^{inf} et T_k^{sup} le maximum et le minimum de la fonction de température pour k donné ($\forall C \in \mathcal{C}: T_k^{inf} \leq T(k, C) \leq T_k^{sup}$).

(ii) Pour tous les $k \geq k_0$ et pour un entier $k_0 \geq 2$ quelconque: $T_k^{inf} \geq N\Sigma_\Delta^+ / \ln(k)$.

(iii) Si $\Sigma^-(\omega, \omega') \neq 0$ pour une $\omega \in \Omega \setminus \Omega_{opt}$ quelconque et une $\omega' \in \Omega_{opt}$ quelconque, nous devons imposer une condition supplémentaire:
$\forall k: \frac{T_k^{sup} - T_k^{inf}}{T_k^{inf}} \leq R$, avec:

$$R = \min_{\substack{\omega \in \Omega \setminus \Omega_{opt} \\ \omega' \in \Omega_{opt} \\ \Sigma^-(\omega,\omega') \neq 0}} \frac{U(\omega) - U^{inf}}{\mid \Sigma^-(\omega, \omega') \mid}. \tag{3.24}$$

Si les conditions sont satisfaites alors pour n'importe quelle configuration initiale $\eta \in \Omega$ et pour chaque $\omega \in \Omega$:

$$\lim_{k\to\infty} P(X(k) = \omega \mid X(0) = \eta) = \pi_0(\omega). \tag{3.25}$$

Le preuve du théorème est donné en Annexe 3.A.

Remarques:

1 En pratique, nous pouvons rarement déterminer R et Σ_Δ^+.

2 En examinant Σ_Δ^+ dans la condition 3.2.1/ii, nous avons le même problème que dans le cas d'un recuit classique. La seule différence est qu'ici nous utilisons Σ_Δ^+ au lieu de Δ. En conséquence, la même solution pratique peut être utilisée: une loi de température exponentielle et une température initiale assez élevée.

3 Le facteur R est plus intéressant. Nous proposons ici deux méthodes: Soit nous utilisons un intervalle suffisament petit $[T_0^{inf}, T_0^{sup}]$ qui satisfait raisonablement la condition 3.2.1/iii (nous avons utilisé cette méthode), soit nous utilisons une condi-

tion plus stricte mais facilement vérifiable [85] au lieu de la condition 3.2.1/iii:

$$\lim_{k\to\infty} \frac{T_k^{sup} - T_k^{inf}}{T_k^{inf}} = 0. \tag{3.26}$$

4 Que se passe-t-il si $\Sigma^-(\omega, \omega') = 0$ quelques soient ω et ω' dans la condition 3.2.1/iii et en conséquence R n'est pas défini? C'est le meilleur cas car les configurations optimales *globalement* sont également optimales *localement*. Ceci signifie que nous n'avons pas de restriction sur l'intervalle $[T_k^{inf}, T_k^{sup}]$, n'importe quelle loi de température *locale* qui satisfait les conditions 3.2.1/i–3.2.1/ii peut être utilisée.

3.2.1 Application au modèle hiérarchique

Le modèle hiérarchique exige beaucoup plus d'itérations par pixel que les modèles mono-grilles. C'est pourquoi le recuit classique est trop lent même sur une machine parallèle. Pourtant, grâce à la structure pyramidale du modèle, nous pouvons définir un schéma multi-température qui consiste à associer des températures élevées au niveaux les plus hauts afin de diminuer la sensibilité aux minima locaux sur les grilles grossières (voir Figure 3.3). Pour les cliques entre deux niveaux voisins, nous utilisons la température associée au plus haut niveau ou celle du niveau inférieur (mais nous gardons le même niveau pendant l'exécution de l'algorithme). Dans le Paragraphe 3.4, nous comparons le recuit multi-température au recuit inhomogène classique. Les expériences montrent que le recuit multi-température converge beaucoup plus rapidement que le recuit classique.

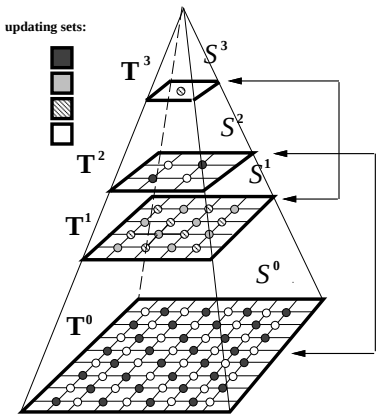

Figure 3.3: *Schéma de relaxation sur la pyramide.*

3.3 Relaxation déterministe

Les algorithmes stochastiques atteignent un minimum global mais exigent un calcul important. D'un autre côté, le minimum n'est obtenu que théoriquement. En pratique, nous mettons toujours en œuvre une *approximation* du recuit simulé, la convergence vers un minimum global n'est donc pas assurée.

Pour augmenter la rapidité de la convergence, plusieurs auteurs proposent des algorithmes *déterministes*: *Iterated Conditional Modes* (ICM) [8],*Graduated Non-Convexity*

(GNC) [10], *Deterministic Pseudo Annealing* (DPA) [7], *Game Strategy Annealing* (GSA) [61], etc... La propriété commune de ces algorithmes est qu'ils ne réalisent qu'une descente d'énergie et, en conséquence, convergent toujours vers un minimum *local* qui dépend plus ou moins de la configuration initiale [8, 10, 53, 52, 54].

Nous proposons dans le paragraphe suivant un algorithme déterministe.

3.3.1 Dynamique de Metropolis modifiée (MMD)

MMD [53, 52, 54] est une variante déterministe de l'algorithme de Metropolis. Pour des températures élevées, le comportement de notre algorithme est similaire aux algorithmes stochastiques mais si la température est inférieure à un certain seuil, il devient déterministe. La "longueur" de la phase "pseudo-stochastique" est contrôlée par un constant utilisé dans les dynamiques modifiés. La différence entre l'algorithme de Metropolis et notre approche est le choix de ξ dans l'Étape ③ de l'Algorithme 3.1.1. Pour l'algorithme original, ξ est choisi aléatoirement à chaque itération, pour notre algorithme ξ est un seuil constant, disant α, qui est fixé au début de l'algorithme.

Algorithme 3.3.1 (MMD)

① *Soit ω^0 une configuration initiale aléatoire, $k = 0$ et $T = T_0$.*

② *Choisir une configuration η aléatoirement avec une distribution uniforme qui est différente de ω^k en un seul élément.*

③ **(Critère de Metropolis modifié)** *Calculer $\Delta U = U(\eta) - U(\omega)$ et accepter η en utilisant la règle suivante:*

$$\omega^{k+1} = \begin{cases} \eta & \text{si } \Delta U \leq 0, \\ \eta & \text{si } \Delta U > 0 \text{ et } \ln(\alpha) \leq \left(-\frac{\Delta U}{T}\right), \\ \omega^k & \text{sinon} \end{cases} \tag{3.27}$$

où α est un seuil constant fixé au début de l'algorithme ($\alpha \in (0, 1)$).

④ *Décroitre la température $T = T_{k+1}$ et répéter l'Étape ② jusqu'à la convergence (par exemple ΔU est inférieur à un certain seuil).*

L'algorithme MMD est plus rapide que l'algorithme original (voir Paragraphe 3.4), car dans l'Étape ②, nous ne calculons que $\Delta U/T$ qui est comparé à $\ln(\alpha)$, alors que pour la méthode originale, nous devons calculer $\exp(-\Delta U/T)$ à chaque itération. L'initialisation n'est pas vitale car la *phase pseudo-stochastique* donne une bonne configuration

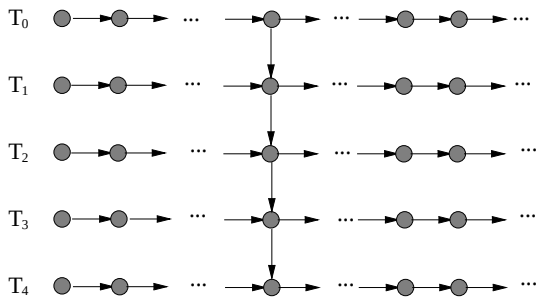

Figure 3.4: *Schème systolique.*

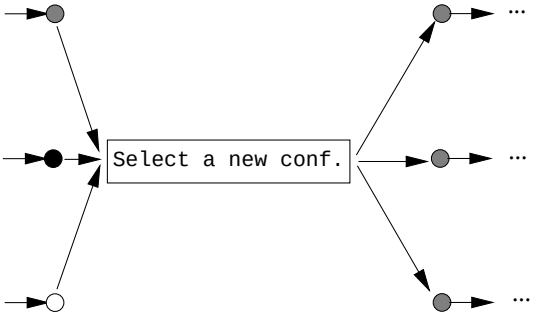

Figure 3.5: *Schème groupé.*

initiale pour la *phase déterministe*. Il n'existe pas de formule explicite pour calculer α. En pratique, une α plus importante (proche de 1) est choisie si l'énergie est lisse, sinon une valeur plus faible doit être assignée à α.

Théorème 3.3.1 (MMD) *Pour chaque $\alpha \in (0,1)$, il existe un seuil de température*

$$T_\alpha = -\frac{\Delta U_{min}}{\ln(\alpha)} \tag{3.28}$$

$$\text{où } \Delta U_{min} = \min_{\substack{\omega,\eta \in \Omega \\ U(\omega) \neq U(\eta)}} \mid U(\omega) - U(\eta) \mid \tag{3.29}$$

tel que si $T_k < T_\alpha$, les configurations avec une énergie décroissante seront acceptées, et l'algorithme converge vers un minimum local.

Si $T_k = \Gamma / \ln(k)$ alors $T_k < T_\alpha$ si et seulement si $k > K_\alpha$ où K_α est un seuil donné par:

$$K_\alpha = \exp\left(-\frac{\Gamma \cdot \ln(\alpha)}{\Delta U_{min}}\right). \tag{3.30}$$

Autrement dit, après K_α itérations, l'algorithme entre dans la *phase déterministe*. La démonstration du théorème se trouve dans l'Annexe 3.B.

3.3.2 Parallélisation

La méthode la plus naturelle dans la plupart des problèmes de traitement d'images est la *méthode de codage* [8]. Elle consiste à construire *des ensembles de codage* tels

que les pixels appartenant aux mêmes ensembles soient conditionnellement indépendants (voir Figure 3.6). L'avantage de cette méthode est qu'elle garde les propriétés de convergence de l'algorithme séquentiel.

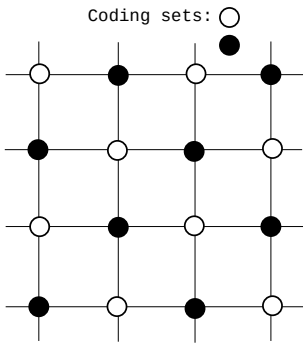

Figure 3.6: *Ensembles de codage dans le cas d'un modèle markovien d'ordre 1.*

Une autre méthode a été proposée dans [2]: à chaque itération k, chaque site appartient à l'ensemble des site actifs A_k avec une probabilité de τ ($\tau \in (0, 1]$ est fixé). Ensuite, les sites actifs sont mis à jour au même temps. Dans ce cas, la convergence a été démontrée par Trouvé [82] pour $\tau \in (0, 1)$, sans montrer que la limite est la même que dans le cas séquentiel.

Les méthodes décrites supposent que l'espace des configurations peut être partitionné (c'est souvent le cas en traitement d'images). Dans le cas contraire, il existe deux approches: *l'algorithme systolique* [57, 33, 32, 4] dont le but est de générer plusieurs chaînes de Markov (possibilité d'utiliser des températures différentes [32, 4, 33]) avec des intéractions entre les différentes chaînes (cf. Figure 3.4); et la seconde est l'algorithme groupé où tous les processeurs génèrent cooperativement la même chaîne de Markov [16, 57].

Pour les modèles multi-échelles, quelques méthodes plus spécifiques ont été proposées dans [36, 65, 66]. Dans le paragraphe suivant, nous nous intéressons à la parallélisation de notre modèle hiérarchique.

3.3.3 Algorithme parallèle hiérarchique

De point de vue de l'optimisation. il n'y a pas de différence entre les modèles monogrilles et hiérarchiques. Nous devons minimiser une fonction non-convexe définie sur un espace de configurations qui peut être partitionné. Nous pouvons donc utiliser les mêmes techniques de parallélisation. Cependant, il existe une situation spécifique pour le modèle hiérarchique: le recuit multi-température en utilisant la méthode de codage. Dans le Paragraphe 3.2.1, nous avons déjà expliqué comment mettre en œuvre le recuit multi-température sur la pyramide (cf. Figure 3.3). Pour la parallélisation, nous pouvons définir des ensembles de codage décrits dans le Paragraphe 3.3.2. Dans la Figure 3.3, en considérant les interactions entre deux niveaux voisins, les couches reliées par pointers seront mis à jour en même temps. Bien sûr, à chaque niveau, nous devons définir des ensembles supplémentaires pour les communications intra-niveaux.

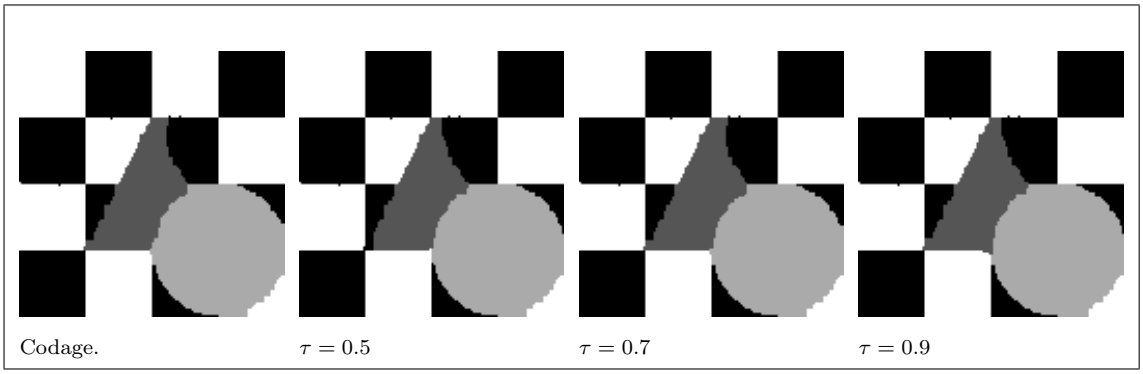

Figure 3.7: *Résultats de l'échantillonneur de Gibbs avec différent techniques de parallélisation.*

Echantillonneur de Gibbs	VPR	Nb. d'Iter	Temps total	Temps per It.	Energie
Codage	2	342	14.21 sec.	0.042 sec.	44190.63
$\tau = 0.5$	2	333	14.12 sec.	0.042 sec.	44195.77
$\tau = 0.7$	2	337	14.10 sec.	0.041 sec.	44190.28
$\tau = 0.9$	2	366	15.49 sec.	0.042 sec.	44192.86

Tableau 3.1: *Résultats de l'échantillonneur de Gibbs avec différent techniques parallélisation.*

3.4 Résultats expérimentaux

Le but de ce paragraphe est d'évaluer les performances des algorithmes décrits, en particulier sur les problèmes de segmentation d'images synthétiques et réelles. Les tests ont été réalisés sur une machine à connexions CM200 [39] en utilisant la méthode de codage pour la parallélisation (voir Paragraphe 3.3.2). Nous avons testé également la méthode de parallelisation de rapport τ décrit dans le Paragraphe 3.3.2, mais nous avons obtenu pratiquement les mêmes résultats (voir Figure 3.7 et Tableau 3.1). Étant donné que la convergence vers un minimum global n'est pas assurée, nous avons décidé d'utiliser le codage pour les tests.

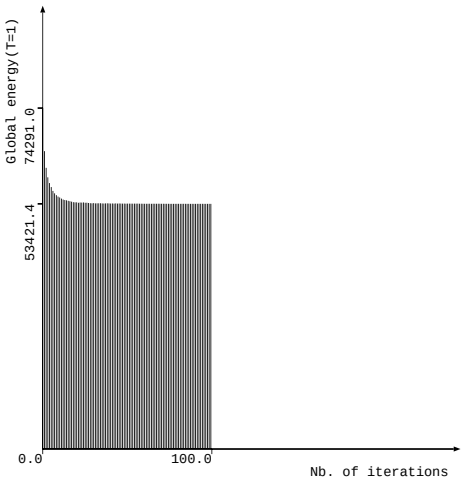

Figure 3.8: *Décroissance d'énergie avec le recuit multi-température.*

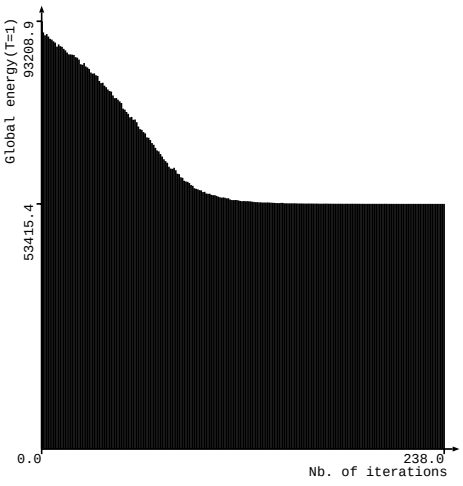

Figure 3.9: *Décroissance d'énergie avec le recuit inhomogène.*

3.4.1 Comparaison les recuits classique et multi-température

Dans la Figure 3.10, nous comparons le recuit inhomogène et multi-température sur une image synthétique bruitée en utilisant l'échantillonneur de Gibbs. La fonction d'énergie du modèle hiérarchique est définie dans le Paragraphe 2.4.4. Nous avons utilisé strictement les mêmes paramètres pour les deux algorithmes: la pyramide contient 4 niveaux, le VPR est égal à 4. La température initiale a été respectivement 4 pour le plus haut niveau , 3, 2 et 1 pour le plus bas niveau et 4 pour tous les niveau dans le cas du recuit inhomogène. Le potentiel β est égal à 0.7 et γ est égal à 0.1. Dans la Figure 3.9 et Figure 3.8, nous montrons l'énergie globale calculée à une température fixée en fonction du nombre d'itérations. Tous les deux atteinent pratiquement le même minimum (53415.4 pour le recuit inhomogène et 53421.4 pour le recuit multi-température), mais l'algorithme inhomogène exige 238 itérations (796.8 sec. d'exécution) quant au recuit multi-température, il suffit de 100 itérations (340.6 sec. d'exécution).

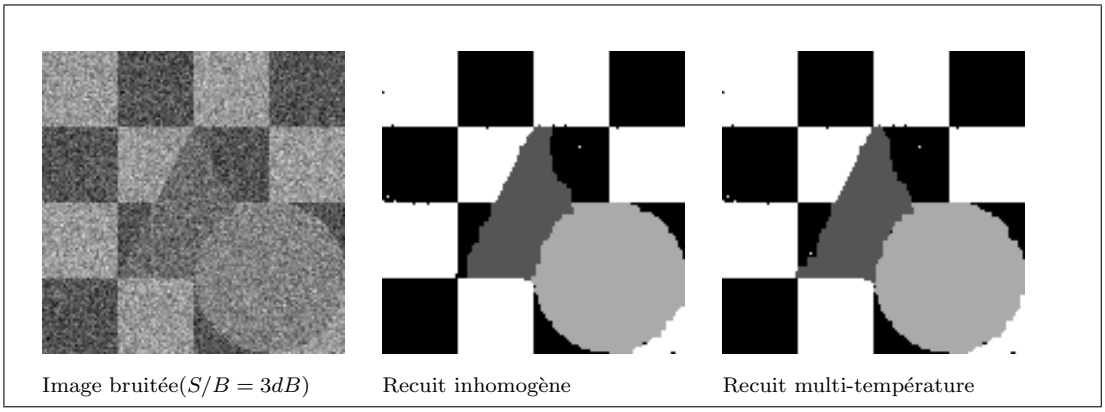

Figure 3.10: *Résultats de l'échantillonneur de Gibbs sur une image synthétique.*

3.4.2 Les algorithmes stochastiques et déterministiques

Pour tester les algorithmes, nous avons utilisé un modèle markovien d'ordre un avec la fonction d'énergie suivante (voir Paragraphe 2.2):

$$U(\omega, f) = \sum_{s \in \mathcal{S}} \left(\ln(\sqrt{2\pi}\sigma_{\omega_s}) + \frac{(f_s - \mu_{\omega_s})^2}{2\sigma_{\omega_s}^2} \right) + \sum_{\{s,r\} \in \mathcal{C}} \beta \delta(\omega_s, \omega_r) \tag{3.31}$$

où

$$\delta(\omega_s, \omega_r) = \begin{cases} -1 & \text{si } \omega_s = \omega_r \\ +1 & \text{si } \omega_s \neq \omega_r \end{cases} \tag{3.32}$$

et β est le paramètre qui contrôle l'homogénité des régions. Chaque classe $\lambda \in \Lambda$ est représentée par sa valeur moyenne μ_λ et son écart-type. Les paramètres sont donnés par le Tableau 3.3 dans l'Annexe 3.D.

La température initiale pour les algorithmes utilisant une loi de température (échantillonneur de Gibbs, Metropolis, MMD, GSA) a été $T_0 = 4$ et la loi a été $T_{k+1} = 0.95 \cdot T_k$. Pour MMD et GSA, le paramètre α est donné par Tableau 3.2. ICM et DPA ont été initialisés par le terme gaussien de la fonction d'énergie. Pour les autres algorithmes, une valeur aléatoire a été choisie.

Image	α
checkerboard	0.3
triangle	0.3
bruit	0.7
SPOT	0.7

Tableau 3.2: *Le paramètre α pour MMD et GSA.*

Les résultats obtenus sont présentés dans l'Annexe 3.C et l'Annexe 3.D. Nous pouvons constater que les algorithmes stochastiques donnent les plus basses énergies finales

mais ils sont plus lents que les méthodes déterministes. ICM est le plus rapide mais le minimum atteint est supérieur à celui des autres méthodes. DPA, MMD et GSA sont des bons compromis entre la qualité finale des résultats et le temps d'exécution. Un autre avantage est qu'ils sont moins dépendants des conditions initiales que ICM.

Annexe

3.A Démonstration du théorème de recuit multi-température

3.A.1 Notations

Soit $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ l'ensemble des sites. $\Lambda = \{0, 1, \ldots, L-1\}$ dénote l'espace d'états commun et $\omega, \eta, \eta' \ldots \in \Omega$ dénotent les configurations, où $\Omega = \Lambda^N$ est fini. Les sites sont mis à jour dans l'ordre $\{n_1, n_2, \ldots\} \subset \mathcal{S}$. Les configurations générées constituent une chaîne de Markov inhomogène $\{X(k), k = 0, 1, 2, \ldots\}$, où $X(0)$ est la configuration initiale. La transition $X(k-1) \to X(k)$ est gouvernée par la distribution de Gibbs $\pi_{T(k,C)}$, en utilisant la matrice de transition à l'itération k:

$$P_{\omega,\eta}(k) == \begin{cases} \pi_{T(k,C)}(X_{n_k} = \eta_{n_k} \mid X_s = \eta_s, s \neq n_k), & \text{si } \eta = \omega|_{\omega_{n_k}=\lambda} \text{ pour } \lambda \in \Lambda \\ 0, & \text{sinon} \end{cases} \tag{3.33}$$

$\pi_{T(k,C)}(\omega)$ désigne la distribution de Gibbs à l'itération k

$$\pi_{T(k,C)}(\omega) = \frac{\exp(-U(\omega) \oslash T(k,C))}{Z} \tag{3.34}$$

$$\text{avec } U(\omega) \oslash T(k,C) = \sum_{C \in \mathcal{C}} \frac{V_C(\omega)}{T(k,C)} \,. \tag{3.35}$$

Les caractéristiques locales sont notées par:

$$\pi_{T(k,C)}(X_s = \omega_s \mid X_r = \omega_r, s \neq r) = \frac{1}{Z_s} \exp\left(- \sum_{C \in \mathcal{C}: s \in C} \frac{V_C(\omega)}{T(k,C)} \right) \tag{3.36}$$

$$\text{avec } Z_s = \sum_{\lambda \in \Lambda} \exp\left(- \sum_{C \in \mathcal{C}: s \in C} \frac{V_C(\omega|_{\omega_s=\lambda})}{T(k,C)} \right) \tag{3.37}$$

La décomposition de $U(\omega) - U(\eta)$ pour ω et η quelconques ($\omega \neq \eta$) est donnée par:

$$U(\omega) - U(\eta) = \sum_{C \in \mathcal{C}} (V_C(\omega) - V_C(\eta)). \tag{3.38}$$

Si $\Sigma^+(\omega, \eta)$ denote la somme sur les cliques positives et $\Sigma^-(\omega, \eta)$ la somme sur les cliques négatives, on obtient:

$$\sum_{C \in \mathcal{C}} (V_C(\omega) - V_C(\eta))$$

$$= \underbrace{\sum_{C\in\mathcal{C}:(V_C(\omega)-V_C(\eta))<0} (V_C(\omega)-V_C(\eta))}_{\Sigma^-(\omega,\eta)} + \underbrace{\sum_{C\in\mathcal{C}:(V_C(\omega)-V_C(\eta))\geq 0} (V_C(\omega)-V_C(\eta))}_{\Sigma^+(\omega,\eta)}. \quad (3.39)$$

De plus, soit

$$U^{sup} = \max_{\omega\in\Omega} U(\omega), \quad (3.40)$$

$$U^{inf} = \min_{\omega\in\Omega} U(\omega), \quad (3.41)$$

$$\text{et } \Delta = U^{sup} - U^{inf}. \quad (3.42)$$

et soit Σ^+_Δ le minimum des sommes positifs:

$$\Sigma^+_\Delta = \min_{\substack{\omega'\in\Omega_{sup}\\ \omega''\in\Omega_{opt}}} \Sigma^+(\omega',\omega''). \quad (3.43)$$

Évidemment, $\Delta \leq \Sigma^+_\Delta$.

Étant donné une distribution initiale μ_0, la distribution de $X(k)$ est donnée par le vecteur $\mu_0 \prod_{i=1}^k P(i)$:

$$P_{\mu_0}(X(k)=\omega) = \left(\mu_0 \prod_{i=1}^k P(i)\right)\Big|_\omega \quad (3.44)$$

$$= \sum_\eta P(X(k)=\omega|X(0)=\eta)\mu_0(\eta) \quad (3.45)$$

Nous utilisons la notation suivante pour les transitions: $\forall l < k$ and $\omega,\eta\in\Omega$:

$$P(k,\omega|l,\eta) = P(X(k)=\omega|X(l)=\eta),$$

et pour n'importe quelle distribution μ sur Ω:

$$P(k,\omega|l,\mu) = \sum_\eta P(X(k)=\omega|X(l)=\eta)\mu(\eta).$$

Parfois, nous utilisons la notation $P(k,\cdot|l,\mu)$, où "$\cdot$" se réfère à *n'importe quelle* configuration de Ω. Finalement, soit $\|\mu-\nu\|$ la distance suivante entre deux distributions sur Ω:

$$\|\mu-\nu\| = \sum_\omega |\mu(\omega)-\nu(\omega)|.$$

Il est clair que $\lim_{n\to\infty}\mu_n = \mu$ en distribution (c.a.d. $\forall\omega : \mu_n(\omega)\to\mu(\omega)$) si et seulement si $\|\mu_n-\mu\|\to 0$.

3.A.2 Démonstration

Tout d'abord, nous prouvons deux lemmes qui impliquent le Théorème 3.2.1:

Lemme 3.A.1 *$\forall k_0 = 0, 1, 2 \ldots$:*

$$\lim_{k\to\infty} \sup_{\omega,\eta',\eta''} |P(X(k) = \omega | X(k_0) = \eta') - P(X(k) = \omega | X(k_0) = \eta'')| = 0. \qquad (3.46)$$

Démonstartion du Lemme 3.A.1:
Soit $k_0 = 0, 1, 2, \ldots$ fixé et définissons $K_l = k_0 + l\kappa$, $l = 0, 1, 2, \ldots$, où κ est le nombre de transitions nécessaires pour une mise à jour complète de $\mathcal{S}$ (pour chaque $k = 0, 1, 2, \ldots$: $\mathcal{S} \subseteq \{n_{k+1}, n_{k+2}, \ldots, n_{k+\kappa}\}$). Soit $\delta(k)$ la plus petite probabilité entre les caractéristiques locales:

$$\delta(k) = \inf_{\substack{1\le i\le N \\ \omega\in\Omega}} \pi_{T(k,C)}(X_{s_i} = \omega_{s_i} | X_{s_j} = \omega_{s_j}, j \neq i).$$

Un seuil bas de $\delta(k)$ est le suivant:

$$\delta(k) \geq \frac{\exp(-U^{sup} \oslash T(k,C))}{L \exp(-U^{inf} \oslash T(k,C))} = \frac{\exp(-\Delta \oslash T(k,C))}{L} \geq \frac{1}{L} \exp(-\Sigma_\Delta^+ \oslash T(k,C)$$

$$\geq \frac{1}{L} \exp(-\Sigma_\Delta^+ / T_k^{inf}) \ ,$$

où $L = \mid \Lambda \mid$ est le nombre des états possibles en un site. Fixons maintenant l et soit m_i l'indice de la dernière mise à jour du site s_i avant $K_l + 1$ (c'est a dire avant la $l^{\text{ième}}$ mise à jour complète de $\mathcal{S}$):

$$\forall i: 1 \leq i \leq N : m_i = \sup\{k : k \leq K_l, n_k = s_i\}.$$

Si nous supposons que $m_1 > m_2 \cdots > m_N$, alors:

$$P(X(K_l) = \omega | X(K_{l-1}) = \omega')$$

$$= P(X_{s_1}(m_1) = \omega_{s_1}, X_{s_2}(m_2) = \omega_{s_2}, \ldots, X_{s_N}(m_N) = \omega_{s_N} | X(K_{l-1}) = \omega')$$

$$= \prod_{i=1}^{N-1} P(X_{s_i}(m_i) = \omega_{s_i} | X_{s_{i+1}}(m_{i+1}) = \omega_{s_{i+1}}, \ldots, X_{s_N}(m_N) = \omega_{s_N}, X(K_{l-1}) = \omega')$$

$$\geq \prod_{i=1}^{N} \delta(m_i) \geq L^{-N} \prod_{i=1}^{N} \exp(-\Delta / T_{m_i}^{inf}) \geq L^{-N} \exp\left(-\frac{\Sigma_\Delta^+ N}{T_{k_0+l\kappa}^{inf}}\right) \qquad (3.47)$$

car $m_i \leq K_l = k_0 + l\kappa, i = 1, 2 \ldots, N$ et T_k^{inf} est décroissante. Si $k_0 + l\kappa$ est suffisement grand alors $T_{k_0+l\kappa}^{inf} \geq N\Sigma_\Delta^+ / \ln(k_0 + l\kappa)$ (condition 3.2.1/ii) et l'Équation (3.47) peut être continué:

$$P(X(K_l) = \omega | X(K_{l-1}) = \omega') \geq L^{-N} \exp\left(-\frac{\Sigma_\Delta^+ N}{N\Sigma_\Delta^+ / \ln(k_0 + l\kappa)}\right) = L^{-N}(k_0 + l\kappa)^{-1}.$$

Nous pouvons donc supposer pour un constant Γ $(0 < \Gamma \leq 1)$ suffisement grand que:

$$\inf_{\omega,\omega'} P(X(K_l) = \omega | X(K_{l-1}) = \omega') \geq \frac{\Gamma L^{-N}}{k_0 + l\kappa} \tag{3.48}$$

pour chaque $k_0 = 0, 1, 2, \ldots$ et $l = 1, 2, \ldots$.

Considérons maintenant la limite donnée par l'Équation (3.46) et pour chaque $k > k_0$, définissons $K^{sup}(k) = \sup\{l : K_l < k\}$ tel que $\lim_{k\to\infty} K^{sup}(k) = \infty$. Soit $k > K_1$ fixé:

$$\sup_{\omega,\eta',\eta''} |P(X(k) = \omega | X(0) = \eta') - P(X(k) = \omega | X(0) = \eta'')|$$

$$= \sup_\omega \left(\sup_\eta P(X(k) = \omega | X(0) = \eta) - \inf_\eta P(X(k) = \omega | X(0) = \eta)\right)$$

$$= \sup_\omega \left(\sup_\eta \sum_{\omega'} P(X(k) = \omega | X(K_1) = \omega') P(X(K_1) = \omega' | X(0) = \eta)\right.$$

$$\left. - \inf_\eta \sum_{\omega'} P(X(k) = \omega | X(K_1) = \omega') P(X(K_1) = \omega' | X(0) = \eta)\right)$$

$$\doteq \sup_\omega Q(k, \omega).$$

De plus, pour chaque $\omega \in \Omega$:

$$\sup_\eta \sum_{\omega'} P(X(k) = \omega | X(K_1) = \omega') P(X(K_1) = \omega' | X(0) = \eta)$$

$$\leq \sup_\mu \sum_{\omega'} P(X(k) = \omega | X(K_1) = \omega') \mu(\omega'),$$

où μ est une mesure de probabilité sur Ω. En utilisant l'Équation (3.48), on a:

$$\mu(\omega') \geq \frac{\Gamma L^{-N}}{k_0 + l\kappa}.$$

Supposons que $P(X(k) = \omega | X(K_1) = \omega')$ a un maximum pour $\omega' = \omega^{sup}$ et un minimum pour $\omega' = \omega^{inf}$. On a donc:

$$\sup_{\mu} \sum_{\omega'} P(X(k) = \omega | X(K_1) = \omega') \mu(\omega') \leq$$

$$\left(1 - (L^N - 1)\frac{\Gamma L^{-N}}{k_0 + l\kappa}\right) P(X(k) = \omega | X(K_1) = \omega^{sup})$$

$$+\frac{\Gamma L^{-N}}{k_0 + l\kappa} \underbrace{\sum_{\omega' \neq \omega^{sup}} P(X(k) = \omega | X(K_1) = \omega')}_{P(X(k)=\omega|X(K_1)=\omega^{inf}) + \sum_{\omega' \neq \omega^{sup}, \omega^{inf}} P(X(k)=\omega|X(K_1)=\omega')} ,$$

et de la même façon:

$$\inf_{\mu} \sum_{\omega'} P(X(k) = \omega | X(K_1) = \omega') \mu(\omega') \geq$$

$$\left(1 - (L^N - 1)\frac{\Gamma L^{-N}}{k_0 + l\kappa}\right) P(X(k) = \omega | X(K_1) = \omega^{inf})$$

$$+\frac{\Gamma L^{-N}}{k_0 + l\kappa} \underbrace{\sum_{\omega' \neq \omega^{inf}} P(X(k) = \omega | X(K_1) = \omega')}_{P(X(k)=\omega|X(K_1)=\omega^{sup}) + \sum_{\omega' \neq \omega^{sup}, \omega^{inf}} P(X(k)=\omega|X(K_1)=\omega')} .$$

Il est donc clair que

$$Q(k, \omega) \leq \left(1 - \frac{\Gamma}{k_0 + l\kappa}\right) \left(P(X(k) = \omega | X(K_1) = \omega^{sup}) - P(X(k) = \omega | X(K_1) = \omega^{inf})\right),$$

désormais:

$$\sup_{\omega, \eta', \eta''} |P(X(k) = \omega | X(0) = \eta') - P(X(k) = \omega | X(0) = \eta'')| \leq$$

$$\left(1 - \frac{\Gamma}{k_0 + l\kappa}\right) \sup_{\omega, \eta', \eta''} |P(X(k) = \omega | X(K_1) = \eta') - P(X(k) = \omega | X(K_1) = \eta'')| \leq$$

$$\left(1 - \frac{\Gamma}{k_0 + l\kappa}\right) \left(\left(1 - \frac{\Gamma}{k_0 + l\kappa}\right) \sup_{\omega, \eta', \eta''} |P(X(k) = \omega | X(K_2) = \eta') - P(X(k) = \omega | X(K_2) = \eta'')|\right)$$

En continuant, nous obtenons le seuil suivant:

$$\leq \prod_{k=1}^{K^{sup}(k)} \left(1 - \frac{\Gamma}{k_0 + l\kappa}\right) \sup_{\omega, \eta', \eta''} \left|P(X(k) = \omega | X(K_{K^{sup}(k)}) = \eta')\right.$$

$$-P(X(k)=\omega|X(K_{K^{sup}(k)})=\eta'')\Big|$$

Finalement, puisque la valeur maximale suprême est 1, on a:

$$\sup_{\omega,\eta',\eta''}|P(X(k)=\omega|X(0)=\eta')-P(X(k)=\omega|X(0)=\eta'')|\leq\prod_{k=1}^{K^{sup}(k)}\left(1-\frac{\Gamma}{k_0+l\kappa}\right).$$

Il suffit donc de montrer que:

$$\lim_{m\to\infty}\prod_{k=1}^{m}\left(1-\frac{\Gamma}{k_0+l\kappa}\right)=0.$$

Ce qui est une conséquence bien connue de la divergence de la série

$$\sum_{l}(k_0+l\kappa)^{-1}$$

$\forall k_0$ et κ. **Q.E.D.**

Lemme 3.A.2

$$\lim_{k_0\to\infty}\sup_{k\geq k_0}\parallel P(k,\cdot|k_0,\pi_0)-\pi_0\parallel=0. \tag{3.49}$$

Démonstration du Lemme 3.A.2:
Notons $P(k,\cdot|k_0,\pi_0)$ par $P_{k_0,k}(\cdot)$ tel que pour chaque $k\geq k_0>0$:

$$P_{k_0,k}(\omega)=\sum_{\eta}P(X(k)=\omega|X(k_0)=\eta)\pi_0(\eta).$$

Dans un premier temps, nous montrons que pour n'importe quel $k>k_0\geq 0$:

$$\|P_{k_0,k}-\pi_{T(k,C)}\|\leq\|P_{k_0,k-1}-\pi_{T(k,C)}\|. \tag{3.50}$$

Afin de faciliter les notations, nous supposons que $n_k=s_1$:

$$\|P_{k_0,k}-\pi_{T(k,C)}\|=$$

$$\sum_{(\omega_{s_1},\ldots\omega_{s_N})}\Big|\pi_{T(k,C)}(X_{s_1}=\omega_{s_1}|X_s=\omega_s,s\neq s_1)P_{k_0,k-1}(X_s=\omega_s,s\neq s_1)$$

$$-\pi_{T(k,C)}(X_s=\omega_s,s\in\mathcal{S})\Big|$$

$$=\sum_{(\omega_{s_2},\ldots\omega_{s_N})}\left(\sum_{\omega_{s_1}\in\Lambda}\pi_{T(k,C)}(X_{s_1}=\omega_{s_1}|X_s=\omega_s,s\neq s_1)\,|P_{k_0,k-1}(X_s=\omega_s,s\neq s_1)\right.$$

$$-\pi_{T(k,C)}(X_s = \omega_s, s \neq s_1)\Big|\Big)$$

$$= \sum_{(\omega_{s_2},\dots\omega_{s_N})} \Big|P_{k_0,k-1}(X_s = \omega_s, s \neq s_1) - \pi_{T(k,C)}(X_s = \omega_s, s \neq s_1)\Big|$$

$$= \sum_{(\omega_{s_2},\dots\omega_{s_N})} \left|\sum_{\omega_{s_1}} (P_{k_0,k-1}(X_s = \omega_s, s \in \mathcal{S}) - \pi_{T(k,C)}(X_s = \omega_s, s \in \mathcal{S}))\right|$$

$$\leq \sum_{(\omega_{s_1},\dots\omega_{s_N})} \Big|P_{k_0,k-1}(X_s = \omega_s, s \in \mathcal{S}) - \pi_{T(k,C)}(X_s = \omega_s, s \in \mathcal{S})\Big|$$

$$= \|P_{k_0,k-1} - \pi_{T(k,C)}\|.$$

Dans un deuxième temps, nous montrons que $\pi_{T(k,C)}$ converge vers π_0 (la distribution uniforme sur Ω_{opt}):

$$\lim_{k\to\infty} \|\pi_0 - \pi_{T(k,C)}\| = 0.$$

Soit $|\Omega_{opt}|$ le nombre des configurations optimales:

$$\lim_{k\to\infty} \pi_{T(k,C)}(\omega)$$

$$= \lim_{k\to\infty} \frac{\exp(-U(\omega) \oslash T(k,C))}{\sum_{\omega'\in\Omega_{opt}} \exp(-U(\omega') \oslash T(k,C)) + \sum_{\omega'\notin\Omega_{opt}} \exp(-U(\omega') \oslash T(k,C))}$$

$$= \lim_{k\to\infty} \frac{\exp(-(U(\omega) - U^{inf}) \oslash T(k,C))}{|\Omega_{opt}| + \sum_{\omega'\notin\Omega_{opt}} \exp(-(U(\omega) - U^{inf}) \oslash T(k,C))} = \begin{cases} 0 & \omega \notin \Omega_{opt} \\ \frac{1}{|\Omega_{opt}|} & \omega \in \Omega_{opt} \end{cases} \tag{3.51}$$

L'équation précédente est vrai si $(U(\omega) - U^{inf}) \oslash T(k,C) \geq 0$. Cette inégalité peut se mettre sous la forme suivante:

$$\sum_{C\in\mathcal{C}} \frac{V_C(\omega) - V_C(\omega')}{T(k,C)} \geq 0 \tag{3.52}$$

où ω' est une configuration optimale (c.a.d. $\omega' \in \Omega_{opt}$). $V_C(\omega) - V_C(\omega')$ peut être négatif, mais $U(\omega) - U^{inf}$ est toujours positif ou null. Désignons par $\Sigma(\omega)$ la différence d'énergie dans l'Équation (3.52) sans tenir compte de la température. Évidemment, elle est non-negative:

$$\Sigma(\omega) = \sum_{C\in\mathcal{C}} V_C(\omega) - V_C(\omega') = U(\omega) - U^{inf} \geq 0$$

Puis, décomposons $\Sigma(\omega)$ en utilisant l'Équation (3.21):

$$\Sigma(\omega) = \Sigma^+(\omega,\omega') + \Sigma^-(\omega,\omega').$$

ce qui nous donne:

$$\Sigma^+(\omega,\omega') = \Sigma(\omega) - \Sigma^-(\omega,\omega').$$

Examinons maintenant l'Équation (3.52):

$$\sum_{C\in\mathcal{C}} \frac{V_C(\omega) - V_C(\omega')}{T(k,C)} = \Sigma^-(\omega,\omega') \oslash T(k,C) + \Sigma^+(\omega,\omega') \oslash T(k,C)$$

$$\geq \Sigma^-(\omega,\omega')/T_k^{inf} + \Sigma^+(\omega,\omega')/T_k^{sup} = \frac{\Sigma^-(\omega,\omega')\cdot T_k^{sup} + \Sigma^+(\omega,\omega')\cdot T_k^{inf}}{T_k^{inf} T_k^{sup}} \geq 0$$

De plus:

$$\Sigma^-(\omega,\omega')\cdot T_k^{sup} + \Sigma^+(\omega,\omega')\cdot T_k^{inf} = \Sigma^-(\omega,\omega')\cdot T_k^{sup} + (\Sigma(\omega) - \Sigma^-(\omega,\omega'))T_k^{inf}$$

Par conséquence:

$$\Sigma^-(\omega,\omega')(T_k^{sup} - T_k^{inf}) - \Sigma(\omega)\cdot T_k^{inf} \geq 0$$

En divisant par le terme négatif $\Sigma^-(\omega,\omega')$, on a:

$$T_k^{sup} - T_k^{inf} \leq \frac{\Sigma(\omega)}{\mid \Sigma^-(\omega,\omega') \mid} T_k^{inf}$$

Ceci est vrai à cause de la condition 3.2.1/iii du théorème.

Finalement, nous montrons que:

$$\sum_{k=1}^{\infty} \left\| \pi_{T(k,C)} - \pi_{T(k+1,C)} \right\| < \infty \tag{3.53}$$

car

$$\sum_{k=1}^{\infty} \left\| \pi_{T(k,C)} - \pi_{T(k+1,C)} \right\| = \sum_{\omega} \sum_{k=1}^{\infty} \left| \pi_{T(k,C)}(\omega) - \pi_{T(k+1,C)}(\omega) \right|$$

et parce que

$$\forall\omega : \pi_{T(k,C)}(\omega) \longrightarrow \pi_0(\omega),$$

il suffit de montrer que $\pi_T(\omega)$ est monotone pour chaque ω. Ceci est évident (Équation (3.51)):

- si $\omega \notin \Omega_{opt}$ alors $\pi_T(\omega)$ est strictement croissante pour $0 < T \leq \epsilon$ pour une ϵ suffisement faible,
- si $\omega \in \Omega_{opt}$ alors $\pi_T(\omega)$ est strictement décroissante pour toutes les $T > 0$.

Fixons $k > k_0 \geq 0$. En utilisant l'Équation (3.50) et l'Équation (3.53), on obtient:

$$\|P_{k_0,k} - \pi_0\| \leq \|P_{k_0,k} - \pi_{T(k,C)}\| + \|\pi_{T(k,C)} - \pi_0\|$$

$$\leq \|P_{k_0,k-1} - \pi_{T(k,C)}\| + \|\pi_{T(k,C)} - \pi_0\| \text{ par l'Équation (3.50)}$$

$$\leq \|P_{k_0,k-1} - \pi_{T(k-1,C)}\| + \|\pi_{T(k-1,C)} - \pi_{T(k,C)}\| + \|\pi_{T(k,C)} - \pi_0\|$$

$$\leq \|P_{k_0,k-2} - \pi_{T(k-2,C)}\| + \|\pi_{T(k-2,C)} - \pi_{T(k-1,C)}\| + \|\pi_{T(k-1,C)} - \pi_{T(k,C)}\| + \|\pi_{T(k,C)} - \pi_0\|$$

$$\leq \cdots \leq \|P_{k_0,k_0} - \pi_{T(k_0,C)}\| + \sum_{l=k_0}^{k-1} \|\pi_{T(l,C)} - \pi_{T(l+1,C)}\| + \|\pi_{T(k,C)} - \pi_0\|.$$

d'autre part:

$$P_{k_0,k_0} = \pi_0$$

et

$$\lim_{k\to\infty} \|\pi_{T(k,C)} - \pi_0\| = 0.$$

On a donc:

$$\overline{\lim_{k_0\to\infty}} \sup_{k\geq k_0} \|P_{k_0,k} - \pi_0\| \leq \overline{\lim_{k_0\to\infty}} \sup_{k>k_0} \sum_{l=k_0}^{k-1} \|\pi_{T(l,C)} - \pi_{T(l+1,C)}\|$$

$$= \overline{\lim_{k_0\to\infty}} \sum_{l=k_0}^{\infty} \|\pi_{T(l,C)} - \pi_{T(l+1,C)}\| = 0$$

où le dernier terme vaut 0 par l'Équation (3.53) **Q.E.D.**

Théorème 3.2.1 (Recuit multi-température) *Supposons qu'il existe un entier $\kappa \geq N$ tel que pour chaque $k = 0, 1, 2, \ldots$, $\mathcal{S} \subseteq \{n_{k+1}, n_{k+2}, \ldots, n_{k+\kappa}\}$. Pour tous les $C \in \mathcal{C}$, soit $T(k, C)$ une séquence décroissante des températures en k telle que:*

(i) $\forall C: \lim_{k\to\infty} T(k, C) = 0$.
Notons respectivement par T_k^{inf} et T_k^{sup} le maximum et le minimum de la fonction de température pour k donné ($\forall C \in \mathcal{C}: T_k^{inf} \leq T(k, C) \leq T_k^{sup}$).

(ii) Pour tous les $k \geq k_0$ et pour un entier $k_0 \geq 2$ quelconque: $T_k^{inf} \geq N\Sigma_\Delta^+ / \ln(k)$.

(iii) Si $\Sigma^-(\omega, \omega') \neq 0$ pour une $\omega \in \Omega \setminus \Omega_{opt}$ quelconque et une $\omega' \in \Omega_{opt}$ quelconque, nous devons imposer une condition supplémentaire:
$\forall k: \frac{T_k^{sup} - T_k^{inf}}{T_k^{inf}} \leq R$, *avec:*

$$R = \min_{\substack{\omega \in \Omega \setminus \Omega_{opt} \\ \omega' \in \Omega_{opt} \\ \Sigma^-(\omega,\omega') \neq 0}} \frac{U(\omega) - U^{inf}}{\mid \Sigma^-(\omega, \omega') \mid}. \tag{3.54}$$

Si les conditions sont satisfaites alors pour n'importe quelle configuration initiale $\eta \in \Omega$ et pour chaque $\omega \in \Omega$:

$$\lim_{k\to\infty} P(X(k) = \omega \mid X(0) = \eta) = \pi_0(\omega). \tag{3.55}$$

Démonstration:
En utilisant les deux lemmes, nous pouvons facilement valider le théorème:

$$\overline{\lim_{k\to\infty}} \| P(X(k) = \cdot | X(0) = \eta) - \pi_0 \| = \overline{\lim_{k_0\to\infty} \lim_{\substack{k\to\infty \\ k\geq k_0}}} \| \sum_{\eta'} P(k, \cdot | k_0, \eta') P(k_0, \eta' | 0, \eta) - \pi_0 \|$$

$$\leq \overline{\lim_{k_0\to\infty} \lim_{\substack{k\to\infty \\ k\geq k_0}}} \| \sum_{\eta'} P(k, \cdot | k_0, \eta') P(k_0, \eta' | 0, \eta) - P(k, \cdot | k_0, \pi_0) \|$$

$$+ \overline{\lim_{k_0\to\infty} \lim_{\substack{k\to\infty \\ k\geq k_0}}} \| P(k, \cdot | k_0, \pi_0) - \pi_0 \| .$$

où le dernier terme vaut 0 (Lemme 3.A.2). En plus, $P(k_0, \cdot | 0, \eta)$ et π_0 a une masse totale de 1:

$$\| \sum_{\eta'} P(k, \cdot | k_0, \eta') P(k_0, \eta' | 0, \eta) - P(k, \cdot | k_0, \pi_0) \|$$

$$= \sum_{\omega} \sup_{\eta''} |\sum_{\eta'} (P(k,\omega|k_0,\eta') - P(k,\omega|k_0,\eta''))(P(k_0,\eta'|0,\eta) - \pi_0(\eta'))|$$

$$\leq 2 \sum_{\omega} \sup_{\eta',\eta''} |P(k,\omega|k_0,\eta') - P(k,\omega|k_0,\eta'')| \ .$$

Finalement:

$$\overline{\lim_{k\to\infty}} \|P(X(k) = \cdot | X(0) = \eta) - \pi_0\|$$

$$\leq 2 \sum_{\omega} \overline{\lim_{k_0\to\infty} \lim_{\substack{k\to\infty \\ k\geq k_0}}} \sup_{\eta',\eta''} |P(k,\omega|k_0,\eta') - P(k,\omega|k_0,\eta'')| = 0$$

où le dernier terme vaut 0 (Lemme 3.A.1). **Q.E.D.**

3.B Démonstration du théorème MMD

3.B.1 Notations

Soit Ω l'ensemble de toutes les configurations. Les éléments de Ω sont notés par $\omega, \eta, \ldots$, $U(\omega)$ dénote l'énergie de ω. Une nouvelle configuration η est acceptée si:

$$\omega^{k+1} = \begin{cases} \eta & \text{si } \Delta U \leq 0, \\ \eta & \text{si } \Delta U > 0 \text{ et } \alpha \leq \exp\left(-\frac{\Delta U}{T}\right), \\ \omega^k & \text{sinon} \end{cases} \tag{3.56}$$

où α est un seuil constant ($\alpha \in (0,1)$).

3.B.2 Démonstration du théorème

Théorème 3.3.1 (MMD) *Pour chaque $\alpha \in (0,1)$, il existe un seuil de température*

$$T_\alpha = -\frac{\Delta U_{min}}{\ln(\alpha)} \tag{3.57}$$

$$\text{où } \Delta U_{min} = \min_{\substack{\omega, \eta \in \Omega \\ U(\omega) \neq U(\eta)}} \mid U(\omega) - U(\eta) \mid \tag{3.58}$$

tel que si $T_k < T_\alpha$, les configurations avec une énergie décroissante seront acceptées, et l'algorithme converge vers un minimum local.

Démonstration:
Examinons la transition $\omega \to \eta$ quand $U(\eta) > U(\omega)$. La transition est permie (voir Équation (3.56)) si:

$$\alpha \leq \exp\left(-\frac{U(\eta) - U(\omega)}{T}\right). \tag{3.59}$$

Mais en utilisant l'Équation (3.57), on a:

$$\alpha \leq \exp\left(-\frac{U(\eta) - U(\omega)}{T}\right) \leq \exp\left(-\frac{\Delta U_{min}}{T}\right), \tag{3.60}$$

et d'autre part, $T \to 0$, ce qui nous donne:

$$\lim_{k \to \infty} \exp\left(-\frac{\Delta U_{min}}{T}\right) = 0. \tag{3.61}$$

En conséquence, si $T < T_\alpha$, on obtient:

$$\exp\left(-\frac{\Delta U_{min}}{T}\right) \leq \exp\left(-\frac{\Delta U_{min}}{T_\alpha}\right) = \alpha. \tag{3.62}$$

C'est à dire que les configurations avec énergie plus élevée ne sont plus acceptées. **Q.E.D.**

3.C Images

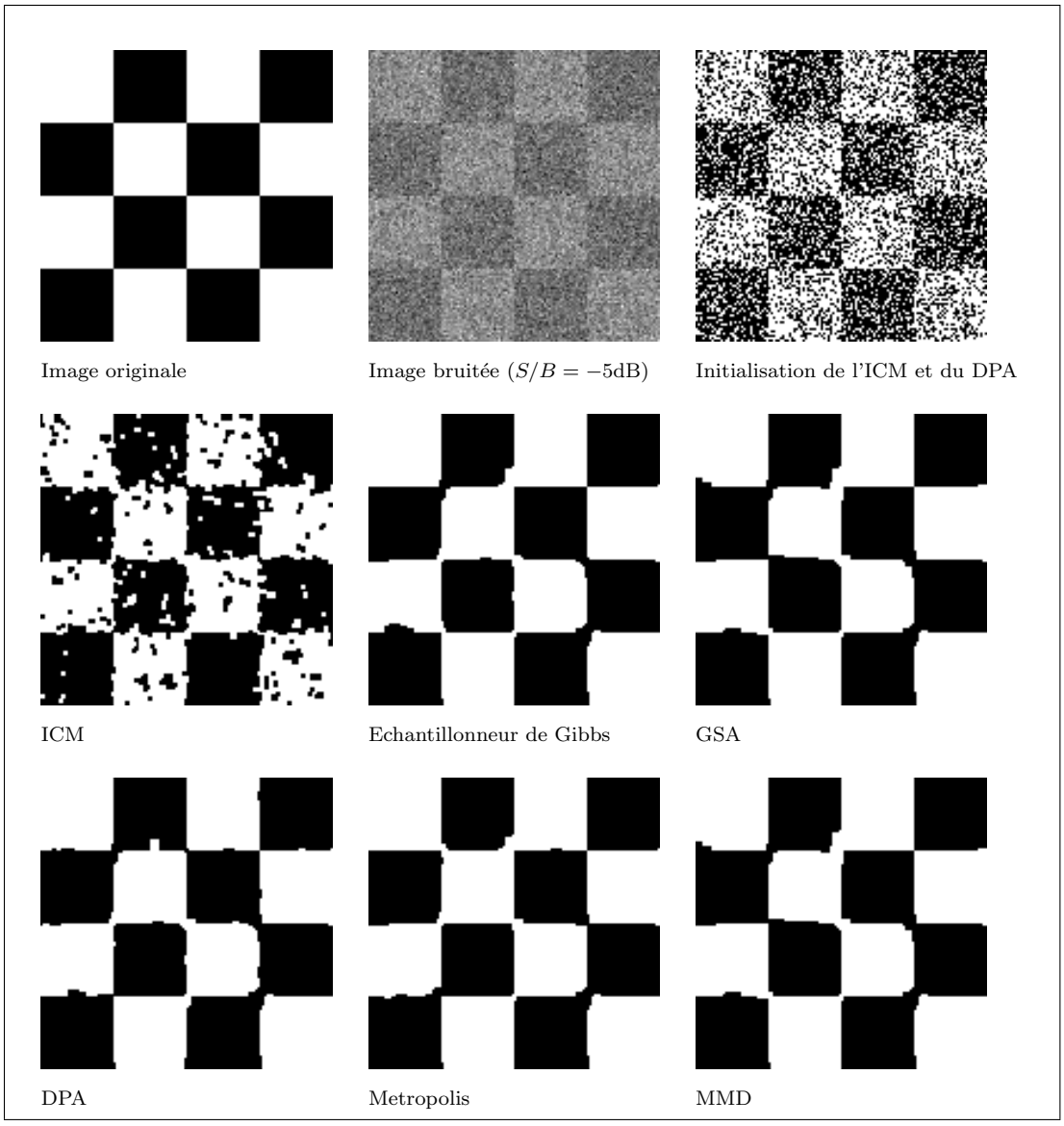

Figure 3.11: *Résultats sur l'image "checkerboard" avec* 2 *classes.*

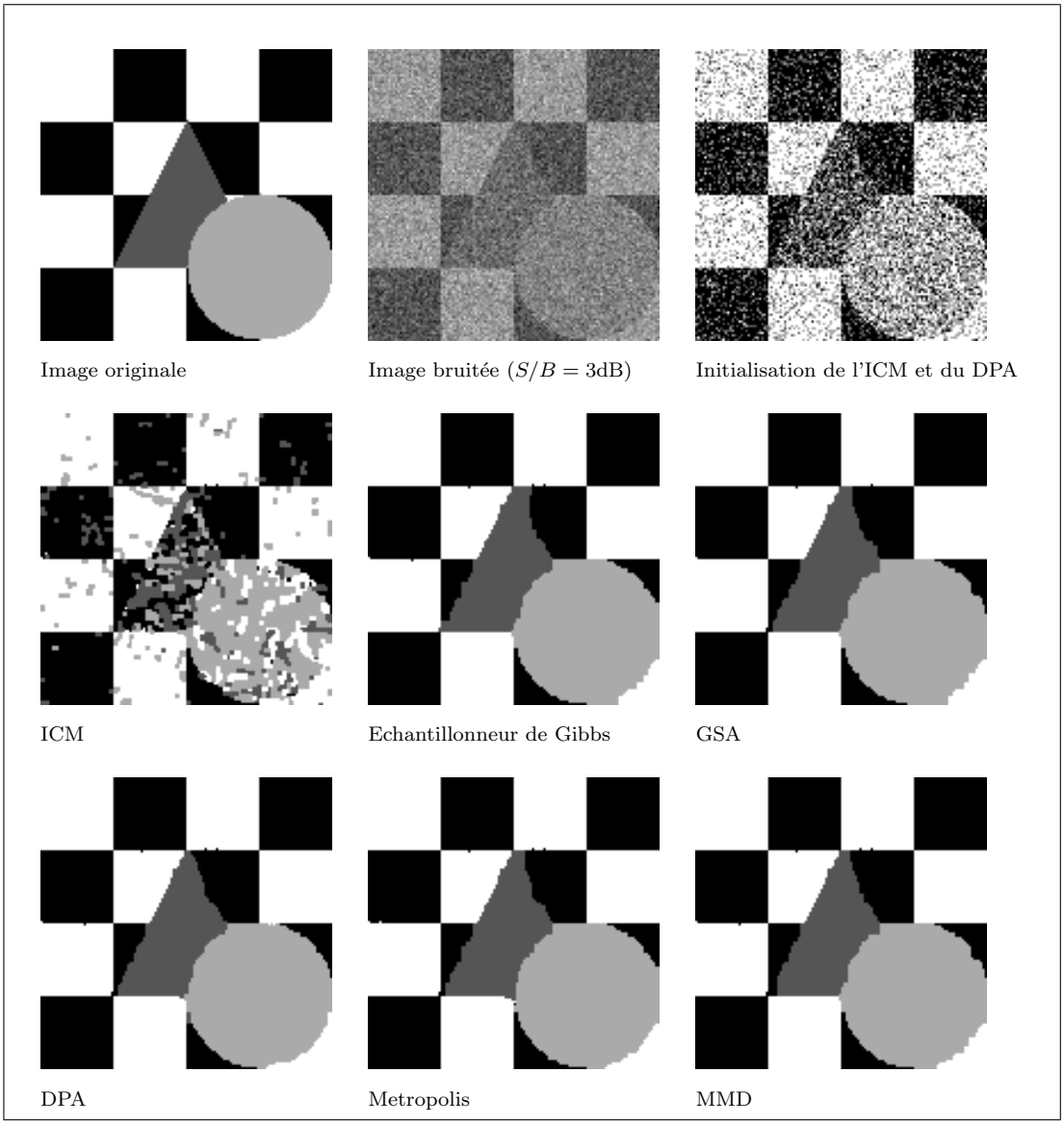

Figure 3.12: *Résultats sur l'image "triangle" avec* 4 *classes.*

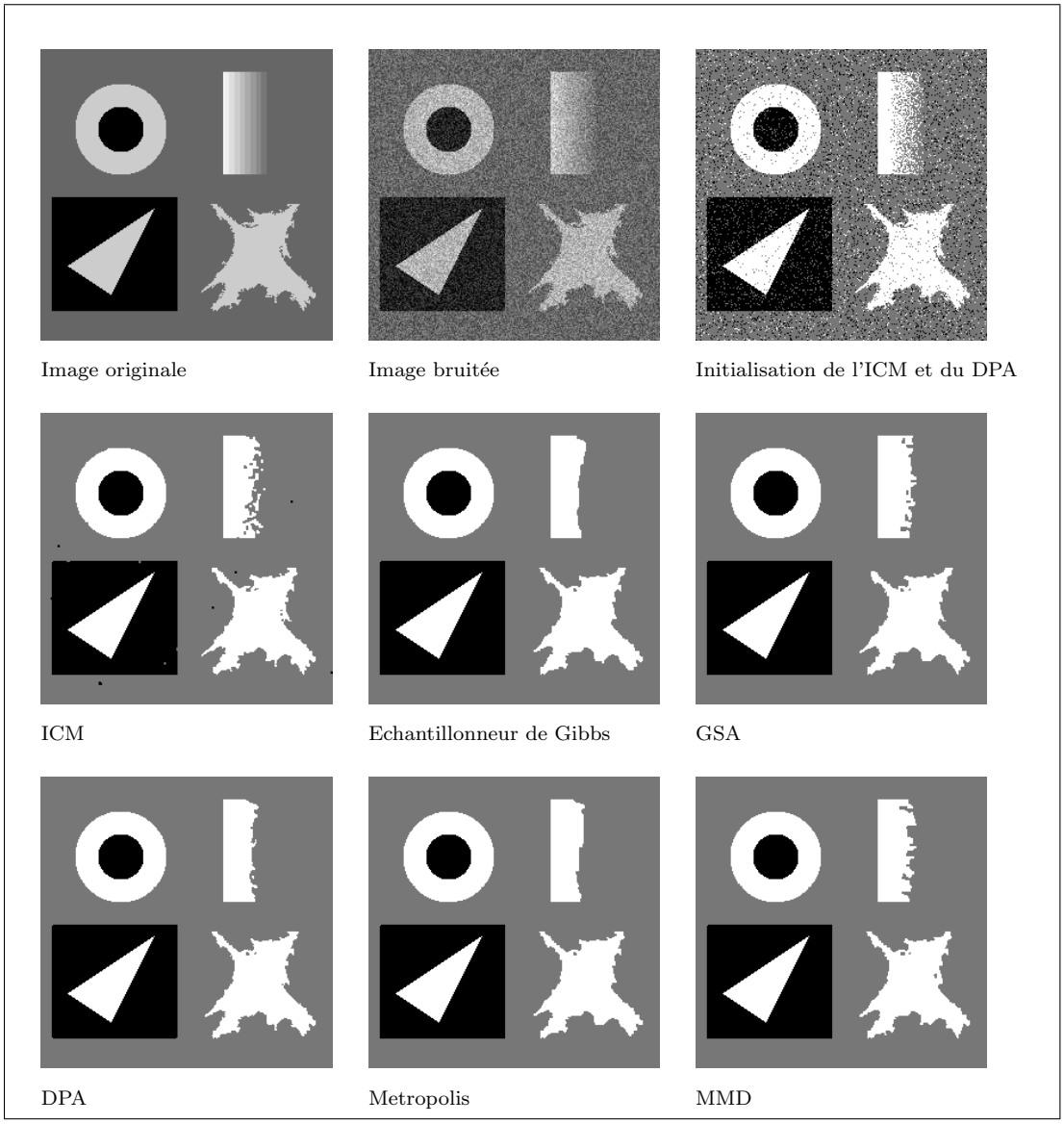

Figure 3.13: *Résultats sur l'image "bruit" avec* 3 *classes.*

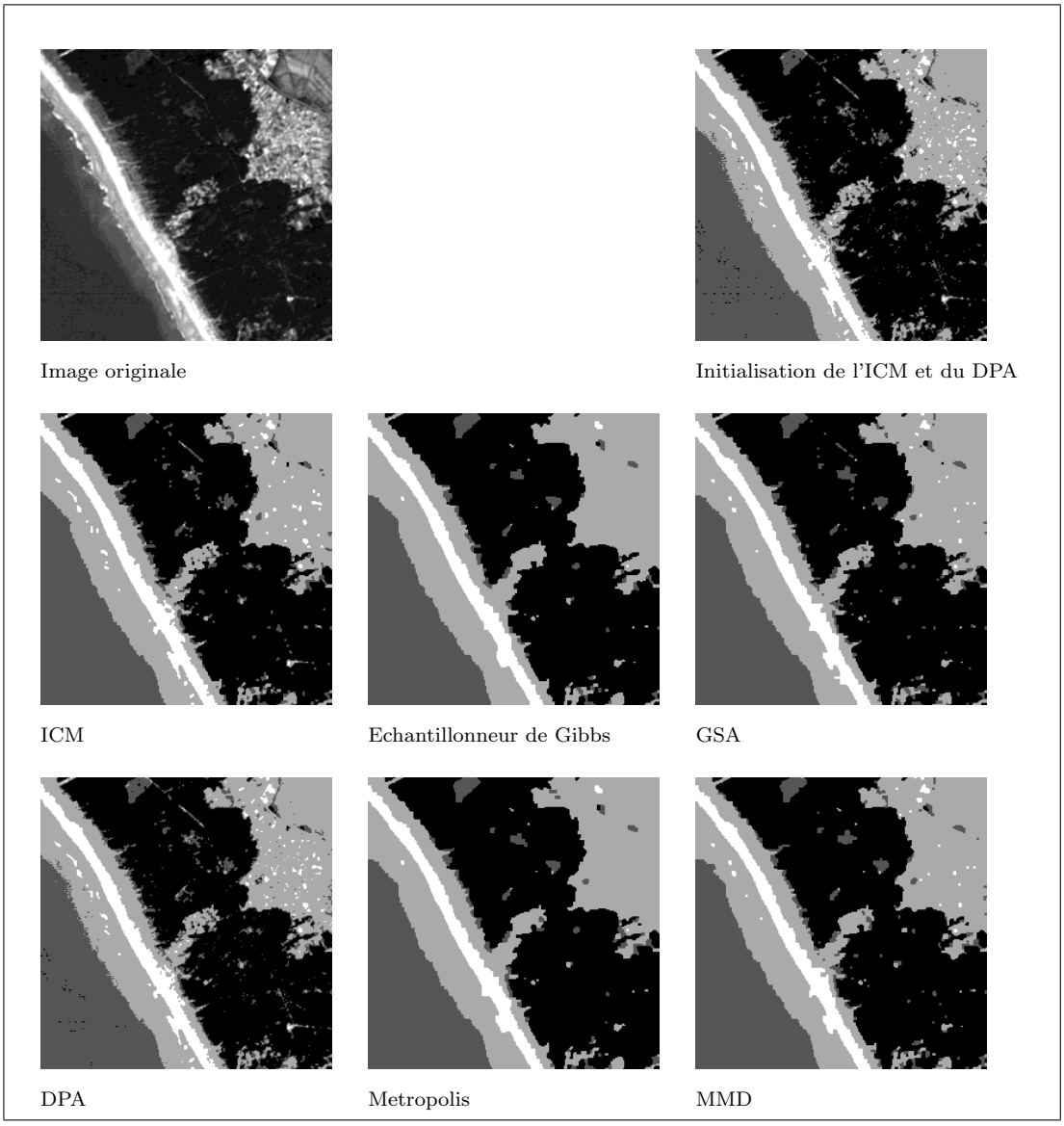

Figure 3.14: *Résultats sur l'image "SPOT" (4 classes).*

3.D Tableaux

Image	β	μ_1	σ_1^2	μ_2	σ_2^2	μ_3	σ_3^2	μ_4	σ_4^2
checkerboard	0.9	119.2	659.5	149.4	691.4	—	—	—	—
triangle	1.0	93.2	560.6	116.1	588.2	139.0	547.6	162.7	495.3
bruit	2.0	99.7	94.2	127.5	99.0	159.7	100.1	—	—
SPOT	2.0	30.3	8.2	37.4	4.6	61.3	128.1	98.2	127.1

Tableau 3.3: *Les paramètres.*

	VPR	Nb. d'Iter.	Temps total	Time/It.	Energie
ICM	2	8	0.078 sec.	0.009 sec.	52011.35
Metropolis	2	316	7.13 sec.	0.023 sec.	49447.60
Gibbs	2	322	9.38 sec.	0.029 sec.	49442.34
MMD	2	357	4.09 sec.	0.011 sec.	49459.60
GSA	2	357	7.59 sec.	0.021 sec.	49459.60
DPA	2	164	2.82 sec.	0.017 sec.	49458.02

Tableau 3.4: *Résultats sur l'image "checkerboard" avec 2 classes.*

	VPR	Nb. d'Iter.	Temps total	Temps/It.	Energie
ICM	2	9	0.146 sec.	0.016 sec.	49209.07
Metropolis	2	202	7.31 sec.	0.036 sec.	44208.56
Gibbs	2	342	14.21 sec.	0.042 sec.	44190.63
MMD	2	292	7.41 sec.	0.025 sec.	44198.31
GSA	2	191	5.44 sec.	0.028 sec.	44198.88
DPA	2	34	1.13 sec.	0.033 sec.	44237.36

Tableau 3.5: *Résultats sur l'image "triangle" avec 4 classes.*

	VPR	Nb. d'Iter.	Temps total	Temps/It.	Energie
ICM	2	8	0.302 sec.	0.037 sec.	-5552.06
Metropolis	2	287	37.33 sec.	0.130 sec.	-6896.59
Gibbs	2	301	35.76 sec.	0.118 sec.	-6903.68
MMD	2	118	10.15 sec.	0.086 sec.	-6216.50
GSA	2	242	17.84 sec.	0.073 sec.	-6256.00
DPA	8	15	1.33 sec.	0.089 sec.	-6685.52

Tableau 3.6: *Résultats sur l'image "bruit" avec 3 classes.*

	VPR	Nb. d'Iter.	Temps total	Temps/It.	Energie
ICM	8	8	0.381 sec.	0.048 sec.	-52751.71
Metropolis	8	323	42.37 sec.	0.131 sec.	-58037.59
Gibbs	8	335	46.73 sec.	0.139 sec.	-58237.32
MMD	8	125	10.94 sec.	0.087 sec.	-56156.53
GSA	8	273	23.03 sec.	0.084 sec.	-56191.61
DPA	8	15	1.78 sec.	0.119 sec.	-40647.96

Tableau 3.7: *Résultats sur l'image "SPOT" avec 4 classes.*

Dans ce chapitre:

4.

Estimation des paramètres

Dans les applications réelles, les paramètres sont souvent inconnus, il faudra les estimer [5] à partir de l'image observée. D'un point de vue statistique, ce problème est équivalent au problème de l'estimation des paramètres à partir d' un mélange de distribution. Si nous avons une réalisation du champ des étiquettes, la tâche est alors relativement facile, il existe plusieurs méthodes standards (maximum de vraisemblance, codage [8], etc. . .). Malheureusement, nous n'avons pas un tel échantillon, il est donc impossible d'utiliser directement ces méthodes. Nous devons les approximer par une fonction de l'image observée.

La plupart des méthodes utilisées sont itératives [74, 64, 15]. Pour une telle méthode, nous avons besoin d'une bonne initialisation pour chaque paramètre. Étant donné que les classes sont représentées par une distribution gaussienne, l'initialisation des valeurs moyennes et des variances des classes est très importante car ils ont une grande influence sur les étiquetages sous-jacents et donc sur le résultat final. Il existe plusieurs approches: la méthode des moments [25], la méthode de Prony [19] ou l'analyse géométrique de l'histogramme [76].

Dans ce chapitre, nous présentons des algorithmes d'estimation pour le modèle monogrille et proposons une méthode pour l'estimation des paramètres du modèle hiérarchique.

4.1 Le problème de l'estimation

Rappelons les notations définies aux Paragraphe 2.1 et Paragraphe 2.2. $\mathcal{F} = \{F_s : s \in \mathcal{S}\}$ dénote l'ensemble des données (l'image observée) sur les sites (ou pixels) $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$. A chaque site, on peut attribuer une étiquette appartenant à $\Lambda = \{0, 1, \ldots, L-1\}$. L'espace des configurations Ω est l'ensemble des étiquetage $\omega = (\omega_{s_1}, \ldots, \omega_{s_N}), \omega_s \in \Lambda$. Le processus d'étiquettes est noté par $\mathcal{X}$. De plus, nous avons n paramètres donnés par le vecteur Θ:

$$\Theta = \begin{pmatrix} \vartheta_1 \\ \vdots \\ \vartheta_n \end{pmatrix} \tag{4.1}$$

Jusqu'ici, Θ était connu et nous avons cherché l'étiquetage qui maximise la distribution a posteriori:

$$\widehat{\omega} = \arg\max_{\omega \in \Omega} P_\Theta(\omega \mid \mathcal{F}, \Theta). \tag{4.2}$$

où $\widehat{\omega}$ est l'estimateur MAP des étiquettes sachant $\mathcal{F}$, en utilisant le modèle P_Θ (pour faciliter les notations, nous omettons l'indice Θ). Si Θ ainsi que ω sont inconnus, le problème de maximisation dans l'Équation (4.2) devient [28, 58]:

$$(\widehat{\omega}, \widehat{\Theta}) = \arg\max_{\omega, \Theta} P(\omega, \mathcal{F} \mid \Theta). \tag{4.3}$$

La paire $(\widehat{\omega}, \widehat{\Theta})$ est le maximum global de la probabilité jointe $P(\omega, \mathcal{F} \mid \Theta)$. Si nous supposons que Θ est une variable aléatoire, la maximisation précédente devient un simple estimateur MAP de la manière suivante [28]: Supposons que Θ est restreint au domaine fini $\mathcal{D}_\Theta$ et que Θ est uniforme sur $\mathcal{D}_\Theta$ (c'est à dire $P(\Theta)$ est constant) [28]:

$$\begin{aligned} \arg\max_{\omega, \Theta} P(\omega, \Theta \mid \mathcal{F}) &= \arg\max_{\omega, \Theta} \frac{P(\omega, \mathcal{F} \mid \Theta) P(\Theta)}{P(\mathcal{F})} && (4.4) \\ &= \arg\max_{\omega, \Theta} \frac{P(\omega, \mathcal{F} \mid \Theta)}{\int_{\mathcal{D}_\Theta} \sum_{\omega \in \Omega} P(\omega, \mathcal{F} \mid \Theta) d\Theta} && (4.5) \\ &= \arg\max_{\omega, \Theta} P(\omega, \mathcal{F} \mid \Theta). && (4.6) \end{aligned}$$

Pratiquement, nous ne pouvons pas résoudre cette maximisation. Même le recuit simulé n'est pas utilisable car les caractéristiques locales par rapport aux paramètres Θ ne peuvent pas être calculées à partir de $P(\omega, \mathcal{F} \mid \Theta)$. Une solution possible peut être obtenu en appliquant le critère suivant [28, 58]:

$$\begin{aligned} \widehat{\omega} &= \arg\max_{\omega} P(\omega, \mathcal{F} \mid \widehat{\Theta}) && (4.7) \\ \widehat{\Theta} &= \arg\max_{\Theta} P(\widehat{\omega}, \mathcal{F} \mid \Theta) && (4.8) \end{aligned}$$

Il est clair que l'Équation (4.7) est équivalente à l'Équation (4.3) pour $\Theta = \widehat{\Theta}$ et l'Équation (4.8) est équivalente à l'Équation (4.3) avec $\omega = \widehat{\omega}$. En plus, l'Équation (4.7) est équivalente à l'estimateur MAP de ω lorsque les paramètres sont connus:

$$\arg\max_{\omega} P(\omega, \mathcal{F} \mid \widehat{\Theta}) = \arg\max_{\omega} P(\omega \mid \mathcal{F}, \widehat{\Theta}) P(\mathcal{F} \mid \widehat{\Theta}) = \arg\max_{\omega} P(\omega \mid \mathcal{F}, \widehat{\Theta}). \quad (4.9)$$

La solution du problème MAP a déjà été étudiée en détail au chapitre précédent.

D'un autre côté, la solution de l'Équation (4.8) est l'estimateur du maximum de vraisemblance (MV) des paramètres sachant l'étiquetage $\widehat{\omega}$. La solution de ce problème est relativement facile, il existe plusieurs algorithmes (pseudo-maximum de vraisemblance [5, 8, 58], codage [9]).

4.2 Le problème des données incomplètes

Dans les applications réelles, nous devons estimer les paramètres à partir des données non-étiquetées. Des méthodes variées ont été proposées pour résoudre ce problème: *Expectation – Maximization* (EM) [18], recuit simulé adaptatif [28, 58], *Iterated Conditional Estimation* (ICE) [64, 74]. Nous présentons ici les deux dernières que l'on a utilisées pour les tests.

4.2.1 Recuit simulé adaptatif

Algorithme 4.2.1 (Recuit simulé adaptatif)

① *Soit $k = 0$, initialiser $\widehat{\Theta}^0$.*

② *Faire n itérations ($n \geq 1$) de recuit simulé en utilisant $P(\omega \mid \mathcal{F}, \widehat{\Theta}^k)$. Le résultat est dénoté par $\widehat{\omega}^{k+1}$.*

③ *Mettre à jour les paramètres $\widehat{\Theta}^{k+1}$ par l'estimateur du maximum de vraisemblance ML en utilisant l'étiquetage $\widehat{\omega}^{k+1}$.*

④ *Continuer l'Étape ② en utilisant $k = k + 1$ jusqu'à ce que $\widehat{\Theta}$ se stabilise.*

Si l'estimateur du maximum de vraisemblance n'est pas calculable (c'est souvent le cas), on peut utiliser d'autres méthodes approximatives (pseudo-maximum de vraisemblance, par exemple).

4.2.2 Estimation conditionnelle itérative (ICE)

Considérons un estimateur $\mathcal{E}_\Theta(\mathcal{F}, \omega)$ de Θ (maximum de vraisemblance, par exemple). Comme les réalisations du champ des étiquettes sont inconnues, nous ne pouvons pas directement appliquer $\mathcal{E}_\Theta(\mathcal{F}, \omega)$, nous devons donc l'approximer. La meilleure approximation, au sens des moindres carrés, est *l'espérance conditionnelle*. Puisque $E\{\mathcal{E}_\Theta \mid \mathcal{F}, \omega\}$ dépend des paramètres Θ, nous avons besoin d'une estimation $\widehat{\Theta}^k$ définie a priori. Ceci nous donne un algorithme itératif [74, 14, 75]:

Algorithme 4.2.2 (ICE)

① *Soit $k = 0$, initialiser $\widehat{\Theta}^0$.*

② *Générer n réalisations (n est fixé a priori) $\widehat{\omega}^i (1 \leq i \leq n)$ du champ des étiquettes en utilisant $\widehat{\Theta}^k$.*

③ *A partir de $\widehat{\omega}^i (1 \leq i \leq n)$, $\widehat{\Theta}^{k+1}$ est obtenu par l'espérance conditionnelle:*

$$\widehat{\Theta}^{k+1} = E\{\mathcal{E}_\Theta \mid \mathcal{X} = \omega\} \approx \frac{1}{n} \sum_{i=1}^{n} \mathcal{E}_\Theta(\mathcal{F}, \widehat{\omega}^i). \tag{4.10}$$

④ *Continuer l'Étape ② jusqu'à ce que $\widehat{\Theta}$ se stabilise.*

4.3 Détermination des modes d'un mélange de gaussiennes

Dans ce paragraphe, nous présentons une méthode non-itérative pour déterminer les modes d'un mélange de gaussiennes qui utilise l'analyse géométrique des domaines concaves du mélange [76].

Un domaine concave d'un mélange de gaussiennes $f(x)$ est déterminé de la façon suivante: à chaque point x, une famille de domaines $D_i(x)$ centrées autour de x est associée (par exemple une séquence de plus en plus grande d'hypercubes). La valeur moyenne de $f(x)$ dans le domaine $D_i(x)$ est définie par:

$$E\{D_i(x)\} = \frac{\int_{D_i(x)} f(\xi) d\xi}{\int_{D_i(x)} d\xi}. \tag{4.11}$$

Le test de convexité est basé sur le fait que $E\{D_i(x)\}$ est une fonction décroissante de i pour n'importe quelle famille de domaines $D_i(x)$ qui sont dans un région concave de $f(x)$.

En traitement d'image, nous regardons l'histogramme comme un mélange de gaussiennes. Notons les valeurs de l'histogramme par $h_0, h_1, \ldots, h_G$ ($h_x \in [0,1]$), où G est le nombre des niveaux de gris possibles. Le domaine $D_i(x)$ est simplement un segment de droite centré en x. L'algorithme est le suivant:

Algorithme 4.3.1 (Identification d'un mélange de gaussiennes)

① *Définissons en chaque point $x \in [0, G]$ un petit voisinage D_1 et un voisinage D_2 plus grand. Calculer $E\{D_1(x)\}$ et $E\{D_2(x)\}$ par l'approximation de Équation (4.11):*

$$E\{D_i(x)\} \approx \frac{\sum_{\xi \in D_i(x)} h_\xi}{l(D_i(x))} \tag{4.12}$$

où $l(D_i(x))$ est la longeur du segment de droite D_i. Calculer $\delta(x) = E\{D_2(x)\} - E\{D_1(x)\}$ à chaque x. Si $\delta(x) \leq 0$, alors l'histogramme est concave dans le domaine correspondant $D_1(x)$.

② *Ensuite, nous devons cumuler tous les domaines sous-jacents où l'histogramme est concave. Les domaines obtenus $\widehat{D}_k (k = 1, \ldots, L)$ donnent les modes du mélange et L est le nombre des modes ou classes.*

③ *La moyenne μ_k du mode k est le centre du domaine $\widehat{D}_k$ et σ_k égale à la longeur de $\widehat{D}_k$. En représentant les domaines concaves $\widehat{D}_k$ par (x_1^k, x_2^k), on a:*

$$\mu_k = \frac{x_2^k - x_1^k}{2} \quad \text{et } \sigma_k = x_2^k - x_1^k. \tag{4.13}$$

4.4 Segmentation non-supervisée d'images

Dans ce paragraphe, nous établissons un modèle mono-grille de segmentation non-supervisée [51]. Pour cela, nous utilisons le modèle markovien d'ordre 1 proposé dans le Paragraphe 2.2, où les classes sont représentées par des distributions gaussiennes:

$$P(\widehat{\omega}, \mathcal{F} \mid \Theta) \;=\; \prod_{s \in \mathcal{S}} \frac{1}{\sqrt{2\pi}\sigma_{\widehat{\omega}_s}} \exp\left(-\frac{(f_s - \mu_{\widehat{\omega}_s})^2}{2\sigma^2_{\widehat{\omega}_s}} \right)$$

$$\cdot \frac{\exp(-2\beta \sum_{\{s,r\}\in\mathcal{C}} \delta(\widehat{\omega}_s, \widehat{\omega}_r))}{Z(\beta)} \tag{4.14}$$

$$\text{avec } Z(\beta) = \sum_{\omega\in\Omega} \exp\left(-2\beta \sum_{\{s,r\}\in\mathcal{C}} \delta(\omega_s, \omega_r)\right) \tag{4.15}$$

$$\text{et } \delta(\widehat{\omega}_s, \widehat{\omega}_r) = \begin{cases} 0 & \text{si } \widehat{\omega}_s = \widehat{\omega}_r \\ 1 & \text{sinon} \end{cases} \tag{4.16}$$

Nous avons $2L + 1$ paramètres (deux pour chaque classe et un hyperparamètre β):

$$\Theta = \begin{pmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_{L-1} \\ \sigma_0 \\ \vdots \\ \sigma_{L-1} \\ \beta \end{pmatrix} \tag{4.17}$$

Les premiers $2L$ paramètres sont estimés à partir du terme gaussien et β est calculé à partir du terme markovien. Pour l'estimation, nous utilisons la fonction de vraisemblance suivante:

$$\begin{aligned} \ln(L(\Theta)) &= \sum_{s\in\mathcal{S}} \left(-\ln(\sqrt{2\pi}\sigma_{\widehat{\omega}_s}) - \frac{(f_s - \mu_{\widehat{\omega}_s})^2}{2\sigma^2_{\widehat{\omega}_s}}\right) \\ &\quad -2\beta \sum_{\{s,r\}\in\mathcal{C}} \delta(\widehat{\omega}_s, \widehat{\omega}_r) - \ln(Z(\beta)) \qquad (4.18) \\ &= \sum_{\lambda\in\Lambda} \underbrace{\sum_{s\in\mathcal{S}_\lambda} \left(-\ln(\sqrt{2\pi}\sigma_\lambda) - \frac{(f_s - \mu_\lambda)^2}{2\sigma^2_\lambda}\right)}_{\mathcal{G}(\mu_\lambda, \sigma_\lambda)} \\ &\quad \underbrace{-2\beta \sum_{\{s,r\}\in\mathcal{C}} \delta(\widehat{\omega}_s, \widehat{\omega}_r) - \ln(Z(\beta))}_{\mathcal{M}(\beta)} \qquad (4.19) \end{aligned}$$

où $\mathcal{S}_\lambda$ est l'ensemble des pixels où $\widehat{\omega} = \lambda$. Nous cherchons le vecteur $\widehat{\Theta}$ qui maximise la fonction de vraisemblance:

$$\forall \lambda \in \Lambda: \quad \frac{\partial \mathcal{G}(\mu_\lambda, \sigma_\lambda)}{\partial \mu_\lambda} = 0 \tag{4.20}$$

$$\frac{\partial \mathcal{G}(\mu_\lambda, \sigma_\lambda)}{\partial \sigma_\lambda} = 0 \tag{4.21}$$

$$\text{et } \frac{\partial \mathcal{M}(\beta)}{\partial \beta} = 0 \tag{4.22}$$

La solution pour μ_λ et σ_λ est simplement la moyenne empirique et la variance empirique:

$$\begin{aligned} \forall \lambda \in \Lambda: \quad \mu_\lambda &= \frac{1}{\mid \mathcal{S}_\lambda \mid} \sum_{s \in \mathcal{S}_\lambda} f_s, \\ \sigma_\lambda^2 &= \frac{1}{\mid \mathcal{S}_\lambda \mid} \sum_{s \in \mathcal{S}_\lambda} (f_s - \mu_\lambda)^2. \end{aligned} \tag{4.23}$$

Pour β, nous examinons la dérivée de $\mathcal{M}(\beta)$:

$$\frac{\partial}{\partial \beta} \left(-2\beta N^{ih}(\widehat{\omega}) - \ln \left(\sum_{\omega \in \Omega} \exp(-2\beta N^{ih}(\omega)) \right) \right)$$

$$= -N^{ih}(\widehat{\omega}) + \frac{\sum_{\omega \in \Omega} N^{ih}(\omega) \exp(-2\beta N^{ih}(\omega))}{\sum_{\omega \in \Omega} \exp(-2\beta N^{ih}(\omega))} = 0 \tag{4.24}$$

où $N^{ih}(\widehat{\omega}) = \sum_{\{s,r\} \in \mathcal{C}} \delta(\widehat{\omega}_s, \widehat{\omega}_r)$ est le nombre des cliques inhomogènes dans $\widehat{\omega}$. Nous obtenons (Équation (4.24)):

$$N^{ih}(\widehat{\omega}) = \frac{\sum_{\omega \in \Omega} N^{ih}(\omega) \exp(-2\beta N^{ih}(\omega))}{\sum_{\omega \in \Omega} \exp(-2\beta N^{ih}(\omega))} \tag{4.25}$$

Puisque $\ln(Z(\beta))$ est *convexe* en Θ [6, 28], le gradient peut être approximé par relaxation stochastique [28]:

Algorithme 4.4.1 (Estimation de l'hyperparamètre)

① *Soit $k = 0$, initialiser $\widehat{\beta}^0$ et soit $N^{ih}(\widehat{\omega})$ le nombre des cliques inhomogènes dans l'estimateur de l'étiquetage.*

② *En utilisant le recuit simulé pour une température fixée T, générer un nouvel étiquetage η avec la distribution suivante:*

$$P(\mathcal{X} = \omega) = \frac{\exp\left(-\frac{\widehat{\beta}^k}{T} \sum_{\{s,r\} \in \mathcal{S}} \delta(\omega_s, \omega_r)\right)}{Z(\widehat{\beta}^k)}. \tag{4.26}$$

Calculer le nombre des cliques inhomogènes $N^{ih}(\eta)$ en η.

③ *Si $N^{ih}(\eta) \approx N^{ih}(\widehat{\omega})$ arrêter l'exécution, sinon $k = k + 1$ et et faire decroître $\widehat{\beta}^k$ si $N^{ih}(\eta) < N^{ih}(\widehat{\omega})$ ou l'augmenter si $N^{ih}(\eta) > N^{ih}(\widehat{\omega})$, puis recommencer à l'Étape ②.*

L'algorithme utilisé pour les test est le suivant:

Algorithme 4.4.2 (Segmentation non-supervisée)

① *Étant donné une image $\mathcal{F}$, calculer son histogramme et pour chaque $\lambda \in \Lambda$, initialiser μ_λ et σ_λ par l'Algorithme 4.3.1. β est initialisé aléatoirement entre 0 et 1.*

② **(Estimation)** *Trouver l'estimateur $\widehat{\Theta}$ des paramètres par l'Algorithme 4.2.2 (ICE).*

③ **(Segmentation)** *Exécuter une segmentation supervisée avec les paramètres $\widehat{\Theta}$, ce qui nous donne l'estimateur MAP du champ des étiquettes sachant $\mathcal{F}$ et $\widehat{\Theta}$.*

4.4.1 Estimation des paramètres du modèle hiérarchique

Nous utilisons le modèle proposé dans le Paragraphe 2.4.4. La fonction de vraisemblance est la suivante:

$$\sum_{i=0}^{M} \sum_{s^i \in \mathcal{S}^i} \sum_{s \in b^i_{s^i}} \left(-\ln(\sqrt{2\pi}\sigma_{\omega_s}) - \frac{(f_s - \mu_{\omega_s})^2}{2\sigma^2_{\omega_s}} \right)$$

$$-2\beta \underbrace{\sum_{i=0}^{M} q^i \sum_{C^i \in \mathcal{C}^i} \delta(\widehat{\omega}_{C^i})}_{N^{ih}(\widehat{\omega})} - 2\gamma \underbrace{\sum_{C \in \vec{\mathcal{C}}_3} \delta(\widehat{\omega}_C)}_{\bar{N}^{ih}(\widehat{\omega})} - \ln(Z(\beta,\gamma)) \tag{4.27}$$

où q^i est le nombre des cliques entre deux blocs voisins sur l'échelle $\mathcal{B}^i$ (pour plus de détails, voir Paragraphe 2.3.3). $N^{ih}(\widehat{\omega})$ dénote le nombre des cliques inhomogènes qui sont sur la même couche et $\bar{N}^{ih}(\widehat{\omega})$ dénote le nombre des cliques inhomogènes qui sont entre deux couches voisines. Tous d'abord, considérons le premier terme:

$$\begin{aligned}
&\sum_{i=0}^{M} \sum_{s^i \in \mathcal{S}^i} \sum_{s \in b^i_{s^i}} \left(-\ln(\sqrt{2\pi}\sigma_{\omega_s}) - \frac{(f_s - \mu_{\omega_s})^2}{2\sigma^2_{\omega_s}} \right) \\
&= \sum_{\lambda \in \Lambda} \sum_{i=0}^{M} \sum_{s^i \in \mathcal{S}^i_\lambda} \sum_{s \in b^i_{s^i}} \left(-\ln(\sqrt{2\pi}\sigma_\lambda) - \frac{(f_s - \mu_\lambda)^2}{2\sigma^2_\lambda} \right)
\end{aligned} \tag{4.28}$$

où $\mathcal{S}^i_\lambda$ dénote l'ensemble des sites sur le niveau i où $\widehat{\omega}_{s^i} = \lambda$. Par dérivation, nous obtenons:

$$\begin{aligned}
\forall \lambda \in \Lambda: \quad \mu_\lambda &= \frac{1}{\sum_{i=0}^{M} \mid \mathcal{S}^i_\lambda \mid} \sum_{i=0}^{M} \sum_{s^i \in \mathcal{S}^i_\lambda} \sum_{s \in b^i_{s^i}} f_s \\
\sigma^2_\lambda &= \frac{1}{\sum_{i=0}^{M} \mid \mathcal{S}^i_\lambda \mid} \sum_{i=0}^{M} \sum_{s^i \in \mathcal{S}^i_\lambda} \sum_{s \in b^i_{s^i}} (f_s - \mu_\lambda)^2
\end{aligned} \tag{4.29}$$

Remarquons que le niveau de gris f_s peut être pris en compte plusieurs fois. Plus exactement, f_s est considéré m fois pour un λ donné s'il existe m échelles où $\widehat{\omega}$ associe l'étiquette λ au site s. La dérivée de la fonction de vraisemblance par rapport à β et γ est donnée par:

$$\frac{\partial}{\partial\beta}\Big(-2\beta N^{ih}(\widehat{\omega}) - \ln(Z(\beta,\gamma))\Big) = -N^{ih}(\widehat{\omega}) - \frac{\partial}{\partial\beta}\ln(Z(\beta,\gamma)) \tag{4.30}$$

$$\frac{\partial}{\partial\gamma}\Big(-2\gamma \bar{N}^{ih}(\widehat{\omega}) - \ln(Z(\beta,\gamma))\Big) = -\bar{N}^{ih}(\widehat{\omega}) - \frac{\partial}{\partial\gamma}\ln(Z(\beta,\gamma)) \tag{4.31}$$

D'où on a:

$$N^{ih}(\widehat{\omega}) = \frac{\sum_{\omega\in\Omega} N^{ih}(\omega)\exp(-2\beta N^{ih}(\omega) - 2\gamma \bar{N}^{ih}(\omega))}{\sum_{\omega\in\Omega}\exp(-2\beta N^{ih}(\omega) - 2\gamma \bar{N}^{ih}(\omega))} \tag{4.32}$$

$$\bar{N}^{ih}(\widehat{\omega}) = \frac{\sum_{\omega\in\Omega} \bar{N}^{ih}(\omega)\exp(-2\beta N^{ih}(\omega) - 2\gamma \bar{N}^{ih}(\omega))}{\sum_{\omega\in\Omega}\exp(-2\beta N^{ih}(\omega) - 2\gamma \bar{N}^{ih}(\omega))} \tag{4.33}$$

La solution de ces équations, comme dans le cas mono-grille, est obtenue par l'algorithme suivant:

Algorithme 4.4.3 (Estimation des hyperparamètres hiérarchiques)

① *Soit $k = 0$, initialiser $\widehat{\beta}^0$ et $\widehat{\gamma}^0$. Soient $N^{ih}(\widehat{\omega})$ le nombre des cliques inhomogènes sur la même échelle et $\bar{N}^{ih}(\widehat{\omega})$ le nombre des cliques inhomogènes entre les niveaux.*

② *En utilisant le recuit simulé pour une température fixée T, générer un nouvel étiquetage η avec la distribution suivante:*

$$P(\mathcal{X} = \omega) = \frac{\exp\left(-\frac{\widehat{\beta}^k}{T}\sum_{i=0}^{M}\sum_{\{s,r\}\in\mathcal{C}^i}\delta(\omega_s,\omega_r) + \frac{\widehat{\gamma}^k}{T}\sum_{\{s,r\}\in\bar{\mathcal{C}}}\delta(\omega_s,\omega_r)\right)}{Z(\widehat{\beta}^k,\widehat{\gamma}^k)}. \quad (4.34)$$

Calculer le nombre des cliques inhomogènes $N^{ih}(\eta)$ et $\bar{N}^{ih}(\eta)$ dans η.

③ *Si $N^{ih}(\eta) \approx N^{ih}(\widehat{\omega})$ et $\bar{N}^{ih}(\eta) \approx \bar{N}^{ih}(\widehat{\omega})$ arrêter l'exécution, sinon $k = k + 1$ et faire décroître $\widehat{\beta}^k$ si $N^{ih}(\eta) < N^{ih}(\widehat{\omega})$ ou l'augmenter si $N^{ih}(\eta) > N^{ih}(\widehat{\omega})$. $\widehat{\gamma}^k$ est obtenu de la même façon. Recommencer avec l'Étape ② en utilisant $(\widehat{\beta}^k,\widehat{\gamma}^k)$.*

L'algorithme de segmentation est le suivant:

Algorithme 4.4.4 (Segmentation hiérarchique non-supervisée)

① *Étant donné une image $\mathcal{F}$, calculer son histogramme et pour chaque $\lambda \in \Lambda$, initialiser μ_λ et σ_λ par l'Algorithme 4.3.1. β et γ sont initialisés aléatoirement.*

② **(Estimation)** *L'estimateur $\widehat{\Theta}$ des paramètres est obtenu par l'Algorithme 4.2.1.*

③ **(Segmentation)** *Exécuter une segmentation supervisée avec les paramètres $\widehat{\Theta}$, ce qui nous donne l'estimateur MAP du champ des étiquettes sachant $\mathcal{F}$ et $\widehat{\Theta}$.*

4.5 Résultats expérimentaux

Nous avons testé les algorithmes proposés sur des images synthétiques bruitées et réelles. Les algorithmes ont été mis en œuvre sur une machine à connexions CM200 [39]. Nous avons comparé les paramètres et les résultats obtenus aux résultats supervisés présentés dans le Chapitre 2. En général, la qualité des résultats non-supervisés est aussi bonne et parfois meilleure que ceux des algorithmes supervisés. Cependant, nous avons observé que les algorithmes non-supervisés sont plus sensibles au bruit à cause de l'initialisation des paramètres, en particulier la moyenne et la variance des classes.

Le seul paramètre que nous avons supposé être connu est le nombre des classes. Tous les autres paramètres ont été estimés automatiquement à partir des données. Nous avons utilisé l'Algorithme 4.4.2: Des valeurs initiales des hypermaramètres sont les suivantes: $\beta = 0.7$ et $\gamma = 0.1$. Les expériences montrent que l'initialisation de ces paramètres n'est pas très importante. Pratiquement, n'importe quelle valeur entre 0.5 et 1 pour β et une valeur proche de zéro pour γ peuvent être admises.

Dans l'étape suivante(Étape ② de l'Algorithme 4.4.2), nous utilisons l'algorithme ICE (Algorithme 4.2.2) pour obtenir les estimations finales. Pour générer les étiquetages nécessaires, nous avons choisi l'ICM car il est très rapide. Connaissant les paramètres $\widehat{\Theta}^n$, l'ICM est utilisé pour maximiser la probabilité a posteriori de ω. Supposons que l'ICM converge en N itérations (en pratique $N < 10$), ce qui nous donne N étiquetage $\omega_i (0 < i < N)$. Pour chaque ω_i, on calcule les paramètres par le maximum de vraisemblance.

Pour le modèle hiérarchique, nous avons utilisé le recuit simulé adaptatif car l'ICE a été trop lent. Une autre modification est que les paramètres gaussiens ont été calculer uniquement sur le plus bas niveau et pas sur la pyramide entière car les variances obtenues ont été trop importantes avec l'algorithme original. Cette modification réduit également le temps de calcul.

Les algorithmes ont été testés sur l'images "checkerboard" (Figure 4.1), "triangle" (Figure 4.2) et "holland" (Figure 4.3–Figure 4.5). Pour les images synthétiques, nous donnons également l'histogramme qui détermine l'initialisation. (l'histogramme de l'image "holland" se trouve dans la Figure 2.13). Dans Tableau 4.2, Tableau 4.4 et Tableau 4.6, nous comparons les paramètres obtenus par les algorithmes non-supervisés à ceux utilisés pour les segmentations supervisées. Nous remarquons que les paramètres supervisés ne sont pas forcement corrects. Ils ont été calculés par l'algorithme décrit dans le Paragraphe 2.2, à partir des ensembles d'apprentissage choisis par un expert (cf. Figure 4.3). Dans Tableau 4.3, Tableau 4.5 et Tableau 4.7, nous donnons le temps de calcul de l'estimation et de la segmentation. Nous pouvons constater que l'estimation exige beaucoup plus de temps que la segmentation. La plus grand partie du temps de calcul est prise par l'estimation des hyperparamètres qui utilise le recuit simulé.

Le Tableau 4.1 donne une comparaison des résultats supervisés et non-supervisés par le nombre des pixels mal-classifiés. Les performances sont pratiquement les mêmes pour la segmentation supervisée et non-supervisée.

En conclusion, les algorithmes proposés donnent des résultats comparables à ceux obtenus par les algorithmes supervisés mais ils exigent un temps de calcul plus élevée et ils sont plus sensibles au bruit. L'avantage principal est que les méthodes non-supervisées sont autonomes, le seul paramètre à préciser est le nombre des classes.

Annexe

4.A Images

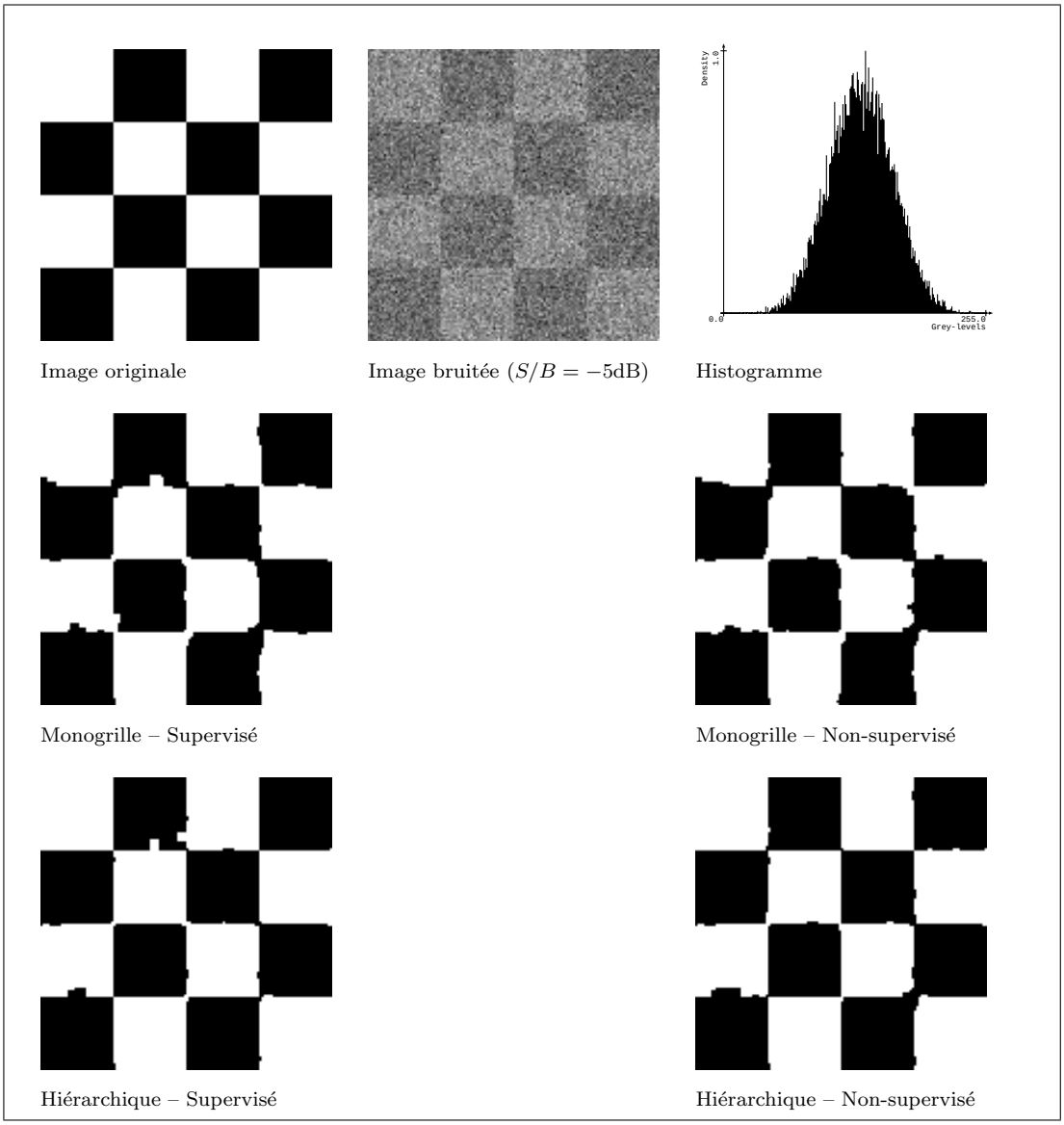

Figure 4.1: *Résultats de segmentation supervisée et non-supervisée sur l'image "checkerboard" avec* 2 *classes.*

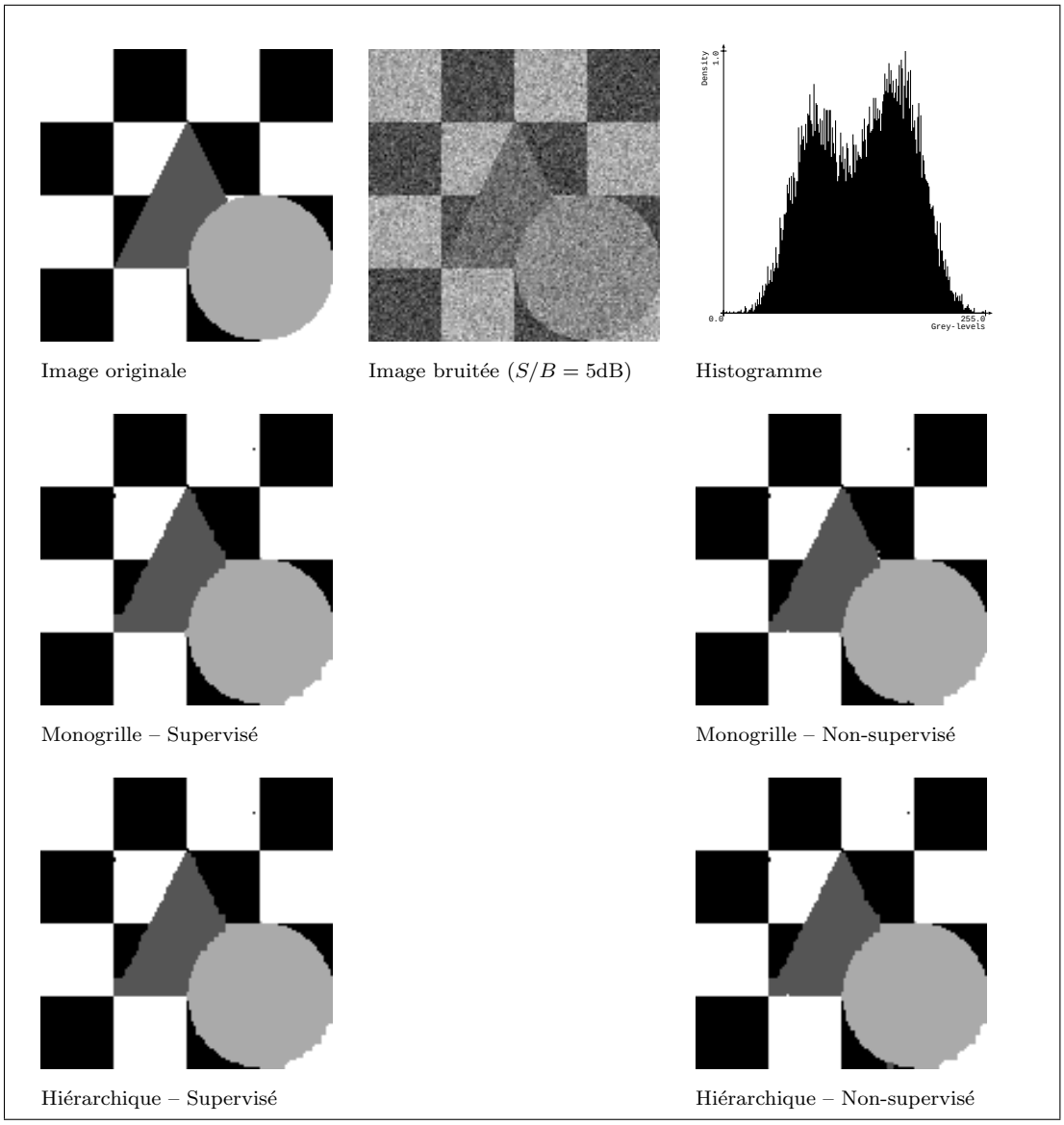

Figure 4.2: *Résultats de segmentation supervisée et non-supervisée sur l'image "triangle" avec* 4 *classes.*

Figure 4.3: *Les ensembles d'apprentissage sur l'image "holland".*

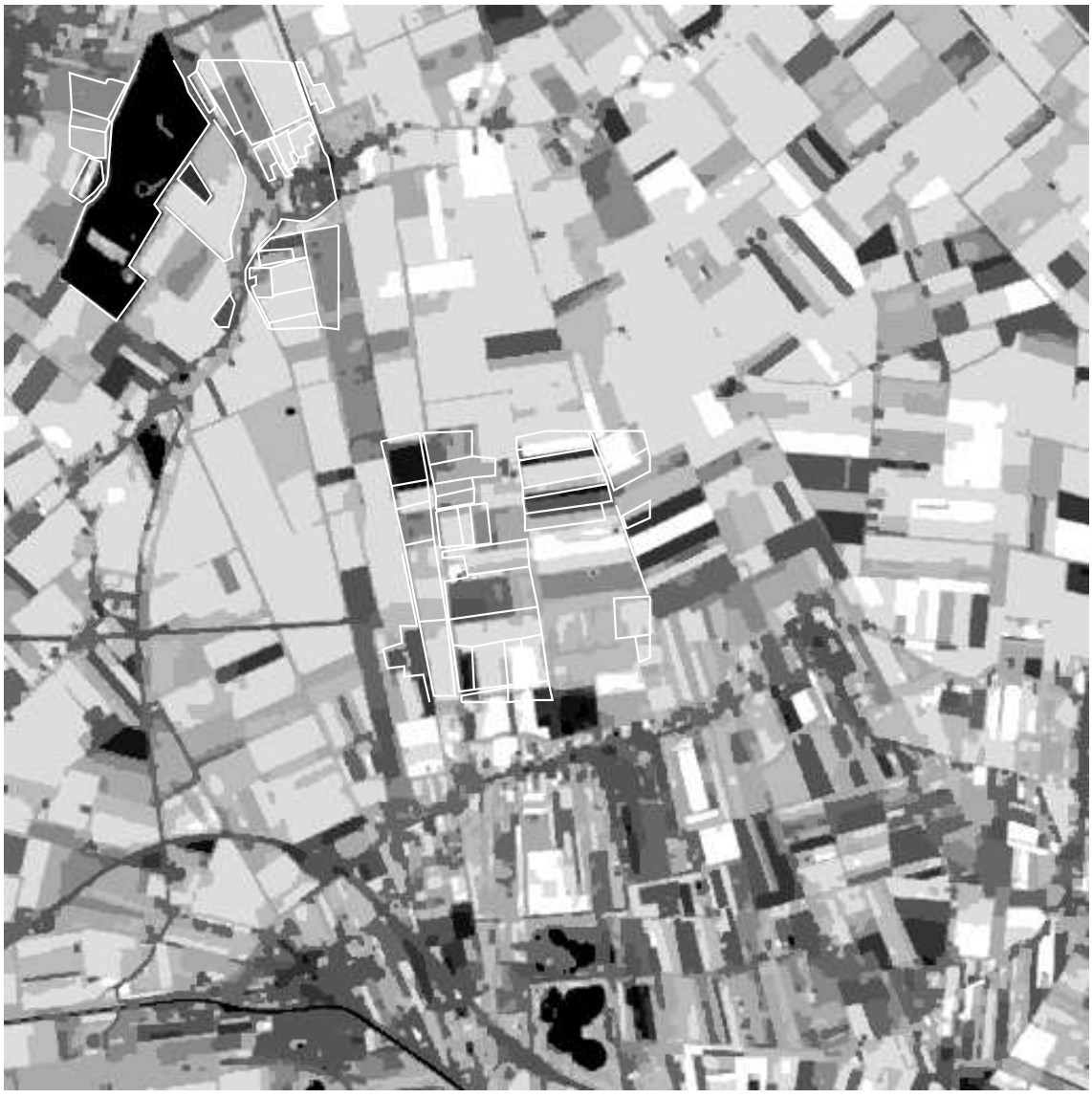

Figure 4.4: *Résultats de segmentation supervisée avec* 10 *classes (Échantilloneur de Gibbs).*

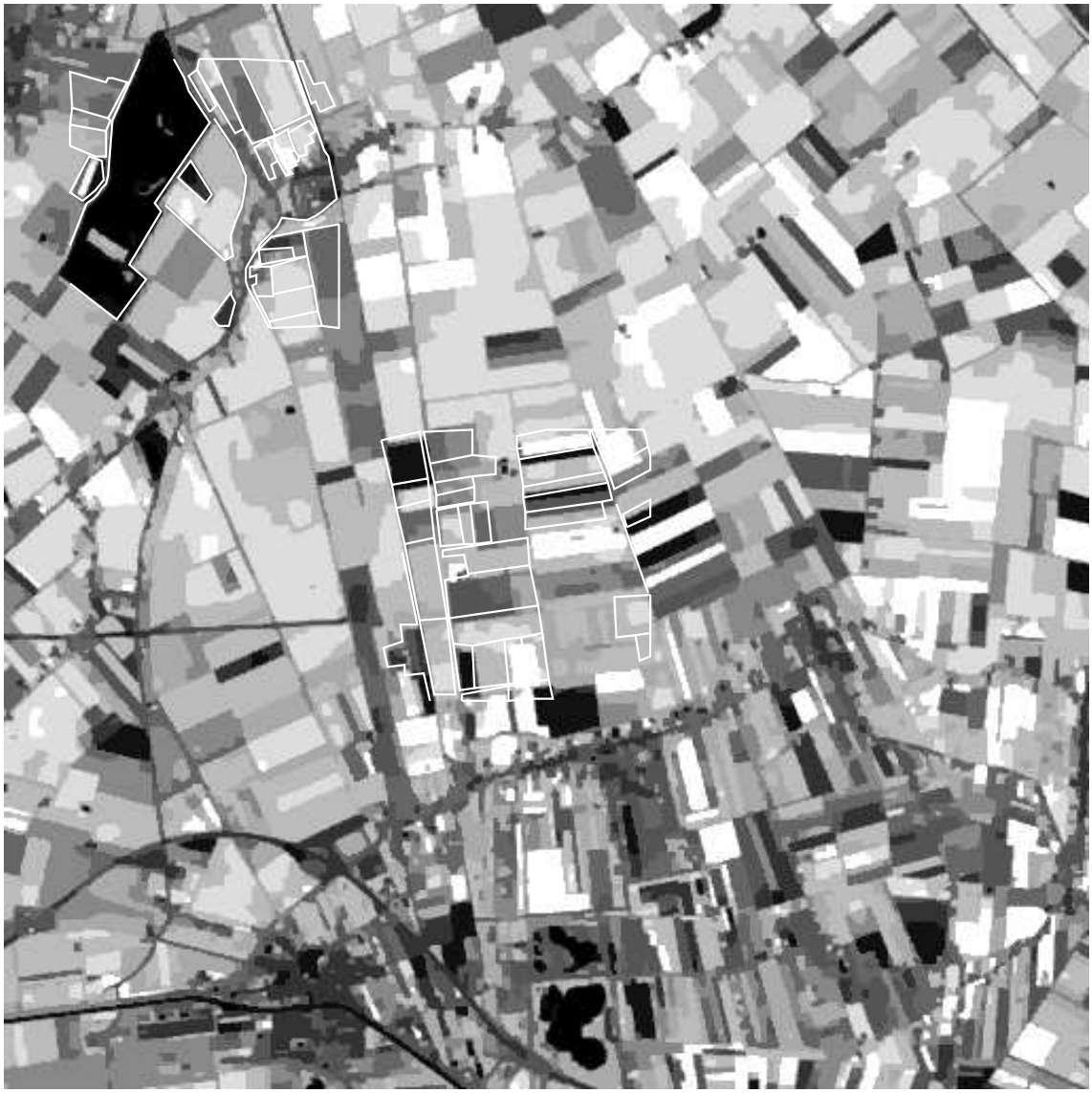

Figure 4.5: *Résultats de segmentation non-supervisée avec* 10 *classes (Échantilloneur de Gibbs).*

4.B Tableaux

Modèle	Image	Supervisé	Non-supervisé
Monogrille	checkerboard	260 (1.59%)	213 (1.41%)
	triangle	112 (0.68%)	103 (0.63%)
Hiérarchique	checkerboard	115 (0.7%)	147 (0.9%)
	triangle	104 (0.63%)	111 (0.68%)

Tableau 4.1: *Résultats supervisés et non-supervisés.*

Modèle monogrille			
	Non-supervisé		
Paramètre	Initial	Final	Supervisé
μ_0	123.5	117.3	119.2
σ_0^2	256.0	680.0	659.5
μ_1	170.0	151.5	149.4
σ_1^2	169.0	668.2	691.4
β	0.7	0.7	0.9

Modèle hiérarchique			
	Non-supervisé		
Paramètre	Initial	Final	Supervisé
μ_0	123.5	126.7	119.2
σ_0^2	256.0	903.4	659.5
μ_1	170.0	151.5	149.4
σ_1^2	169.0	689.3	691.4
β	0.7	0.7	0.7
γ	0.1	0.1	0.3

Tableau 4.2: *Les paramètres de l'image "checkerboard".*

Modèle	VPR	Temps total	Estimation	Segmentation
Monogrille	2	142.73 sec.	133.57 sec.	9.16 sec.
Hiérarchique	4	1551.93 sec.	1042.46 sec.	446.52 sec.

Tableau 4.3: *Temps de l'exécution sur l'image "checkerboard".*

Modèle monogrille			
	Non-supervisé		
Paramètre	Initial	Final	Supervisé
μ_0	83.5	84.3	85.48
σ_0^2	256.0	480.5	446.60
μ_1	100.0	117.3	115.60
σ_1^2	169.0	416.3	533.97
μ_2	152.5	148.1	146.11
σ_2^2	676.0	457.8	540.32
μ_3	181.5	178.5	178.01
σ_3^2	100.0	490.9	504.34
β	0.7	1.0	1.0

Modèle hiérarchique			
	Non-supervisé		
Paramètre	Initial	Final	Supervisé
μ_0	83.5	84.3	85.48
σ_0^2	256.0	483.9	446.60
μ_1	100.0	115.5	115.60
σ_1^2	169.0	444.6	533.97
μ_2	152.5	146.7	146.11
σ_2^2	676.0	502.1	540.32
μ_3	181.5	177.9	178.01
σ_3^2	100.0	500.0	504.34
β	0.7	1.0	0.7
γ	0.1	0.1	0.1

Tableau 4.4: *Les paramètres de l'image "triangle".*

Modèle	VPR	Temps total	Estimation	Segmentation
Monogrille	2	249.75 sec.	237.00 sec.	12.75 sec.
Hiérarchique	4	1762.23 sec.	1232.82 sec.	529.41 sec.

Tableau 4.5: *Temps de l'exécution sur l'image "triangle".*

Paramètre	Non-supervisé		Supervisé
	Initial	Final	
μ_0	51.5	53.1	54.6
σ_0^2	36.0	10.3	93.1
μ_1	60.0	77.2	73.5
σ_1^2	49.0	64.3	4.1
μ_2	70.5	89.6	82.5
σ_2^2	49.0	30.7	35.5
μ_3	80.5	102.5	93.8
σ_3^2	64.0	35.7	93.7
μ_4	97.5	116.2	100.5
σ_4^2	441.0	27.6	308.8
μ_5	122.5	127.2	122.8
σ_5^2	484.0	18.9	8.9
μ_6	136.0	138.6	129.9
σ_6^2	1.0	20.2	37.4
μ_7	152.5	152.7	146.6
σ_7^2	625.0	18.0	15.3
μ_8	169.0	162.4	159.9
σ_8^2	1.0	7.4	31.3
μ_9	181.5	174.2	182.3
σ_9^2	25.0	54.1	73.1
β	0.7	1.3	1.0

Tableau 4.6: *Les paramètres de l'image "holland".*

Modèle	VPR	Temps total	Estimation	Segmentation
Monogrille	32	3576.58 sec.	3270.78 sec.	305.81 sec.

Tableau 4.7: *Temps d'exécution sur l'image "holland".*

Conclusion

Nous avons discuté les trois étapes principales du traitement statistique d'images: la modélisation, l'optimisation et l'estimation des paramètres. Nous avons étudié les problèmes de la vision pré-attentive dans un cadre général appelé étiquetage d'images. Notre approche est probabiliste, nous utilisons les champs de Markov et l'estimation bayesienne, en particulier l'estimation par le Maximum A Posteriori (MAP). L'avantage d'une telle modélisation est que l'information a priori peut être "codée" $localement$ par les potentiels des cliques. Nous avons développé les modèles markoviens pyramidaux qui réduisent le temps de calcul et améliorent la qualité du résultat final. Nous avons proposé des méthodes pour l'estimation des paramètres des modèles hiérarchiques et mono-grilles. Les premiers résultats sont satisfaisants mais il reste beaucoup de travail à faire.

Tous les modèles markoviens exigent la minimisation d'une fonction d'énergie non-convexe qui est résolue par recuit simulé ou par relaxation déterministe. Nous avons présenté également les méthodes de parallélisation de ces algorithmes.

Notre résultat principal est un modèle markovien hiérarchique et un algorithme de recuit multi-température proposé pour la minimisation de l'énergie du modèle hiérarchique. La convergence de cet algorithme vers un optimum $global$ a été prouvé dans le cas général où chaque clique a sa propre loi de température $locale$. Il reste quelques problèmes ouverts comme la relation entre les estimateurs MAP du modèle mono-grille et celui du modèle hiérarchique. la mise en œuvre du modèle hiérarchique sur un ordinateur pyramidal et l'implantation d'une méthode beaucoup plus rapide pour l'estimation des paramètres.

Sommaire

Nous avons étudié les trois étapes du traitement d'images bas niveau:

- Modélisation
- Optimisation
- Estimation des paramètres

Pour la *modélisation*, nous avons étudié les différents problèmes du traitement d'images dans un cadre commun appelé étiquetage d'images. Nous avons proposé de résoudre ce problème en utilisant les champs markoviens et l'estimation bayesienne. L'avantage de la modélisation markovienne est de fournir un modèle simple qui permet de définir localement les informations a priori par des potentiels de clique. Un autre avantage est que le comportement local des champs markoviens permet de mettre en œuvre des algorithmes parallèles. Malheureusement, l'optimisation de la fonction d'énergie exige un temps de calcul important. Pour éviter cet inconvénient, nous avons proposé des modèles multi-grilles qui réduisent le temps de calcul.

En ce qui concerne *l'optimisation*, nous devons minimiser une fonction non-convexe pour trouve l'estimateur MAP des étiquettes. Nous avons deux choix pour résoudre ce problème: soit par *recuit simulé* soit par *relaxation déterministe*. Les algorithmes stochastiques convergent vers un minimum *global* mais exigent beaucoup de calcul. Les méthodes déterministes sont plus rapides mais ne donnent qu'un minimum *local*. La parallélisation des algorithmes est une autre possibilité pour améliorer le temps d'exécution.

Dans les applications réelles, les paramètres sont souvent inconnus, nous devons donc les déterminer à partir des données. Nous avons proposé quelques algorithmes itératifs et nous avons implanté un algorithme mono-grille et hiérarchique de segmentation non-supervisée. Les premiers résultats sont encourageants mais nous avons observé que les algorithmes non-supervisés exigent plus de temps d'exécution que les méthodes supervisées à cause de l'estimation des hyperparamètres (β and γ) qui sont calculés par recuit simulé dans l'implantation courante. Un autre point important est l'initialisation des paramètres gaussiens des classes. Nous avons remarqué que les méthodes non-supervisées sont plus sensibles au bruit que les algorithmes supervisés. Ceci est dû à la mauvaise initialisation des paramètres gaussiens.

Résultats et problèmes ouverts

Nos principals résultats sont un nouveau modèle markovien hiérarchique et un recuit multi-température proposé pour la minimisation de la fonction d'énergie. Le modèle hiérarchique est basé sur un modèle hiérarchique proposé par Perez *et al.* : nous avons utilisé le même processus pour définir les grilles grossières mais nous avons introduit d'un nouveau schéma de communication inter-niveaux. Le modèle obtenu est un champ de Markov complètement connecté sur toute la pyramide. Les nouvelles interactions permettent de propager des interactions locales d'une façon plus efficace mais le modèle devient plus complexe et exige beaucoup plus de temps de calcul. Nous remarquons que le modèle a été mis en œuvre sur une machine à connexions, qui n'est pas l'architecture optimale pour les modèles pyramidaux. De meilleurs résultats peuvent être obtenus, sur une architecture pyramidale.

Un autre sujet intéressant peut être l'étude de la relation entre l'estimateur MAP du modèle mono-grille et celui du modèle hiérarchique. Un point souvent critiqué est que la fonction d'énergie est définie sur toute la pyramide mais dans le résultat, uniquement le plus bas niveau est pris en compte. Nous remarquons qu'une idée similaire a été utilisé dans le modèle de restauration de Geman et Geman: ils ont introduit un processus de ligne, qui n'était pas utilisé dans le résultat final. Le modèle hiérarchique permet de travailler avec les cliques des pixels les plus éloignés pour un prix raisonnable.

Le recuit multi-température a été proposé pour résoudre le problème d'optimisation du modèle hiérarchique mais c'est un algorithme beaucoup plus général. En effet, l'étude mathématique de l'algorithme est très générale et ne suppose pas une structure pyramidale. La convergence vers un optimum *global* a été prouvée dans le cas général où chaque clique a sa propre loi de température locale.

Les résultats présentés dans le Chapitre 4 ne sont que primaires.Il reste encore beaucoup de travail à faire. Il y a deux problèmes principaux qui doivent être étudiés en détail. Il faut trouver un algorithme plus efficace pour l'estimation des hyperparamètres, par exemple une approximation par les champs moyens [60, 84]. Un autre problème à étudier est l'initialisation des paramètres gaussiens.

Publications

1. Z. Kato, M. Berthod, J. Zerubia, W. Pieczynski: Unsupervised Adaptive Image Segmentation. *ICASSP'95, Detroit, Mitchigan, USA, May 1995.*

2. Z. Kato, M. Berthod, J. Zerubia: A hierarchical Markov random Field Model and Multi-Temperature Annealing for parallel image classification. *Submitted to CVGIP.*

3. J. Zerubia, Z. Kato, M. Berthod: Multi-Temperature Annealing: a new approach for the energy-minimization of hierarchical Markov Random Field models. *ICPR-Computer Vision and Applications, Jerusalem, Oct. 1994.*

4. Z. Kato, M. Berthod, J. Zerubia: A hierarchical Markov random Field Model and Multi-Temperature Annealing for parallel image classification. *Research Report 1938, INRIA, Aug. 1993.*

5. Z. Kato, M. Berthod, J. Zerubia: A Hierarchical Markov Random Field Model for Image Classification. *In Proc.* 8^{th} *IMDSP Workshop, Cannes, France, September 1993.*

6. Z. Kato, M. Berthod, J. Zerubia: Multi-scale Markov Random Field models for parallel image classification. *In Proc. ICCV, Berlin, May 1993.*

7. Z. Kato, M. Berthod, J. Zerubia: Parallel image classification using multi-scale Markov Random Fields. *In Proc. ICASSP, Minneapolis, Apr. 1993.*

8. M. Berthod, Z. Kato, J. Zerubia: DPA: A Deterministic Approach to the MAP. *Accepted for publication in IEEE Trans. on Image Processing*, 1994.

9. Z. Kato, J. Zerubia, M. Berthod: Bayesian image classification using Markov Random Fields. In A. M.-D. G. Demoments, editor, *Maximum Entropy and Bayesian Methods*, pages 375–382. Kluwer Academic Publisher, 1993.

10. Z. Kato, J. Zerubia, M. Berthod: Image classification using Markov Random Fields with two new relaxation methods: Deterministic Pseudo Annealing and Modified Metropolis Dynamics. *Research Report 1606, INRIA, Feb. 1992.*

11. Z. Kato, J. Zerubia, M. Berthod: Satellite image classification using a Modified Metropolis Dynamics. *In Proc. ICASSP, San-Francisco, California, USA, Mar. 1992.*

Bibliographie

[1] R. Azencott. Image Analysis and Markov Fields. In *Proc. ICIAM'87*, pages 53–61, Paris, France, June 29-July 3 1987.

[2] R. Azencott. Parallel Simulated Annealing: An Overview of Basic Techniques. In R. Azencott, editor, *Simulated Annealing: Parallelization Techniques*, pages 37–46. John Wiley & Sons, Inc., 1992.

[3] R. Azencott, editor. *Parallel Simulated Annealing. Parallelization Techniques.* John Wiley & Sons, Inc., 1992.

[4] R. Azencott and C. Graffigne. Parallel Annealing by Periodically Interacting Multiple Searches: Acceleration Rates. In R. Azencott, editor, *Simulated Annealing: Parallelization Techniques*, pages 81–90. John Wiley & Sons, Inc., 1992.

[5] Y. Bard. *Nonlinear Parameter Estimation.* Academic Press, Inc., 1974.

[6] R. J. Baxter. *Exactly Solved Models in Statistical Mechanics.* Academic Press, 1990.

[7] M. Berthod, G. Giraudon, and J. Stromboni. Determistic Pseudo-Annealing : a Suboptimal Scheme for optimization in Markov Random Fields. An application to pixel classification. In *Proc. ECCV*, Santa Margherita Ligure, Italy, May 1992.

[8] J. Besag. On the statistical analysis of dirty pictures. *Jl. Roy. Statis. Soc. B.*, 1986.

[9] J. E. Besag. Spatial interaction and the statistical analysis of lattice systems (with discussion). *Jl. Roy. Statis. Soc. B. 36.*, pages 192–236, 1974.

[10] A. Blake and A. Zisserman. Visual reconstruction. *MIT Press, Cambridge - MA*, 1987.

[11] C. Bouman. A Multiscale Image Model for Bayesian Image Segmentation. Technical Report TR-EE 91-53, Purdue University, 1991.

[12] C. Bouman and B. Liu. Multiple Resolution segmentation of Texture Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:99–113, 1991.

[13] C. Bouman and M. Shapiro. Multispectral Image Segmentation using a Multiscale Model. In *Proc. ICASSP'92*, San Francisco, USA, March 1992.

[14] B. Braathen, W. Pieczynski, and P. Masson. Global and Local Methods of Unsupervised Bayesian Segmentation of Images. *Machine Graphics and Vision*, 2(1):39–52, 1993.

[15] H. Caillol, A. Hillion, and W. Pieczynski. Fuzzy Random Fields and Unsupervised Image Segmentation. *IEEE Geoscience and Remote Sensing*, 31(4):801–810, July 1993.

[16] O. Catoni and A. Trouvé. Parallel Annealing by Multiple Trials. In R. Azencott, editor, *Simulated Annealing: Parallelization Techniques*, pages 129–144. John Wiley & Sons, Inc., 1992.

[17] V. Černy. Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *J. Opt. Theory Appl.*, 45(1):41–51, January 1985.

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. Roy. Statist. Soc., ser. B*, vol. 39(1):1–38, 1977.

[19] H. Derin. Estimation Components of Univariate Gaussian Mixtures Using Prony's Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):142–148, January 1987.

[20] H. Derin, H. Elliott, R. Cristi, and D. Geman. Bayes Smoothing Algorithms for Segmentation of Binary Images Modeled by Markov Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):707–720, November 1984.

[21] H. Derin and C. S. Won. A Parallel Segmentation Algorithm Using Relaxation with Varying Neighborhoods and its Mapping to Array Procesors. *CVGIP*, 40:54–78, 1987.

[22] P. L. Dobruschin. The Description of a Random Field by Means of Conditional Probabilities and Constructions of its Regularity. *Theory of Probability and its Applications*, XIII(2):197–224, 1968.

[23] R. Duda and P.Hart. *Pattern Classification and Scene Analysis.* Wiley-Interscience, 1973.

[24] T. S. Ferguson. *Mathematical Statistics. A Decision Theoretic Approach.* Probability and Mathematical Statistics. Academic Press, 1967.

[25] K. Fukunaga and T. Flick. Estimation of the Parameters of a Gaussian Mixture Using the Method of Moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(4):410–416, July 1983.

[26] D. Geiger and A. Yuille. A Common Framework for Image Segmentation. Technical Report 89-7, Harvard Robotics Lab., 1989.

[27] S. B. Gelfand and S. K. Mitter. On Sampling Methods and Annealing Algorithms. In R. Chellappa, editor, *Markov Random Fields*, pages 499–515. Academic Press, Inc., 1993.

[28] D. Geman. Bayesian image analysis by adaptive annealing. In *Proc. IGARSS'85*, pages 269–277, Amherst, USA, Oct. 1985.

[29] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[30] B. Gidas. A Renormalization Group Approach to Image Processing Problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):164–180, February 1989.

[31] R. C. Gonzalez and P. Wintz. *Digital Image Processing.* Addison-Wesley Pub. Co., 1987.

[32] C. Graffigne. A parallel simulated annealing algorithm. Research report, CNRS, Université Paris-Sud, 1984.

[33] C. Graffigne. Parallel Annealing by Periodically Interacting Multiple Searches: An Experimental Study. In R. Azencott, editor, *Simulated Annealing: Parallelization Techniques*, pages 47–80. John Wiley & Sons, Inc., 1992.

[34] B. Hajek. A Tutorial Survey of Theory and Applications of Simulated Annealing. In *Proc. 24. Conf. on Decision and Control*, pages 755–760, Lauderdale, FL, December 1985.

[35] F. R. Hansen and H. Elliott. Image Segmentation Using Simple Markov Field Models. *CVGIP*, 20:101–132, 1982.

[36] F. Heitz, E. Memin, P. Pérez, and P. Bouthemy. A Parallel Multiscale Relaxation Algorithm for Image Sequence Analysis. In *Proc. ICPIP*, Paris, France, June 1991.

[37] F. Heitz, P. Perez, and P. Bouthemy. Multiscale Minimization of Global Energy Functions in Some Visual Recovery Problems. *CVGIP:IU*, 59(1):125–134, 1994.

[38] F. Heitz, P. Perez, E. Memin, and P. Bouthemy. Parallel Visual Motion Analysis Using Multiscale Markov Random Fields. In *Proc. Workshop on Motion*, Princeton, Oct. 1991.

[39] W. D. Hillis. The connection machine. *MIT press*, 1985.

[40] H. P. Hiriyannaiah, G. L. Bilbro, and W. E. Snyder. Restoration of piecewise-constant images by mean-field annealing. *Opt. Soc. Am. A*, 6(12):1901–1912, December 1989.

[41] R. Hummel and S. Zucker. On the foundations of relaxation labeling processes. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 5(3), 1983.

[42] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering*. Academic Press, 1970.

[43] F. C. Jeng and J. M. Woods. Compound Gauss - Markov Random Fields for Image Estimation. *IEEE Trans. Acoust., Speech and Signal Proc.*, ASSP-39:638–697, 1991.

[44] B. Jeon and D. A. Landgrebe. Classification with Spatio-Temporal Interpixel Class Dependency Contexts. *IEEE Trans. on Geoscience and Remote Sensing*, 30(4):663–672, July 1992.

[45] J. M. Jolion and A. Rosenfeld. *A Pyramid Framework for Early Vision*. Series in Engineering and Computer Science. Kluwer Academic Publishers, 1994.

[46] Z. Kato, M. Berthod, and J. Zerubia. A Hierarchical Markov Random Field Model for Image Classification. In *Proc. IMDSP Workshop*, Cannes, France, September 1993.

[47] Z. Kato, M. Berthod, and J. Zerubia. A hierarchical markov random field model and multi-temperature annealing for parallel image classification. Research Report 1938, INRIA, Aug. 1993.

[48] Z. Kato, M. Berthod, and J. Zerubia. Multiscale markov random field models for parallel image classification. In *Proc. ICCV*, Berlin, May 1993.

[49] Z. Kato, M. Berthod, and J. Zerubia. Parallel Image Classification using Multiscale Markov Random Fields. In *Proc. ICASSP*, Minneapolis, Apr. 1993.

[50] Z. Kato, M. Berthod, and J. Zerubia. A Hierarchical Markov Random Field Model and Multi-Temperature Annealing for Parallel Image Classification. *CVGIP:GMIP*, submitted.

[51] Z. Kato, M. Berthod, J. Zerubia, and W. Pieczynski. Unsupervised Adaptive Image Segmentation. In *ICASSP'95*, Detroit, USA, May 1995.

[52] Z. Kato, J. Zerubia, and M. Berthod. Image classification using Markov Random Fields with two new relaxation methods: Deterministic pseudo annealing and modified Metropolis dynamics. Research Report 1606, INRIA, Feb. 1992.

[53] Z. Kato, J. Zerubia, and M. Berthod. Satellite image classification using a modified Metropolis dynamics. In *Proc. ICASSP*, San-Francisco, California, USA, Mar. 1992.

[54] Z. Kato, J. Zerubia, and M. Berthod. Bayesian image classification using markov random fields. In A. M.-D. G. Demoments, editor, *Maximum Entropy and Bayesian Methods*, pages 375–382. Kluwer Academic Publisher, 1993.

[55] R. Kindermann and J. L. Snell. Markov Random fields and their applications. *Amer. Math. Soc.*, 1:1–142, 1980.

[56] S. Kirkpatrick, C. Gellatt, and M. Vecchi. Optimization by simulated annealing. *Science 220, pp 671-680*, 1983.

[57] P. V. Laarhoven and E. Aarts. Simulated annealing : Theory and applications. *Reidel Pub., Dordrecht, Holland*, 1987.

[58] S. Lakshmanan and H. Derin. Simultaneous parameter estimation and segmentation of gibbs random fields using simulated annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):799–813, Aug. 1989.

[59] S. Lakshmanan and H. Derin. Gaussian Markov Random Fields at Multiple Resolution. In *Markov Random Fields*, pages 131–157. Academic Press, Inc., 1993.

[60] D. A. Langan, K. J. Molnar, J. W. Modestino, and J. Zhang. Use of the Mean-Field Approximation in an EM-Based Approach to Unsupervised Stochastic Model-Based Image Segmentation. In *Proceedings ICASSP'92*, pages III–57–III–60, San Francisco, March 1992.

[61] S. Liu-Yu. *Reconnaissance de formes par vision par ordinateur : application à l'identification de foraminifères planctoniques.* Thèse, Université de Nic,e, Sophia Antipolis, June 1992.

[62] F. Marques, J. Cunillera, and A. Gasull. Hierarchical Segmentation Using Compound Gauss-Markov Random Fields. In *Proc. ICASSP*, San Francisco, Mar. 1992.

[63] J. L. Marroquin. *Probabilistic solution of inverse problems*. PhD thesis, MIT-Artificial Intelligence Lab., 1985.

[64] P. Masson and W. Pieczynski. SEM Algorithm and Unsupervised Statistical Segmentation of Satellite Images. *IEEE Geoscience and Remote Sensing*, 31(3):618–633, May 1993.

[65] E. Memin. *Algorithmes et architectures parallèles pour les approches markoviennes en analyse d'image*. PhD thesis, Université de Rennes I, 1993.

[66] E. Memin, F. Heitz, and F. Charot. Efficient Parallel Non-Linear Multigrid Relaxation Algorithms for Low-Level Vision Applications. *Journal of Parallel Distributed Computing, accepted*, 1994.

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

[68] J. Moussouris. Gibbs and Markov Random System with Constrainsts. *Journal of Statistical Physics*, 10(1):11–33, Jan. 1974.

[69] G. K. Nicholls and M. Petrou. A Generalisation of Renormalisation Group Methods for Multi Resolution Image Analysis. In *Proc. ICPR'92*, pages 567–570, 1992.

[70] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. Series in Systems Science. McGraw-Hill Book Company, 1984.

[71] P. Pérez. *Champs markoviens et analyse multirésolution de l'image: application à l'analyse du mouvement*. PhD thesis, Université de Rennes I, 1993.

[72] P. Perez and F. Heitz. Multiscale Markov Random Fields and Constrained Relaxation in Low Level Image Analysis. In *Proc. ICASSP*, San-Francisco, Mar. 1992.

[73] P. Pérez and F. Heitz. Restriction of Markov Random Fields on Graphs. Application to Multiresolution Image Analysis. Research Report 2170, INRIA-Rennes, March 1994.

[74] W. Pieczynski. Statistical image segmentation. In *Machine Graphics and Vision, GKPO'92*, pages 261–268, Naleczow, Poland, May 1992.

[75] W. Pieczynski. Champs de Markov cachés et estimation conditionnelle itérative. *Traitement du signal*, 11(2), 1994.

[76] J. G. Postaire and C. P. A. Vasseur. An approximate solution to normal mixture identification with application to unsupervised pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(2):163–179, March 1981.

[77] W. Pratt. *Digital Image Processing*. Wiley-Interscience, 1978.

[78] S. Rajasekaran. On The Convergence Time of Simulated Annealing. Research Report MS-CIS-90-89, University of Pennsylvania, Department of Computer and Information Science, November 1990.

[79] Y. Rosanov. Markov Random Fields. *Springer Verlag*, 1982.

[80] M. Sigelle, C. Bardinet, and R. Ronfard. Relaxation of Classification Images by a Markov Field Technique - Application to the Geographical Classification of Bretagne Region. In *Proc. European Association of Remote Sensing Lab. Conf.*, Eger, Hungary, Sept 1992.

[81] H. L. Tan, S. B. Gelfand, and E. J. Delp. A Cost Minimization Approach to Edge Detection Using Simulated Annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):3–18, January 1991.

[82] A. Trouvé. Massive Parallelization of Simulated Annealing: A Mathematical Study. In R. Azencott, editor, *Simulated Annealing: Parallelization Techniques*, pages 145–164. John Wiley & Sons, Inc., 1992.

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

[84] J. Zerubia and R. Chellappa. Mean field annealing using Compound Gauss-Markov Random fields for edge detection and image estimation. *IEEE Trans. on Neural Networks*, 8(4):703–709, July 1993.

[85] J. Zerubia, Z. Kato, and M. Berthod. Multi-Temperature Annealing: A New Approach for the Energy-Minimization of Hierarchical Markov Random Field Models. In *Proc. ICPR'94*, Jerusalem, Israel, Oct. 1994.

[86] J. Zerubia and F. Ployette. Detection de contours et restauration d'image par des algorithmes deterministes de relaxation. Mise en oeuvre sur la machine a connexions CM2. *Traitement du Signal*, Sept. 1991.

ABSTRACT

The main concern of this thesis is Markovian modelization in early vision. We consider low level vision tasks in a common framework, called image labeling, where the problem is reduced to assigning labels to pixels. Our approach is probabilistic, using Markov Random Fields (MRF) and Bayesian estimation, in particular Maximum A Posteriori (MAP) estimation. The advantage of MRF modelization is that a priori information can be "coded" *locally* through clique potentials. We also discuss pyramidal MRF models, which reduce the computing time and increase the quality of final results. Parameter estimation is an important problem in real-life applications in order to implement completely data-driven algorithms. We apply some methods to the estimation of monogrid model-parameters and propose a new algorithm for the hierarchical model. The preliminary results are encouraging but there is still a lot of work to do.

All MRF models result in a non-convex energy function. The minimization of this function is done by Simulated Annealing or deterministic relaxation. We also discuss the possible parallelization techniques of optimization algorithms.

Our main result is a new hierarchical MRF model and a Multi-Temperature Annealing algorithm proposed for the energy minimization of the model. The convergence of the MTA algorithm has been proved towards a *global* optimum in the most general case, where each clique may have its own *local* temperature schedule. There are still some open problems such as the relation between monogrid and hierarchical MAP estimates, implementing the hierarchical model on a pyramidal computer, or looking for a faster parameter estimation method.

Keywords: computer vision, early vision, Markovian model, multiscale model, hierarchical model, parallel combinatorial optimization algorithm, multi-temperature annealing, parameter estimation.

RÉSUMÉ

Dans cette thèse, nous nous intéressons aux modèles markoviens appliqués aux problèmes de vision pré-attentive. Nous considerons ces problèmes dans un cadre générale, appellé étiquetage d'images, où le problème consiste à attribuer des étiquettes aux pixels. Notre approche est fondée sur les champs de Markov et l'estimation bayesienne, en particulier l'estimation de Maximum A Posteriori (MAP). L'avantage de la modèlisation markovienne est de fournir un modèle simple qui nous permet de définir les informations a priori par des potentiels locaux. Nous présentons aussi les modèles pyramidaux qui réduisent le temps de calcul et améliorent le résultat final. L'estimation des paramètres est un autre problème important pour les applications réelles. Nous appliquons quelques méthodes connues à l'estimation de paramètres du modèle monogrille et proposon des nouveaux algorithms d'estimation pour le modèle hiérarchique. Les premiers resultats sont satisfaisantes mais il reste beaucoup de travail à faire sur le sujet.

Tous les modèles markoviens nécessitent la minimisation d'une fonction d'énergie non-convexe. Nous avons deux choix pour résoudre ce problème: soit par recuit simulé soit par relaxation déterministe. Nous discutons la possibilité de paralleliser ces algorithmes.

Notre principal résultat est un modèle markovien hiérarchique et un algorithme de récuit multi-température (MTA) pour la minimisation de la fonction d'énergie du modèle hiérarchique. Pour le MTA, nous avons prouvé la convergence vers un minimum global dans le cas le plus général où chaque clique a sa propre loi de température. Il reste quelques problèmes ouverts comme la relation entre les valeurs estimées au sens du MAP pour les modèles monogrille et hiérarchique, l'implantation de l'algorithme MTA sur une architecture pyramidale ou bien la mise en œuvre une méthode beaucoup plus rapide pour l'estimation de paramètres.

Mots clefs: vision par ordinateur, vision pré-attentive, modèlisation markovien, modèle multiéchelle, modèle hiérarchique, algorithmes d'optimisation massivement parallèles, recuit multi-température, etimation de paramètres.