

École des Ponts
ParisTech

PROJET D'INITIATION À LA RECHERCHE 2017

RESTAURATION MARKOVIENNE D'IMAGES

M. KARPE, A. SADACA, N. SOUSSI, C. ZENG

sous la direction de
A. MARUANI

ABSTRACT : Images can be subject to many imperfections during acquisition phases and transmission, resulting in noise, blurring, color saturation and noise, amidst other nuisances. In this paper, we focus on noise imperfections and study a post-processing technique in image denoising based on a model inspired from the statistical physics Ising and Potts model, and the theory of Markov Random Fields. Given simple grayscale images, the aim is to add a gaussian noise and to retrieve the original image.

After demonstrating the relevance of adopting the Markov Random Fields approach in modeling the image and detailing the critical properties of the Ising Model using separately Mean-Field and Markovian perspectives, we implement the Markov Random Field model successively with Potts potential – obtained from a generalisation of the Ising model to an arbitrary number of states – and gaussian markovian energy. The resulting algorithms allow a performance comparison between two minimisation formalisms : simulated annealing and Iterated Conditional Mode (ICM), coupled with one sampling and updating formalism : Metropolis algorithm or Gibbs sampling. The resulting images show the relative accuracy of the gaussian markovian model and superior performance of the ICM minimisation, and also exhibit the influence of increasing the value of the homogenising factor present in both potentials' formula, in improving the quality of the restauration.

KEYWORDS : Image Processing, Markovian Random Fields, Probability, Ising Model.

1 Introduction

L'acquisition d'une image numérique est le plus souvent accompagnée de l'apparition de bruit, souvent dû à une imperfection au niveau de la détection, de la transmission ou de la compression du signal, ou encore à des défauts inhérents à l'environnement comme la présence d'éclairage insuffisant ou trop prononcé. La suppression de ce bruit est même un enjeu vital dans plusieurs domaines, notamment l'imagerie médicale, et la recherche d'un algorithme efficace de débruitage d'image demeure un défi persistant, à la croisée de plusieurs domaines scientifiques : analyse fonctionnelle, probabilités, statistiques et sciences physiques. Dans cette étude, nous implémentons différents algorithmes de restitution d'une image contenant des motifs simples, en niveaux de gris, bruitée selon un bruit gaussien. Pour cela, nous adoptons une description probabiliste de l'image, considérant ses pixels comme des variables aléatoires X_i , d'énergie $U(X_i)$ qui dépend de son voisinage. Cette approche permet d'envisager l'image comme un champ de Markov. L'expression de ces énergies viendra de l'analogie que l'on fera, dans la description de l'image, avec le modèle d'Ising en dimension 2 (plus précisément avec le modèle de Potts, sa généralisation à un système de n états), que l'on confrontera avec le modèle markovien gaussien. Enfin, on discutera des performances des algorithmes de minimisation utilisés (recuit simulé ou modes conditionnels itérés) ainsi que de l'influence des différents paramètres du problème.

2 Méthodes de restauration d'image par champs de Markov

2.1 Champs de Markov

2.1.1 Pourquoi une telle approche ?

Dans le cadre de ce projet, nous nous concentrons sur le traitement d'images. En quoi consiste ce travail ? Nous voulons modéliser une image dont tous les pixels ne sont pas déterminés à l'avance. Il existe plusieurs manières d'envisager le problème. On peut tenter de rendre l'image plus nette, ou au contraire la flouter pour masquer le bruit. Cette démarche correspond par exemple à la restauration d'une image bruitée.

Détaillons le cadre de ce problème. On se donne une image initiale bruitée. Pour améliorer sa netteté, il nous faut changer la valeur de certains de ses pixels. Ce processus est soumis à des contraintes : nous souhaitons obtenir une image proche de celle d'origine. Cependant, nous ne connaissons pas l'image d'origine. Pour chaque pixel, il nous faudra tester des valeurs probables à prendre en fonction de notre intention (rendre l'image très nette, conserver une image très proche de l'originale...).

Comment modéliser ce problème mathématiquement ? Nous souhaitons construire un modèle probabiliste qui va affecter à chaque pixel une valeur selon son environnement et selon des critères pré-établis. La valeur prise par chaque pixel est alors vue comme une variable aléatoire. L'image totale est donc considérée comme un N -uplet de variables aléatoires.

Définition 1 Nous appellerons dans la suite X une réalisation possible de l'image, c'est-à-dire un N -uplet $(X_1, \dots, X_i, \dots, X_n)$ de variables aléatoires.

Nous pouvons munir l'espace des réalisations d'une probabilité P telle que :

$$P(x = y) = P(X_1 = Y_1, \dots, X_n = Y_n)$$

où $y = (Y_1, \dots, Y_n)$ est une réalisation de l'image.

Expliquons pourquoi cette modélisation conduit à introduire la notion de champ de Markov. Afin de construire une image interprétable, la valeur attribuée à un pixel dépendra des valeurs des pixels voisins. En effet, il est peu probable qu'une réalisation constituée de pixels tous très différents puissent représenter une vraie image interprétable. On définit alors la notion de voisinage comme ci-après :

Définition 2 V_s est le voisinage de $s \Leftrightarrow V_s = \{t \mid s \notin V_s \text{ et } t \in V_s \Rightarrow s \in V_t\}$

Pour représenter l'influence des autres pixels sur un pixel s donné, on peut utiliser la notion de potentiel. Ainsi, le potentiel du pixel s noté U_s peut être déterminé selon les potentiels des autres pixels. Il en découle que la probabilité que le pixel s prenne une certaine valeur dépend des potentiels de ses voisins (tout comme la couleur du pixel dépend des couleurs voisines). On peut assimiler la couleur du pixel à une énergie, fonction de son potentiel. De plus, ce sont les voisins du pixel s (plutôt que les pixels lointains) qui joueront un rôle important sur sa valeur. Ce raisonnement, qui n'est pour l'instant qu'intuitif, nous amène à considérer une image comme un champ de Markov.

Quelles sont les caractéristiques d'un tel champ ?

Définition 3 X est un champ de Markov si et seulement si la probabilité conditionnelle locale en un site n'est fonction que de la configuration du voisinage du site considéré. Cette définition est cohérente avec celle généralement utilisée pour une suite de variables aléatoires : la suite (X_n) est une chaîne de Markov si et seulement si :

$$P(X_n = x_n \mid X_{n-1} = x_{n-1}, \dots, X_1 = x_1) = P(X_n = x_n \mid X_{n-1} = x_{n-1})$$

Dans le cadre de notre étude, il est possible de se placer sous des hypothèses markoviennes, où l'image totale x a une énergie $U(x)$ et une probabilité de réalisation $P(x)$.

2.1.2 Étude d'un champ de Markov

Pour ce modèle, nous donnons au pixel i une couleur dépendant des couleurs de ses voisins. On peut donc écrire que la probabilité pour le pixel i d'être d'une certaine couleur, c'est-à-dire la probabilité pour la variable aléatoire X_i de prendre une valeur x_i , dépend de l'énergie de ses voisins. Ce résultat fut prouvé par le théorème de Hammersley-Clifford (1974), qui permet même une expression de cette probabilité. Ce théorème met en jeu une nouvelle notion, celle de champ de Gibbs.

Pour la définir, nous introduisons les notions de mesure de Gibbs et de clique pour un point donné du champ.

Définition 4 Une clique est un ensemble de sites qui sont tous voisins les uns des autres, ou bien un singleton.

Le nombre et les cardinaux possibles des cliques sont donc fonction du critère de voisinage choisi. Un système à 4 voisins ne permet l'existence que de cliques à 1 ou 2 éléments, tandis qu'un système à 8 voisins autorise des cliques à 3 ou 4 éléments.

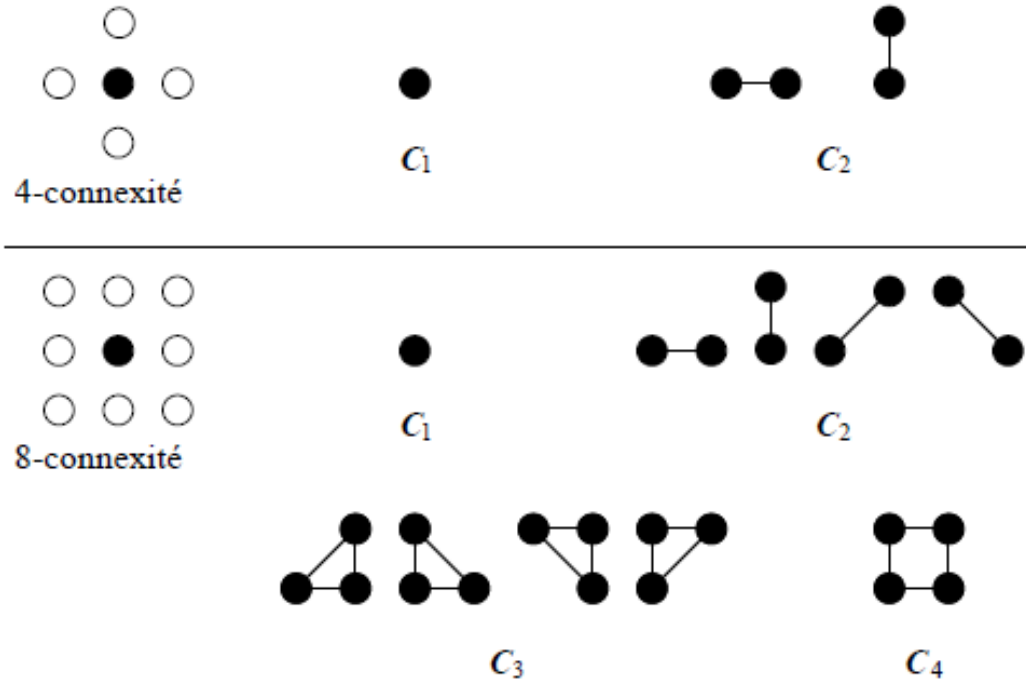


FIGURE 1 – Représentation de cliques C_n d'ordre 1 à 4 pour des voisinages 4 et 8-connexe

Définition 5 La mesure de Gibbs donne la probabilité de réalisation d'un champ aléatoire X , et est donnée par :

$$P(X = x) = \frac{\exp(-U(x))}{Z}$$

où $U(x)$ est l'énergie d'un tel champ et peut se réécrire sous la forme additive

$$U(x) = \sum_{c \in \mathcal{C}} U_c((x_{j_1}, \dots, x_{j_l}), x_{j_k} \in x, j_k \in c)$$

La constante Z , nommée fonction de partition, se détermine par normalisation, et vaut :

$$Z = \sum_{x \in A} \exp(-U(x))$$

On s'aperçoit que le nombre de termes dans la somme croît de manière exponentielle avec la taille de l'image (chaque pixel ajouté peut prendre 2 valeurs, ce qui multiplie par 2 le nombre de réalisations de l'image, et donc le nombre de termes dans la somme). Pour calculer ces probabilités, il faudra donc développer des calculs ne faisant pas intervenir ce terme Z , ou bien trouver une formule sommatoire.

Définition 6 Un champ de Gibbs est un champ aléatoire X dont la probabilité de réalisation est donnée par la mesure de Gibbs :

$$P(X = x) = \frac{1}{Z} \exp(-U(x)) = \frac{1}{Z} \exp\left(-\sum_{c \in C} U_c((x_{j_1}, \dots, x_{j_l}), x_{j_k} \in x, j_k \in c)\right)$$

Théorème Le théorème de Hammersley-Clifford énonce que sous les hypothèses suivantes :

- Le système de voisinage choisi pour les 2 champs est identique.
- Tout système de voisinage V est borné.
- L'espace E des états pouvant être pris par chaque particule est discret

On a l'équivalence entre :

- X est un champ de Markov tel que : $x \in A \Rightarrow P(X = x) > 0$
- X est un champ de Gibbs

On note que ces hypothèses sont vérifiées dans le cadre d'un champ correspondant à une image.

2.1.3 Préparation en vue de l'algorithme

Nous allons établir des résultats sur les probabilités de réalisation des images. En effet, ceux-ci permettront de construire des algorithmes de construction d'images. Comme vu plus tôt, le calcul de la probabilité $P(X = x)$ se révèle impossible à réaliser du fait de la taille de Z . Pour s'affranchir de ce terme, nous étudierons des probabilités conditionnelles. En effet, une probabilité conditionnelle du type $P(A|B) = \frac{P(A \cap B)}{P(B)}$ fait intervenir un quotient de 2 probabilités. Il est donc possible de simplifier les termes Z à condition de bien choisir les probabilités en question.

Le calcul qui nous intéresse dans cette étude est celui de la probabilité pour un pixel de prendre une valeur particulière connaissant les valeurs de ses voisins. Par exemple, si tous ses voisins sont noirs, le pixel doit présenter une probabilité importante d'être également noir.

Notation 1 On s'intéresse donc à la probabilité que le pixel X_i prenne la valeur x_i connaissant son voisinage, probabilité qu'on écrira $P(X_i = x_i|V_i)$.

Notation 2 Par définition d'un champ markovien, cette probabilité est la même que la probabilité que le pixel X_i prenne la valeur x_i connaissant tout le reste de l'image, probabilité que l'on notera $P(X_i = x_i|X^i = x^i)$ avec x^i le vecteur contenant l'ensemble des pixels sauf le i -ème.

L'intersection des événements $(X_i = x_i)$ et $(X^i = x^i)$ permet de définir une réalisation de l'image $X = x$:

$$P(X_i = x_i | X^i = x^i) = \frac{P(X = x)}{P(X^i = x^i)}$$

L'événement $X^i = x^i$ peut se réécrire comme $\bigcup_{a_i \in A_i} X = y_i$ où y_i est le champ tel que le pixel i vaut a_i ($y_i[i] = a_i$), les autres pixels étant fixés par x^i ($y_i[j] = x^i[j]$ pour $j \neq i$), et où A_i est l'ensemble des valeurs pouvant être prises par le pixel i . On a ici une réunion disjointe d'événements indépendants :

$$\begin{aligned} P(X^i = x^i) &= \sum_{a_i \in A_i} P(X = y_i) \\ P(X^i = x^i) &= \frac{\sum_{a_i \in A_i} \exp(-U(y_i))}{Z} \\ P(X_i = x_i | V_i) &= \frac{\exp(-U(x))}{\sum_{a_i \in A_i} \exp(-U(y_i))} \end{aligned}$$

Nous allons simplifier cette expression afin de pouvoir l'implémenter dans un algorithme. Définissons d'abord l'énergie locale d'un site x_j . Celle-ci correspond à la somme des énergies de chacune des cliques auxquelles x_j appartient, c'est-à-dire :

$$U_j(x_j | V_j) = \sum_{c \in C / j \in c} U_c((x_{i_1}, \dots, x_{i_l}), i_k \in c)$$

On rappelle de plus que :

$$U(x) = \sum_{c \in C} U_c((x_{j_1}, \dots, x_{j_l}), j_k \in c, x_{j_k} \in x)$$

Après avoir fixé un site i , on peut diviser l'ensemble des cliques en 2 sous-ensembles : celles qui contiennent le site x_i , et celles qui ne le contiennent pas, c'est-à-dire :

$$U(x) = \sum_{c \in C / i \in c} U_c((x_{j_1}, \dots, x_{j_l}), x_{j_k} \in x, j_k \in c) + \sum_{c \in C / i \notin c} U_c((x_{j_1}, \dots, x_{j_l}), x_{j_k} \in x, j_k \in c)$$

On peut donc réécrire U sous la forme :

$$U(x) = \sum_{c \in C / i \notin c} U_c((x_{j_1}, \dots, x_{j_l}), x_{j_k} \in x, j_k \in c) + U_i(x_i | V_i, x_i = x[i])$$

Ce résultat permet de simplifier l'expression de la probabilité calculée précédemment :

$$\sum_{a_i \in A_i} \exp(-U(y_i)) = \sum_{a_i \in A_i} \exp\left(-\left(\sum_{c \in C / i \notin c} U_c((x_{j_1}, \dots, x_{j_l}), x_{j_k} \in y_i, j_k \in c) + U_i(a_i | V_i)\right)\right)$$

Le terme

$$\sum_{c \in C/i \notin c} U_c((x_{j_1}, \dots, x_{j_l}), x_{j_k} \in y_i, j_k \in c)$$

ne dépend pas de la valeur a_i choisi pour le pixel i , par définition (on s'intéresse uniquement à des x_j pris dans des cliques qui ne contiennent pas le pixel i). On a donc :

$$\sum_{a_i \in A_i} \exp(-U(y_i)) = \exp\left(-\sum_{c \in C/i \notin c} U_c((x_{j_1}, \dots, x_{j_l}), x_{j_k} \in y_i, j_k \in c)\right) \cdot \sum_{a_i \in A_i} \exp(-U_i(a_i|V_i))$$

Résultat Finalement, on obtient :

$$P(X_i = x_i | X^i = x^i) = \frac{\exp(-U_i(x_i|V_i))}{\sum_{a_i \in A_i} \exp(-U_i(a_i|V_i))}$$

Expliquons pourquoi cette expression est intéressante. Son calcul ne nécessite pas le calcul de la constante de normalisation Z . Pour pouvoir l'appliquer, il suffit de connaître l'expression de l'énergie, ainsi que l'état des voisins du pixel i (on est bien dans le cadre d'un champ markovien). Ainsi, il est possible de choisir une énergie U qui favorise certaines valeurs x_i en fonction des valeurs des pixels voisins. Cela nous permettra de traiter des images en choisissant (par exemple) des gradients très faibles sur l'image (une très faible différence de couleur entre 2 pixels voisins).

2.2 Modèle d'Ising

La forme de la probabilité choisie ainsi que la notion de voisinage rappelle le modèle d'Ising que nous avons étudié dans le cadre du cours de Physique statistique de F. Chevoir [2]. Nous poussons l'analogie un peu plus loin en essayant d'interpréter le modèle d'Ising en termes d'images. Nous restons donc naturellement dans le cas de deux dimensions.

Pour ce faire, il convient de faire quelques rappels sur le modèle d'Ising en deux dimensions. C'est un réseau de spins dont chaque élément peut prendre deux valeurs ± 1 et pointe conventionnellement soit vers le haut, soit vers le bas. En voici une représentation :

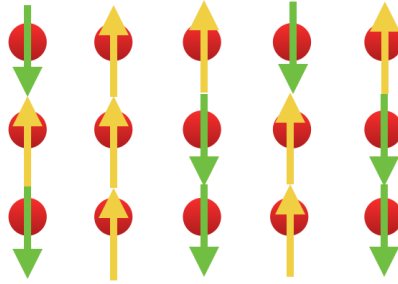


FIGURE 2 – Assemblée de spins. Ici, le moment total est 8 et correspond à une magnétisation de $2/3$.

Nous savons qu'en l'absence de champ extérieur et d'interactions entre les voisins, les deux valeurs ± 1 sont équiprobables.

Cependant, quelle est précisément l'analogie entre des systèmes mécaniques statistiques (ici magnétiques) et des images? Nous considérons l'image comme un tableau de pixels modélisé par un réseau cristallin, tandis que les niveaux de gris de chaque pixel sont vus comme les états des atomes ou molécules du réseau. Dans le cadre de l'analogie avec le modèle d'Ising, nous devons restreindre la plage de niveaux de gris à deux valeurs seulement : 0 et 255, qui correspondent au noir et au blanc.

Formalisons le modèle d'Ising dans le cas général. Considérons un réseau régulier et infini (pour négliger les effets de bord) dont les N spins s_i occupent un site i . Chaque site i possède une valeur $\sigma_i = \pm 1$. Ce réseau est plongé dans un champ extérieur h et on considère qu'il y a une interaction d'intensité $J > 0$ entre chaque spin et ses plus proches voisins. Notons z la coordinence du réseau, soit le nombre de voisins de chaque site. L'énergie totale a alors pour expression :

$$E(s_1, s_2, \dots, s_N) = -J \sum_{\langle i,j \rangle} s_i s_j - h \sum_i s_i \quad (1)$$

Remarque Cette formule reflète bien l'intuition. Des spins voisins tendent à pointer dans la même direction. De plus, l'énergie est petite lorsque les spins voisins sont dans le même sens, et grande lorsqu'ils sont de sens opposés.

2.2.1 Propriétés critiques du modèle d'Ising

Nous avons réalisé une simulation du modèle d'Ising pour mieux comprendre son comportement.

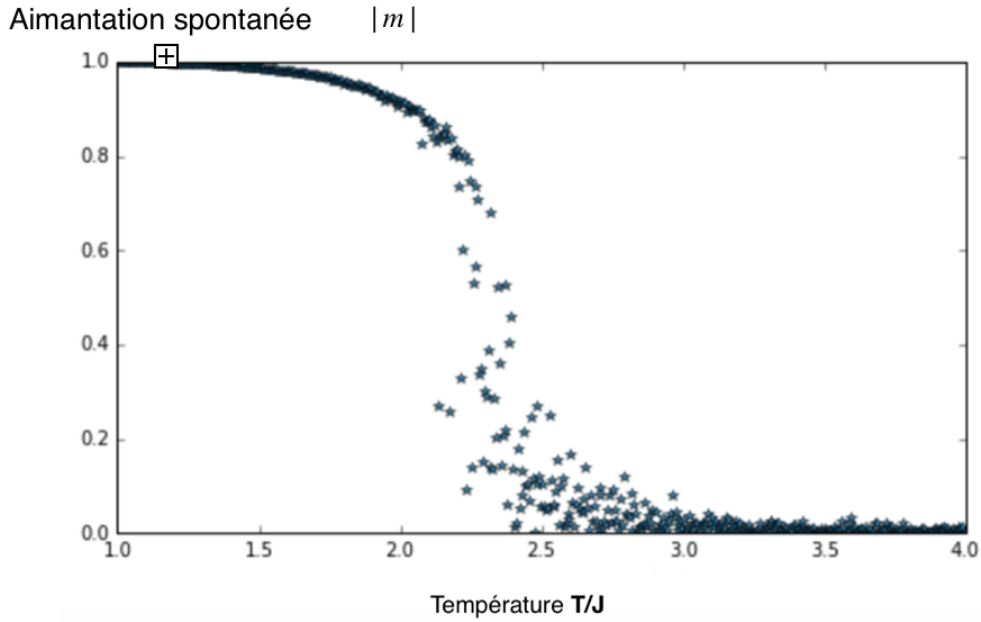


FIGURE 3 – Aimantation spontanée en fonction de la température

Nous observons alors certaines propriétés critiques. Notre simulation prédit une transition de phase ayant lieu à la température de Curie. Il existe en effet une transition para-ferromagnétique que l'on peut décrire par la méthode de champ moyen. D'après le graphique, cette transition a lieu pour une température critique qui semble être de l'ordre de $2.3J K$ où J est le paramètre de Heisenberg de la relation (1).

En 1944, Lars Onsager a résolu le modèle carré 2D en l'absence d'un champ extérieur ($h = 0$) grâce à une démonstration extrêmement technique reposant sur la théorie algébrique des matrices. Onsager décrit l'apparition d'une aimantation à la température critique T_c telle que :

$$\frac{T_c}{J} \approx 2.269 K.J^{-1}$$

Depuis ce résultat, de nombreuses méthodes ont été élaborées pour retrouver cette valeur. Nous nous intéressons à la méthode du champ moyen pour obtenir un ordre de grandeur de cette température critique. Grâce à la théorie des champs de Markov que nous avons exposée dans la partie précédente, nous adopterons une approche markovienne pour estimer la valeur de la température critique.

2.2.2 Calcul en champ moyen

Nous utilisons la méthode du champ moyen pour trouver une approximation de la valeur critique de la température T_c . Nous faisons l'hypothèse que les seules interactions considérées sont celles avec

les 4 plus proches voisins, représentés sur le schéma ci-dessous :

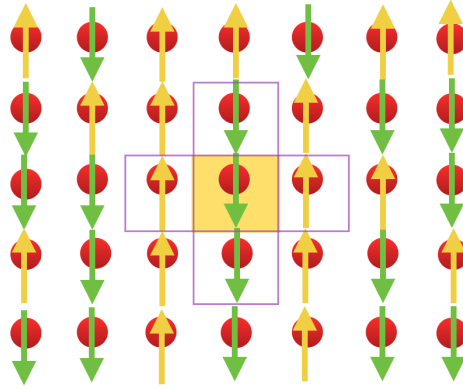


FIGURE 4 – Plus proches voisins dans le modèle d’Ising 2D

On cherche ici la valeur moyenne m de chaque spin à la température T et dans un champ h . Nous remplaçons donc les valeurs locales de l’aimantation spontanée des spins par leur valeur moyenne. Ainsi, tous les spins voisins du spin s_i ont pour aimantation cette même valeur m . Celui-ci est donc soumis à un champ effectif $h' = h + zJm$. D’après la distribution de Boltzmann, nous pouvons calculer cette valeur moyenne avec $\beta = \frac{1}{k_B T}$:

$$m = \frac{\exp(\beta h') - \exp(-\beta h')}{\exp(\beta h') + \exp(-\beta h')} = \tanh \beta(h + zJm)$$

En particulier pour $h = 0$, l’aimantation m vérifie donc $m = \tanh(\beta zJm)$. Résolvons graphiquement cette équation.

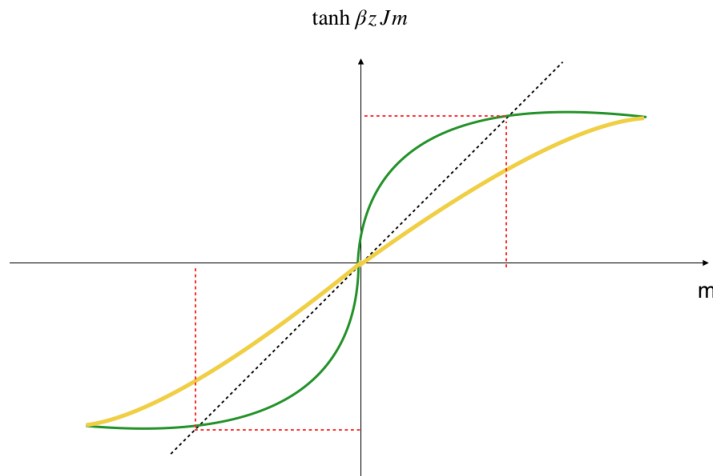


FIGURE 5 – Discussion de l’intersection de la fonction $\tanh \beta(h + zJm)$ avec la droite $y = m$ passant par l’origine selon le régime

La résolution graphique montre un changement de comportement à la température critique $T_c = zJ/k_B$. Comme $z = 4$, nous obtenons : $\frac{T_c}{J} = 2.898 \text{ K.J}^{-1}$. L'écart relatif au calcul exact est donc de :

$$\epsilon = \frac{2.899 - 2.269}{2.269} = 27.8\%$$

Le résultat est du bon ordre de grandeur, mais l'écart relatif est trop important pour être acceptable. En effet, les fluctuations jouent un rôle important dans l'hypothèse du champ moyen qui constitue donc une hypothèse grossière. Nous cherchons donc à obtenir une meilleure approximation par une méthode markovienne.

2.2.3 Une approche markovienne

Hypothèses Nous nous plaçons dans un champ magnétique extérieur nul ($h = 0$) et supposons l'existence d'interactions avec les plus proches voisins sur un réseau de maille carrée. En se servant de la théorie des champs de Markov exposée précédemment, nous allons tenter d'améliorer les résultats obtenus par la méthode des champs moyens.

Dans le modèle d'Ising 2D en champ extérieur nul ($h = 0$), l'énergie totale du système peut s'écrire comme la somme sur toutes les paires de voisins (i, j) :

$$U = \sum_{(i,j)} U_{ij}$$

où l'énergie d'interaction U_{ij} s'écrit : $U_{ij} = -J\sigma_i\sigma_j$ avec $\sigma_i = \pm 1$ et J désigne le paramètre de couplage d'Ising-Heisenberg.

Le modèle d'Ising est un exemple de champ de Markov. Dans ce modèle, la probabilité qu'un spin occupant un site s du réseau ait pour valeur x_s connaissant le reste du réseau S_s est égale à la probabilité conditionnelle connaissant la configuration du voisinage immédiat N_s .

D'après la distribution de probabilité et le théorème de Hammersley-Clifford, pour une température T fixée, cette probabilité s'écrit :

$$P(x_s|S_s) = P(x_s|N_s) = \frac{1}{Z_{N_s}} \exp\left(\frac{-U(x_s|N_s)}{k_B T}\right)$$

où $U(x_s|N_s)$ est la somme des termes énergétiques impliquant le site s selon la configuration du voisinage N_s , soit l'énergie conditionnelle du site s . Z_{N_s} est la fonction de partition locale de la configuration et vaut :

$$Z_{N_s} = \sum_{\{x_s\}} \exp\left(\frac{-U(x_s|N_s)}{k_B T}\right)$$

Soit p la probabilité qu'un spin s ait pour valeur $+1$. La règle de Bayes conduit à :

$$p = P(x_s = 1) = \sum_{\{N_s\}} P(x_s = 1_s)P(N_s) = \sum_{\{N_s\}} \frac{\exp(\frac{-U(x_s|N_s)}{k_B T})}{Z_{N_s}} P(N_s)$$

où $P(N_s)$ est la probabilité de la configuration N_s , la somme se faisant sur toutes les configurations du voisinage immédiat N_s .

Notons $i(j)$ la configuration locale où i spins voisins ont la valeur j , ainsi que $K = \frac{2J}{k_B T}$ où K est la constante de Potts. En considérant toutes les configurations possibles, la probabilité précédente s'écrit :

$$\begin{aligned} p &= \sum_k P[k(\pm 1)] \binom{4}{k} \frac{e^K}{e^K + e^{4-K}} \\ &= \frac{e^{4K}}{1 + e^{4K}} P(4(1)) + \binom{4}{1} \frac{e^{3K}}{e^K + e^{3K}} P(3(1)) + \dots + \binom{4}{3} \frac{e^K}{e^K + e^{3K}} P(3(-1)) + \frac{e^{4K}}{1 + e^{4K}} P(4(-1)) \end{aligned}$$

En effet, il s'agit de la somme sur k du nombre de configurations où k spins ont la valeur ± 1 fois la probabilité que le voisinage soit dans la configuration $i(j)$ fois la probabilité de Boltzmann. Un exemple de configurations correspondant au calcul précédent est donné ci-dessous :

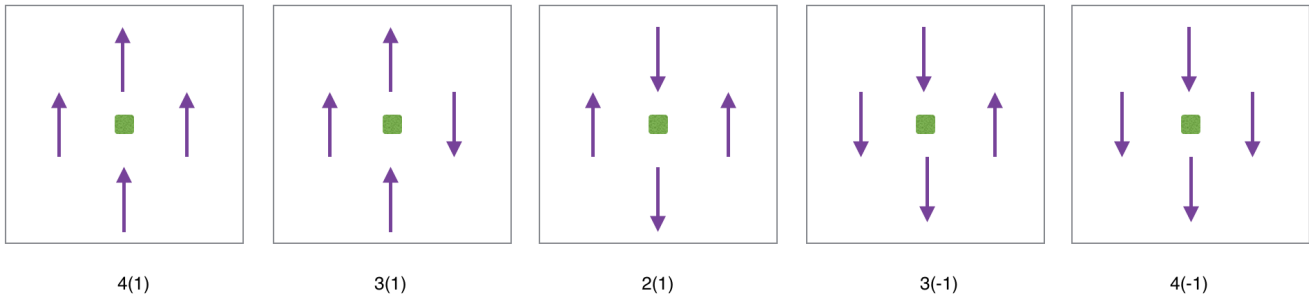


FIGURE 6 – Exemple de configuration des spins

Nous ajoutons deux hypothèses :

- Les spins du voisinage N_s sont décorrélés, ce qui constitue une hypothèse forte ;
- La probabilité p est invariante par translation sur le réseau de spin, ce qui est une hypothèse faible.

Comme N_s suit une loi de Bernoulli, nous pouvons reformuler l'équation précédente :

$$p = \frac{e^{4K}}{1 + e^{4K}} p^4 + \binom{4}{1} \frac{e^{3K}}{e^K + e^{3K}} p^3(1-p) + \binom{4}{2} \frac{e^{2K}}{2e^{2K}} p^2(1-p)^2 + \binom{4}{3} \frac{e^K}{e^K + e^{3K}} p(1-p)^3 + \frac{e^{-4K}}{1 + e^{-4K}} (1-p)^4$$

Remarquons que l'équation est invariante par la transformation $p \rightarrow 1 - p$. D'après la théorie des jauges, cette transformation échange les valeurs de spins entre 0 et 1. On en déduit les propriétés suivantes :

- Cette équation met en avant un polynôme du troisième degré plutôt que quatre et $p = \frac{1}{2}$ est une solution pour tout K , c'est-à-dire pour toute température ;
- Le graphe de la fonction ϕ est symétrique par rapport au point $(\frac{1}{2}, \frac{1}{2})$.

Donc $p = \phi(p)$ et $1 - p = \phi(1 - p)$. En sommant ces deux équations, nous obtenons :

$$\phi(p) + \phi(1 - p) = 1$$

D'où $2\phi(\frac{1}{2}) = 1$, soit $\phi(\frac{1}{2}) = \frac{1}{2}$. De plus, cette égalité prouve que ϕ est de degré 3.

Nous pouvons à présent écrire : $\phi(p) = (p - \frac{1}{2})\mathbb{P}_2(p)$ où \mathbb{P}_2 est un polynôme de degré 2. Calculons $p - (1 - p)$, il vient : $2p - 1 = \phi(p) - \phi(1 - p)$. D'après l'expression de ϕ , nous obtenons :

$$\mu = \frac{e^{4K} - 1}{1 + e^{4K}} [p^4 - (1 - p)^4] + \binom{4}{1} \frac{e^{3K} - e^K}{e^{3K} + e^K} [p^3(1 - p) - p(1 - p)^3]$$

En utilisant l'identité $p^2 - (1 - p)^2 = 2p - 1$, nous obtenons pour tout $p \neq \frac{1}{2}$, l'identité vérifiée par les solutions non triviales :

$$(p^2 + (1 - p)^2) \tanh 2K + 4p(1 - p) \tanh K = 1$$

Il s'agit d'une équation du second degré dont les solutions sont symétriques par rapport au point $(\frac{1}{2}, \frac{1}{2})$. Résolvons cette équation. Posons $u = \exp(2x)$. Comme

$$\forall x \in \mathbb{R}, \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1} = \frac{u - 1}{u + 1}$$

L'équation devient :

$$\frac{1}{2} \frac{u^2 - 1}{u^2 + 1} + \frac{u - 1}{u + 1} - 1 = 0$$

En réduisant au même dénominateur, en simplifiant et en définissant la fonction f , nous obtenons :

$$f(u) = u^3 - 3u^2 - u - 5 = 0$$

Une implémentation numérique nous fournit la courbe de cette fonction dans la figure suivante.

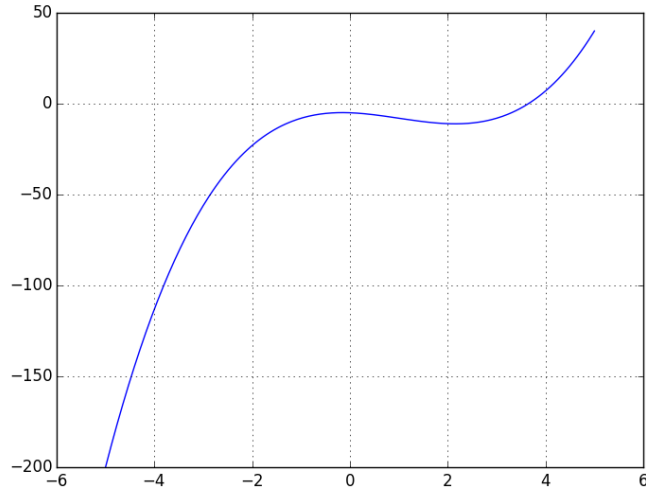


FIGURE 7 – Représentation graphique de la fonction $f : x \rightarrow x^3 - 3x^2 - x - 5$

Les deux zéros de f' sont dans $[-5, 5]$, donc f est monotone (strictement croissante) hors de cet intervalle. Essayons d'appliquer la méthode de Newton-Raphson à partir d'une valeur dans $[-5, 5]$:

```
In [4]: (executing lines 1 to 39 of "Untitled.py")
Pour u0 = -3 on obtient 3.6494359144894917 au bout de 66 itérations
Pour u0 = -2 on obtient 3.649435914489492 au bout de 23 itérations
Pour u0 = -1 on obtient 3.649435914489492 au bout de 22 itérations
Pour u0 = 0 on obtient 3.649435914489492 au bout de 21 itérations
Pour u0 = 1 on obtient 3.649435914489492 au bout de 23 itérations
Pour u0 = 2 on obtient 3.649435914489492 au bout de 21 itérations
Pour u0 = 3 on obtient 3.649435914489492 au bout de 6 itérations
```

Nous trouvons donc $\exp(2K_c) = 3.649$, soit $K_c = 0.647$ puis $\frac{T_c}{J} = \frac{2}{k_B K_c} = 2.239 K.J^{-1}$. Comparons avec le résultat exact d'Onsager qui établit : $\frac{T_c}{J} = 2.269 K.J^{-1}$.

Nous avons donc un écart relatif de :

$$\epsilon = \frac{2.269 - 2/239}{2.269} = 1.32\% < 5\%$$

La méthode par champs de Markov fournit donc un excellent résultat pour la température critique. L'analogie entre les images et les systèmes physiques nous suggère de traiter le problème du traitement des images en s'inspirant de méthodes probabilistes.

2.3 Implémentation numérique des champs de Markov

Nous formulons à présent les méthodes probabilistes que nous avons implémentées numériquement pour les appliquer au traitement d'images.

Dans le cadre de notre étude, nous considérerons des images rectangulaires en niveau de gris (de 0 à 255), de longueur w et de hauteur h . Chaque pixel de l'image est représenté par ses coordonnées (i, j) et une valeur correspondant au niveau de gris du pixel, cette valeur appartenant à $E = \llbracket 0, 255 \rrbracket$. L'image totale peut donc être représentée par un tableau de valeurs de E , de dimension $w \times h$.

Pour chaque pixel, on peut définir un système de cliques tel qu'expliqué en (2.1.2). Dans nos codes informatiques, la présence d'une variable c qui prendra la valeur 4 ou 8 permettra de choisir si l'on considère des voisinages 4 ou 8-connexe, et on considèrera des cliques d'ordre 2.

L'implémentation des algorithmes de traitement d'image comporte trois phases principales :

- le choix d'un champ de Markov adapté
- le tirage d'une configuration selon le champ de Markov choisi
- l'implémentation de l'algorithme qui converge vers une image correcte après un certain nombre de tirages.

Nous décrivons ces trois phases de façon théorique, puis nous analyserons les résultats obtenus avec les différentes méthodes.

2.3.1 Champs de Markov en traitement d'image

Nous présentons ici les champs de Markov les plus utilisés en traitement d'image ainsi que quelques variantes donnant de meilleurs résultats.

- **Le modèle d'Ising :**

Comme nous l'avons indiqué en (2.2), ce modèle n'est applicable qu'à une image comprenant 2 niveaux de gris. Par une transformation affine, on peut associer à ces 2 valeurs les valeurs de l'ensemble $E = \{-1, 1\}$. On rappelle l'énergie de ce modèle :

$$U(x) = - \sum_{c=(s,t) \in C} \beta x_s x_t - \sum_{s \in S} B x_s$$

- **Le modèle de Potts :**

Il s'agit de la généralisation du modèle d'Ising, adaptée à un ensemble E de cardinal N , comme $E = \llbracket 0, 255 \rrbracket$. La différence principale avec le modèle d'Ising est que seuls les potentiels liés aux cliques d'ordre 2 sont définis. Il n'y a pas de terme d'énergie lié aux cliques d'ordre 1, correspondant à un champ magnétique externe. L'énergie de ce modèle est :

$$U(x) = \beta \sum_{c=(s,t) \in C} (\mathbf{1}_{\{x_s \neq x_t\}} - \mathbf{1}_{\{x_s = x_t\}})$$

Un tel modèle tend à créer des zones homogènes de taille d'autant plus grande que β est grand.

- **Le modèle markovien gaussien :**

Ce modèle n'est utilisable que pour les images en niveaux de gris, ce qui est parfaitement adapté à notre étude. On considère, ici encore, des voisinages 4 ou 8-connexes et seulement des cliques d'ordre 2. L'énergie de ce modèle est :

$$U(x) = \beta \sum_{c=(s,t) \in C} (x_s - x_t)^2 + \alpha \sum_{s \in S} (x_s - \mu_s)^2$$

Pour $\beta > 0$, le premier terme quadratique favorise les faibles différences de niveaux de gris, puisqu'il s'agit de minimiser une énergie qui augmente de façon quadratique avec l'écart en niveaux de gris. Le second terme fait intervenir un terme μ_s qui correspond à une image de référence. Si l'on connaît une approximation de l'image que l'on veut obtenir, ou si l'on veut rester proche de l'image initiale, ce terme permet à l'image solution x de ne pas s'éloigner de l'image de référence μ .

2.3.2 Tirage d'une configuration selon le champ de Markov

Après le choix d'un champ de Markov, il faut procéder à la minimisation de son énergie totale. On procède par tirage de configurations. L'idée générale est de tirer pour chaque pixel une valeur aléatoire et de lui attribuer cette valeur si elle permet la diminution de l'énergie totale.

Les algorithmes les plus utilisés pour réaliser ces tirages, l'échantillonneur de Gibbs et l'algorithme de Metropolis, fonctionnent de façon similaire. On procède à n itérations, où l'on choisit aléatoirement un pixel s puis on associe à s une expérience aléatoire en fonction de l'image à l'itération $n - 1$. On met à jour ou non s selon le résultat de l'expérience aléatoire.

Comme il faut balayer l'ensemble des pixels s un grand nombre de fois, on balaye généralement l'ensemble des pixels ligne par ligne et de gauche à droite sans réaliser de tirage aléatoire, pour être sûr que tous les pixels ont été soumis à des mises à jour. L'algorithme cesse après un grand nombre n d'itérations, ou lorsqu'il y a peu de changements de pixels pour une itération.

Pour ces deux algorithmes, on parle d'algorithme de relaxation probabiliste : relaxation car l'algorithme réalise des mises à jour successives des différents pixels, et probabiliste car l'algorithme simule des tirages aléatoires.

Voici le contenu des algorithmes.

- **L'échantillonneur de Gibbs :**

Pour chacune des n itérations de l'algorithme, on balaye l'ensemble des pixels. Pour chaque pixel (noté s) :

- Calcul de la probabilité locale, connaissant la configuration des voisins V_s pour l'image à l'itération $n - 1$:

$$\mathbb{P}(X_s = x_s | V_s) = \frac{\exp(-U_s(x_s | V_s))}{\sum_{a_s \in A_s} \exp(-U_s(a_s | V_s))}$$

- Mise à jour du site par tirage aléatoire selon la loi $\mathbb{P}(X_s = x_s | V_s)$

- **L'algorithme de Metropolis :**

Pour chacune des n itérations de l'algorithme, on balaye l'ensemble des pixels. Pour chaque pixel (noté s) :

- Tirage aléatoire de λ dans E selon une loi uniforme sur E :
- Calcul de la variation d'énergie si la valeur de s , $x_s^{(n-1)}$, est remplacée par λ :

$$\Delta U = U_s(\lambda | V_s^{(n-1)}) - U_s(x_s^{(n-1)} | V_s^{(n-1)})$$

- Si $\Delta U < 0$, alors on met à jour le pixel : $x_s^{(n)} = \lambda$
- Sinon, on met à jour selon la probabilité de succès $p = \exp(-\Delta U)$

2.3.3 Implémentation de l'algorithme de restauration d'image

Après avoir implémenté un algorithme de tirage de configuration, il faut implémenter un algorithme convergeant vers une image solution du problème de minimisation de l'énergie totale. Deux algorithmes sont principalement utilisés dans le cadre du traitement d'image par champs de Markov.

- **Le recuit simulé :**

Le recuit simulé est une méthode classique de minimisation d'énergie fréquemment utilisée en physique. En traitement d'image, l'algorithme consiste en n itérations au cours desquelles on réalise des tirages de configuration tels que décrits en **(2.3.2)**. Cependant, ces tirages ne dépendent plus que de l'énergie de configuration, mais aussi d'une quantité $T^{(n)}$ qui mesure le degré d'aléatoire introduit dans ces tirages, qu'on nomme température et qui décroît à chaque itération. Partant d'une température $T^{(0)}$ assez grande et de l'image à traiter, l'algorithme est le suivant :

Pour chaque itération n ,

- Tirage d'une configuration en remplaçant les énergies $U(x)$ par les quantités $U(x)/T^{(n)}$ pour les tirages de mise à jour des pixels
- Diminution de la température selon une décroissance logarithmique : $T^{(n)} > \frac{c}{\log(2+n)}$

La décroissance logarithmique est nécessaire pour obtenir la convergence en probabilité de l'algorithme vers l'image qui minimise l'énergie. En pratique, pour des images complexes ou grandes, cette décroissance est trop lente et on préfère utiliser une décroissance linéaire ou quadratique, ce qui peut provoquer une convergence vers un minimum seulement local de l'énergie. Cependant, la taille (200×200 pixels) et la simplicité des images que nous traitons nous autorise à utiliser cette décroissance logarithmique. On a observé de légères différences entre la décroissance logarithmique et la décroissance linéaire, ce qui est cohérent avec les notions de minimum local et minimum global de l'énergie.

• **Les modes conditionnels itérés :**

La méthode *Iterated Conditional Mode (ICM)* consiste à tester *toutes* les nuances de gris pour chaque pixel et à mettre à jour avec la configuration qui permet la diminution d'énergie la plus importante. Même si l'on teste à chaque itération toutes les nuances de gris, l'absence du caractère probabiliste (présent dans le recuit simulé) permet à l'ICM de converger bien plus rapidement. En revanche, on ne converge pas forcément vers le minimum global.

L'algorithme ICM est le suivant : Pour chacune des n itérations de l'algorithme, on balaye l'ensemble des pixels. Pour chaque pixel (noté s) :

- $\forall \lambda \in E$, calcul de la variation d'énergie si la valeur de s , $x_s^{(n-1)}$, est remplacée par λ :

$$\Delta U = U_s(\lambda | V_s^{(n-1)}) - U_s(x_s^{(n-1)} | V_s^{(n-1)})$$

- Si $\Delta U < 0$, alors mise à jour le pixel avec λ qui minimise ΔU : $x_s^{(n)} = \lambda$

3 Résultats pratiques et discussion

Avant de pouvoir restaurer des images, il nous a fallu les détériorer. Pour cela, nous avons utilisé un bruit gaussien d'amplitude 50. Il s'agit de simuler une variable aléatoire gaussienne et de modifier la valeur des pixels de l'image en ajoutant la valeur de la variable si cet ajout permet à la valeur du pixel de rester dans $E = \llbracket 0, 255 \rrbracket$.

La *figure 8* montre l'image que nous avons étudiée, ainsi que la même image brouillée avec un bruit gaussien d'amplitude 50. Cette image est composée d'un fond blanc (pixels de valeur $x_s = 255$), d'un carré noir ($x_s = 0$), d'une étoile gris foncé ($x_s = 70$), d'un coeur gris ($x_s = 140$) et d'un rond gris clair ($x_s = 210$).

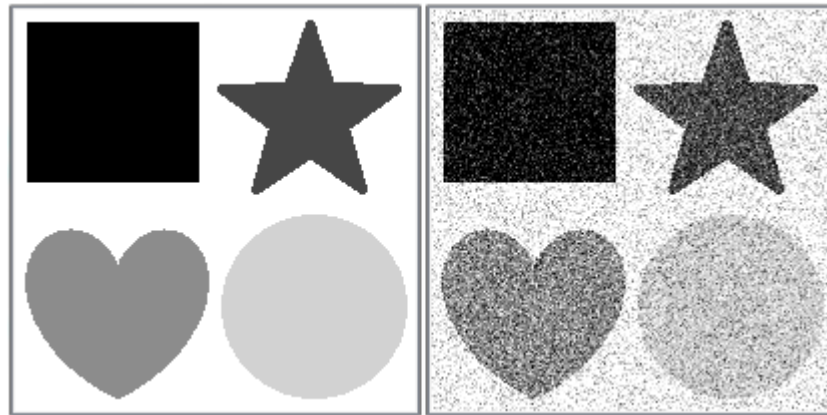


FIGURE 8 – Image initiale sans bruit (à gauche) et avec bruit gaussien d'amplitude 50 (à droite).

Pour évaluer quantitativement la restauration d'image, nous considérons le rapport signal sur bruit (*signal-to-noise ratio*, SNR), qui s'exprime en décibels (dB) et est donné par :

$$SNR = 10 \log \left(\frac{\sum_{s \in S} x_s^2}{\sum_{s \in S} (y_s - x_s)^2} \right)$$

où x_s est la valeur du pixel s de l'image initiale non bruitée, et y_s de même pour l'image débruitée. Plus SNR est grand, moins le bruit détériore l'image initiale. Pour savoir si le débruitage est efficace, il nous faudra comparer les valeurs avec le SNR obtenu avec l'image bruitée. Le tableau 1 recense notamment le SNR pour un bruit gaussien d'amplitude 50.

Amplitude	SNR (dB)
10	63.9516
25	45.2884
50	31.9855
100	20.3203

TABLE 1 – SNR en fonction de l'amplitude du bruit gaussien.

On rappelle qu'on considère des images en niveaux de gris, car le temps de calcul est trop important pour des images en couleur ($E = \llbracket 0, 255 \rrbracket^3$). Même pour des images très simples en niveaux de gris, les résultats obtenus sont loin d'être parfaits, ce qui justifie de limiter notre étude à ce type d'images.

Comme expliqué en (2.3), il existe plusieurs méthodes de traitement d'image par champs de Markov et chacune de ces méthodes possède ses propres paramètres. Pour réaliser un traitement d'image, nous devons choisir les valeurs données à ces paramètres. Le *tableau 2* recense l'ensemble des paramètres avec les choix par défaut effectués, qui sont les choix qui ont donné les meilleurs résultats lors de l'exécution de nos algorithmes.

Paramètres à choisir	Choix possibles	Choix effectué par défaut
Taille $w \times h$ de l'image	$\mathbb{N} \times \mathbb{N}$	200×200
Amplitude du bruit gaussien	\mathbb{N}	50
Algorithme de minimisation de l'énergie	ICM Recuit simulé	Pas de choix par défaut
Nombre d'itérations de la minimisation	\mathbb{N}	10
Température initiale pour recuit	\mathbb{R}^+	100
Evolution de la température pour recuit	Logarithmique Linéaire Quadratique	Oui Non Non
Algorithme de tirage de configuration	Gibbs Metropolis	Non Oui
Nombre d'itérations du tirage	\mathbb{N}	10
Voisinage	4-connexe 8-connexe	Non Oui
Potentiel	Potts Markovien gaussien	Pas de choix par défaut
Paramètre β pour Potts	\mathbb{R}^+	50
Paramètre β pour Markovien gaussien	\mathbb{R}^+	50
Paramètre α pour Markovien gaussien	\mathbb{R}^+	50

TABLE 2 – Paramètres des différentes méthodes et choix par défaut effectués

Les principales différences dans nos résultats résident dans le choix de l'algorithme de minimisation et dans le choix du modèle de potentiel.

On va appliquer à l'image brouillée les différents algorithmes décrits précédemment, pour différentes valeurs des paramètres. On va d'abord étudier le modèle de Potts, puis le modèle markovien gaussien.

3.1 Première série d'essais

On considère dans un premier temps que l'on ne connaît rien de l'image que l'on doit obtenir. On implémente les algorithmes tels que décrits en (2.3) et on les exécute.

3.1.1 Le modèle de Potts

- Avec recuit simulé :

Étant donné la taille de l'image, un balayage de l'image bruitée montre que toutes les nuances de gris sont présentes sur l'image : $E = \llbracket 0, 255 \rrbracket$. On exécute alors le recuit simulé avec un algorithme de Metropolis qui tire $\lambda \in E$.

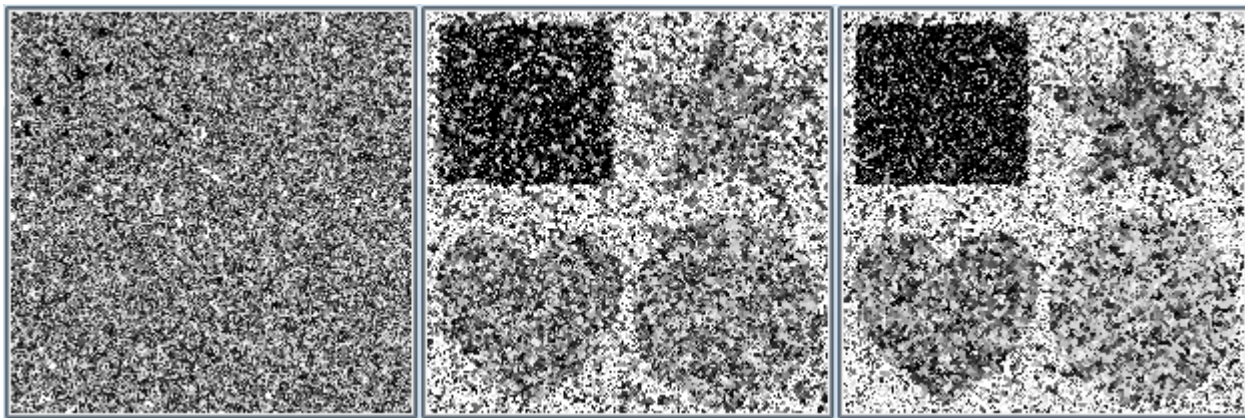


FIGURE 9 – Image traitée par recuit simulé et modèle de Potts pour $\beta \in \{50, 100, 500\}$ (de gauche à droite).

β	SNR (dB)
50	2.73104
100	8.48355
500	11.3079
10000	11.6187

TABLE 3 – SNR en fonction de β .

On constate que les résultats s'améliorent lorsque β augmente, avec un phénomène de saturation pour β grand ($\beta \approx 10^4$). Cependant, les résultats sont pires que l'image bruitée elle-même ! L'image étant très bruitée, l'algorithme tend à remplacer les pixels par n'importe quelle valeur entre 0 et 255, ce qui ne permet pas de débruiter l'image.

- Avec ICM :

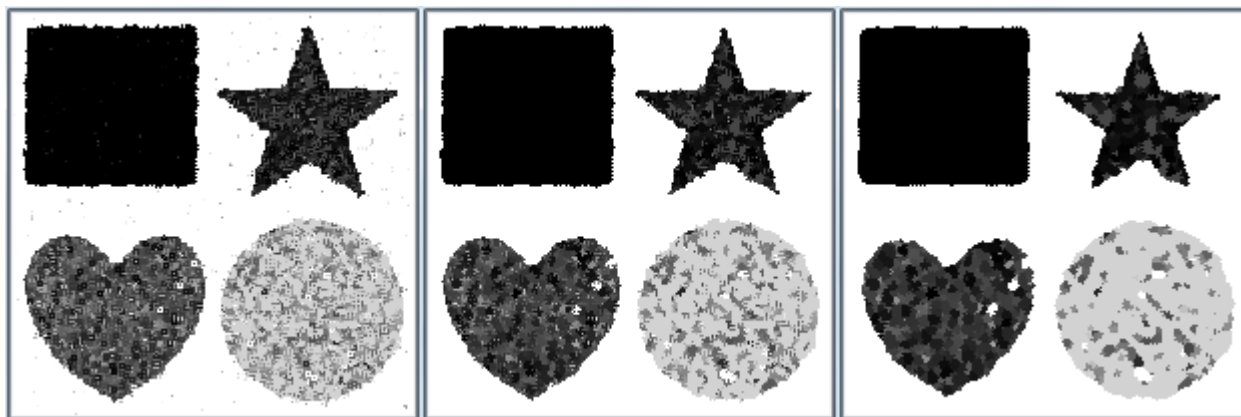


FIGURE 10 – Image traitée par ICM et modèle de Potts pour 1, 2 et 4 itérations (de gauche à droite).

Itérations	SNR (dB)
1	29.2689
2	27.3294
4	25.5964
10	23.1687

TABLE 4 – SNR en fonction du nombre d’itérations de l’ICM.

Les résultats sont très bons après quelques itérations. Même après une seule itération, le bruit a presque entièrement disparu. Toutefois, l’ICM tend à rogner les figures, d’où un SNR qui diminue lorsque le nombre d’itérations augmente. Le SNR ne varie pas lorsqu’on change β , donc β ne semble pas avoir d’influence sur le résultat.

3.1.2 Le modèle markovien gaussien

On considère désormais le modèle markovien gaussien, et on teste les deux algorithmes de minimisation d’énergie avec et sans attache aux données initiales.

On constate que le modèle markovien gaussien floute les images. En effet, il tend à effectuer une sorte de moyenne locale. Or, l’image étant brouillée par un bruit qui prend toutes les valeurs de $E = \llbracket 0, 255 \rrbracket$, le moyennage homogénéise les différentes zones de gris en un gris moyen. Ainsi, même si le tableau 5 montre que le SNR est meilleur que sur l’image brouillée, le rendu visuel et la disparition du bruit sur la zone blanche ne sont pas bons.

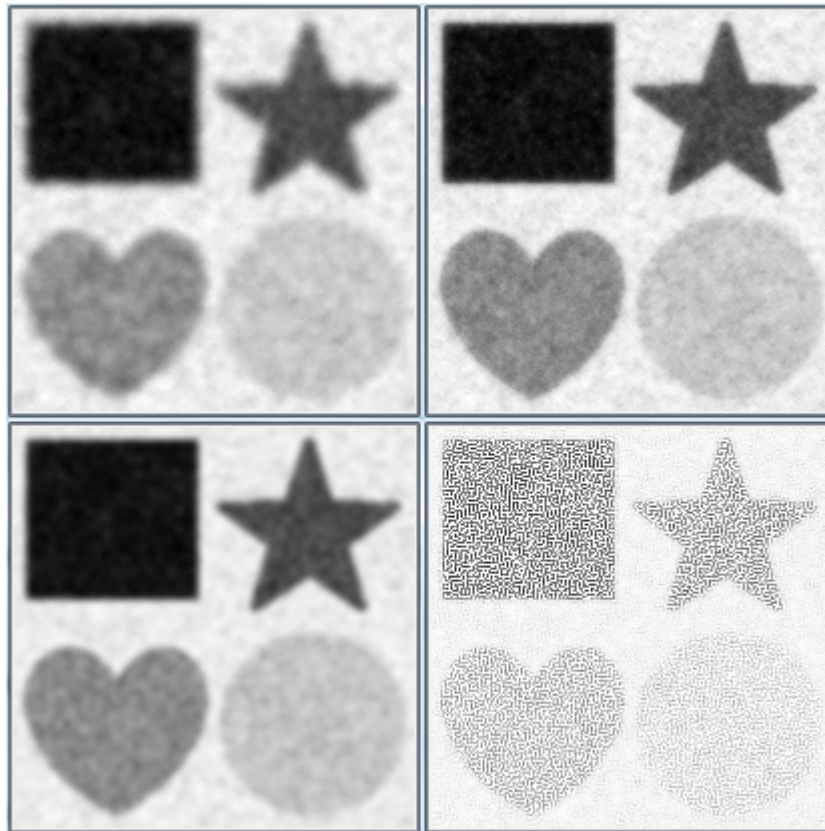


FIGURE 11 – Image traitée par recuit simulé (en haut) et ICM (en bas) et modèle markovien gaussien sans (à gauche) et avec (à droite) attache aux données.

Méthode	SNR (dB)
Recuit sans attache	35.2715
Recuit avec attache	38.9796
ICM sans attache	38.0792
ICM avec attache	17.2107

TABLE 5 – SNR en fonction de la méthode utilisée.

Cette première série d'essais montre que, mis à part pour l'ICM avec modèle de Potts, les résultats de restauration d'image sont mauvais. Pour le recuit simulé avec Potts, les mauvais résultats s'expliquent par le tirage aléatoire sur les 255 nuances de gris, alors que dans l'image initiale non bruitée, seules 5 nuances sont présentes.

3.2 Seconde série d'essais

On réalise désormais les tirages de configuration seulement parmi les nuances présentes sur l'image initiale non bruitée : $\lambda \in E = \{0, 70, 140, 210, 255\}$.

3.2.1 Le modèle de Potts

- Avec recuit simulé :

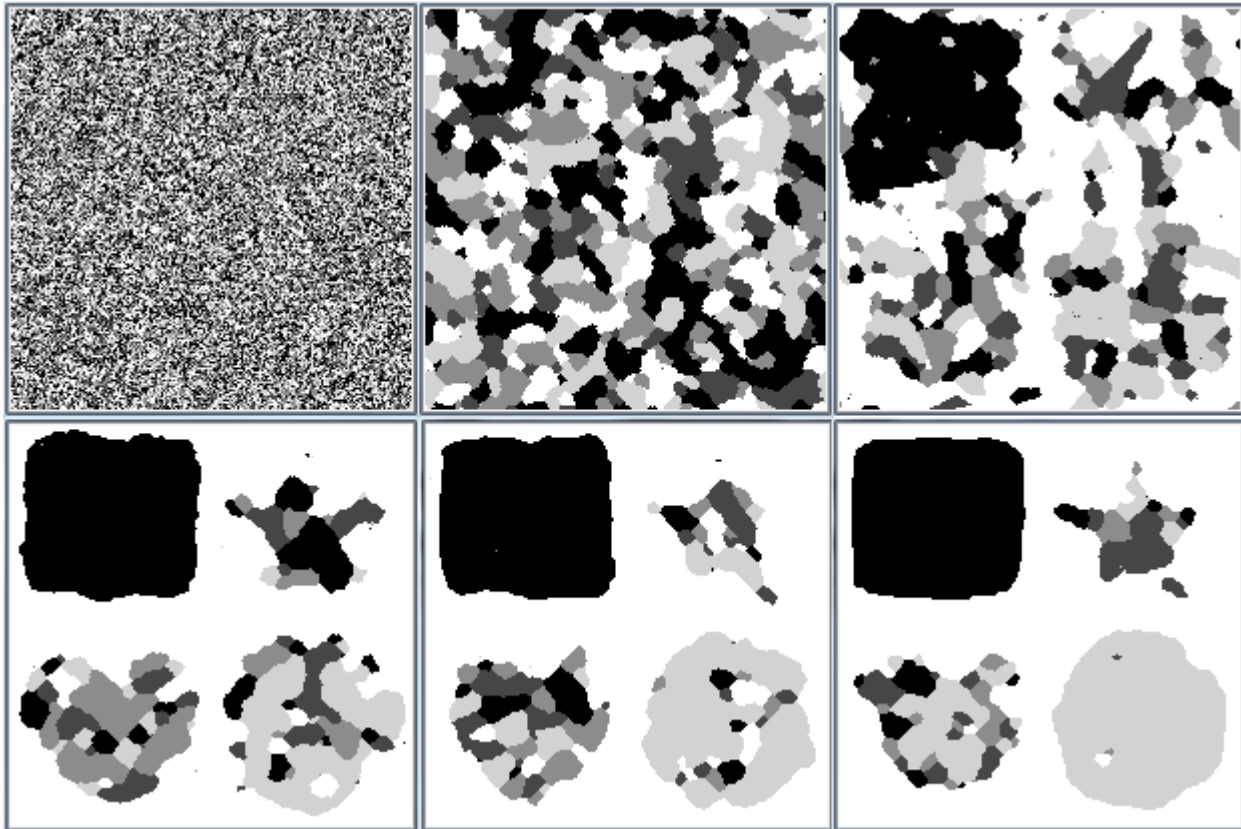


FIGURE 12 – Image traitée par recuit simulé et modèle de Potts pour $\beta \in \{5, 25, 35, 50, 100, 500\}$ (de gauche à droite puis de haut en bas).

β	SNR (dB)
5	3.38126
25	3.18447
35	14.8631
50	22.8984
100	23.9018
500	25.9764

TABLE 6 – SNR en fonction de β .

Les résultats sont bien meilleurs que lorsqu'on ne connaît rien de l'image initiale, même si le SNR reste inférieur à celui de l'image bruitée. On constate ici l'importance du paramètre β : plus β est grand, plus la taille des zones homogènes augmente. On a cependant toujours un phénomène de saturation pour β grand. L'image converge vers un état proche de l'état initial.

- Avec ICM :

Les résultats deviennent excellents : l'image obtenue est quasiment l'image initiale et le SNR est meilleur que pour l'image bruitée. Le SNR montre aussi que β n'a toujours pas d'influence.

3.2.2 Le modèle markovien gaussien

Le modèle markovien gaussien donne aussi de biens meilleurs résultats. Les résultats sont légèrement meilleurs pour l'ICM que pour le recuit simulé. L'image obtenue lorsqu'on ajoute l'attache aux données initiales, qui permet notamment de corriger les effets néfastes de l'ICM sur les bords, est extrêmement proche de l'image initiale non bruitée.

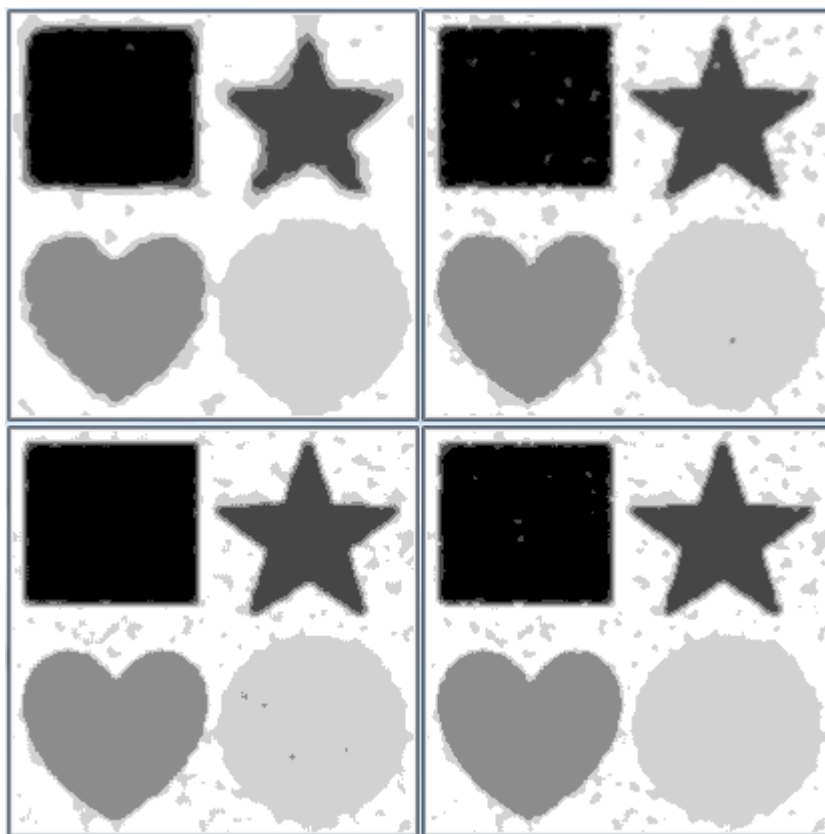


FIGURE 13 – Image traitée par recuit simulé (en haut) et ICM (en bas) et modèle markovien gaussien sans (à gauche) et avec (à droite) attache aux données et connaissant les nuances de gris présentes dans l'image initiale non brouillée.

Méthode	SNR (dB)
Recuit sans attache	40.3498
Recuit avec attache	44.4864
ICM sans attache	43.9946
ICM avec attache	44.7768

TABLE 7 – SNR en fonction de la méthode utilisée.

- Avec recuit simulé :

L'influence de β et le phénomène de saturation sont conservés : plus β est grand, meilleure est la restauration. Le SNR est meilleur que pour l'image bruitée.

Des modifications du paramètre α ont été réalisées sur les tests avec attache aux données, cependant les résultats étant déjà très bons pour α petit, α ne semble pas avoir une grande influence. On rappelle qu'en théorie, plus α est grand, plus l'attache aux données initiales est importante.

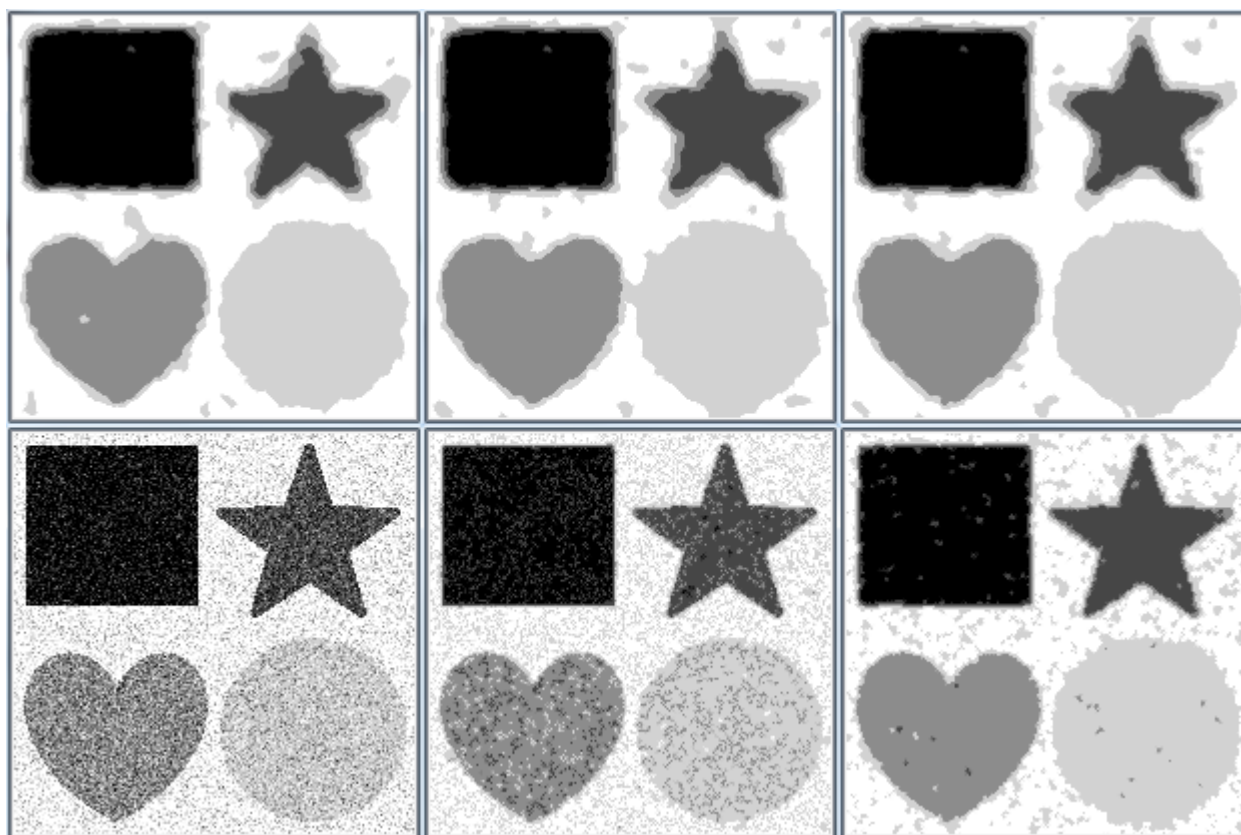


FIGURE 14 – Image traitée par recuit simulé et modèle markovien gaussien sans (en haut) et avec (en bas) attache aux données pour $\beta \in \{1, 10, 25\}$ (de gauche à droite).

β	SNR (dB)
1	40.7834
10	40.0753
25	40.0668
5000	39.6827

TABLE 8 – SNR sans attache en fonction de β .

β	SNR (dB)
1	32.5507
10	38.673
25	43.5519
5000	43.4116

TABLE 9 – SNR avec attache en fonction de β .

- Avec ICM :

On réalise les mêmes constats qu’avec le recuit simulé. Le meilleur SNR parmi tous les essais est obtenu ici, pour $\beta = 20$ (tableau 10).

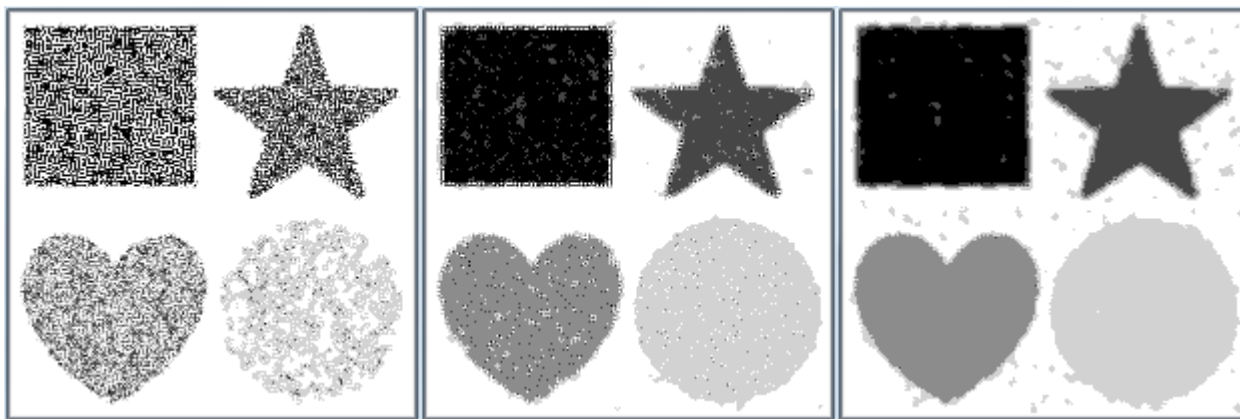


FIGURE 15 – Image traitée par ICM et modèle markovien gaussien avec attache aux données pour $\beta \in \{1, 10, 25\}$ (de gauche à droite).

β	SNR (dB)
1	19.7154
10	39.8086
20	46.9805
25	46.474
1000	43.4296
5000	43.3621

TABLE 10 – SNR en fonction de β .

3.3 Limites des méthodes markoviennes

L'image traitée en (3.1) et (3.2) comportait des formes séparées avec des nuances de gris éloignées. Nous étudions désormais une image composée d'un dégradé de gris pour montrer les limites des méthodes markoviennes.

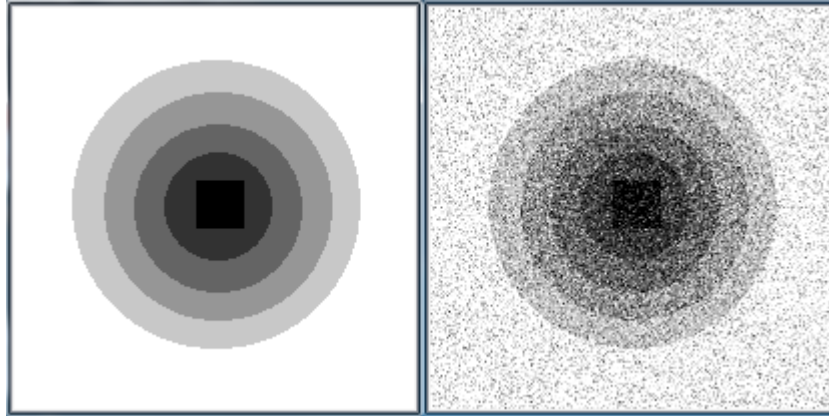


FIGURE 16 – Image initiale sans bruit (à gauche) et avec bruit gaussien d'amplitude 50 (à droite).

On applique à l'image les méthodes de recuit simulé et l'ICM avec β grand, modèle markovien gaussien avec attache aux données et connaissance des nuances initiales car ce sont les choix qui ont donné les meilleurs résultats jusqu'à présent. Les résultats sont représentés en *figure 17*.

On constate que le recuit simulé avec modèle de Potts donne un résultat analogue à celui de la *figure 12* pour $\beta = 25$. Lorsqu'on augmente β , le résultat ne s'améliore pas et reste proche de celui en *figure 17*.

L'ICM avec Potts tend à détériorer les bords comme constaté en (3.1) et (3.2). Toutefois, les nuances de gris étant proches, cela provoque la disparition de certaines.

Le modèle markovien gaussien donne de très bons résultats quel que soit le potentiel. Le SNR obtenu est le double de celui de l'image bruitée en *figure 16*, ce qui est même meilleur que pour les résultats de la *figure 15*.

Pour conclure cette étude et montrer les limites des méthodes markoviennes sur des nuances proches, nous avons testé les différents algorithmes sur une photo avec les 255 nuances de gris. La méthode donnant le meilleur SNR est le modèle markovien gaussien avec attache aux données, bien que ce modèle ait pour conséquence de flouter l'image.

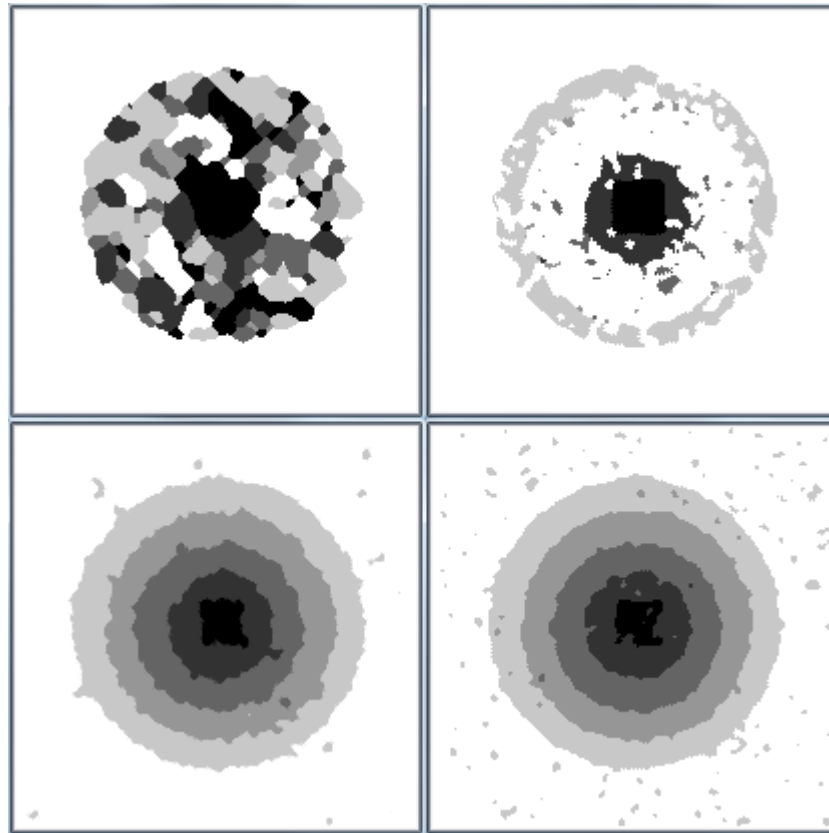


FIGURE 17 – Image traitée par recuit simulé (à gauche) et ICM (à droite) pour le modèle de Potts (en haut) et le modèle markovien gaussien avec attache aux données (en bas) pour β grand.



FIGURE 18 – Image initiale sans bruit (à gauche), avec bruit gaussien d’amplitude 50 (au milieu) et image traitée par modèle markovien gaussien avec attache aux données (à droite) pour β grand.

4 Conclusion

Débruiter une image est une étape importante dans plusieurs domaines avancés comme la médecine ou la cartographie. Notre projet présente une méthode de débruitage d'images en noir et blanc reposant sur la théorie des champs de Markov et sur les modèles physiques d'Ising et de Potts. L'étude théorique du modèle markovien met en avant la nécessité de choisir la bonne approche de minimisation, ainsi que de l'algorithme d'échantillonnage. Nous avons été confrontés au problème de calibrage de différents paramètres. Les résultats montrent l'efficacité de l'algorithme ICM avec les potentiels issus du modèle markovien gaussien, qui, par comparaison aux autres modèles, présente une meilleure restitution de l'image détériorée. La restauration est d'autant meilleure que l'on connaît des informations sur l'image initiale.

5 Remerciements

Nous remercions tout d'abord M. Alain Maruani pour l'encadrement qu'il nous a apporté lors de nos tâtonnements en quête de projet, pour avoir transformé nos idées et intérêts en pistes de recherche et pour son suivi, à la fois pendant et en dehors des séances tout au long de ces six semaines.

Nos remerciements vont également à MM. Bernard Lapeyre et Jean-François Delmas qui nous ont orientés vers de nouvelles pistes de réflexion lorsque nous étions dans une impasse.

Enfin, un grand merci à M. Pascal Monasse pour son aide précieuse, sans laquelle nos codes informatiques n'auraient pas fonctionné.

Références

- [1] M. Sigelle et A. Maruani, *Critical Properties of the 2D Ising Model : A markovian approach*, dans *Maximum Properties and Bayesian Methods*, pp. 389 – 392, Kluwer Academic Publishers, 1992.
- [2] F. Chevoir, *Cours de Physique Statistique*
- [3] M. Sigelle et F. Tupin, *Champs de Markov en Traitement d'Image*, École Nationale Supérieure des Télécommunications, 1999.
- [4] G. Weisbuch, *Dynamique des systèmes complexes*, Chapitres 8 – 9, CNRS/InterEditions, 1989.