

Méthodes statistiques pour la segmentation d'images - Partie 3

MAT5011: Modèles de Markov en Signal et Image

Réalisé pour le 12 novembre 2021

par

Lorenzo HERMEZ

Coordinateurs : Wojciech PIECZYNSKI & Clément FERNANDES

Table des matières

Introduction	1
1 Champs de Markov cachés	1
1.1 Définition du modèle	1
1.2 Échantillonneur de Gibbs	2
2 Segmentation supervisée	2
3 Segmentation non supervisée	3
Conclusion	6

Introduction

Lors du TP précédent, les modèles utilisés pour la segmentation d'images supposent que les pixels étaient indépendants les uns des autres. Toutefois, cette hypothèse est plus que discutable dans le cas d'images. C'est pour cela que l'on modélisera nos images par des champs de Markov cachés (on aurait pu également modéliser les images par des chaînes de Markov cachées). Le but de ce TP est donc d'analyser le gain de précision obtenu lors de la segmentation par le critère du MPM en choisissant une modélisation par les champs de Markov.

1 Champs de Markov cachés

1.1 Définition du modèle

Dans ce modèle, la loi de X s'écrit de la manière suivante :

$$\mathbb{P}_X[x] = \gamma e^{-\sum_{(s,t)} \varphi(x_s, x_t)}$$

où $\sum_{(s,t)} \varphi(x_s, x_t)$ est prise sur tous les couples de pixels voisins (horizontaux et verticaux dans le cas des 4 plus proches voisins).

De plus, la fonction φ est définie par :

$$\varphi(x_s, x_t) = \alpha - 2\alpha \mathbb{1}\{x_s = x_t\}$$

De cette manière, la loi du couple (X, Y) est donnée par :

$$\mathbb{P}(x, y) = \gamma e^{-\sum_{(s,t)} \varphi(x_s, x_t) + \sum_{s \in S} \log(f_{x_s}(y_s))}$$

et on peut ainsi définir la loi a posteriori :

$$\mathbb{P}_{X, Y=y}[x] = \frac{\mathbb{P}(x, y)}{\sum_{x \in \Omega^N} \mathbb{P}(x, y)} = \frac{\gamma}{p(y)} e^{-\sum_{(s,t)} \varphi(x_s, x_t) + \sum_{s \in S} \log(\mathbb{P}(y_s | x_s))}$$

1.2 Échantillonneur de Gibbs

L'échantillonneur de Gibbs est l'un des algorithmes les plus utilisés pour la simulation de réalisations de X . Rappelons les étapes de cet algorithme :

1. On se donne, tout d'abord, une configuration initiale $x^0 = (x_s^0)_{s \in S}$
2. Puisque l'on connaît la loi de X_1 selon tous les autres X_s , on effectue un tirage selon cette loi et on note x_1^1 la classe obtenue. On effectue la même chose pour X_2 , en prenant en compte la nouvelle valeur de X_1^1 . On fait de même pour tous les $s \in S$. On obtient alors une nouvelle image x^1 . On recommence de manière à obtenir une suite de réalisations de X , notée x^0, x^1, \dots, x^n qui converge en loi vers $\mathbb{P}_X[x]$.

2 Segmentation supervisée

La mise en place de l'échantillonneur de Gibbs est l'une des étapes les plus importantes de ce TP. En effet, la simulation de réalisations de X ou de X sachant $Y = y$ est primordiale et permet de calculer tous les paramètres du modèle.

Tentons de simuler des réalisations de X dans le cas où $\alpha = 1$ pour voir à quoi ressemblent nos images. Profitons également de ces simulations pour segmenter ces mêmes réalisations lorsqu'un des bruits vient bruiteur notre image. On obtient le résultat suivant :

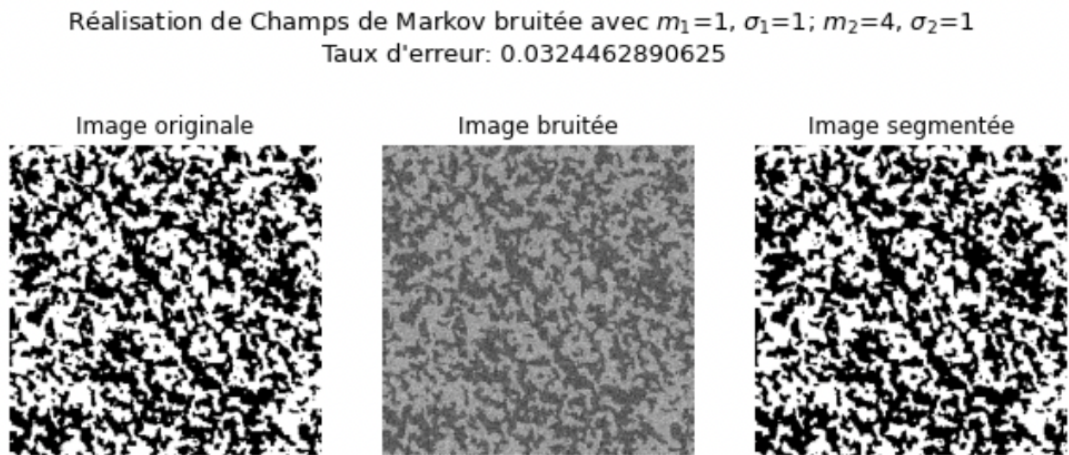


FIGURE 1 – Réalisation de la loi de X dans le cas où $\alpha = 1$, puis la même réalisation bruitée d'un bruit avec $m_1 = 1, \sigma_1 = 2, m_2 = 4$ et $\sigma_2 = 1$ pour paramètres.

On remarque que la segmentation de l'image fonctionne très bien puisque l'on obtient

le taux d'erreur le plus bas obtenu durant l'ensemble de ce TP. On pouvait s'y attendre car la connaissance des pixels voisins apportent beaucoup plus d'informations que dans le cas indépendants.

Notons les taux d'erreur obtenus en fonction des différents bruits appliqués dans un tableau :

Bruit appliqué	Taux d'erreur obtenu
$\mathcal{N}(1, 1^2)$ et $\mathcal{N}(4, 1^2)$	0.032
$\mathcal{N}(1, 1^2)$ et $\mathcal{N}(2, 1^2)$	0.264
$\mathcal{N}(1, 1^2)$ et $\mathcal{N}(1, 3^2)$	0.108

TABLE 1 – Taux d'erreur moyens obtenus en fonction des différents bruits appliqués sur 10 segmentations

3 Segmentation non supervisée

Dans cette section, on ne connaît plus les différents paramètres des bruits et du modèle. C'est ce qui se passe en pratique, nous n'avons la valeurs des paramètres que dans très rares cas. Ainsi, la mise en place d'un algorithme d'estimation des paramètres et de rigueur et c'est pour cela que nous avons mis en place l'algorithme EM-Gibbsien.

Cet algorithme, de très grande complexité, nous permet d'estimer les paramètres en utilisant des méthodes de Monte Carlo et la pseud-vraisemblance, i.e. la vraisemblance conditionnelle à $Y = y$

Dans l'algorithme EM-Gibbsien, il nous faut estimer les paramètres du bruit et le paramètre α . Après un long moment, on obtient le résultat de segmentation suivant :

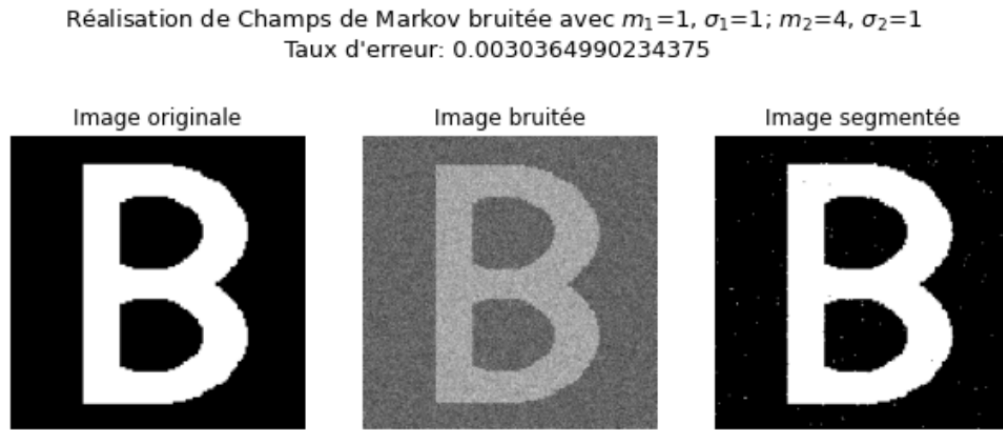


FIGURE 2 – Segmentation de l'image beee2.bmp bruitée par le bruit n°1

On fait de même pour toutes les images et tous les bruits. Dans un premier temps, on remarque que la segmentation s'effectue parfaitement pour les images bruitées avec un faible bruit. Cependant, ce n'est pas le cas pour les 2 autres bruits. Il faut jouer sur les paramètres d'itération afin d'obtenir les bonnes estimations.

Bruits	alfa2	beee2	country2	veau2	zebre2	Méthode utilisée
$\mathcal{N}(1, 1^2)$ et $\mathcal{N}(4, 1^2)$	0.083	0.076	0.071	0.067	0.068	K-Means
	0.055	0.059	0.063	0.067	0.066	MPM
	0.055	0.06	0.063	0.067	0.066	EM + MPM
	0.055	0.058	0.063	0.066	0.065	SEM + MPM
	0.003	0.010	0.006	0.004	0.03	EM + Champs Markov
$\mathcal{N}(1, 1^2)$ et $\mathcal{N}(2, 1^2)$	0.353	0.338	0.323	0.309	0.311	K-Means
	0.22	0.248	0.278	0.307	0.302	MPM
	0.418	0.382	0.369	0.309	0.333	EM + MPM
	0.380	0.365	0.336	0.310	0.343	SEM + MPM
	0.231	0.254	0.281	0.298	0.334	EM + Champs Markov
$\mathcal{N}(1, 1^2)$ et $\mathcal{N}(1, 3^2)$	0.5	0.501	0.501	0.499	0.5	K-Means
	0.154	0.175	0.204	0.265	0.236	MPM
	0.152	0.176	0.205	0.265	0.236	EM + MPM
	0.152	0.175	0.202	0.267	0.232	SEM + MPM
	0.164	0.184	0.213	0.259	0.303	EM + Champs Markov

TABLE 2 – Tableau récapitulatif des taux d’erreur moyens (sur 100 itérations) entre la méthode du K-Means, le critère du MPM avec ou sans estimation de θ par 2 méthodes différentes, le critère du MPM par les champs de Markov cachés.

On remarque une nette amélioration du taux d’erreur dans le cas des champs de Markov cachés. Cependant, dans le cas de bruits plus forts, la segmentation est plus compliquée en temps raisonnable. C’est pour cela que les taux d’erreur dans le tableau sont élevés et non pas parce la méthode est mauvaise.

Conclusion

Nous avons mis en oeuvre différentes méthodes tout au long du TP pour la segmentation d'images. Nous avons utilisés des méthodes très simples et d'autres plus compliquées (dans le cas des champs de Markov cachés). On peut remarquer que la modélisation globale améliore grandement les résultats de la segmentation. On pourrait observer une amélioration en modélisant nos images par des chaînes de Markov cachées. L'estimation des paramètres est un pilier dans la segmentation d'images et ce n'est pas une tâche facile. Il faudrait d'autant plus affiner le nombre d'itérations afin d'obtenir une bonne estimation de θ et ainsi avoir une bonne segmentation. L'usage de calculateur cloud pourrait être une solution au problème de complexité.