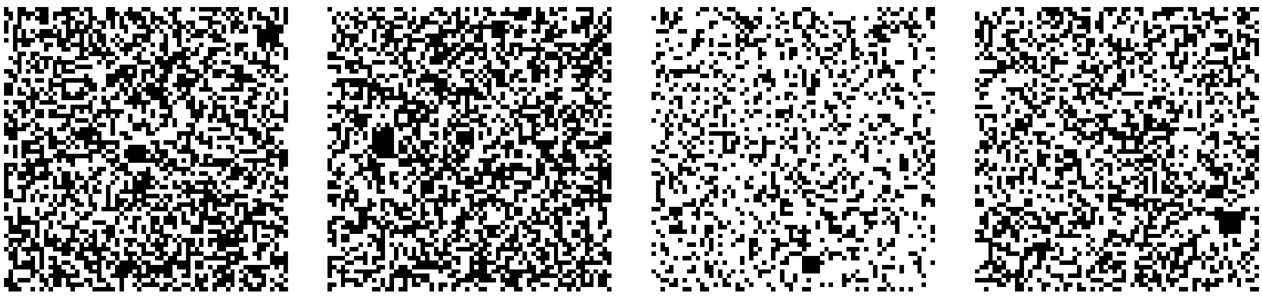
TP3 – Perception des formes dans une image

Introduction

Dans une image binaire de synthèse, on considère les pixels comme des variables aléatoires iid. Un pixel reçoit la valeur 0 (pixel noir) avec la probabilité p ou la valeur 1 (pixel blanc) avec la probabilité $1-p$ (loi de Bernouilli de paramètre p). Malgré l'indépendance entre pixels, des agglomérats de pixels noirs, en forme de carrés, peuvent apparaître par le fait du hasard. La probabilité $\mathcal{P}(p,t)$ pour qu'un carré de côté t soit noir est égale à p^{t^2}. Elle diminue très rapidement si les carrés deviennent plus grands, c'est-à-dire lorsque t croît, ou si les pixels noirs se raréfient, c'est-à-dire lorsque p décroît. Or, un carré noir se distingue d'autant mieux qu'il est plus grand ou que l'image est plus claire (cf. figure 1). Cet exemple très simple illustre le principe général de la perception des formes dans une image : une forme est d'autant plus perceptible qu'elle est moins probable.



$\mathcal{P}(0.5,4) = 1.5\,10^{-5}$ $\mathcal{P}(0.5,5) = 2.9\,10^{-8}$ $\mathcal{P}(0.3,4) = 4.3\,10^{-9}$ $\mathcal{P}(0.4,5) = 1.1\,10^{-10}$

FIGURE 1 – Chacune de ces images binaires de taille 64×64, tirées selon une loi de Bernouilli de paramètre p, contient un et un seul carré noir de côté t. La probabilité d'apparition $\mathcal{P}(p,t) = p^{t^2}$ du carré décroît de gauche à droite, ce qui le rend de plus en plus perceptible : notre œil est attiré par les événements les moins probables !

Lancez le script `carres_noirs.m`, qui simule une séquence d'images binaires aléatoires de taille 64×64 tirées selon une loi de Bernouilli de paramètre p, jusqu'à ce qu'une de ces images contienne au moins un carré noir de côté t. Ce script teste plusieurs valeurs de t, à p fixé. Attention : la recherche est de plus en plus longue, puisque la probabilité d'apparition $\mathcal{P}(p,t) = p^{t^2}$ d'un carré noir décroît fortement lorsque t croît. Néanmoins, plus l'attente est longue, plus la forme « saute aux yeux » dans l'image affichée.

Le script `carres_noirs.m` affiche également, sous la forme d'une pseudo-image en niveaux de gris, la probabilité $\mathcal{P}(p,t) = p^{t^2}$ pour qu'un carré soit noir. Des « courbes de niveau » correspondant aux valeurs 0.1, 0.01, 0.001 et 0.0001 de $\mathcal{P}(p,t)$ sont superposées à cette pseudo-image, de façon à montrer la décroissance très rapide de cette probabilité lorsque t croît ou lorsque p décroît.

Comme cela a déjà été vu dans le TP1, les variables aléatoires que contituent les pixels d'une image naturelle ne sont pas iid, mais sont au contraire fortement corrélées. C'est la raison pour laquelle les formes perceptibles sont beaucoup plus courantes dans une image naturelle que dans une image aléatoire. En revanche, il s'agit en général de formes approchées. Il est donc nécessaire de donner une définition plus souple de la notion de forme.

Dans une image binaire aléatoire de taille $T \times T$ tirée selon une loi de Bernouilli de paramètre p, la probabilité pour qu'un carré de côté t, c'est-à-dire constitué de $N = t^2$ pixels, contienne *exactement* $n \leq N$ pixels noirs, est donnée par la loi binomiale $\mathcal{B}(p,N,n) = C_N^n\,p^n(1-p)^{N-n}$, où C_N^n désigne le nombre de combinaisons de n éléments parmi N (fonction `nchoosek` de Matlab). Par conséquent, la probabilité pour qu'un tel carré contienne *au moins* n pixels noirs s'écrit :

$$\sum_{l=n}^{N} \mathcal{B}(p,N,l) = \sum_{l=n}^{N} C_N^l\,p^l(1-p)^{N-l} \qquad (1)$$

Exercice 1 : détection de pixels voisins ayant des gradients parallèles

Détecter les « alignements » dans une image consiste à en faire un croquis constitué de segments de droites. De tels segments se situent là où le module du gradient ∇I du niveau de gris est élevé. Pour détecter les alignements, il faut donc commencer par calculer le gradient du niveau de gris, par exemple par « différences finies centrées » (axe des abscisses orienté vers le bas, axe des ordonnées vers la droite, longueurs en pixels) :

$$\nabla I(i,j) \approx \left[\frac{I(i+1,j)-I(i-1,j)}{2}, \frac{I(i,j+1)-I(i,j-1)}{2}\right]^\top \tag{2}$$

Cette expression n'est pas utilisable pour les pixels du bord. Pour la première ligne, on doit la rempacer par :

$$\nabla I(1,j) \approx \left[I(2,j)-I(1,j), \frac{I(1,j+1)-I(1,j-1)}{2}\right]^\top \tag{3}$$

Une fois le gradient du niveau de gris calculé en chaque pixel de l'image, seuls ceux où la norme du gradient est supérieure à un seuil sont sélectionnés. Parmi ces pixels, on cherche à constituer des ensembles E tels que :

- l'ensemble E soit connexe, au sens des « 8 plus proches voisins » ;
- les gradients $G_1 = \nabla I(i_1, j_1)$ et $G_2 = \nabla I(i_2, j_2)$ en deux pixels de E aient des directions quasi-parallèles :

$$|G_1 \cdot G_2| \geq \|G_1\| \, \|G_2\| \, \cos\alpha \tag{4}$$

ce qui signifie que $\alpha > 0$ constitue la valeur limite de l'angle formé par les vecteurs G_1 et G_2.

Complétez la fonction `recursion`, qui permet de construire un ensemble E à partir d'un « germe » par appels récursifs. Complétez ensuite le script `exercice_1.m`, où deux paramètres supplémentaires `card_min` et `card_max` permettent de définir les valeurs extrêmes du cardinal de E.

Remarque

Pour cet exercice et le suivant, il est conseillé de consulter la documentation des fonctions `ind2sub` et `sub2ind`.

Exercice 2 : détection des alignements dans une image

Si les gradients du niveau de gris constituaient des variables aléatoires iid, alors la probabilité p pour qu'un gradient fasse un angle inférieur à α, relativement à une direction de référence, vaudrait :

$$p = \frac{2\,\alpha}{2\,\pi} = \frac{\alpha}{\pi} \tag{5}$$

Soit E un des ensembles de pixels constitués à l'étape précédente. On appelle R le plus petit rectangle englobant de E. Ce rectangle, de cardinal N, contient généralement d'autres pixels que ceux de E. Parmi les pixels de E, on compte ceux où le gradient ∇I fait, avec la direction du petit axe de R, un angle inférieur à α. La probabilité pour que cette propriété soit vérifiée en n pixels de R *exactement* suit une loi binomiale. La probabilité pour qu'elle soit vérifiée en n pixels de R *au moins* est donc donnée par (1), p étant défini par (5). Le critère permettant de sélectionner les ensembles E formant des alignements détectables s'écrit donc :

$$\sum_{l=n}^{N} C_N^l \, p^l (1-p)^{N-l} < \epsilon \tag{6}$$

où n désigne le nombre de pixels de E où le gradient est parallèle au petit axe de R, et où $\epsilon \ll 1$ est un seuil.

Complétez le script `exercice_2.m` de façon à mettre en œuvre cette méthode probabiliste de détection d'alignements. Une fois mis au point, vous pourrez tester ce script sur une image simulée, où les niveaux de gris sont tirés selon une loi uniforme. Enfin, vous pourrez le tester sur l'image `Morlaix.png` (attention : le calcul est un peu long), ou sur toute autre image de votre choix.