

## 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 2). 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.

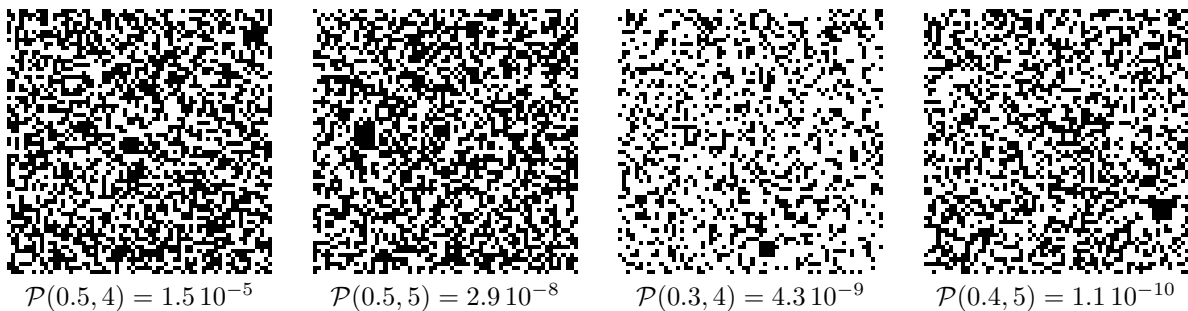


FIGURE 1 – Chacune de ces images binaires de taille  $64 \times 64$ , tirées selon une loi de Bernouilli de paramètre  $p$ , contient un et un seul carré noir de côté  $t$ . La probabilité d'apparition  $\mathcal{P}(p, t) = p^{t^2}$  du carré décroît de gauche à droite, ce qui le rend de plus en plus 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  $64 \times 64$  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  $t$ , pour  $p = 0,5$  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 constituent 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  $N$ . Par conséquent, la probabilité pour qu'un tel carré contienne *au moins*  $n$  pixels noirs s'écrit :

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

La somme (1) peut s'écrire à l'aide de la fonction `binocdf` de Matlab (doc `binocdf`).

## Exercice 1 : détection de pixels voisins ayant des gradients parallèles

Détecter les « alignements » dans une image consiste à faire une esquisse constituée de segments de droites. De tels segments se situent là où le module  $\|\nabla I\|$  du gradient du niveau de gris est élevé. Pour détecter les alignements, il faut donc commencer par calculer le gradient du niveau de gris, par exemple avec la fonction **gradient** (axe des abscisses orienté vers la droite, axe des ordonnées vers le bas, unité de longueur = pixel).

Une fois le gradient du niveau de gris calculé, seuls les pixels où la norme du gradient est supérieure à un seuil sont sélectionnés. Parmi ces pixels, on cherche à constituer des ensembles  $E$  tels que :

- chaque 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 voisins de  $E$  aient des directions parallèles, soit  $|G_1 \cdot G_2| \geq \|G_1\| \|G_2\| \cos \alpha$ , où  $\alpha > 0$  constitue la limite de l'angle formé par les vecteurs  $G_1$  et  $G_2$ .

Après avoir lu attentivement le script **exercice\_1.m**, complétez la fonction **recursion**, qui permet de construire, par appels récursifs, les ensembles  $E$  à partir de « germes ». La figure 2 montre un exemple d'exécution de ce script sur l'image **Piree.png** (pour les distinguer, les ensembles  $E$  sont affichés avec des couleurs différentes).

**Remarque** – Il est conseillé de consulter la documentation des fonctions **ind2sub** et **sub2ind**.

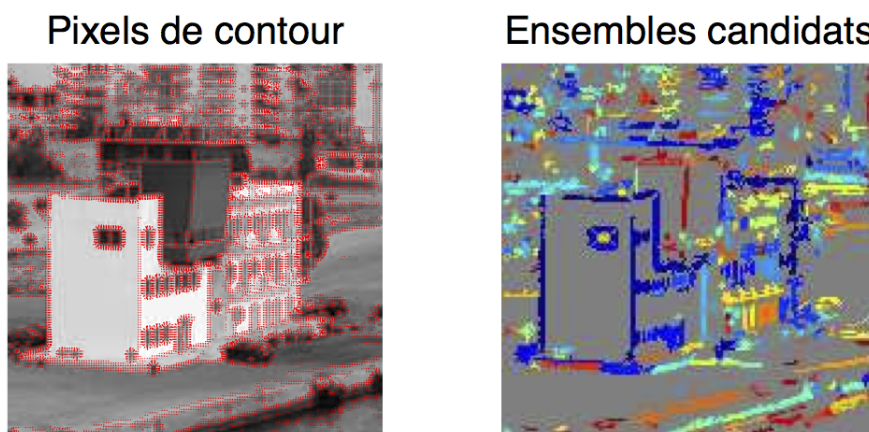


FIGURE 2 – À gauche : image sur laquelle le gradient du niveau de gris est superposé aux pixels de contour (sous la forme de flèches rouges). À droite : ensembles « candidats »  $E$ , constitués de pixels de contour.

## Exercice 2 : détection des alignements dans une image

Si la variable aléatoire « gradient du niveau de gris » en un pixel suivait une loi uniforme, la probabilité  $p$  pour qu'une réalisation fasse un angle inférieur à  $\alpha$ , relativement à une direction de référence, vaudrait  $p = \frac{2\alpha}{2\pi} = \frac{\alpha}{\pi}$ .

Soit  $E$  un des ensembles de pixels constitués à l'étape précédente. On définit  $R$  comme le plus petit rectangle englobant de  $E$ , ayant pour axes les axes principaux de sa *matrice d'inertie*. Cette matrice est une généralisation de la matrice de variance/covariance, pour laquelle les coordonnées  $(x, y)$  sont pondérées. En l'occurrence, un poids  $\|\nabla I(i, j)\|$  est attribué à chaque pixel  $(i, j)$  de  $E$ . Une fois ce rectangle déterminé, on compte le nombre  $n$  de pixels  $(i, j)$  de  $E$  où le gradient  $\nabla I(i, j)$  fait, avec la direction du petit axe de  $R$ , un angle inférieur à  $\alpha$ . La probabilité pour que cela se produise en  $n$  pixels de  $R$  *exactement* suit une loi binomiale. La probabilité pour que cela se produise en  $n$  pixels de  $R$  *au moins* est donc donnée par (1), où  $p = \frac{\alpha}{\pi}$  et  $N = \text{card}(R)$ . Le critère pour déterminer si un ensemble candidat  $E$  forme un alignement détectable s'écrit donc ( $\epsilon \ll 1$  est un seuil) :

$$\sum_{k=n}^N C_N^k p^k (1-p)^{N-k} < \epsilon \quad (2)$$

Complétez le script **exercice\_2.m** de façon à mettre en œuvre cette méthode probabiliste de détection d'alignements. Une fois mis au point sur l'image **Piree.png**, vous pourrez tester ce script sur les images **chaises.png** et **Morlaix.png** (soyez patient : le calcul est un peu long), ainsi que sur toute autre image de votre choix.