TP1 – Corrélation entre pixels dans une image

Initiation à Matlab

Notions générales

- Le résultat d'une affectation = est affiché, sauf si cette affectation se termine par un caractère `;`
- Les commandes `format short` et `format long` permettent de modifier le format d'affichage des variables.
- Les commandes `who` et `whos` permettent d'afficher l'ensemble des variables utilisées.
- La commande `clear` efface le contenu de toutes les variables utilisées.
- Il est fortement déconseillé d'utiliser des mots-clés de Matlab comme noms de variables.
- La commande `help <fonction>` affiche la description de `<fonction>` (exemple : `help plot` décrit la syntaxe de l'affichage de graphiques 2D).

Manipulation de vecteurs et de matrices

- Les composantes d'un vecteur ligne sont séparées par des virgules ou des espaces : `v1 = [x1 y1 z1];`
- Les composantes d'un vecteur colonne sont séparées par des points-virgules : `v2 = [x2 ; y2 ; z2];`
- Vecteur à incrément constant : `v3 = x_min:dx:x_max;` (vecteur ligne de dimension variable, qui contient les valeurs `x_min+i*dx`, où `i` est un entier positif ou nul tel que `x_min+i*dx` est inférieur à `x_max`).
- Les matrices utilisent la même syntaxe que les vecteurs : `M = [m11 m12 m13 ; m21 m22 m23];`
- La sous-matrice de `M` constituée par les lignes de numéros pairs et les colonnes de numéros impairs s'écrit : `N = M(2:2:size(M,1),1:2:size(M,2));`
- Vectorisation d'une matrice (les colonnes de `M` sont concaténées) : `v = M(:);`

Quelques matrices utiles

- `zeros(m,n)` : matrice nulle de taille $m \times n$.
- `ones(m,n)` : matrice de taille $m \times n$ dont tous les éléments sont égaux à 1.
- `eye(m,n)` : matrice de taille $m \times n$ dont les éléments diagonaux sont égaux à 1, les autres à 0.
- `rand(m,n)` : matrice de taille $m \times n$ d'éléments tirés aléatoirement selon la loi uniforme sur $[0, 1]$.
- `randn(m,n)` : matrice de taille $m \times n$ d'éléments tirés aléatoirement selon la loi normale centrée réduite.
- Appeler ces fonctions avec un seul argument équivaut à les lancer avec deux arguments identiques.

Opérations sur les matrices

- Addition `A+B` ; soustraction `A-B` ; produit `A*A'` ; puissance `A^3` ; transposition `A'` ou `transpose(A)`.
- Inverse `inv(A)` ; pseudo-inverse `pinv(A)`.
- Multiplication **élément par élément** `A.*B` (chaque élément `A(i,j)` est multiplié par l'élément `B(i,j)`) ; division **élément par élément** `A./B` (chaque élément `A(i,j)` est divisé par l'élément `B(i,j)`) ; puissance **élément par élément** `A.^3` (chaque élément de `A` est élevé à la puissance 3).

Exercice d'initiation à Matlab

Le script `exercice_Matlab.m` lit une image en niveau de gris, la stocke dans la matrice bidimensionnelle `image_originale` et l'affiche. Les niveaux de gris des pixels de cette image sont obtenus après traversée d'une mosaïque de filtres colorés, appelée *matrice de Bayer*, qui est placée devant le récepteur photosensible des appareils photographiques (cf. figure 1-a).

Complétez ce script de façon à créer une matrice `image_RVB` à 3 dimensions, contenant deux fois moins de lignes et deux fois moins de colonnes que `image_originale`. Chaque pixel de `image_RVB` correspond à un ensemble de quatre pixels de `image_originale` : V_1, R, V_2, B. Les valeurs R et B sont recopiées telles quelles dans les canaux rouge et bleu de `image_RVB`. En revanche, la valeur dans le canal vert est égale à la moyenne des valeurs V_1 et V_2. Affichez `image_RVB` et vérifiez visuellement l'exactitude du résultat.

Exercice 1 : mise en évidence des corrélations entre pixels voisins

Cet exercice constitue une introduction au cours de probabilités consacré à un couple de variables aléatoires. Ici, chaque pixel d'une image numérique est considéré comme une variable aléatoire. On s'intéresse à la corrélation entre pixels voisins.

Le script Matlab de nom `exercice_1.m` affiche une image interne à Matlab (cf. figure 1-b) ainsi que son histogramme (cf. figure 1-c). Complétez ce script de telle sorte qu'il affiche les paires de niveaux de gris d'un pixel et de son voisin de droite, sous la forme d'un nuage de points. Calculez et affichez le coefficient de corrélation linéaire de ces données. Enfin, affichez la droite de régression.

(a) (b) (c)

FIGURE 1 – (a) Mosaïque de filtres colorés en rouge, vert et bleu (matrice de Bayer). (b) Exemple d'image interne à Matlab. (c) Histogramme de l'image (b).

Rappels

- Moyenne : $\bar{x} = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} x_i$
- Variance : $\sigma_x^2 = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}(x_i - \bar{x})^2 = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} x_i^2 - \bar{x}^2$
- Écart-type : $\sigma_x = \sqrt{\sigma_x^2}$
- Covariance : $\sigma_{xy} = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} x_i\, y_i - \bar{x}\,\bar{y}$
- Coefficient de corrélation linéaire : $r = \dfrac{\sigma_{xy}}{\sigma_x\, \sigma_y}$
- Équation de la droite de régression : $y = \dfrac{\sigma_{xy}}{\sigma_x^2}\,(x - \bar{x}) + \bar{y}$

Exercice 2 : décorrélation des niveaux de gris d'une image

La décorrélation des niveaux de gris consiste, par exemple, à soustraire au niveau de gris d'un pixel le niveau de gris de son voisin de gauche.

Dupliquez le script `exercice_1.m` sous le nom `exercice_2.m`, puis modifiez ce dernier de façon à calculer une version décorrélée de l'image de la figure 1-a. Pour que les deux images aient les mêmes dimensions, collez la première colonne de l'image initiale à gauche de la première colonne de l'image décorrélée. Attention : les niveaux de gris de l'image décorrélée pouvant être positifs ou négatifs, il est nécessaire de modifier la valeur de la variable `I_min`.

Affichez le nuage de points analogue à celui de l'exercice 1. Calculez enfin le nouveau coefficient de corrélation linéaire. Conclusion ?