TP1-2013. Premiers pas avec les images sous `MATLAB`

Cette première feuille de TP du cours 2013 prépare en même temps qu'elle illustre les premiers cours introductifs au traitement des images digitales sous `MATLAB`.

Exercice 1 (Codage RGB).

(1) Rédigez un code sous `MATLAB`

```
>> [IR,IG,IB] = decompRGB(I);
```

qui, étant donnée une image couleur téléchargée depuis le web comme une image au format `.jpeg,.jpg,.gif,.bmp` (*etc.*) grâce à la routine `I=imread('toto.format')`, isole les trois composantes (chacune sous la forme d'une variable de taille `(M1,M2,3)`) correspondant aux trois couleurs (respectivement `R,G,B`). Convertissez ces trois composantes au format `double`, puis ensuite transformez les en des variables de taille `(M1,M2)` prenant leurs valeurs dans 255 canaux de l'intervalle $[0,1]$, correspondant aux valeurs `[0:255]/255`. Testez ce code sur l'exemple de l'image `venise.jpg`.

(2) Réalisez par moyenne une image au format `gray` (noir et blanc) modélisée par une matrice de taille (M,N) à entrées dans [0,1] (au format flottant en double précision), codée sur les 256 canaux `[0:255]/255`.

(3) Réalisez un code permettant de déduire de la variable obtenue depuis une image couleur `I` *via*

```
>> I = imread('toto.format');
>> I = double(I)/255;
```

trois composantes (chacune de taille `M1,M2,3`) correspondant cette fois aux trois couleurs complémentaires `C,M,Y` (`cyan,magenta,yellow`) de l'image RGB `I`.

Exercice 2 (histogramme et histograme cumulé d'une image). Cet exercice est l'occasion de voir rapidement et de manière pour l'instant qualitative quelques rudiments de probabilité. Pour un support plus approfondi, on pourra (ultérieurement) consulter les notes du cours de probabilité pour Math-Info en ligne sur mon site :

`http://www.math.u-bordeaux1.fr/~ayger/Proba6031.pdf`

Étant donnée une image `I` de taille `(M1,M2)` codée sur N niveaux de gris (dans la pratique pour nous $N = 256$, ce qui correspond au codage initial au format `uint8`) et ramenée à une image au format flottant (`double`) prenant ses valeurs dans $[0,1]$

(en fait prenant les valeurs $k/(N-1)$ avec $k = 0 : N-1$), l'*histogramme* de l'image est l'application définie sur $\{0, ..., N-1\}$:

$$H \ : k \in \{0, ..., N-1\} \longmapsto \frac{\text{card}\,\{\text{pixels}\,;\, I(\text{pixel}) \in [k/(N-1), (k+1)/(N-1)[\}}{M_1 \cdot M_2}$$

Cela revient à associer à k la ≪ proportion ≫ de pixels tombant dans la boite correspondant à l'intervalle $[k/(N-1), (k+1)/(N-1)[$, avec la convention que, lorsque $k = N-1$, le cardinal au numérateur figure le nombre de pixels auquels est attribuée la valeur 1.

(1) Examinez les routines `H=histc(V,X)` et `H=hist(V,X)` sous `MATLAB` et calculez grâce à la routine `histc` l'histogramme des images `bordeaux.jpg`, `venise.jpg`, `alsace.jpg`, `lena.jpg` mises en ligne sur le site (une fois converties au format noir et blanc, puis double). Rédigez un code (que vous appellerez `histoIMAG`) qui permette de calculer (en exploitant la routine `histc`) l'histogramme d'une image I codée sur N canaux de gris). L'affichage sous forme d'histogramme de la fonction $k \mapsto M_1 M_2 H(k)$ est aussi fournie par la commande `hist`. Intégrez cet affichage dans le code.

(2) Transformez le code `histoIMAG` de manière à y intégrer la construction de l'*histogramme cumulé*, c'est-à-dire celle de la fonction

$$\widetilde{H} \ : k \in \{0, ..., N-1\} \longmapsto \sum_{\ell=0}^{k} H(\ell).$$

En termes de théorie des probabilités, il faut penser à H comme à une *densité de probabilité* sur $[0, 1]$ et à l'histogramme cumulé $\widetilde{H}$ comme à la *fonction de répartition* correspondante.

(3) Que valent la *moyenne* et la *variance* de l'image ? Rédigez un code permettant de calculer ces deux paramètres. Il faut juste avoir en tête ici que dans le cadre continu des variables aléatoires I ayant une densité f sur $\mathbb{R}$ et donc de fonction de répartition

$$x \mapsto \int_{-\infty}^{x} H(t)\, dt$$

(ici la densité matérialisée par H vit dans $[0, 1]$), la moyenne est

$$E[I] := \int_{\mathbb{R}} tH(t)\, dt = \int_0^1 t\, H(t)\, dt$$

tandis que la variance est

$$\sigma^2(I) := E\big[I - E[I]\big]^2 = E[I^2] - (E[I])^2 = \int_0^1 t^2\, H(t)\, dt - \Big(\int_0^1 t\, H(t)\, dt\Big)^2.$$

À vous de discrétiser ces formules pour faire le calcul demandé.

Exercice 3 (Quelques transformations d'image).

(1) Qu'est-ce qui différencie graphiquement les graphes des fonctions $t \mapsto t^\gamma$ lorsque $\gamma \in]0, +\infty[$ (suivant que γ est strictement inférieur à 1, égal à 1 ou strictement supérieur à 1) ? Rédigez un code `puissance` qui, étant donnée une image noir et blanc I codée entre 0 et 1 sur N canaux de gris, transforme cette image en l'image I^γ. Faites des tests sur les diverses

images proposées. Que se passe-t-il suivant que $\gamma < 1$ ou $\gamma > 1$? Regardez l'effet sur les histogrammes et expliquez.

(2) Comment transformer (*via* une transformation `T`) une image `I` (codée sur N niveaux de gris entre $[0,1]$ de manière à ce que l'histogramme de `T(I)` soit celui d'une variable aléatoire distribuée de manière uniforme sur les L niveaux de gris $[0:L-1]/(L-1)$? Réalisez un code opérant cette transformation. On observera d'abord que si $L = N$ et si l'histogramme cumulé de `I` est une fonction struictement croissante, alors la transformation `T` en question consiste a prendre $T = \widetilde{H}$. Cette opération s'appelle l'*égalisation d'image* et vous testerez ce code `Iequal = egalisation (I,L,N)` sur les diverses images proposées.

Exercice 4 (Seuillage binaire et algorithme d'Otsu).

(1) Réalisez un code

```
Ibinaire = seuillagebinaire (I,lambda,N) ;
```

qui, étant donné un entier $\lambda \in \{0, ..., N-1\}$ transforme une image `I` (codée en noir et blanc sur N canaux de $[0,1]$, c'est-à-dire aux points `k=[0:N-1]/(N-1)`) en une image binaire en mettant à 0 les pixels tels que la valeur de `I` au pixel soit strictement inférieure à $\lambda/(N-1)$ et à 1 ceux tels que cette valeur soit supérieure ou égale à ce seuil $\lambda/(N-1)$.

(2) Réalisez un code qui, pour chaque $t = 2 : N-2$, calcule les moyennes $\mu_t^{\pm}$ et $(\sigma_t^2)^{\pm}$ des variables aléatoires correspondant respectivement aux histogrammes H_t^- et H_t^+ correspondant à la « partition » des pixels en deux classes : celle où la valeur au pixel est strictement inférieure au seuil $t/(N-1)$ et celle où la valeur au pixel est supérieure où égal à ce seuil $t/(N-1)$. Calculez en même temps les proportions $p_t^{\pm}$ et faites en sorte que le code renvoie le vecteur

$$p_t^-\,(\sigma_t^2)^- + p_t^+\,(\sigma_t^2)^+, \qquad t = 2 : ... : N-2.$$

Faites en sorte que ce code renvoie aussi la valeur $\lambda = t_{\min}$ où cette fonction de $t \in \{2, ..., N-2\}$ réalise son minimum (ou la plus petite valeur de t où c'est le cas au cas où il y en aurait plusieurs). On construira ce code sous la syntaxe

```
>> [varintra,lambda] = algorithmeOTSU(I,N);
```

(3) Pourquoi la valeur λ trouvée à la question 2 vous parait-elle le meilleur compromis pour effectuer un seuillage binaire de l'image précisément à ce seuil λ ? Réalisez ce seuillage sur ces des exemples (`bordeaux`, `alsace` ou `venise`) et comparez avec d'autres seuillages pour une valeur λ' différente de cette valeur λ optimale.