M4201Cin TP 2

Histogrammes, quantification et échantillonnage

23 janvier 2018 – B. COLOMBEL

Les réponses aux questios des exercices 1 à 6 seront écrites dans l'éditeur Scinote dans un seul fichier.
Les réponses autres que des fonctions Scilab seront commentées. Il sera enregistré sous le nom :

`Nom_Prenom-TP2.sci`

et déposé dans le dossier prévu à cet effet dans AMETICE (module M3201Cin, TP2)

Si le module SIVP de **Scilab** est bien installé, vous verrez au démarrage :

```
Initialisation :
  Chargement de l'environnement de travail

SIVP - Scilab Image and Video Processing Toolbox
        load macros
        load gateways
        load help
        load demos
```

Sinon, installez le :

Applications » Gestionnaire de modules - ATOMS » Traitements des images » SIVP

Cliquer sur [installer] puis redémarrer **Scilab**.

1 Histogramme d'une image — Traitement ponctuel

Exercice 1 :

1. Ouvrir l'image `lena.pgm` et la stocker dans la variable `image0`.
2. Afficher son histogramme avec la commande :

```
--> [counts, cells] = imhist(uint8(image0))
--> plot(cells, counts);
```

Que contiennent les variables `counts` et `cells` ?

Une image étant un tableau nous pouvons lui appliquer une fonction (bien définie) pour en modifier les niveaux de gris. Par exemple, si on peut utiliser les fonctions suivantes :

$$f(x) = 255\sqrt{\frac{x}{255}} \quad \text{et} \quad g(x) = 255\left(\frac{x}{255}\right)^2$$

```
--> I1 = 255 * sqrt(double(image0) ./255);
--> imshow(unit8(I1))
--> I2 = 255 * (double(image0) ./255) .^2;
--> imshow(uint8(I2))
```

3. le négatif d'une image I_0 est une autre image I_{neg} dont les niveaux sont définis par :

$$I_{\text{neg}} = I_{\max} - I_0(i,j) = I_{\max}\left(1 - \frac{I_0(i,j)}{I_{\max}}\right)$$

   Faire le négatif de l'image `image0` avec scilab (afficher la nouvelle image pour vérifier).
   une seule ligne de commande nécessaire

4. Écrire une fonction **Scilab** `res = teinte(image, h)` qui ajoute la valeur h à chaque pixel de l'image `image` pour $-255 \leqslant h \leqslant 255$.

⚠ on ne peut pas avoir de pixels négatifs ou supérieurs à 255. On pourra utiliser les commandes suivantes pour modifier les valeurs d'une matrice selon un critère :

```
--> X(X<0)=0; // Met toutes les valeurs negatives de la matrice a 0
--> X(X==9)=5; // Met tous les pixels de l'image valant 9 a la valeur 5
--> X(2<=X<=10)=6; // Met tous les pixels de l'image compris entre 2 et 10 a la valeur
 6
```

Look Up Table

Soit $I(x,y)$ le niveau de gris du point P de l'image source et $I'(x,y)$ le niveau de gris de l'image résultat. L'opération ponctuelle réalise l'application suivante de $\mathbb{R}_+$ dans $\mathbb{R}_+$:

$$I'(x;y) = \varphi\left[I(x;y)\right]$$

Les coordonnées du point résultat sont supposée identiques à celle du point source dans cette étude. L'opération peut se représenter par un graphe ou une table (LUT).

La table LUT peut être utilisée comme suit :

Conctruction et utilisation d'une Look Up Table (LUT)

Entrées : Une image I de dimension $p \times n$;
Une transformation locale φ ;
début
   // Initialisation de la LUT
   pour *i variant de 1 à 256* **faire**
      $\text{LUT}(i) = \varphi(i)$;
   fin
   // Utilisation de la LUT
   pour *i variant de 1 à p* **faire**
      pour *j variant de 1 à n* **faire**
         $I'(i,j) = \text{LUT}(I(i,j))$;
      fin
   fin
fin

Exercice 2 : **Recadrage de l'histogramme**

1. Écrire une fonction qui modifie les niveaux de gris des pixels de telle sorte que la dynamique des niveaux de gris soit comprise entre 0 et 255 (recadrage dynamique).

   On pourra utiliser les fonctions `min(M)` (et `max(M)`) qui retourne la valeur minimale (et maximale) des coeffcients de la matrice M.
2. Tester la fonction avec l'image `hotel-de-ville.pgm`.

Exercice 3 : **Égalisation de l'histogramme**

1. Écrire une fonction qui modifie les niveaux de gris des pixels par une égalisation d'histogramme et qui affiche l'histogramme et l'histogramme cumulé de l'image résultante.
2. Tester votre fonction sur l'image `port.pgm`.

2 Quantification

Exercice 4 :

1. (a) Ouvrir l'image `cameraman.pgm` et la stocker dans une matrice `I1`.
   (b) Quelle est sa taille ? Sur combien de bits est-elle codée ?
   (c) Afficher l'image.
   (d) Tracer son histogramme et commenter.

Étant donné une image quantifiée sur `b1` bits, on peut utiliser la commande suivante pour la requantifier sur `b2` bits :

```
--> image = floor(image ./ 2^(b1-b2));
```

2. Créer une nouvelle image `I2` correspondant à l'image `I1` quantifiée sur 6 bits. Afficher l'image obtenue et son histogramme. Quel est le lien entre l'histogramme de `I1` et celui de `I2` ? Commenter.
3. Créer une nouvelle image `I3` correspondant à l'image `I1` quantifiée sur 4 bits. Afficher l'image obtenue et son histogramme. Qu'observe-t-on sur l'image ? Dans quelles zones ce phénomène est-il particulièrement visible ?
4. Calculer la différence `I4 = abs(I3-I1)` (en utilisant les versions renormalisées de `I1` et `I3`) et l'afficher sous la forme d'une image. Confronter avec les observations de la question précédente : ces résultats sont-ils cohérents ? Comment peut-on expliquer cela ?

3 Échantillonnage

L'une des toutes premières étapes du traitement numérique des images est la tâche d'échantillonnage qui réduit l'ensemble continu du monde observable en une série de valeurs discrètes. L'échantillonnage apparaît aussi en de nombreuses autres occasions, par exemple lorsqu'on redimensionne une image, lorsqu'on la convertit en un autre format, etc.

Par défaut, une image I de dimension $n \times m$ est indexée par les indices i et j (qui sont des entiers). On définit alors les *vecteurs spatiaux* $x = (1,2,\cdots,n)$ et $y = (1,2,\cdots,m)$.
Il existe plusieurs conventions, mais nous allons supposer ici que le vecteur x correspond à la verticale et le vecteur y correspond à l'horizontale.

Remarque. Pour certaines images de synthèse, on définit une image $I'(x,y)$ avec des vecteurs spatiaux x et y ne contenant plus nécessairement des valeurs entières.

Exercice 5 : **Sous-échantillonnage et sur-échantilonnage**

1. Ouvrir l'image `cameraman.pgm` et la stocker dans une matrice `Z`. Récupérer le nombre de lignes n et le nombre de colonnes m de l'image.

Les *vecteurs spatiaux* définissant l'image `Z` sont donc `x = 1 : n` et `y = 1 : m`.

3. Si l'on souhaite créer une image de taille deux fois plus petite, on peut utiliser les vecteurs spatiaux `x_sous = 1, 3, 5, ..., n` et `y_sous = 1, 3, 5, ..., m`.
   Écrire une fonction `Z_sous = sous_ech(Z)` correspondant à une version sous-échantillonnée de `Z` grâce aux vecteurs `x_sous` et `y_sous` définis précédemment. Commenter.
4. Si l'on souhaite créer une image de taille deux fois plus grande, on peut utiliser les vecteurs spatiaux `x_sur = 1,1,2,2,3,3, ..., n, n` et `y_sur = 1,1,2,2,3,3, ..., m, m`.
   Écrire une fonction `Z_sur = sur_ech(Z)` correspondant à une version sur-échantillonnée de `Z` grâce aux vecteurs `x_sur` et `y_sur` définis précédemment. Commenter.

Exercice 6 : On souhaite générer l'image de synthèse (normalisée) suivante :

$$z(x;y) = 255 \times \left(\frac{1}{2} + \frac{1}{2}\cos\left(2\pi(3x+4y)\right)\right)$$

Pour cela, on va utiliser des pas d'échantillonnage $\Delta x = 0{,}05$ et $\Delta y = 0{,}05$ respectivement pour les axes x et y. On souhaite générer une image ayant $M = 200$ lignes et $N = 300$ colonnes.

1. Définir les vecteurs spatiaux x et y à utiliser (on prendra $x(1) = 0$ et $y(1) = 0$).
2. Créer l'image `Z1` correspondnate.
3. Faire varier les pas d'échantillonnage Δx et Δy et commenter.