

## **Evaluation TP Traitement d'images**

### **UE HAI605I Données multimédia**

- 
- Durée : 1h15'
  - Note sur 20 comptant pour 20% de la note totale de l'UE.
  - TP à faire obligatoirement sur les ordinateurs des salles machines de l'Université
  - Utilisation possible des codes développés lors des 5 séances de TP
  - Tout moyen d'échange et de communication entre étudiants est interdit
  - Rédiger et envoyer par email à la fin de la séance votre compte-rendu au format pdf : CR\_TP\_IMAGES\_NOM\_PRENOM.pdf
- 

#### **1) Histogramme d'une image**

- a) Prendre l'image pgm (entre 01.pgm et 15.pgm) indiquée par l'enseignant dans le répertoire suivant :  
[https://www.lirmm.fr/~wpuech/enseignement/master\\_informatique/Analyse\\_Traitement\\_Image/TP/images/](https://www.lirmm.fr/~wpuech/enseignement/master_informatique/Analyse_Traitement_Image/TP/images/)
- b) Ecrire un programme histo.cpp (argument d'entrée : 1 image au format pgm) permettant d'afficher à l'écran les occurrences des niveaux de gris de l'image. Rediriger l'affichage dans un fichier histo.dat qui contiendra 2 colonnes : indice et occurrence des niveaux de gris.
- c) A l'aide de gnuplot, visualiser l'histogramme.

**Dans votre compte rendu :**

- Insérer l'image originale en niveaux de gris au format pgm.
- Insérer l'histogramme de l'image originale
- Insérer 2 à 3 commentaires concernant l'histogramme obtenu.

#### **2) Densité de probabilité (ddp) d'une image**

- a) A partir du programme histo.cpp écrire un programme ddp.cpp permettant de la même manière d'afficher à l'écran la densité de probabilité (ddp) des niveaux de gris d'une image au format pgm. La ddp est égale aux occurrences divisées par le nombre de pixels de l'image, la somme des ddp (pour  $0 \leq i < 256$ ) est égale à 1. Compiler et exécuter votre programme avec la même image.
- b) Tracer cette ddp avec gnuplot. Que constatez-vous entre la forme de la ddp et celle de l'histogramme ?

**Dans votre compte rendu :**

- Insérer la densité de probabilité obtenue
- Décrire ce que vous constatez

#### **3) Fonction de répartition**

- a) A partir du programme `ddp.cpp` écrire un programme `repartition.cpp` permettant de la même manière d'afficher à l'écran la fonction de répartition des niveaux de gris d'une image au format `pgm`. Pour rappel la fonction de répartition est une fonction croissante de 0 à 1 pour  $0 \leq i < 256$ .  $F[0] = ddp[0]$  et  $F[i] = F[i-1] + ddp[i]$  pour  $0 < i < 256$  (au final  $F[255]$  doit être égal à 1). Compiler et exécuter votre programme avec la même image.
- b) Tracer la fonction de répartition avec `gnuplot`. Que constatez-vous ?

**Dans votre compte rendu :**

- Insérer la fonction de répartition obtenue
- Décrire ce que vous constatez

#### 4) Augmentation du contraste d'une image par égalisation d'histogramme

L'objectif de cette partie est d'augmenter le contraste d'une image par égalisation de son histogramme, cela à partir de la fonction de répartition obtenue à la question 3.

- a) A partir du programme `repartition.cpp` écrire un programme `egalisation.cpp` avec 2 arguments : 1 image en entrée au format `pgm` et une image en sortie au format `pgm`. L'image de sortie doit correspondre à l'image d'entrée de manière contrastée.  
A tout pixel  $p(i)$  de l'image d'entrée doit correspondre un pixel  $p'(i)$  de l'image de sortie tel que  $p'(i) = \text{partie\_entiere}(F[p(i)] \times 255)$ .  
Compiler et exécuter votre programme avec la même image. Que constatez-vous ?
- c) A l'aide de `gnuplot`, afficher sur le même graphique les histogrammes de l'image d'entrée et de l'image contrastée. Que constatez-vous ?

**Dans votre compte rendu :**

- Insérer cote à cote l'image originale et l'image contrastée.
- Insérer sur le même graphique les histogrammes des deux images.
- Insérer 2 à 3 commentaires concernant les résultats obtenus.