

Travaux dirigés Traitement d'images n°3

Histogrammes et lookup tables

—Master 1—

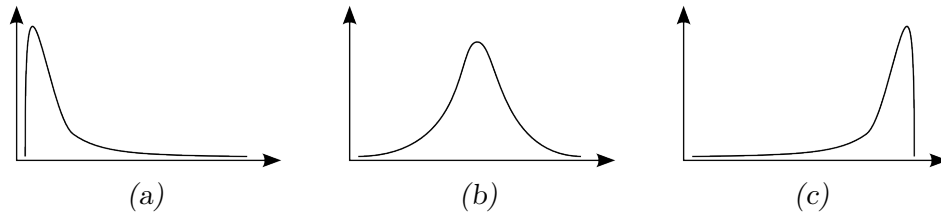
► Exercice 1. Histogrammes et images en niveaux de gris

1. Affichage

- (a) Récupérez le fichier `tp03.zip`.
- (b) Ouvrez le fichier `tp03/tp03_ex01.cpp` et exécutez le. Ce programme converti une image en niveaux de gris, affiche l'image et son histogramme. Testez ce programme avec une image. Lire le code.

2. Histogrammes

- (a) Trouvez des images sur internet telle que leur histogramme soit :



► Exercice 2. Normalisation d'images

- 1. Ouvrez le fichier `tp03/tp03_ex02.cpp` et exécutez le. Ce programme applique une lookup table `LUTplop` à une image en niveau de gris. Testez le.
- 2. Quelle fonction cette lookup table représente-t-elle?
- 3. Inspirez-vous de `LUTplop` pour générer une lookup table qui, appliquée à l'image, normalise son histogramme. L'image finale utilise alors toute la dynamique disponible (de 0 à 255). Vous utiliserez la formule :

$$I'(x, y) = \frac{I(x, y) - I_{\min}}{I_{\max} - I_{\min}} \cdot 255$$

et le prototype suivant :

```
cv::Mat LUTimageNormalize(const cv::Mat &image);
```

► **Exercice 3. Egalisation d’histogramme**

1. Faites une fonction qui calcule l’histogramme cumulé d’une image en niveaux de gris. La fonction aura le prototype suivant :

```
cv::Mat computeCumulatedHistogramGS(const cv::Mat &imageGS);
```

2. Faites une fonction calculant une LUT qui, appliquée à une image, égalise son histogramme. Vous utiliserez la formule suivante :

$$\text{LUT}(x) = \frac{255}{w \times h} H_c(x)$$

La fonction aura le prototype suivant :

```
cv::Mat generateLUTHistogramNormalize(const cv::Mat &imageGS);
```

3. Testez vos fonctions sur l’image `théCafé.png`.