

## Mise en œuvre de traitements d'image à partir de caractéristiques statistiques

---

### DESCRIPTION DES ACTIVITÉS

#### Présentation

**Objectifs :** calculer globalement et localement les principales caractéristiques statistiques d'une image monochrome et mettre en œuvre des traitements basés sur ces caractéristiques.

**Description générale du travail attendu :** calcul global et local des principales caractéristiques statistiques d'une image monochrome (histogramme, moyenne, variance, entropie) et mise en œuvre de traitements adaptatifs utilisant ces caractéristiques.

**Activité notée :** écriture de huit fichiers en langage C : `histogram.c`, `stif.c`, `test-lhist.c`, `test-lstif.c`, `threshold.c`, `al-blur.c`, `al-enhance.c` et `al-thresh.c`.

**Modalités :** activité effectuée en binôme sur une durée de trois semaines.

#### Grille de notation.

- D** (calcul des différents histogrammes : standard, normalisé, cumulé et normalisé cumulé) :
- le fichier `histogram.c` est compilable avec le fichier `test-hist.c` et un environnement standard ISO C99/POSIX (par exemple en utilisant la commande `gcc`);
  - les histogrammes produits en utilisant la fonction `test-hist` sont corrects.
- C** (calcul des caractéristiques statistiques et seuillage standard) :
- les fichiers `stif.c` et `threshold.c` sont compilables avec les fichiers de test correspondants et un environnement standard ISO C99/POSIX (par exemple en utilisant la commande `gcc`);
  - les statistiques produites sont correctes (C-)
  - l'histogramme local est correct (C)
  - les statistiques locales sont correctes (C+)
- B** (seuillage adaptatif, global ou local) :
- les fichiers fournis sont compilables avec un environnement standard ISO C99/POSIX (par exemple en utilisant la commande `gcc`);
  - les seuillages par rapport à un seuil fixe, à la moyenne et la médiane produisent des résultats corrects (B-)
  - les seuillages par rapport à un pourcentage et par rapport à la moyenne locale produisent des résultats corrects (B et B+)
- A** (traitements) :
- fonctionnement correct de un (A-), deux (A) ou trois (A+) des programmes `al-blur.c`, `al-thresh.c` et `al-enhance.c`

# 1 Rappels sur les caractéristiques statistiques d'une image monochrome

Les caractéristiques seront calculées sur des images monochromes pour  $M = 256$  niveaux de gris. Il s'agit :

— des histogrammes standard  $h$ , cumulé  $h_c$  et normalisé  $h'$  calculés sur un domaine  $D$

$$h(i) = \sum_{p \in D \text{ et } I(p)=i} 1 \quad (1)$$

$$h_c(i) = \sum_{j=0}^i h(j) \quad (2)$$

$$h'(i) = \frac{h(i)}{\sum_{i=0}^{M-1} h(i)} \quad (3)$$

— de l'histogramme normalisé cumulé  $h'_c$ , obtenu en cumulant l'histogramme normalisé  $h'(I)$  ;

— de la moyenne :

$$\mu = \sum_{i=0}^{M-1} h'(i) \times i \quad (4)$$

— de la médiane :

$$m = \min \left\{ i \in [0..M-1] \text{ tel que } \sum_{j=0}^i h(j) \geq \frac{\sum_{j=0}^{M-1} h(j)}{2} \right\} \quad (5)$$

— de la variance :

$$v = \sum_{i=0}^{M-1} h'(i) \times (i - \mu)^2 \quad (6)$$

— de l'entropie :

$$h'(i) = - \sum_{i=0}^{M-1} h'(i) \times \log_2(h'(i)) \quad \left( \text{par convention, on prend : } 0 \log_2(0) = 0 \right) \quad (7)$$

Par défaut, le domaine  $D$  utilisé pour calculer l'histogramme standard est le domaine entier de l'image. On peut aussi, pour chaque pixel  $p$  de l'image, restreindre le domaine de calcul de l'histogramme standard à un voisinage  $V(p)$  de ce pixel. Cela permet de calculer les différents paramètres statistiques localement à chaque pixel.

## 2 Activités

### Activité 1

Écrire le fichier `histogram.c` qui permet de calculer, pour une image monochrome  $I$ , les histogrammes  $h(I)$ ,  $h_c(I)$ ,  $h'(I)$  au moyen des fonctions `histogram_make`, `histogram_cumulate` et `histogram_normalize` (voir interface dans le fichier `histogram.h`). La fonction `histogram_make` alloue dynamiquement l'histogramme qu'elle calcule et retourne son adresse. Les autres fonctions modifient directement l'histogramme qu'elles reçoivent en paramètre.

Ce programme sera testés avec le fichier `test-hist.c` qui utilise lui-même le fichier `img.c`. Le fichier `Makefile.targets` contient les définitions de toutes les cibles permettant de compiler les programmes de cette activité.

#### Conseils pour l'activité 1.

1. La constante `HISTOGRAM_NLEV` définit le nombre de niveaux de gris de l'image, et donc la taille de l'histogramme.
2. Un résultat d'histogramme  $h(I)$  est fourni à titre d'exemple pour le fichier `quais.png` de la diapositive 9 du thème B-1. L'histogramme  $h'(I)$  doit avoir la même forme que  $h(I)$ , mais à valeurs dans  $[0, 1]$ . L'histogramme  $h_c(I)$  doit vérifier  $h_c(255) = \text{size}$ .
3. Il est possible de visualiser les histogrammes produits par la commande `test-hist` au moyen de la commande `gnuplot` :

```
test-hist FILE.png > FILE.hist
gnuplot -p -e "plot [0:255] \"FILE.hist\" with histogram"
```

et de les comparer avec les histogrammes affichés par Gimp (Couleurs->Informations-Histogramme). On peut utiliser d'autres mode d'affichage avec Gnuplot, par exemple `histeps`.

### Activité 2

Écrire le programme `stif.c` (*statistical image feature*) contenant les quatre fonctions

```
float stif_mean(unsigned char *channel, int size);
int stif_median(unsigned char *channel, int size);
float stif_variance(unsigned char *channel, int size);
float stif_entropy(unsigned char *channel, int size);
```

calculant respectivement la moyenne, la médiane, la variance et l'entropie d'une image monochrome.

Ces fonctions seront testées avec le fichier `test-stif.c`.

#### Conseils pour l'activité 2.

1. Comparer les résultats obtenus pour la moyenne, la variance et l'entropie avec ceux présentés dans le diaporama du thème A-01 (voir planche en fin de l'énoncé). Les fichiers correspondants sont fournis.
2. Pour vérifier la médiane, il est possible d'utiliser l'affichage d'histogramme de Gimp.

### Activité 3

- Ajouter dans le fichier `histogram.c` la fonction

```
float *histogram_make_local(unsigned char *channel, int width, int height,  
                           int i, int j, int half_width);
```

qui construit l'histogramme du canal monochrome défini par `{channel, width, height}` localement au point `(i, j)` sur un voisinage carré de demi-largeur `half_width`.

- Ajouter au fichier `stif.c` les fonctions `stif_lmean`, `stif_lmedian`, `stif_lvariance` et `stif_lentropy` calculant les caractéristiques locales à partir des mêmes paramètres (interface dans le fichier `stif.h`).

- À partir des fichiers `test-hist.c` et `test-stif.c`, écrire les fichiers `test-lhist.c` et `test-lstif.c` permettant de tester ces nouvelles fonctionnalités :

```
Usage: test-lhist [-c] [-n] I J HALF-WIDTH IMAGE
```

```
Usage: test-lstif I J HALF-WIDTH IMAGE
```

et qui produit les mêmes résultats que `test-hist.c`, mais pour un domaine de calcul de l'histogramme défini par `{I,J,HALF-WIDTH}`.

#### Conseils pour l'activité 3.

1. Pour déterminer le domaine de calcul de l'histogramme local, on pourra reprendre le principe de boîte englobante utilisé pour mettre en œuvre les primitives morphologiques (on pourra éventuellement utiliser pour cela le module `bbox` fourni).
2. Il est possible de construire un histogramme local avec Gimp en utilisant l'outil de sélection de région. La vérification des histogrammes cumulés et normalisés peut se faire selon les mêmes principes que pour l'activité 1.
3. Si le calcul des histogrammes locaux est correct, celui des paramètres statistiques locaux qui en découle directement doit l'être également.

#### Activité 4

Écrire le fichier `threshold.c` comportant les cinq fonctions (interface dans `threshold.h`) :

— `threshold_std` : seuillage d'un canal monochrome par rapport à un seuil `th`

$$I'(p) = \begin{cases} 0, & \text{si } I(p) < th \\ 255, & \text{sinon} \end{cases}$$

— `threshold_mean` et `threshold_median` : seuillage en prenant pour valeur de seuillage respectivement la moyenne et la médiane ;

— `threshold_percent` : seuillage par rapport à un seuil `t` tel que  $p\%$  des pixels aient la valeur zéro ( $p$  étant un paramètre de la fonction) ;

— `threshold_lmean` : en chaque pixel, seuillage par rapport à la moyenne locale en ce pixel :

$$I'(p) = \begin{cases} 0, & \text{si } I(p) < \mu(V(p)) \\ 255, & \text{sinon} \end{cases}$$

Ces fonctions seront testées avec le fichier `test-thresh.c`.

#### Conseils pour l'activité 4.

1. Le calcul du seuil  $t_p$  à partir de  $p$  peut se faire simplement avec l'histogramme normalisé cumulé :

$$t_p = \min \left\{ i \text{ tel que } h'_c(i) > p \right\}$$

2. Plusieurs exemples de résultats de seuillage sont fournis à partir des images `quais.png`, `txt1.png` et `txt2.png` :
  - `quais-th127.png` et `quais-th-med.png` : relativement au milieu et à la médiane ;
  - `quais-th-mean.png`, `txt1-th-mean.png` et `txt2-th-mean.png` : relativement à la moyenne ;
  - `quais-th-p25.png` et `quais-th-p75.png` : relativement aux pourcentages de 25% et 75% ;
  - `txt1-thlm-4.png`, `txt1-thlm-16.png` et `txt2-thlm-16.png` : relativement à la moyenne locale sur des voisinages de demi-largeur 4 et 16 pixels.

## Activité 5

*Application à la mise en œuvre de traitements adaptatifs. Écrire les sources des commandes :*

- `al-blur HALF-WIDTH T IMAGE RESULT` qui construit l'image  $I'$  résultant du lissage adaptatif suivant :

$$I'(p) = \begin{cases} \mu(V(p)) & \text{si } |I(p) - \mu(V(p))| \leq T \\ I(p) & \text{sinon} \end{cases}$$

- `al-thresh HALF-WIDTH K IMAGE RESULT` qui construit l'image  $I'$  résultant du seuillage adaptatif suivant :

$$\begin{cases} T(p) = \mu(V(p)) - K \times \sigma(V(p)) \\ I'(p) = \begin{cases} 0 & \text{si } I(p) \leq T(p) \\ 1 & \text{sinon} \end{cases} \end{cases}$$

- `al-enhance HALF-WIDTH D IMAGE RESULT` qui construit l'image  $I'$  résultant du rehaussement adaptatif de contraste suivant :

$$I'(p) = \mu(V(p)) + \frac{D}{\sigma(V(p)) + \varepsilon} (I(p) - \mu(V(p)))$$

où  $\varepsilon$  est une constante arbitraire de petite taille permettant d'éviter les divisions par zéro (par exemple  $\varepsilon = 10^{-2}$ ).

Les constantes  $T$ ,  $K$  et  $D$  sont des paramètres de réglage passés en paramètres et ajustés en fonction de l'image et du résultat attendu (voir exemples).

Chacun des trois fichiers `al-blur.c`, `al-thresh.c` et `al-enhance.c` doit être compilable en utilisant seulement les fichiers définis précédemment.

### Conseils pour l'activité 5.

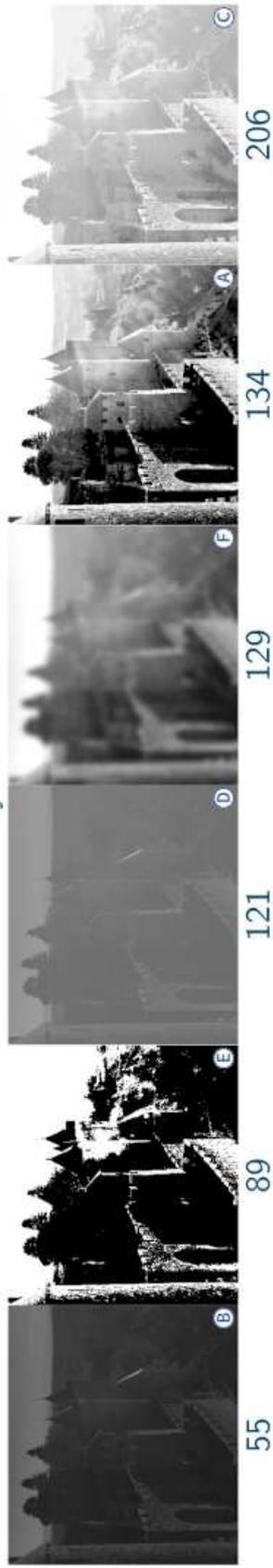
1. Les fichiers `lac-4-256.png` et `lac-4-60.png` sont des exemples de lissage adaptatif du fichier `lac.png`, pour une demi-largeur de quatre pixels et un seuil respectivement de 256 et de 60.
2. Les fichiers `txt1-ath-4-05.png` et `txt1-ath-16-09.png` sont des exemples de seuillage adaptatif du fichier `text1.png` pour des demi-largeurs respectivement de quatre et seize pixels, et des valeurs de  $K$  de 0.5 et 0.9 (à comparer avec les résultats de seuillage par rapport à la moyenne et à la moyenne locale de l'activité 4). Des exemples analogues sont fournis pour le fichier `text2.png`.
3. Le fichier `toits-8-64.png` est un exemple de rehaussement adaptatif obtenu avec le fichier `toit.png`, une demi-largeur de voisinage 8 et une valeur de  $D$  égale à 64.

### Récapitulatif des fichiers à remettre

- activité 1 : `histogram.c`
- activité 2 : `stif.c`
- activité 3 : `test-lhist.c` et `test-lstif.c` (avec les modifications apportées aux deux fichiers `histogram.c` et `stif.c` pour ajouter les traitements locaux)
- activité 4 : `threshold.c`
- activité 5 : `al-blur.c`, `al-thresh.c` et `al-enhance.c`

Le fichier `Makefile` fourni doit permettre de compiler séparément les programmes de chaque activité.

Moyennes



Écart types



Entropies

