

TP2

Manipulations d'histogramme

Analyse et transformation d'histogrammes

Exercice 1. Analyse et export d'histogramme

On va s'intéresser à l'amélioration des images `objects-dark.pgm` et `len_dark.pgm`

1. Reprenez le code du TP1 pour le chargement d'une image PGM
2. Construisez un petit programme qui charge une image et construit l'histogramme normalisé de cette dernière (vous exportez cet histogramme dans un fichier texte tabulé (1ere colonne = intensité, 2ième colonne = nombre de pixels ayant cette intensité / nombre total de pixels))
3. Générer les tracés des histogrammes via gnuplot (cf annexe). Qu'observez-vous sur l'histogramme des images PGM précédentes ?

Exercice 2. Transformation

Dans ce qui suit, nous considérons que l'histogramme est normalisé sur les intensités. Ainsi, les intensités de l'histogramme de l'intervalle $[0, M]$ sont normalisées dans $[0, 1]$.

1. Implémentez les transformations d'histogramme suivantes
 - Inversion d'histogramme
 - Gamma-correction : $i' = i^{\frac{1}{\gamma}}$. Vérifier son comportement sur $\gamma = 2.2$, $\gamma = \frac{1}{2.2}$
 - Interpolation linéaire $[a, b] \subset [0, 1] \rightarrow [0, 1]$

À chaque fois, vous illustrez la transformation par:

- un tracé gnuplot de la fonction ϕ de transformation d'histogramme
- deux tracés avant/après sur les histogrammes
- l'image PGM modifiée

Exercice 3. Egalisation

Comme décrit dans le cours, l'objectif est de construire une fonction de transfert ϕ entre l'histogramme de l'image, vue comme une distribution empirique de probabilité $p(i)$, vers une distribution cible. Dans le cas présent, nous cherchons à maximiser l'entropie de l'image en ciblant une distribution uniforme.

Etant donné un histogramme sur des intensités dans l'intervalle $[0, M]$, nous cherchons donc à cibler une distribution telle que la probabilité de chaque niveau de gris $i' = \phi(i)$ d'apparaître dans l'image est $p'(i') = \frac{1}{M}$.

Dans le cas de variables et distributions continues et pour une transformation ϕ croissante, nous avons :

$$p'(\phi(i)) = p(i) \frac{di}{di'} \quad (1)$$

Ainsi,

$$di' = Mp(i)di \quad (2)$$

Et donc,

$$\phi(i) = M \int_0^i p(\omega) d\omega \quad (3)$$

Dans le cas discret, la transformation ϕ s'écrit:

$$\phi(i) = M \frac{\sum_{j=0}^i \text{hist}(j)}{\sum_{j=0}^M \text{hist}(j)} \quad (4)$$

Implémentez l'égalisation d'histogramme décrite précédemment et testez cette transformation sur quelques images (avec tracés gnuplot de ϕ et des histogrammes avant/après)

Segmentation d'images par histogramme

Exercice 1. Seuillage naïf

- Construisez un petit programme qui charge une image, calcule son histogramme et seuil l'image initiale selon une certaine intensité.

Exercice 2. Minimisation de variance

On va s'intéresser différentes techniques pour automatiser la recherche d'un *bon* seuil.

L'objectif de la méthode de Fisher est de minimiser la variance intra-classe des deux classes (objet / fond). Plus formellement, nous considérons un histogramme normalisé H d'intensités $[0, M]$ et un seuil t définissant deux classes ($B=\text{background}$, $O=\text{object}$) de valeurs dans l'histogramme. Nous cherchons à minimiser la fonctionnelle

$$\sigma_{intra}^2(t) = n_B(t)\sigma_B^2(t) + n_O(t)\sigma_O^2(t) \quad (5)$$

avec

$$n_B(t) = \sum_{i=0}^{t-1} H(i) \quad (6)$$

$$n_O(t) = \sum_{i=t}^M H(i) \quad (7)$$

$$\mu_B(t) = \frac{1}{n_B(t)} \sum_{i=0}^{t-1} H(i) \quad (8)$$

$$\mu_O(t) = \frac{1}{n_O(t)} \sum_{i=t}^M H(i) \quad (9)$$

$$\sigma_B^2(t) = \frac{1}{n_B(t)} \sum_{i=0}^{t-1} (j - \mu_B(t))^2 H(i) \quad (10)$$

$$\sigma_O^2(t) = \frac{1}{n_O(t)} \sum_{i=t}^M (j - \mu_o(t))^2 H(i) \quad (11)$$

Question 1 Écrivez le bout de code qui permet de calculer $\sigma_{intra}^2(t)$. Pour une image donnée, calculez le $\operatorname{argmin}_{t \in 0..M}(\sigma_{intra}^2(t))$ et effectuez le seuillage avec ce seuil.

Otsu a observé que minimiser la variance intra-classe est équivalente à maximiser la variance entre les classes définies (après réécriture) par :

$$\sigma_{entre}^2(t) = n_B(t)n_O(t) (\mu_B(t) - \mu_O(t))^2 \quad (12)$$

Question 2 Ecrivez le code correspondant à la maximisation de $\sigma_{entre}^2(t)$. En pratique, ça ne doit pas changer le seuil optimal trouvé mais vous donner un algorithme plus efficace. Afin d'être efficace, n'oubliez pas de mettre à jour incrémentalement les quantités n_i et μ_i . Par exemple

$$n_B(t+1) = n_B(t) + H(t+1) \quad (13)$$

$$\mu_B(t+1) = \dots \quad (14)$$

Annexe Crash-course Gnuplot

Given a text file with tabular data (several columns, separated by spaces or tabs..) `toto.txt`

- Display a graph where abscissa/ordinate are mapped to first/third column

```
plot "toto.txt" using 1:3 with points
```

- Same graph with lines and points between datum

```
plot "toto.txt" using 1:3 with linespoints
```

- Y-axis is the sum of the second and third column and we limit the x-axis range to $[0, 10]$

```
plot [0:10] "toto.txt" using 1:($3+$2) with linespoints
```

- Labels

```
set xlabel "Abscissa"  
set ylabel "Abscissa"
```

- The next “plot” command will output the graph in a PDF file (with enhanced and color properties)

```
set terminal pdf enhanced color  
set output "glop.pdf"
```

- To use the X11 terminal (default one)

```
set terminal X11  
unset output
```

- Need some help on the plot command

```
help plot
```