# TP2
# Histogram manipulation

## Histogram analysis and transformation

**Exercise 1.** Histogram analysis and export

We want to improve the images `objects-dark.pgm` and `len_dark.pgm`.

1. Go back to the code from TP1 to load a PGM image.
2. Build a small program that load an image and that build its normalized histogram. You will export this histogram into a text file containing a table whose first column is the intensity, and whose second column is [number of pixels which have this intensity] / [Total number of pixels].
3. Use `gnuplot` to generate the drawing of this histogram (cf Appendix). What do you observe in the histograms of the considered PGM pictures?

**Exercise 2.** Transformation

In the following, we suppose that the histogram intensities are mapped from $[0, M]$ to $[0, 1]$.

Implement the following histogram transformations:

- Histogram inversion
- Gamma-correction : $i' = i^{\frac{1}{\gamma}}$. Check its behavior for $\gamma = 2.2$ and $\gamma = \frac{1}{2.2}$ [1]
- Linear interpolation $[a, b] \subset [0, 1] \to [0, 1]$

Every times, you will illustrate the transformation by:

- A `gnuplot` drawing of the histogram transformation function $\phi$.
- Two drawing (before/after) of the histograms.
- The modified PGM picture.

[1] Hint: $a^i$ = `pow(a,i)`, don't forget the `#include<math.h>`

**Exercise 3.** Equalization

As presented during the lesson, our aim is to build a transfer function $\phi$ between the picture histogram (seen as an empirical probabilistic distribution $P(i)$), toward a targeted distribution. Here, we want to maximize the entropy of the image by targeting a uniform distribution.

Given an histogram over the intensities of the interval $[0, M]$, we want a distribution such that the probability of each gray level $i' = \phi(i)$ is $P'(i') = \frac{1}{M}$.

In the case of continuous variables and distributions and for an increasing transformation $\phi$, we have:

$$P'(\phi(i)) = P(i)\frac{di}{di'} \tag{1}$$

Thus,

$$di' = M.P(i).di \tag{2}$$

And,

$$\phi(i) = M \int_0^i P(\omega) d\omega \tag{3}$$

In the discrete case, the transformation $\phi$ is the following:

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

**Implement the histogram equalization transformation previously described and test this transformation over several pictures (with the corresponding `gnuplot` drawings of $\phi$ and their histograms before/after transformation).**

## Image segmentation by using the histogram

**Exercise 1.** Naive Threshold (fr: Seuillage naif)

- Build a small program that loads a picture, computes its histogram and threshold the initial image according to a given intensity.

**Exercise 2.** Variance minimization

We will interest ourselves to different techniques to automatize the search of a *good* threshold.

Fisher's method aims at minimizing the intra-class variance of the two classes (objects / background). More formally, let us consider a normalized histogram $H$ whose intensities is in $[0, M]$. Let us consider a threshold $t$ defining 2 classes (*B=background, O=object*) of values in the histogram. We want to minimize the functional:

$$\sigma^2_{intra}(t) = n_B(t)\sigma^2_B(t) + n_O(t)\sigma^2_O(t) \tag{5}$$

where

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

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

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

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

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

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

**Question 1** Write a function which computes $\sigma^2_{intra}(t)$. Then, for a given image, write down a program which computes $argmin_{t \in 0..M}(\sigma^2_{intra}(t))$, and use the corresponding threshold.

Otsu observed that minimizing the intra-class variance is equivalent to maximizing the variance between the classes defined by:

$$\sigma^2_{entre}(t) = n_B(t)n_O(t)\left(\mu_B(t) - \mu_O(t)\right)^2 \tag{12}$$

**Question 2** Similarly, write down a program that maximize $\sigma^2_{entre}(t)$. You should not find another optimal threshold value, but you should compute it more effectively. To be effective, do not forget to update incrementally $n_i$ et $\mu_i$, for example by using the following identities:

$$n_B(t+1) = n_B(t) + H(t+1) \tag{13}$$

$$\mu_B(t+1) = .... \tag{14}$$

## **Appendix -** 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 "Abscisse"
set ylabel "Ordonnee"
```

- 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
```