Contents

1 Introduction

In this document we present the differents traitement make in the librairy.

2 GrayScale

The last operation made on the image before displaying it is a grayscale conversion. It converts the image (in float format) in an 8 bits image (in int format). The number of gray value can be edited as a value of 2^N, with $N \in [1, 8]$. Whatever the value of N, the gray values are equally reparted between 0 and 255:

- $\{0, 255\}$ for $N = 2$,
- $\{0, 85, 170, 255\}$ for $N = 3$ ...

The lower and upper bounds (LB and UB respectively) can be edited, they define the maximum input value such as:

$$\forall P^{\text{in}}_{i,j} < LB, P^{\text{out}}_{i,j} = 0, \tag{1}$$

$$\forall P^{\text{in}}_{i,j} > UB, P^{\text{out}}_{i,j} = 255, \tag{2}$$

where $P^{\text{in}}_{i,j}$ is the pixel i, j of input image and $P^{\text{out}}_{i,j}$ is the pixel i, j of output image. The function defining the curve relating input and output can be:

- linear,
- exponential,
- logarithmic,
- sigmoid.

The shape of this curbe is shown in Figure 1.

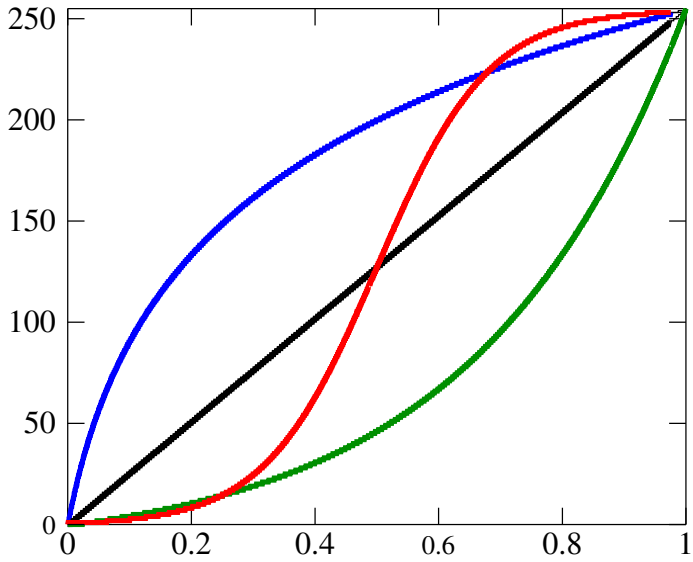

Figure 1: The four grayscale conversion implemented: linear (black), exponential(green), logarithmic (blue) and sigmoid (red). $N = 8$, $LB = 0$, $UB = 1$.

2.1 Linear gray scale

Simple gray scale conversion from 0 to 255.

2.2 Exponential gray scale

Here, the gray scale conversion is an exponential function. The distance between the linear curve and exponential curve can be modified internally by changing fac parameter (= 3 by default). The green curve y in Figure 1 is defined by:

$$y(x) = \mathrm{e}^x, x \in [0, fac]\,, \tag{3}$$

a change of variable is made such that the input for LB to UB correspond to x in eq. 3. Then a post traitement is made on $y(x)$ such that it varies from 0 to 255 (with step of 1 on Figure 1).

2.3 Logarithmic gray scale

Here the gray scale conversion is the bijective function of the exponential gray scale. So it is defined by:

$$y(x) = \log_{10}(x), x \in \left[1, \mathrm{e}^{fac}\right], \tag{4}$$

a change of variable is made such that the input for LB to UB correspond to x in eq. 3. Then a post traitement is made on $y(x)$ such that it varies from 0 to 255 (with step of 1 on Figure 1).

2.4 Sigmoid gray scale

Here the gray scale conversion is a sigmoid function (see: `https://fr.wikipedia.org/wiki/Sigmo%C3%AFde_(math%C3%A9matiques)`). It is defined by:

$$y(x) = \frac{1}{1 + \mathrm{e}^{-x}}, x \in [-5.54126, 5.54126]\,, \tag{5}$$

the bounds $[-5.54126, 5.54126]$ has been chosen because when we rescale $y(x) \to Y(x)$ such that $Y(x)$ varies from 0 to 255 ($Y(x) = \mathrm{E}\left(255y(x)\right)$), we have:

$$Y(x) >= 1, x > -5.54126, \tag{6}$$

$$Y(x) <= 254, x < 5.54126, \tag{7}$$

this mean that $Y(x) = 0$ only if input $<=\ LB$ and $Y(x) = 255$ only if input $>=\ UB$

3 Filter

We use three possible filters on the pixels of the image:

- no filter,
- convolution,
- median.

3.1 No filter

No filter is applied, the output image is a copy of the input image.

3.2 Convolution filter

The convolution filter take into acount the direct attenant pixels (see: `https://en.wikipedia.org/wiki/Kernel_(image_processing)` for example), a pixel of the output image is calculated using the value of 9 pixels (included itself), with a 3x3 matrix kernel K:

$$K = \begin{bmatrix} W_{0,0} & W_{0,1} & W_{0,2} \\ W_{1,0} & W_{1,1} & W_{1,2} \\ W_{2,0} & W_{2,1} & W_{2,2} \end{bmatrix}, \tag{8}$$

The value of output pixel is given by:

$$P_{i,j}^{\text{out}} = \sum_{k=0}^{k=2} \sum_{l=0}^{l=2} W_{k,l} P_{i-1+k,j-1+l}^{\text{in}}, \tag{9}$$

where $P_{i,j}^{\text{out}}$ is the pixel (i,j) of the output image and $P_{i,j}^{\text{in}}$ is the pixel (i,j) of the input image. The matrix kernel K must be normalized:

$$\sum_{i=0}^{i=2} \sum_{j=0}^{j=2} W_{i,j} = 1, \tag{10}$$

to keep the amplitude of the pixel.

In case we are at a border or corner of the image we can't use the whole kernel matrix, we use a submatrix kernel. For example for the top line of an image $i = 0$, we use the subkernel k^{top} defined by:

$$K^{\text{top}} = \begin{bmatrix} W_{1,0}^{\text{top}} & W_{1,1}^{\text{top}} & W_{1,2}^{\text{top}} \\ W_{2,0}^{\text{top}} & W_{2,1}^{\text{top}} & W_{2,2}^{\text{top}} \end{bmatrix}, \tag{11}$$

knowing that this subkernel must also be normalized:

$$\sum_{i=1}^{i=2} \sum_{j=0}^{j=2} W_{i,j}^{\text{top}} = 1. \tag{12}$$

3.3 Median filter

The median filter such as the convolution filter is applied on 9 pixels (the center pixel and the 8 adjacent pixels). Output is the median value of the 9 pxiels (for 9 values, the median is the fifth lowest value). In case we are at a border or corner we apply the median filter on 6 or 4 values respectively (the median will be the third and second lowest value respectively);

4 Scan conversion