

Master Automatique, Robotique,
MU4RBI08 / MU4RBI07 – Traitement d'images
UE Traitement des images (3 ECTS)
Master 1 – 2^{ème} semestre

TP 1

Implémenter les transformations ci-dessous.
Implement the transformation suggested below.

1. Implémenter une fonction qui extrait la **composante rouge** de l'image.
*Implement a function able to extract the **red component** of an image.*



"Red" component generation.

2. Implémentez une fonction qui va créer le **négatif d'une image** (soustrayez de 255 chaque composante RGB de chaque pixel. Cela inverse l'intensité de chaque couleur : le clair devient foncé et vice versa).
*Implement a function able to generate the **negative of an image** (from 255, you will so need to subtract the intensity's component – RGB - of each pixel. This will invert each color's intensity of each pixel: the bright will become dark and vice versa).*



"Negative" image generation.

3. **Convertir une image couleur en noir et blanc (niveaux de gris).** Dans une telle image, chaque pixel est noir, blanc, ou a un niveau de gris entre les deux. Cela signifie que les trois composantes ont la même valeur. La formule standard donnant le niveau de gris en fonction des trois composantes couleur est :

$$\text{gray} = 0.299 * \text{red} + 0.587 * \text{green} + 0.114 * \text{blue}$$

Convert a color image in black and white (gray level). In such an image, each pixel is black, white or has a gray level between 0 and 1. This means that the components have the same value. The standard formula giving the grey level function of the components is:

$$\text{gray} = 0.299 * \text{red} + 0.587 * \text{green} + 0.114 * \text{blue}$$





"B&W" (gray level) image generation

- 4. Convertir une image couleur en nuances de sépia.** En photographie, le sépia est une qualité de tirage qui ressemble au noir et blanc, mais avec des variations de brun, et non de gris. Dans la transformation d'une image couleur en une image en nuances de sépia, on tient compte des transformations suivantes :

$$\text{outputRed} = (\text{inputRed} * .393) + (\text{inputGreen} * .769) + (\text{inputBlue} * .189)$$

$$\text{outputGreen} = (\text{inputRed} * .349) + (\text{inputGreen} * .686) + (\text{inputBlue} * .168)$$

$$\text{outputBlue} = (\text{inputRed} * .272) + (\text{inputGreen} * .534) + (\text{inputBlue} * .131)$$

NB : en cas de saturation (intensité résultante > 255), mettre la valeur à 255.

Convert a color image to sepia tones. In photography, sepia is a print quality that looks like black and white, but with variations of brown, not gray. In the transformation of a color image into an image in sepia tones, we consider the next transformations:

$$\text{outputRed} = (\text{inputRed} * .393) + (\text{inputGreen} * .769) + (\text{inputBlue} * .189)$$

$$\text{outputGreen} = (\text{inputRed} * .349) + (\text{inputGreen} * .686) + (\text{inputBlue} * .168)$$

$$\text{outputBlue} = (\text{inputRed} * .272) + (\text{inputGreen} * .534) + (\text{inputBlue} * .131)$$

NB: in case of saturation (resulting intensity > 255), set the value to 255.



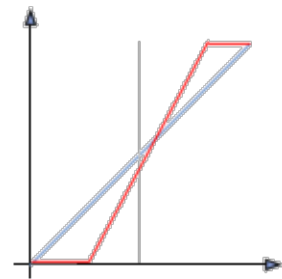


"Sepia" effect

5. **"Contrast"**. Pour chaque composante de chaque pixel, appliquez les règles suivantes :

- Si la valeur est plus petite que 30, assignez la valeur 0.
- Si la valeur est plus grande que 225, assignez la valeur 255.
- Les valeurs c comprises entre 30 et 225 seront recalculées avec la formule : $(255.0 / 195.0) * (c - 30) + 0.5$

Avec une calculatrice, vous constaterez que, par exemple : 30 devient 0, 225 devient 255, 45 devient 20, 180 devient 196, 120 devient 118. Les valeurs "s'écartent" les unes des autres, ce qui augmente le contraste.

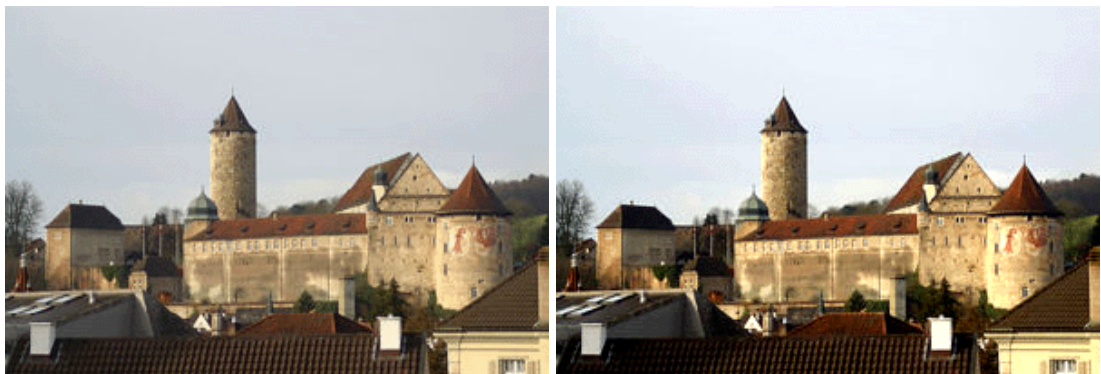


"Contrast". For each component of each pixel, apply the following rules: a. If the value is less than 30, assign the value 0.

b. If the value is greater than 225, assign the value 255.

c. The c values between 30 and 225 will be recalculated with the formula: $(255.0 / 195.0) * (c - 30) + 0.5$.

With a calculator, you will find that, for example: 30 becomes 0, 225 becomes 255, 45 becomes 20, 180 becomes 196, 120 becomes 118. The values "deviate" from each other, which increases the contrast.



Look Up Table LUT (courbe tonale) and "Contrast" effect

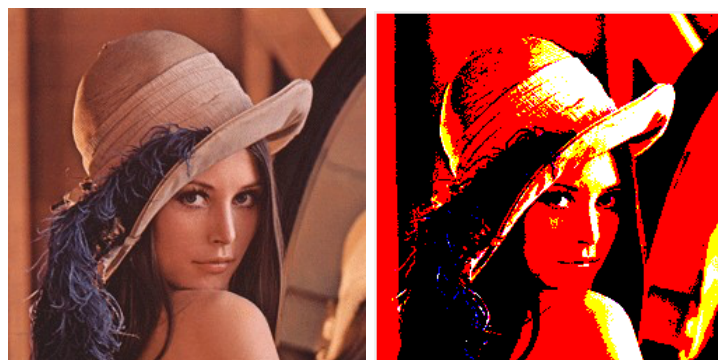
6. **Seuillage.** Le seuillage d'image est la méthode la plus simple de segmentation d'image. À partir d'une image en niveau de gris, le seuillage d'image peut être utilisé pour créer une image comportant uniquement deux valeurs, noir ou blanc (monochrome). On remplace un à un les pixels d'une image par rapport à une valeur seuil fixée (par exemple 123). Ainsi, si un pixel a une valeur supérieure au seuil (par exemple 150), il prendra la valeur 255 (blanc), et si sa valeur est inférieure (par exemple 100), il prendra la valeur 0 (noir).

Thresholding. Image thresholding is the simplest method of image segmentation. From a grayscale image, image thresholding can be used to create an image with only two values, black or white (monochrome). The pixels of an image are replaced one by one with respect to a fixed threshold value (for example 123). Thus, if a pixel has a value greater than the threshold (for example 150), it will take the value 255 (white), and if its value is lower (for example 100), it will take the value 0 (black).



Avec une image en couleur, on fera de même avec les trois composantes rouge, vert et bleu. Il y aura ainsi huit couleurs possibles pour chaque pixel : blanc, noir, rouge, vert, bleu, magenta, jaune et cyan.

With a color image, we will do the same with the three components red, green and blue. There will thus be eight possible colors for each pixel: white, black, red, green, blue, magenta, yellow and cyan.



7. **"Flip"**. Produire une symétrie axiale d'axe horizontal.

Flip the image around the horizontal axis.



"Flip"

8. **"Border"** créera un bord bleu d'une largeur de 5 pixels.

Generate a blue border with a width of 5 pixels.



"Border"

9. **"Relief"** : Créer un bord qui donnera une impression de relief. Il sera constitué de 4 régions polygonales d'une largeur de 10 pixels. Les pixels de la région du haut seront éclaircis de 65, ceux de la région du bas assombris de 65. A droite et à gauche, ils seront assombris de 40. La difficulté de l'exercice réside dans la gestion des coins...

Create an edge that will give an impression of relief. It will consist of 4 polygonal regions with a width of 10 pixels. The pixels of the top region will be lightened by 65, those of the bottom region darkened by 65. On the right and left, they will be darkened by 40. The difficulty of the exercise lies in managing the corners ...



"Relief"

10. "Pixeliser" (x10). L'idée est de diviser l'image en carrés de 10 pixels sur 10 (dans les bords, ce seront des rectangles). Dans un "carré", chaque composante de chaque pixel sera la valeur moyenne du carré.

"Pixelize" (ordinal = 10). The idea is to divide the image into squares of 10 by 10 pixels (in the edges, they will be rectangles). In a "square", each component of each pixel will be the average value of the square.



Pixelize