

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

Prof. Daniel Racocceanu
Sorbonne Université, Faculté des Sciences et Ingénierie
daniel.racocceanu@sorbonne-universite.fr

TP 2

1. **Lisibilité.** Que doit-on faire pour augmenter la lisibilité de l'image aquitain.tif (dans le dossier du TP) ?

Readability. *What should we do to increase the readability of the image aquitain.tif (in lab's folder)?*

2. **Filtrage.** Le principe du filtrage est de modifier la valeur des pixels d'une image, généralement dans le but d'améliorer son aspect. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine. Un filtre est une transformation mathématique (appelée produit de convolution) permettant de modifier la valeur d'un pixel en fonction des valeurs des pixels avoisinants, affectées de coefficients. Les calculs sont faits pour chacune des trois composantes de couleur. Le filtre est représenté par un tableau (une matrice), caractérisé par ses dimensions et ses coefficients, dont le centre correspond au pixel concerné. Habituellement, la somme des coefficients fasse 1, afin d'obtenir une valeur habituelle du niveau de gris résultant, quelle que soit la situation initiale (NB : il est possible de travailler avec des filtres ne vérifiant pas cette propriété mais le résultat devra être plafonné – il y aura donc une perte d'information). Rappelons que les valeurs des composantes des pixels sont des nombres entiers compris entre 0 et 255. Si les nouvelles valeurs ne sont plus des entiers, il faudra les arrondir.

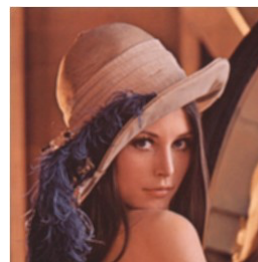
The principle of filtering is to modify the value of the pixels of an image, generally in order to improve its appearance. In practice, this involves creating a new image using the pixel values of the original image. A filter is a mathematical transformation (called a convolution product) allowing the value of a pixel to be modified as a function of the values of neighbouring pixels, assigned coefficients. Calculations are made for each of the three colour components. The filter is represented by a table (a matrix), characterized by its dimensions and its coefficients, the centre of which corresponds to the pixel concerned. Usually, the sum of the coefficients from the convolution kernel to be equal to 1, in order to obtain a usual value of the resulting gray level, whatever the initial situation (NB: it is possible to work with filters not checking this property but the result will have to be limited - so there will be a loss of information). Recall that the values of the components of the

pixels are originally integers between 0 and 255. If the new values are not integers, they will have to be rounded.

- a. Lissage.** Le tableau ci-dessous rend l'image plus floue. On dit que c'est un filtre passe-bas. Appliquer ce tableau revient en fait à remplacer la valeur de chaque pixel par la moyenne des 9 pixels formant un carré autour du pixel visé. Implémenter et tester ce filtre.

Smoothing. The table below makes the image more blurred. This is an example of low pass filter. Applying this kernel consists in replacing the value of each pixel by the average of the 9 pixels forming a square around the targeted pixel. Implement and test this filter.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9



- b. Accentuation.** À l'inverse, le tableau ci-après rendra l'image plus nette. C'est un filtre passe-haut. Attention ! Il peut arriver que la nouvelle valeur ne soit plus comprise entre 0 et 255. Il faudra donc toujours prendre $\min(x, 255)$ et $\max(x, 0)$, où x est la nouvelle valeur. Implémenter et testez ce filtre.

Accentuation. At the opposite, the table below will make the image sharper. This is an example of high pass filter. Remark: it may happen that the new value is no longer between 0 and 255. It will therefore always be necessary to take $\min(x, 255)$ and $\max(x, 0)$, where x is the new value. Implement and test this filter.

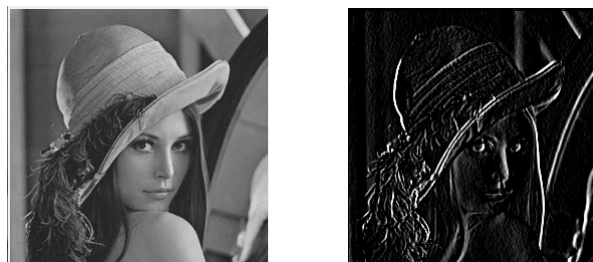
0	-0.5	0
-0.5	3	-0.5
0	-0.5	0



- c. **Gradient** (filtre de Sobel). Pour faire simple, l'opérateur calcule le gradient de l'intensité de chaque pixel. Ceci indique la direction de la plus forte variation du clair au sombre, ainsi que le taux de changement dans cette direction. On connaît alors les points de changement soudain de luminosité, correspondant probablement à des bords. Implémentant ce filtre et testez-le sur Matlab.

Gradient (Sobel filter). To make it simple, the operator calculates the gradient of the intensity of each pixel. This indicates the direction of the greatest change from light to dark, as well as the rate of change in that direction. We then know the points of sudden change in brightness, probably corresponding to edges. Implement and test this filter.

-1	0	1
-2	0	2
-1	0	1



- d. **Le filtre médian.** La technique de filtre médian est largement utilisée en traitement d'images numériques, car elle permet de réduire le bruit tout en conservant les contours de l'image. L'idée principale du filtre médian est de remplacer chaque pixel par la valeur médiane de son voisinage. Ce type de filtre médian est appelé « marginal ». Considérons neuf pixels en niveaux de gris, dont une valeur est aberrante (ici 255):

The median filter. The median filter technique is widely used in digital image processing because it reduces noise while retaining the edges of the image. The main idea of the median filter is to replace each pixel with the median value of its vicinity. This type of median filter is called "marginal". Let us consider nine grayscale pixels, one of which is an outlier (here 255):

2	4	12
2	255	3
7	9	3

Le filtre médian va d'abord trier ces valeurs par ordre croissant :
The median filter will first sort these values in ascending order:

2, 2, 3, 3, 4, 7, 9, 12, 255

et prendre la valeur médiane (la cinquième valeur), ici la valeur 4. La sortie du filtre donnera :

and take the median value (the fifth value), here the value 4. The filter output will give:

2	4	12
2	4	3
7	9	3

Développer un programme en deux étapes. La première ajoutera du bruit à une image : un certain nombre de pixels aléatoires seront remplacés par des pixels blancs. On appellera cette instruction « bruit ». La seconde permettra d'éliminer ce bruit en utilisant un filtre médian. Pour une image en couleurs, on appliquera ce filtre aux trois composantes RVB séparément. On appellera cette instruction « median ».

Develop a script in two steps. The first will add noise to an image: a certain number of random pixels will be replaced by white pixels. This instruction will be called "noise". The second will eliminate this noise using a median filter. For a color image, this filter will be applied to the three RGB components separately. This instruction is called "median".

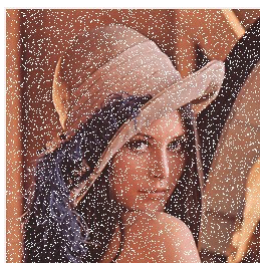


Image bruitée



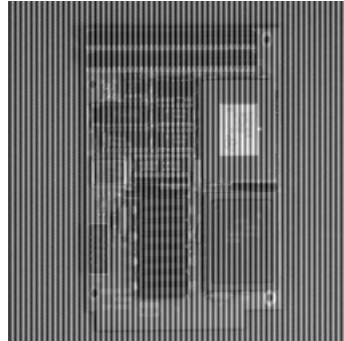
Image filtrée 1 fois



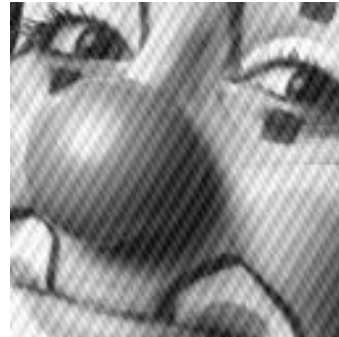
Image filtrée 2 fois

- 3. Filtrage bruit périodique.** Proposez un algorithme / script qui permet de « nettoyer » les images *noise.tif* et *clown.tif*

Clean-up periodical noise. Propose an algorithm / script to suppress the periodical noise in images *noise.tif* and *clown.tif*



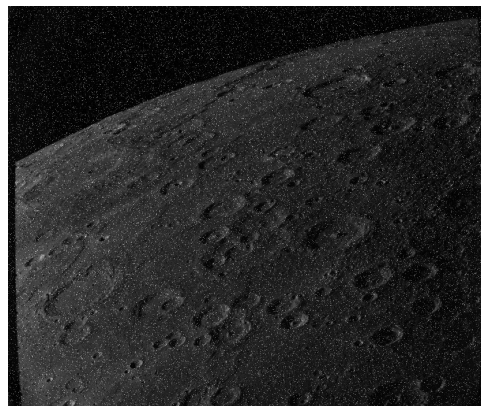
Noise.tif



Clown.tif

4. **Atenuation / élimination du bruit.** Proposez un script qui permet d'éliminer le bruit de l'images *mercury.tif* et *noisy_Lena.png*.

Noise reduction / suppression. Propose a script to clean *mercury.tif* and *noisy_Lena.png* images.



Mercury.tif



noisy_Lena.png