

# TP 1 : Basic tools for image processing

This first tutorial aims at introducing the various basic tools available in the Matlab *image processing* library. You are invited to regularly consult the documentation of the different Matlab functions to understand the operations they perform :

```
>> help function    % short description
>> doc function    % longer description and detailed examples.
```

A certain number of tools are identical to those you used in digital signal processing labs and random signal processing labs, so don't hesitate to consult your previous codes to remember how to use the functions.

## 1 Visualization

The image 'Emphysema\_H\_and\_E.jpg' is an image of haematoxylin and eosin stained lung tissue sample taken from an end-stage emphysema patient. Cell nuclei appear in blue-purple, red blood cells are red, other cell bodies and extracellular material are pink, and air spaces are white. The original image comes from "T Cells Cause Lung Damage in Emphysema", PLoS Med 1(1) : e25, 2004, <https://doi.org/10.1371/journal.pmed.0010025>".

1. Load the image 'Emphysema\_H\_and\_E.jpg' with `imread`, and display it with `imshow`.
2. Convert the color image to grayscale with `rgb2gray`, then display the result.
3. Display each of the red, green and blue components of the color image on separated figures. Display the corresponding colorbar.
  - (a) Can you make the connection between the information carried by the three components and the color image displayed in question 1 ?
  - (b) Can you make the connection between the information carried by the three components and the different biological objects ?
4. Display the components of hue, saturation and value of the color image, computed with `rgb2hsv`. To display the hue as a color image, we can set to 1 the saturation and the value of the HSV image, then convert it into RGB image with the function `hsv2rgb`.
5. Display the image converted into grayscale with a false color representation : `imshow(myImage,'Colormap',jet)`. Display the corresponding colorbar.
6. Convert the grayscale image in double. Display the converted image by specifying the intensity range in `imshow`. Test successively : `[0,1]`, `[0,100]`, `[100,255]` and `[]`. What is the effect of using `[]` on the image display ?

## 2 Load and read different image formats

A digital image is represented by a matrix or an array whose dimension depends on the nature of the image (grayscale, color, multispectral, etc). The format of the image (jpeg, png, tif, nifti, fits, mat, etc) depends on the way the image is stored in memory and the software used to produce/store/read the image.

1. Load the image 'Emphysema\_H\_and\_E.jpg' with `imread`. How the intensities are coded ?

2. Load the image 'fractured\_spine.tif' with `imread`. How the intensities are coded?
3. Load the computed tomography scan contained in file 'CT.mat' with function `load`.
  - (a) Look carefully at how the image is stored in this Matlab structure and how you can access it.
  - (b) How the intensities are coded?
  - (c) Display the image with `imshow` with the corresponding colorbar.
  - (d) Convert the image in double and display the image with `imshow` with the corresponding colorbar.
  - (e) Calculate the min and max intensities of the image. What is the problem on the two previous displayed image?
  - (f) How to set up the `imshow` function so that the display of the intensities is correct?
  - (g) What is the default setting to use if you don't know the image intensity range in advance?

File with .mat extension are Matlab structure used to store different types of data like images, metadata, etc. Other formats like .nifti, .fits, etc offer also the possibility to store images and metadata. Specific functions must be used to open these images. The `imread` function is dedicated to load images stored in classical image formats (.png, .tif, .jpeg) whose intensities are coded on integers.

### 3 Histogram and image enhancement

1. Load the image contained in file 'CT.mat'.
2. When interpreting the images, the radiologist chooses a visualization dynamic (color scale) adapted to the organs he wants to analyze. Test successively : `[]`, `[-150, 250]` and `[-1400, 200]` and analyze the differences.
3. Knowing the min and max intensities of this image, then convert it into an image with gray levels between 0 and 255 (once the scaling is done, take care to convert the image into `uint8`). Display the histogram of this converted image with `imhist` function. We will note thereafter this image  $I_{\text{norm}}$ .
4. We now want to display the histogram of the original image. In this case, `imhist` does not work because the image is not coded on 256 gray levels. Use the `hist` function for this. This function requires a vector as input. We can convert the image into a vector using the syntax `I(:)`. Test the influence of the choice of the number of bins (default value, Sturges rule :  $1 + \ln(n)$  with  $n$  the number of pixels in the image, max-min, ...).
5. On the  $I_{\text{norm}}$  image, apply the following algebraic operations :  $I_{\text{norm}}^2$ ,  $I_{\text{norm}}^4$ ,  $\log(I_{\text{norm}})$ . Visualize and comment on their effects. What happens if we don't take care of converting the  $I_{\text{norm}}$  image into double beforehand?
6. Perform a histogram equalization (`histeq`) of the  $I_{\text{norm}}$  image. Is the objective of the method achieved?

### 4 Fourier Transform

1. Load the image `sin32.png`.
2. Compute its discrete Fourier transform (DFT) with the `fft2` function (after converting the image into double).
3. Display the modulus and the angle of the DFT of the image. The DFT is usually represented with the low frequencies in the center of the image, unlike the result calculated by `fft2`. To move the low frequencies back to the center, use `fftshift` on the result of `fft2`.
4. How is the modulus of the DFT interpreted?
5. Perform the same operations with the images `sin16.png` and `sin8.png`. What do you conclude?

## 5 High and low frequencies of an image

1. Load the image contained in file 'CT.mat'.
2. Compute its discrete Fourier transform (DFT) with the function `fft2` and `fftshift`. Should the Fourier transform be calculated on the color image or on the grayscale transformed image?
3. Multiply its Fourier transform with a rectangle function of width  $\Delta f$  in order to recover only the low frequencies (we will build the rectangle function by using the `zeros` function, then by setting to 1 the values at the center of the image).
4. Calculate the inverse Fourier transform of the result (`fftshift` then `fft2`) in order to visualize the corresponding low frequency image.
5. Repeat the same operations but this time recovering the high frequencies.
6. Vary  $\Delta f$  and watch the influence on the corresponding high and low frequency images.

## 6 Aliasing phenomenon

The aliasing is a phenomenon that appears when the image is undersampled without respecting the Shannon theorem. This phenomenon can be avoided by filtering the image properly before sampling. The purpose of this exercise is to compare subsampling without a filter, with a Gaussian filter and then with an ideal filter.

1. Load and display the image 'onion\_cells.jpg' and convert it to grayscale and double.
2. Subsample the image by a factor of two, this can be done with the instruction `I(1 : 2 : end, 1 : 2 : end)`. How does the aliasing phenomenon translate visually?
3. Display the modulus of the Fourier transform of the subsampled image and compare it to the modulus of the Fourier transform of the original image. How does the aliasing phenomenon translate visually on the Fourier transform? Look at the top left edge of the image and the edge at the bottom right of the image.
4. By subsampling the image, is the sampling rate changed? Is Shannon's theorem respected? What is the one-dimensional equivalent of aliasing called?
5. Apply a Gaussian filter (`imgaussfilt` function with 'circular' option) to the image before subsampling. Adjust the filter parameters to minimize the aliasing. Depending on the value of sigma, choose a sufficiently large value for the parameter `FilterSize`.
6. Apply an ideal filter to the image before subsampling. Choose the filter cutoff frequencies appropriately to reduce aliasing without altering the image. An ideal low-pass filter completely removes frequencies above the cutoff frequency without altering the frequencies below. To implement the ideal filter, it is easiest to work in the Fourier domain to define the ideal filter.