Introduction to Computer Vision

ISAE-SUPAERO

- Emmanuel Zenou -

2019 - 2020

The objective of this course is to introduce Image Processing and Computer Vision principles under MatLab.

1 Basics

Commands `imread` et `imshow` allow to respectively load and display an image.

▶ **Question 1** *Load and Display a grayscale image and a color image. How do you interpret the image coding under* MatLab*? What is the data type?*

The command `figure` open a new window and increment the numbering. Open and display all images. `Close all` allows to close all windows at once.

1.1 Greyscale Image Coding

Now, we will build the matrix whose image could be displayed on the screen. By default, data type is double. Check for min (for black) and max (for white) values in double type.

▶ **Question 2** *Build a matrix with a gradual value of intensity and an horizontal line with a constant value; the representative image is shown Fig. 1.*

Figure 1: Grayscale Illusion

▶ **Question 3** *Build a matrix of black & white stripes, and with a rectangle and disk as shown Fig. 2. All parameters (size of stripes, size of rectangle, radius) should be easily modifiable.*

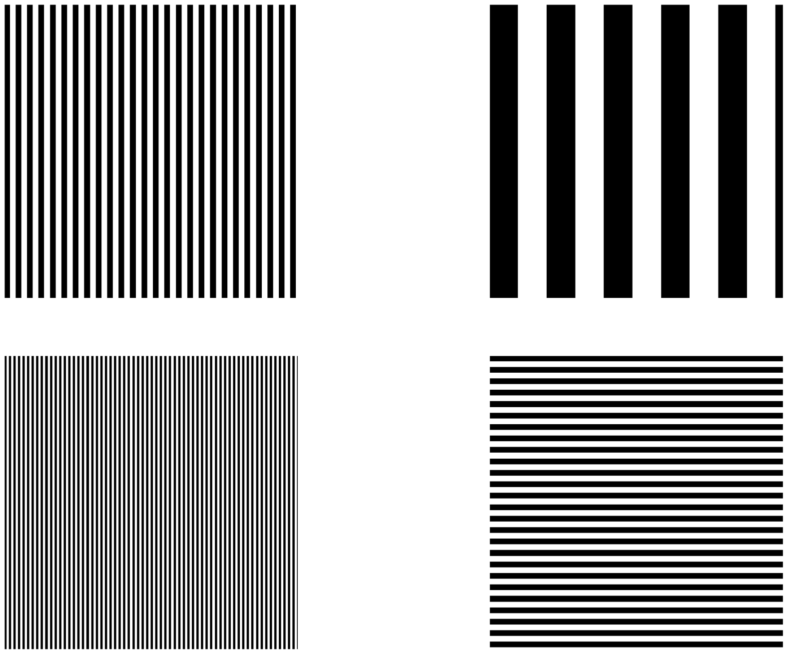

Figure 2: Stripes

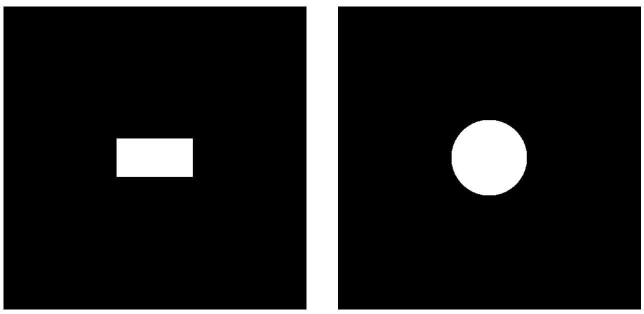

Figure 3: Rectangles & Disks

1.2 Colour Image Coding

▶ **Question 4** *Next, display Teinte.jpg and its red, green and blue components. Interpret and analyse. Same with oeil.jpg, cargo.jpg and CoulAdd.jpg.*

▶ **Question 5** *Build and display the french flag. Build and display your flag.*

▶ **Question 6** *Use the HSV code (with the command* `rgb2hsv` *and interpret images. How is the type of the new matrix? Build and display the image Fig. 4.*

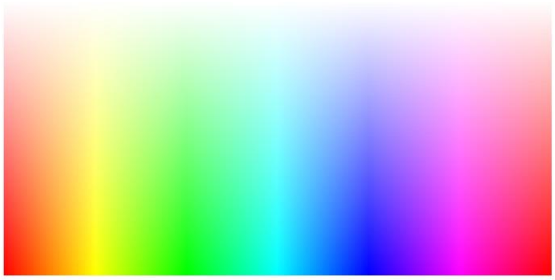

Figure 4: HSV Color Space

`rgb2gray` allows to transform a colour image to a grayscale image:

$$I = \alpha \times r + \beta \times g + \gamma \times b$$

▶ **Question 7** *What are the values of α, β and γ ?*

▶ **Question 8** *Load ans display SpainBeach.png and isolate the beach.*

1.3 Histograms

It is easy to display the histogram of an image under MatLab with the command `imhist`.

▶ **Question 9** *What is an histogram? What is the use? Display and interpret histograms of images?*

It is also easy to manage histogram with the command `histeq`.

▶ **Question 10** *Work the mysterious images called Imagex.bmp and Imagexx.bmp.*

1.4 Filtering

Filtering can be associated to blur and to edge detection. Commands `imfilter` and `fspecial` can be used to respectively filter images and define kernel (which can be also defined as a matrix).

▶ **Question 11** *Apply blur Filtering and Edge filtering on the Stripes images and on a 'real' image. What are the main associated Kernels ?*

▶ **Question 12** *Thanks to successive filtering operators, isolate the main 5 stars of the image Etoile.png.*

Report starts from here

---

2 Fourier Transform

The Fourier transform is a very powerful tool to extract information on an image. Let us recall these formulas:
DFT: Direct Fourier Transform:

$$F(p,q) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} I(n,m).e^{-j(\frac{2\pi}{N})pn}.e^{-j(\frac{2\pi}{M})qm}$$

IFT: Inverse Fourier Transform:

$$I(m,n) = \frac{1}{MN} \sum_{p=0}^{N-1} \sum_{q=0}^{M-1} F(p,q).e^{+j(\frac{2\pi}{N})pn}.e^{+j(\frac{2\pi}{M})qm}$$

MatLab calculates the FT of an image thanks to `fft2` (followed by `fftshift`).

▶ **Question 13** *Get the FT and analyze the spectrum if images with stripes (Fig. 2), rectangle and disks (Fig. 3).*

▶ **Question 14** *Blur the image with different kernels and interpret the spectrum.*

▶ **Question 15** *Write a program that extracts the specific field in the image Champs.jpg (see Fig. 5).*

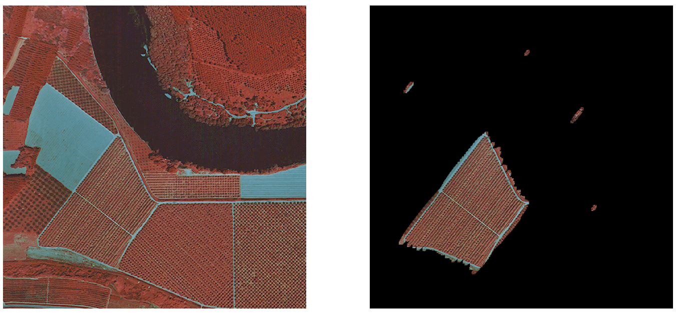

Figure 5: Extraction du champ par transformée de Fourier.

3 Deblurring

In this section, we will see some examples of image deblurring algorithms through linear modelization.

The is main source of blur in an image:

- Bad focalization,
- Moving blur,
- Atmosphere turbulences
- etc.

3.1 Linear Modelization

As a first approximation, a blurred image could be modeled as following:

$$\mathcal{Y} = \mathcal{H} * \mathcal{I} + \mathcal{B}$$

with

- $\mathcal{Y}$ the blur image to be restored
- $\mathcal{H}$ the convolution kernel,
- $\mathcal{I}$ the original image to be estimated,
- et $\mathcal{B}$ a Gaussian additive noise.

▶ **Question 16** *Why do we have chosen this model? What are the limits of this model with respect to a single lens model?*

Load the image of Toulouse (`toulouse.bmp`), blur this image with a $(2T+1)$-sized square kernel (Typically $T = 3$). The convolution kernel is thus $h(x, y) = \alpha$ if $|x| \leq T$ and $|y| \leq T$, with $\alpha = \frac{1}{(2T+1)^2}$.

▶ **Question 17** *Compare the spectrums of the original image and of the blur image: what do you observe? Justify.*

3.2 Blur Estimation

The objective is here to estimate a posteriori the value of T, *i.e.* from the blurred image.

In dim 1, $h(x) = \sqrt{\alpha}$ si $|x| \leq T$. The DFT is

$$\mathcal{H}(u) = \sum_{x=-T}^{+T} h(x) e^{-j2\pi \frac{ux}{N}} = \frac{1}{2T+1} \sum_{x=-T}^{+T} w^x$$

However,

$$\sum_{x=-T}^{+T} w^x = \frac{w^{-T} - w^{T+1}}{1-w} = \frac{w^{-T-\frac{1}{2}} - w^{T+\frac{1}{2}}}{w^{-\frac{1}{2}} - w^{\frac{1}{2}}}$$

Thus

$$\mathcal{H}(u) = \frac{1}{2T+1} \frac{sin(2\pi \frac{u}{N}(T+\frac{1}{2}))}{sin(\pi \frac{u}{N})}$$

In this case, $N = 512$, $u = -512 \ldots 512$. In dim 2, $h(x, y) = h(x)h(y)$

▶ **Question 18** *Propose a cardinal sinus function that have the same zeros than $\mathcal{H}(u)$ (superpose the two functions). What conclusion could you give from this properties? Could you estimate T ?*

Compare the the original image spectrum and the blurred image spectrum (using eventually the log function):

▶ **Question 19** *Estimate T.*

3.3 Image Deblurring

Now, image deblurring (or image restauration) could be done by many methods. Let us focus on two of them

- Inverse Filtering,
- Wiener Filtering.

Let $g(x, y)$ be the inverse filter of $h(x, y)$. In the spectral Domain, we have:

$$\mathcal{G} = \frac{1}{\mathcal{H}}$$

The estimated restaured image $\widehat{\mathcal{I}}$ is:

$$\widehat{\mathcal{I}} = \mathcal{G}\mathcal{Y} = \widehat{\mathcal{I}} = \mathcal{G}\mathcal{H}\mathcal{I} + \mathcal{G}\mathcal{B} = \widehat{\mathcal{I}} = \mathcal{I} + \mathcal{G}\mathcal{B}$$

The following program allows to apply this process:

```
SeuilMax = 11 ;
hh = zeros(TailleImage);
centre = [1 1] + floor(TailleImage/2) ;
ext = (TailleFiltre-[1 1])/2;
ligs = centre(1) + [-ext(1):ext(1)];
cols = centre(2) + [-ext(2):ext(2)];

h = ones(TailleFiltre)/prod(TailleFiltre);
hh(ligs,cols) = h;
hh = ifftshift(hh);

H = fft2(hh);

ind = find(abs(H)<(1/SeuilMax));
H(ind) = (1/SeuilMax)*exp(j*angle(H(ind)));

G = ones(size(H))./H;

(...)
```

▶ **Question 20** *Complete this program (only 2 lines!) to process inverse filtering method.*

▶ **Question 21** *What's happen with the image* `marcheur.jpg` *?*