Master MOSiG - M1

Signal Processing

Signal Processing Applied to 2D Images

Computer Exercises N. 4

Authors

Ky Nguyen

Raquel Oliveira

Grenoble, 08 December 2010

Index

1 Aliasing and Moiré Patterns

Question 1.1.

Complete the corresponding Java code (see paper).

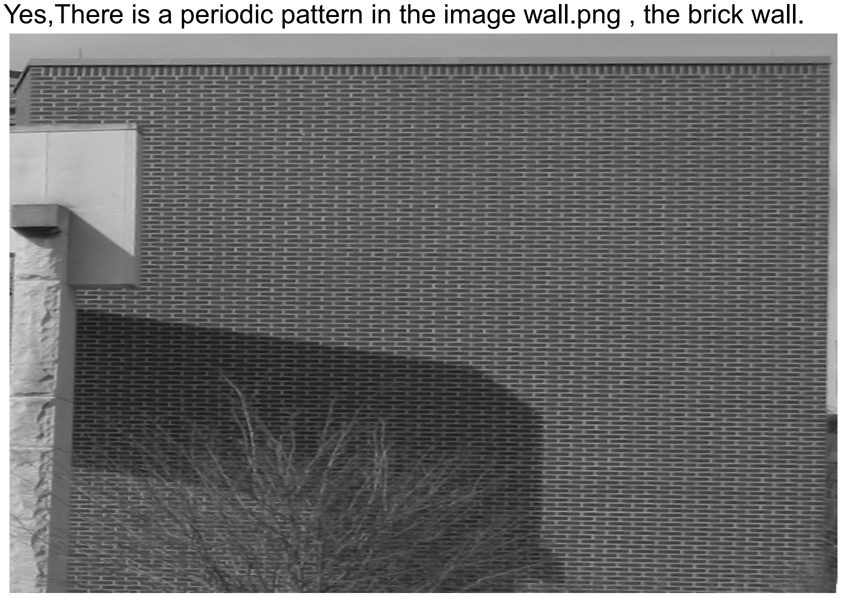
```
public Image subSamples(int factor) {
        int newWidth = this.width / factor;
        int newHeight = this.height / factor;
        Image newImg = new Image(newWidth, newHeight);

        // Please write your code here...
        for (int j = 0; j < newHeight; j++)
            for (int i = 0; i < newWidth; i++) {
                if ((j * factor < this.height) && (i * factor < this.width))
                    newImg.image[j][i] = this.image[j * factor][i * factor];
                else
                    newImg.image[j][i] = (char) 0;
            }
        return newImg;
    }
```

Question 1.2.

In the image wall.pgm, do you see a periodic pattern? Which one?

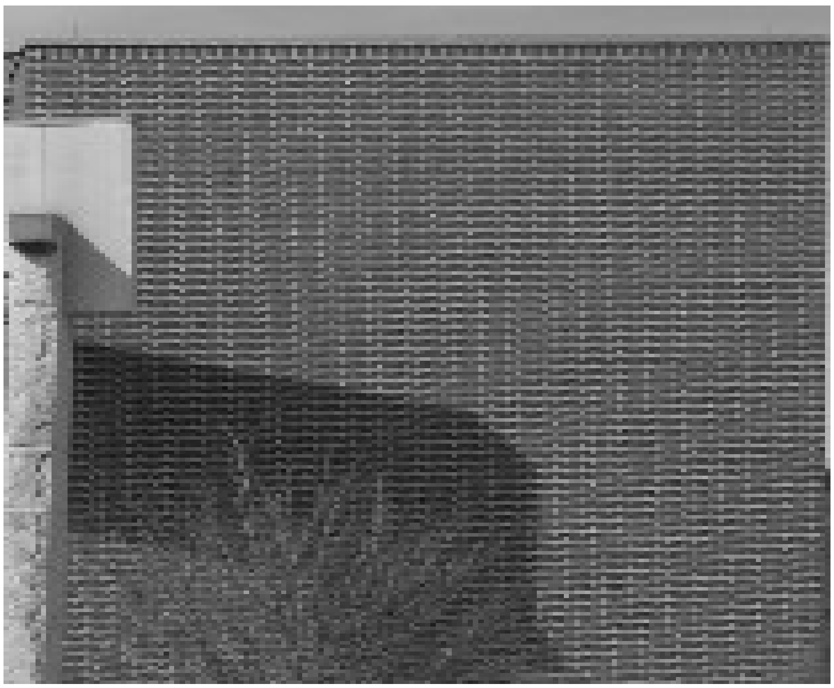
Yes,There is a periodic pattern in the image wall.png , the brick wall.



Question 1.3.

Subsample the image wall.pgm with a factor 4. What do you see in the subsampled image ? Why ?

From the picture of wall.png with a factor 4, we can see that the structure of periodic pattern of brick wall is corrupted because of phenomenon aliasing or aliased frequencies. When the function is sampled at a rate lesser than twice its highest frequencies, the phenomenon aliasing will happen, the sampling rate in image is the number of samples is taken per unit distance. In this case, we divided the number of samples with a factor 4, therefore number of samples that is taken with this factor is lesser than twice samples of original samples, hence this lead a aliased frequencies.



2 Contrast Stretching

Question 2.1.

Complete the method getAsSignal() of the class Image.

```
public Signal getAsSignal() {
      Signal signal = new Signal();
      signal.settName("Image");

      // Please write your code here
      int k = 0;
      for(int j=0; j<this.height;j++)
         for(int i=0;i<this.width;i++)
         {
               signal.addElement(k,this.image[j][i]);
               k++;
         }
      return signal;
   }
```

Question 2.2.

Complete the constructor public Image(Signal signal, int width, int height) of the class Image.

```
public Image(Signal signal, int width, int height) {
    this.width = width;
    this.height = height;
    this.image = new char[height][width];

    // Please writeyour code here
    if(signal.getNbSamples() != width*height)
            System.out.println("The number of samples("+signal.getNbSamples()+") is different of [width("+
width+")*height("+height+")] of the image.");

    int h=0,w=0;
    char ch_var;
    for(int i=0;i<signal.getNbSamples();i++){
        if (signal.getValueOfIndex(i)<0)
            ch_var=(char) 0;
        else if(signal.getValueOfIndex(i)>255)
            ch_var=(char) 255;
        else
            ch_var=(char) Math.round(signal.getValueOfIndex(i));

        this.image[h][w]=ch_var;

        w++;
        if (w==width){
            w=0;
            h++;
        }
    }
}
```

Question 2.3.

Explain equation 1 (see paper).

This equation is a min-max normalization function. Min-max normalization subtracts the minimum value of an attribute from each value of the attribute and then divides the difference by the range of the attribute. These new values are multiplied by the new range of the attribute and finally added to the new minimum value of the attribute. These operations transform the data into a new range.

In equation 1, we have an input signal x[n] with the range of gray level going from actualMin to actualMax, where actualMin is lowest pixel value and actialMax is highest pixel value. After to apply the formula (1), we will have an output signal which is stretched in a new range from lower value (newRangeMin) to upper value (newRangemax). That means if we choose a newRangeMin greater than actualMin, from equation (1) we always get of the output signal a value greater than the value of the input signal, hence in the new range the picture will be much more stretch than the original image. In contrast, if we choose a newRangeMin lesser than the actualMin, then from equation (1) we always will have of the output signal (pixel) a value lesser than the value of original signal (pixel), therefore the output image will be lesser stretch (or lesser contrast) than the original image.

Question 2.4.

Is this system linear ? Why ?

No, it is not. We will explain that with a counter-example.
We know that, if the system is linear, we can apply the scaling property:
H(ax) = aH(x) = ay

Let's take a example:
x[n] = 2
actualMin=1
actualMax=3
newRangeMin=4
newRangeMax=9

$$y[n]=newRangeMin+\frac{(newRangeMax-newRangeMin)}{(actualMax-actualMin)}*(x[n]-actualMin)$$

$$y[n]=4+\frac{(9-4)}{(3-1)}*(2-1) \quad = \quad 4+2{,}5 \quad = \quad 6{,}5$$

Multiplying the system for a constant a=3:

$$3*y[n]=3*6{,}5 \quad = \quad 19{,}5$$

Multiplying the function for a constant a=3 and applying in the system:

$$y[3*n]=4+\frac{(9-4)}{(3-1)}*(3*2-1) \quad = \quad 4+2{,}5(5) \quad = \quad 16{,}5$$

Since ay[n] ǂ y[an], the system is not linear.

Question 2.5.

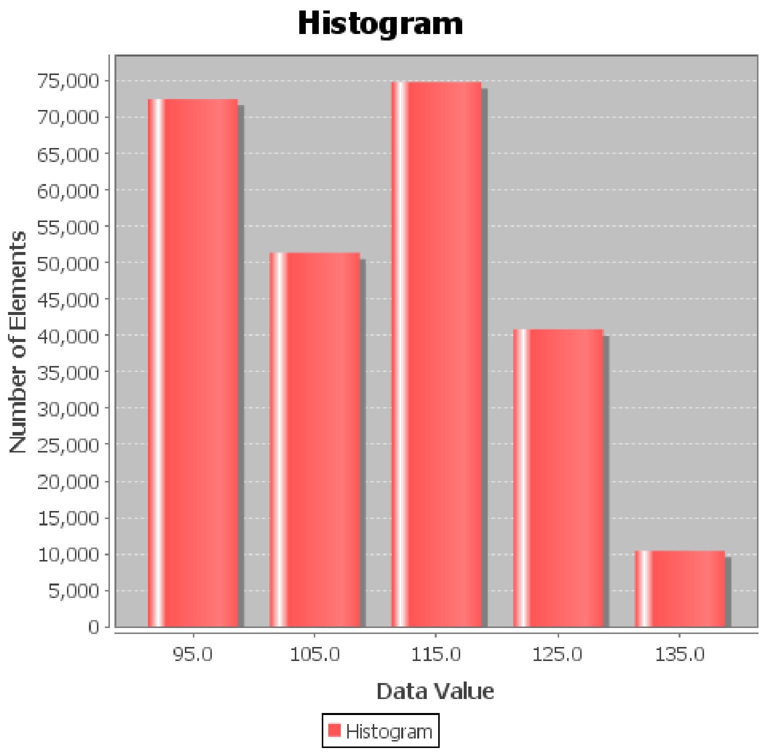
Complete the method public Signal stretchContrast(double newRangeMin, double newRangeMax) of the class Signal.

```
public Signal stretchContrast(double newRangeMin, double newRangeMax) {
   Signal signal = new Signal();

   // Please write your code here
   int N = this.getNbSamples();
   double constant = ( (newRangeMax-newRangeMin)/ (this.getMax()-this.getMin()) );
   double yk;
   for(int i=0; i<N;i++){
        yk = newRangeMin + (constant*(this.getValueOfIndex(i)-this.getMin()) );
        signal.addElement(i, yk);
   }
   return signal;
 }
```

Question 2.6.

Display the histogram of this image. What are the minimum and the maximum values of the pixels? What would be the maximum range of grey levels in an image encoded on char ?



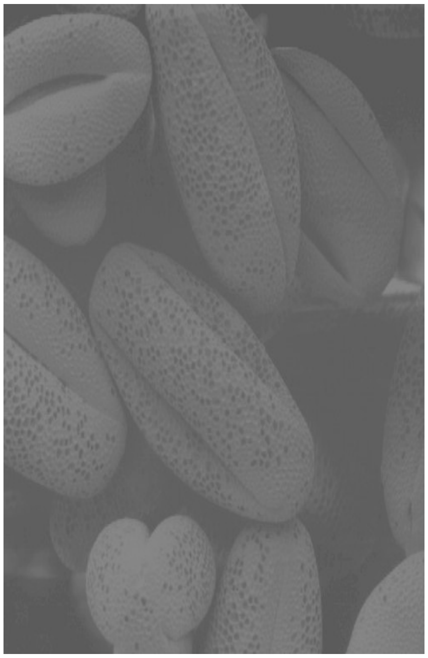
The minimum and the maximum values of the pixels are, respectively, 90 and 138. The maximum range of grey levels in an image encoded on char would be from 0 to 255.

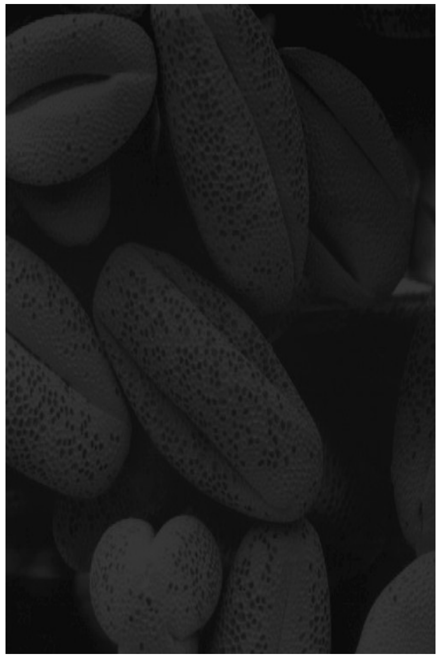
Question 2.7.

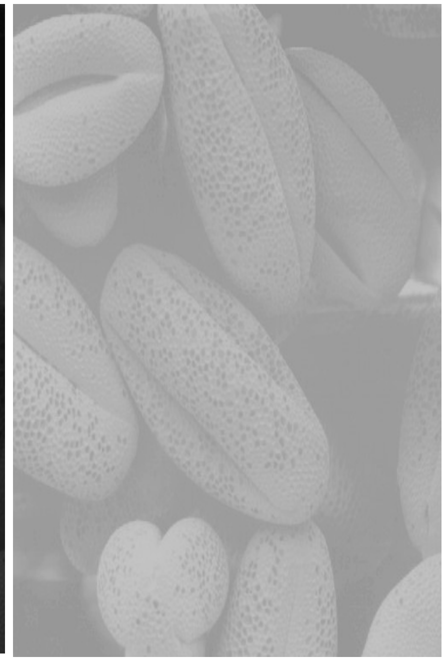
Apply contrast stretching on the image pollen.png. What do you observe? Compare the 2 histograms.

In order to perfectly understand the behavior of the contrast stretching and better answer this question, we performed 3 tests:
First, we applied contrast stretching on the image using the same range of the original image (48).
Second, we applied contrast stretching on the image using a range lesser than the original image (20<48).
Third, we applied contrast stretching on the image using a range greater than the original image (80>48).
For the three cases, we tested with interval both lesser than actualMin and greater than actualMax.

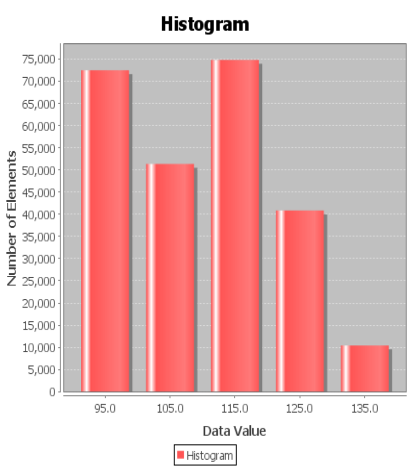
First Test:

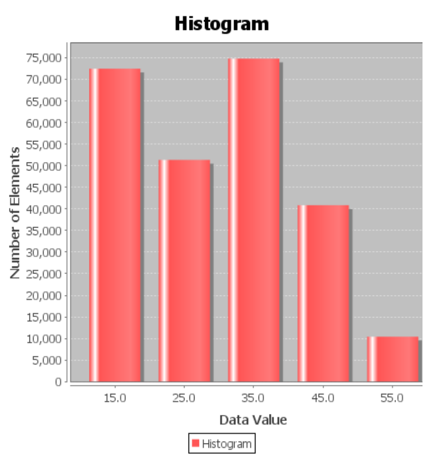


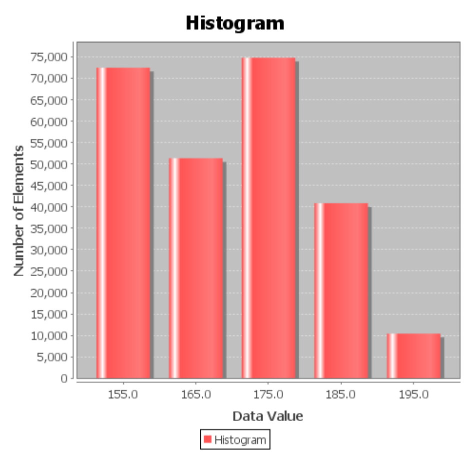




We observed that always that the interval is lesser than actualMin, the result image becomes darker than the original one. On the other hand, always that the interval is greater than actualMax, the result image becomes clearer than the original one.

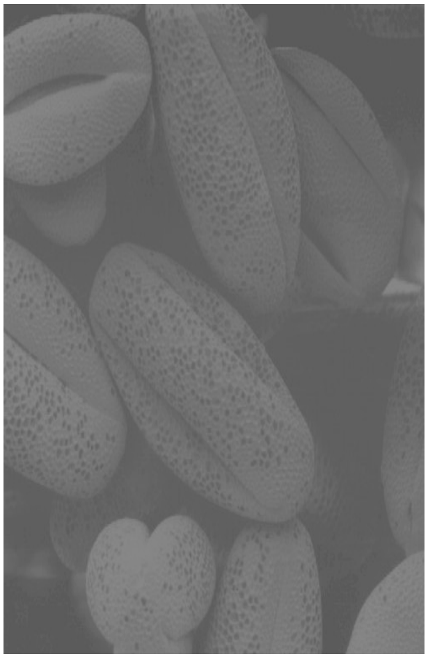




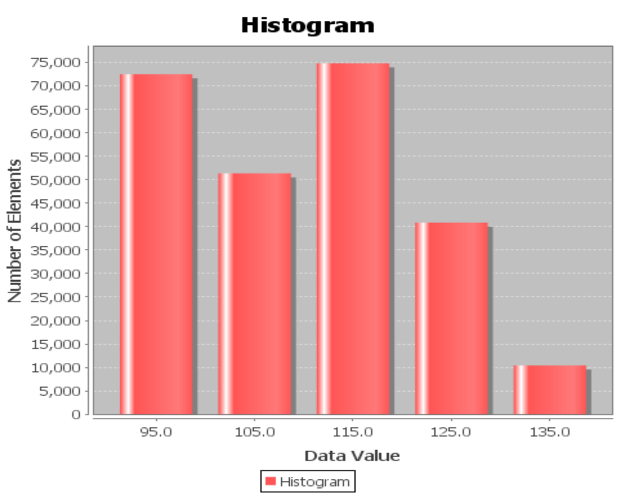


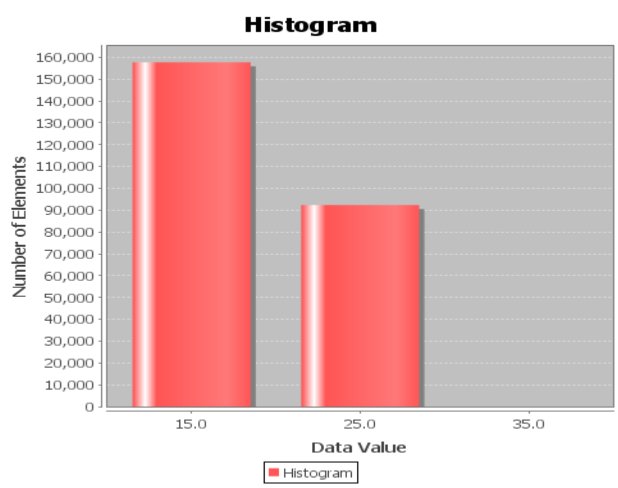
The histograms are equal to the original's image in terms of axis y and number of columns, varing just in axis x, following the variance of the newRangeMin and newRangeMax.

Second test:



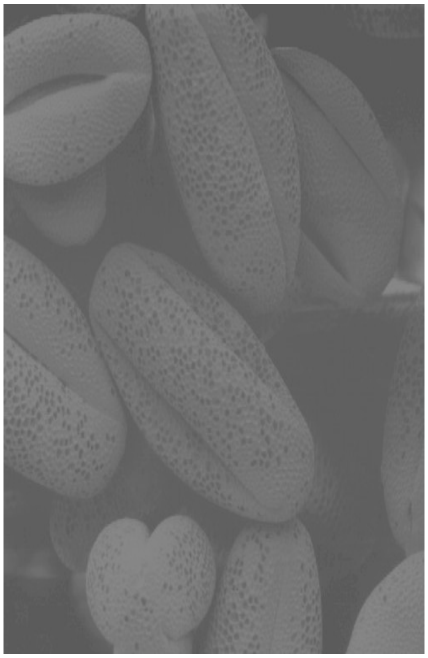
When we decrease the range, the level of the contrast decreases.

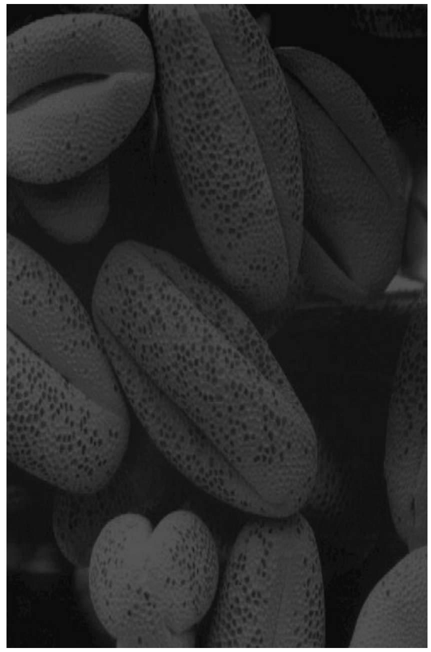


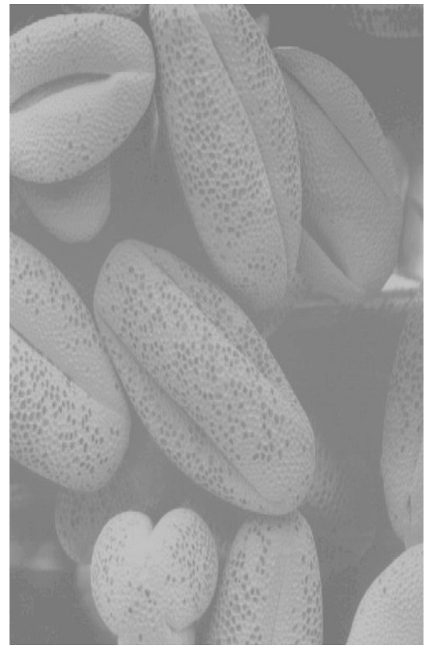


The histogram has fewer columns than the original image, because, with less contrast, there are lesser grey levels than in the original image. The maximum value of the axis y increases, because since the image have less levels of grey, most of the pixels has the same interval of level, so, they appears at the same column at the histogram.

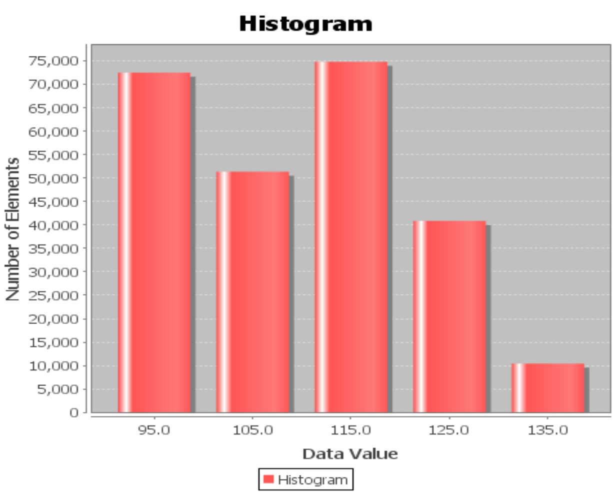
Third test:

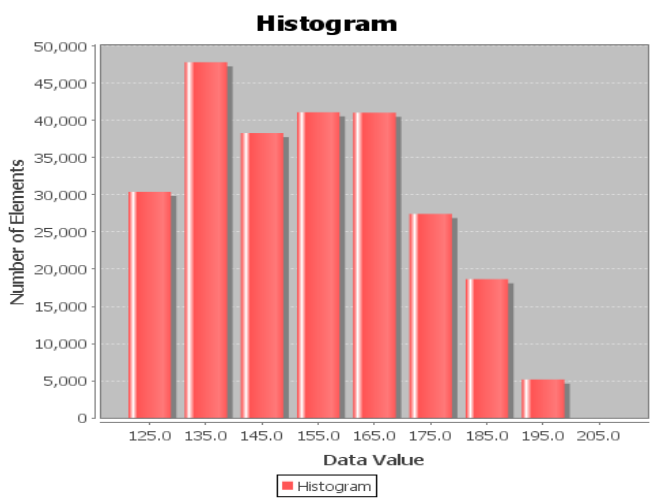






When we increase the range, the level of the contrast becomes wider.

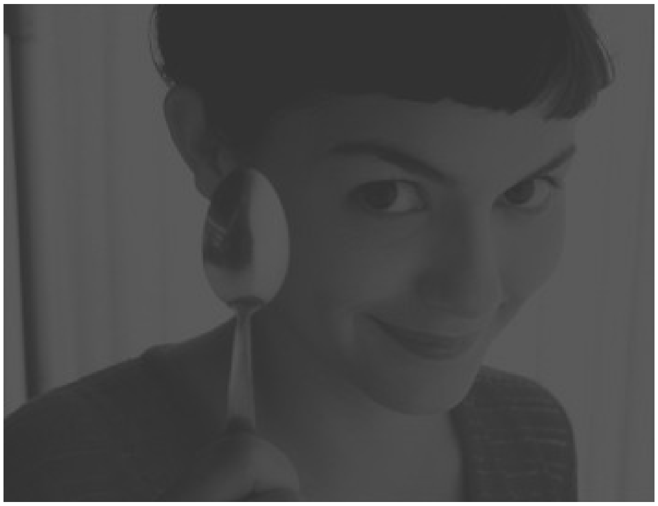




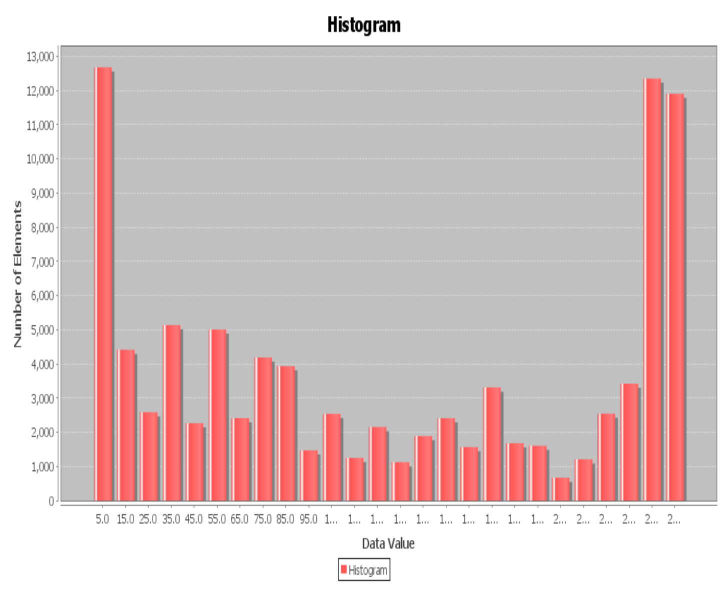
The histogram has more columns than the original image, because, with more contrast, there are more grey levels than in the original image.

Question 2.8.

Same questions for the image amelie.png.



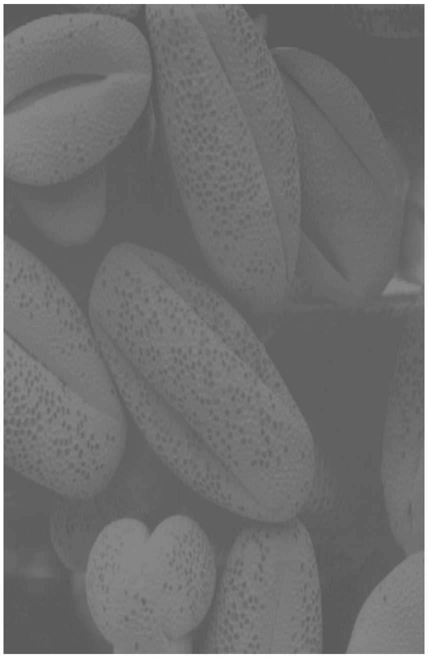
Briefly, the constrast of the image is influencied by the range of the grey levels of the input signal, increasing or decreasing if we increase or decrease the range.
The brightness of the image is influencied by the shift operation at the range. If we apply a shift operation in the range to the right(e.g., greater than the actualMax), the image becomes clearer. On the other hand, If we apply a shift operation in the range to the left(e.g., lesser than the actualMin), the image becomes darker.



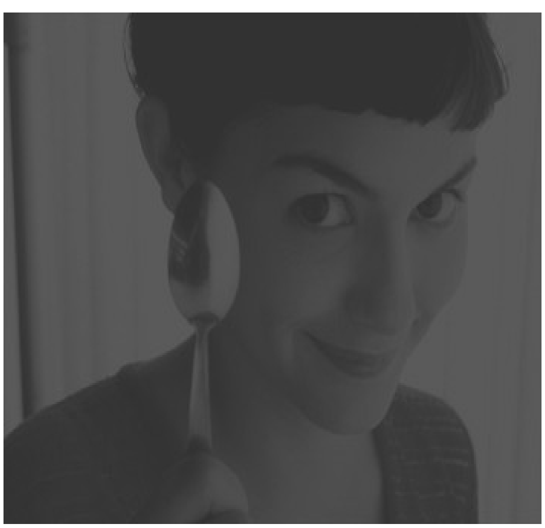
The number of columns of the histogram is influencied by the range of the grey levels of the input signal. If we increasing the range, the number of columns also increases, which means that we have more levels of grey on the image. The maximum value of the axis y decreases, because the image have more diferents levels of grey, and they are spread at the columns. The opposite happens when we decrease the range. The histogram does not change if we apply a shift operation at the range. For example, the histogram of an image with range 20 is the same so if the image has actualMin=30 and actualMax=50 as if the image has actualMin=120 and actualMax=140.

Question 2.9.

What is the missing pattern if you subsample the result image?



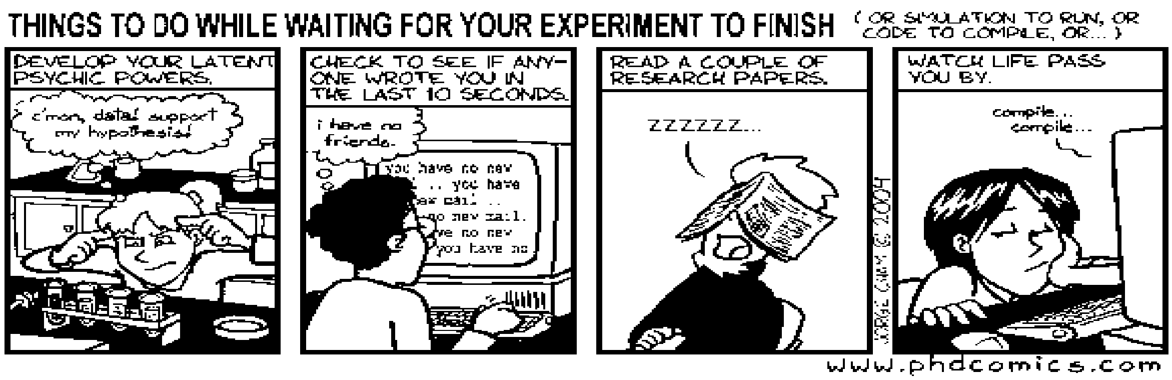
The missing pattern are the points present in the original image of pollen.

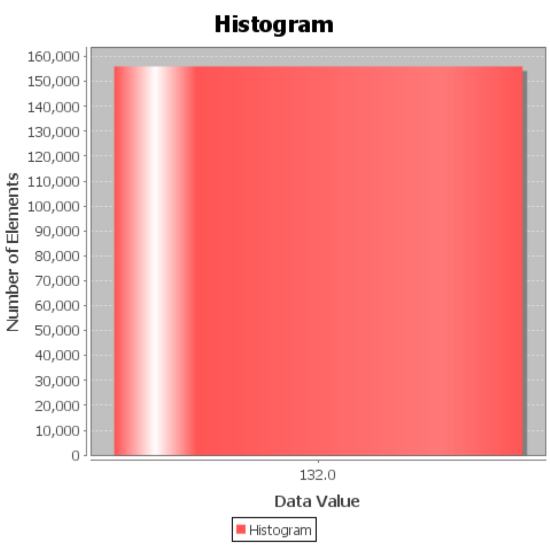


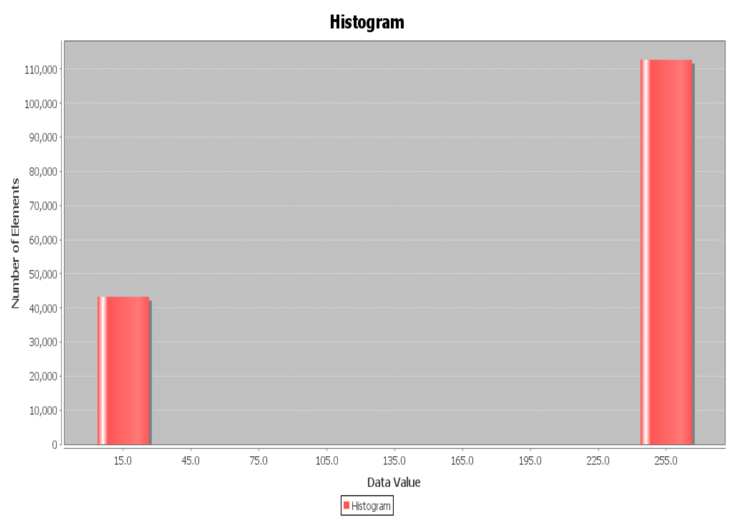
Here, the missing pattern are the stripes clothing amelie.

Question Just For Fun.

What is on the mystery.png image?







As we can see in the result histogram, when we apply the contrast stretching on the image, as a final result we only have pixels with colors white and black, without any other level of grey, making it possible to view the image.

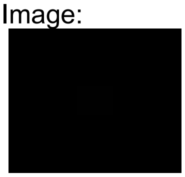
3 Smoothing Spatial Filters

Question 3.1.

Create a 2D impulse of width 5 and store it in a 1D signal file. Use the menu 5 to save it as an image.

File:
#25
0 0
1 0
2 0
3 0
4 0
5 0
6 0
7 0
8 0
9 0
10 0
11 0
12 1
13 0
14 0
15 0
16 0
17 0
18 0
19 0
20 0
21 0
22 0
23 0
24 0

Image:

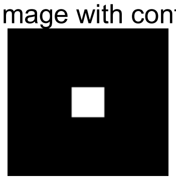


Question 3.2.

Why does this image appear black? Create a new image with contrast stretching to observe the white central pixel.

That image appear black because in the file that caused the image there is only two levels of grey (represented by 0 and 1). Since the scale of gray goes from 0 to 255, 1 is very close to 0, so, it's kind of the same thing seen with the naked eye.

Image with contrast stretching:



Question 3.3.

Create a 2D kernel of a 3 x 3 average filter.

Firstly, we create a file 1-D kernel 3x3 with the below values:
% 3x3 impulse image
#9
0.0 1
1.0 1
2.0 1
3.0 1
4.0 1
5.0 1
6.0 1
7.0 1
8.0 1
Secondly, we use menu 5 to convert a signal 1-D kernel 3x3 to 2-D image. We obtain the following picture:

Question 3.4.

Convolve the image lena with noise.png with a 3 x 3 average filter. What do you observe?

Before convolve the image lena_with_noise.png with the 3x3 average Filter we have the image lena_with_noise looks like below:

After convolve the image lena_with_noise.png with the image 2-D kernel 3x3 average Filter, we have the output image looks like below. Clearly, the output image has no longer the noise, it looks like smoother and brighter than the original image. Generally, the original image is blurred by convolve with a 2-D 3x3 kernel. In other word, convolve the 2-D 3x3 kernel average Filter with the original image reduces the noise in the original image. However, it still have a little bit noise (small white point) in the output image.
The original image does not obtain dark border, but the output image contains a dark border.

Question 3.5.

Convolve the image lena with noise.png with a 5 x 5 average filter. What do you observe? Compare with the previous result.

Firstly, we create the 1-D kernel 5x5 average Filter as below:
% 5x5 impulse image
#25
0.0 1
1.0 1
2.0 1
3.0 1
4.0 1
5.0 1
6.0 1
7.0 1
8.0 1
9.0 1
10.0 1
11.0 1
12.0 1
13.0 1
14.0 1
15.0 1
16.0 1
17.0 1
18.0 1
19.0 1
20.0 1
21.0 1
22.0 1
23.0 1
24.0 1

Secondly, we use the menu 5 to convert 1-D kernel 5x5 average kernel to 2-D kernel 5x5 average Filter. Finally, we convolve the original image lena_with_noise.png with the 2-D kernel 5x5 average Filter, we receive of the output the picture below:

Question 3.6.

What is the equation of a 1-dimensional Gaussian distribution of mean μ = 0 and standard deviation σ?

1-D Gaussian distribution:

$$f(x)=\frac{1}{(\sqrt{2\pi}\sigma)}e^{(\frac{-(x-\mu)^2}{2\sigma^2})}$$

When $\mu=0$:

$$f(x)=\frac{1}{(\sqrt{2\pi}\sigma)}e^{(\frac{-x^2}{2\sigma^2})}$$

Question 3.7.

Explains the numbers in Figure 3 (see paper).

These numbers represent a shape of a 2-D Gaussian distribution within a discrete approximation to the Gaussian function, with mean μ = (0,0) and standard deviation σ=1. Each number is generated by applying the equation of a 2-D Gaussian distribution. The number 273, which the result of the convolution should be divided, is the sum of all the values. The goal of this division is to obtain a unitary gain of each convolved pixel, this is called normalization.

Question 3.8.

Comment what you obtain. Compare the result image with the 2 images obtained in the previous section.

The output image from convolve the original image lena_with_noise.png with the Gaussian kernel give us an image without the noise. It is more blurred and smoother than the original image, it contains a dark border surround the image. However, this quality of the output image is better than 2 images obtained in the previous section. It is not too blurred, and still keep a sharp level in image quality.