1 Matlab correction

1.1 First manipulations

The following function is usefull to display 4 images:

```
function affichePar4(A, B, C, D)
% Display 4 images in the same window
figure();
subplot(2,2,1);
imshow(A);

subplot(2,2,2);
imshow(B);

subplot(2,2,3);
imshow(C);

subplot(2,2,4);
imshow(D);
```

1.1.1 Load and save image

```
I=imread('retine.png');
imagesc(I);
figure(2);
imshow(I);

% data inside image
imfinfo('retine.png');
size(I)
```

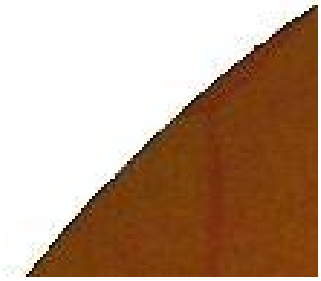
```
Command window

>> imfinfo('retine.png')
ans =
                    Filename: '/home/yann/Documents/Cou...'
                FileModDate: '05-Nov-2013 12:04:24'
                    FileSize: 741963
                      Format: 'png'
                FormatVersion: []
                       Width: 922
                      Height: 911
                    BitDepth: 24
                   ColorType: 'truecolor'
             FormatSignature: [137 80 78 71 13 10 26 10]
                    Colormap: []
                   Histogram: []
               InterlaceType: 'none'
                Transparency: 'none'
    SimpleTransparencyData: []
             BackgroundColor: []
             RenderingIntent: []
              Chromaticities: []
                       Gamma: []
                 XResolution: 2835
                 YResolution: 2835
              ResolutionUnit: 'meter'
                     XOffset: []
                     YOffset: []
                  OffsetUnit: []
             SignificantBits: []
                ImageModTime: []
                       Title: []
                      Author: []
                 Description: []
                   Copyright: []
                CreationTime: []
                    Software: []
                  Disclaimer: []
                     Warning: []
                      Source: []
                     Comment: []
                   OtherText: []
>>
```

1.1.2 JPEG file format

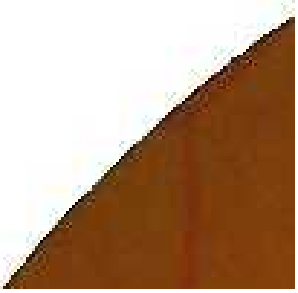
JPEG is a compressed file format that accept loss in quality. The following code illustrates the different quality parameters, shown in Fig.1.

```
% test read/write with loss in quality
imwrite(I, 'retine_lossy_25.jpg', 'jpg', 'Mode', 'lossy', 'Quality', 25);
a1=imread('retine_lossy_25.jpg');

imwrite(I, 'retine_lossy_50.jpg', 'jpg', 'Mode', 'lossy', 'Quality', 50);
a2=imread('retine_lossy_50.jpg');

imwrite(I, 'retine_lossy_75.jpg', 'jpg', 'Mode', 'lossy', 'Quality', 75);
a3=imread('retine_lossy_75.jpg');

imwrite(I, 'retine_lossy_100.jpg', 'jpg', 'Mode', 'lossy', 'Quality',
    ↪ 100);
a4=imread('retine_lossy_100.jpg');

% zoom in particular area
d1=imcrop(a1, [75 68 130 112]);
d2=imcrop(a2, [75 68 130 112]);
d3=imcrop(a3, [75 68 130 112]);
d4=imcrop(a4, [75 68 130 112]);

affichePar4(d1, d1, d3, d4);
```

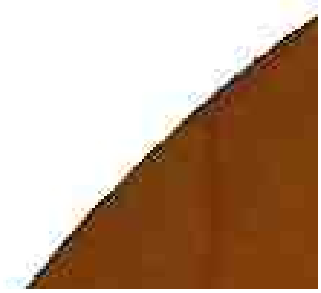
(a) Quality: 100.



(b) Quality: 75.



(c) Quality: 50.



(d) Quality: 25.

Figure 1: Illustration of different quality parameters used in JPEG compression. The original image is zoomed in order to emphasize the quality loss.

1.2 Image histogram

Notice that MATLAB® indices begin at 1!

```
function h=myHist(image)
% histogram function of grayscale image coded in 8 bits
h=zeros(256, 1);

for i=1:size(image, 1)
  for j=1:size(image, 2)
    h(image(i,j)+1) = h(image(i,j)+1) + 1;
  end
end
```

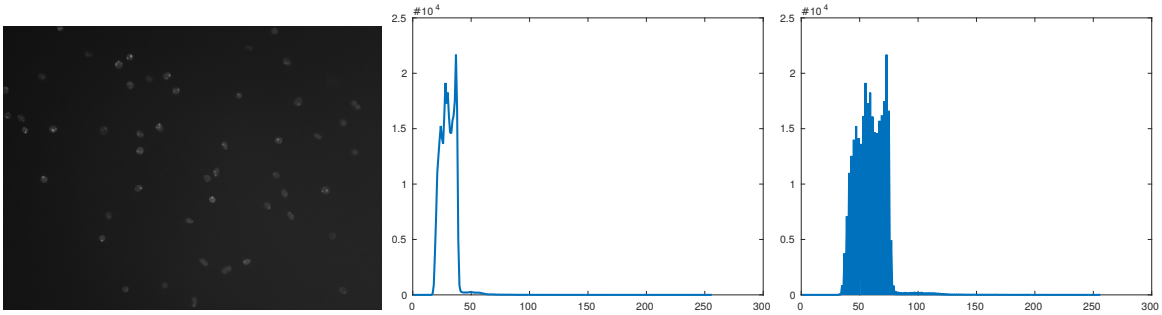
This is the MATLAB® version.

```
muscle=imread('muscle.jpg');
h = imhist(muscle);
figure();plot(h);
```

1.3 Linear mapping of the image intensities

The application of a linear stretching is quite simple. The histograms are illustrated in Fig.2.

```
cornea=imread('cellules_cornee.jpg');
minimum = min(cornee(:));
maximum = max(cornee(:));

a=255/(maximum-minimum);
b=-255*minimum/(maximum-minimum);

cornee2 = a*cornee + b;
figure();
subplot(2,2,1);imshow(cornee); title('cornea');
subplot(2,2,2);imshow(cornee2);title('stretched cornea');

% histograms
subplot(2,2,3);plot(imhist(cornee)); title('histogram of cornea');
subplot(2,2,4);plot(imhist(cornee2));title('stretched histogram');
```



(a) Original image. (b) Histogram of the image cellules_cornee. (c) Stretched histogram.

Figure 2: Result of histogram stretching.

1.4 Color quantization

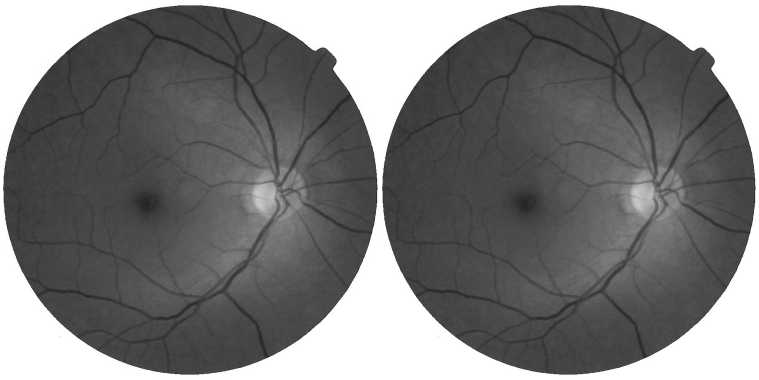
The objective is to reduce the number of colors by 2 (for example). We use the properties of the data types: integer type rounds automatically while dividing. In the proposed example, the gray value is taken as the green channel of a color retina image (Fig.3).

```
image_gris = I(:,:,2); % green channel
q4=image_gris/4*4;
q16=image_gris/16*16;
q32=image_gris/32*32;
affichePar4(image_gris, q4, q16, q32);
```

1.5 Aliasing (Moiré) effect

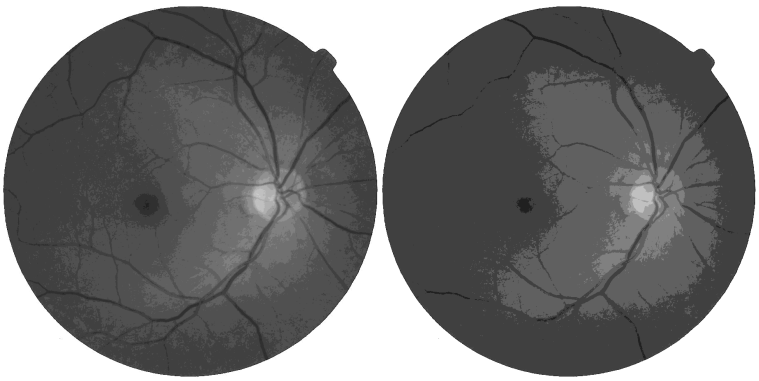
The aliasing effect occurs when two sampling are performed. This is illustrated in Fig.4 with the following code.

```
function C=cercle(fs,f)
% Generates an image with aliasing effect
% fs: sample frequency
% f : signal frequency

% time sampling
t=0:1/fs:1;

C=zeros(size(t,2));
for i=1:size(t,2);
    for j=1:size(t,2);
```



(a) Original retina image, green channel.

(b) 64 grey levels.

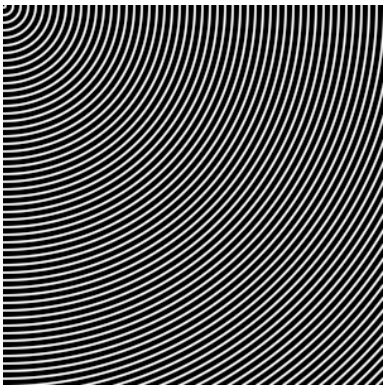


(c) 16 grey levels.

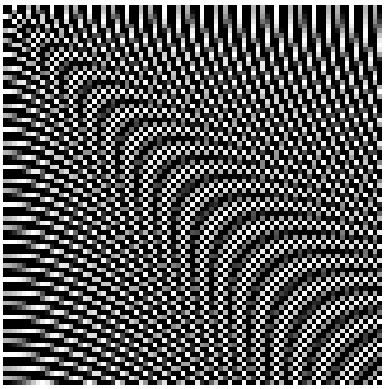
(d) 8 grey levels.

Figure 3: Illustration of different quality parameters used in JPEG compression.

```
            C(i,j)=sin(2*pi*f*sqrt(t(i)^2+t(j)^2));
        end
    end
```



(a) $f_s = 300, f = 50$.



(b) $f_s = 80, f = 50$.

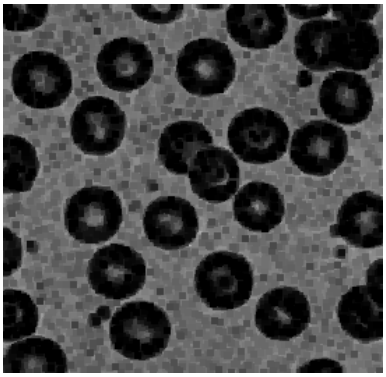
Figure 4: Illustration of the aliasing effect.

1.6 Low-pass filtering

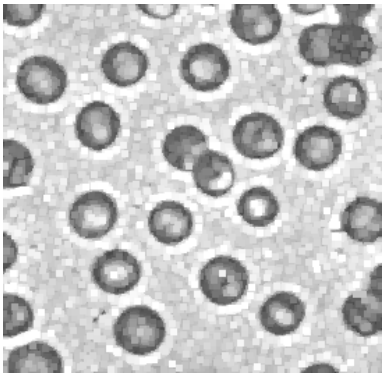
The mean and Gaussian filters are linear filters. Other filters are called rank filters. See Fig.5 for an illustration.

```
% read image and convert to double for convolution computation
A=imread('bloodCells.bmp');
A=double(A)/255;

% display images
figure;
subplot(231);imshow(A);title('Original');

Amin=ordfilt2(A,1,ones(5,5),'symmetric');
subplot(232);imshow(Amin);title('Low-pass filter : min');

Amax=ordfilt2(A,25,ones(5,5),'symmetric');
subplot(233);imshow(Amax);title('Low-pass filter : max');

Amoyen=imfilter(A,1/25*ones(5,5),'symmetric');
subplot(234);imshow(Amoyen);title('Low-pass filter : moyen');

Amedian=ordfilt2(A,13,ones(5,5),'symmetric');
subplot(235);imshow(Amedian);title('Low-pass filter : median');

hgauss=fspecial('gaussian',[5 5],1);
```

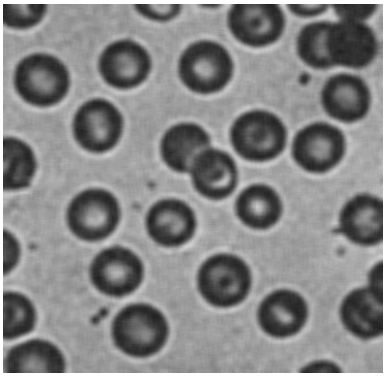
```
Agauss=imfilter(A,hgauss);
subplot(236);imshow(Agauss);title('Low-pass filter : gaussien');
```



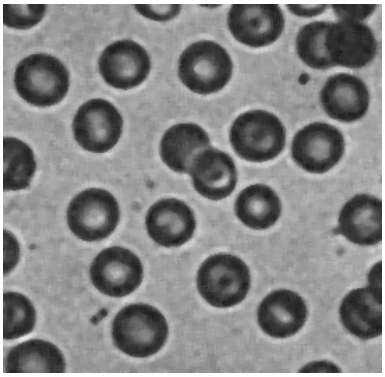
(a) Minimum filter.



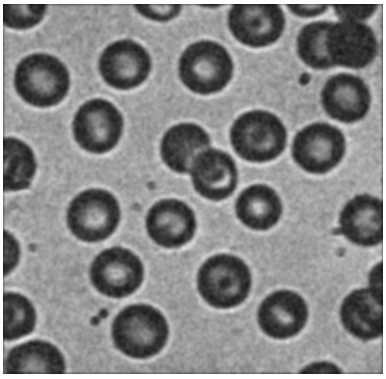
(b) Maximum filter.



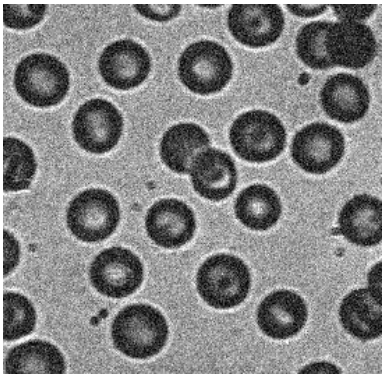
(c) Mean filter.



(d) Median filter.



(e) Gaussian filter, $\sigma = 1$.



(f) Original image.

Figure 5: Low-pass filters computed in a 5×5 neighborhood.

1.7 High pass filters

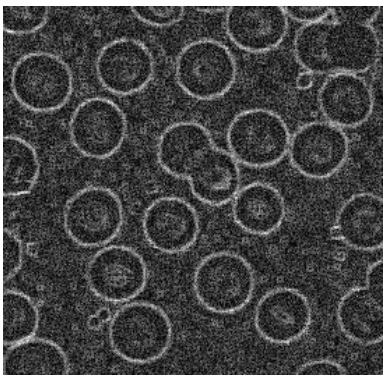
These high-pass filters are simple the difference (residu) between the original image and a low-pass filter (Fig.6).

```
figure;
subplot(231);imshow(A);title('Original');

AminPH=A-Amin;
subplot(232);imshow(AminPH);title('High-pass : min');

AmaxPH=Amax-A;
subplot(233);imshow(AmaxPH);title('High-pass : max');

AmoyenPH=A-Amoyen;
subplot(234);imshow(AmoyenPH);title('High-pass : moyen');

AmedianPH=A-Amedian;
subplot(235);imshow(AmedianPH);title('High-pass : median');

AgaussPH=A-Agauss;
subplot(236);imshow(AgaussPH);title('High-pass : gaussien');
```

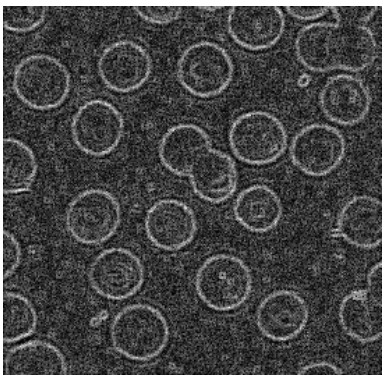
1.7.1 Laplacian filter

The Laplacian filter is based on the second derivative (Fig.7).

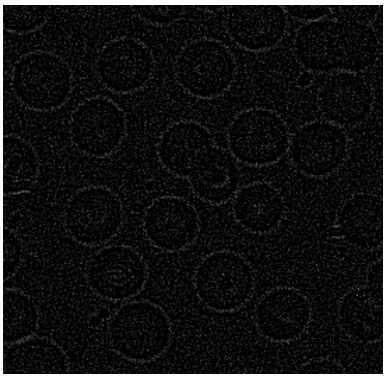
```
B=imread('osteoblaste.bmp');
B=double(B);
B=B/255;
hlaplacien=[-1 -1 -1; -1 8 -1;-1 -1 -1];
Blaplacien=imfilter(B,hlaplacien);
Alaplacien=imfilter(A,hlaplacien);
figure;
subplot(221);imshow(A);title('original image');
subplot(222);imshow(Alaplacien);title('Laplacian filter');
subplot(223);imshow(B);title('originale image');
subplot(224);imshow(Blaplacien);title('Laplacian filter');
```



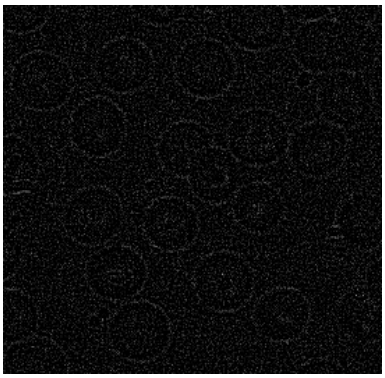
(a) Residu from minimum filter.



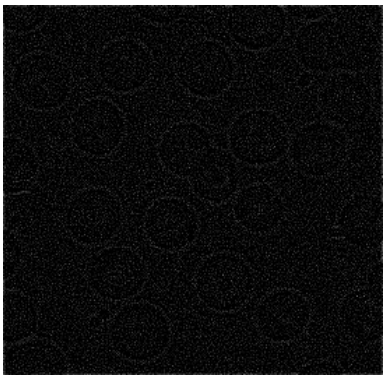
(b) Residu from maximum filter.



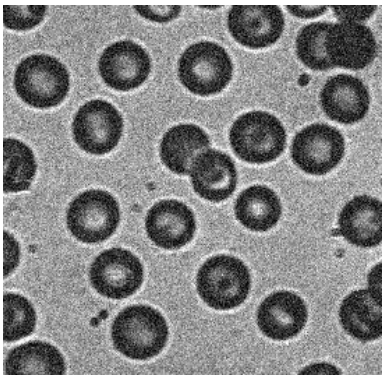
(c) Residu from mean filter.



(d) Residu from median filter.

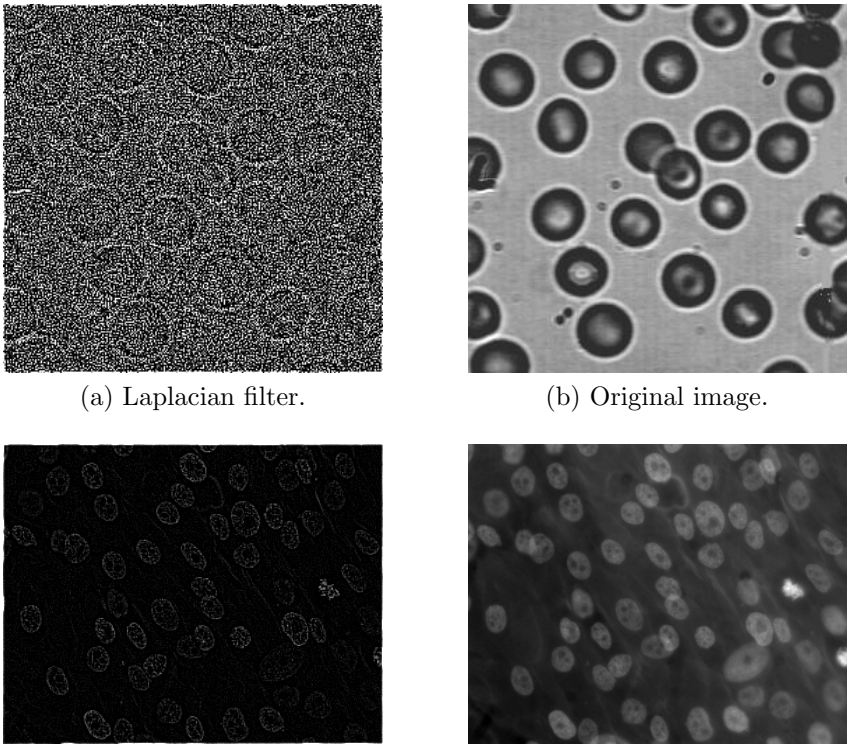


(e) Residu from Gaussian filter, $\sigma = 1$.



(f) Original image.

Figure 6: High-pass filters computed in a 5×5 neighborhood.



(a) Laplacian filter. (b) Original image.

(c) Laplacian filter. (d) Original image.

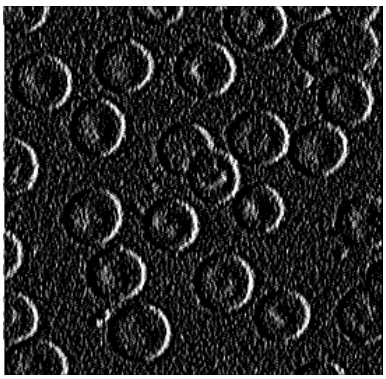
Figure 7: Laplacian filters.

1.8 Derivative filters

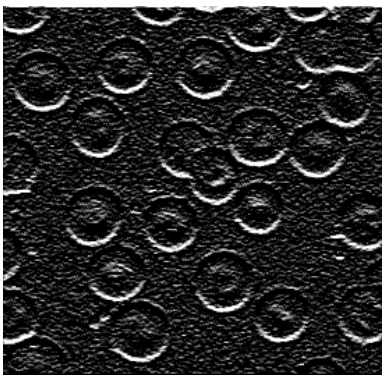
1.8.1 Derivation: Prewitt gradient

A gradient is a vector of the first derivatives. The norm of this vector represents the intensity of the contours (Fig.8 for the Prewitt gradient, and 9 for the Sobel gradient). A derivative filter is very sensitive to noise.

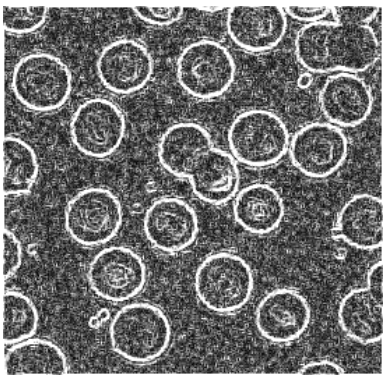
```
hprewittx=[-1 0 1;-1 0 1;-1 0 1];
hprewitty=hprewittx';
Aprewittx=imfilter(A,hprewittx);
Aprewitty=imfilter(A,hprewitty);
Aprewittxy=(Aprewittx.^2+Aprewitty.^2).^(0.5);
subplot(221);imshow(A);title('Original');
subplot(222);imshow(Aprewittxy);title('Prewitt : x and y');
subplot(223);imshow(Aprewittx);title('Prewitt : x');
subplot(224);imshow(Aprewitty);title('Prewitt : y');
```



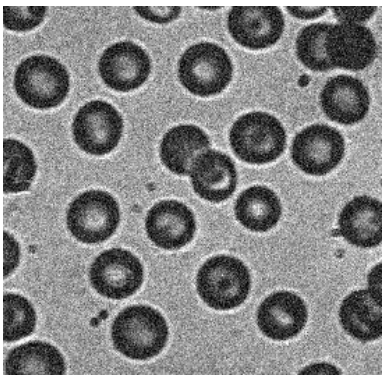
(a) Horizontal direction.



(b) Vertical direction.



(c) Norm of the gradient.

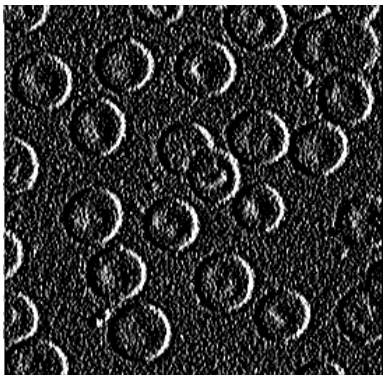


(d) Original image.

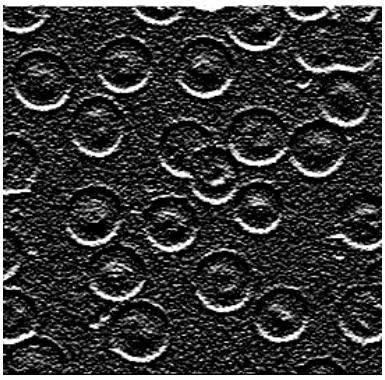
Figure 8: Prewitt derivative filter.

1.8.2 Derivation: Sobel gradient

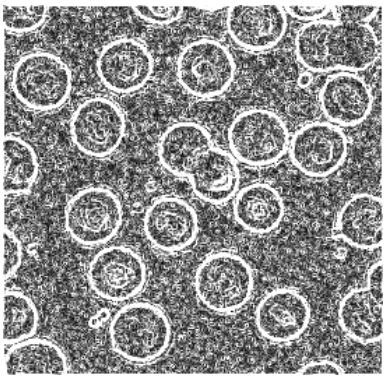
```
figure
hsobelx=[-1 0 1;-2 0 2;-1 0 1];
hsobely=hsobelx';
Asobelx=imfilter(A,hsobelx);
Asobely=imfilter(A,hsobely);
Asobelxy=(Asobelx.^2+Asobely.^2).^(0.5);
subplot(221);imshow(A);title('Original');
subplot(222);imshow(Asobelxy);title('Sobel : x and y');
subplot(223);imshow(Asobelx);title('Sobel : x');
subplot(224);imshow(Asobely);title('Sobel : y');
```



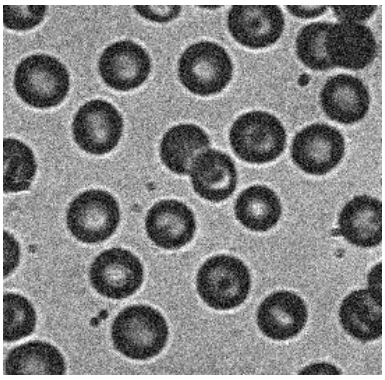
(a) Horizontal direction.



(b) Vertical direction.



(c) Norm of the gradient.



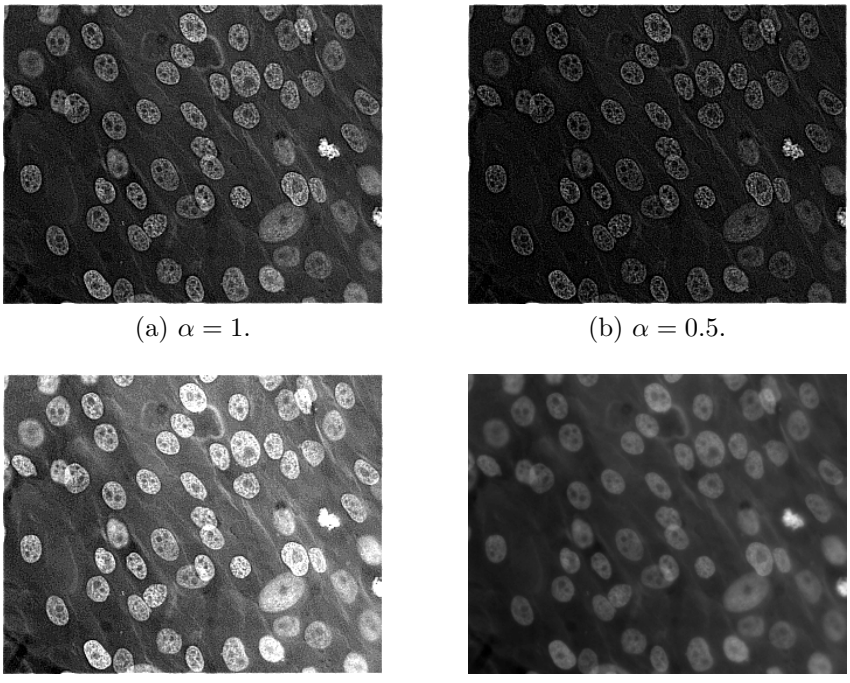
(d) Original image.

Figure 9: Sobel derivative filter.

1.9 Enhancement filters

This quite simple method is used in photo manipulation softwares in order to artificially increase the focus of an image. The human visual perception positively responds to this type of filter.

```
B=imread('osteoblaste.bmp');
B=double(B);
B=B/255;
hlaplacien=[-1 -1 -1; -1 8 -1;-1 -1 -1];
Blaplacien=imfilter(B,hlaplacien);
Benhance1=B+Blaplacien;
Benhance2=0.5*B+Blaplacien;
Benhance3=2*B+Blaplacien;
figure
subplot(221);imshow(B);title('Original');
subplot(222);imshow(Benhance1);title('Enhancement : 1');
subplot(223);imshow(Benhance2);title('Enhancement : 0.5');
subplot(224);imshow(Benhance3);title('Enhancement : 2');
```



(a) $\alpha = 1$. (b) $\alpha = 0.5$.

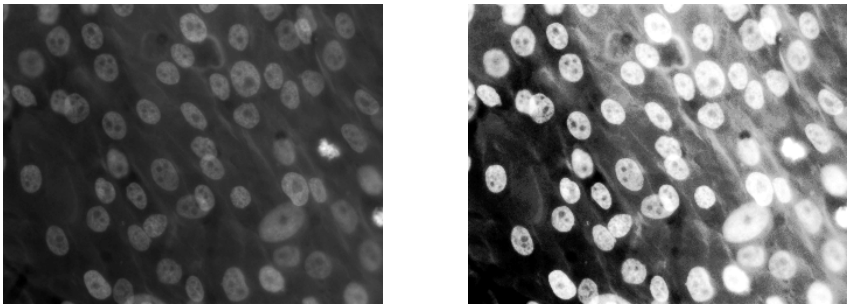
(c) $\alpha = 2$. (d) Original image.

Figure 10: Image enhancement: $I = \alpha \cdot I + HP(I)$, where HP is a high-pass filter (the Laplacian filter in these illustrations).

1.10 Open question

The histogram equalization is a method that will further developped. The result is presented in Fig.11.

```
figure
Bhisteq=histeq(B);
subplot(131);imshow(B);title('original image');
subplot(132);imshow(Benhance3);title('enhancement by laplacian');
subplot(133);imshow(Bhisteq);title('histogram equalization enhancement);
```



(a) Original image. (b) Histogram equalization.

Figure 11: Image enhancement by histogram equalization. When extreme intensity values are present in a image (white or black values), histogram stretching is useless. The histogram equalization can thus be a solution in order to enhance the image.