1 Matlab correction

1.1 Fourier transform

In order to display the Fourier transform, the following function is used to show both spectrum and phase. Notice the logarithmic function.

```
function viewImageSpectrum(I,F)
% I: original image
% F: Fourier transform of I

subplot(1,3,1);
imshow(I,[]);

% phase
subplot(1,3,3);
Im=angle(S);
imshow(Im,[]);

% amplitudes
subplot(1,3,2);
Ia=abs(S);
Ia2=log(1+Ia);
imshow(Ia2,[]);
```

The fftshift functions centers the frequency $(0,0)$ in the image. The following functions will be used:

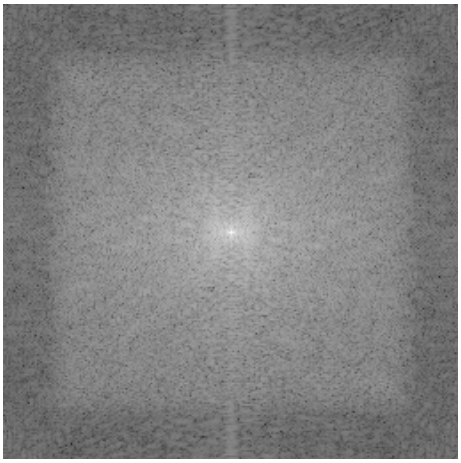
```
% Fourier transform utility of image I
function S=FT(I)
S=fftshift(fft2(double(I)));
```

```
% Inverse Fourier transform utility of spectrum S
function I=iFT(S)
I=real(ifft2(fftshift(S)));
```

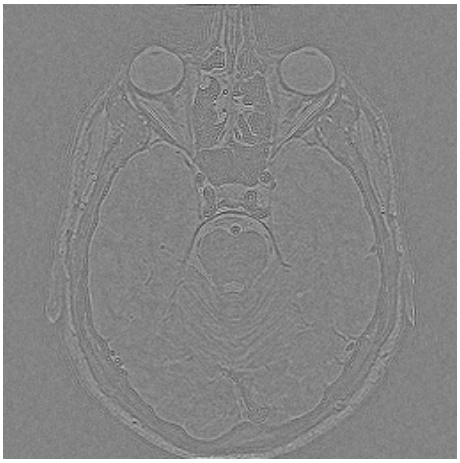
1.2 Inverse Fourier Transform

The important thing to notice in the 2D Fourier transform is that the information is present in the phase, not in the amplitude. The following code highlights this property (see also Fig.1).

```
Spectrum=FT(A);

% amplitude and phase
amplitude = abs(Spectrum);
phase = angle(Spectre);

% reconstruction with amplitude only
C=real(ifft2(fftshift(amplitude)));
figure;imshow(C,[]);title('Reconstruction from amplitude only');

% reconstruction with phase only
D=real(ifft2(fftshift((exp(1i*phase)))));
figure;imshow(D,[]);title('Reconstruction from phase only');
```



(a) Amplitude only.



(b) Phase only.

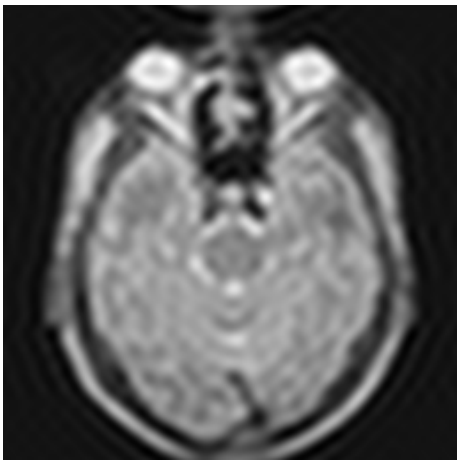
Figure 1: Reconstruction of partial informations (phase or amplitude only). Notice that the main visual informations are contained in the phase, and not in the amplitude.

1.3 Low-pass and high-pass filtering

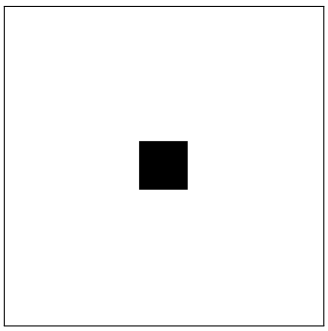
This is done by selecting only low or high frequencies, respectively. A binary window is employed.

```
function Sr=LPfilter(S, fC)
% low pass filtering
% S: spectrum (Fourier Transform)
% fC: cut-off frequency
fC=floor(fC);
if(fC<=0 | 2*fC >= size(S,2) | 2*fC >= size(S,1) )
    disp('Wrong cut-off frequency');
```

(a) High Pass filter.



(b) Low Pass filter.



(c) Filtering (binary) mask.

Figure 2: Fourier basic filtering of the brain image.

```
    Sr=0;
    return;
end

So=zeros(size(S));
So((size(S,1)/2-fC):(size(S,1)/2+fC), (size(S,2)/2-fC):(size(S,2)/2+fC))
    ↪ =1;
Sr=S.*So;
```

```
function Sr=FiltrePH(S, fC)
% High pass filter
% S: spectrum (fourier transform)
% fC: cut-off frequency

fC=floor(fC);
if(fC<=0 | 2*fC >= size(S,2) | 2*fC >= size(S,1) )
    disp('Taille de coupure incorrecte');
    Sr=0;
    return;
end

So=ones(size(S));
So((size(S,1)/2-fC):(size(S,1)/2+fC), (size(S,2)/2-fC):(size(S,2)/2+fC))
    ↪ =0;
Sr=S.*So;
```

The two previous functions are used to filter the image, which is done with:

```
Spectre_PB0=FiltrePB(Spectrum,20);
Spectre_PH0=FiltrePH(Spectrum,110);
% inverse Fourier transform
A_PB0=iFT(Spectre_PB0);
A_PH0=iFT(Spectre_PH0);
% display results
viewImageSpectre(A_PB0,Spectre_PB0);title('Low-pass filter');
viewImageSpectre(A_PH0,Spectre_PH0);title('High-pass filter');
```

1.4 Application: evaluation of corneal cell density

The principle consists in isolating the annulus (see Fig.3) that corresponds to a frequency of repetition of the cells, that can lead us to a cell density. First of all, the image is loaded and the FFT is applied.

```
A=imread('cornee.tif');
% spectrum
Spectrum=FT(A);
% amplitude and phase computation
amplitude = abs(Spectrum);
phase = angle(Spectrum);
```

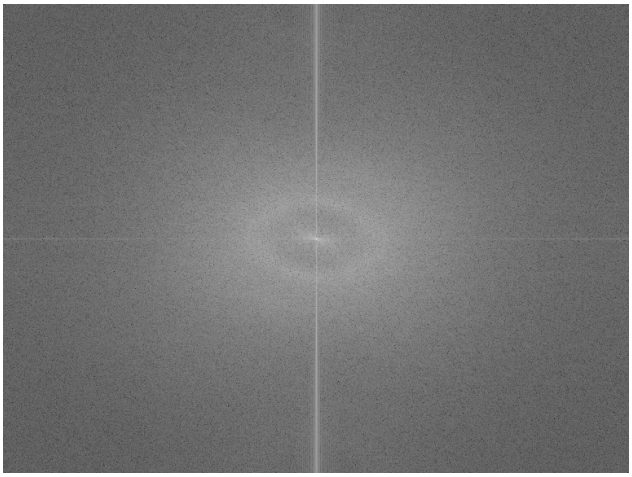


Figure 3: Amplitude of the spectrum of the cornea image. Notice the circular shape that denotes a certain regularity of the cellular pattern, and can be used in order to evaluate the cell density.

Then, the amplitude is filtered. A central line is kept. The objective is now to extract the second peak. The results of the findpeaks function are displayed in Fig.4.

```
% Filtering of
PSF = fspecial('gaussian',30,30);
Blurred = imfilter(amplitude,PSF,'symmetric','conv');

V = Blurred(:,end/2);
plot(V);

% In order to find the peaks, the method is elementary:
% we are looking for the 2nd peak, at length(V)/2
[pks, locs] = findpeaks(V);
hold on
plot(locs, pks, 'sr');
locs2 = sort(abs(length(V)/2-locs))

% result is displayed
disp('frequency of repetition:')
f=locs2(2)/length(V)
```

```
disp('cornea diameter:')
d=1/f
```

Command window

```
locs2 =
     1
    49
    50
   200
   214
   224
   225
   234
   274
   277
   281
   281
   284
   285
   286

frequency of repetition:
f =
    0.0851

cornea diameter:
d =
   11.7551
>>
```

More informations can be see in [2, 1, 3, 4].

References

[1] E. Grisan, A. Paviotti, N. Laurenti, and A. Ruggeri. A lattice estimation approach for the automatic evaluation of corneal endothelium density. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 1700–1703, 2005.

[2] A. Ruggeri, E. Grisan, and J. Jaroszewski. A new system for the automatic estimation of endothelial cell density in donor corneas. *Br J Ophthalmol*, 89(3):306–311, 2005.

[3] Alfredo Ruggeri, Enrico Grisan, and Jan Schroeter. Evaluation of repeatability for the automatic estimation of endothelial cell density in donor corneas. *Br J Ophthalmol*, 0, 2007.

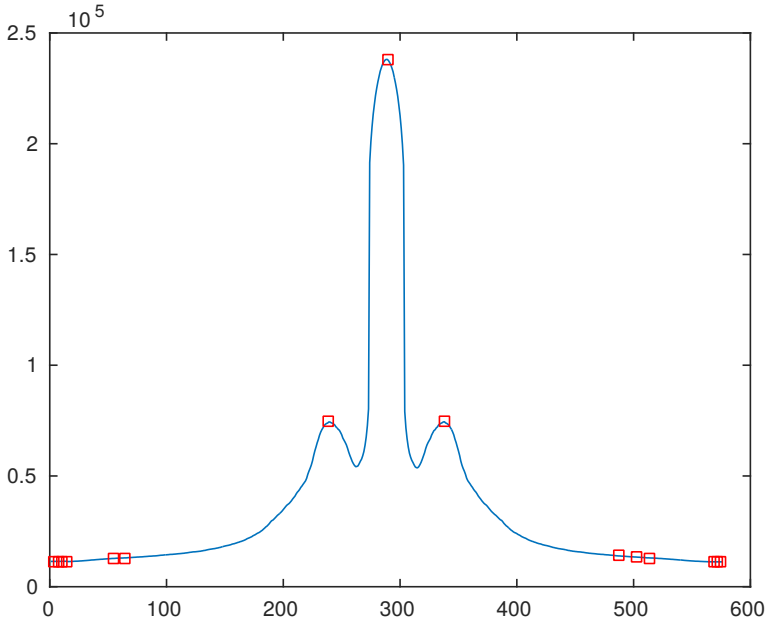


Figure 4: Detected peaks.

[4] B. Selig, K. A. Vermeer, B. Rieger, T. Hillenaar, and C. L. Luengo Hendriks. Fully automatic evaluation of the corneal endothelium from in vivo confocal microscopy. *Biomed central*, 2015.