Correction: Fourier Transform and cornea cell density evaluation

1 Python correction

```python
import numpy as np
from scipy import misc, ndimage #read/write images
import matplotlib.pyplot as plt # plots
```

1.1 Introduction and utilities

To display a spectrum, the following function can be useful:

```python
# Displays spectrum and phase in an image (grayscale)
def viewSpectrumPhase(amplitude, phase):
    plt.figure()
    plt.subplot(1,2,1)
    plt.imshow(np.log(1+amplitude), plt.cm.gray);

    plt.subplot(1,2,2)
    mmax = np.max(phase);
    mmin = np.min(phase);
    if (mmax == mmin):
        B=0;
    else:
        B = 255*(phase-mmin)/(mmax-mmin);

    plt.imshow(B, cmap=plt.cm.gray);
```

1.2 Fourier transform

Results (amplitude and phase) are represented in Fig. 3.

```python
cornea = misc.imread('cornee.png')
print type(cornea)
print cornea.shape, cornea.dtype

plt.subplot(131)
plt.imshow(cornea, cmap=plt.cm.gray)
```

```python
## Fourier transform
#  result is complex
#  fftshift is used by convention the get frequency 0 at center of image
spectre = np.fft.fftshift(np.fft.fft2(cornea));

A = abs(spectre);
G = np.angle(spectre);

viewSpectrumPhase(A, G);
```

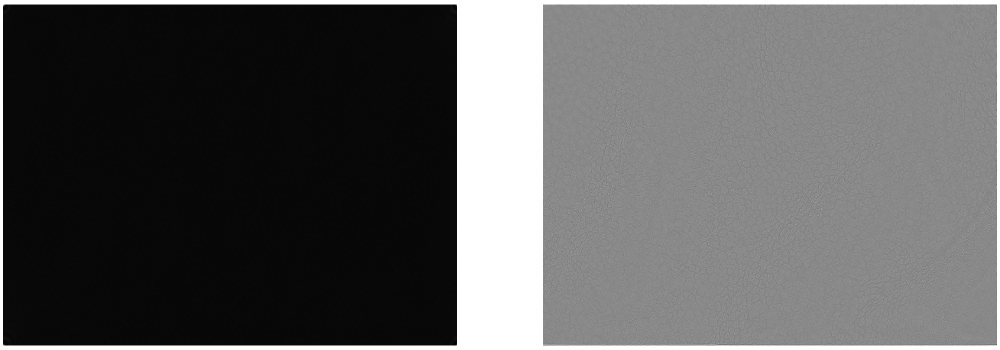
1.3 Inverse Fourier Transform

```python
## inverse Fourier transform
cornee2 = np.real(np.fft.ifft2(np.fft.fftshift(spectre)));
plt.figure()
plt.imshow(cornee2, cmap=plt.cm.gray);
plt.title('inverse FT');
```

The Fourier transform, although very powerful, is really difficult to interpret in 2D. The main information is not contained in the amplitude, but in the phase. The following reconstructions will illustrate it (see Fig.1).

```python
## inverse Fourier transform, without the phase
cornee_amplitude= np.real(np.fft.ifft2(np.fft.fftshift(A)));
plt.figure();
plt.imshow(cornee_amplitude, cmap=plt.cm.gray);
plt.title('inverse FT on amplitude');
```

```python
## inverse Fourier transform on phase only
complex_phase = np.exp(1j*G);
cornee_phase=np.real(np.fft.ifft2(np.fft.fftshift(complex_phase)));
plt.figure()
plt.imshow(cornee_phase, cmap=plt.cm.gray);
plt.title('inverse FT on phase');
```



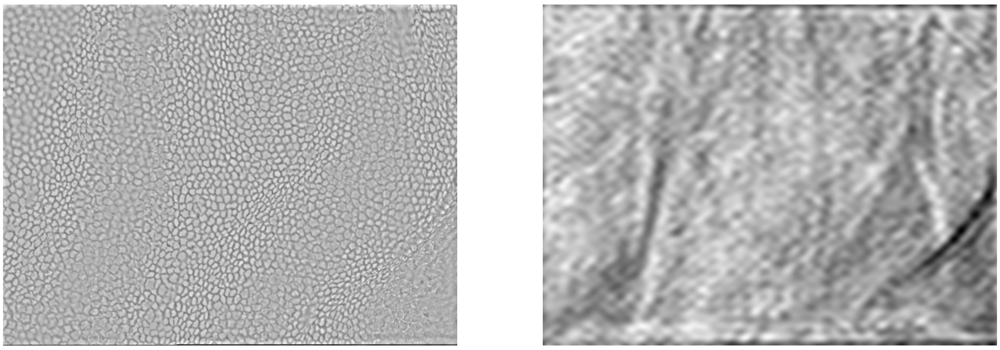
(a) Reconstruction with amplitude only.

(b) Reconstruction with phase only.

Figure 1: Reconstruction of partial informations (phase or amplitude only).

1.4 Low-pass and high-pass filtering

The results are illustrated in Fig. 2. Two functions are defined.



(a) High Pass filter.

(b) Low Pass filter.

Figure 2: Fourier basic filtering of cornea image.

A low pass-filter consists in the suppression of values for frequencies lower than a cut-off frequency.

```python
def LowPassFilter(spectrum, cut):
    """Low pass filter of the FFT (spectrum)
    The shape of this filter is a square. fftshift has been applied so that
    frequency 0 lays at center of spectrum image
    @param spectrum: FFT2 transform
    @param cut      : cut value of filter (no physical unit, only number of pixels)
    """
    X,Y = spectrum.shape;
    mask = np.zeros((X,Y), "int");
    mx = X/2; my = Y/2;
    mask[mx-cut:mx+cut, my-cut:my+cut] = 1;
    f = spectrum * mask;
    plt.figure
    plt.imshow(abs(f)); plt.title('Low pass filter')
    return f;
```

A high pass filter is exactly the opposite: suppression of the values under the cut-off frequency.

```python
def HighPassFilter(spectrum, cut):
    """High pass filter of the FFT (spectrum)
    The shape of this filter is a square. fftshift has been applied so that
    frequency 0 lays at center of spectrum image
    @param spectrum: FFT2 transform
    @param cut      : cut value of filter (no physical unit, only number of pixels)
    """
    X,Y = spectrum.shape;
    mask = np.ones((X,Y), "int");
    mx = X/2; my = Y/2;
    mask[mx-cut:mx+cut, my-cut:my+cut] = 0;
    f = spectrum * mask;
    plt.figure
    plt.imshow(abs(f)); plt.title('High pass filter')
    return f;
```

In the following application, the image is loaded and the effects of a low-pass and high-pass filters are illustrated.

```python
# FT of original image
cornea = misc.imread('cornee.png')
spectre = np.fft.fftshift(np.fft.fft2(cornea));

# low pass filter
L = LowPassFilter(spectre, 30)
viewSpectrumPhase(abs(L), np.angle(L))
corneaLP = np.real(np.fft.ifft2(np.fft.fftshift(L)))

# high pass filter
H = HighPassFilter(spectre, 30)
viewSpectrumPhase(abs(H), np.angle(H))
corneaHP = np.real(np.fft.ifft2(np.fft.fftshift(H)))

# display results and filters
plt.figure();
plt.subplot(1,2,1)
plt.imshow(corneaLP, plt.cm.gray); plt.title('reconstruction after LP
    ↪ filtering')
plt.subplot(1,2,2)
plt.imshow(corneaHP, plt.cm.gray); plt.title('reconstruction after HP
    ↪ filtering')
```

1.5 Application: evaluation of cellular density

The cells can be considered as a pattern, and its frequency repetition can found in the Fourier transform. More precisions on this application can be found in [2, 1, 3, 4] on the relation between the real cellular density and the image pixels.

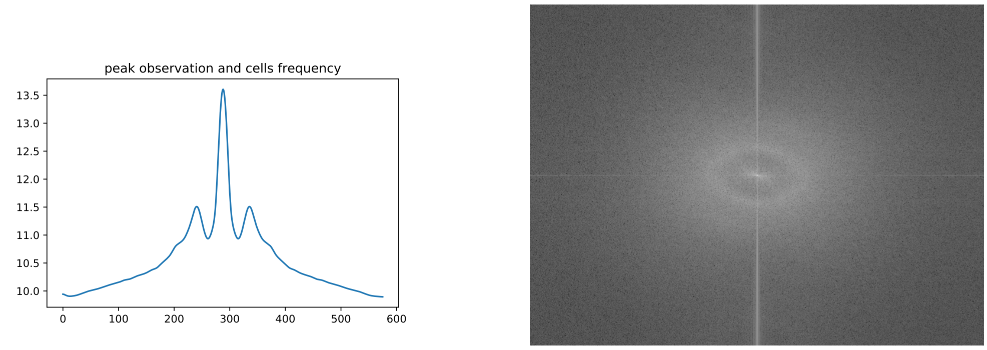
```python
cornea = misc.imread('cornee.tif')

# Fourier Transform
spectre = np.fft.fftshift(np.fft.fft2(cornea));
amplitude = abs(spectre);

# Filter amplitude
Blurred = ndimage.filters.gaussian_filter(amplitude, 5);
plt.figure
plt.subplot(1,2,1);
plt.imshow(np.log(1+Blurred), plt.cm.gray);
plt.title('filtered amplitude')

# Observe frequency peaks
plt.subplot(1,2,2);
plt.plot(np.log(1+Blurred[:,Y/2]));
plt.title('peak observation and cells frequency')
```

The results are illustrated in Fig. 3.



(a) Amplitude of Fourier Transform.

(b) Filtering of amplitude.

Figure 3: The two peaks represent the frequency of repetition of the cellular pattern, i.e. it can be used to compute the cellular density.

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. 5

[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. 5

[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. 5

[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. 5