Computer Vision 1: Exercise Sheet 7

Summary:

1. Inverse Fourier Transform of frequency spectrum and phase spectrum using `numpy`
2. Implement the Notch reject filtering on the Saturn ring image
3. Canny Edge Detector using `skimage`

In this exercise, we will replicate two experiments presented in the lecture, specifically the Inverse Discrete Fourier Transform (IDFT) of the frequency and phase spectra, as well as the simple Notch filtering that helped removing unexpected interfering signals from the image of the Saturn ring. This is our last encounter with the frequency domain. We then go back to the spatial domain with the Canny Edge Detector.

Three images are needed for this exercise: `woman.tif, saturn.tif` and our happy dog `MaruTaro.jpg`. As always, you can find them on Moodle.

Figure 1: A woman. Source: Gonzalez/Woods' DIP book

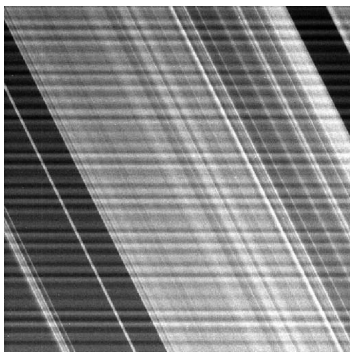


Figure 2: Saturn ring. Source: Gonzalez/Woods' DIP book

1 IDFT of frequency and phase spectra

Recall that the Inverse Discrete Fourier Transform is formally expressed as

$$\begin{aligned} f(x,y) &= \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(\frac{ux}{M}+\frac{vy}{N})} \\ &= \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} |F(u,v)|e^{j\phi(u,v)}e^{j2\pi(\frac{ux}{M}+\frac{vy}{N})} \end{aligned} \tag{1}$$

where $|F(u,v)|$ is the magnitude (frequency spectrum) and $\phi(u,v)$ is the angle (phase spectrum) of the complex number $F(u,v)$.

To compute the IDFT of the phase spectrum only, the frequency spectrum is set to 1:

$$f_{phase}(x,y) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} e^{j\phi(u,v)}e^{j2\pi(\frac{ux}{M}+\frac{vy}{N})} \tag{2}$$

On the other hand, the IDFT of the frequency spectrum only is computed by setting the phase spectrum to zero:

$$f_{frequency}(x,y) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} |F(u,v)|e^{j2\pi(\frac{ux}{M}+\frac{vy}{N})} \tag{3}$$

- Using `numpy.fft.fft2`, compute the DFT of the `woman.jpg` image. Extract the frequency and phase spectra of this DFT.
- Write a function that reconstructs the Fourier transform $F(u,v)$ from a frequency spectrum and a phase spectrum. The function should be similar to the following:

```
def Fuv(frequency, phase):
  """
  Input:
  ------
  frequency : an M x N array of real numbers
              The frequency spectrum.
  phase     : an M x N array of real numbers
              The phase spectrum.

  Output:
  ------
  An M x N array of complex numbers.
  The Fourier transform reconstructed from frequency and phase
   spectra.
  """
  return ... # One line of code is expected
```

  Note: in Python, the imaginary number j in the equations above is represented as `1j`.

- Compute the IDFT of the Fourier transform reconstructed by the *Fuv* function in the previous task. The result should look identical to the original woman image.
- By setting the value for the frequency spectrum parameter to 1, compute the IDFT of the phase spectrum only. The result should contain the shape of the woman as shown in the slides of lecture 6.

- By setting the value for the phase spectrum parameter to 0, compute the IDFT of the frequency spectrum only. Apply `numpy.fft.fftshift` to obtain an output similar to what is shown in the slides of lecture 6.
- Using `numpy.rot90`, rotate the original image 180° and compute the FFT of the rotated image. What can we observe about the frequency and phase spectra of the rotated image, compared to the frequency and phase spectra of the original image? **Hint**: instead of the original image, you might want to start experimenting with a random 2D array of small dimensions (like 3×3 or 4×5), rotate it and observe how the spectra differ between before and after rotation.
- (Optional) If you observe some kind of pattern, can you give a mathematical proof that the pattern is true?

2 The simple Notch reject filtering on the Saturn image

- Read the Saturn image above and compute its DFT. Shift the DFT and visualize the frequency spectrum.

  Note when visualizing the frequency spectrum: if the log of the spectrum is computed, the result might contain infinity values `-inf`. Set all infinity values to 0 before plotting.
- Create an array for the Notch filter (see Figure 3). This array should have the same shape as the Saturn image.

  Note that the vertical black segments in the middle of the image are exactly 1 pixel wide. The space between them is 10 pixels long (i.e. 5 pixels from the center of the image to each of them).
- Apply this filter to the shifted DFT above to obtain a new DFT. Visualize the new frequency spectrum after filtering.
- Using `numpy.fft.ifft2`, compute the inverse of this new DFT (no padding is required). Do not forget to shift the DFT back before computing the inverse. The image result should look nice and clean as displayed in the slides of lecture 6.
- How can we obtain a visualization the horizontal interfering signal instead? Verify this by adding one line of code to the procedure you have written so far.

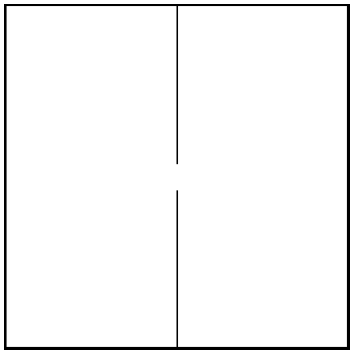
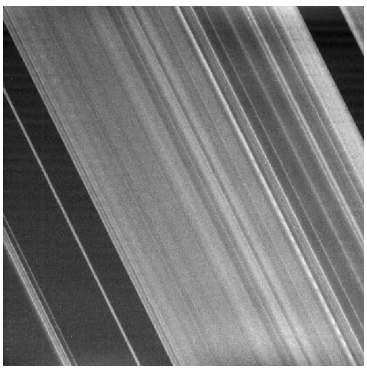


Figure 3: A simple Notch reject filter. Note that the black border is added to aid visualization, not part of filter. Source: Gonzalez/Woods' DIP book

The expected results are shown in Figure 4.



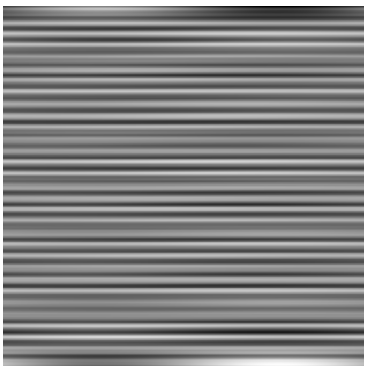


Figure 4: Expected results of the Notch filterings. Left: the signal after filtering. Right: the interfering signal.

3 Canny Edge Detector

- Using `skimage.feature.canny`, compute and visualize the edges in the `MaruTaro.jpg` image. Try different values for the parameter σ from 0.5 to 5. Depending on this parameter, the result might look similar to Figure 5. What can be noticed as σ increases?

Figure 5: Canny Edge Detection on the Maru Taro image