Fourier Transform and Frequency Filtering

Matias Cinera
Computer Science Department, *University of South Florida*
Tampa, Florida, 33612, USA
cinera@usf.edu

I. Introduction

This document is the report of my Assignment 2 from my Fall 2023 class CAP-5400 Digital Image Processing, as a graduate student. The individuals who are overviewing this assignment are our professor Dr. Dmitry Goldgof and Mr. Anthony McCofie (Teaching Assistant). The purpose of this report is to explore the Discrete Fourier transformation in grey-level images along with High/low filter cut-off frequency & unsharp masking. It will also apply these same concepts to HSV images.

II. Discrete Fourier Transform

The Discrete Fourier Transform (DFT) is applied to grayscale images to convert them from the spatial domain to the frequency domain.

A. Implementation

To process an image using the Discrete Fourier Transform (DFT), follow these steps: First, resize the image to an optimal size, ensuring it's a multiple of two, three, or five for improved DFT performance. Then, allocate space for both complex and real values by converting the image to a float format with an extra channel for complex values. Apply the DFT, calculate the magnitude of the complex results, and convert to a logarithmic scale for visualization. Next, crop the image and rearrange quadrants for visualization purposes, and finally, normalize the values to the range of zero to one using `normalize()` for proper display.

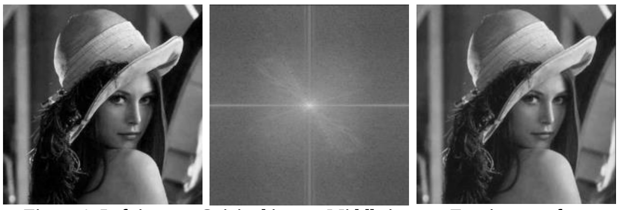

Figure 1: Left image - Original image. Middle image - Fourier transform Right image - Inverse fourier transform

B. Low-pass Filter

Using a lowpass filter in Fourier transform involves attenuating high-frequency components while preserving low-frequency ones. A low-pass filter applied to an image in the frequency domain smooths or blurs the image, reduces noise, and softens sharp edges. This leads to a cleaner appearance but can also result in a loss of fine details and reduced image contrast.

$$H(u,v) = \begin{cases} 0, & \text{if } \sqrt{(u-\frac{M}{2})^2 + (v-\frac{N}{2})^2} \le F \\ 1, & \text{otherwise} \end{cases}$$

Figure 2: Low Pass Filter equation

- **H(u, v) is the filter on the frequency domain.**
- **M and N are the number of rows and columns of the filter**
- **The sqrt expression represents the circular shape of the filter**
- **F cut off frequency.**

Figure 3: Inverse Fourier Transform with a Low Pass Filter (F = 20 & 50)

C. High-pass Filter

Applying a highpass filter in Fourier transform entails emphasizing high-frequency components while suppressing low-frequency ones. In the frequency domain, a high-pass filter enhances image details, highlights edges, and accentuates fine features, resulting in increased image contrast. However, this process can also introduce noise and may make the image appear sharper and less smooth, potentially leading to some visual artifacts and reduced overall image smoothness.

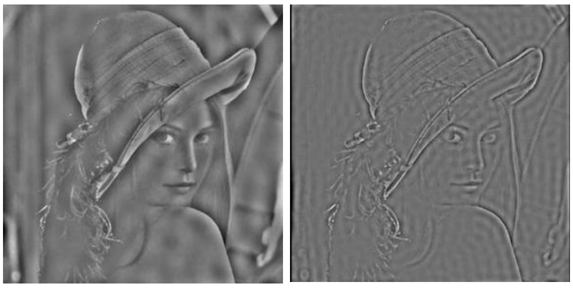

Figure 4: Inverse Fourier Transform with a High Pass Filter (F = 15 & 20)

$$H(u,v) = \begin{cases} 1, & \text{if } \sqrt{(u-\frac{M}{2})^2 + (v-\frac{N}{2})^2} \le F \\ 0, & \text{otherwise} \end{cases}$$

Figure 5: High Pass Filter equation

D. Band-pass Filter

A bandstop filter is typically used to remove interference or unwanted frequencies from an image. When applied to an image in the frequency domain, it selectively eliminates a narrow band of frequencies, effectively removing specific patterns or noise. To implement this filter just take the difference between the High and Low pass filters. Just make sure the cutoff for the High filter is lager than the Low filter.

Figure 6: Cathedral Gray-Scale Image

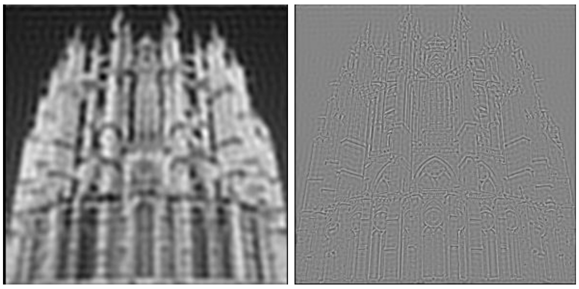

Figure 7: Left – Low Pass (F=20), Right – High Pass (F=50)

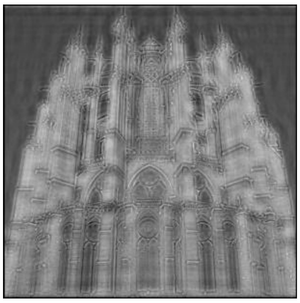

Figure 8: Band Stop Filter Image (Low=20, High=50)

E. Unsharp Masking

Unsharp masking is an image enhancement technique that involves creating a "mask" image by subtracting a blurred version of the original image from the original. This mask is then amplified and added back to the original image to enhance its sharpness and fine details, making edges and features stand out more clearly. To compute it, just filter out an image (I used a gaussian filter, but can be done with a high pass filter) and get the difference with the original. Then multiply the frequency domain with T, after that add the filter.

$$highFreq = image - filter$$
$$sharpened = filter + T \cdot highFreq$$

Figure 9: Unsharp Masking equation

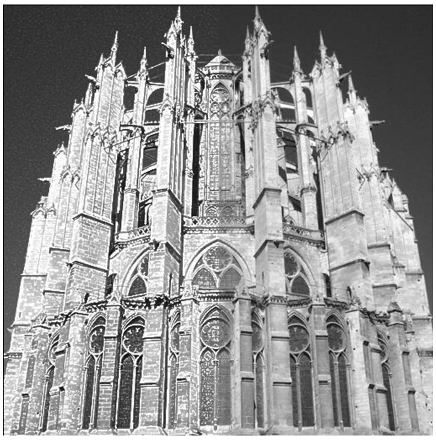

Figure 10: Left side – Unsharp Masking (T=50), Right side - Original image

III. Frequemcy Filtering on the HSV Color Space

Filtering on the HSV color space is very similar to gray scale level images. Just split out the channels, process the desired channel, then merge back with the 2 other channels.

Figure 11: Baboon color image

A. Low-Pass Filter

- **Hue**: Smooth out color transitions and reduce color noise. It can make the colors appear more uniform and less vibrant.
- **Saturation**: Reduces the intensity of color and makes the image appear less saturated.
- **Value**: will make the image appear more uniform in terms of brightness and reduce noise.

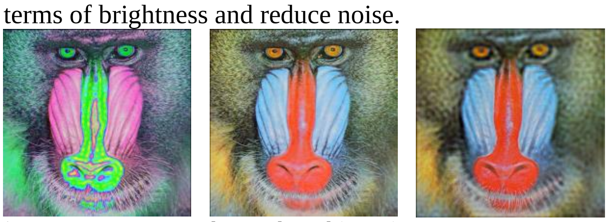

Figure 12: Low Pass on each HSV channel (F = 35)

B. High-Pass Filter

- **Hue**: Highlight the sharp changes in color. This could lead to more pronounced edges between different colors.
- **Saturation**: Enhances the edges or boundaries of objects, making them more distinct.
- **Value**: Enhances fine details and edges in brightness, making the image appear sharper.

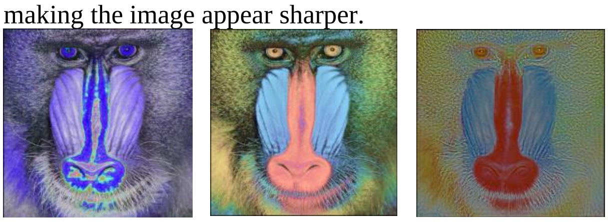

Figure 13: High Pass on each HSV channel (F=40)

C. High-Pass Filter

- **Hue**: remove a range of hues, effectively removing certain colors while leaving others intact.
- **Saturation**: Enhances specific or suppress saturation levels.
- **Value**: Enhance or suppress specific brightness levels.

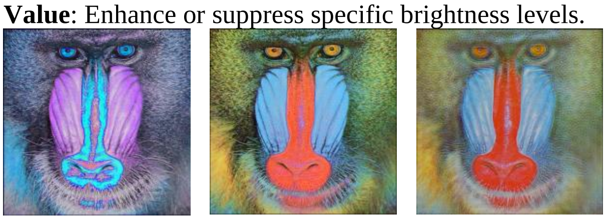

Figure 14: Low Pass on each HSV channel (Low=25, High=55)

IV. References

- Datahacker.rs. (2019, October 26). OPENCV #012 discrete Fourier transform, part 2. Master Data Science. https://datahacker.rs/opencv-discrete-fourier-transform-part2/#id1
- Unsharp masking with python and opencv. Instruments & Data Tools. (2021, May 5). https://www.idtools.com.au/unsharp-masking-with-python-and-opencv/
- Discrete fourier transform. OpenCV. (n.d.). https://docs.opencv.org/4.x/d8/d01/tutorial_discrete_fourier_transform.html