Lab 5 Frequency Domain Filtering

Jincheng Yu

April 4, 2023

1 Introduction

In this experiment, we investigate the effectiveness of different filters in the frequency domain by generating various filter templates in the frequency domain, multiplying the source image, and then applying inverse Fourier transform to obtain the result image. We compare the results of these frequency domain filters with that of spatial domain filtering.

The four filters used in this experiment are the Sobel filter, ideal lowpass filter, Gaussian filter, and notch filter. The Sobel operator takes the weighted difference of the gray value of each pixel in the top, bottom, left, and right areas of the image and detects the edge by reaching the extreme value at the edge. The Sobel operator produces better detection results and has a smooth suppression effect on noise, but the edges obtained are thicker, and false edges may appear. The notch filter is a selective filter that can attenuate a signal at a specific frequency point and block the passage of this frequency signal. It can quickly attenuate the input signal at a certain frequency point to achieve the filtering effect of blocking the passage of this frequency signal. The notch filter is a more useful selective filter. The notch band rejection filter can be constructed by the product of the high-pass filter whose center has been shifted to the center of the notch filter.

Through this experiment, we aim to gain a better understanding of the performance and limitations of different frequency domain filters, and how they compare with traditional spatial domain filtering techniques.

2 Sobel Filter and Butterworth Notch Filters

2.1 Sobel Filter

The Sobel filter is a commonly used edge detection filter in image processing. It works by computing the gradient of the image intensity at each pixel. Specifically, it calculates the first-order derivative of the image intensity along the horizontal and vertical directions separately, and then combines them to get an approximation of the magnitude of the gradient at each pixel.

The Sobel operator is defined as follows: $gx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ and $gy = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$ where G_x and G_y are the kernels for calculating the gradient along the horizontal and vertical directions, respectively. The image is convolved with these two kernels to obtain the gradient images I_x and I_y:

$$I_x = G_x * I$$

$$I_y = G_y * I$$

The magnitude of the gradient at each pixel is then approximated by the following formula:

$$I_{mag} = \sqrt{I_x^2 + I_y^2}$$

The Sobel filter is effective in detecting edges in images because it highlights areas where the intensity changes sharply.

2.2 Butterworth notch filters

Butterworth notch filters are a type of filter used in signal processing to remove unwanted frequency components from a signal. Unlike traditional notch filters that can only remove a single frequency component, Butterworth notch filters can remove multiple frequency components simultaneously.

The Butterworth notch filter is based on the Butterworth filter, which is a type of low-pass filter that has a maximally flat frequency response in the passband. To create a Butterworth notch filter, the Butterworth transfer function is modified by adding one or more notches, which are frequency regions where the filter will attenuate the signal.

The transfer function of a Butterworth notch filter is given by:

$$H(s) = \frac{1}{1 + (\frac{w_0}{s})^{2n}} \prod_{k=1}^{N} \frac{(s^2 - 2\zeta_k \omega_{0k} s + w_{0k}^2)}{(s^2 + 2\zeta_k \omega_{0k} s + w_{0k}^2)}$$

where s is the complex frequency variable, ω_0 is the center frequency of the notch, n is the order of the filter, N is the number of notches, ζ_k is the damping coefficient for the k^{th} notch, and ω_{0k} is the center frequency of the k^{th} notch.

The Butterworth notch filter can be implemented in the frequency domain by taking the Fourier transform of the input signal, multiplying it by the transfer function of the filter, and then taking the inverse Fourier transform to obtain the filtered signal.

Overall, Butterworth notch filters are a useful tool for removing unwanted frequency components from signals in a wide range of applications, including audio processing, image processing, and telecommunications.

3 Experiment

3.1 Implement

3.1.1 Sobel Filter

Sobel filter uses sobel operator to calculate every new pixel grayscale value according to the input image. The algorithm is as following.

Data: Input image I
Result: Output image J

1. Apply zero padding to the input image I_p;
2. Apply a phase shift to the image to center the frequency spectrum around the origin $(-1)^{x+y}$;
3. Compute the Fourier transform of the padded image F(u,v);
4. Generate spatial Sobel x and y filters;
5. Apply the Sobel filters in the frequency domain using convolution E(u,v);
6. Combine the edges obtained from the x and y Sobel filters $J_p(u,v) = E(u,v) + I(u,v)$;
7. Remove the padded edges from the output image J(u,v);
8. Scale the edge image to a range of 0-255 J;

In the python, the following code is zero padding.

```
    pad_image = np.zeros((height + 2, width + 2))  # zero padding
```

To center the frequency spectrum, the following code shows the process of multiply $(-1)^{x+y}$

```
    for i in range(height + 2):
        for j in range(width + 2):
            pad_image[i, j] *= (-1) ** (i + j)
```

The following code is used to extract edge by sobel filter.

```
dge_x = cv.filter2D(np.real(np.fft.ifft2(img_freq)), -1, sobel_calx)
    edge_y = cv.filter2D(np.real(np.fft.ifft2(img_freq)), -1,
        sobel_caly)
    edge = np.abs(edge_x) + np.abs(edge_y)
```

The following code shows the process of normalize the grayscale.

```
    output_image *= (output_image > 0)
    output_image = output_image * (output_image <= 255) + 255 * (
        output_image > 255)
```

3.1.2 Butterworth notch filters

The histogram matching calculates the reflection between the histogram of input image and specific histogram and the new grayscale of every point in output figure according to the reflection. The algorithm is shown as follows.

Data: Input image I

Result: Output image J

Apply zero padding to the input image I_p;

Apply 2D Fourier Transform $F(u, v)$;

Generates the Butterworth notch filter in the frequency domain $D(u, v)$;

Apply the filter $O_p = F * D$;

Inverse Fourier transform

J$_p$ $Removepaddingzeros J(u, v)$ $Scaletheedgeimagetoarangeof 0 - 255$J

Based on the notch pairs we choose, we can generate the notch filter matrix H_{NF}.

```
for v_k, u_k in notch_pairs:
    D = 30
    D_k = np.sqrt((np.arange(P)[:, np.newaxis] - u_k) ** 2 + (np.
        arange(Q) - v_k) ** 2)
    D_k[D_k == 0] = 1
    H_NF *= 1 / (1 + (D / D_k) ** 8)
```

The following code shows the process of fft and ifft to get the output image.

```
F = np.fft.fft2(img)
G = np.fft.fftshift(F) * H_NF
g = np.fft.ifft2(np.fft.ifftshift(G)).real
```

3.2 Result and Analysis

Figure 1: Original image

(a) Spatial domain filter

(b) Frequency domain filter

Figure 2: Original image

The Sobel filter result shows in the Fig.1. The figure presented above illustrates the filtering outcomes of the Sobel filter, which were produced in both the spatial and frequency domains. The Sobel operator effectively extracted the edge component of the image. When the edge component was merged with the initial image, a sharpened version of the image was obtained. As discussed in the 'Experiment' section, the results generated by both filters were identical, indicating the success of the experiment. It is important to note that while the edge image was scaled, the final image was not.

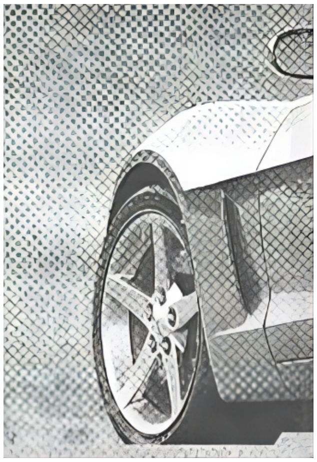
When we convert the Sobel operator from the spatial domain to the frequency domain, we need to maintain the odd symmetry of the Sobel template. To achieve this, we place the center of the Sobel template at the center of the array, after filling f and h as 602*602. In the spatial domain, this is equivalent to multiplying the frequency domain template by $(-1)^{x+y}$. In the frequency domain, we multiply the frequency domain template by $(-1)^{x+y}$ to offset the effect of the previous operation. If we do not perform the decentralized operation, the image will be cut into four pieces, with the centers of the pieces located at the four corners. The output image will also become four blocks from the original image, with the center point being abnormal because it is not decentralized. The result shown in Fig.3.

Figure 3: Output image without center shifting

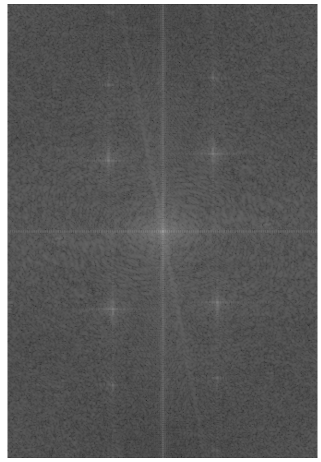
The lab pictures of butterworth filer are in Fig.4. The depicted image showcases the outcome of employing Butterworth notch filter to process an image with moiré. The illustration presents the spectra of the original image, filter, and output image. Analysis of these spectra reveals that the filtering operation has successfully eliminated the moiré frequencies from the original image spectrum, leading to a significant improvement in the clarity and size of the output image while retaining certain details.

The presented image showcases the outcome of utilizing a Butterworth notch filter to filter an image with moiré patterns. The figure includes the spectra of the original image, the filter, and the resultant image, respectively. Upon closer examination, it becomes apparent that the moiré components present in the original image spectrum have been efficiently eliminated by the Butterworth notch filter. As a consequence, the filtered image displays a higher level of clarity and enlarged details, while still preserving some of the original image's finer details.
In terms of selecting the notch parameters, the frequency spectrum of the original image indicates the existence of four pairs of moiré ripples. Thus, to define four notch pairs, we set n=4. To determine the positions of these notch pairs, we calculate the coordinates of the four notch centers from the center of the spectrum rectangle. Through rough estimation and experimentation, we have identified the four coordinate points: (55,80), (55,150), (-55,85), and (-55,150), with their opposite numbers corresponding to the center of another notch. For the selection of the parameter D0, we measure the radius of the moiré ripple in the frequency spectrum, and then choose D0=30 to ensure that it can encompass the entire energy of the pulse.

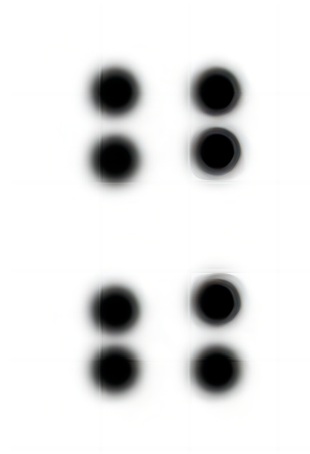
Furthermore, the main reason why the frequency domain function of a filter is real and symmetric is to simplify the computation. The imaginary part of a real symmetric filter is zero, and the filter can be regarded as a real function. This means that when filtering in the frequency



(a) Original figure

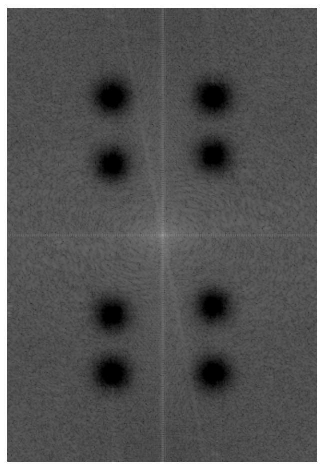


(b) Spectrum



(c) Filter

(d) Filtered figure



(e) Output spectrum

Figure 4: The figures of processing butterworth notch filter

domain, the complex part can be ignored, reducing the computational load and increasing processing speed. Additionally, in the case of real symmetry, the amplitude response of the filter's frequency domain function is an even function, and the phase response is an odd function, so there is no phase distortion introduced. As for the exception, When it is necessary to perform bandpass or bandstop filtering on an image, complex filters are usually used because these filters need to remove or preserve a certain frequency range in the frequency domain, and these frequency ranges may not be on the real axis.

4 Conclusion

In conclusion, the Sobel frequency domain filter and the Butterworth notch filter are powerful image processing tools with unique features. The Sobel filter is particularly useful for detecting edges and enhancing image details in the frequency domain, while the Butterworth filter is effective in selectively eliminating specific frequency ranges, especially when dealing with moiré patterns. By combining these filters, we can achieve more precise and comprehensive image processing results. These tools have great potential for various applications, including medical image analysis, surveillance systems, and scientific imaging.

This laboratory exercise demonstrates the implementation of the Sobel frequency domain filter and compares its results with those obtained from the Sobel spatial domain filter used in the previous lab. The results prove that the two outputs are identical, indicating that the spatial template is properly centered, Fourier transformed, decentralized, and the result obtained by inverse Fourier transform after multiplying the Fourier transform of the input image in the frequency domain is equivalent to the result of convolution in the spatial domain. Additionally, a selective filter, the Butterworth notch filter, is utilized to precisely eliminate the moiré present in the input image using four notch pairs. The filter and its output are displayed intuitively through the frequency spectrum.