# Digital Image Processing, 2024 Spring

# Homework 2

Name: Zhou Shouchen
Student ID: 2021533042

Due 23:59 (CST), May. 3, 2024

## Problem 1: Image Sharping

(a) Let $\boldsymbol{a} = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}^T$, and $\boldsymbol{b} = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}^T$.

Sobel operator among $x$ direction: $S_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = \boldsymbol{a}\boldsymbol{b}^T$

Sobel operator among $y$ direction: $S_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} = \boldsymbol{b}\boldsymbol{a}^T$

So both the Sobel operators can be represented as the outer product of two vectors, i.e. its separable.
From what we have learned, if a separable filter can be represented as $\boldsymbol{w} = \boldsymbol{w}_1\boldsymbol{w}_2^T$, then the convolution of the filter with an image can be computed as $\boldsymbol{I} * \boldsymbol{w} = (\boldsymbol{I} * \boldsymbol{w}_1) * \boldsymbol{w}_2^T$, where '$*$' represents convolution.
So we can implement the separated kernels sequencially to the origin image to get the results.
The $S\_x\_a$ represents the convolution of origin image and $\boldsymbol{a}$, $S\_x\_ab$ represents convolution of $S\_x\_a$ and $\boldsymbol{b}^T$, which also means that the convolution of origin image and the $S_x$ operator.
And $S\_y\_b$ represents the convolution of origin image and $\boldsymbol{b}$, $S\_y\_ba$ represents convolution of $S\_y\_b$ and $\boldsymbol{a}^T$, which also means that the convolution of origin image and the $S_y$ operator.
Specifically, for the convenience of checking, the negative values are turned to 0, and then the image pixel values are normalized to $[0, 255]$ after filtering.
And the results are shown in Figure 1.

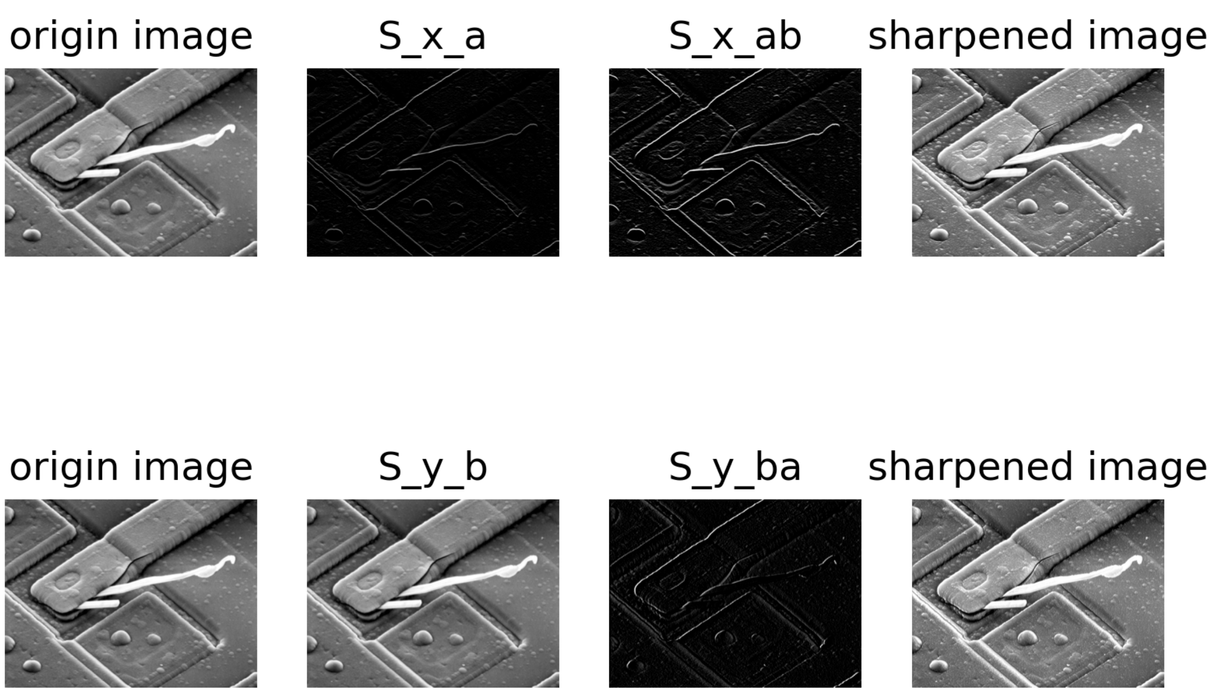



Figure 1. Processed images by Sobel operators among $x$ and $y$ directions.

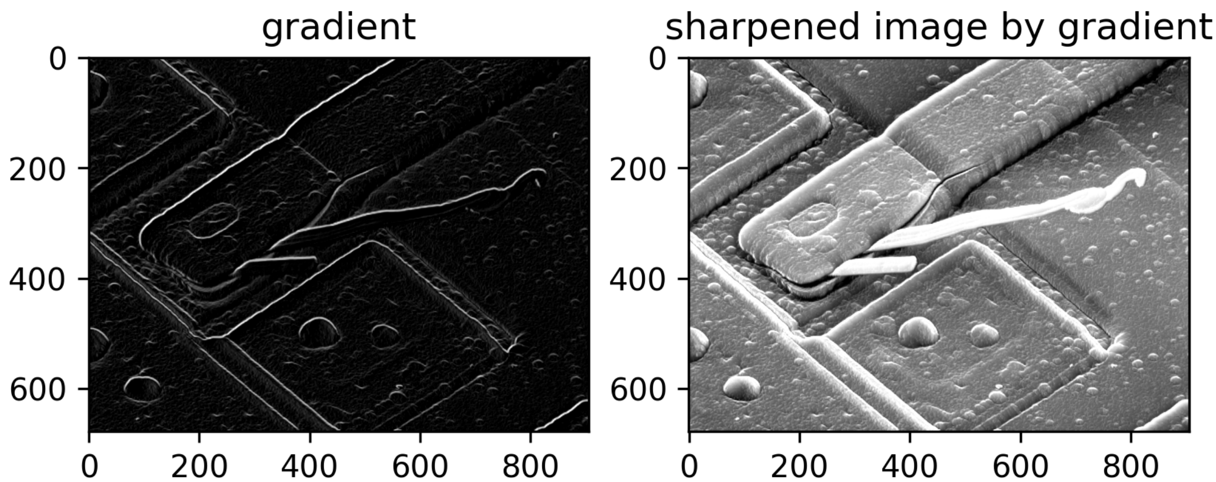



Figure 2. p1a gradient

(b) Gaussian Highpass Filter:

$$D(u,v) = \left[ \left(u - \frac{P}{2}\right)^2 + \left(v - \frac{Q}{2}\right)^2 \right]^{\frac{1}{2}}$$

$$H(u,v) = 1 - \exp\left(-\frac{D(u,v)^2}{2D_0^2}\right)$$

In order to do FFT, we need to pad the image to the size of $2^m * 2^n$, where $m$ and $n$ are integers.
So the images size varies: $678 * 906 \Rightarrow 1024 * 1024$.
And aftering filtering, the image is cropped to the original size.

The Gaussian high pass filter is shown in Figure 3.

To have a better effect of visualization on the frequency domain, the image is shown by applying the log transformation on the magnitude of the Fourier transform of the image.
i.e. $I_{\log} = 20 \log(1 + |I|)$, where $I$ is the spectrum generated by Fourier transform of the image.
The results of filtering the image with the Gaussian high pass filter and their frequency domain results are shown in Figure 5.
The filtered image is generated by using the inverse Fourier transform to the frequency domain results in the real domain.

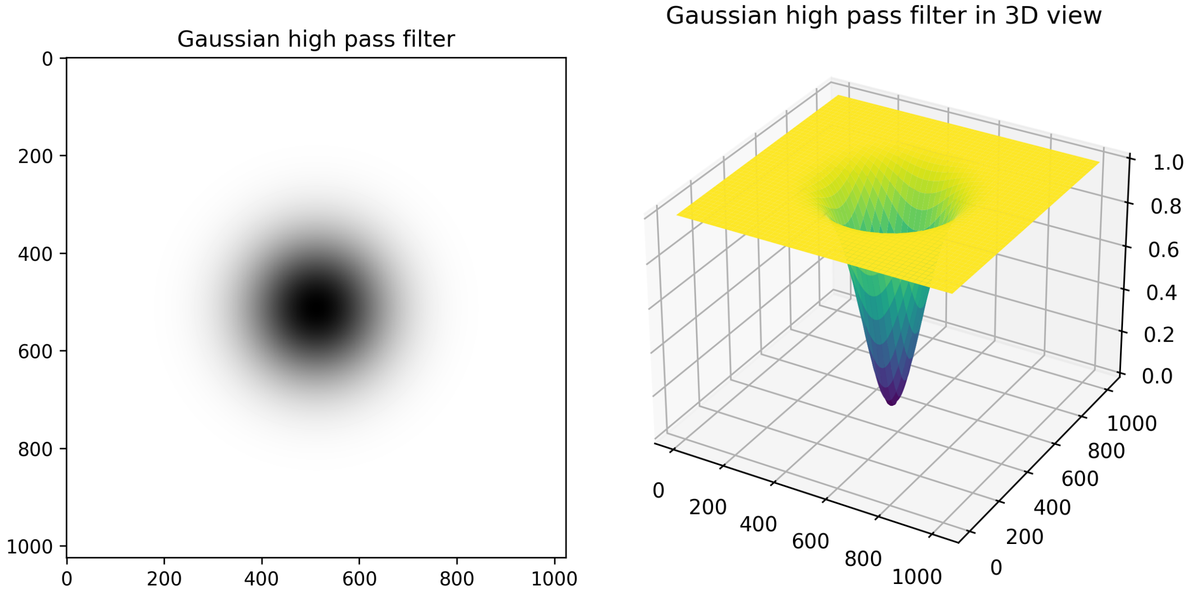



Figure 3. Gaussian high pass filter

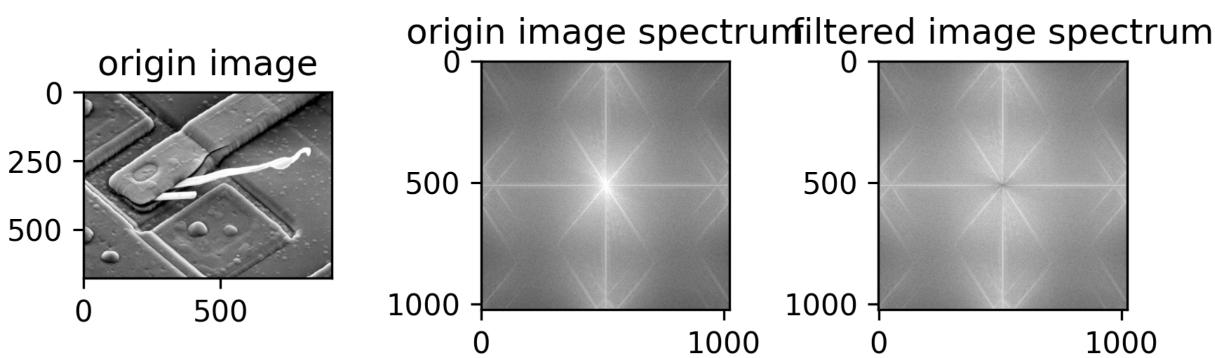



Figure 4. Results of filtering the image with the Gaussian high pass filter.

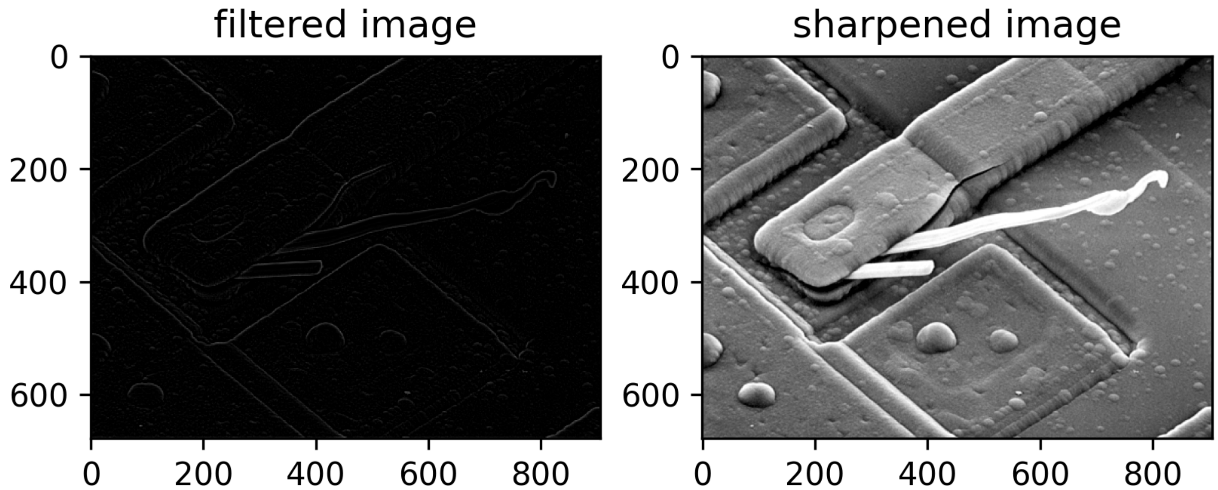



Figure 5. Results of filtering the image with the Gaussian high pass filter.

## Problem 2: Homomorphic Filtering

The Homomorphic filter:

$$H(u,v) = (\gamma_H - \gamma_L) * \left[1 - \exp\left(-c\left[\frac{D(u,v)}{D_0}\right]^2\right)\right] + \gamma_L$$

Where

$$\gamma_L < 1, \gamma_H \geq 1, D(u,v) = \left[\left(u - \frac{P}{2}\right)^2 + \left(v - \frac{Q}{2}\right)^2\right]^{\frac{1}{2}}$$

In this specific situation, we take $\gamma_L = 0.5, \gamma_H = 2.0, D_0 = 80, c = 1$.

In order to do FFT, we need to pad the image to the size of $2^m * 2^n$, where $m$ and $n$ are integers.

So the images size varies: $1162 * 746 \Rightarrow 2048 * 1024$.

And aftering filtering, the image is cropped to the original size.

The Homomorphic filter is shown in 6.

The results of filtering the image with the Homomorphic filter is shown in Figure 7.

The variance of pixel values with in the white box $[500, 1100] \times [90, 180]$ is 0.0017427355, which exceeds $3 \times 10^{-4}$.

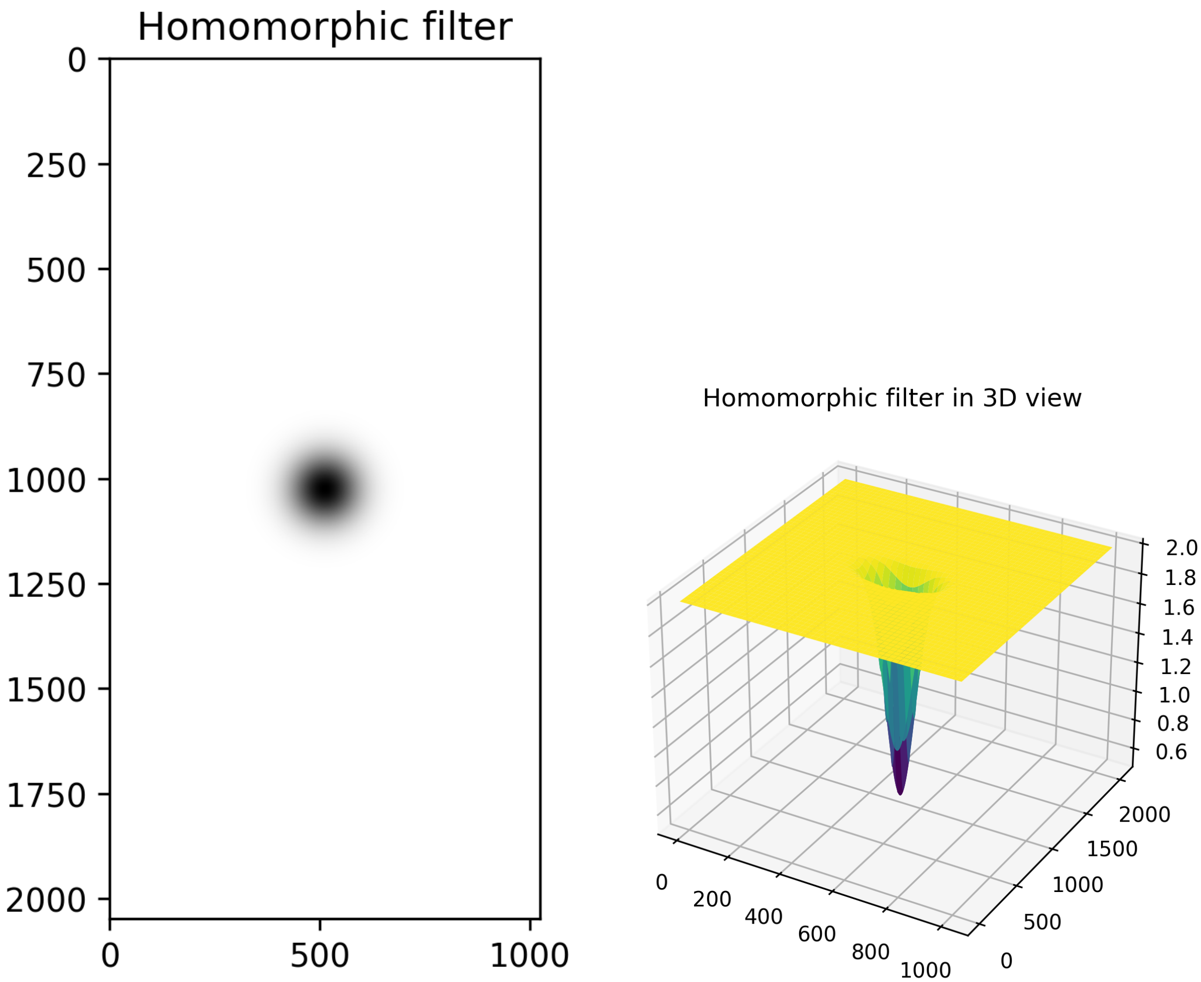



Figure 6. Homomorphic filter

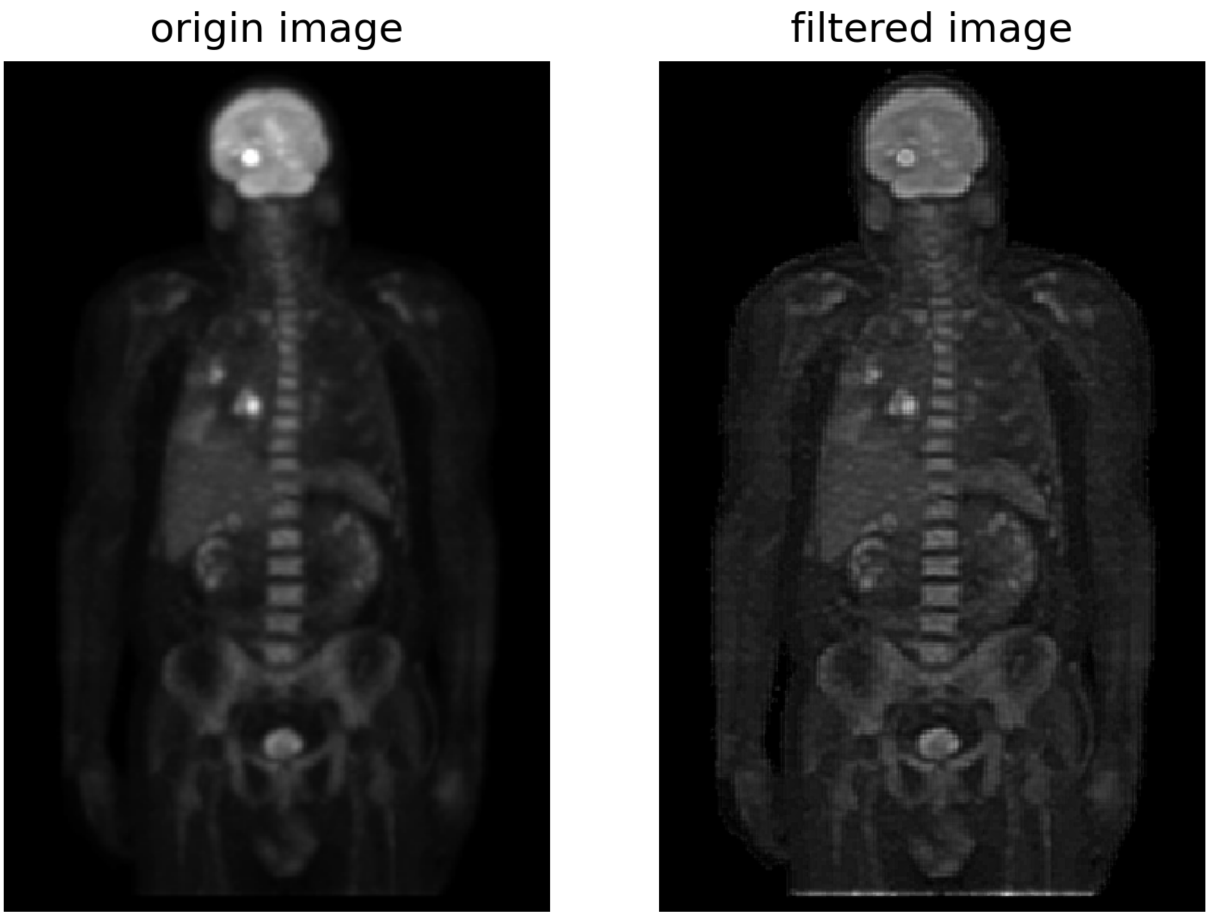



Figure 7. Result after Homomorphic filtering

## Problem 3: Color space conversion

1. Convert RGB image to HSI image.

The formula to convert RGB to HSI is as follows:

$$\theta = \arccos\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{\frac{1}{2}}}\right\}$$

$$H = \begin{cases} \theta & \text{if } B \le G \\ 360-\theta & \text{if } B > G \end{cases}$$

$$S = 1 - \frac{3}{R+G+B}\min\{R,G,B\}$$

$$I = \frac{R+G+B}{3}$$

And the HSI channels should be normalized to $[0,1]$.
Since $H$ represents the hue, which is a circular value, and its range is $[0,360]$.
$S$ represents the saturation, its range is $[0,1]$.
$I$ represents the intensity, its range is $[0,255]$.
So to normalize the HSI channels, we can use the following formula:

$$H \leftarrow \frac{H}{360}, S \leftarrow \frac{S}{1}, I \leftarrow \frac{I}{255}$$

For details, since we need to avoid the situation that the denominator is 0, so we can add $\epsilon = 10^{-9}$ on the denominator when calulating $\theta$ and $S$.
Also, in order to fit the domain of the function arccos, we need to clip the values into $[-1,1]$ when calulating $\theta$.
So with above formulas, we can convert the RGB image to HSI image.

2. Convert HSI image to RGB image.

The formula to convert HSI to RGB is as follows:

$$0^\circ \le H < 120^\circ$$

$$B = I(1-S), \quad R = I\left[1+\frac{S\cos(H)}{\cos(60^\circ - H)}\right], \quad G = 3I-(R+B)$$

$$> 120^\circ \le H < 240^\circ$$

$$R = I(1-S), \quad G = I\left[1+\frac{S\cos(H-120^\circ)}{\cos(180^\circ - H)}\right], \quad B = 3I-(R+G)$$

$$> 240^\circ \le H < 360^\circ$$

$$G = I(1-S), \quad B = I\left[1+\frac{S\cos(H-240^\circ)}{\cos(300^\circ - H)}\right], \quad R = 3I-(G+B)$$

And the RGB channels should be normalized to $[0,255]$.
So to normalize the RGB channels, we need to recover the HSI by using the following formula:

$$H \leftarrow H * 360, S \leftarrow S * 1, I \leftarrow I * 255$$

For specific details, all values of $H$ are using the radians, so the numbers of the degrees above should be converted to radians while implementing.

With these formulas, we can convert the HSI image to RGB image.

With the analysis above, we can convert the RGB image to HSI image, and recover the RGB image from the HSI image.

The Figure 8 is the result of the conversions.

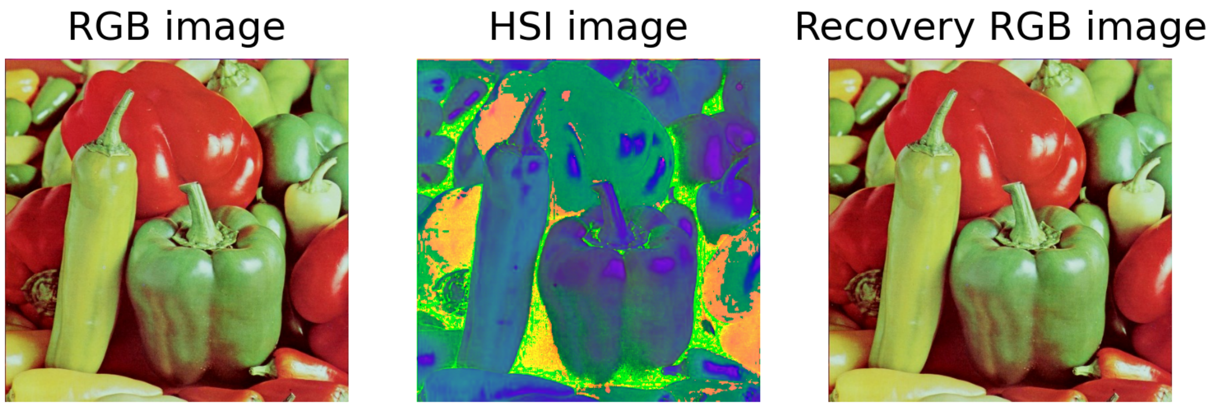



Figure 8. RGB to HSI conversion and the recovery

## Problem 4: Image Restoration

(a) The spectrum of the origin image is generated through FFT.
To shift the zero frequency to the center of the spectrum, we times $(-1)^{u+v}$ for each pixel $(u, v)$ on the origin image before applying FFT.

To have a better effect of visualization on the frequency domain, the image is shown by applying the log transformation on the magnitude of the Fourier transform of the image.
i.e. $I_{\log} = 20\log(1 + |I|)$, where $I$ is the spectrum generated by Fourier transform of the image.

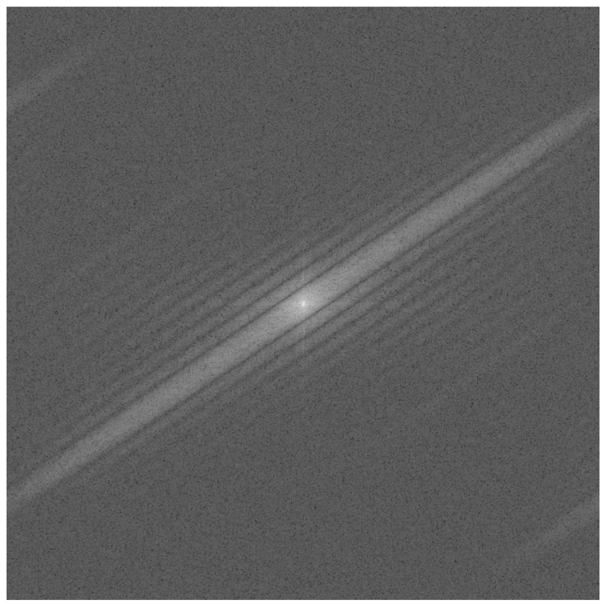



Figure 9. FFT shifted spectrum

(b) The image by applying Radon transform on the origin image is shown in Figure 11.
The Radon transform:

$$g(\rho, \theta) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)\delta(x\cos\theta + y\sin\theta - \rho)$$

where $\delta$ is the Dirac delta function.

We can get the coordinates with the highest intensity in the Radon transformed image, which is $(\theta, d) = (, )$.

So above all, the angle between the strip and the vertical direction is $\theta =$ and the distance between two similar dark strips is $d =$

$N = 640, L = \dfrac{N}{d} =$

(c) We construct the frequency domain Wiener filter using the formula:

$$W = \frac{H^*}{|H|^2 + \dfrac{S_n}{S_f}}$$

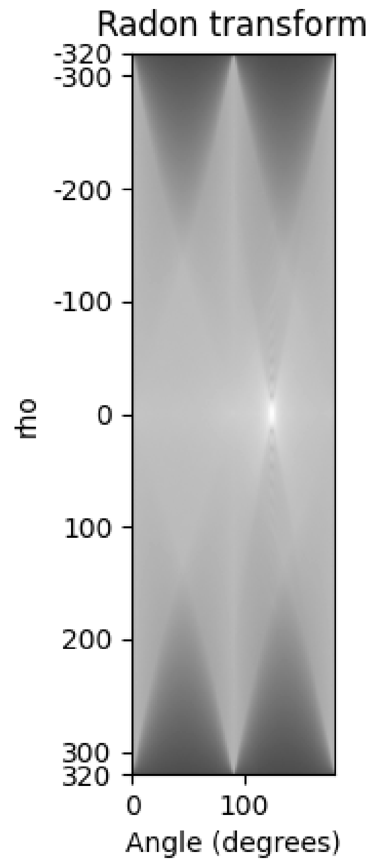
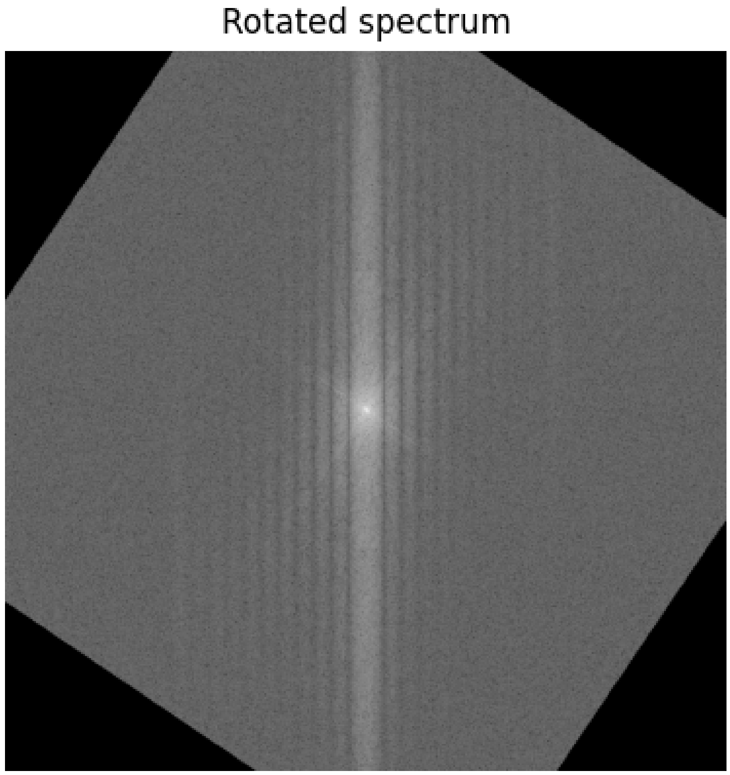



Figure 10. Radon Transformed image



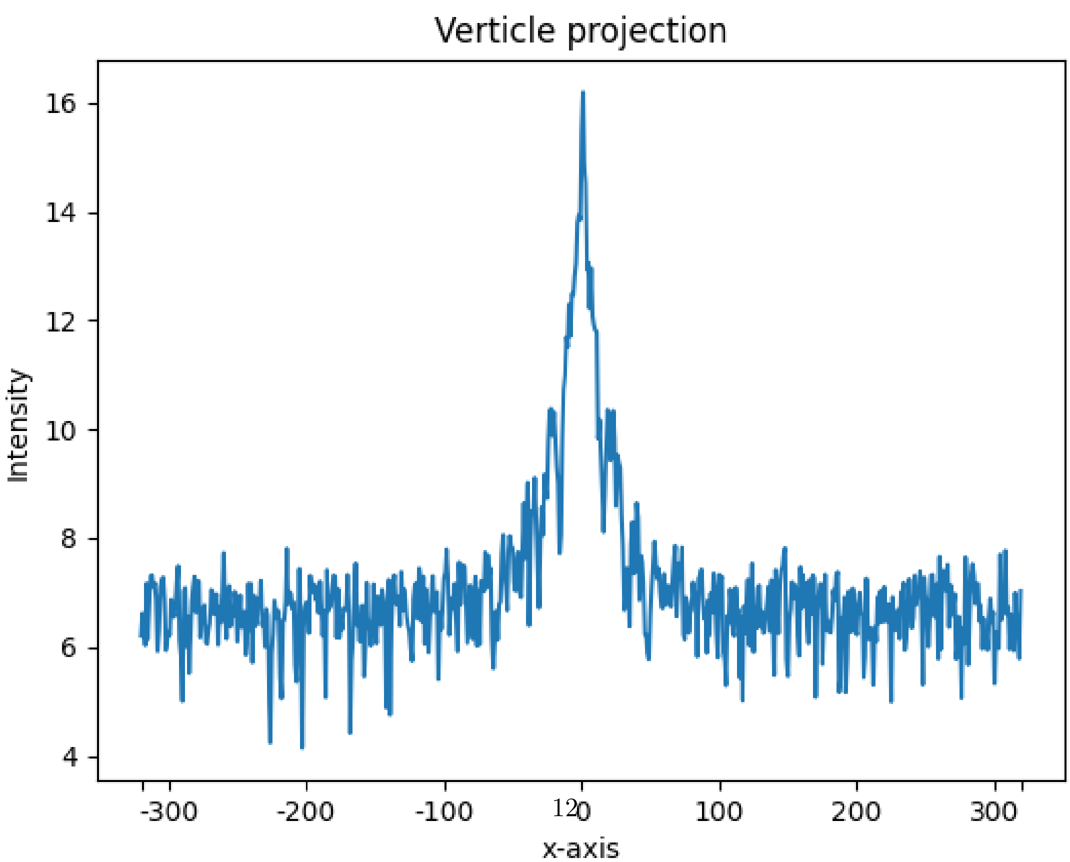






Figure 11: Radon Transformed image

where $H$ is the Fourier transform of the point spread function, $S_n$ is the noise power spectrum, and $S_f$ is the signal power spectrum.
We here take $K = 0.004$, and we do not apply ‘fftshift’ during the filtering.


Figure 12. p4c