Image Restoration – A Use Case

Thai Quang Phat

Supervisor: Prof. Nguyen An Khuong, PhD.Candidate Nguyen Hoang Minh

November 29, 2024

Outline

- Problem
- Comments
- Procedure
- Experimental Results
- Conclusion
- Limitations & Future Work

Problem

Problem

Problem Statement

Given a blurred image of a heart, the task is to recover the degraded function. It is known that, the bottom right corner crosshair image before degraded, is 3 pixels wide, 30 pixels long, and had an intensity of 255. Provide a step-by-step procedure indicating how you would use the information just given to obtain the blurring function $H(u,v)$.

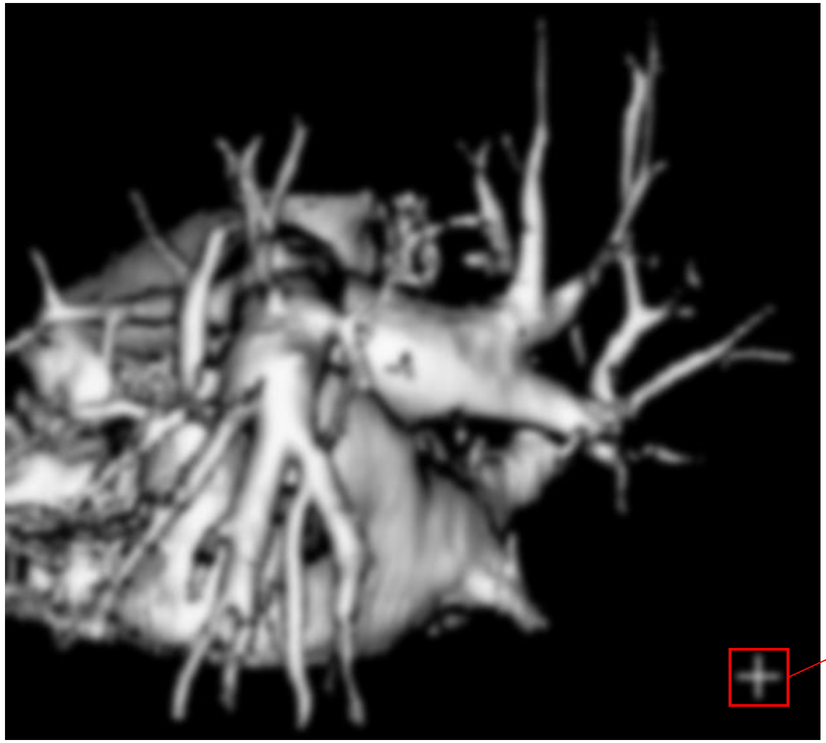


Figure. Blurred heart

(Original image courtesy of GE Medical Systems.)

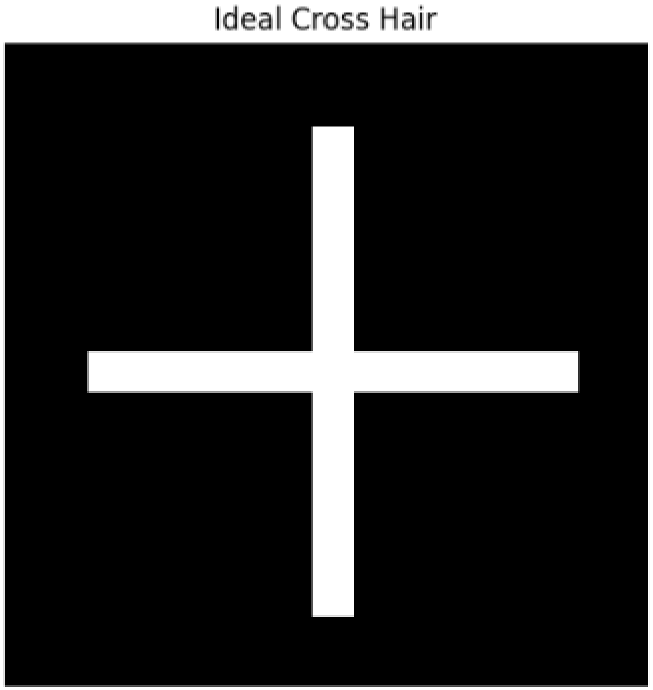


Figure. Crosshair before blur

Assumptions

Upon solving the problem, we consider the following assumption

1. **No knowledge of the clear heart image:** We will recover the degraded image as to verify the correctness of the degraded function, with not having the original **clear** heart image, thus at the restored result, we accept the result to our understanding.

Approach Problem

We divide our approach into three main sections

Give some comments from observation and experimental

Calculate the estimated degraded function based on comments above

Re apply the degraded function to obtain the recovered image, and give conclusions about the correctness of the obtained function

Comments

Comments

Frequency Domain Encountered Comments

As far as we consider, blurring caused in the image is mostly affected in the frequency domain. With this comment, we approach to fully exploit the Fourier transform.

No Noise Corrupted

We observe the histogram of a segment of the original image.

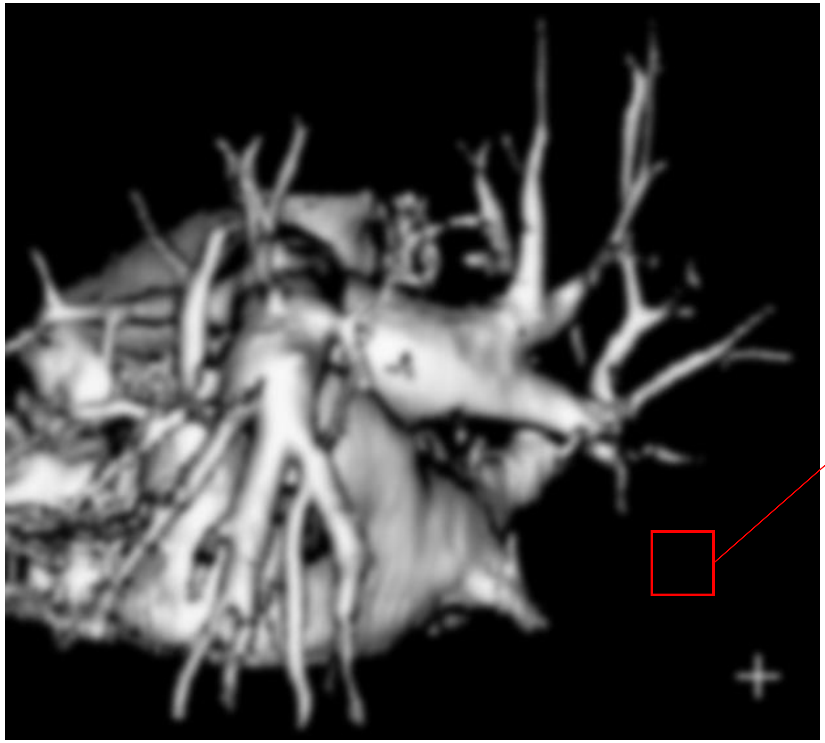


Figure. Input heart image.

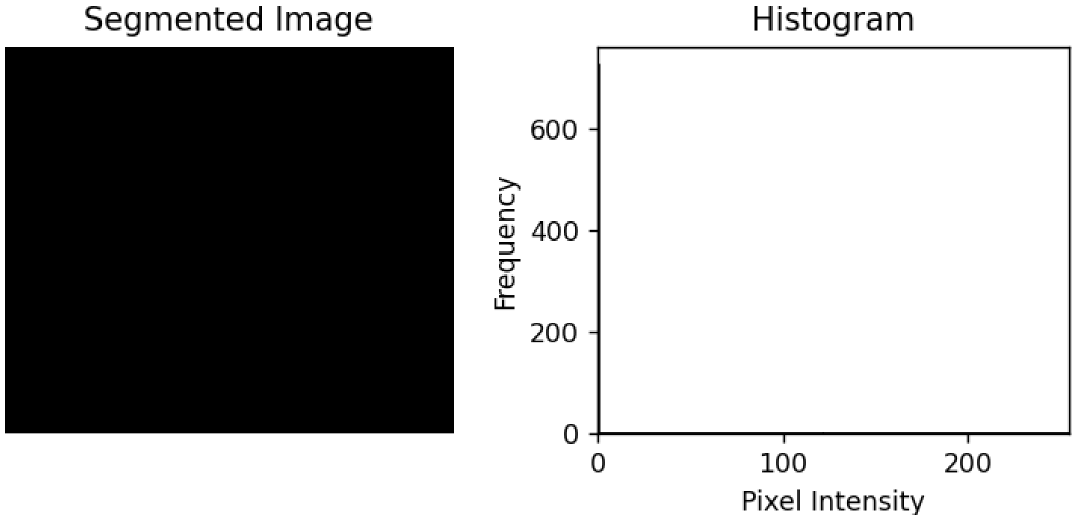


Figure. Histogram of segmented image.

Noise typically affects high-frequency components. Blurring the image attenuates these components, resulting in a smoother appearance. It can be seen in the histogram that, whether the original image have noise or not, after blurring, the appearance of noise is absence. In the absence of specific information about noise, it is reasonable to assume the image is not corrupted by noise.

Gaussian Filter Comments

We first denote some variables

- F_{blurred}: the Fourier transform of the blurred cropped crosshair image.
- F_{ideal}: the Fourier transform of the ideal crosshair image.
- H: the Fourier transform of the degraded function.

An approach to obtain H is by the following equation

$$F_{\text{blurred}} = F_{\text{ideal}} \cdot H \Leftrightarrow H = \frac{F_{\text{blurred}}}{F_{\text{ideal}}}$$

This technique is called the inverse filtering

Comments

In most cases, we cannot obtain directly H using the inverse filtering, a division is normally applied to a real number. Recall a very common action, let a, b be two complex numbers, then we have

$$\frac{a}{b} = \frac{a \cdot \overline{b}}{|b|^2}$$

This motivates us to come up with the following approximation.

$$H \approx \frac{F_{\text{blurred}} \cdot \overline{F_{\text{ideal}}}}{\left|F_{\text{ideal}}\right|^2 + \varepsilon}$$

Where ε is the estimated constant to make sure the denominator is non-zero. This is known as the Wiener filtering.

Gaussian Filter Comments

Using this approximation, we can obtain the Fourier spectrum of the degraded function as below.

$$H \approx \frac{F_{\text{blurred}} \cdot \overline{F_{\text{ideal}}}}{\left|F_{\text{ideal}}\right|^2 + \varepsilon}$$

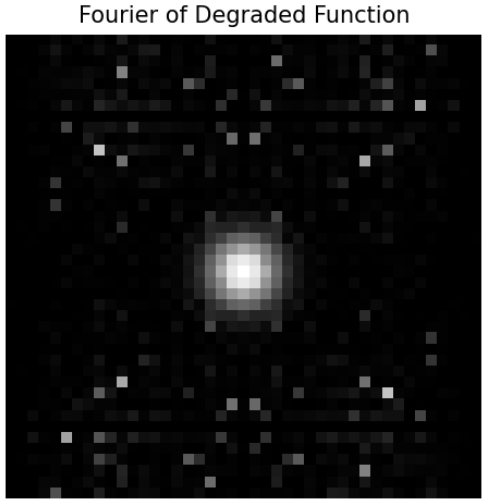


Figure. The Fourier spectrum of the degraded function.

Gaussian Filter Comments

Let’s ignore the white dots, and focus on the white circle in the middle. This image resembles a lot of the Gaussian filter, which has the formula

$$G(u,v) = e^{\frac{-D(u,v)^2}{2D_0{}^2}}$$

Where D_0 is the cutoff frequency, $D(u,v)$ is the distance from each pixel to the center of the image, in this case the center of the Fourier Transformed image.

Here is the visualization plot of the Gaussian filter.

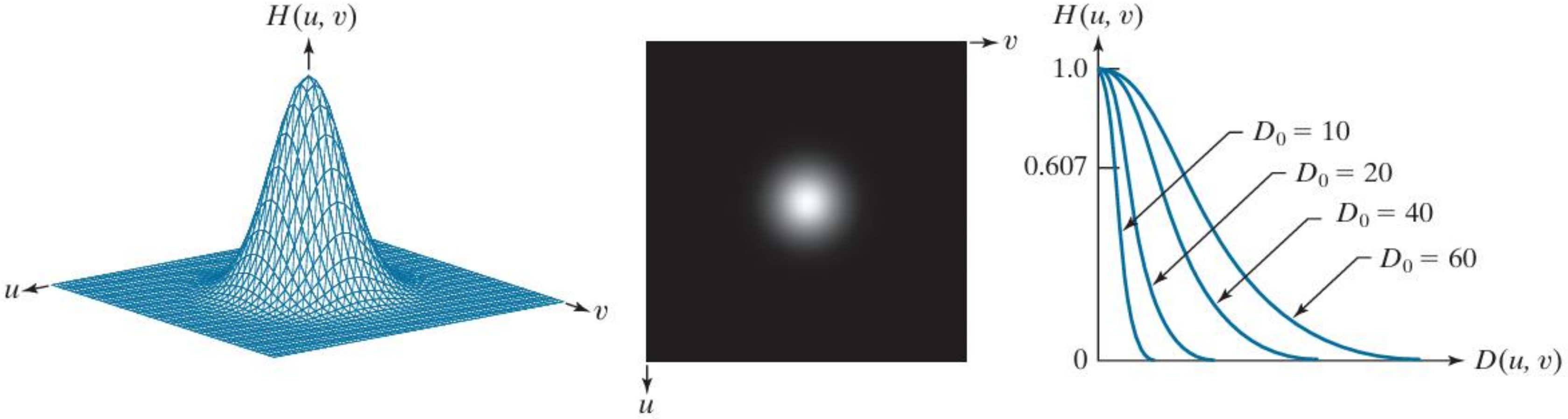


Figure. The Gaussian Lowpass Filter visualization and the cutoff frequency

Gaussian Filter Comments

Take a comparison of the Gaussian filter with cutoff frequency $D_0 = 2.227$ and the obtained degraded function.

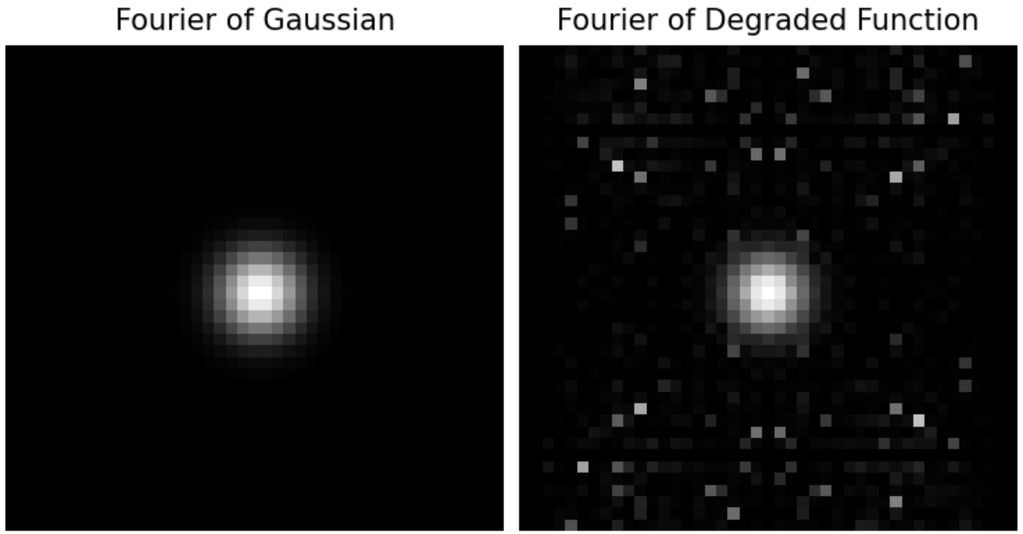


Figure. The Fourier spectrum of the Gaussian filter compared to the degraded function

This makes it reasonable for us to approximate our degraded function with the Gaussian filter. We will explain how we got the value 2.227 in the procedure below.

Summary Comments

We have a short summary of information before getting into the main procedure:

1. The image is not corrupted by noise.
2. The blurred is caused in the frequency domain.
3. The estimated degraded function is a Gaussian filter function.

Procedure

Procedure

Blurred Crosshair Cropped

This step is an important step, as it is the only information of the image that we have. We attempt by cropping out the blurred crosshair.

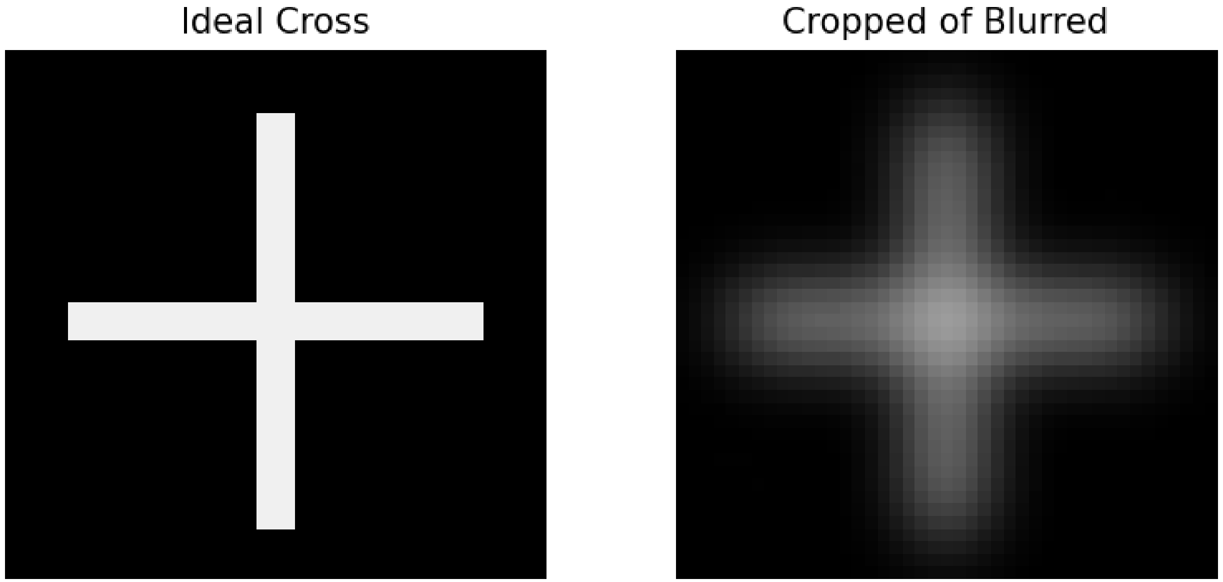


Figure. The cropped image compared to its original

Fourier Transformed Procedure

Since we are working on the frequency domain, it is crucial to apply Fourier transform before applying any filters.

Let f be any input image size M x N, we denote F as its Fourier transform, which is obtained by

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-i2\pi(ux/M + vy/N)}$$

Fourier transform allows us to convert from the spatial domain, to the frequency domain. We do not discuss about why in this particular presentation.

Estimated Gaussian Filter Procedure

Previously we have stated that we can approximate the degraded function with the Gaussian filter function, which means, we want to have

$$H(u,v) \approx e^{\frac{-D(u,v)^2}{2D_0{}^2}}$$

This reduces the problem to find D_0 that fits best our assumption.

Estimated Gaussian Filter Procedure

Recall that we have the following equivalent

$$H(u,v) \approx e^{\frac{-D(u,v)^2}{2D_0{}^2}} \Rightarrow D_0 \approx \sqrt{\frac{-D(u,v)^2}{2\ln H(u,v)}}$$

To obtain the estimated cutoff frequency, the simplest approach is to take the average

$$\overline{D_0} \approx \frac{1}{M \cdot N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \sqrt{\frac{-D(u,v)^2}{2\ln H(u,v)}}$$

Where *M*, *N* are the sizes of the input crosshair blurred image.

Estimated Gaussian Filter Procedure

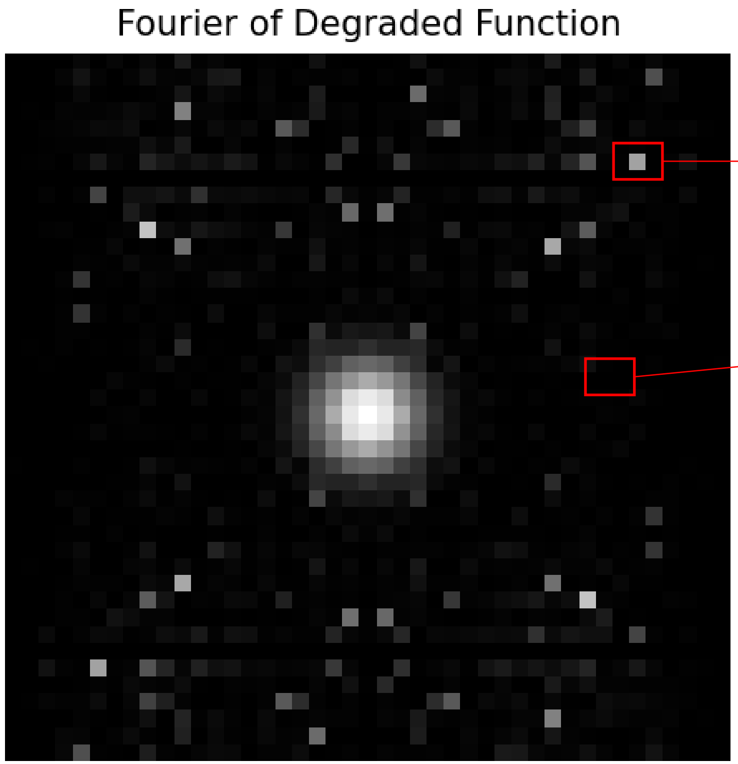


Figure. The Fourier spectrum of the degraded function.

Some problem arises if we apply the estimation algorithm above.

1. **Corrupted noise.** The white dots outside the Gaussian shape appeared as we don't have knowledge about noise. Including these dots in the calculation may interrupt the estimation.
2. **Natural logarithm of zero.** If $H(u, v)$ equals to zero, we can't attempt to take the natural logarithm of it (since the domain of natural logarithm are positive real numbers).

This encourages us to develop another method for estimating D_0.

Estimated Gaussian Filter Procedure

Notice that the Gaussian shape only occurs around the center of the image. And we can estimate the cutoff frequency by using only a 1-D cut of the image, this motivates us to selects the diameter segment of the image, which is the middle row and column of the image.

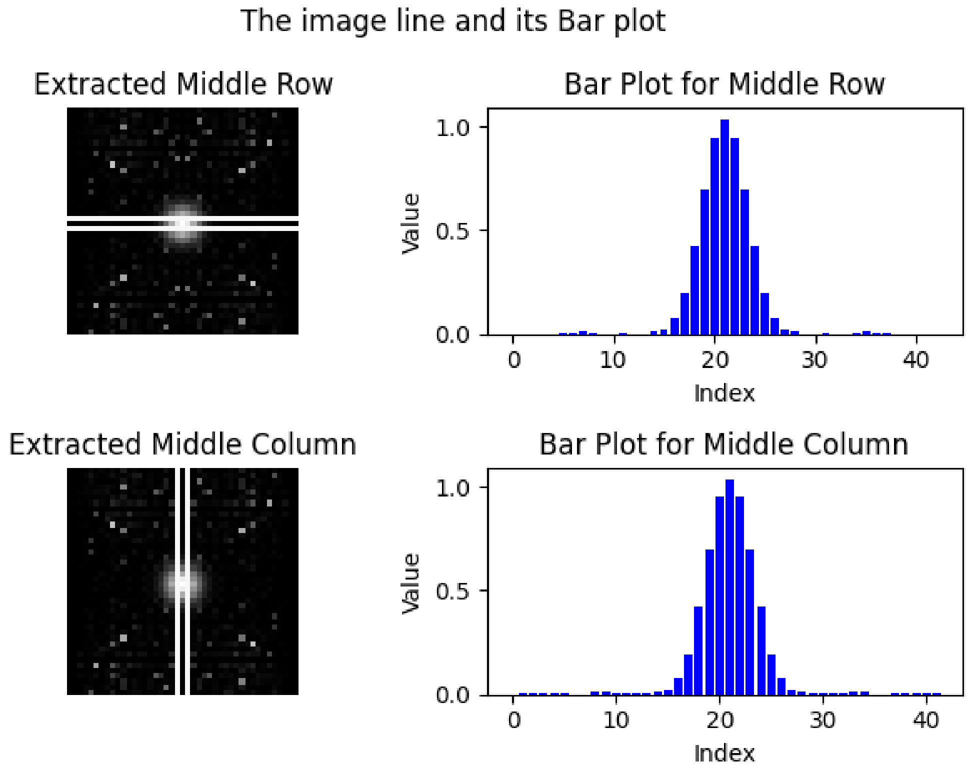


Figure. The middle lines and its bar plot

Estimated Gaussian Filter

This image once again proofs our statement that the estimated degraded function can be a Gaussian function.

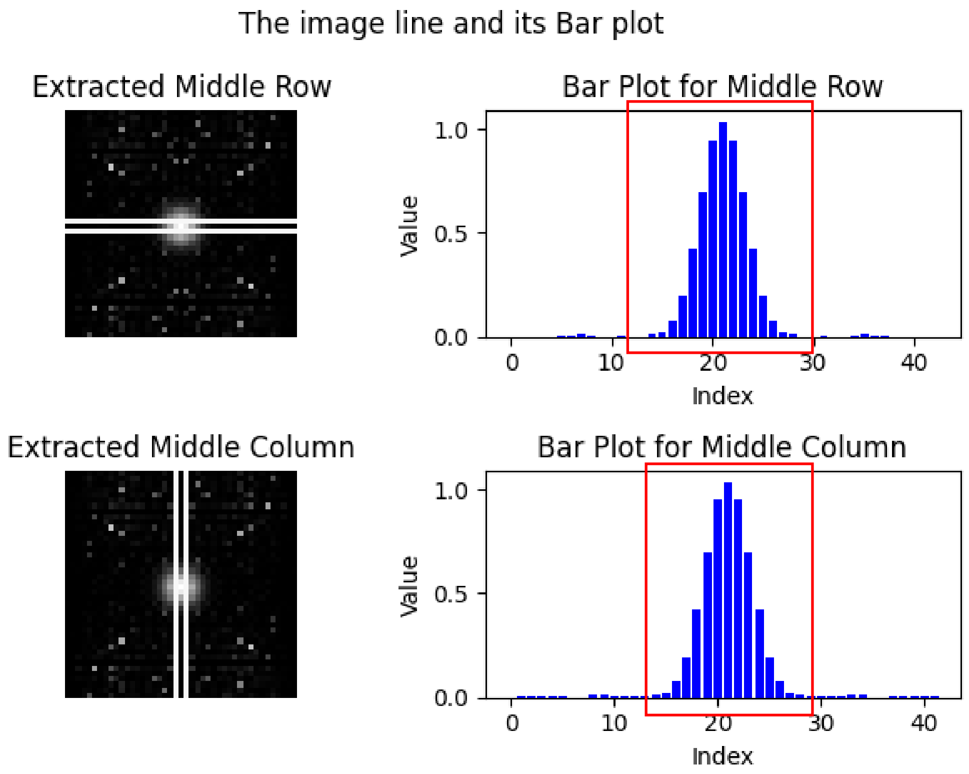


Figure. The middle lines and its bar plot

We only want to use the "Gaussian shape" of the image. Thus this will only be the center area of the bar plot. Other values can be treated as noise.

Specifically, in the middle row, we are using values with index from 12 to 30, and for the middle column, we are using values with index from 14 to 28.

Estimated Gaussian Filter Procedure

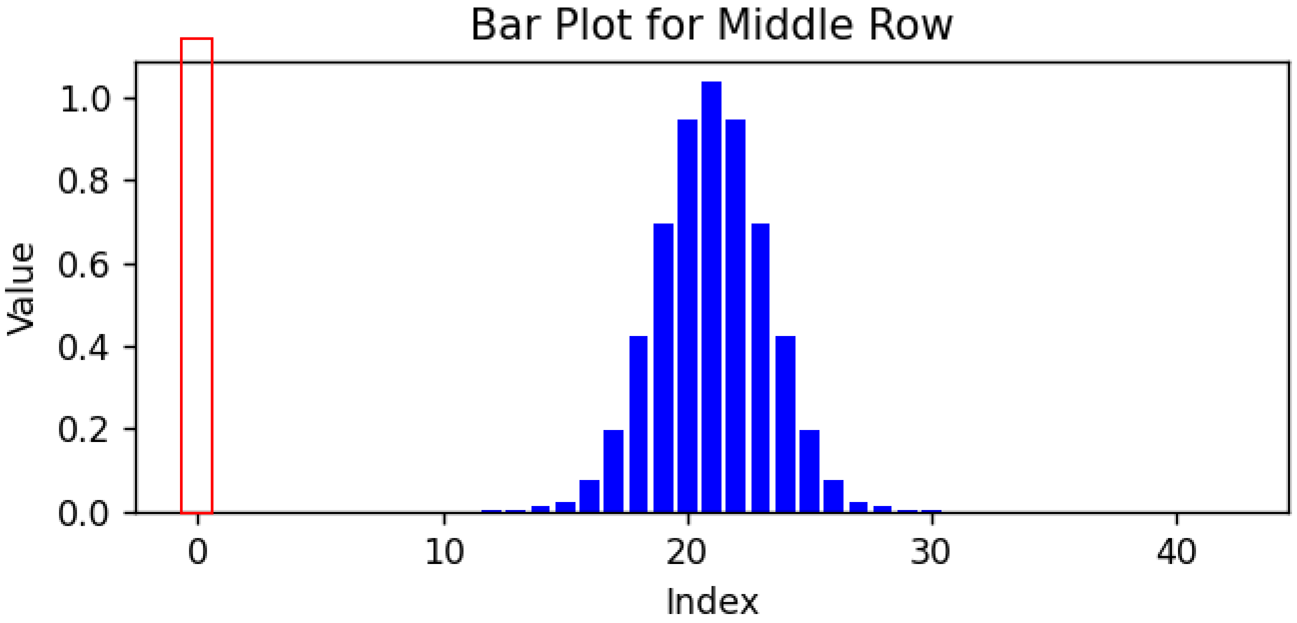
Using comments above, we came up with the following method

Let $H(u, v)$ be the estimated degraded function, $D_0(u, v)$ be the estimated cutoff frequency at index (u, v), D_0 is the final cutoff frequency (initialize as 0).

1. Extract the list obtained from the middle row and the middle column of the degraded function image, let it be $H(u, v)$.
2. If $H(u, v)$ equals to zero, then move to the next index.
3. Otherwise, calculate $D_0(u, v)$ using the formula previously.
4. Add $D_0(u, v)$ to D_0.
5. If there are still indexes not calculate, go back to step 2.
6. Obtain D_0 by dividing with the number of non zero values added.

Estimated Gaussian Filter Procedure

Let’s do an example with the middle row of how we sequently increase D_0. The procedure for the middle column is the similar. The center in this case is at index 21.

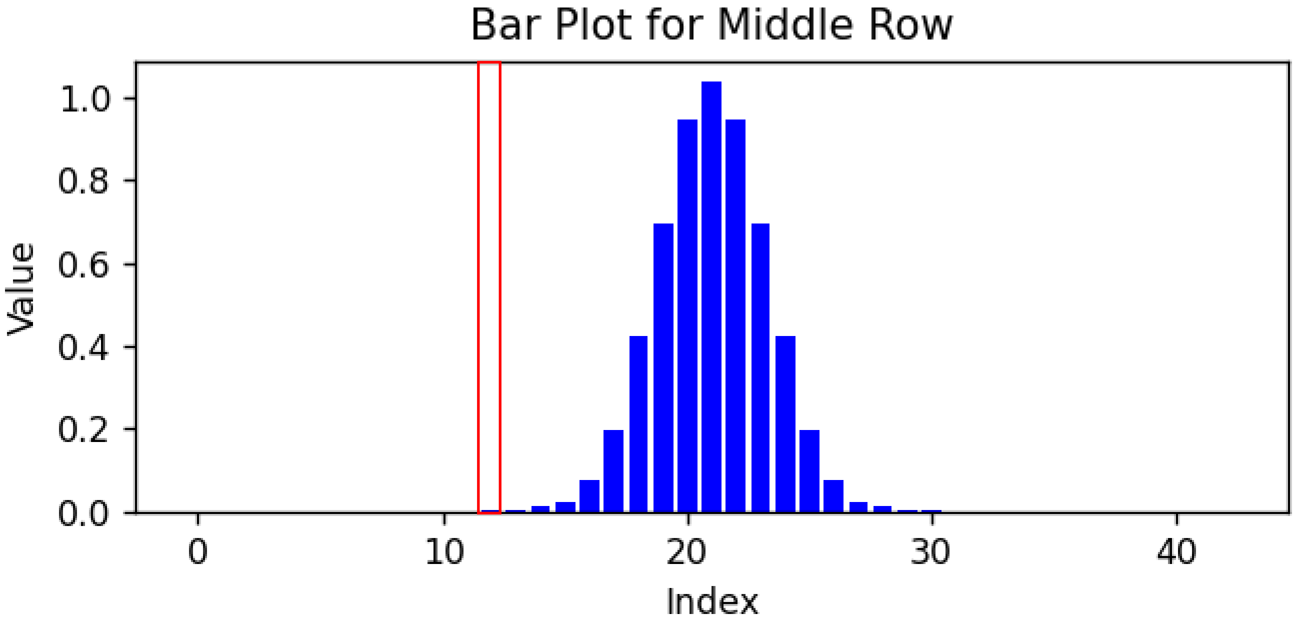


- Index: 0
- $H(u, v)$ value: 0

$\Rightarrow$ Skip this index

Estimated Gaussian Filter Procedure

It is reasonable for us to skip forward to the 12th index, since others are zero.



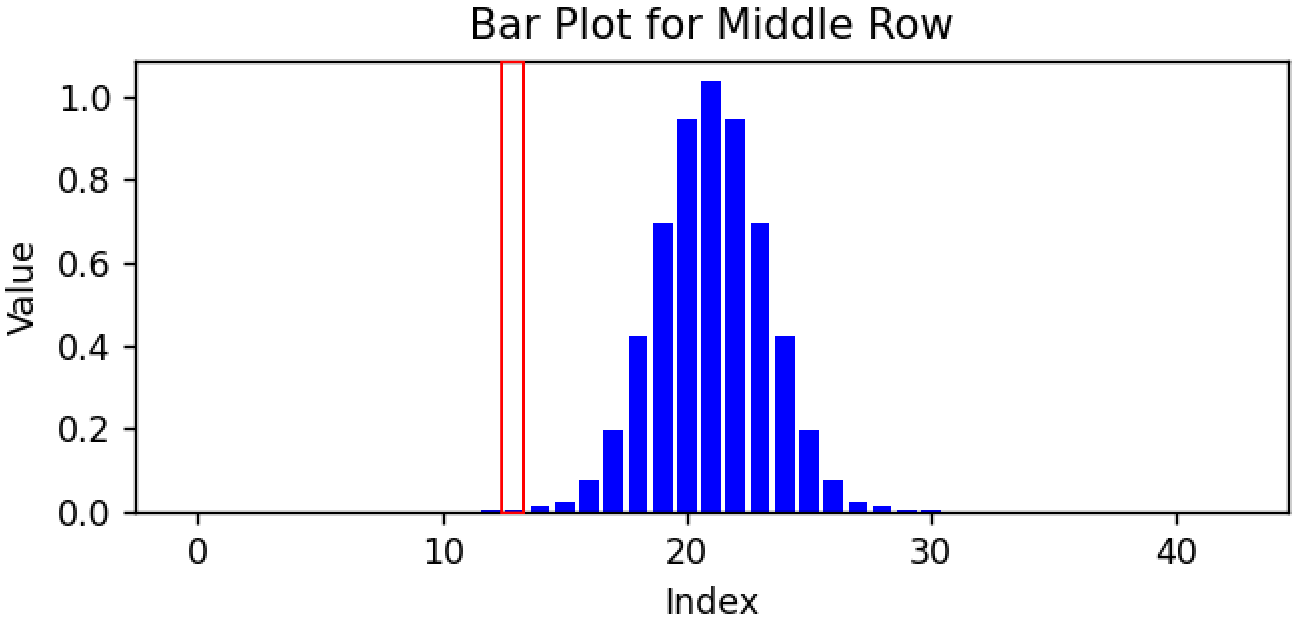
- Index: 12
- $H(u, v)$ value: 0.00085
- $D^2(u, v)$: 81
- $D_0(u, v)$: 2.393
- D_0: 2.393

$$D_0 \approx \sqrt{\frac{-D(u,v)^2}{2\ln H(u,v)}}$$

$\Rightarrow$ Update D_0 successfully.

Estimated Gaussian Filter Procedure

Here we show how to process at the 13th index, which is where there is an update in D_0.



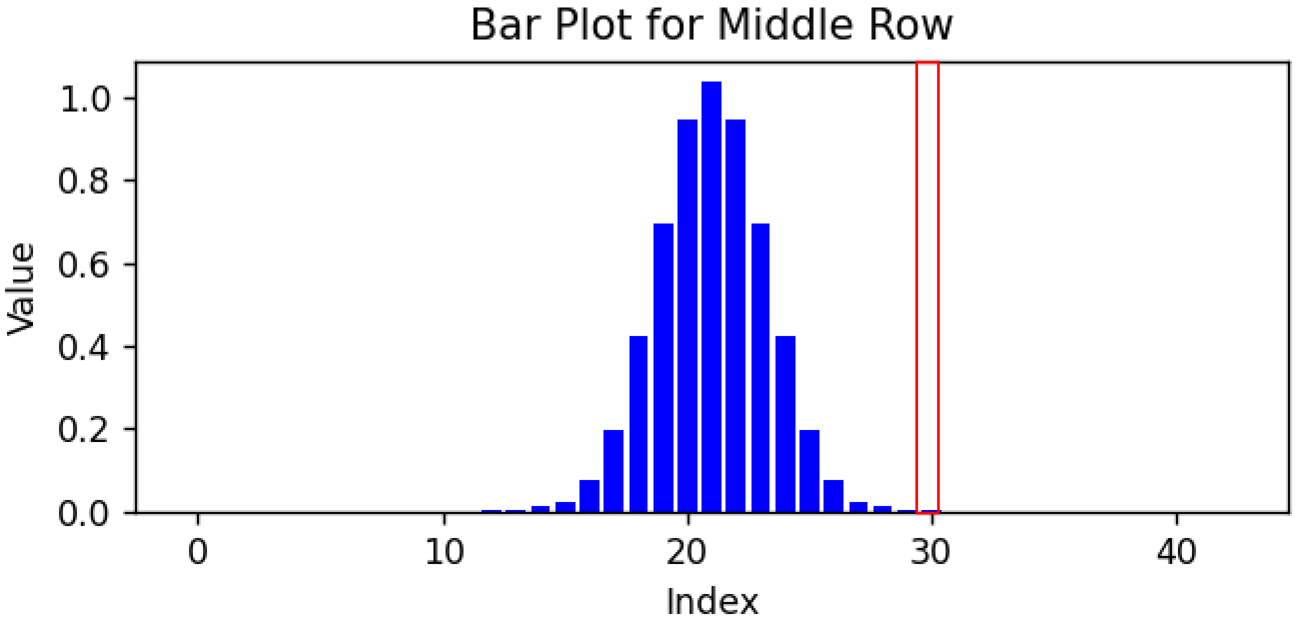
- Index: 13
- $H(u, v)$ value: 0.00087
- $D^2(u, v)$: 64
- $D_0(u, v)$: 2.131
- D_0: 4.524

$\Rightarrow$ Update D_0 successfully.

$$D_0 \approx \sqrt{\frac{-D(u,v)^2}{2\ln H(u,v)}}$$

Estimated Gaussian Filter Procedure

At the last index 30, we have successfully update D_0 from the middle row.



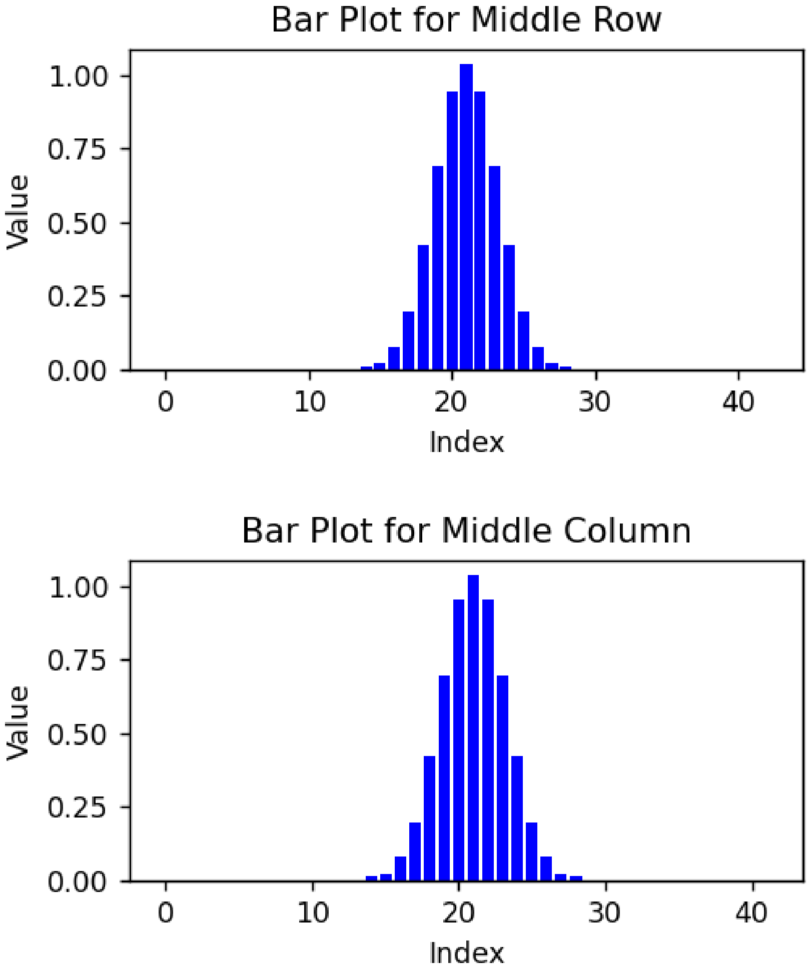
- Index: 30
- $H(u, v)$ value: 0.00085
- $D^2(u, v)$: 81
- $D_0(u, v)$: 2.393
- D_0: 45.162

$$D_0 \approx \sqrt{\frac{-D(u,v)^2}{2 \ln H(u,v)}}$$

$\Rightarrow$ Update D_0 successfully.

Estimated Gaussian Filter Procedure

After processing for the middle and the middle column, we can finally estimate D_0 as



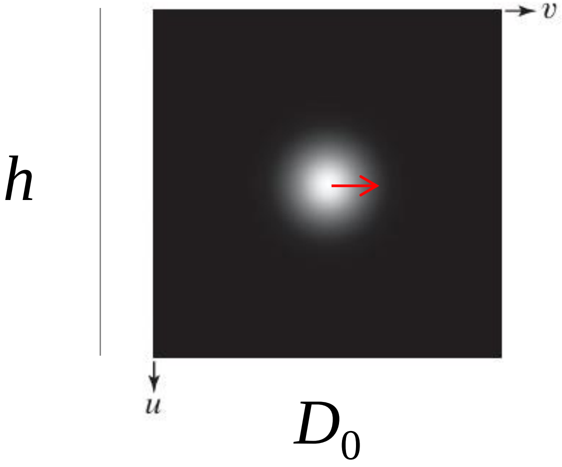
- D_0: 75.74
- Number of non-zero values: 34
- Final estimated D_0:

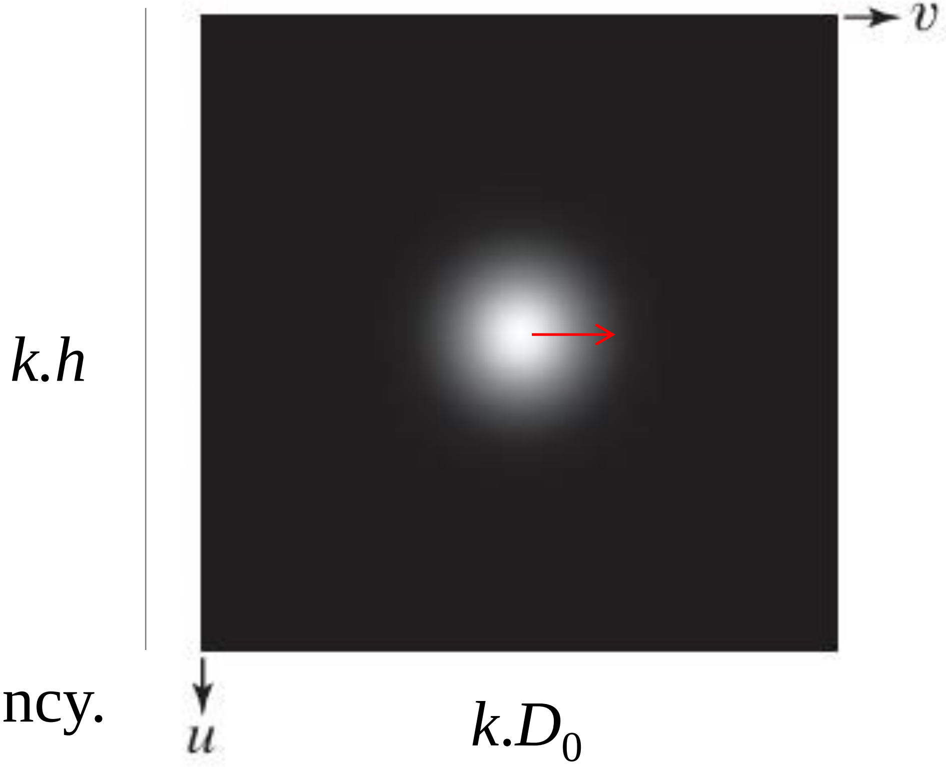
$$D_0 = \frac{D_0}{\#\_non\_zero} = \frac{75.74}{34} \approx 2.227$$

This demonstration shows how we obtain the value 2.227 mentioned above.

Scale for Input Image

We take advantage of the cutoff frequency mentioned earlier, for a larger scale image, we would like to scale the cutoff frequency approximately as the ratio between the large and cropped image.





This encourage us to just scale the cutoff frequency.

Scale for Input Image

Thus, the estimated *H’* Gaussian filter for the input heart image will have the following formula

$$H(u,v) \approx e^{\frac{-D(u,v)^2}{2D_0{}^2}} \xrightarrow{\text{Scaling}} H'(u,v) \approx e^{\frac{-D(u,v)^2}{2(kD_0)^2}}$$

Restored Image Procedure

This phase is not necessary for the problem itself, but it is a way to verify whether the obtained function is acceptable or not.

Restored Image

Having the estimated *H’*, obtained from scaling the cutoff frequency D_0 from *H*, we can obtain the restored image, using the Wiener filtering

$$F_{\text{restored}} = \frac{F_{\text{input}} \cdot \bar{H}'}{|H'|^2 + K}$$

Where F_{input}, F_{restored} respectively be the Fourier transform of the input and the restored heart image, *K* will be a constant being estimated.

Convert to Spatial Domain

Lastly, we use the inverse Fourier transform to bring the image back from the frequency domain to its spatial domain.

$$f_{\text{restored}}(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{n=0}^{N-1}F_{\text{restored}}(u,v)e^{i2\pi(ux/M+vy/N)}$$

Post Processing Procedure

We introduce two extra steps in enhancing the image quality: brightening and sharpening.

For brightening, we use thresholding, a technique which selects pixels at a certain level of intensity and add to its intensity.

$$f_{\text{brighten}} = \begin{cases} f_{\text{restored}} + \alpha & \text{if } \; f_{\text{restored}} \geq t \\ f_{\text{restored}} & \text{otherwise} \end{cases}$$

Where α is the added intensity, t is the threshold level.

Post Processing Procedure

For sharpening, we opt to use the Laplacian sharpening kernel, and apply convolution to the restored image

$$L = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Apply convolution to obtain the final sharpened image

$$f_{\text{sharpened}} = f_{\text{restored}} \star L$$

Here * denotes convolution.

Summary Procedure

We have a short summary of the main procedure:

1. Crop out the blurred crosshair, as well as creating the ideal crosshair image.
2. Get the Fourier transform of images involved.
3. Use averaging to estimate the degraded function by Gaussian filter function.
4. Obtain the scaled degraded function for the heart image.
5. Use Wiener filtering to obtain the obtain Fourier transformed image.
6. Apply inverse Fourier transform to bring the image back to spatial domain and observe its result.
7. (Optional). Brightening and sharpening the recovered image for visualization.

Experimental Results

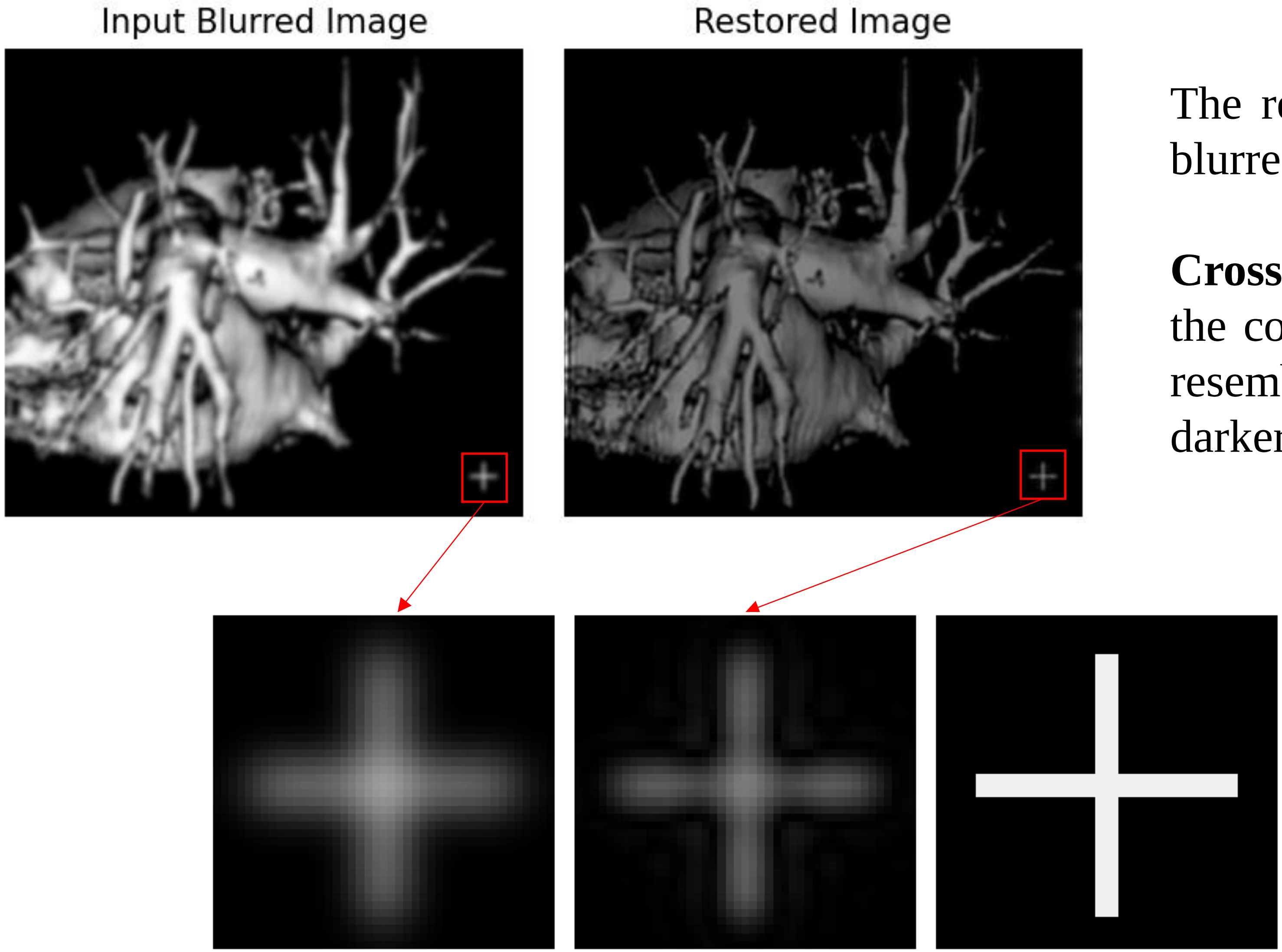
Experimental Results

Parameter Settings

By explanations above, combine with some experiment, we show below the parameters for the problem.

Parameter	Meaning	Value
D_0	Cutoff frequency for the blurred crosshair	2.227
ε	The estimate constant for finding the degraded function	0.0001
K	The estimate constant for Wiener filter	0.005
k	The scaling coefficient	13.404
α	The added intensity	20.000
t	The threshold level	5.000

Crosshair Comparison

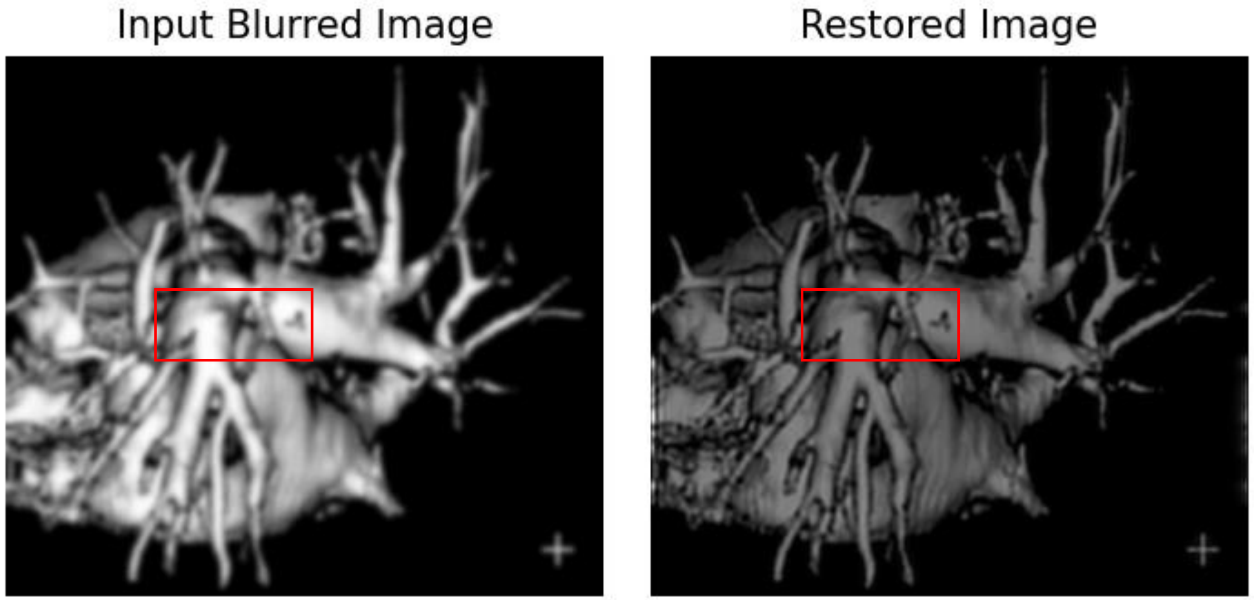


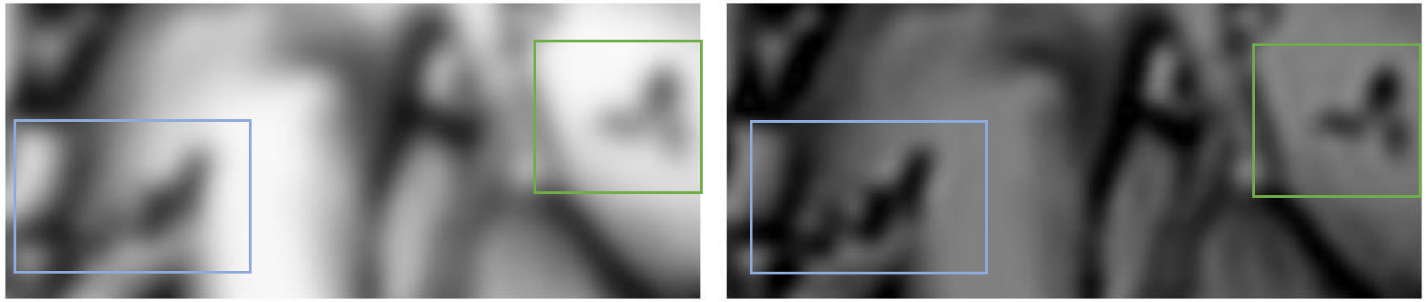
The restored image is said to be better than the blurred image, here are the reasons:

Crosshair restored: Observing the crosshair at the corner, it is seen that the recovered crosshair resembles more of the ideal crosshair (although darker, this is a result of the Wiener filter).

Figure. The blurred, restored and ideal crosshair image

Crack Comparison





Information gained: take a look at the areas which appears to be a crack in the heart. It is seen in the blurred image, the shape of the triangle crack (in the green rectangle) is not clear, whether in the restored image, we can clearly see the shape of the crack, which resembles a triangle.

Another similar area lies in the blue rectangles. In the blurred image, we barely see the shape of the crack. But in the restored image, we can see details of it.

Enhancement

For visualization purpose, we provide two resulted image using brightening and sharpening.

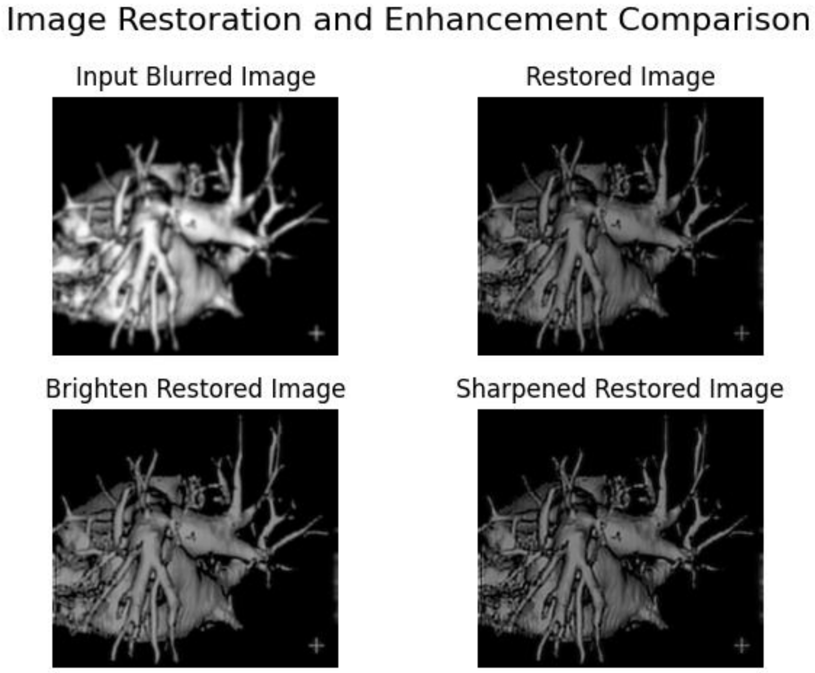


Figure. Comparison of four images

Since these two images are optional steps to enhance the image quality, we only care about the difference between them and the restored image, other conclusions are similar to ones made for the restored image and the blur image.

Crack Comparison

We provide a side-by-side comparison image of the input image and three results, and observe its improvement.

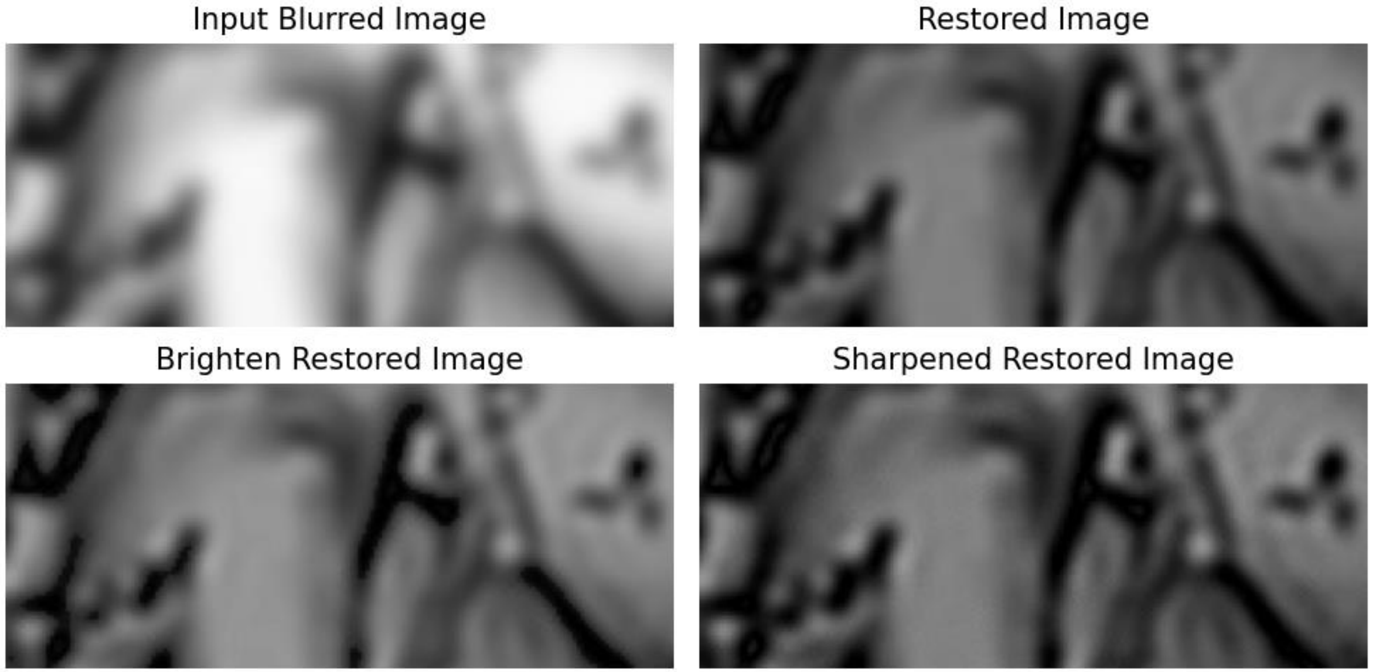


Figure. Crack areas in four images

With the brightened image, the cracks are shown more easier. We notice in the sharpened image, the details shows a more concise shape of how it suppose to be.

Experimental Results

We have a short summary of the image results:

1. The restored image have successfully restored the crosshair image at its best.
2. The restored heart image gives more information than the original blurred image. Shapes are more accurate.
3. The brightened image gives a brighter version of the restored image, while the sharpened image gives more sharp details, making the image more easily for visualization.

Conclusion

Conclusion

Conclusion

Throughout solving the problem, we could see that

- The estimated Gaussian function is good enough to predict the degraded function of the input heart image.
- Different cutoff radius is applied for different image size.
- Enhancement of an image does not give more information as it already shows. Such work is done for better visualization.

Limitations & Future Work

Limitations & Future Work

Limitations

1. **Limited filters:** Gaussian filter is a good filters to estimate the original image in our scenario, there might be other filters which may give better results.

Future Work

1. **More filters:** explore other filters rather than Gaussian.
2. **Post processing methods:** as to enhance the image quality, better post processing methods can be encountered.

Reference

1. Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing, Fourth Edition*. Pearson, 2018.