VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Digital Image Processing and Computer Vision (CO3057)

Fourier Transform for Image Restoration in Frequency Domain A Mathematical Approach

Advisor: Nguyen Duc Dung
Student: Thai Quang Phat - 2252606

HO CHI MINH CITY, NOV 2025

Fourier Transform for Image Restoration in Frequency Domain: A Mathematical Approach

Thai Quang Phat

Update on November 17, 2025

Contents

Preface

0.1 Declaration

I certify that this study[1] "Fourier Transform: An Overview", is my study under the supervision of Dr. Nguyen Duc Dung derived from practical needs and my desire to study.

0.2 Acknowledgements

I would like to express our sincere appreciation for Dr. Nguyen Duc Dung, my report supervisors. Under their guidance, I have learned the background knowledge and be able to complete this project.

0.3 Abstract

The Fourier Transform is a critical tool in digital image processing, particularly for enhancing and restoring images affected by blur. Its ability to convert spatial domain information into the frequency domain makes it invaluable for isolating and addressing specific frequency components that contribute to image degradation. In this study, I explored an overview of various applications that utilize the Fourier Transform, emphasizing its effectiveness in analyzing, filtering, and restoring images by targeting and reducing blurring effects. Through these applications, the Fourier Transform demonstrates its versatile role in improving image quality and facilitating more accurate digital imaging solutions.

Keywords: Fourier Transform, digital image processing, filtering, restoration.

[1] Code implementation: https://github.com/thaiquangphat/fourier_transform_for_image_restoration

Chapter 1

Background & Related Works

1.1 Background

Image restoration in the frequency domain represents a fundamental paradigm in digital image processing, aimed at recovering an original image from its degraded observation. The degradation process can be mathematically modeled in the spatial domain as a convolution operation between the original image and a point spread function (PSF), accompanied by additive noise. When transformed to the frequency domain via the Fourier transform, this convolution becomes a simple multiplication, significantly simplifying the restoration problem and enabling efficient computational implementations [1].

The discrete Fourier transform (DFT) serves as the cornerstone of frequency domain image processing. For a digital image $f(x,y)$ of size $M \times N$, the two-dimensional DFT is defined as:

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M+vy/N)}$$

where $u = 0,1,2,\ldots,M-1$ and $v = 0,1,2,\ldots,N-1$. The inverse DFT reconstructs the spatial domain image:

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

The image degradation model in the frequency domain can be expressed as:

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

where $G(u,v)$ represents the degraded image, $H(u,v)$ is the degradation function (optical transfer function), $F(u,v)$ is the original image, and $N(u,v)$ denotes the additive noise, all in the frequency domain. This multiplicative relationship in the frequency domain, contrasted with the convolution in the spatial domain, provides significant advantages for both analysis and computational efficiency, as it can be implemented using the Fast Fourier Transform (FFT) algorithm with $O(N \log N)$ complexity.

The magnitude spectrum of the Fourier transform contains most of the geometric information about the spatial structure of the image, while the phase spectrum is crucial for accurate reconstruction. The DC component $F(0,0)$ represents the image mean, and frequency components further from the origin correspond to higher spatial frequencies, typically associated with edges and fine details in the image.

1.2 Related Work

1.2.1 Inverse Filtering

The simplest approach to image restoration in the frequency domain is direct inverse filtering. Given the degradation model, an estimate of the original image can be obtained by:

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

While conceptually straightforward, inverse filtering suffers from severe limitations. The method is extremely sensitive to noise, particularly at frequencies where $H(u,v)$ is small or approaches zero, leading to catastrophic noise amplification [1]. Additionally, when $H(u,v) = 0$, the filter becomes undefined, making the restoration problem ill-posed. Despite these limitations, inverse filtering provides the theoretical foundation for more sophisticated restoration techniques.

1.2.2 Wiener Filtering

The Wiener filter represents a landmark advancement in image restoration, providing an optimal solution in the minimum mean square error (MMSE) sense. Developed by Norbert Wiener in the 1940s and published in 1949, the filter executes an optimal trade-off between inverse filtering and noise smoothing by incorporating statistical knowledge of both the signal and noise processes [2].

The Wiener filter in the frequency domain is expressed as:

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + S_n(u,v)/S_f(u,v)} \right] G(u,v)$$

where $H^*(u,v)$ is the complex conjugate of the degradation function, $S_n(u,v)$ represents the power spectrum of the noise, and $S_f(u,v)$ denotes the power spectrum of the original image. The term $S_n(u,v)/S_f(u,v)$ can be interpreted as the reciprocal of the signal-to-noise ratio.

In practical applications where the exact power spectra are unknown, a simplified parametric Wiener filter is commonly employed:

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + K} \right] G(u,v)$$

where K is a constant parameter representing the noise-to-signal power ratio (NSR). This parameter can be determined experimentally or adaptively based on local image characteristics. Where the signal is very strong relative to the noise ($K \to 0$), the Wiener filter approaches the inverse filter, while in regions of weak signal, the filter provides smoothing to suppress noise amplification.

Kutay and Ozaktas extended the classical Wiener filtering framework to fractional Fourier domains, demonstrating that optimal filtering in fractional Fourier domains can achieve error reduction compared to ordinary Fourier domain Wiener filtering for certain types of degradation, particularly chirp-like degradations and non-constant velocity motion blur, while maintaining $O(N \log N)$ computational complexity [3].

Hillery and Chin proposed iterative Wiener filtering methods that refine the power spectrum estimates between iterations, achieving improved restoration quality over single-pass implementations [4]. Various adaptive extensions of the Wiener filter have been developed, including frequency-adaptive forms where K becomes a function of frequency $K(u,v)$, and spatially adaptive forms where K varies locally as $K(x,y)$.

1.2.3 Constrained Least Squares Filtering

Constrained least squares (CLS) restoration addresses the limitation that exact noise characteristics required by Wiener filtering are often unavailable in practice. Hunt pioneered this approach in 1973, formulating the restoration problem as an optimization task that seeks a solution balancing fidelity to the observed data with smoothness constraints on the restored image [5].

The CLS method minimizes the following objective function using Lagrange multipliers:

$$J = \|C\hat{f}\|^2 + \lambda \|g - H\hat{f}\|^2$$

where C represents a high-pass filter operator (typically the discrete Laplacian), λ is the regularization parameter controlling the trade-off between smoothness and fidelity, $\hat{f}$ is the estimated image, g is the degraded image, and H is the degradation operator.
The frequency domain solution for CLS filtering is given by:

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \lambda |P(u,v)|^2} \right] G(u,v)$$

where $P(u,v)$ is the Fourier transform of the Laplacian operator. The typical choice for the discrete Laplacian is:

$$c(n_1,n_2) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

The regularization parameter λ plays a crucial role in determining restoration quality. Reddi improved the computational efficiency of Hunt's original formulation, demonstrating that constrained least squares solutions can be computed iteratively with significantly reduced computational complexity [6]. While analytical approaches exist for estimating λ, it is typically considered a user-tunable parameter [7]. The CLS filter implementation in the frequency domain using block-circulant matrices and FFT achieves computational efficiency comparable to Wiener filtering.

Iterative forms of CLS, known as the Tikhonov-Miller method, provide improved convergence properties for certain classes of degradation problems. Katsaggelos and colleagues developed adaptive regularization techniques where λ varies spatially to accommodate local image characteristics, ensuring tighter regularization in smooth regions while allowing greater flexibility near edges [8]. Matakos et al. extended CLS methods to handle uncertain point spread functions through regularized constrained total least-squares (RCTLS) approaches, which account for errors in both the degraded image and the degradation operator [9].

1.2.4 Richardson-Lucy Deconvolution

The Richardson-Lucy algorithm represents an iterative, non-linear approach to image restoration based on maximum likelihood estimation under Poisson noise statistics. Independently proposed by Richardson [10] and Lucy [11] in the early 1970s, this method has become particularly valuable for applications in astronomical imaging, microscopy, and other domains where photon-counting noise dominates.

The Richardson-Lucy iterative formula is expressed as:

$$\hat{f}^{(k+1)} = \hat{f}^{(k)} \left[\frac{g}{h * \hat{f}^{(k)}} * h^* \right]$$

where $\hat{f}^{(k)}$ represents the estimate at iteration k, g is the observed degraded image, h is the PSF, h^* denotes the mirrored PSF (or equivalently, the inverse Fourier transform of the complex conjugate of the optical transfer function), and $*$ indicates convolution.

The algorithm inherently enforces non-negativity constraints, making it particularly suitable for physical imaging systems where negative intensities are meaningless. Each iteration of the Richardson-Lucy algorithm can be efficiently implemented in the frequency domain by replacing convolutions with multiplications:

$$\hat{F}^{(k+1)}(u,v) = \hat{F}^{(k)}(u,v) \cdot \mathscr{F}^{-1}\left[\frac{G(u,v)}{H(u,v)\hat{F}^{(k)}(u,v)} \cdot H^*(u,v)\right]$$

A significant advantage of the Richardson-Lucy method is that convergence does not require prior knowledge of noise statistics, only the PSF. However, the method exhibits relatively slow convergence, often requiring numerous iterations to achieve acceptable restoration quality. Furthermore, in the presence of significant noise, the algorithm can amplify noise artifacts in later iterations, necessitating careful selection of stopping criteria or incorporation of regularization techniques. Accelerated variants and regularized extensions, such as those incorporating total variation constraints, have been developed to address these limitations while maintaining the method's fundamental maximum likelihood framework [12].

1.2.5 Homomorphic Filtering

Homomorphic filtering represents a specialized frequency domain technique developed in the 1960s by Stockham, Oppenheim, and Schafer at MIT for processing signals combined in multiplicative or non-linear relationships [13]. In image processing, homomorphic filtering addresses the illumination-reflectance model of image formation, where the observed image intensity is the product of scene illumination and object reflectance:

$$f(x,y) = i(x,y) \cdot r(x,y)$$

where $i(x,y)$ represents illumination (typically varying slowly across the image) and $r(x,y)$ denotes reflectance (often changing abruptly at object boundaries).

The homomorphic filtering approach transforms this multiplicative relationship into an additive one through logarithmic transformation:

$$\ln[f(x,y)] = \ln[i(x,y)] + \ln[r(x,y)]$$

The logarithmic image is then transformed to the frequency domain:

$$\mathscr{F}\{\ln[f(x,y)]\} = \mathscr{F}\{\ln[i(x,y)]\} + \mathscr{F}\{\ln[r(x,y)]\}$$

A high-pass filter $H(u,v)$ is applied in the frequency domain to attenuate low-frequency illumination components while preserving or enhancing high-frequency reflectance components:

$$S(u,v) = H(u,v) \cdot \mathscr{F}\{\ln[f(x,y)]\}$$

The enhanced image is recovered through inverse Fourier transform followed by exponentiation:

$$g(x,y) = \exp\{\mathscr{F}^{-1}[S(u,v)]\}$$

Common filter functions for homomorphic filtering include Gaussian, Butterworth, and custom high-pass filters designed to simultaneously compress the illumination dynamic range (by attenuating low frequencies) and enhance contrast (by amplifying high frequencies). Ramponi demonstrated that Butterworth high-pass filters

are particularly effective for homomorphic applications due to their smooth transition characteristics, which avoid the ringing artifacts associated with ideal filters [14]. The Butterworth high-pass filter is expressed as:

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

where γ_L and γ_H control the degree of low and high frequency modification, $D(u,v)$ is the distance from the frequency origin, D_0 is the cutoff frequency, and n determines the filter slope.

Homomorphic filtering has proven effective for correcting non-uniform illumination in various applications, including medical imaging, satellite imagery, and forensic analysis. However, the method assumes separability of illumination and reflectance components in the frequency domain, which may not hold for all imaging scenarios [15].

1.2.6 Power Spectrum Equalization

Power spectrum equalization represents another classical approach to frequency domain restoration, particularly applicable when the degradation function introduces spectral distortions that can be characterized and compensated through spectral shaping. This technique is especially relevant for ultrasonic imaging, X-ray radiography, and other applications where the system response can be estimated from the degraded image itself.

The fundamental premise of power spectrum equalization is that both the image and noise belong to homogeneous random fields, with the noise being uncorrelated with the image and having zero mean. The restoration filter aims to equalize or whiten the power spectrum of the degraded image by suppressing correlation between image samples introduced by the degradation process [16].
The power spectrum equalization filter can be expressed as:

$$H_{PSE}(u,v) = \frac{1}{\sqrt{S_g(u,v)}}$$

where $S_g(u,v)$ represents the power spectrum of the degraded image. In practice, modifications are introduced to avoid excessive amplification of noise-dominated frequencies:

$$H_{PSE}(u,v) = \frac{S_{ref}(u,v)}{\sqrt{S_g(u,v) + \varepsilon}}$$

where $S_{ref}(u,v)$ is a reference power spectrum (often estimated from typical images in the application domain) and ε is a small constant preventing division by zero.

This approach requires no explicit knowledge of the PSF but relies on statistical characterization of the degradation effects through power spectral analysis. Modified versions incorporating spatial adaptation and frequency-dependent regularization have been developed to improve restoration quality while preserving edge information.

1.2.7 Frequency Domain Filtering Techniques

Beyond specific restoration algorithms, general frequency domain filtering provides fundamental tools for image enhancement and noise reduction. The basic taxonomy includes low-pass, high-pass, band-pass, and band-reject filters, each serving distinct purposes in image processing applications [1].

Low-pass filters attenuate high frequencies while preserving low frequencies, effectively smoothing images and suppressing high-frequency noise. Common implementations include ideal, Butterworth, and Gaussian low-pass filters. The Gaussian low-pass filter is expressed as:

$$H_{LP}(u,v) = e^{-D^2(u,v)/2\sigma^2}$$

where $D(u,v) = \sqrt{u^2+v^2}$ represents the distance from the frequency origin and σ controls the filter spread. The Butterworth low-pass filter of order n provides tunable smoothness:

$$H_{LP}(u,v) = \frac{1}{1+[D(u,v)/D_0]^{2n}}$$

where D_0 is the cutoff frequency. As n increases, the Butterworth filter approaches the behavior of an ideal filter, though at the cost of increased ringing artifacts in the spatial domain.

High-pass filters preserve high frequencies associated with edges and fine details while attenuating low-frequency components. These filters can be derived as complements of low-pass filters:

$$H_{HP}(u,v) = 1 - H_{LP}(u,v)$$

The Gaussian high-pass filter is given by:

$$H_{HP}(u,v) = 1 - e^{-D^2(u,v)/2\sigma^2}$$

while the Butterworth high-pass filter takes the form:

$$H_{HP}(u,v) = \frac{1}{1+[D_0/D(u,v)]^{2n}}$$

Band-pass and band-reject (notch) filters selectively pass or attenuate specific frequency bands. Notch filters are particularly effective for removing repetitive spectral noise patterns that appear as distinct peaks in the frequency domain. An ideal notch reject filter is defined as:

$$H_{NR}(u,v) = \begin{cases} 0 & \text{if } D_1(u,v) \le D_0 \text{ or } D_2(u,v) \le D_0 \\ 1 & \text{otherwise} \end{cases}$$

where D_1 and D_2 represent distances from the notch centers at symmetric frequency locations, and D_0 defines the notch radius. Butterworth notch filters provide smoother transitions:

$$H_{NR}(u,v) = \prod_{k=1}^{Q} \left[1 - \frac{1}{1+[D_k(u,v)/D_0]^{2n}} \right]$$

where Q is the number of notch filter pairs and $D_k(u,v)$ represents the distance from point (u,v) to the center of the k-th notch.

These fundamental filtering operations serve as building blocks for more sophisticated restoration algorithms and can be adapted to specific degradation characteristics observed in the frequency spectrum of degraded images. The choice of filter type involves trade-offs between computational efficiency, ringing artifacts, and transition sharpness [17].

Chapter 2

Mathematics Preliminaries

2.1 Complex Numbers

2.1.1 Definitions

The Fourier Transform is a transform which has its domain are real values ($\mathbb{R}$) and ranges in the complex values ($\mathbb{C}$), we need to study some concepts about complex numbers.

Definition 2.1. A **complex number** is an ordered pair (x,y) where $x,y \in \mathbb{R}$, written as $x+iy$, where i is an imaginary number, that is $i = \sqrt{-1}$. The set of complex numbers is denoted as $\mathbb{C}$

$$\mathbb{C} = \{x+iy : \ x,y \in \mathbb{R}\}$$

Let $z = x+iy$ be any complex number, then

- x is called the **real part**, denoted by $\text{Re}\{z\} := x$.
- y is called the **imaginary part**, denoted by $\text{Im}\{z\} := y$.
- The **complex conjugate** of z, denoted by $\bar{z}$, is defined as $\bar{z} = x - iy$.

Example 2.1. We have some names for a complex number, for example

- $z_1 = 2+7i$ is a general complex number.
- $z_2 = 0-9i$ has its real part equals to zero, this is called the **pure imaginary number**.
- $z_3 = 3+0i$ has its imaginary part equals to zero, this is called the **real number**.

Definition 2.2. A **complex plane** is an Oxy plane where Ox is the *real axis*, Oy is the imagine axis. A number

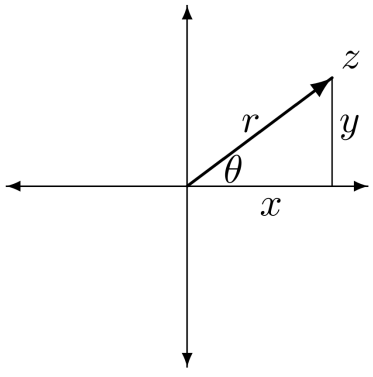

Figure 2.1: Complex number z

Definition 2.3. Let $z = x + iy$ be a non-zero complex number, let $r = |z| = \sqrt{x^2 + y^2}$, $x = r\cos\theta$ and $y = r\sin\theta$

- r is called the **absolute value** (or **magnitude**, or **modulus**) of z, it is also known as the distance to the origin.
- θ is called the **argument** of z, and has the range of $-\pi \leq \theta \leq \pi$.

The representation $z = x + iy = r(\cos\theta + i\sin\theta)$ is called the polar representation of z.

Theorem 2.1. *Let θ be any real number, the complex number*

$$e^{i\theta} = \cos\theta + i\sin\theta$$

*is obtained using the **Euler's formula**.*

2.1.2 Complex Integration

Working with Fourier Transform means dealing with complex integrations. In this part I give a brief introduction about the complex integration.

Definition 2.4. Let $f(t) = x(t) + iy(t)$ be any complex function, here $x(t), y(t)$ are real functions. We define its integration as

$$\int_a^b f(t)dt = \int_a^b x(t)dt + i\int_a^b y(t)dt$$

Let $f(t) = x(t) + iy(t)$ be any complex function, the following definition holds

- The **real part** of $\int_a^b f(t)dt$ is $\int_a^b x(t)dt$.
- The **imaginary part** of $\int_a^b f(t)dt$ is $\int_a^b y(t)dt$.

Keep in mind that complex integration involves integrating over a contour C, for this document, we only consider the simple integration over an axis.

2.2 Fourier Series

Definition 2.5. A function $f(t)$ is said to be **periodic** of period T if there exist a number $T > 0$ such that

$$f(t + T) = f(t)$$

for all t. The smallest value for T is called the *fundamental period* of the function f.

Example 2.2. Consider the function

$$f(t) = \cos 2\pi t + \frac{1}{2}\cos 4\pi t$$

whose graph is shown as below

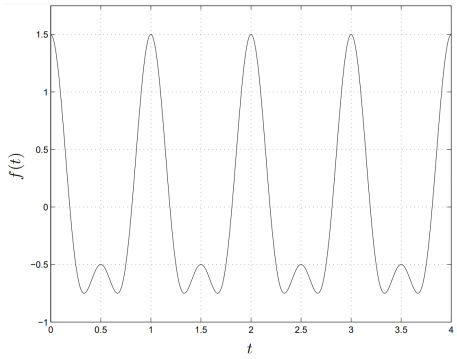

Figure 2.2: Graph of $\cos 2\pi t + \frac{1}{2}\cos 4\pi t$ [18]

This function has a period of 1 since

$$\begin{aligned} f(t+1) &= \cos 2\pi(t+1) + \frac{1}{2}\cos 4\pi(t+1) \\ &= \cos(2\pi t + 2\pi) + \frac{1}{2}\cos(4\pi t + 4\pi) = \cos 2\pi t + \frac{1}{2}\cos 4\pi t = f(t) \end{aligned}$$

It can also be shown that there is no smaller T satisfies this.

Theorem 2.2. *A function $f(t)$ of a continuous variable t, that is periodic with a period T, can be expressed as the sum of sines and cosines, multiplied by appropriate coefficients. This sum, known as the Fourier series, has the form*

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{i\frac{2\pi n}{T}t}$$

where

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} f(t) e^{-i\frac{2\pi n}{T}t}, \text{ for } n \in \mathbb{Z}$$

are the coefficients.

Proof. We first show that why $f(t)$ is an expansion of sines and cosines, this follows from the Euler formula we discussed above. We have

$$e^{\frac{2\pi nt}{T}} = \cos\left(\frac{2\pi nt}{T}\right) + i\sin\left(\frac{2\pi nt}{T}\right)$$

To proof for the coefficients, we simply plug the function into coefficients

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} dx \left[\sum_{m=-\infty}^{\infty} c_m e^{i\frac{2\pi m}{T}t}\right] e^{-i\frac{2\pi n}{T}t} = \frac{1}{T}\sum_{m=-\infty}^{\infty} c_m \int_{-T/2}^{T/2} dx e^{i\frac{2\pi m}{T}t} e^{-i\frac{2\pi n}{T}t}$$

This is equivalent to

$$c_n = \frac{1}{T}\sum_{m=-\infty}^{\infty} c_m \int_{-T/2}^{T/2} dx e^{i\frac{2\pi(m-n)}{T}t} = \frac{1}{T}\sum_{m=-\infty}^{\infty} c_m T \delta_{mn} = c_n$$

Where δ_{mn} is the **Kronecker delta function** which we will discussed later. □

The Fourier series is just an approximation. We know take a look at functions which is not sine or cosine, but can be expressed as sum of them using sines and cosines.

Example 2.3. Consider the function

$$f(t) = \begin{cases} 1 & \text{if } 0 \le t < 1/2 \\ -1 & \text{if } 1/2 \le t < 1 \end{cases}$$

We plot out the function as below

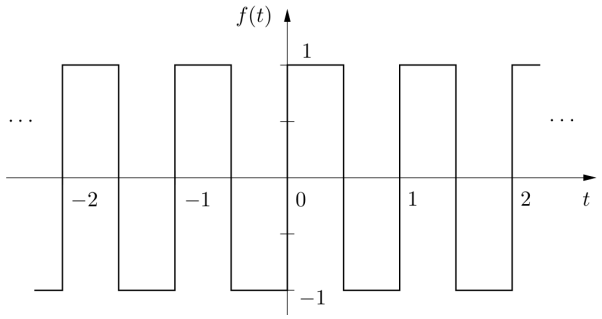

Figure 2.3: Square wave function [18]

This function is periodic of period 1. So we consider

$$\begin{aligned} \hat{f}(n) &= \int_0^1 e^{-2\pi int} f(t)dt \\ &= \int_0^{1/2} e^{-2\pi int} dt - \int_{1/2}^1 e^{-2\pi int} dt \\ &= \left[-\frac{1}{2\pi ine^{-2\pi int}} \right]_0^{1/2} - \left[-\frac{1}{2\pi ine^{-2\pi int}} \right]_{1/2}^1 = \frac{1}{\pi in}(1 - e^{-\pi in}) \end{aligned}$$

By taking the sum, we obtain the Fourier series as

$$\sum_{n \neq 0} \frac{1}{\pi in} \left(1 - e^{-\pi in}\right) e^{2\pi int}$$

Notice that $1 - e^{-\pi in} = \begin{cases} 0 & \text{if } n \text{ even} \\ 2 & \text{if } n \text{ odd} \end{cases}$. The series becomes $\sum_{n \text{ odd}} \frac{2}{\pi in} e^{2\pi int}$. And since n is odd, we write $n = 2k+1$ and notice from the Euler formula that this is equivalent to $e^{2\pi int} = 2i \sin 2\pi nt$, we have the final Fourier series is

$$\frac{4}{\pi} \sum_{k=0}^{\infty} \frac{1}{2k+1} \sin 2\pi(2k+1)t$$

By summing up to 9 and 39 terms, we obtain the following diagrams

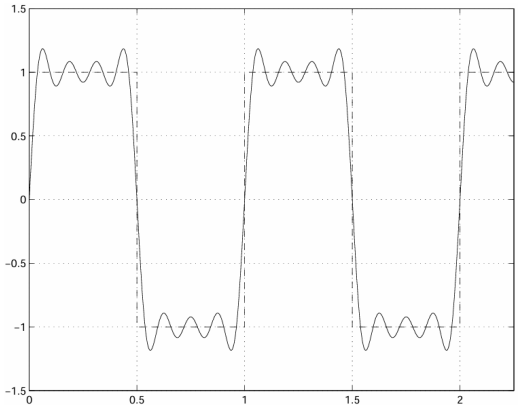

Figure 2.4: Summing up to 9 [18]

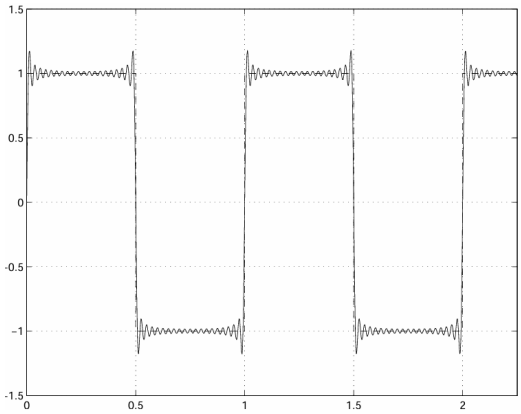

Figure 2.5: Summing up to 39 [18]

2.3 Impulses and Sifting

Definition 2.6. A *unit impulse* of a continuous variable t, located at $t = 0$, denoted $\delta(t)$, is defined as

$$\delta(t) = \begin{cases} \infty & \text{if } t = 0 \\ 0 & \text{if } t \neq 0 \end{cases}$$

and is constrained to satisfy the identity

$$\int_{-\infty}^{\infty} \delta(t)dt = 1$$

Physically, if we interpret t as time, an impulse may be viewed as a spike of infinity amplitude and zero duration, having unit area. An impulse has the so-called sifting property with respect to integration

$$\int_{-\infty}^{\infty} f(t)\delta(t) = f(0)$$

A more general way of this sifting property is that we can locate the impulse at an arbitrary point t_0.

$$\int_{-\infty}^{\infty} f(t)\delta(t - t_0) = f(t_0)$$

We are interested in applying the impulse at periodic locations, an approach is to use the sum

$$s_{\Delta T} = \sum_{k=-\infty}^{\infty} \delta(t - k\Delta T)$$

This is called an *impulse train*.

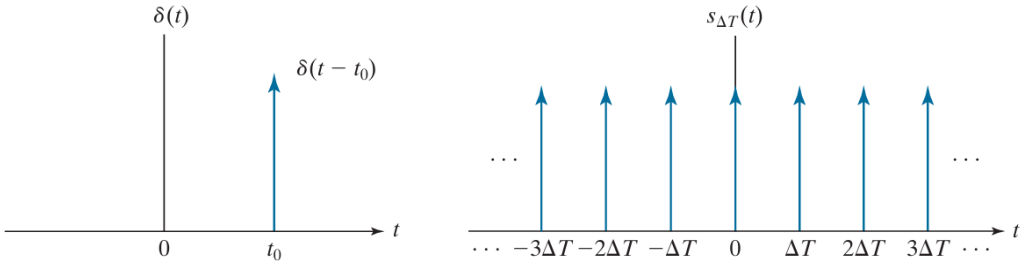

Figure 2.6: A single impulse and its train [1]

Definition 2.7. A *unit impulse* of a discrete variable t, located at $t = 0$, denoted $\delta(x)$, is defined as

$$\delta(x) \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases}$$

and is constrained to satisfy the identity

$$\sum_{x=-\infty}^{\infty} \delta(x) = 1$$

The sifting property also holds in the discrete domain

$$\sum_{x=-\infty}^{\infty} f(x)\delta(x) = f(0) \quad , \quad \sum_{x=-\infty}^{\infty} f(x)\delta(x - x_0) = f(x_0)$$

2.4 Fourier Transform

Definition 2.8. The Fourier transform of a function $f(t)$ is defined as

$$\mathscr{F}(f(t)) = \int_{-\infty}^{\infty} e^{-2\pi iut} f(t) dt$$

We also define $\mathscr{F}(f(t)) = F(u)$, we have $F(u) : \mathbb{R} \mapsto \mathbb{C}$.

The Fourier transform analyzes a signal into its frequency components. In digital image processing, it transforms an image in spatial domain to its frequency domain. We will discuss more about its applications later.

Definition 2.9. The **inverse** Fourier transform of a function $F(u)$ is defined as

$$f(t) = \int_{-\infty}^{\infty} F(u) e^{2\pi iut} du$$

Proof. We assume that the function $f(t)$ have a very large period T, which means $f(t)$ is zero outside the range $[-T/2, T/2]$, now use the Fourier series, we can rewrite the Fourier coefficients as

$$\begin{aligned} c_n &= \frac{1}{T} \int_{-T/2}^{T/2} e^{-2\pi int/T} f(t) dt = \frac{1}{T} \int_{-\infty}^{\infty} e^{-2\pi int/T} f(t) dt \\ &= \frac{1}{T} F\left(\frac{n}{T}\right) = \frac{1}{T} F(s_n) \end{aligned}$$

Pluggin this into the Fourier series, we have

$$f(t) = \sum_{n=-\infty}^{\infty} \frac{1}{T} F(s_n) e^{2\pi i s_n t}$$

Notice that this resembles the Riemann sum approximating the integral $\int_{-\infty}^{\infty} F(u) e^{2\pi iut} du$. Now taking $T \to \infty$, this leads us to

$$f(t) = \int_{-\infty}^{\infty} F(u) e^{2\pi iut} du$$

which completes the proof. □

Note that the Fourier transform is an expansion of $f(t)$ multiplied by sinusoidal terms whose frequencies are determined by the values of u. Thus, because the only variable left after integration is frequency, we say that **the domain of the Fourier transform is the frequency domain**.

Even though the function $F(u)$ itself is complex, we can see that

$$F(0) = \int_{-\infty}^{\infty} f(t)dt$$

This by mean is the integral part of $f(t)$, which is always real. The term $F(0)$ somehow demonstrates how the Fourier transform is frequency domain transform.

Example 2.4. Consider a simple rectangle function $\Pi(t)$ defined as

$$\Pi(t) = \begin{cases} 1 & |t| < 1/2 \\ 0 & |t| \geq 1/2 \end{cases}$$

Its diagram is shown as below

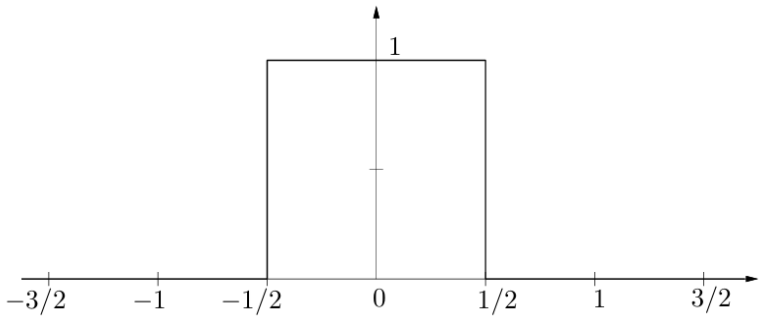

Figure 2.7: Rectangle function

Now apply the Fourier transform, we obtain the Fourier coefficient for c_n as

$$\begin{aligned} c_n &= \frac{1}{T}\int_{-T/2}^{T/2} e^{-2\pi int/T}\Pi(t)dt = \frac{1}{T}\int_{-1/2}^{1/2} e^{-2\pi int/T}\cdot 1 dt \\ &= \frac{1}{T}\left[\frac{1}{-2\pi int/T}e^{-2\pi int}\right]_{-1/2}^{1/2} = \frac{1}{2\pi in}\left(e^{\pi in/T} - e^{-\pi in/T}\right) = \frac{1}{\pi n}\sin\left(\frac{\pi n}{T}\right) \end{aligned}$$

We can scale this c_n by T, to obtain the Fourier transform, let $s = n/T$

$$\mathscr{F}(\Pi(t)) = \frac{\sin(\pi s)}{\pi s}$$

This function is also called the **sinc** function. The diagram of this function is shown below

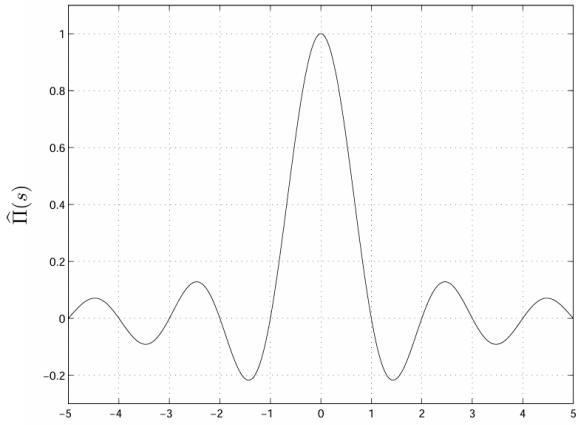

Figure 2.8: $\mathscr{F}(\Pi(t))$ [18]

Notice that $\text{sinc}(0) = 1$ as $\lim_{x\to 0} \dfrac{\sin x}{x} = 1$, we clearly see that

$$\text{sinc}(0) = \int_{-1/2}^{1/2} \Pi(t)dt$$

Which is our desired.

2.4.1 Properties

In this part we discuss some properties of the Fourier transform

Fourier Transform Pairs And Duality

Consider the following

$$\mathscr{F}f(-s) = \int_{-\infty}^{\infty} e^{-2\pi i(-s)t} f(t)dt = \int_{-\infty}^{\infty} e^{2\pi ist} f(t)dt = \mathscr{F}^{-1}f(s)$$
$$\mathscr{F}^{-1}f(-t) = \int_{-\infty}^{\infty} e^{2\pi is(-t)} f(s)ds = \int_{-\infty}^{\infty} e^{-2\pi ist} f(s)ds = \mathscr{F}f(t)$$

This is called the ”duality” property of the transforms. Meaning that you can obtain one from another.

Example 2.5. Knowing that

$$\mathscr{F}\Pi = \text{sinc}$$

and hence that

$$\mathscr{F}^{-1}(\text{sinc}) = \Pi$$

Using the duality property, we can obtain $\mathscr{F}(\text{sinc})$ as follow

$$\mathscr{F}(\text{sinc}(t)) = \mathscr{F}^{-1}\text{sinc}(-t) = \Pi(-t)$$

We sometimes denote the reversed signal $f(t)$ by $f^{-}(t)$, which means

$$f^{-}(t) = f(-t)$$

Even And Odd Symmetries

Recall that a function f is even iff $f = f^-$, and odd iff $f = -f^-$. From here, we obtain similar property for the Fourier transform

$$(\mathscr{F}f)^- = \mathscr{F}f^- = \begin{cases} \mathscr{F}f, & \text{if } f \text{ is even} \\ \mathscr{F}(-f) = -\mathscr{F}f, & \text{if } f \text{ is odd} \end{cases}$$

which shows that if f is even or odd, then so is its Fourier transform.

Linearity

For this, we need to check if

$$\mathscr{F}(f+g)(s) = \mathscr{F}f(s) + \mathscr{F}g(s)$$
$$\mathscr{F}(\alpha f)(s) = \alpha \mathscr{F}f(s)$$

We show a proof for the first property, the second one follows similarly

Proof. We have

$$\begin{aligned} \mathscr{F}(f+g)(u) &= \int_{-\infty}^{\infty} (f(s)+g(s))e^{-2\pi iut} st \\ &= \int_{-\infty}^{\infty} f(t)e^{-2\pi iut}dt + \int_{-\infty} g(t)e^{-2\pi iut}dt = \mathscr{F}f(u) + \mathscr{F}g(u) \end{aligned}$$

This completes the proof. □

The Shift Theorem

It is sometimes needed to shift the variable t over a value b, which now we examine its Fourier transform

$$\begin{aligned} \int_{-\infty}^{\infty} f(t+b)e^{-2\pi iut}dt &= \int_{-\infty}^{\infty} f(s)e^{-2\pi u(s-b)}ds \\ &= \int_{-\infty}^{\infty} f(s)e^{-2\pi ius}e^{2\pi ibu}ds \\ &= e^{2\pi ibu}\int_{-\infty}^{\infty} f(s)e^{-2\pi ius}ds = e^{2\pi iub}F(u) \end{aligned}$$

The Stretch (Similarity) Theorem

We want to know if we scale t to at what happens to the Fourier transform of $f(at)$, we obtain

$$\begin{aligned} \int_{-\infty}^{\infty} f(at)e^{-2\pi iut}dt &= \int_{-\infty}^{\infty} f(s)e^{-2\pi u(s/a)}\frac{1}{a}ds \\ &= \frac{1}{a}\int_{-\infty}^{\infty} f(s)e^{-2\pi i(s/a)u}ds = \frac{1}{a}F\left(\frac{u}{a}\right) \end{aligned}$$

This means changing the variable from x to ax is a change of scale, also known as a similarity.

2.5 Convolution

In this section, we consider the properties when applying the Fourier transform to the convolution.

Definition 2.10. The convolution of two functions $g(t)$ and $f(t)$ is the function

$$h(t) = \int_{-\infty}^{\infty} g(t-x)f(x)dx$$

We use the notation

$$(g * f) = \int_{-\infty}^{\infty} g(t-x)f(x)dx$$

Theorem 2.3. *Applying the Fourier transform on both sides of the equation gives the result*

$$\mathscr{F}(g * f)(s) = \mathscr{F}g(s)\mathscr{F}f(s)$$

Proof. We start by going from $\mathscr{F}g(u)\mathscr{F}f(u)$, which gives the following

$$\mathscr{F}g(u)\mathscr{F}f(u) = \int_{-\infty}^{\infty} g(t)e^{-2\pi iut}dt \int_{-\infty}^{\infty} f(x)e^{-2\pi iux}dx$$

Different variables on both sides in the integral because we're going to combine the product into an iterated integral

$$\begin{aligned}\int_{-\infty}^{\infty} g(t)e^{-2\pi iut}dt \int_{-\infty}^{\infty} f(x)e^{-2\pi iux}dx &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(t)f(x)e^{-2\pi iut}e^{-2\pi iux}dtdx \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(t)f(x)e^{-2\pi iu(t+x)}dtdx \\ &= \int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} g(t)e^{-2\pi iu(t+x)}dt\right)f(x)dx\end{aligned}$$

Let $s = t + x$, hence $ds = dt$ and we have

$$\int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} g(t)e^{-2\pi iu(t+x)}dt\right)f(x)dx = \int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} g(s-x)e^{-2\pi ius}ds\right)f(x)dx$$

which now follows that

$$\begin{aligned}\int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} g(s-x)e^{-2\pi ius}ds\right)f(x)dx &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(s-x)f(x)e^{-2\pi ius}dsdx \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(s-x)f(x)e^{-2\pi ius}dxds \\ &= \int_{-\infty}^{\infty} e^{-2\pi ius}\left(\int_{-\infty}^{\infty} g(s-x)f(x)dx\right)ds\end{aligned}$$

Since $h(s) = \int_{-\infty}^{\infty} g(s-x)f(x)dx$, we continue as

$$\int_{-\infty}^{\infty} e^{-2\pi ius}\left(\int_{-\infty}^{\infty} g(s-x)f(x)dx\right)ds = \int_{-\infty}^{\infty} h(s)e^{-2\pi ius}ds = \mathscr{F}h(u)$$

This completes the proof which shows that $\mathscr{F}(g * f)(s) = \mathscr{F}g(s)\mathscr{F}f(s)$. □

It is not surprising that we can also use the inverse Fourier transform to obtain a similar result

$$\mathscr{F}^{-1}(g * f) = \mathscr{F}^{-1}g\mathscr{F}^{-1}f$$

We also have

$$\mathscr{F}(gf)(s) = (\mathscr{F}g * \mathscr{F}f)(s)$$

This shows that multiplication in the spatial domain corresponds to convolution in the frequency domain.

2.6 The Discrete Fourier Transform

Since this document mainly focus on digital image processing, which is a discrete domain, we want to focus on this.

2.6.1 Sampling And The Fourier Transform Of Sampled Functions

Earlier we discuss about the impulse function, whose concept is similar to a discrete domain, we construct some theory about it.

The basic idea of the impulse function is to "spike" a continuous functions as periodic locations for sampling. This allows us to use discrete values rather than continuous ones.

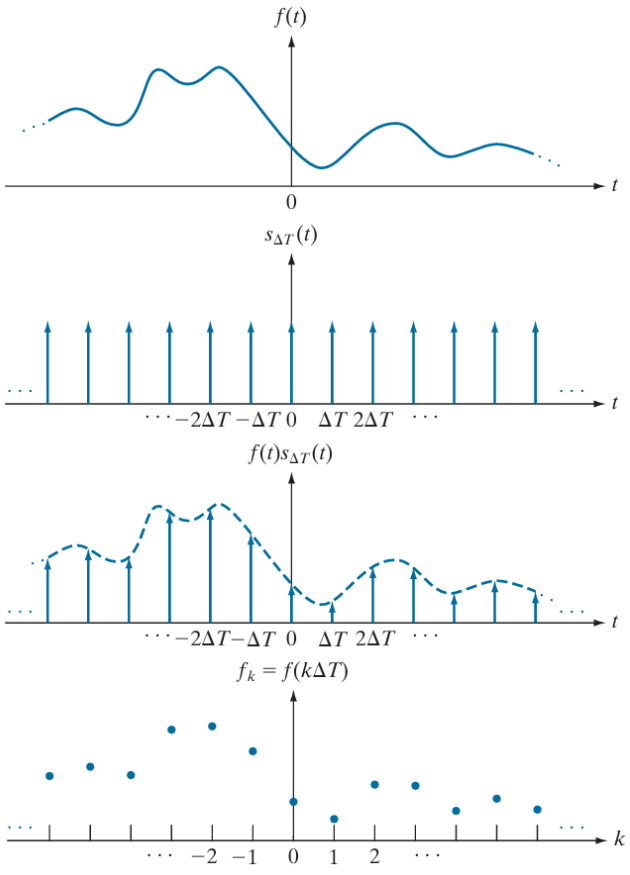

Figure 2.9: Sampling a continuous function [1]

As the figure above shows, we want to sample function $f(t)$ at locations $k\Delta T$, one simple approach is to multiply $f(t)$ with the impulse train function $s_{\Delta T}$ to obtain discrete values as shown in the last figure. Hence we derive the sampled function $\tilde{f}(t)$ as

$$\tilde{f}(t) = f(t)s_{\Delta T} = \sum_{n=-\infty} \infty f(t)\Delta(t - n\Delta T)$$

Note the the Fourier transform of the impulse train function $s_{\Delta T}$ is $\frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} \Delta\left(u - \frac{n}{\Delta T}\right)$, we can obtain the

Fourier transform of the sampled function as below

$$\begin{aligned}\tilde{F}(u) = (F \star S)(u) &= \int_{-\infty}^{\infty} F(t)S(u-t)dt \\ &= \frac{1}{\Delta T}\int_{-\infty}^{\infty} F(t)\sum_{n=-\infty}^{\infty}\delta\left(u-t-\frac{n}{\Delta T}\right)dt \\ &= \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty}\int_{-\infty}^{\infty} F(t)\delta\left(u-t-\frac{n}{\Delta T}\right)dt \\ &= \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} F\left(u-\frac{n}{\Delta T}\right)\end{aligned}$$

where the final step follows from the sifting property of the impulse.

2.6.2 The Discrete Transform

When working with images, we are interested in the discrete domain, hence discrete transform is very important.

Recall the Fourier transform of sampled function is

$$\tilde{F}(u) = \int_{-\infty}^{\infty}\tilde{f}(t)e^{-i2\pi ut}dt$$

By using $\tilde{f}(t) = \sum_{n=-\infty}^{\infty} f(t)\delta(t-n\Delta T)$, we obtain

$$\begin{aligned}\tilde{F}(u) = \int_{-\infty}^{\infty}\tilde{f}(t)e^{-j2\pi ut}dt &= \int_{-\infty}^{\infty}\sum_{n=-\infty}^{\infty} f(t)\delta(t-n\Delta T)e^{-j2\pi ut}dt \\ &= \sum_{n=-\infty}^{\infty}\int_{-\infty}^{\infty} f(t)\delta(t-n\Delta T)e^{-j2\pi ut}dt \\ &= \sum_{n=-\infty}^{\infty} f_n e^{-j2\pi un\Delta T}\end{aligned}$$

The last equal is obtained by letting $g(t) = f(t)e^{-i2\pi ut}$. Using $\int_{-\infty}^{\infty} g(t)\delta(t-n\Delta T)dt = g(n\Delta T)$ gives us the desired equation.

Suppose we want to obtain M equal spaced samples of $\tilde{F}(u)$ taken over one period interval from $u = 0$ to $u = 1/\Delta T$. This is accomplished by taking $u = \frac{m}{M\Delta T}$ where $m = 0, 1, 2, \ldots, M-1$. Substituting this u into $\tilde{F}(u)$ gives us

$$F_m = \sum_{n=0}^{M-1} f_n e^{-i2\pi mn/M}$$

This is called the **Discrete Fourier Transform (DFT)**. We can recover the sample set $\{f_m\}$ by using the inverse discrete Fourier function

$$f_n = \frac{1}{M}\sum_{m=0}^{M-1} F_m e^{i2\pi mn/M}$$

To have a better look of how this is done, it is best why examine the following example

Example 2.6. Assume we have the following function sampled as below

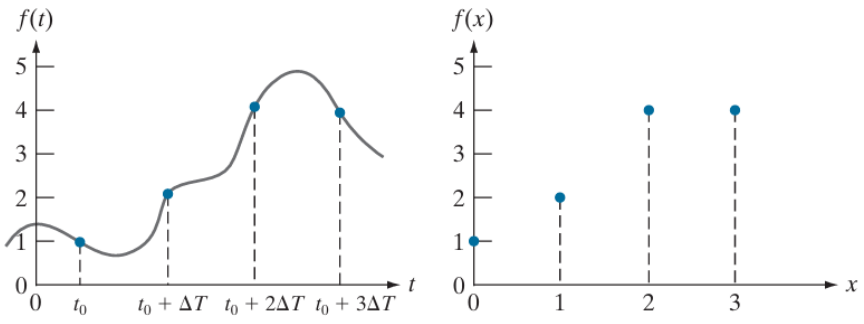

Figure 2.10: DFT Sampled [1]

As known, the first value of $F(u)$, which is $F(0)$, is the sum of all sampled values

$$F(0) = \sum_{x=0}^{3} f(x) = [f(0) + f(1) + f(2) + f(3)] = 11$$

The next value is

$$F(1) = \sum_{x=0}^{3} f(x)e^{-2\pi ix/4} = 1e^{0} + 2e^{-i\pi/2} + 4e^{-i\pi} + 4e^{-i3\pi/2} = -3 + 2i$$

Continue this process to obtain $F(2) = -1$ and $F(3) = -3 - 2i$. To construct $f(t)$ again, we use the inverse Fourier transform. For example, with $f(0)$, we have

$$f(0) = \frac{1}{4}\sum_{x=0}^{3} F(u) = \frac{1}{4}[11 - 3 + 2i - 1 - 3 - 2i] = 1$$

Which proofs the correctness of the theorem.

The discrete equivalent of the 1-D convolution is

$$f(x) * h(x) = \sum_{m=0}^{M-1} f(m)h(x-m) \; x = 0, 1, 2, \cdots, M-1$$

2.7 The 2-D Fourier Transform

The 2D transform mainly derives from the 1D transform, in this section we walk through some concepts in the 2D domain.

2.7.1 The 2-D Impulse And Its Sifting Property

Definition 2.11. The impluse $\delta(t,z)$ of two continuous variables t, z is defined as

$$\delta(t,z) = \begin{cases} 1 & \text{if } t = z = 0 \\ 0 & \text{otherwise} \end{cases}$$

Which is also constrained

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(t,z)dtdz = 1$$

The sifting property also works similar here, where we have

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)\delta(t,z)dtdz = f(0,0)$$

and

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)\delta(t-t_0,z-z_0)dtdz = f(t_0,z_0)$$

To obtain properties for the discrete domain, we have the following

Definition 2.12. The impluse $\delta(t,z)$ of two discrete variables t,z is defined as

$$\delta(t,z) = \begin{cases} 1 & \text{if } t = z = 0 \\ 0 & \text{otherwise} \end{cases}$$

Which is also constrained

$$\sum_{t=-\infty}^{\infty}\sum_{z=-\infty}^{\infty} \delta(t,z) = 1$$

Its sifting property is

$$\sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty} f(x,y)\delta(x,y)dxdy = f(0,0)$$

and

$$\sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty} f(x,y)\delta(x-x_0,y-y_0)dtdz = f(x_0,y_0)$$

2.7.2 The 2-D Continuous Fourier Transform

Let $f(t,z)$ be a continuous function of two variables t and z. The two dimensional, continuous Fourier transform pair is given by the expressions

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)e^{-i2\pi(ut+vz)}dtdz$$

and

$$f(t,z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{i2\pi(ut+vz)}dudv$$

Equations above are the forward and inverse Fourier transform in 2-D. For discrete values, we simply sample the values to obtain something similar to what described in the 1-D domain. Which is, the 2-D discrete Fourier transform (DFT) is

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

Where $f(x,y)$ is a digital image of size $M \times N$. We can also obtain $f(x,y)$ by using the inverse discrete Fourier transform

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

2.8 Some Properties

In this section we discuss some of the most important factors of the Fourier transform.

2.8.1 Periodicity

As in the 1-D case, the 2-D Fourier transform and its inverse are infinitely periodic in the u and v directions; that is

$$F(u,v) = F(u+k_1M,v) = F(u,v+k_2N) = F(u+k_1M,v+k_2N)$$

and

$$f(x,y) = f(x+k_1M,y) = f(x,y+k_2N) = f(x+k_1M,y+k_2N)$$

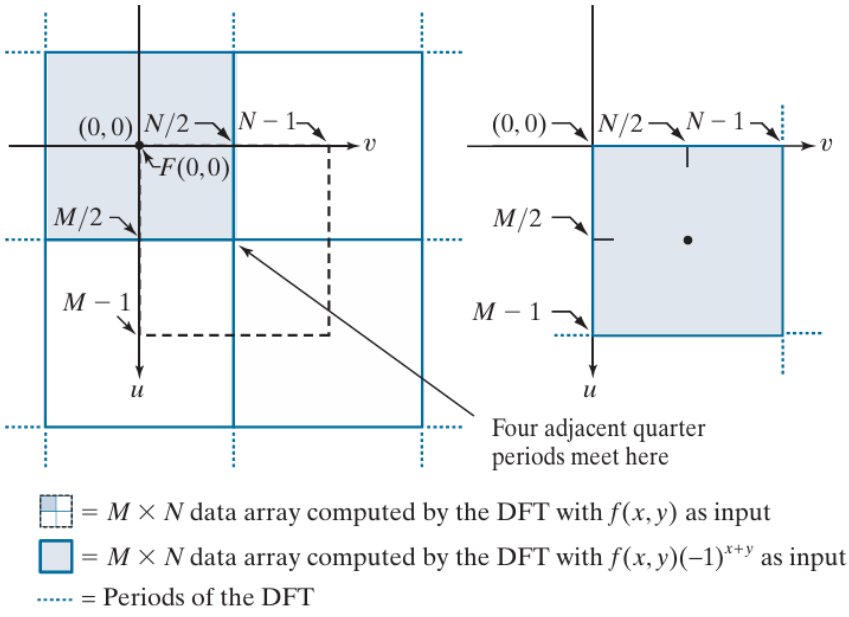

Figure 2.11: Centering the image [1]

Recall that the value $F(0,0)$ has the sum of all sampled values. For display purpose, it is more convenient to have this interval a complete period of the transform in which the data are contiguous and ordered properly. It follows that

$$f(x)e^{i2\pi(u_0x/M)} \Leftrightarrow F(u-u_0)$$

This concept, as for the purpose of centering in 1-D and 2-D we let $u_0 = M/2$ to obtain

$$f(x)(-1)^x \Leftrightarrow F(u-M/2)$$

and

$$f(x,y)(-1)^{x+y} \Leftrightarrow F(u-M/2,v-N/2)$$

2.8.2 Fourier Spectrum And Phase Angle

Let $R(u,v)$ and $I(u,v)$ respectively be the real and imaginary part of $F(u,v)$, which we have

$$F(u,v) = R(u,v) + iI(u,v) = |F(u,v)|e^{i\phi(u,v)}$$

Where we have the magnitude

$$|F(u,v)| = \left[R^2(u,v) + I^2(u,v)\right]^{1/2}$$

and the phase angle

$$\phi(u,v) = \arctan\left[\frac{R(u,v)}{I(u,v)}\right]$$

Moreover, the power spectrum is denoted by

$$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$$

These values shows the properties of the sampled function $f(x,y)$.

2.9 The 2-D Discrete Convolution Theorem

In the next chapter, when working with the Fourier transforms, we focus on using the convolution, which is why now we introduce about the convolution.

Definition 2.13. The 2-D discrete convolution is defined as

$$(f * h)(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)h(x-m,y-n)$$

for $x = 0,1,\cdots,M-1$ and $y = 0,1,\cdots,N-1$.

From here, we also obtain the following theorem

Theorem 2.4. *The 2-D convolution theorem stated that*

$$(f * h)(x,y) \Leftrightarrow (F \cdot H)(u,v)$$

and

$$(f \cdot h)(x,y) \Leftrightarrow \frac{1}{MN}(F * H)(u,v)$$

Where F,H are the Fourier transforms of f and h.

Chapter 3

Applications in Digital Image Processing

3.1 Filtering

3.1.1 Fundamentals

Filtering in the frequency domain consists of modifying the Fourier transform of an image, then computing the inverse transform to obtain the spatial domain representation of the processed result. A basic filtering equation in which we are interested has the form

$$g(x,y) = \mathrm{Re}\left\{\mathscr{F}^{-1}\left[H(u,v)F(u,v)\right]\right\}$$

where

- $F(u,v)$ is the DFT of the input image.
- $H(u,v)$ is the *filter transfer function* (also called as *filter* or *filter function*).
- $g(x,y)$ is the output image.

Since most functions $H(u,v)$ are symmetric about their center, this requires $H(u,v)$ to be centered as well.

Another important feature needed to be considered is that when applying DFT, it is multiplication in the frequency domain, thus if we don't "extend" the image, this will cause the output result to be cropped to match the size of the filter. Which is why we need to apply zero padding on the edges of an image.

From here, we have a summary of steps for filtering in the frequency domain.

1. Given an input image $f(x,y)$ of size $M \times N$, obtain the padding sizes $P = 2M, Q = 2N$.
2. Form a padded image $f_p(x,y)$ of size $P \times Q$ using zero padding.
3. Multiply $f_p(x,y)$ by $(-1)^{x+y}$ to center the Fourier transform on the $P \times Q$ frequency triangle.
4. Compute the DFT $F(u,v)$ of the above image.
5. Construct a real, symmetric filter transfer function, $H(u,v)$ of size $P \times Q$ with center at $(M/2, N/2)$.
6. Form the product, which is element wise $G(u,v) = H(u,v)F(u,v)$.
7. Obtain the filtered image (of size $P \times Q$) by computing the IDFT of $G(u,v)$
$$g_p(x,y) = \mathrm{Re}\left\{\mathscr{F}^{-1}\left[H(u,v)F(u,v)\right]\right\}$$
8. Obtain the final result, $g(x,y)$ of the same size of the original image, by extracting the $M \times N$ region on the top, left quadrant of $g_p(x,y)$.

3.1.2 Smoothing

In this section we discuss about three main filter functions: ideal lowpass filter, gaussian lowpass filter, and the butterworth lowpass filter.

Ideal Lowpass Filter

A 2-D lowpass filter that passes without attenuation all frequencies within a circle of radius from the origin, and "cuts off" all frequencies outside this, circle is called an ideal lowpass filter (ILPF); it is specified by the transfer function

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

Where D_0 is the cutoff frequency, and $D(u,v) = \sqrt{(u-M/2)^2 + (v-N/2)^2}$ is the distance from each pixel to the center of the image.

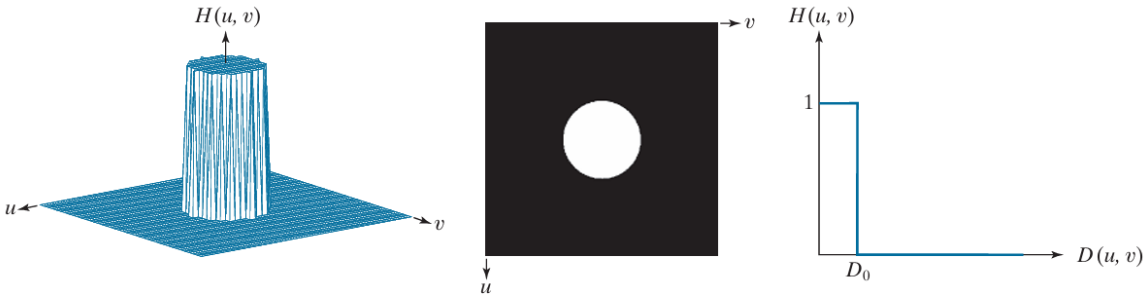

Figure 3.1: Ideal Lowpass Filter

The ILPF indicates that all frequencies inside the circle of radius D_0 are passed with no attenuation.

We can see the following effects of ILPF:

- Blurring effect which decreases as the cutoff frequency inceases.
- Ringing effect which becomes finer as the cutoff frequency increases.

The higher the radius, the more the image remains the same as the original image.

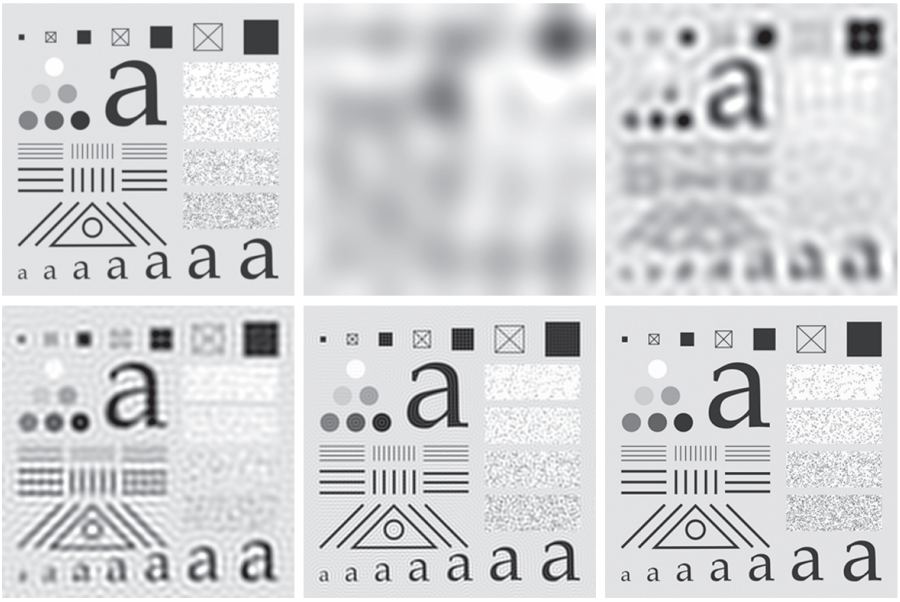

Figure 3.2: The top left is the original image, the rest are results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460

Gaussian Lowpass Filter

The Gaussian lowpass filter with cutoff frequency D_0 is defined as

$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

The Gaussian lowpass filter shows a more smoother transition passing the cutoff frequency, making the result image more smooth.

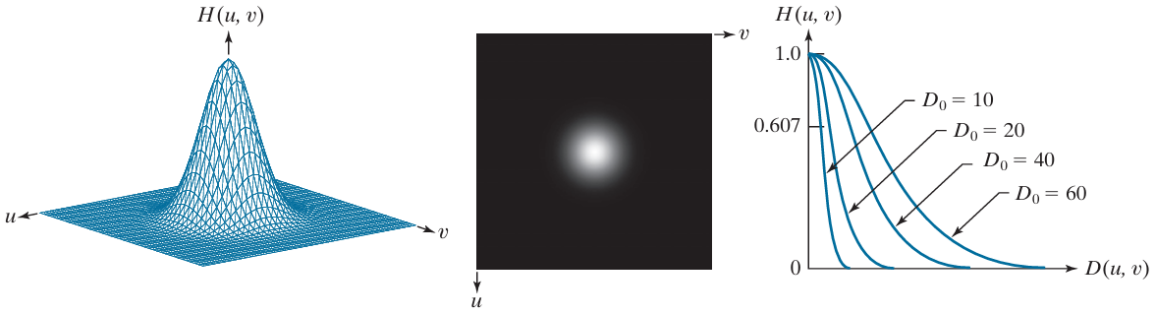

Figure 3.3: Gaussian Lowpass Filter **digitalimage**

We can see the following effects of GLPF:

- Smooth transition in blurring as a function of increasing cutoff.
- No ringing effect.

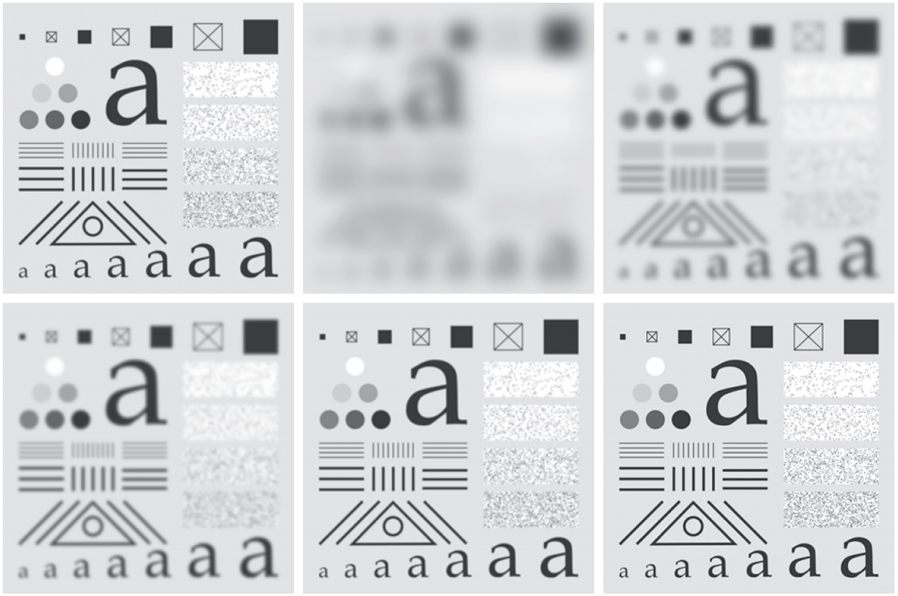

Figure 3.4: The top left is the original image, the rest are results of filtering using GLPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460 **digitalimage**

Butterworth Lowpass Filter

The Butterworth lowpass filter (BLPF) of order n with cutoff frequency D_0 is defined as

$$H(u,v) = \frac{1}{1+[D(u,v)/D_0]^{2n}}$$

The Butterworth lowpass filter shows smooth, gradual transition between pass and stop bands. Unlike an ideal lowpass filter that has a sharp cutoff, the Butterworth filter's response smoothly decreases from the passband to the stopband.

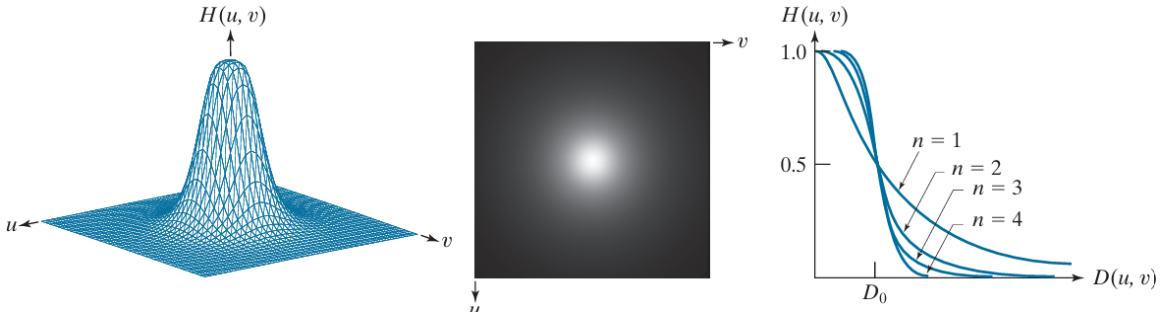

Figure 3.5: Butterworth Lowpass Filter **digitalimage**

We can see the following effects of BLPF:

- Gradually reduces high frequencies rather than cutting them off abruptly
- A higher order results in a sharper transition, while a lower order yields a more gradual cutoff.

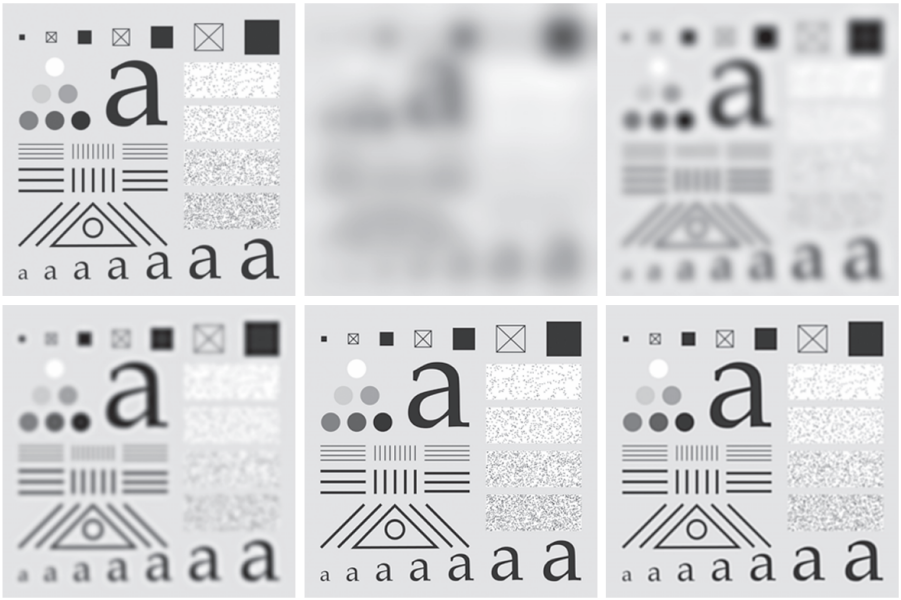

Figure 3.6: The top left is the original image, others are results of filtering using BLPFs with cutoff frequencies at the radii 10, 30, 60, 160, and 460 and n = 2.25 **digitalimage**

Summary

We show a summary of the lowpass filters.

Filter Type	**Pros**	**Cons**
Ideal Lowpass Filter	- Perfectly passes all low frequencies and completely blocks high frequencies.	- Sharp cutoff causes ringing artifacts (Gibbs phenomenon) in the spatial domain, leading to unwanted oscillations near edges.
Gaussian Lowpass Filter	- Smooth, gradual transition between passband and stopband, reducing artifacts and avoiding ringing.	- Does not provide a sharp cutoff, so some high frequencies close to the cutoff may still pass through.
Butterworth Lowpass Filter	- Balances between sharp cutoff and smooth transition, with adjustable sharpness via the filter order.	- Higher-order filters may still introduce some ringing artifacts, though less pronounced than the ideal filter.

Table 3.1: Pros and Cons of Different Lowpass Filters

Based on the analysis above, the following table outlines the specific scenarios in which each filter type—Ideal, Gaussian, and Butterworth lowpass filters—is most effective. This summary provides guidance on choosing the appropriate filter depending on the desired outcome, whether it be strict frequency separation, smooth blurring, or a balanced approach to edge preservation and noise reduction.

Filter Type	Best Scenario
Ideal Lowpass Filter	Situations where a strict frequency cutoff is needed, and ringing artifacts are not a concern (e.g., theoretical studies, simulations).
Gaussian Lowpass Filter	Applications requiring smooth, artifact-free blurring, like noise reduction in sensitive images (e.g., medical imaging, photography with smooth gradients).
Butterworth Lowpass Filter	General-purpose filtering where a balance between sharpness and smoothness is required, and minimal ringing artifacts are acceptable (e.g., feature extraction, real-world image processing).

Table 3.2: Comparison of Lowpass Filters and Their Best Scenarios

3.1.3 Sharpening

We showed in the previous section that an image can be smoothed by attenuating the high-frequency components of its Fourier transform. Because edges and other abrupt changes in intensities are associated with high-frequency components, image sharpening can be achieved in the frequency domain by highpass filtering, which attenuates low-frequencies components without disturbing high-frequencies in the Fourier transform. In this section, we introduce about the ideal highpass filter, Gaussian highpass filter and the Butterworth highpass filter.

All the highpass filter later will have the same feature, that is we are subtracting from the lowpass filter, meaning that

$$H_{\text{HP}}(u,v) = 1 - H_{\text{LP}}(u,v)$$

where

- $H_{\text{HP}}(u,v)$ is the highpass filter.
- $H_{\text{LP}}(u,v)$ is the lowpass filter.

Ideal, Gaussian and Butterworth Highpass Filter

Given the concept of the highpass filter above, we have the following table that describes each filter

Ideal Highpass Filter	Gaussian Highpass Filter	Butterworth Highpass Filter
$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \leq D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$	$H(u,v) = \dfrac{1}{1+[D_0/D(u,v)]^{2n}}$

Table 3.3: Highpass Filters

The following figure illustrates the perspective plot, the corresponding image, and the radial cross-section of a highpass transfer function. The perspective plot provides a three-dimensional view of how the transfer function varies with frequency, displaying the attenuation and passing regions in a spatial context. The image representation shows the transfer function's frequency response in a visual format, allowing for a clearer understanding of its behavior. Additionally, the radial cross-section gives a one-dimensional view along the radial direction, showing how the transfer function attenuates frequencies as a function of distance from the origin, highlighting the cutoff behavior between the low and high frequencies.

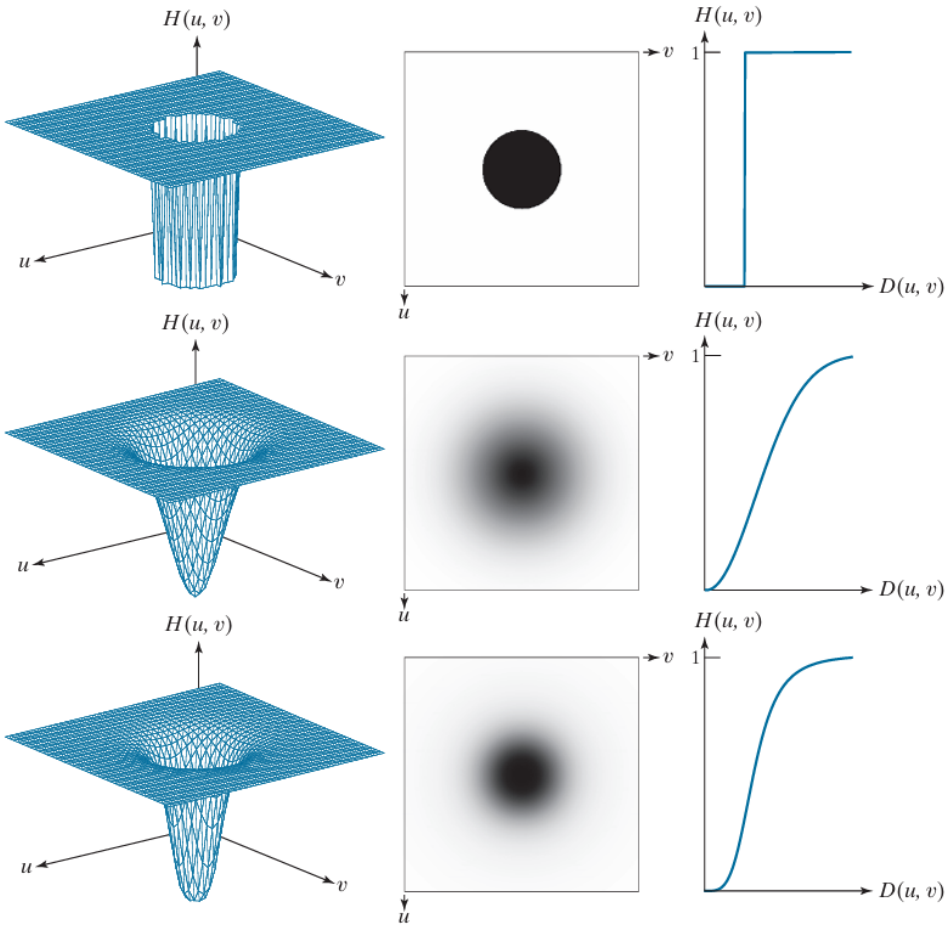

Figure 3.7: Highpass Filters **digitalimage**

The results of using the highpass filters are shown in the following figure. Generally

- **Ideal highpass filter.** The Ideal highpass filter allows frequencies above a specified cutoff frequency to pass unchanged and completely blocks frequencies below it. It is defined by a sharp, abrupt transition at the cutoff.

- **Gaussian highpass filter.** The Gaussian highpass filter provides a smooth, continuous transition between the passband and stopband. It attenuates low frequencies gradually and allows high frequencies to pass with a smooth roll-off.

- **Butterworth highpass filter.** he Butterworth highpass filter offers a tunable transition between the passband and stopband, controlled by the filter's order n. A higher order results in a sharper cutoff, while a lower order creates a more gradual transition.

Summary

We show a summary of the highpass filters.

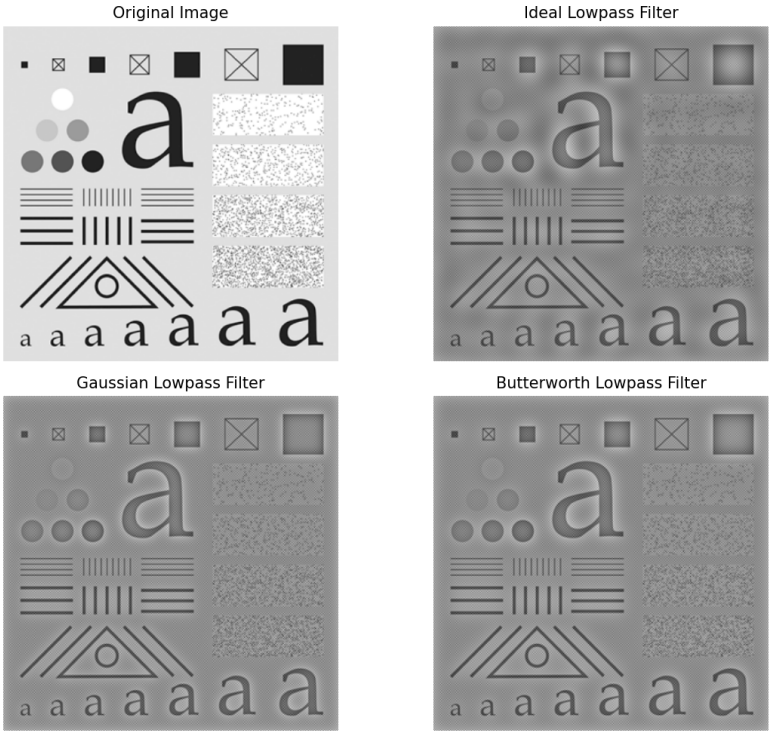

Figure 3.8: Results of the filters. The first image is the original image, others respectively shows the ideal, Gaussian and Butterworth (order 2) highpass filter

Filter Type	Pros	Cons
Ideal Highpass Filter	- Perfect frequency separation between low and high frequencies.	- Sharp cutoff causes significant ringing artifacts (Gibbs phenomenon) in the spatial domain.
Gaussian Highpass Filter	- Smooth transition between passband and stopband, minimizing artifacts.	- Does not provide a sharp cutoff, allowing some low frequencies near the cutoff to pass.
Butterworth Highpass Filter	- Adjustable smoothness with filter order, providing a balance between sharpness and smoothness.	- Higher-order filters can introduce minor ringing artifacts and increase computational complexity.

Table 3.4: Comparison of Pros and Cons of Different Highpass Filters

Based on the analysis above, the following table outlines the specific scenarios in which each filter type—Ideal, Gaussian, and Butterworth highpass filters—is most effective. This summary provides guidance on choosing the appropriate filter depending on the desired outcome, whether it be strict frequency separation, smooth blurring, or a balanced approach to edge preservation and noise reduction.

Filter Type	Best Scenario
Ideal Highpass Filter	Best for theoretical studies or simulations where a sharp frequency separation is required, and the ringing artifacts are acceptable. Often used in cases where precise control over the frequency cutoff is necessary.
Gaussian Highpass Filter	Ideal for general-purpose image processing where smooth transitions and minimal artifacts are prioritized. Commonly used in edge enhancement, high-frequency noise reduction, or applications that require a natural and artifact-free result.
Butterworth Highpass Filter	Best when a balance between sharpness and smoothness is needed, with minimal artifacts. Frequently used in image enhancement, feature extraction, and noise reduction, especially when flexibility in controlling filter sharpness is desired.

Table 3.5: Best Scenarios for Using Different Highpass Filters

3.2 Restoration

The main objective of restoration is to obtain an estimated image which is closest to the original image. Its concept can be describe as in the image below

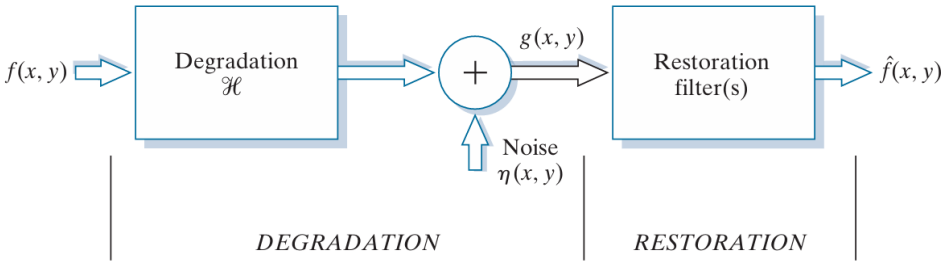

Figure 3.9: Restoration Process **digitalimage**

Notations in the figure above will be introduce shortly. Since this document mainly focus on the Fourier transform, thus we don't dive in too deep about the noise models. In order to understand the usage of the following restoration filters, we need to know what is a noise model, and what does it do to an image.

3.2.1 Noise Models

Given an input image $f(x,y)$, degraded image $g(x,y)$, and some knowledge about $\mathscr{H}$, and some knowledge about the additive noise term $\eta(x,y)$. The objective is to obtain an estimate $\hat{f}(x,y)$ of the original image.

If $\mathscr{H}$ is a linear, position-invariant operator, then the degraded image is given in the spatial domain by

$$g(x,y) = (h * f)(x,y) + \eta(x,y)$$

In case of the frequency domain, we use convolution as

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

Where

- $G(u,v)$ is the DFT of $g(x,y)$, which is the output image.
- $H(u,v)$ is the DFT of $h(x,y)$, which is the degraded function.
- $F(u,v)$ is the DFT of $f(x,y)$, which is the input image.
- $N(u,v)$ is the DFT of $\eta(x,y)$, which is the noise added.

Noises can be constructed based on distributions. The purpose of noise is to degrade the image to a certain level, making it hard to display the main features.

The figure below shows an original image compared to some of the noise images.

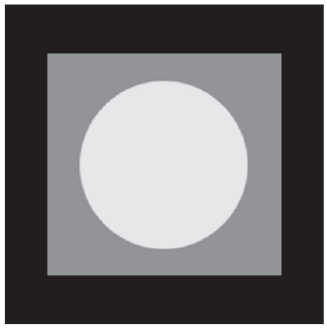

Figure 3.10: Original Image **digitalimage**

These three images below shows the result after applying Gaussian, Rayleigh, and Erlang noise. Again, we are not discussing details how the noise is added, but more about the restoration process, which is where the Fourier transform is applied.

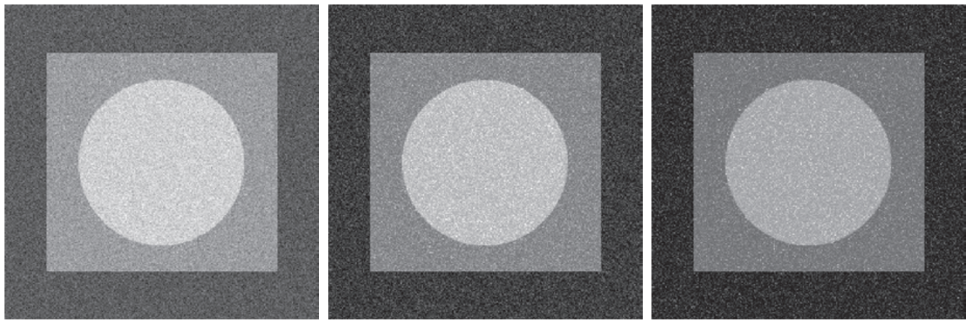

Figure 3.11: Images resulting from adding Gaussian, Rayleigh, and Erlang noise **digitalimage**

3.2.2 Inverse Filtering

This technique is considered most trivial, recall from the degraded image after applying the Fourier transform

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

Consider we ignore noise, the estimated restored image is

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

Plug into the first equation, we have

$$\hat{F}(u,v) + F(u,v) + \frac{N(u,v)}{H(u,v)}$$

This is the inverse filtering. It can easily be seen that this technique has some issues

1. **Noise Amplification**
    - If the degradation function $H(u,v)$ has zeros or near-zero values in the frequency domain, the inverse filter may amplify the noise significantly. This is a common issue when $H(u,v)$ has low-frequency components or is ill-conditioned.
    - In practice, the noise term $N(u,v)$ in the frequency domain complicates inverse filtering because the inverse filter will amplify noise at frequencies where $H(u,v)$ is small.
2. **Division by Zero**
    - If $H(u,v) = 0$ for certain frequencies, the inverse filter will fail because division by zero occurs. This leads to instability in the restoration process.
3. **Ill-posed Problem**
    - Inverse filtering is often ill-posed, especially when the degradation function $H(u,v)$ is not perfectly known or when $H(u,v)$ has large regions of low or zero values in the frequency domain.

Applying the inverse filtering gives the following result.

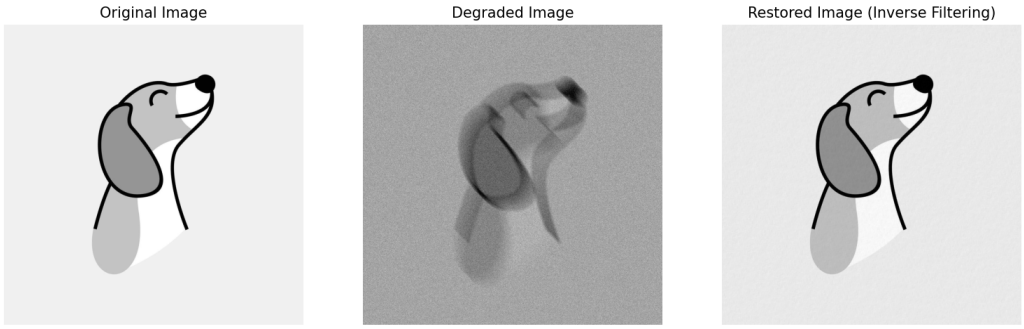

Figure 3.12: Inverse Filtering

The left-most image is the original image, the second image is obtained by using the degraded function which shifts an image, finally, the right-most image is obtained by applying inverse filtering.

Advantages of Inverse Filtering:

- **Simplicity.** Inverse filtering is straightforward to implement. It directly uses the degradation model and applies the inverse operation to recover the original image.
- **Mathematical Foundation.** It provides an optimal solution when the degradation process is perfectly known and no noise is present.
- **Computationally Efficient.** Since it operates in the frequency domain and only requires division, inverse filtering is computationally simpler when compared to Wiener filtering, which requires additional steps to estimate noise and signal power spectral densities.

3.2.3 Wiener Filtering

Wiener filtering is a technique used to restore an image that has been degraded by noise and blurring. It is an optimal filter in the least mean square error (LMS) sense, meaning it minimizes the mean square error between the estimated (restored) image and the original image. The Wiener filter is particularly effective when the noise and degradation process are known.

The Wiener filter works by balancing between two competing objectives:

1. Noise reduction: It attempts to suppress the noise present in the degraded image.
2. Edge preservation: It tries to retain important features, like edges, in the image.

The objective of the Wiener filtering is to minimize the error

$$e^2 = E\left\{\left(f-\hat{f}\right)^2\right\}$$

where $f, \hat{f}$ respectively are the input and the estimated image.

The result for the best estimated image is

$$\begin{aligned}\hat{F}(u,v) &= \left[\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)|H(u,v)|^2+S_\eta(u,v)}\right]G(u,v)\\ &= \left[\frac{H^*(u,v)}{|H(u,v)|^2+S_\eta(u,v)/S_f(u,v)}\right]G(u,v)\\ &= \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2+S_\eta(u,v)/S_f(u,v)}\right]G(u,v)\end{aligned}$$

where

1. $\hat{F}(u,v)$ = Fourier transform of the estimate of the undergraded image.
2. $G(u,v)$ = Fourier transform of the degraded image.
3. $H(u,v)$ = degradation transfer function (Fourier transform of the spatial degradation).
4. $H^*(u,v)$ = complex conjugate of $H(u,v)$.
5. $|H(u,v)|^2 = H^*(u,v)H(u,v)$.
6. $S_\eta(u,v) = |N(u,v)|^2$ = Power spectrum of the noise.
7. $S_f(u,v) = |F(u,v)|^2$ = Power spectrum of the undergraded image.

Proof. We start by applying the proof for the 1-D domain, the 2-D domain is just an extension.

Recall that now we have the input image f, the degraded image g, we want to find y that satisfies

$$\min e^2 = \min\left\{(f-\hat{f})^2\right\} = \min\left\{(f-g*y)^2\right\}$$

where $*$ denotes convolution. By definition, the convolution $g(n)*y(n)$ is

$$g(n)*y(n) = \sum_{i=0}^{N} y(i)g(n-i)$$

which leaves to optimizing the following

$$E(y) = \left\{ \left(f(n) - \sum_{i=0}^{N} y(i)g(n-i) \right)^2 \right\}$$

Taking the derivative of $y(k)$ where $k \in [0,N]$, this leads to

$$\frac{\partial E(y)}{\partial y(k)} = 0 \Leftrightarrow -2E \left[g(n-k) \left(f(n) - \sum_{i=0}^{N} y(i)g(n-i) \right) \right] = 0$$

This is equivalent to

$$\sum_{i=0}^{N} y(i)E\left[g(n-i)g(n-k)\right] = E\left[f(n)g(n-k)\right]$$

Let $r_{fg}(k) = E\left[f(n)g(n-k)\right]$ be the cross-correlation between $f(n)$ and $g(n)$, and $r_{gg}(k-i) = E\left[g(n-i)g(n-k)\right]$ is the auto-correlation of $g(n)$. The equation above is equivalent to

$$\sum_{i=0}^{N} y(i)r_{gg}(k-i) = r_{fg}(k) \Leftrightarrow y(k) * r_{gg}(k) = r_{fg}(k)$$

Now take the DFT on both sides gives

$$Y(u) = \frac{S_{fg}(u)}{S_{gg}(u)}$$

From here on, we have

$$\begin{aligned}
g(n) &= h(n) * f(n) + \eta(n) \\
r_{gg}(k) &= g(k) * g(-k) \\
&= (h(k) * f(k) + \eta(k)) * (h(-k) * f(-k) + \eta(-k)) \\
&= h(k) * h(-k) * r_{gg}(k) + r_{\eta\eta}(k) \\
r_{fg}(k) &= f(k) * g(-k) \\
&= f(k) * (h(-k) * f(-k) + \eta(-k)) \\
&= h(-k) * r_{ff}(k)
\end{aligned}$$

Back to $Y(u)$, we now have

$$\begin{aligned}
S_{gg}(u) &= |H(u)|^2 S_{ff}(u) + S_{\eta\eta}(u) \\
S_{fg}(u) &= H^*(u) S_{ff}(u)
\end{aligned}$$

For simplicity, we let $S_f(u) = S_{ff}(u)$ and $S_\eta(u) = S_{\eta\eta}(u)$. We obtain $Y(u)$ as

$$Y(u) = \frac{H^*(u)S_f(u)}{S_f(u)|H(u)|^2 + S_\eta(u)}$$

Which completes the proof. □

It can easily be seen, that without the noise, this is a general formula for the inverse filtering, since the power spectrum of the noise equals to zero.

$$\hat{F}(x,y) = \frac{G(u,v)}{H(u,v)}$$

Below is the original image used for the experiment

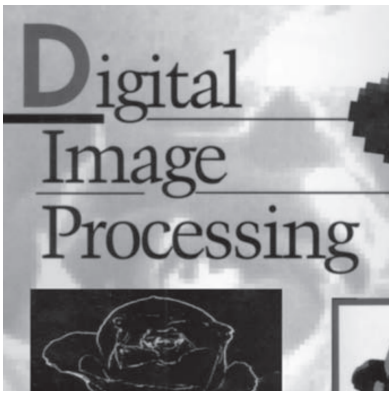

Figure 3.13: Book Cover **digitalimage**

Below are some results obtained from the experiments

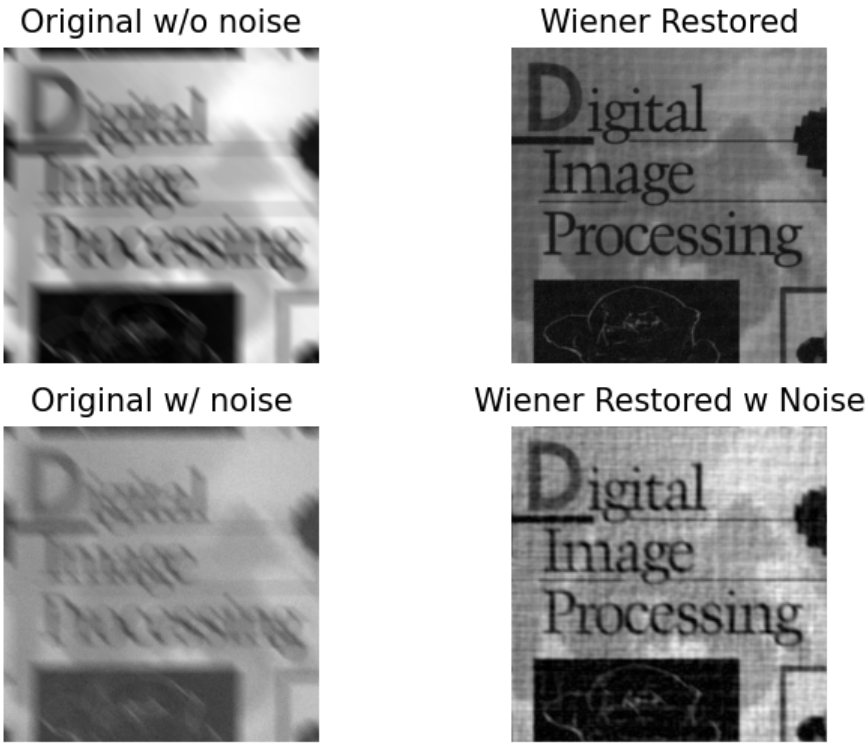

Figure 3.14: Wiener Filtering

The top-left image is degraded by using the same shift function discuss above, and this time, there is no noise added. As observed, the result is not as close, this is because the Wiener filter works best with images that contains noise. The left image on the second row shows a degraded image with noise, the its right is the result after applying Wiener filter. This result is much better, as it is more close to the original image.

Advantages of Wiener Filtering:

- **Noise Suppression.** Wiener filtering is more robust in the presence of noise, as it adapts to both the signal and noise characteristics in the image.

- **Optimal Restoration.** It minimizes the mean square error between the restored image and the original image, leading to better results in real-world scenarios with noise.
- **Edge Preservation.** The Wiener filter balances between noise reduction and preserving edges, making it suitable for noisy images where details should be retained.
- **Adaptability.** The filter adapts based on the local signal-to-noise ratio, making it more effective in heterogeneous environments (different levels of noise across the image).

3.2.4 Summary

We have discussed about the inverse and the Wiener filtering. The table below shows a short summary about when should we use each of the filter.

Criteria	**Inverse Filtering**	**Wiener Filtering**
Purpose	To restore an image degraded by a known blur.	To restore an image degraded by both blur and noise.
Degradation Model	Assumes exact knowledge of the degradation.	Requires estimation of both noise and degradation.
Robustness to Noise	Not robust to noise, can amplify noise.	Robust to noise, as it adjusts based on noise levels.
Restoration Quality	Performs well when the degradation is known, and there's minimal noise.	Performs better in noisy environments by suppressing noise and preserving edges.
Computational Complexity	Computationally simpler and faster.	More complex and computationally expensive due to power spectral density estimation.
Edge Preservation	Can result in loss of edges if the degradation is not precisely known.	Preserves edges while reducing noise, making it better for real-world images.
Practical Use	Works well when degradation is well-characterized, with little noise.	Preferred for real-world scenarios where noise and degradation are present.
Dependence on Parameters	Requires exact kernel information.	Requires estimation of noise and signal characteristics, which can be challenging.
Implementation	Simple and easy to implement.	More complex due to need for noise and signal modeling.
Application Scenario	Best for controlled conditions with little or no noise.	Best for noisy environments or when the degradation is unknown.

Table 3.6: Comparison of Inverse and Wiener Filtering

Chapter 4

Restoring A Blurred Image - A Use Case

4.1 Problem

4.1.1 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 $H(u,v)$.

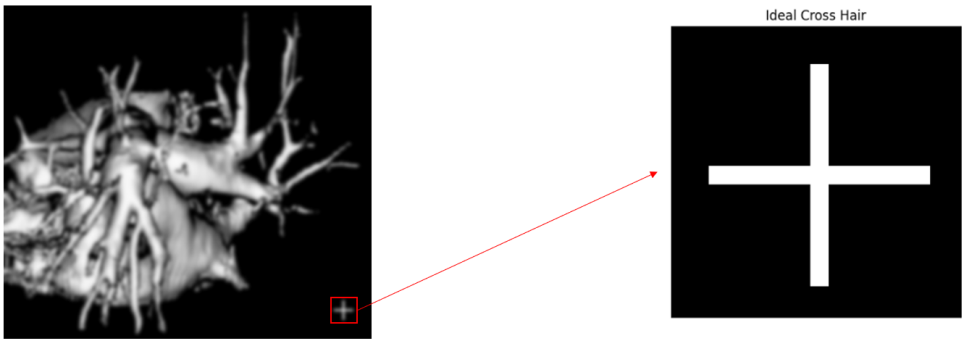

Figure 4.1: The blurred heart image **digitalimage** and the ideal crosshair image.

4.1.2 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.

4.1.3 Approach

We divide our approach into three main sections

1. Give some comments from observation and experimental.

2. Calculating the estimated degraded function based on comments above.

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

4.2 Comments

We state some comments that we have made to make our approach reasonable.

4.2.1 Frequency Domain Encoutered

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.

4.2.2 No Noise Corrupted

We proceed to segment a region in the image and observe its histogram.

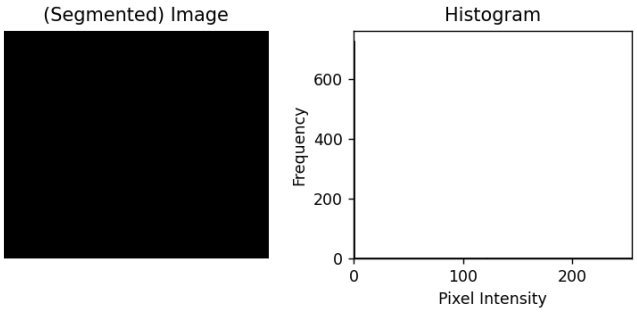

Figure 4.2: Histogram of a dark region in the blurred image

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.

4.2.3 Gaussian Filter Estimated

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.

As we know, the Fourier transform of the blurred image is obtained by $F_{\text{blurred}} = F_{\text{ideal}} \cdot H$, thus, an approach to obtain H is by using the equation

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

This is called the inverse filtering, as we mentioned above.

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.

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

Where ε is the estimated constant to make sure the denominator is non-zero. This is known as the Wiener filtering. Using this approximation, we can obtain the Fourier spectrum of the degraded function as below.

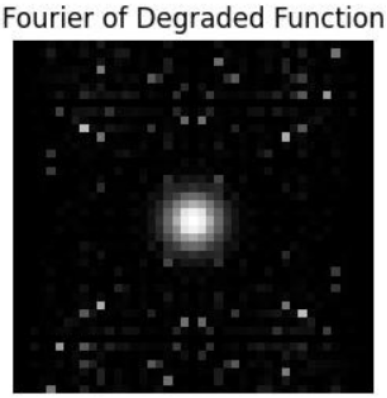

Figure 4.3: The Fourier spectrum of the degraded function

And as we might recall, this is very similar to the Fourier spectrum of the Gaussian filter function. We take a comparison of the Fourier spectrum of a Gaussian filter function with a cutoff frequency of $D_0 = 2.227$ and the degraded function above.

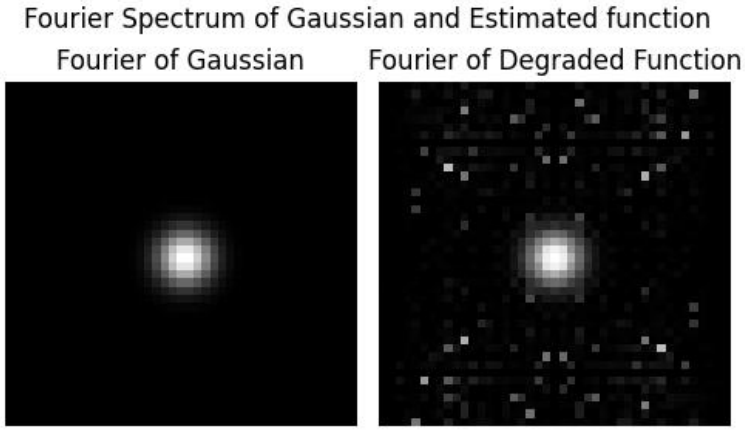

Figure 4.4: The Fourier spectrum of the Gaussian filter compared to the degraded function

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.

4.2.4 Summary

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.

4.3 Procedure

4.3.1 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.

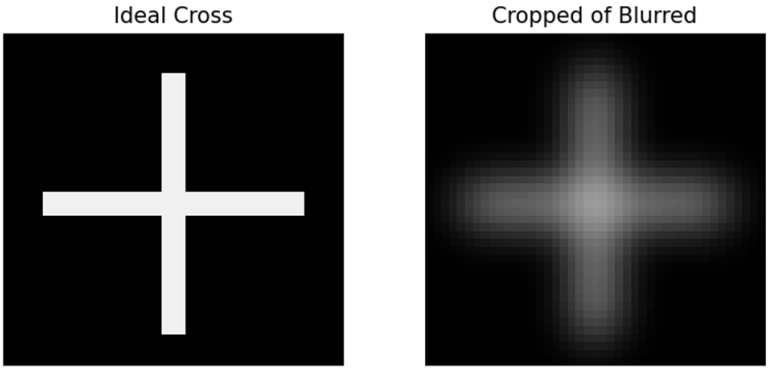

Figure 4.5: The cropped image compared to its original

4.3.2 Estimating Gaussian Filter

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

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

This reduces the problem to find D_0 that fits best our assumption. Now that we have the following equivalent

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

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 is the size of the crosshair blurred image. Some problem arises when using the approximation mentioned above

- **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.

- **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.

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.

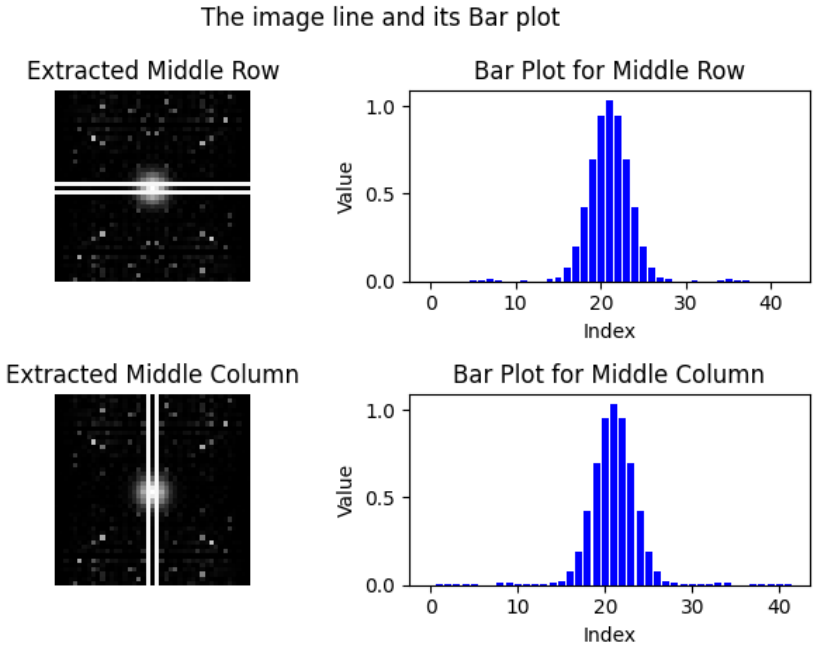

Figure 4.6: The middle lines and its bar plot

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

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.

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).

Let us do a walk-through example to understand how the procedure works. We first proceeds for the middle row, the middle column if proceeded similarly. The center in this case is at index 21.

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 it with the number of non zero values added.

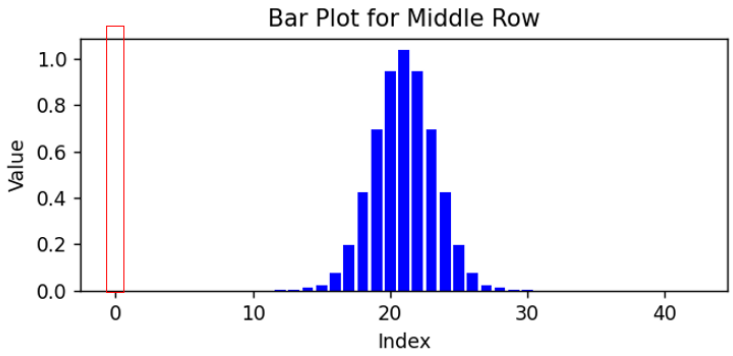

Figure 4.7: Index 0 of the middle row

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

We can skip this index, and move on to the next one. It is reasonable for us to skip forward to the 12^{th} index, since others are zero.

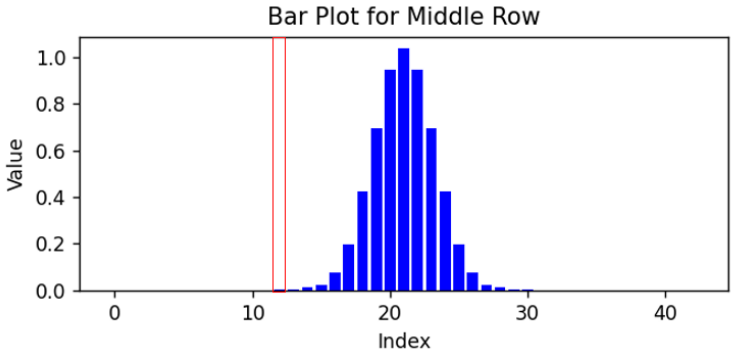

Figure 4.8: Index 12 of the middle row

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

- $D_0 : 2.393$

D_0 is updated at this step. Here we show how to process at the 13^{th} index, which is where there is an update in D_0.

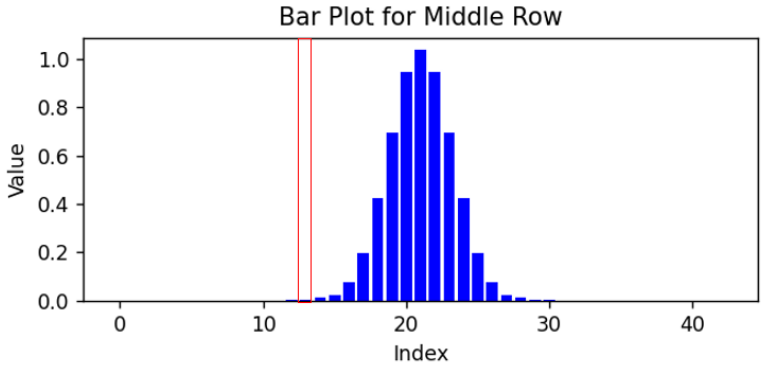

Figure 4.9: Index 13 of the middle row

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

D_0 is updated at this step. At the last index 30, we have successfully update D_0 from the middle row.

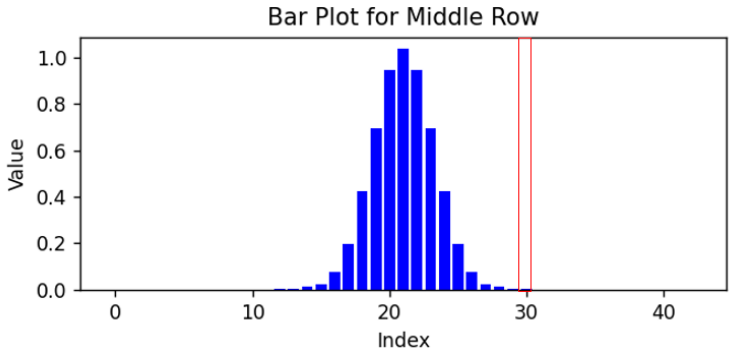

Figure 4.10: Index 30 of the middle row

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

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

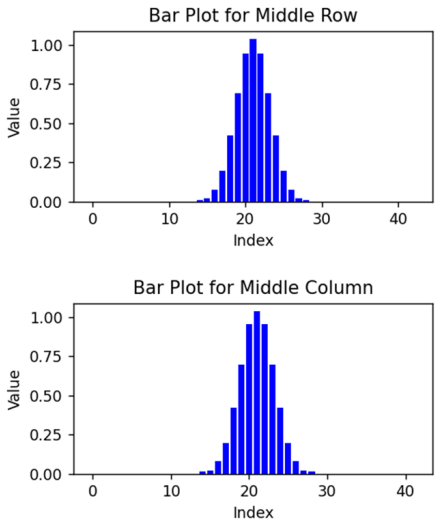

Figure 4.11: Final middle row and middle column

We summarize the values

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

Thus the final D_0 is

$$\hat{D}_0 = \frac{D_0}{\text{no\_non\_zero}} = \frac{74.74}{34} \approx 2.227$$

This is how we can obtain $D_0 = 2.227$ mentioned earlier.

4.3.3 Scale The Cutoff Frequency

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

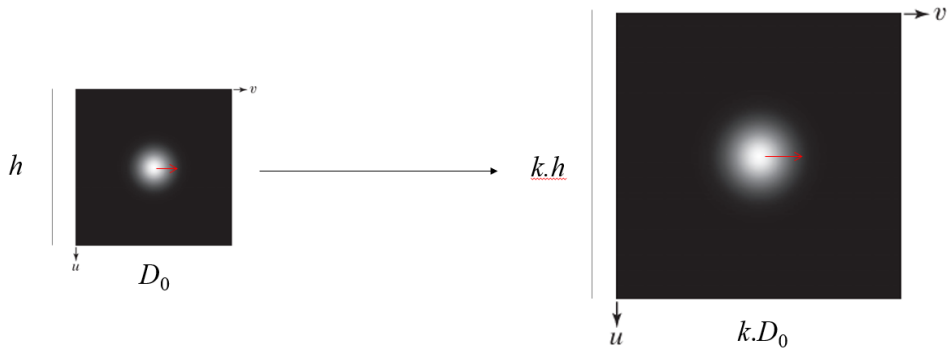

Figure 4.12: Scaling the cutoff frequency

Which means, to obtain the Gaussian filter H' for the full heart image, we can simply scale the cutoff frequency.

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

4.3.4 Restore Heart Image

As mentioned above, we do opt to use the Wiener inverse filter to restore the heart image, instead of using the direct inverse filtering. The Fourier transform of the restore heart image, denoted as F_{restored} is calculated as

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

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

We can proceed to apply the inverse Fourier transform to obtain the restored heart image.

4.3.5 Post Processing

We introduce two extra steps in enhancing the image quality, mostly for visualization purpose: brightening and sharpening.

Brightening

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

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

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

Sharpening

For sharpening, we opt to use the Laplacian sharpening kernel, and apply convolution to the restored image. The Laplacian we are using here have the form

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

The sharpened image $f_{\text{sharpened}}$ can be obtained as

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

Where $*$ denotes convolution.

To further enhance the contrast of the image, we opt to use the following kernel

$$L = \begin{bmatrix} 0 & -1 & 0 \\ -1 & \mathbf{6} & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Where the center value is 6 instead of 5, as we want to not only sharpen the image, but to also enhance its contrast.

Chan-Vese Segmentation

In order to assess the effectiveness of the Fourier-based image restoration, we employ **Chan-Vese active contour segmentation** as a post-processing step. The primary purpose of this method is to extract the main object from the restored image by partitioning regions of approximately homogeneous intensity. Unlike edge-based segmentation methods, Chan-Vese is capable of accurately delineating objects even in the presence of weak or diffuse boundaries, making it particularly suitable for evaluating restoration quality.

Mathematically, the Chan-Vese model seeks a contour C that minimizes the following energy functional:

$$F(c_1, c_2, C) = \mu \,\text{Length}(C) + \nu \,\text{Area}(\text{inside}(C)) + \lambda_1 \int_{\text{inside}(C)} |I(x,y) - c_1|^2 \, dxdy + \lambda_2 \int_{\text{outside}(C)} |I(x,y) - c_2|^2 \, dxdy$$

where $I(x,y)$ denotes the intensity at pixel (x,y), c_1 and c_2 represent the average intensities inside and outside the contour, respectively, and μ, ν, λ_1, λ_2 are weighting parameters that control the influence of contour smoothness, area, and region homogeneity. The algorithm iteratively updates a level-set function to evolve the contour C towards a minimum of this functional.

The expected output of this procedure is a binary mask in which the object of interest is clearly delineated from the background. When applied to the restored images, Chan-Vese segmentation produces neat and well-defined contours, thereby providing a visual confirmation of the restoration performance and highlighting the regions that have been effectively recovered from blurring.

4.3.6 Summary

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, sharpening and segmentation the recovered image for visualization.

4.4 Experimental Results

4.4.1 Parameters Setting

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

Table 4.1: Parameters for for functions mentioned above

4.4.2 Image Results

Crosshair Restored
The image below shows a side-by-side comparison between the input blurred image and the restored image.

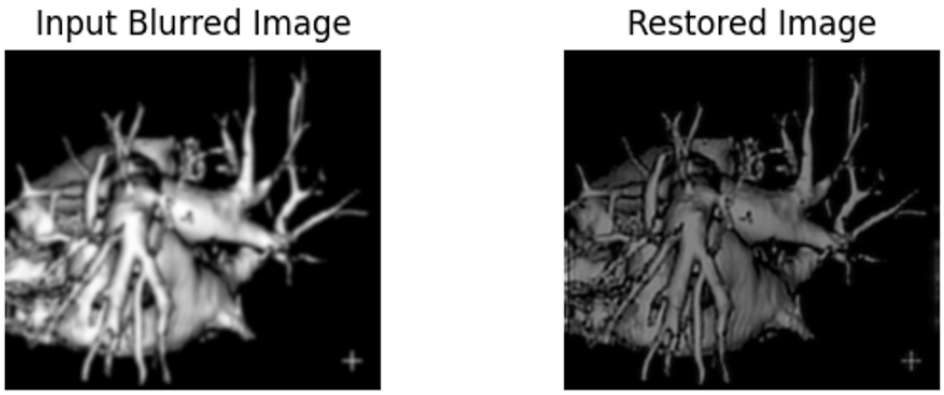

Figure 4.13: Side by side comparison of the input and the restored heart image

We also crop out the crosshair section to verify the correctness of our function.

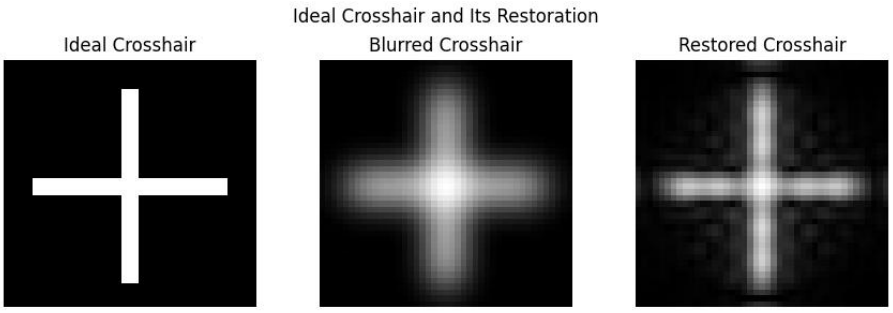

Figure 4.14: The ideal, blurred and restored crosshair image

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).

From this observation, we conclude that our Gaussian filter function is acceptable to be treated as the degraded function.

Crack Restored
Below is another region which we chose to compare, which is a crack in the heart image. We chose this region as it contains some of the most complicated information.

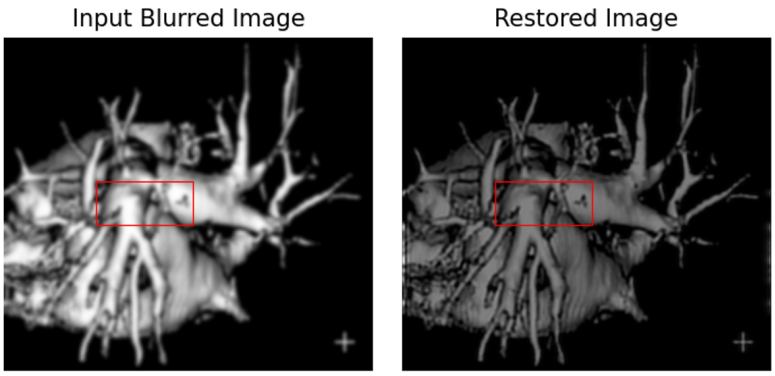

Figure 4.15: The cracked region

Below are its zoom in images for better visualization.

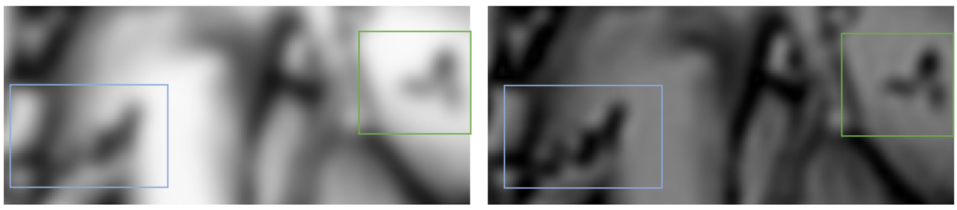

Figure 4.16: Crack region in the input blurred image and the restored image

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

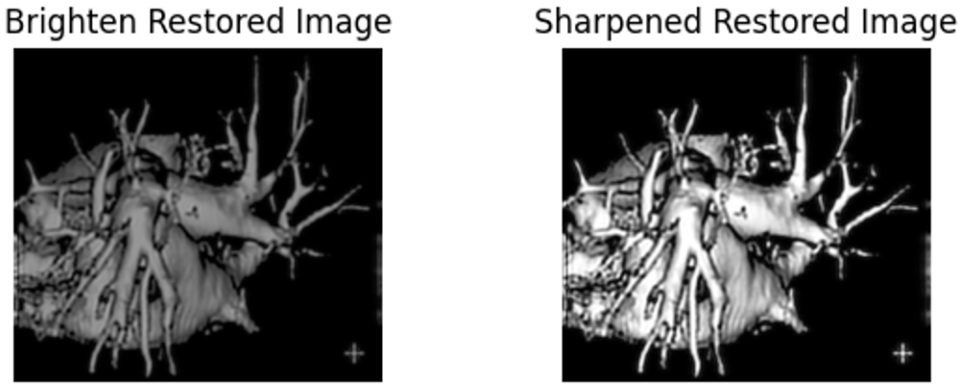

Figure 4.17: Brightened and sharpened image

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.

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

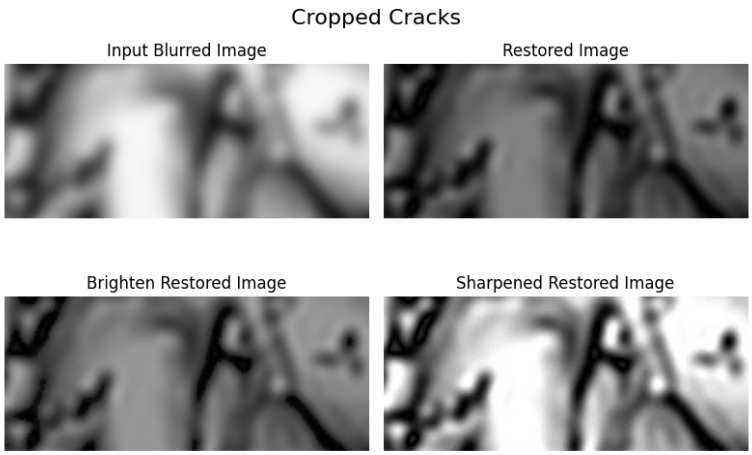

Figure 4.18: Crack areas in four images

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, this helps clarify the shapes and boundaries, offering a clearer understanding of the original image's features.

Chan-Verse Segmentation Visualization

As mentioned above, we use Chan-Verse segmentation on four images to visualize the performance of our approach. We first visualize the segmentation on all four images.

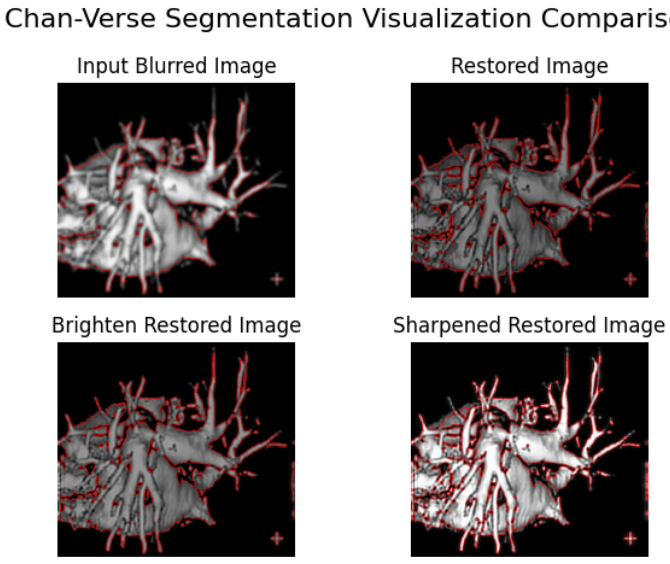

Figure 4.19: Four images segmented with Chan-Verse Segmentation

As observed, the contours extracted from the blurred image exhibit reduced accuracy, with boundary locations appearing diffuse and poorly defined. In contrast, the restored image demonstrates substantially improved line separation, and the corresponding contours are sharper and more clearly delineated.

We also visualize the lines on the four cracks.

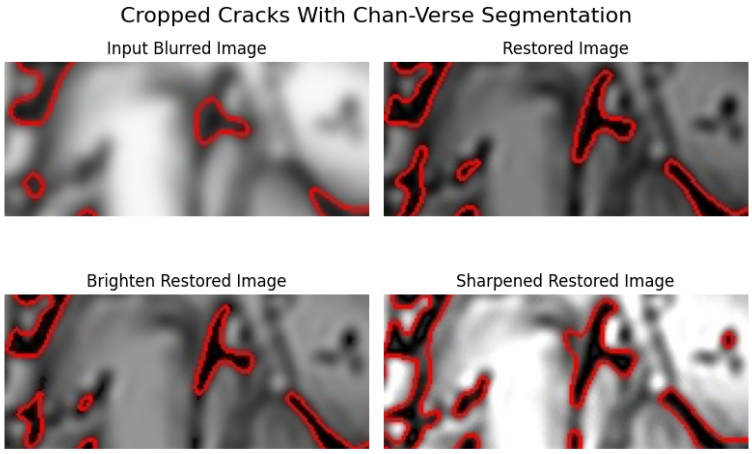

Figure 4.20: Crack areas in four images segmented with Chan-Verse Segmentation

The final sharpened image provides the most effective segmentation, with the separation boundaries clearly distinguishable. This enhanced delineation reveals substantially more structural information within the image.

4.4.3 Summary

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.

4.5 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.

The code for this project can be found on GitHub at https://github.com/thaiquangphat/Image-Restoration-in-Frequency-Domain.

4.6 Limitations and Future Work

4.6.1 Limitations

We encounter the following limitation while solving the problem.

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.

4.6.2 Future Work

We see some potential for the solution to be improved, some of which includes

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.

Chapter 5

Conclusion

We have conducted an in-depth study of the mathematical concepts underlying the Fourier transform, exploring its fundamental principles, properties, and a variety of applications, particularly in the field of image processing. This exploration covered the theoretical foundation of the Fourier transform, including how it decomposes signals into their frequency components and how this powerful tool can be leveraged to analyze and manipulate images in both spatial and frequency domains.

When it comes to image restoration in the frequency domain, several essential techniques are required. These include filtering, which involves modifying specific frequency components to enhance or suppress certain image features, and more advanced restoration methods that aim to reconstruct images degraded by various factors. Filtering in the frequency domain relies on the application of filters such as low-pass, high-pass, or band-pass filters, tailored to address specific degradation issues. These techniques are often complemented by mathematical restoration methods designed to correct distortions or noise in an image.

To better understand the practical application of Fourier transform in image processing, we considered a use-case problem focused on recovering a blurred image. Through this experiment, it became evident that achieving effective image restoration requires a deep understanding of the noise characteristics within the image, as well as precise knowledge of the mathematical form of the degradation function. This includes identifying whether the degradation is caused by motion blur, lens aberrations, or other factors, and formulating appropriate corrective measures. The experiment underscored the importance of combining theoretical knowledge with practical implementation to address complex image restoration challenges effectively.

References

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing, Fourth Edition*. Pearson, 2018.

[2] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series, with engineering applications*, 1949.

[3] M. A. Kutay and H. M. Ozaktas, *Optimal image restoration with the fractional fourier transform*, Computer Standards & Interfaces, vol. 20, pp. 452–453, 1998.

[4] A. Hillery and R. T. Chin, *Iterative wiener filters for image restoration*, Signal Processing, IEEE Transactions on, vol. 39, pp. 1892–1899, Sep. 1991.

[5] B. R. Hunt, *The application of constrained least squares estimation to image restoration by digital computer*, IEEE Transactions on Computers, vol. C-22, pp. 805–812, 1973.

[6] S. Reddi, *Constrained least-squares image restoration: An improved computational scheme*, Applied Optics, vol. 17, pp. 2340–2341, Aug. 1978.

[7] N. Galatsanos and A. Katsaggelos, *Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation*, Image Processing, IEEE Transactions on, vol. 1, pp. 322–336, Aug. 1992.

[8] A. Katsaggelos, J. Biemond, and R. Schafer, *A regularized iterative image restoration algorithm*, Signal Processing, IEEE Transactions on, vol. 39, pp. 914–929, May 1991.

[9] V. Mesarovic, N. Galatsanos, and A. Katsaggelos, *Regularized constrained total least squares image restoration*, IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, vol. 4, pp. 1096–108, Feb. 1995.

[10] W. H. Richardson, *Bayesian-based iterative method of image restoration∗*, J. Opt. Soc. Am., vol. 62, no. 1, pp. 55–59, Jan. 1972.

[11] L. B. Lucy, *An iterative technique for the rectification of observed distributions*, The Astronomical Journal, vol. 79, pp. 745–754, 1974.

[12] D. Biggs and M. Andrews, *Acceleration of iterative image restoration algorithms*, Applied Optics, vol. 36, pp. 1766–1775, Mar. 1997.

[13] *Nonlinear filtering of multiplied and convolved signals*, IEEE transactions on audio and electroacoustics, vol. 16, no. 3, pp. 437–466, 2003.

[14] H. Adelmann, *Butterworth equations for homomorphic filtering of images*, Computers in Biology and Medicine, 1998.

[15] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[16] T. S. Huang, Ed., *Picture Processing and Digital Filtering*. Berlin: Springer-Verlag, 1975.

[17] K. R. Castleman, *Digital Image Processing*. Upper Saddle River, NJ: Prentice Hall, 1996.

[18] B. Osgood, *Lecture Notes for EE 261 The Fourier Transform and its Applications*. Stanford University, 2007, Available at: https://see.stanford.edu/materials/lsoftaee261/book-fall-07.pdf.