MANIPAL
ACADEMY of HIGHER EDUCATION
(Institution of Eminence Deemed to be University)

MDS6401 / Segment 01

IMAGE REPRESENTATION AND ENHANCEMENT – PART II

Table of Contents

Introduction

Enhancing digital images through frequency domain processing is crucial for various applications in digital image processing (DIP). Images are transformed from spatial to frequency using the Fourier Transform, we achieve efficient filtering, improved image compression, and effective pattern recognition. This topic will explore the necessity of frequency domain processing, the basics of image enhancement, spatial domain limitations, the concepts of Fourier series, and the real-time applications of different types of filters.

Learning Objectives

At the end of this topic, you will be able to:

- Explain the basics of image enhancement from the frequency domain perspective
- Illustrate the spatial domain techniques and their limitations
- Outline the concepts and working of the Fourier series
- List the various types of low-pass filters and high-pass
- Explain the various real-time applications of filters

1. Enhancing Digital Images: The Necessity of Frequency Domain Processing

To comprehend the necessity of image enhancement and the relevance of the frequency domain, it is imperative to address why reliance solely on the spatial domain is insufficient. Firstly, exploring the reasons behind frequency domain image processing is essential. In Digital Image Processing (DIP), transforming an image from the spatial domain to the frequency domain using mathematical transformations, such as the Fourier Transform, is crucial. The Fourier Transform is a well-known method that we utilise to understand the advantages of frequency domain processing.

One of the primary reasons for using frequency domain image processing is the ease of filtering. Regarding time complexity, filtering operations are significantly more efficient in the frequency domain than in the spatial domain. In the spatial domain, the time complexity of operations increases dramatically as the image size grows. For instance, in computer science, the time complexity of spatial domain operations can escalate polynomially, which is evident in vision transformers. When processing images, these transformers face increased computational demands, hence the development of patch-based transformers that reduce the time complexity.

The second key advantage is image compression. The frequency domain allows for the efficient separation of significant signal components from noise. While it may not completely prevent noise, it offers better relative performance than the spatial domain in noise reduction. This is achieved by isolating and removing noise components to a certain extent.

Furthermore, pattern recognition benefits from frequency domain processing. In some scenarios, spatial domain differentiation is insufficient, necessitating a transformation to another domain to identify patterns. Although frequency domain transformation is not typically used in deep learning, transforming data to different dimensional spaces helps in pattern differentiation, particularly in classification tasks.

Lastly, image enhancement is more effective in the frequency domain. Operations such as sharpening and blurring are more efficiently performed. Advanced techniques benefit significantly from the frequency domain approach, including edge detection and object identification in images.

2. Challenges in the Spatial Domain

The spatial domain presents several significant challenges, particularly regarding computational complexity. When applying a customised filter to an entire image, the filter must process each pixel individually. This procedure increases the time complexity exponentially, posing substantial computational difficulties. In modern research, particularly in deep learning and machine learning, the efficiency of models concerning time and space complexity is crucial. A model must be more efficient to ensure its practical applicability and the likelihood of publication in scholarly papers are maintained.

Another major issue in the spatial domain is edge artefacts. Artefacts are imperfections or distortions that appear in images due to defects in the imaging process. These may display unexpected white patches or other anomalies, indicating issues with image acquisition. Effective filtering to remove these artefacts is essential for maintaining image quality.

Moreover, the design of filters in the spatial domain is limited and complex. Crafting effective filters for image processing tasks is more challenging in the spatial domain than in the frequency domain, where the process is more straightforward.

Performance issues for specific tasks also affect the spatial domain. Some image processing tasks may result in the amplification of noise, leading to poorer performance. The spatial domain's inability to adequately handle these tasks underscores the frequency domain's superiority in image enhancement.

3. Concept of Fourier Transformation

To understand the image enhancement process through the frequency domain, we employ the Fourier Transform or Fourier Series, a concept introduced by the mathematician John

Fourier. Initially dismissed for its complexity, Fourier's work was recognised and formalised, becoming a cornerstone in signal processing.

Fourier's fundamental contribution is that any univariate function involving a single-dimensional feature can be decomposed into a weighted sum of sine and cosine functions with different frequencies. Any periodic function may be represented as sines and cosines multiplied by a weighting function coefficient.

This principle is analogous to mathematical approximations like the Maclaurin or Taylor series, where complex functions are represented as the sum of simpler base functions. Regardless of a function's complexity, it can be defined as a sum if it is periodic and meets certain conditions.

Consider the Figure 1. The image on the left shows a complex waveform that appears as a single, superimposed wave. Fourier Transform lets us decompose this complex wave into its constituent sine and cosine components, each with specific frequencies and amplitudes.

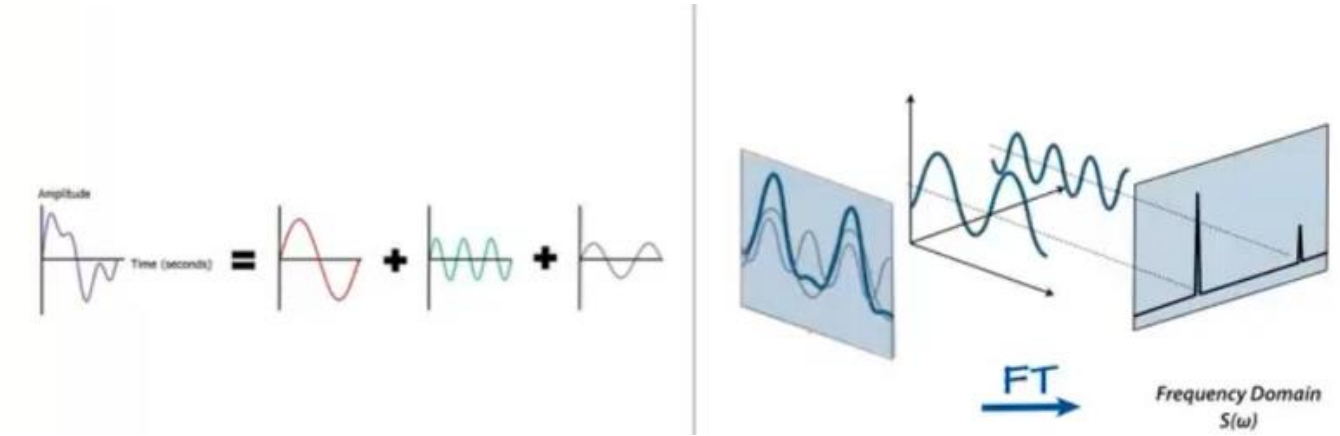


Figure 1: Decomposing Complex Signals: Understanding Fourier Transform in Frequency Domain Analysis

Imagine you have a smoothie containing various ingredients. Typically, you consume it as a single blend. However, if you were provided with a breakdown of each ingredient and its quantity, you could understand the composition better. Similarly, the Fourier Transform delivers a breakdown of a complex wave into its simpler components.

In Figure 1, the frequency domain representation (right side) shows distinct spikes, each representing a frequency component and its amplitude. These spikes indicate the frequencies used and their respective amplitudes, which, when combined, form the complex waveform shown on the left side of the image.

Thus, the Fourier Transform dissects a unified, superimposed signal into its parts, offering a detailed understanding of the signal's composition. This decomposition is crucial for various applications in image processing, such as filtering, noise reduction, and pattern recognition, where analysing the frequency components can lead to more efficient and effective results.

3.1 Basic Building Blocks

In the context of the Fourier Series, it is essential to understand the basic building blocks, mainly the sine function, represented as $A\,sin(\omega x + \phi)$. This function converts a signal into a periodic function.

- Amplitude (A): The amplitude represents the height of the wave, indicating the signal's strength or intensity.
- Angular Frequency (ω): This parameter describes how many oscillations occur in a unit time period, dictating the wave's frequency.
- Phase (ϕ): The phase indicates the horizontal shift of the wave, specifying where the wave starts within its cycle.
- Variable (x): This represents the independent variable, often time or spatial coordinates, over which the function is defined.

These components collectively define the behaviour and characteristics of the periodic function, allowing us to decompose complex signals into simpler sinusoidal components for analysis and processing.

4. Time and Frequency Domain Comparisons

Figure 2 illustrates fundamental comparisons between various signals' time and frequency domain representations. Each time domain signal has a corresponding alternative form in the frequency domain, offering a different perspective for analysis.

Consider this comparison as similar to the concept of alternate universes depicted in popular culture, such as the show "Stranger Things." In this analogy, the time domain represents the familiar world, while the frequency domain represents an alternate universe where the signals exhibit different characteristics.

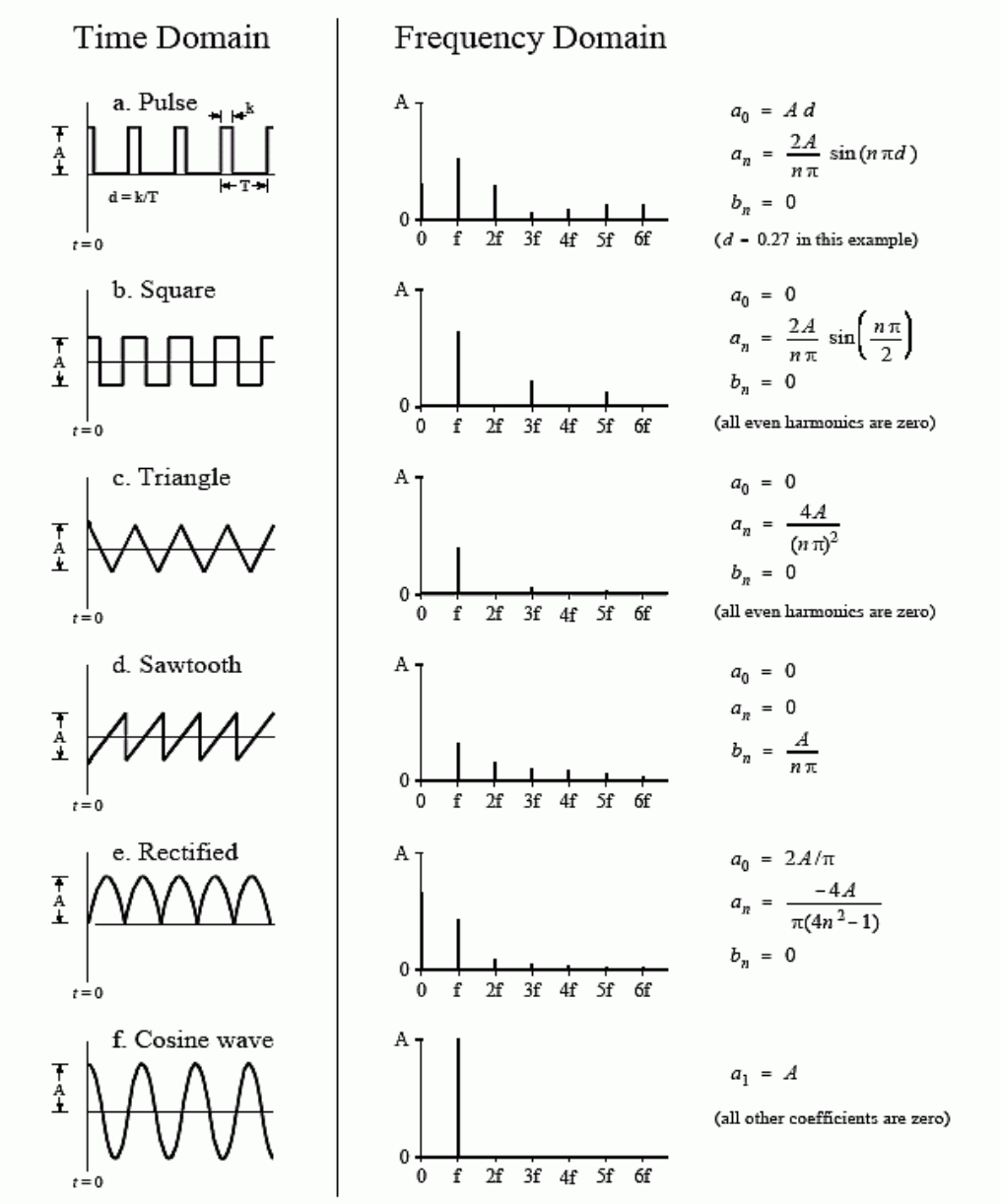


Figure 2: Time and Frequency Domain Comparison

The image presents several examples:

- Pulse (a): In the time domain, a pulse is represented by a sharp, discrete signal. The frequency domain decomposes it into a series of sine and cosine functions with different frequencies and amplitudes.
- Square Wave (b): A square wave in the time domain is characterised by periodic, binary state changes. Its frequency domain representation includes the fundamental frequency and odd harmonics, with amplitudes decreasing as frequency increases.
- Triangle Wave (c): The triangle wave, known for its linear rise and fall in the time domain, is represented in the frequency domain with odd harmonics, where amplitudes decrease quadratically with frequency.
- Sawtooth Wave (d): The sawtooth wave, exhibiting a linear rise followed by a sudden drop in the time domain, has a frequency domain representation that includes all harmonics with amplitudes decreasing linearly.
- Rectified Sine Wave (e): A rectified sine wave, which only includes positive half-cycles of a sine wave, shows a combination of fundamental and even harmonics in the frequency domain.
- Cosine Wave (f): A cosine wave, a fundamental sine function shifted in phase, is represented in the frequency domain by a single frequency component at its fundamental frequency.

The Fourier series approach allows any time domain signal to be represented as a sum of sinusoidal components at various frequencies. This transformation facilitates a different and often more insightful analysis of the signal's characteristics and behaviour.

5. Analysing Composite Waveforms through Fourier Series

One can determine the constituent waveforms within a single complex waveform by utilising the Fourier series.

Decomposing a Complex Waveform

Figure 3 aims to identify the components of the given complex waveform. The top image shows a complex wave; the challenge is finding its constituent sine waves. It becomes evident that the waveform is a superposition of multiple sine waves.

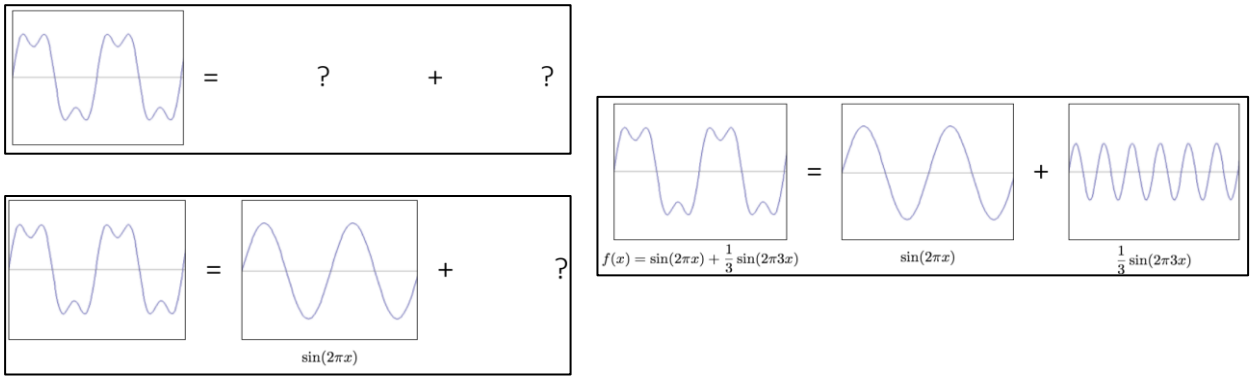


Figure 3: Decomposing Complex Waveforms into Sine Wave Components

Upon decomposition, we see that the waveform is formed by superimposing a primary sine wave, $sin(2\pi x)$, and another sine wave, $\frac{1}{3}sin(2\pi 3x)$, with a higher frequency. These two waves combine to create the observed complex waveform. This demonstrates how a composite wave can be broken down into simpler sine waves of different frequencies and amplitudes.

Constructing a Square Wave

Figure 4 depicts the process of constructing a square wave from sinusoidal components:

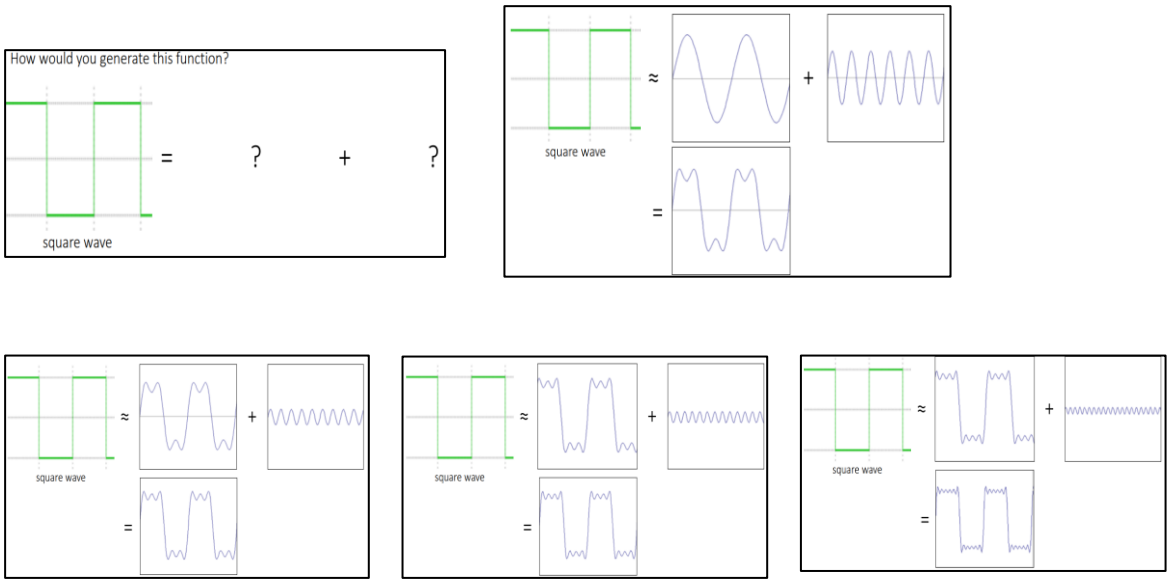


Figure 4: Progressive Construction of Square Waveform

In these waveforms, we start with a simple sinusoidal wave and incrementally add higher frequency sine waves to approximate a square wave. By adding sine waves with odd harmonics and decreasing amplitudes, we move closer to the desired square waveform.

For instance, the initial sine wave might be $sin(2\pi x)$. Adding another sine wave with a higher frequency, such as $\frac{1}{3}sin(2\pi 3x)$, and continuing this process in a more balanced way, we refine the waveform. Each added sine wave brings us closer to the square wave, as shown in Figure 4.

Infinite Sum Representation

Figure 5 illustrates how a square wave can be represented as an infinite sum of sine waves:

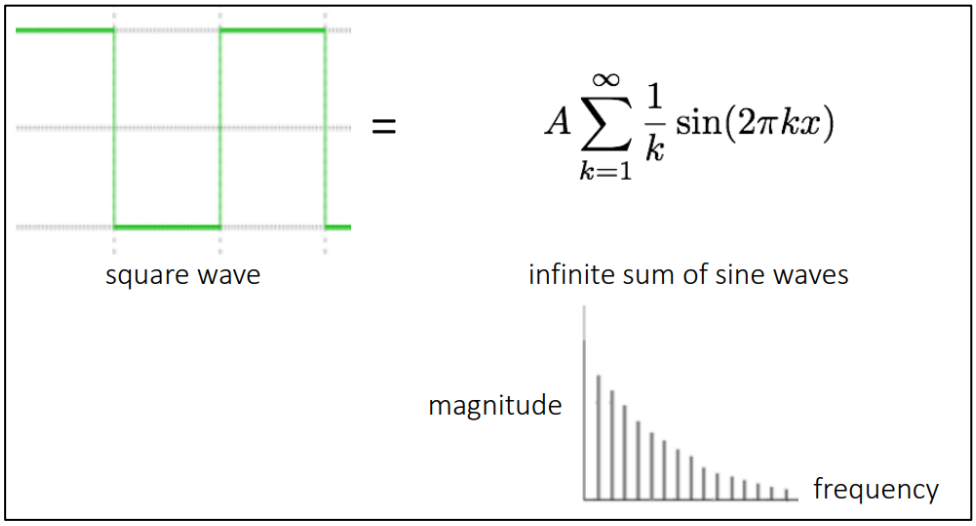


Figure 5: Infinite Sum of Sine Wave

In these representations, the square wave is depicted as an infinite series:

$$A\sum_{k=1}^{\infty}\frac{1}{k}sin(2\Pi kx)$$

Here, A represents the amplitude, and k denotes the harmonic number. The series converges to the square wave as more terms are added. The amplitude of each harmonic decreases as the frequency increases, eventually approaching zero but theoretically never quite reaching it. This concept resembles an asymptotic function where the amplitude diminishes towards zero at infinity.

The graph in Figure 5 visually demonstrates this concept, plotting magnitude against frequency. As more sine wave components are added, the waveform approximates the square wave more closely.

6. Working of Fourier Transformation

The Discrete Fourier Transform (DFT) is a discretised version of the Fourier series. Unlike the continuous Fourier transform, which includes all frequencies that form an image, the DFT samples the Fourier transform, capturing a set of frequency samples sufficient to accurately describe the spatial domain image.

As previously mentioned, the continuous Fourier transform theoretically extends to infinity, providing an exact representation of the waveform or image. However, In actual applications, we use a limited number of samples. The frequency samples correspond to the number of pixels in the spatial domain image, allowing us to reconstruct the image accurately.

The formula for the two-dimensional DFT of an $n \times n$ image is given by:

$$F(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) e^{-\iota 2\pi \left(\frac{ki}{N} + \frac{lj}{N}\right)}$$

In this formula:

- $f(i,j)$ represents the image in the spatial domain.
- The exponential term $e^{-\iota 2\pi \left(\frac{ki}{N} + \frac{lj}{N}\right)}$ is the basis function corresponding to each point $F(k,l)$ in the Fourier space.
- The double summation accounts for the contributions of all pixels (i,j) in the spatial domain.

For a square image of size $n \times n$ the two-dimensional DFT involves summing over all pixel values, weighted by the complex exponential functions (basis functions). These basis functions are periodic and can be represented by sine and cosine functions.

The key points to understand are:

- **Double Summation:** The summation is performed twice, once for each spatial dimension i and j, to account for the two-dimensional nature of the image.
- **Basis Functions:** The exponential terms serve as basis functions for the transformation, determining how each frequency component contributes to the overall image.

By using DFT, we convert the image from spatial to frequency, where each point $F(k,l)$ represents a specific frequency component of the original image. This transformation facilitates various image-processing tasks, such as filtering and image enhancement.

Although the DFT is computationally intensive due to the double summation, it provides a powerful tool for analysing and processing images in the frequency domain.

6.1 Challenges in Applying the Fourier Transform

Applying the Fourier Transform, especially for larger images, presents significant challenges due to its computational complexity.

Computational Complexity

- The Fourier Transform can still be computed reasonably well for small images, such as 64x64 or 128x100 pixels. However, the complexity of this operation is $O(n^2)$, where n is the number of pixels in one dimension.
- The $O(n^2)$ complexity arises because the transformation involves double summation over all pixels in the image, resulting in a quadratic growth of computational requirements.
- $O\ (n^2)$ complexity is considered high and inefficient for larger images, making the brute force application of the Fourier Transform impractical.

Reduction of Complexity with Fast Fourier Transform (FFT)

- Computational complexity can be greatly lowered to $O(n\log n)$ by using the Fast Fourier Transform (FFT) algorithm. This reduction is crucial for practical applications, as it drastically lowers the time required for computation.

- $O(n\log n)$ is one of the most efficient complexities achievable for such algorithms, comparable to the best-known sorting algorithms. The FFT makes it feasible to apply Fourier Transforms to large images efficiently.

Without employing FFT, performing Fourier Transforms on large images would be computationally prohibitive due to the $O(n^2)$ complexity. Thus, the FFT is essential for making the application of Fourier Transforms practical in real-world scenarios, providing a much faster and more efficient means to perform the necessary computations.

7. Fast Fourier Transform

The popularity of Fourier-based techniques has surged due to the development of the Fast Fourier Transform (FFT). FFT performs the Fourier Transform approximately 100 to 600 times faster than the traditional, basic implementation. This speed is crucial as high-resolution images, with more pixels, make the algorithm computationally expensive. FFT optimises this process, making complex calculations more manageable and efficient.

Fourier Transform Output

The Fourier Transform of an image produces complex numbers, resulting in two components: the real part and the imaginary part, often referred to as the magnitude and phase of the image, respectively. In image processing, the magnitude usually contains most of the information, and thus, it is often displayed. However, to re-transform a processed Fourier image back into the spatial domain correctly, both the magnitude and phase must be preserved.

Visualising Magnitude and Phase

Figure 6 depicts the application of Fourier transforms.

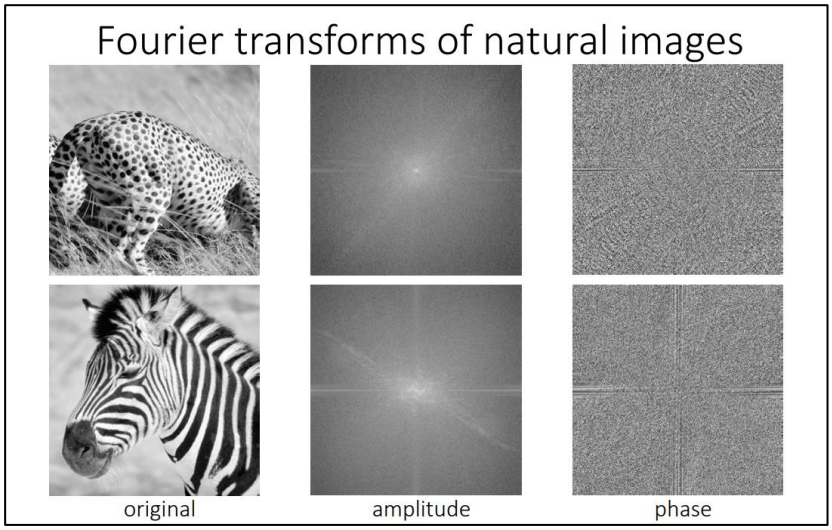


Figure 6: Fourier Transforms on Images

- Original Images: The first column shows the original natural images.
- Amplitude: The second column displays the amplitude (magnitude) of the Fourier Transform.
- Phase: The third column represents the phase component.

Visually, the amplitude and phase images might seem obscure and difficult to interpret directly. The central part of the amplitude image shows a concentration of frequencies that form the original image. The phase image, though seemingly chaotic, contains crucial structural information about the image.

Importance of Phase in Image Reconstruction

To emphasise the importance of the phase, consider the following experiment as depicted in Figure 7:

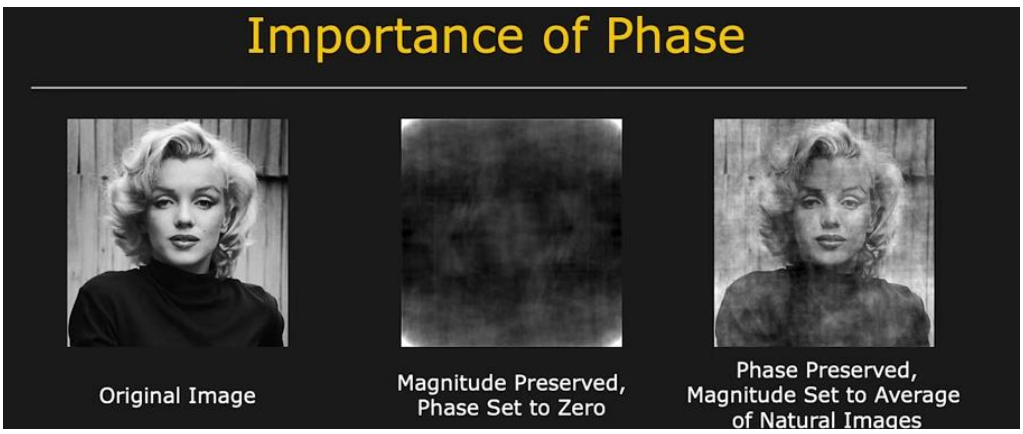


Figure 7: Importance of Phase

- Original Image: The first image is the original.
- Magnitude Preserved, Phase Set to Zero: The second image retains the magnitude while setting the phase to zero. The resulting image becomes unrecognisable.
- Phase Preserved, Magnitude Averaged: The third image retains the phase but averages the magnitude. Despite the smoothing of the magnitude, the image remains recognisable.

This experiment highlights that preserving the phase is essential for maintaining the structural integrity of the image. Without the phase, even with the correct amplitude, it becomes challenging to identify the image accurately.

8. Visualising Frequency Spectrum

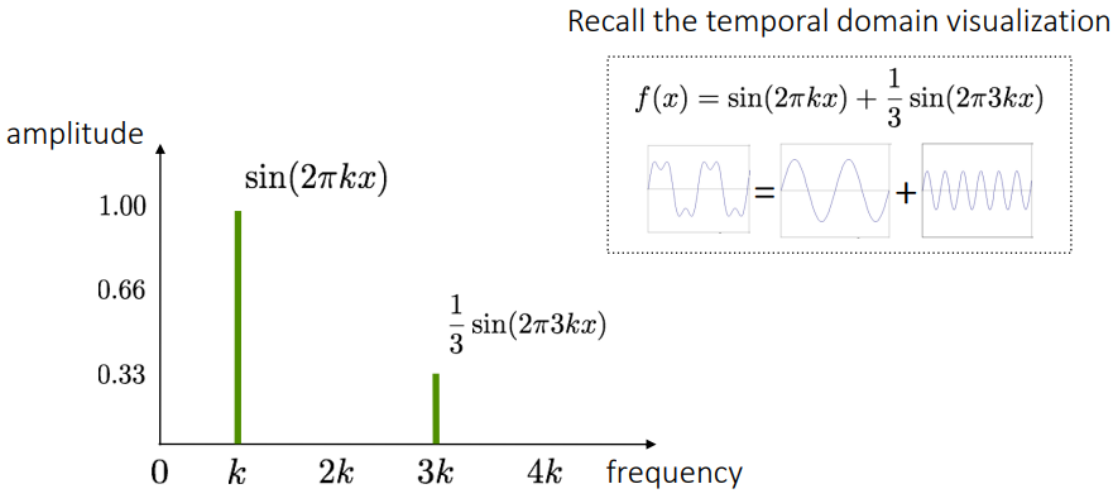


Figure 8: Temporal Domain Visualisation- Decomposing a Function into Frequency Components

In Figure 8, f(x) represents our final image, which can be considered as a combination of functions, such as $\sin(2\pi kx)$ and $\frac{1}{3}sin(2\pi 3kx)$. When visualised in the frequency domain, we observe a histogram where the x-axis denotes frequency and the y-axis represents amplitude.

- **Amplitude Peaks:** The first function, $\sin(2\pi kx)$, has an amplitude of 1, represented by the peak at frequency k. The second function, $\frac{1}{3}sin(2\pi 3kx)$, has an amplitude of 0.33,

indicated by the lower peak at frequency $3k$. This visualisation helps us identify the constituent frequencies and their contributions to the final waveform.

Frequency Domain Visualisation in 1D and 2D

The application of frequency domain analysis can be extended to both one-dimensional (1D) and two-dimensional (2D) data:

- **1D Visualisation:** The top part of the image shows a single sine wave in the spatial domain and its corresponding frequency spectrum with a single frequency peak.
- **2D Visualisation:** The bottom part illustrates a 2D spatial domain image with vertical lines and its frequency domain representation. The frequency domain image shows three main dots, with the central dot representing the origin. The dots on either side represent the frequency components and their mirror images.

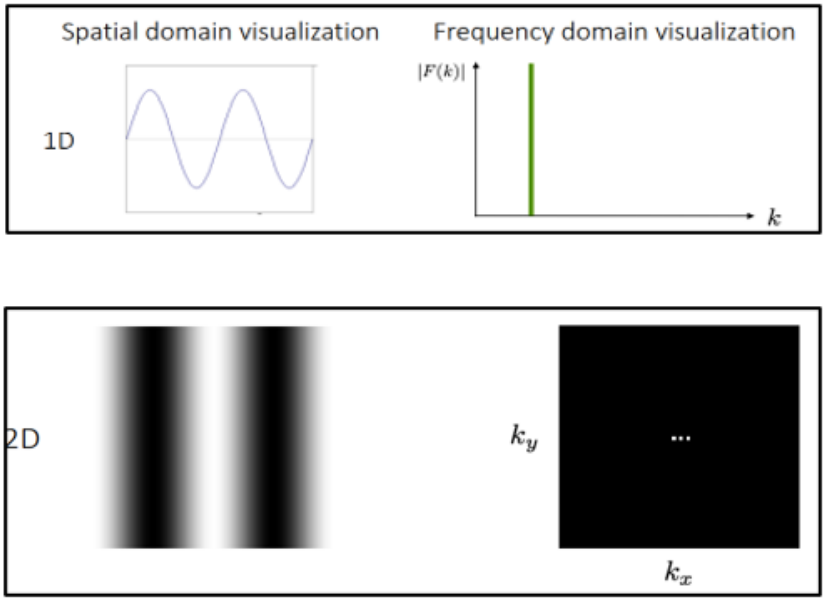


Figure 9: Spatial and Frequency Domain Visualisation in 1D and 2D

Increased Frequency and Spacing in 2D Visualisation

As we examine more complex images, the spacing of the frequency components changes, indicating variations in frequency as depicted in Figure 10:

- **Higher Frequency:** The image shows vertical lines with increased frequency, resulting in more spaced-out dots in the frequency domain. This indicates that the image contains higher frequency components.

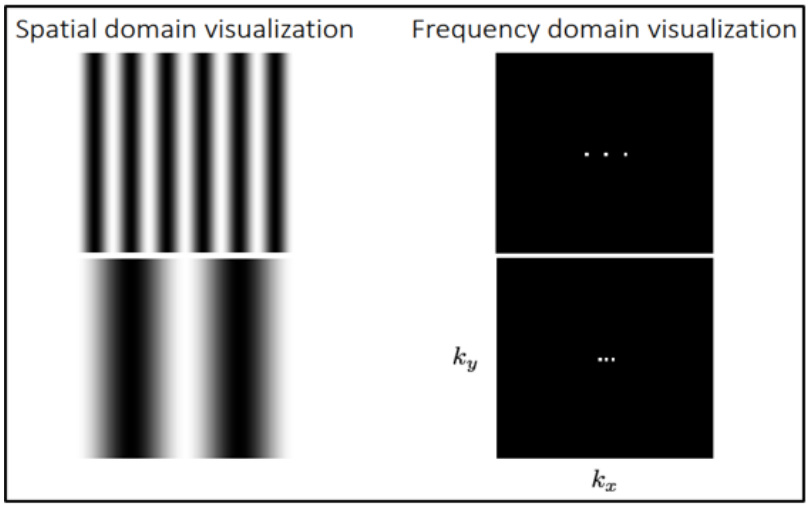


Figure 10: High-Frequency Components in Spatial and Frequency Domain Visualisation

Frequency Representation of Cosine-like Images

Figure 11 shows how the Fourier Transform sums all images as cosine-like images:

- **Pure Cosines and Image Center:** The image of pure cosines helps visualise the horizontal and vertical frequency components. The centre of the image represents the origin, with high frequencies in the vertical direction causing bright dots away from the center. Similarly, high frequencies in the horizontal direction cause bright dots away from the centre horizontally.

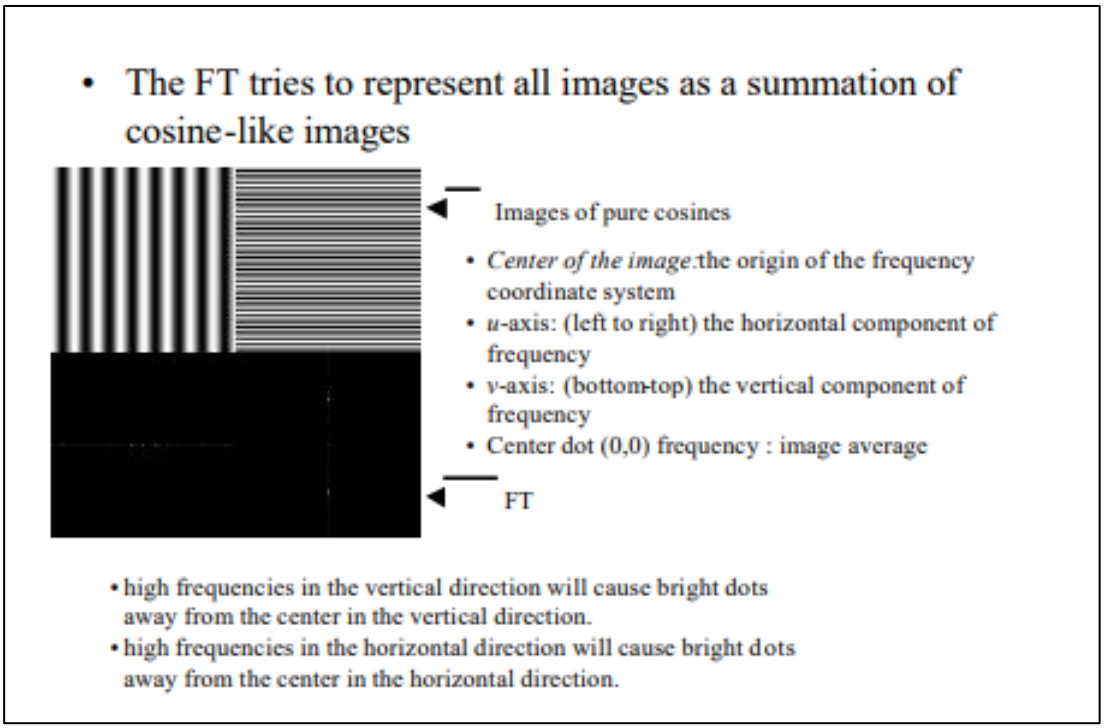


Figure 11: Summation of Cosine-like Images

Understanding Frequency Components

Analysing these images helps us understand the relationship between spatial domain features and their frequency domain counterparts:

- **Vertical Edges:** Vertical edges in the spatial domain correspond to horizontal frequency components in the frequency domain.
- **Horizontal Edges:** Horizontal edges in the spatial domain correspond to vertical frequency components in the frequency domain.

9. Understanding the Discrete Fourier Transform (DFT) and its Visualisation

Frequency Spectrum of a Square

Figure 12 shows the Discrete Fourier Transform (DFT) of a square within an image. The DFT reveals the frequency components of the square:

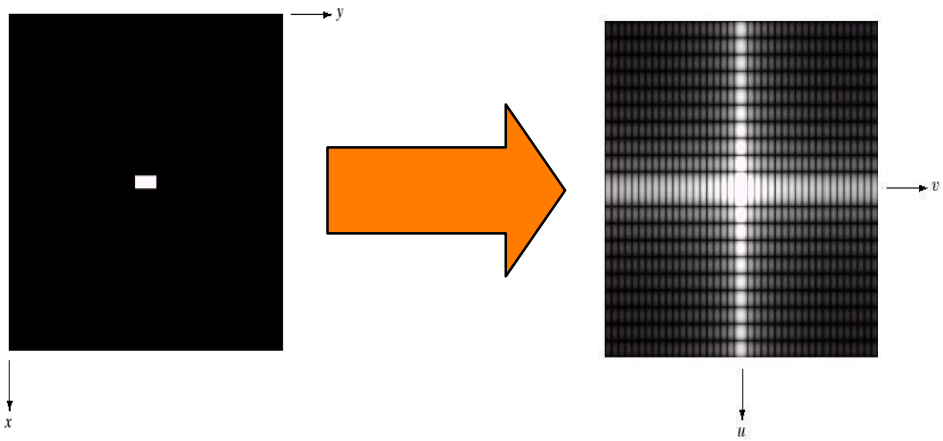


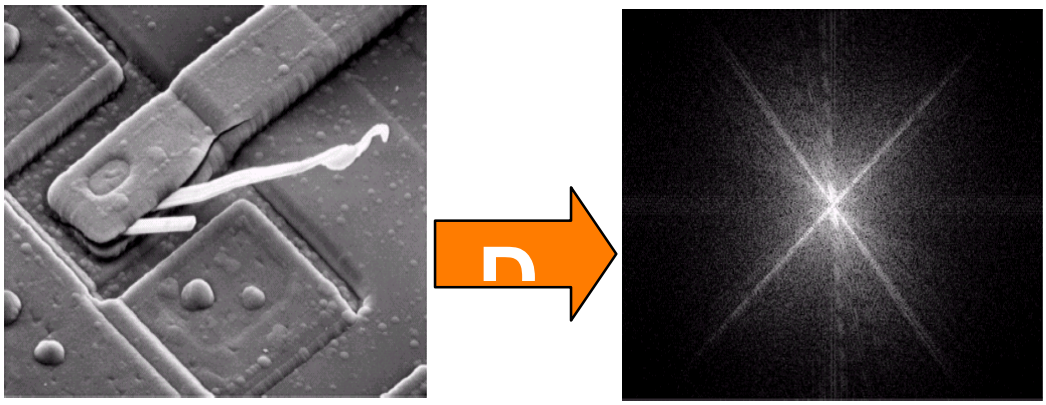
Figure 12: Discrete Fourier Transform of a Rectangular Shape

- Low Frequencies: Located in the centre of the frequency domain image.
- High Frequencies: As we move away from the centre, higher frequencies become more prominent, especially along the longer edges of the rectangle. These high frequencies correspond to the sharp transitions in the spatial domain image.

In a rectangular shape, the longer side will have more high frequencies along its length, visible as bright lines in the frequency domain. The shorter side has smoother frequencies due to fewer transitions.

Examples of DFT Outputs

Figure 13 shows two examples of DFT outputs:



Scanning electron microscope image of an integrated circuit magnified ~2500 times

Fourier spectrum of the image

Figure 13: Fourier Transform of a Complex Image with High-Frequency Edges

- **Central Part:** Contains many low frequencies representing gradual changes in image intensities.
- **Edges and Lines:** Represented by high frequencies, indicating sharp transitions or edges in the spatial domain.

High frequencies are associated with edges, while low frequencies correspond to smooth regions with gradual intensity changes.

Fourier Transform and Inverse Fourier Transform Equations

The Fourier Transform and its inverse allow conversion between the spatial and frequency domains:

Continuous Fourier Transform:

$$F(k) = \int_{\infty}^{-\infty} f(x)e^{-j2\pi kx}\, dx$$

$$f(x) = \int_{\infty}^{-\infty} F(k)e^{j2\pi kx}\, \mathrm{dk}$$

Discrete Fourier Transform:

$$F(k) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)e^{-j2\pi kx/N}$$

$$f(x) = \frac{1}{N^2}\sum_{k=0}^{N-1} F(k)e^{j2\pi kx/N}$$

These equations enable the transformation from the spatial domain $f(x)$ to the frequency domain $F(k)$ and vice versa. The discrete form is used for digital images.

Key Concepts

- **Low Frequencies:** Represent smooth regions with gradual intensity changes.
- **High Frequencies:** Correspond to edges or sharp transitions in the image.
- **Transformation:** The Fourier Transform decomposes an image into its frequency components, and the inverse Fourier Transform reconstructs the image from these components.

Understanding these principles is essential for various image processing tasks, including filtering and edge detection. By analysing the frequency domain, we can gain insights into the structural composition of images, enabling more effective image enhancement and analysis.

10. Understanding Convolution

Derived from the convolution theorem which states that convolution in the spatial domain is equivalent to multiplication in the frequency domain, this concept is widely utilized in signal processing, image processing and techniques such as Convolutional Neural Networks (CNNs). The process involves combining two functions to produce a third function that represents how the shape of one is modified by the other.

Definition:

The mathematical definition of convolution for two functions f and g is given by:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(y)g(x-y)\,dy$$

In this equation:

- $f(y)$ is the filter function.
- $g(x-y)$ is the input signal (or image).
- The result of the convolution is the filtered signal, $(f*g)(x)$.

Application in Image Processing

In image processing, convolution is used to apply filters to images. Here is how it works:

- Filter (Kernel): A small matrix used to modify the image. The filter values (kernel) slide over the entire image.
- Image: The function being processed by the filter.

For example, consider a median filter applied to an image:

- The kernel (filter) is a small, predefined matrix, such as a 3x3 matrix with values.
- This kernel moves across the image, element by element, multiplying its values with the corresponding pixel values of the image.
- The sum of these multiplications gives the new value for the central pixel of the image under the kernel.

Characteristics of Convolution

- Linear Function: Convolution is a linear operation, meaning it modifies the pixel values but does not change their relative positions. This is why convolution is also known as a linear shift-invariant filter.
- Shift Invariance: The filter's effect is the same, regardless of where it is applied in the image.

Example on Convolution:

If the filter kernel is a 3x3 matrix with values like [1, 2, 0.5, 0.3, etc.], the convolution process would:

- Multiply the kernel values with the corresponding image pixel values.
- Sum these products to get a new pixel value in the output image.

Key Points

- Filter: $f(y)$ is the filter applied to the image.
- Image: $g(x-y)$ is the image being processed.
- Output: The result of the convolution, $(f*g)(x)$, is the filtered image.

Practical Use

Convolution is essential in image processing tasks such as edge detection, blurring, sharpening, and more. Understanding convolution helps in designing and applying filters effectively to enhance image features or extract useful information.

Convolution's role in CNNs illustrates its importance in deep learning, where multiple layers of convolutions help in automatically learning hierarchical features from raw images, making it a powerful tool in computer vision applications.

11. Convolution in Image Processing

Convolution can be understood through its operation in both the spatial and frequency domains. The mathematical representation shows that the Fourier transform of the convolution of two functions g and h is the product of their Fourier transforms:

$$\mathcal{F}\{g * h\} = \mathcal{F}\{g\}\mathcal{F}\{h\}$$

Similarly, the inverse Fourier transform of the product of two functions g and h in the frequency domain is equivalent to their convolution in the spatial domain:

$$\mathcal{F}^{-1}\{gh\} = \mathcal{F}^{-1}\{g\} * \mathcal{F}^{-1}\{h\}$$

This property allows us to switch between spatial and frequency domains for efficient computation of convolutions.

Practical Example of Convolution with Image Filtering

This practical example as depicted in Figure 14 demonstrates convolution in image processing using a specific kernel (filter) to detect vertical edges:

- **Intensity Image:** The original grayscale image to be processed.
- **Kernel (Filter):** A 3x3 matrix, typically used for edge detection. In this example, the kernel emphasises vertical edges:

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

- **Filtered Image:** The output image after applying the kernel. This process enhances vertical edges in the original image, as the kernel moves across the image and computes the weighted sum of the neighbouring pixels.

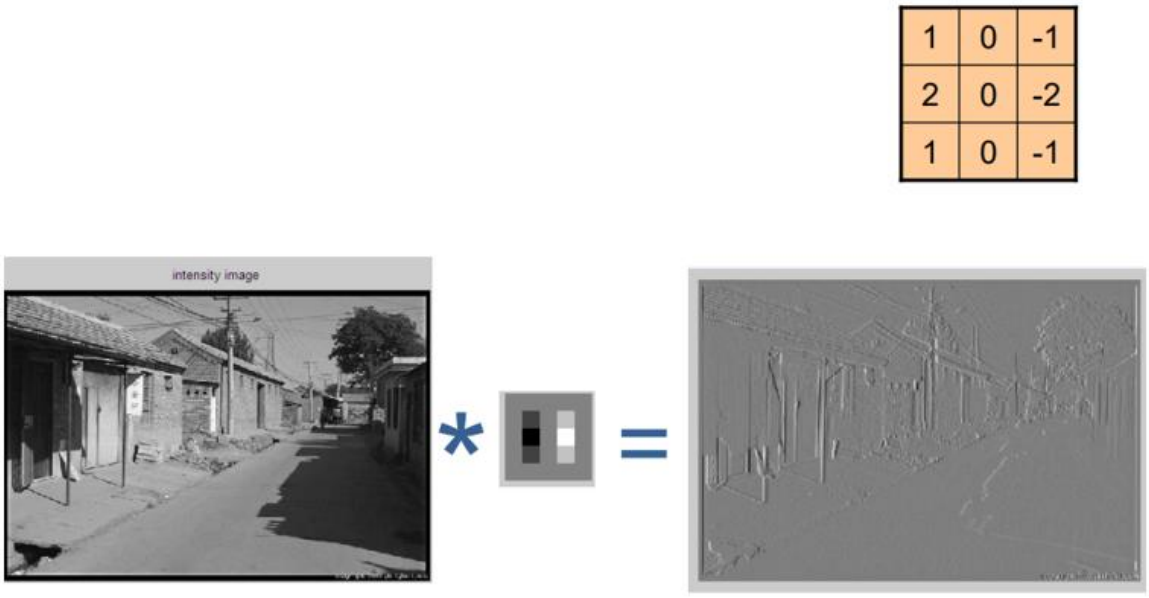


Figure 14: Vertical Edge Detection Using Convolution

Convolution Process

- Kernel (Filter): The kernel is a small matrix that slides over the image, applying its values to the corresponding pixels.
- Multiplication and Summation: For each position of the kernel, its values are multiplied by the pixel values of the image. The results are summed to produce the new pixel value in the output image.
- Edge Detection: The example kernel detects vertical edges by emphasising differences in pixel values in the vertical direction while ignoring horizontal changes.

12. Examples of Fourier Transform in Images

Analysis of Simple Shapes

Let us examine how the Fourier transform appears for different simple shape images as depicted in Figure 15.

1. Tilted White Bar

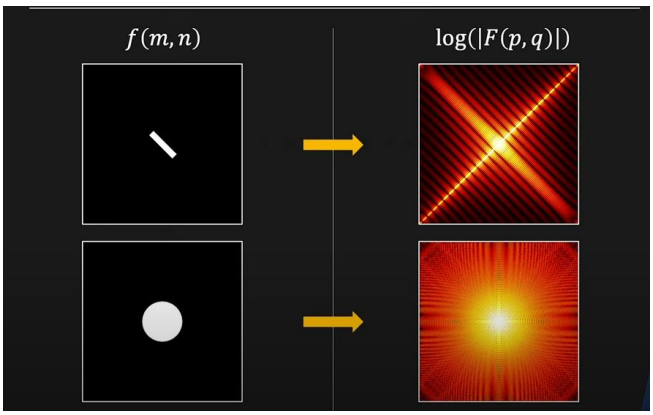


Figure 15: Fourier Transform Visualisation of Simple Geometric Shapes

- Spatial Domain: The image on the left shows a tilted white bar.
- Frequency Domain: The Fourier transform of this image shows strong lines perpendicular to the direction of the bar. The central part, which contains low frequencies, is relatively stable. However, as we move away from the centre, high frequencies become more prominent, reflecting edge information and noise. The orientation of the image is mirrored in the frequency domain.

2. Circular Shape

- **Spatial Domain**: The lower image shows a white circle.
- **Frequency Domain:** The Fourier transform of the circle is round, highlighting that the transform aligns with the shape's orientation. This demonstrates that an image's orientation affects its Fourier transform.

Analysis of Complex Images

Now let us look at more complex images as illustrated in Figure 16.

3. Rubik's Cube

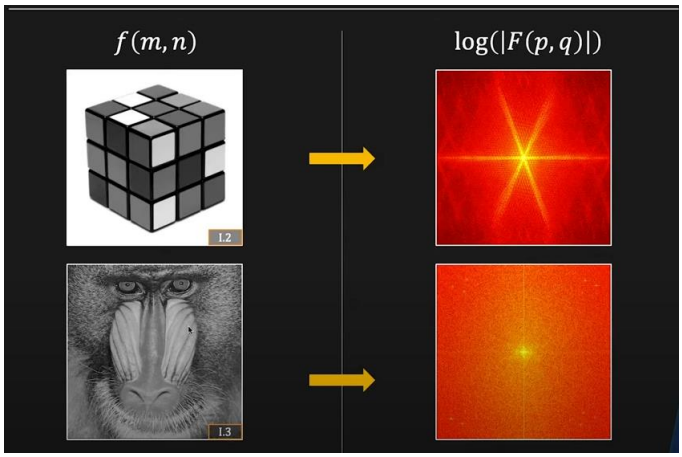


Figure 16: Fourier Transform Visualisation of Complex Images

- **Spatial Domain:** The image of a Rubik's Cube shows clear horizontal and vertical lines.
- **Frequency Domain:** The Fourier transform reveals these lines as prominent frequencies. For visualisation purposes, a logarithmic function is applied to compress the range of frequency amplitudes. This helps in better visualising the frequency components.

4. Monkey Image

- **Spatial Domain:** The image of a monkey is complex with many edges and features.
- **Frequency Domain:** The Fourier transform of this image is more intricate. It contains numerous high-frequency components, indicating the presence of many edges and fine details. As the image complexity increases, interpreting the Fourier transform becomes challenging.

Key Observations

- **Orientation and Shape**

  The Fourier transform reflects the orientation and shape of the objects in the image. For example, a tilted line results in a transform with perpendicular lines.
- **Complexity**

  Simple shapes produce more straightforward Fourier transforms, while complex images result in intricate transforms with many high-frequency components.

- **Noise Susceptibility**

  High-frequency components are more prone to noise. As seen in the complex images, the presence of high frequencies often indicates susceptibility to noise.

13. Filtering in the Frequency Domain

Image processing operations, such as denoising, can be performed in either the spatial or frequency domains. The example as depicted in Figure 17 illustrates the following:

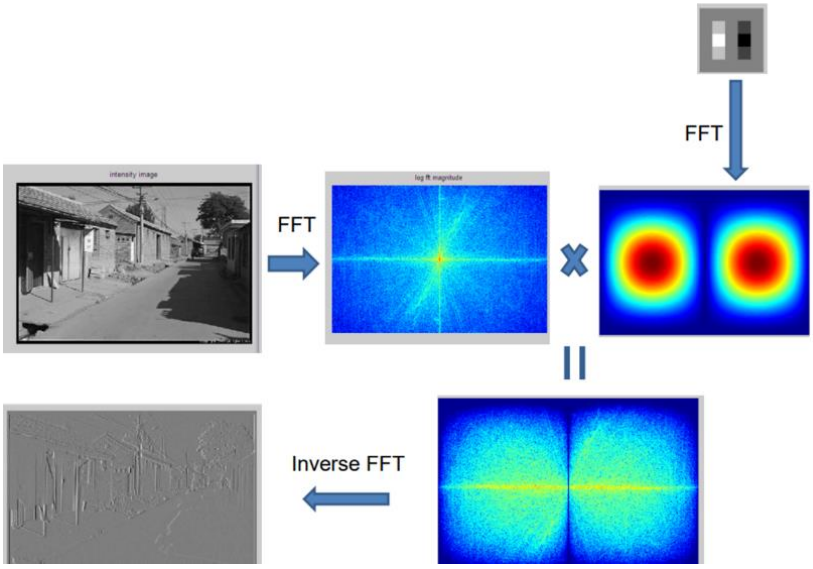


Figure 17: Image Filtering Process Using Fast Fourier Transform (FFT)

- **Original Image:** A grayscale image.
- **FFT and Filtering:** Applying Fast Fourier Transform (FFT) converts the image into the frequency domain. Multiplying this with a filter allows for specific frequency manipulation.
- **Inverse FFT:** Applying inverse FFT converts the filtered image back to the spatial domain, showing the effects of the frequency domain filtering.

This process demonstrates that operations feasible in the spatial domain can also be performed in the frequency domain, often with different visualisations for better understanding.

13.1 Low Pass Filtering

Low pass filtering allows only low-frequency components to pass through while eliminating high frequencies. Figure 18 depicts the effect of Low Pass Filtering on the Frequency Spectrum:

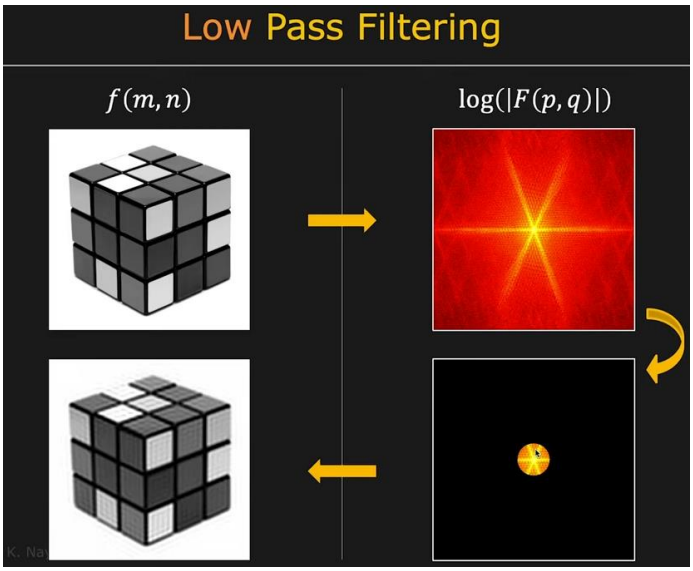


Figure 18: Effect of Low Pass Filtering on Image Frequency Spectrum

- Fourier Transform: The Fourier transform shows a detailed frequency spectrum.
- When a low pass filter is applied as a result of which high frequencies are removed, resulting in a slightly blurred image upon inverse transform, indicating the loss of sharp edges.

Let us consider an extended example as depicted in Figure 19:

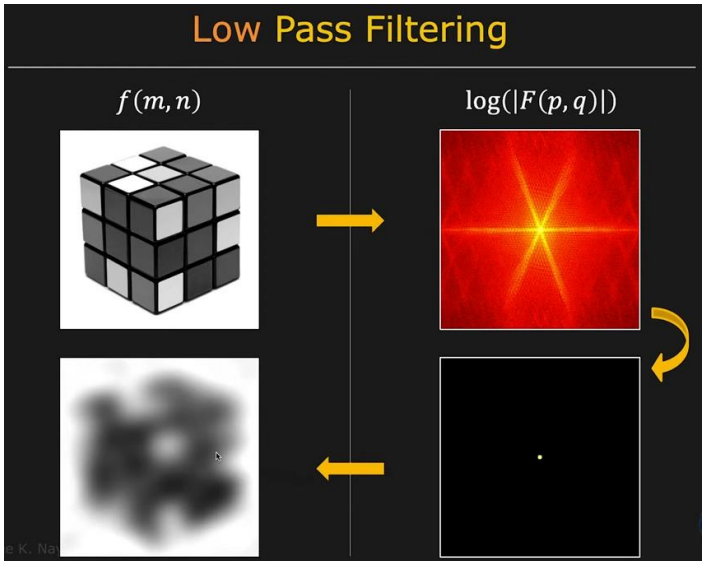


Figure 19: Impact of Low Pass Filtering on Image Clarity and Frequency Spectrum

Further reducing the frequency components to a smaller central area results in a more blurred image. This process confirms that removing high frequencies eliminates edge details, leading to a smooth, blurred image.

Low pass filters are effective for blurring as they allow low frequencies and eliminate high frequencies.

13.2 High Pass Filtering

High pass filtering allows high-frequency components to pass through while eliminating low frequencies, Figure 20 depicts the simple as well as extended aspects of high pass filtering on the image frequency spectrum. When a high pass filter is applied removing the central low-frequency components results in an image with enhanced edges and reduced brightness upon inverse transform.

Considering the extended version of the example as shown on the right side of Figure 20, the following points need to be kept in mind:

- Increased Radius: Expanding the radius of the high pass filter eliminates more low frequencies, further emphasising edges and diminishing smooth areas.

High-pass filters are effective for edge detection as they highlight high-frequency components and remove low-frequency components.

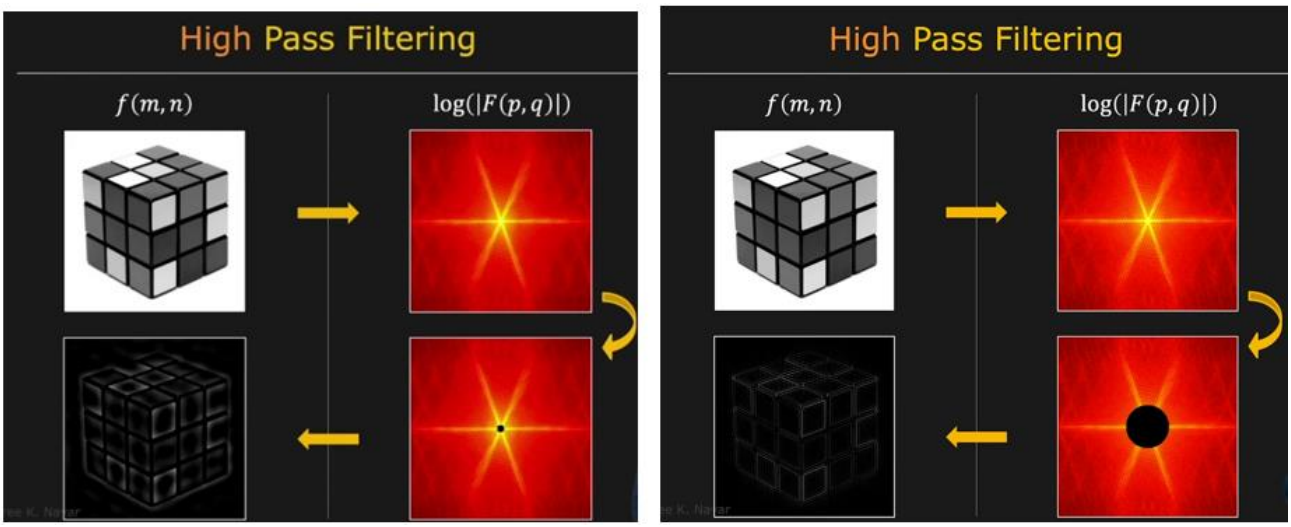


Figure 20: Effect of High Pass Filtering on Image Frequency Spectrum

13.3 Gaussian Smoothing

Gaussian Smoothing Example

The example depicted in Figure 21 demonstrates Gaussian smoothing, a common technique used for blurring images. The process is visualised both in the spatial and frequency domains and the Figure illustrations are as follows:

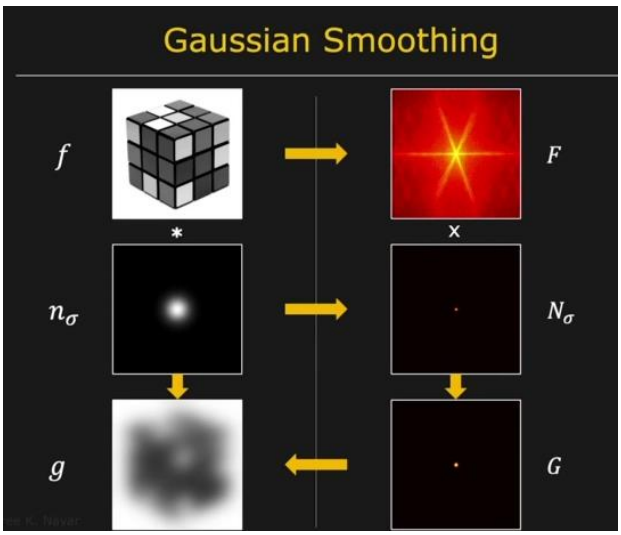


Figure 21: Illustration of Gaussian Smoothing in Spatial and Frequency Domains

- **Original Image (f):** The image of a Rubik's Cube.
- **Fourier Transform (F):** The Fourier transform of the original image.

- **Gaussian Filter (nσ):** A Gaussian smoothing filter is applied, which is essentially a low-pass filter.
- **Filtered Image (g):** The resulting blurred image after applying the Gaussian filter in the frequency domain.

This illustrates how Gaussian smoothing works as a low-pass filter, removing high-frequency components and resulting in a blurred image.

Frequency Domain Filtering Pipeline

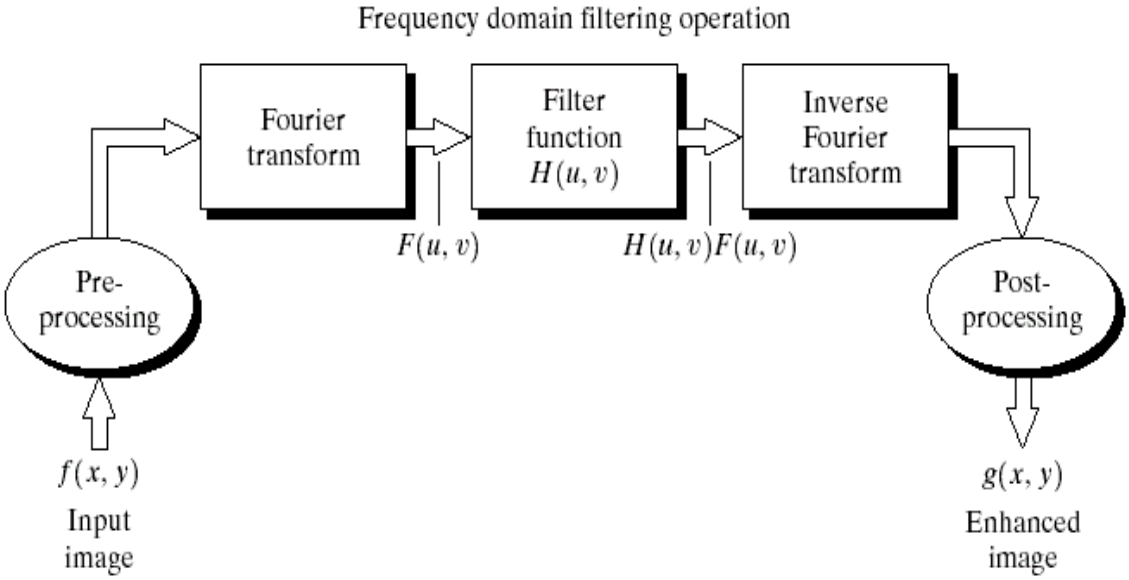


Figure 22: Frequency Domain Pipeline

The pipeline for frequency domain filtering as depicted in Figure 22 consists of the following steps:

- **Pre-processing:** Initial processing of the input image $f(x,y)$.
- **Fourier Transform:** Converting the image to the frequency domain $F(u,v)$.
- **Filtering:** Applying a filter function $H(u,v)$ to manipulate the frequency components.
- **Inverse Fourier Transform:** Converting the filtered image back to the spatial domain.
- **Post-processing:** Additional morphological operations, if necessary, to enhance the final output image $g(x,y)$.

Notch Filter Example

A notch filter specifically targets and removes certain frequencies, typically those at the centre of the frequency spectrum, which correspond to low-frequency components. The result is an image with enhanced high-frequency details and sharper edges as depicted in Figure 23.

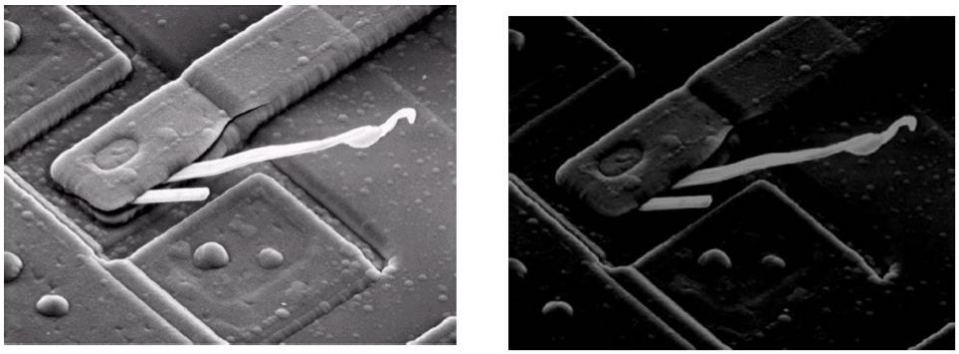


Figure 23: Edge Detection Result: Input vs. Output Image

- **Notch Filter Operation:** The central region of the frequency domain is set to zero, eliminating low-frequency components while preserving high frequencies.
- **Output:** The resulting image is sharper, with low-frequency components removed, emphasising high-frequency details and edges.

13.4 Mathematical Aspects of Frequency Domain Filters

13.4.1 Smoothing Frequency Domain Filters

Smoothing in the frequency domain is achieved by eliminating the high frequencies, which are typically associated with the edges in an image. The basic model of this operation is represented by the function $H(u,v)\cdot F(u,v)$, where F is the Fourier transform of the image being filtered, and H is the filter applied. Here, u and v correspond to the frequency domain coordinates.

A low pass filter allows low frequencies to pass while attenuating the high frequencies, resulting in a blurred image. Conversely, a high pass filter allows high frequencies to pass and reduces the low frequencies, enhancing the edges and details in the image.

13.4.2 Ideal Low Pass Filter

The ideal low pass filter can be visualised as shown in the Figure 24 below:

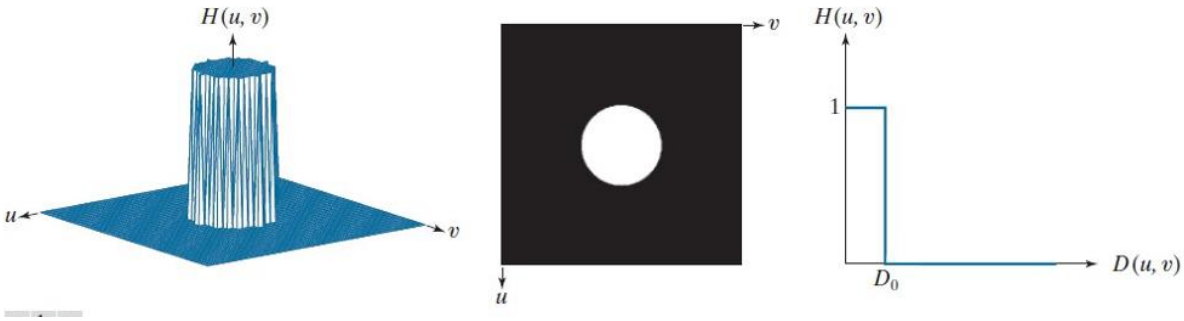


Figure 24: (a) Perspective plot of an ideal low pass-filter transfer function (b) Function displayed as an image (c) Radial cross-section

In this domain, the filter is represented by a mask where the central low-frequency components are retained (white region, multiplied by 1) and the high-frequency components are discarded (black region, multiplied by 0).

Effect of Ideal Low Pass Filter on Images

The impact of applying an ideal low-pass filter with varying radii can be seen in the following image:

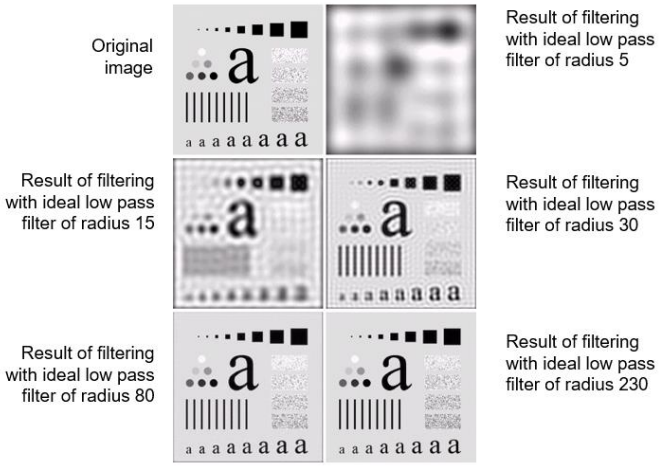


Figure 24: Effect of Varying Radii in Ideal Low Pass Filtering on Image Quality

(Source: Gonzalez & Woods, Digital Image Processing (2002))

As the radius of the low pass filter increases (5, 15, 30, 80, 230), the image transitions from being heavily blurred (low radius) to sharper (high radius). This occurs because a larger radius includes more high-frequency components, reducing the overall blurring effect.

While an ideal low pass filter is theoretical and not feasible for practical applications due to its sharp transition, it serves as a foundation for understanding how frequency domain filtering operates. In practice, more feasible approximations like the Butterworth and Gaussian low pass filters are used.

13.4.3 Butterworth Lowpass Filters

The Butterworth filter is a type of low-pass filter that aims to closely approximate an ideal filter. The Butterworth filter equation is given by:

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

In this equation:

- $D(u,v)$ represents the distance from the origin in the frequency domain.
- D_0 is the cut off frequency.
- n is the order of the filter, which controls the sharpness of the filter's cut off.

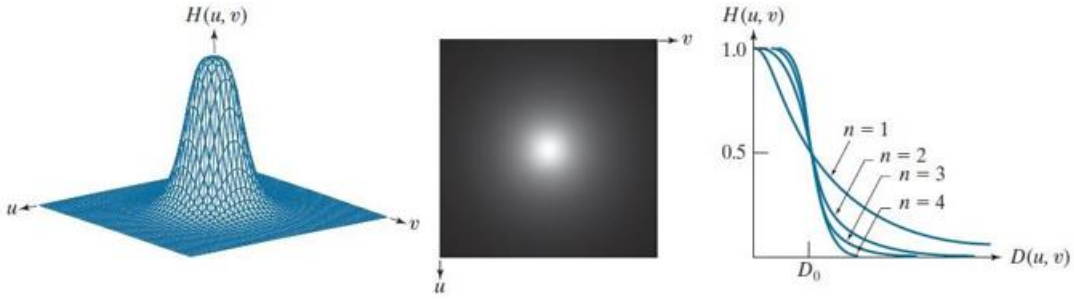


Figure 25: (a) Perspective plot of a Butterworth lowpass filter function (b) Function displayed as an image (c) Radial cross-section of BLPFs of orders 1 through 4.

As seen in Figure 25, the parameter n controls the smoothness of the curve. A lower order (n=1) results in a smooth transition, while higher orders make the cut off more abrupt.

Effects of Butterworth Lowpass Filtering on an Image

The results of filtering with Butterworth filters of different cut off radii and order 2 are as below.

From the Figure 26, it is clear that:

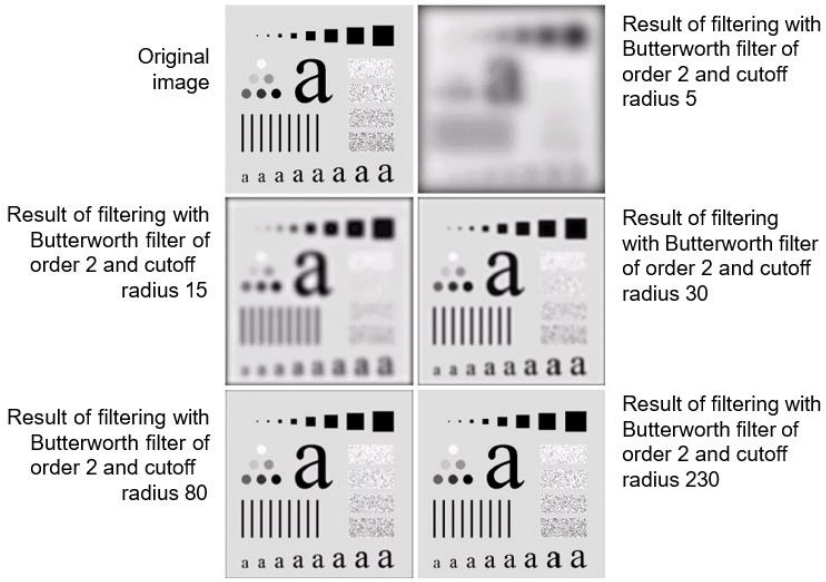


Figure 26: Effect of Butterworth Lowpass Filter with Varying Cutoff Radii on Image Blurring

- A smaller cut off radius (e.g., radius 5) results in a highly blurred image.
- As the cut off radius increases (e.g., radius 15, 30, 80, 230), the image retains more details because higher frequency components are included, leading to sharper images.

The Butterworth filter provides a more gradual transition compared to an ideal filter, which results in fewer artefacts in the processed images. As the cutoff radius and order increase, the filter approaches the characteristics of an ideal low-pass filter, allowing more high-frequency components to pass through and resulting in clearer images.

13.4.4 Gaussian Lowpass Filters

Building a filter using a Gaussian low pass filter, the equation looks like:

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

This equation, $H(u,v) = e^{-D^2(u,v)/2D_0^2}$, is very similar to the Gaussian filtering function. Here, D represents the distance between the centre point. Increasing the radius decreases the smoothness but focusing on the centre part allows only the low-frequency components to pass through, leading to a significant amount of blurring.

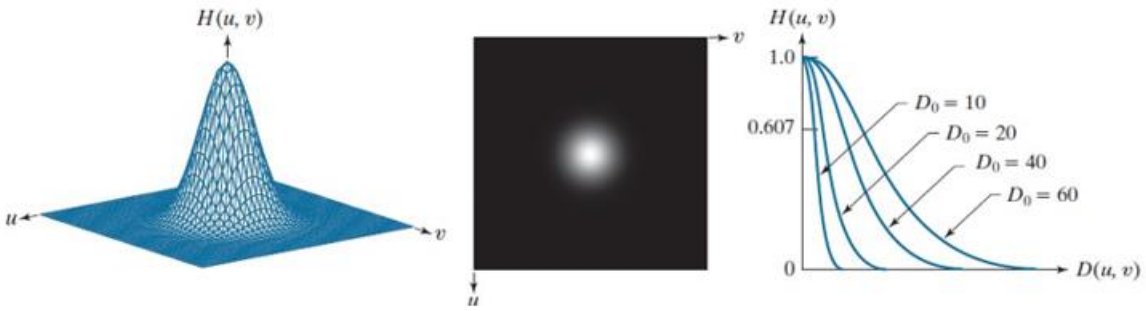


Figure 27: (a) Perspective plot of a GLPF transfer function. (b) Function displayed as an image. (c) Radial cross sections for various values of D_0

Figure 27 illustrates a Gaussian Lowpass Filter in the frequency domain. It includes a 3D plot showing low frequencies passing while high frequencies are reduced, a 2D image visualizing this effect, and cross-sectional plots for different cut off frequencies (D_0), showing broader filters allow more high frequencies through.

The resulting images depicted in Figure 28 show the results after applying Gaussian filters with different cut off radii. As the cut off radius increases, the blurriness decreases, but sharpness in certain regions remains.

As you increase the cut off value, you get more blurring, but at a smaller radius, you achieve higher image sharpness due to the retention of low frequencies while eliminating high frequencies.

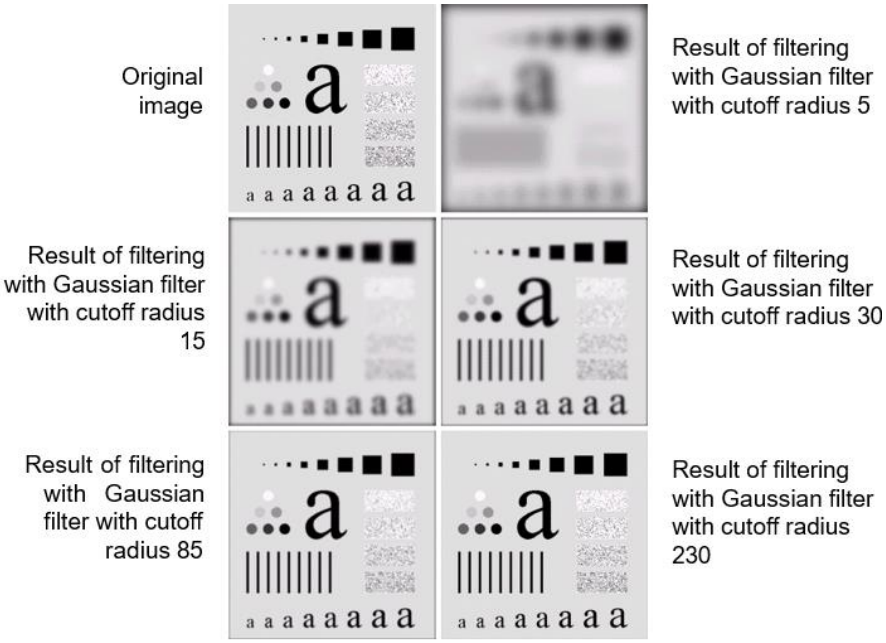


Figure 28: Effect of Gaussian Lowpass Filtering with Varying Cut off Radii

Figure 29 illustrates the comparison of the results of filtering with an ideal low pass filter, Butterworth filter, and Gaussian filter, each with a cut off radius of 15. The Gaussian filter provides a smoother transition compared to the other filters.

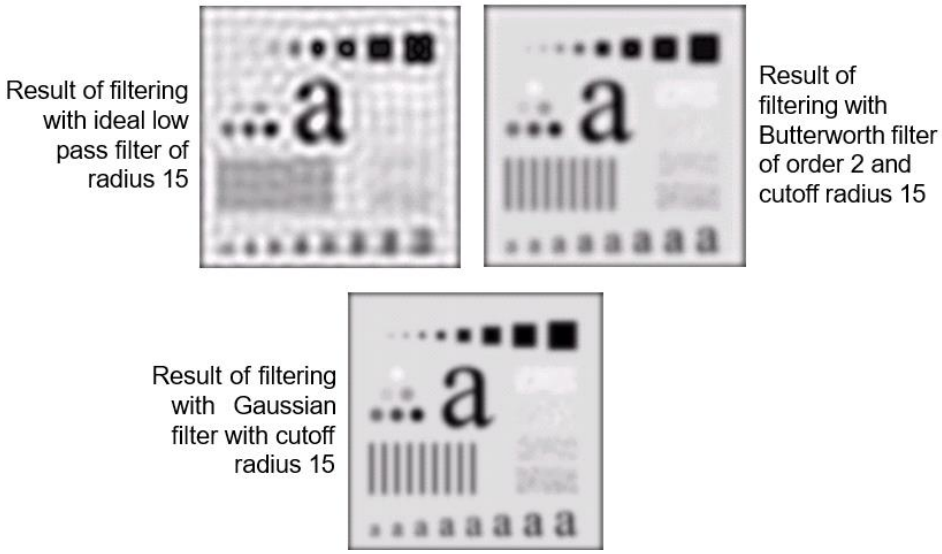


Figure 29: Comparison of Low Pass Filters with Cutoff Radius 15

Applications of Low Pass Gaussian Filter

Gaussian filters have various practical applications, particularly in text processing.

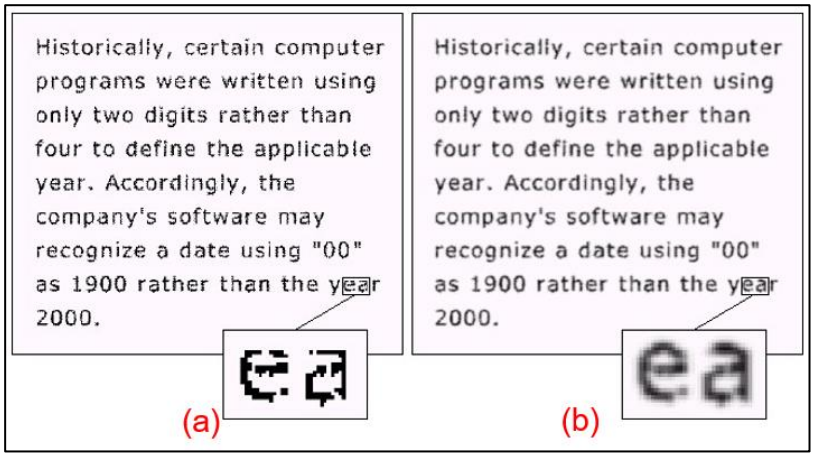


Figure 30: Text Smoothing Comparison: Unfiltered vs. Gaussian Filtered

In this Figure 30, the text is broken in the original version. Applying Gaussian smoothing blends, the pixel values, making the text more readable. Although it might not be perfect, it significantly improves clarity.

13.4.5 Ideal High Pass Filters

This method allows only the high-frequency components to pass through, effectively enhancing the edges and details of the image, contrary to low-pass filtering. The ideal high pass filter is given as:

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

In Figure 31, the mask retains the high-frequency components (white regions) while the centre black region is multiplied by zero, preventing any low-frequency components from passing through. As a result, the high frequencies in the image are preserved, leading to the following effects:

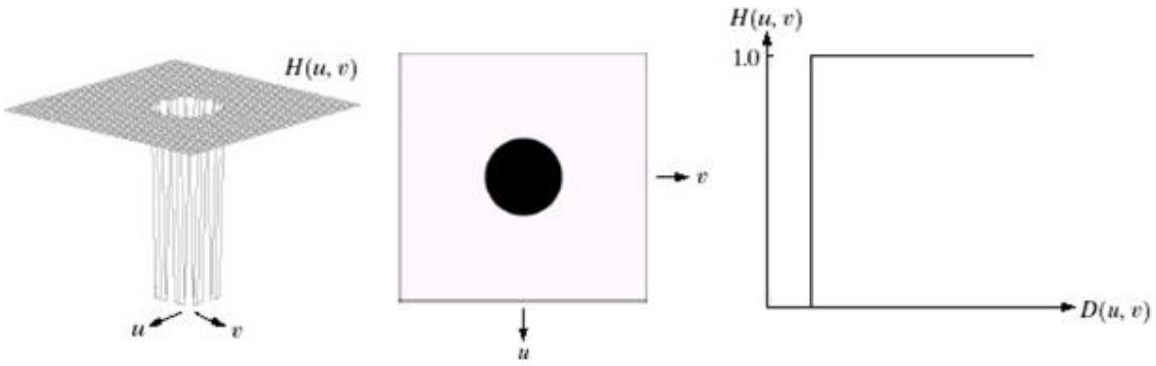


Figure 31: Ideal High Pass Filter

When the cut off distance D_0 is set to 15, fewer high frequencies are allowed, resulting in a somewhat blurry image. As the distance increases to 80, more high-frequency components are included, significantly sharpening the image as depicted in Figure 32. This process removes the low-frequency regions, which correspond to brightness, and retains only the sharp edges, enhancing the overall sharpness and detail of the image.

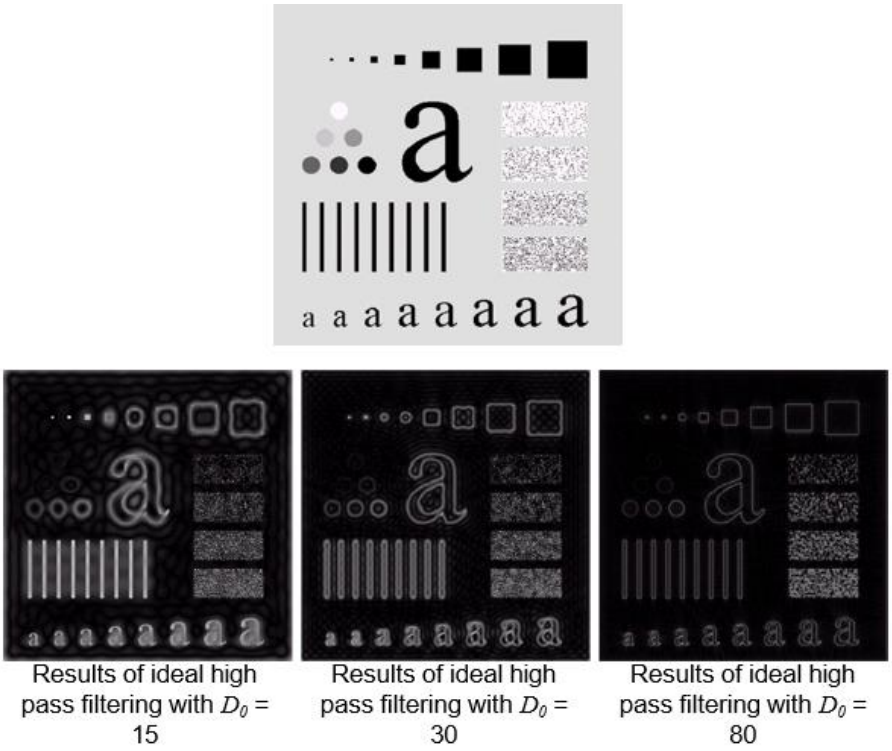


Figure 32: Results of Ideal High Pass Filtering with Varying D_0 Values

13.4.6 Butterworth High Pass Filter

To approximate an ideal high-pass filter, we use a Butterworth filter, which is defined by the following function:

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

Here, n is the order of the filter, and D_0 is the cut off distance, similar to the low pass Butterworth filter but inverted.

Figure 32 of the Butterworth high pass filter is depicted below:

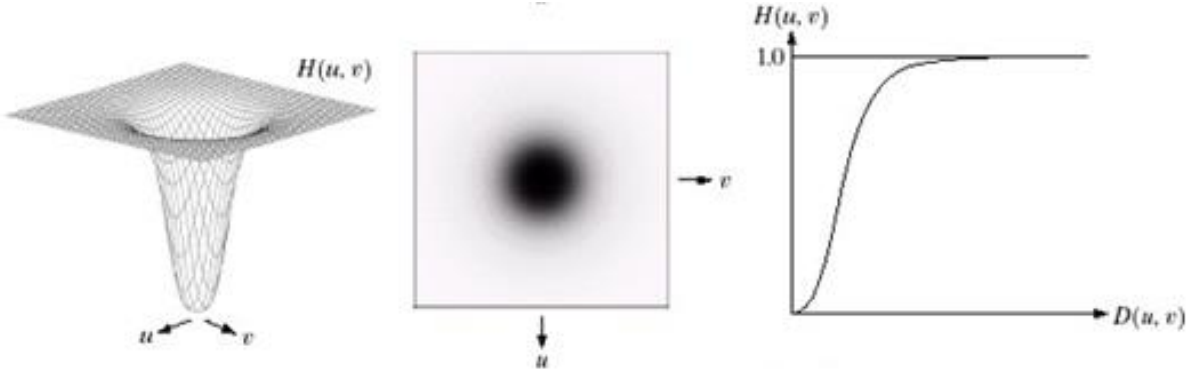


Figure 32: Visualisation of the Butterworth High Pass Filter Function

- The leftmost image shows a 3D perspective plot of the Butterworth high pass filter function.
- The middle image displays the filter function as an image, where the black region represents zero values (low frequencies are attenuated), and the white region represents the high frequencies that are allowed to pass.
- The rightmost image shows the radial cross-section, where the curve becomes steeper as the order n increases.

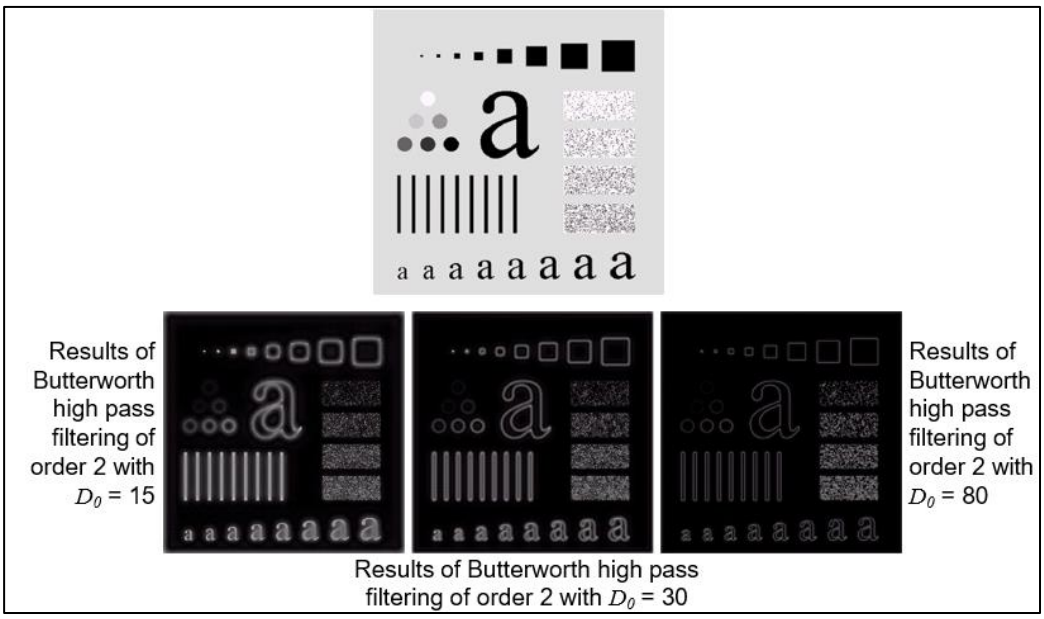


Figure 33: Butterworth High Pass Filtering with Varying Cutoff Distances

Figure 33 illustrates the results of Butterworth high pass filtering with varying cut off distances and can be interpreted as below:

- The top image is the original image for reference.

- The three images below show the results of applying Butterworth high pass filtering of order 2 with D_0 values of 15, 30, and 80 respectively.
- With a lower D_0 value, fewer high-frequency components pass through, making the image less sharp.
- As D_0 increases, more high-frequency components are allowed, enhancing the edges and making the image appear sharper, emphasising the high-frequency content.

13.4.7 Gaussian High Pass Filter

The Gaussian high pass filter is given as:

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

The term $e^{-D^2(u,v)/2D_0^2}$ the above equation is a representation of the Gaussian function, where D is the distance from the origin in the frequency domain. The Gaussian high pass filter allows high-frequency components to pass through while reducing low-frequency components. This results in a sharper image by emphasising edges and details.

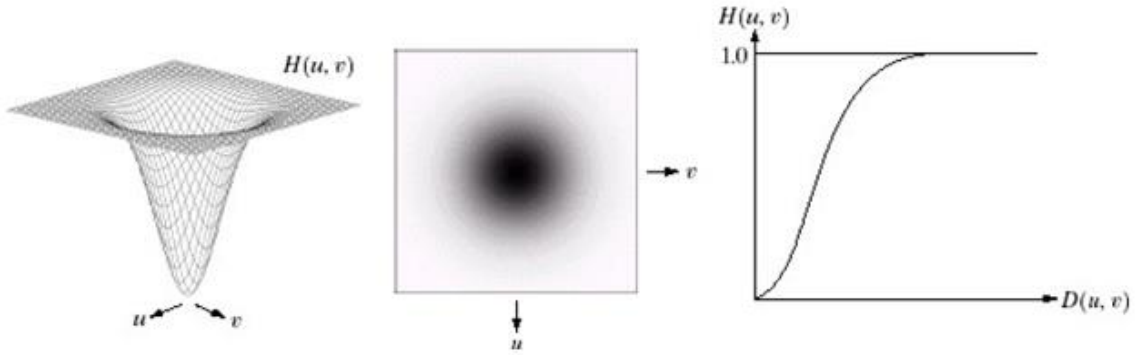


Figure 34: Gaussian High Pass Filter: Visual Representation and Characteristics

Figure 34 can be interpreted as below:

- The original image undergoes filtering with different cutoff distances (D_0).
- The results of applying the Gaussian high pass filter with increasing cutoff distances (D_0) demonstrate that images become darker and sharper.

13.4.8 Comparison of High Pass Filters

To understand the differences between various high pass filters, let us compare the results of Ideal, Butterworth, and Gaussian high pass filters as depicted in Figure 35.

Ideal High Pass Filter

- Provides the sharpest edges but may introduce artefacts.
- Results in a very stark contrast between the filtered and unfiltered regions.

Butterworth High Pass Filter

- Offers a smoother transition compared to the Ideal filter.
- The image appears somewhat blurred but retains significant high-frequency details.

Gaussian High Pass Filter

- Balances between the sharpness of the Ideal filter and the smoothness of the Butterworth filter.
- The image remains sharp while minimising artefacts.

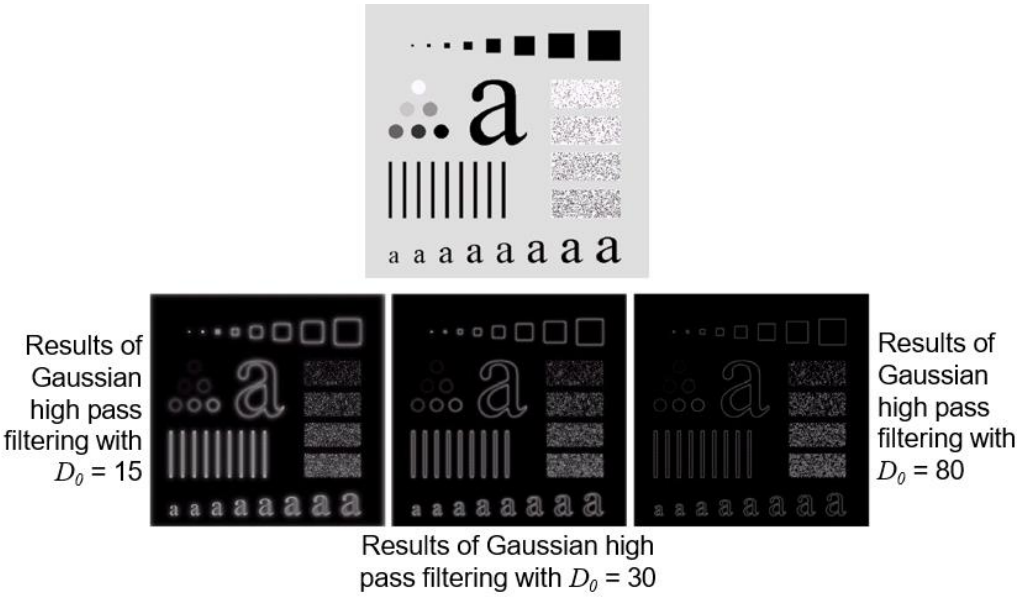


Figure 35: Gaussian High Pass Filtering with Various Cutoff Distances

Analysis

The results indicate that while the Ideal high pass filter provides the highest sharpness, it may also cause unwanted artefacts. The Butterworth filter offers a compromise with less sharpness but smoother transitions. The Gaussian high pass filter effectively combines the benefits of both, providing sharpness without significant artefacts.

13.4.9 Real-Life Application of High-Pass Filter

In this real-life application, the original chest X-ray image as depicted in Figure 36 undergoes various high-pass filtering techniques and histogram equalisation to enhance and distribute pixel values more evenly. Initially, the original image provides limited information, but the final processed image reveals detailed edge information, organ structures, and bone details. This enhanced image is significantly more useful for making inferences and can be effectively utilised in deep learning models for improved analysis.

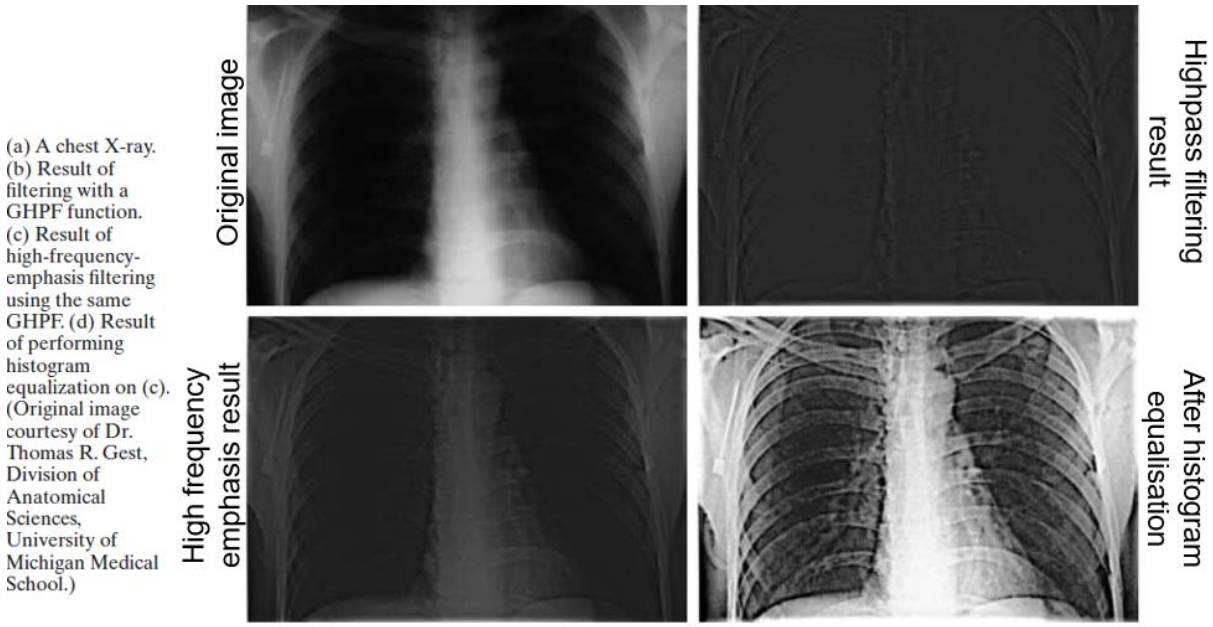


Figure 36: Enhancement of Chest X-ray Using High-Pass Filtering and Histogram Equalisation

14. Principles of Linearity

The principle aligns with the concepts of additivity and homogeneity in the context of linear functions and is crucial for understanding the nature of certain filters.

1. Additivity

A function R is additive if it satisfies the following condition:

$$R(x_1+x_2)=R(x_1)+R(x_2)$$

This means that applying the function to the sum of two inputs yields the same result as applying the function separately to each input and then summing the results. This property is essential for determining if a function is linear.

2. Homogeneity (Scaling)

A function R satisfies homogeneity if:

$$R(\alpha x_1)=\alpha R(x_1)$$

Here, applying the function to a scalar multiple of an input results in the scalar multiple of the function's output. This property, combined with additivity, characterises a linear function.

3. Combined Property

When both additivity and homogeneity are satisfied, the function is linear:

$$R(\alpha\, x_1+\beta\, x_2)=\alpha R(x_1)+\beta R(x_2)$$

This combined property ensures that the function consistently adheres to the principles of linearity for any inputs x_1 and x_2, and scalars α and β.

15. Linear and Non-Linear Filters

Linear Filters

Linear filters, such as low-pass and high-pass filters, are essential tools in image processing. Here is a brief overview of their functionalities and properties:

- **Low Pass Filters:** These filters allow low-frequency components to pass through while blocking high-frequency components. They smoothen or blur an image by reducing sharp transitions and details.
- **High Pass Filters:** These filters allow high-frequency components to pass through while blocking low-frequency components. They enhance edges and fine details in an image by emphasising high-frequency components.

Both low-pass and high-pass filters are considered linear because their output is a linear combination of the input. They do not alter the fundamental structure of the waveform but combine existing waveforms to produce the output.

Non-Linear Filters

Non-linear filters, such as median filters and bilateral filters, work differently:

- **Median Filters:** These filters replace each pixel's value with the median value of the intensities in the neighbourhood of that pixel. This effectively removes noise while preserving edges.

- **Bilateral Filters:** These filters smooth images while preserving edges by averaging pixel values with weights assigned based on spatial and intensity differences.

Non-linear filters change the composition or distribution of an image, which means they do not satisfy the principle of linearity. They do not follow the additive and homogeneity principles that define linear functions.

16. Summary

In this topic, we discussed:

- Image Enhancement Basics: Understanding the necessity of frequency domain processing for efficient filtering and improved image compression.
- Spatial Domain Limitations: Recognising the computational challenges and edge artefacts in the spatial domain.
- Fourier Series Concepts: Learning how Fourier series decompose complex waveforms into simpler sine and cosine components for analysis.
- Low-Pass and High-Pass Filters: Exploring various types of filters and their effects on image blurring and sharpening.
- Real-Time Applications: Discuss the practical uses of filters in tasks like text smoothing and edge detection.
- Principles of Linearity: Distinguishing between linear (low pass and high pass) and non-linear (median and bilateral) filters based on additivity and homogeneity.
- Fourier Transform Advantages: Understanding the efficiency and importance of the Fourier Transform and Fast Fourier Transform (FFT) in image processing.
- Frequency Domain Operations: Analysing images in the frequency domain for tasks like filtering, noise reduction, and pattern recognition.
- Convolution Understanding: Applying convolution in both spatial and frequency domains for effective image processing tasks.
- Frequency domain processing: Allows for efficient filtering, effective image compression, and enhanced pattern recognition, with operations like sharpening and blurring being more effective.

- The Fourier Transform: Facilitates image transformation from spatial to frequency domain, aiding various processing tasks.
- Linearity principles: Include additivity and homogeneity; both must be satisfied for a function to be considered linear.
- Linear filters (low pass and high pass): Maintain waveform structure, allowing specific frequency components to pass and altering the image accordingly.
- Non-linear filters (median and bilateral): Change image composition, effectively removing noise and preserving edges without satisfying linearity principles.