1 Introduction

This set of lectures introduces the use of the Fourier transform in image processing. The main aim is to show how frequency representation helps in analyzing image structure and how it supports several key processing tasks. The lectures explain the principles of the *Discrete Fourier Transform (DFT)* and the *Fast Fourier Transform (FFT)*, and they demonstrate how these tools are applied in filtering, enhancement, and restoration of images.

The lecture series is organized into three parts. The first lecture presents the DFT and discusses sampling, aliasing, and windowing, which link continuous signals to their discrete representations. The second lecture focuses on the FFT algorithm and its efficiency. The third lecture is devoted to applications of the FFT in image processing.

2 Motivation: Why the Fourier Transform?

The Fourier transform is based on the idea that any sufficiently regular signal or image can be expressed as a superposition of sinusoidal components of different frequencies, amplitudes, and phases. Representing data in terms of these components reveals periodicity, orientation, and scale, which are often difficult to observe directly in the spatial or temporal domain.

In the continuous setting, the Fourier transform of a function $f(t)$ is defined as

$$F(u) = \int_{-\infty}^{\infty} f(t)\, e^{-j2\pi ut}\, dt,$$

and the inverse transform reconstructs f from its frequency representation,

$$f(t) = \int_{-\infty}^{\infty} F(u)\, e^{j2\pi ut}\, du.$$

Using Euler's formula,

$$e^{-j2\pi ut} = \cos(2\pi ut) - j\sin(2\pi ut),$$

we see explicitly that the transform decomposes a signal into cosine and sine waves. The function $F(u)$ is defined on the frequency domain, where frequency is measured in cycles per unit of the independent variable of the original signal.

Several fundamental properties make the Fourier transform indispensable for image processing. The transform is linear, so sums of signals correspond to sums of their transforms. Through the convolution theorem, convolution in space or time becomes simple multiplication in the frequency domain:

$$f * g \xrightarrow{\mathcal{F}} F(\omega)G(\omega),$$

which forms the basis of fast filtering via the FFT. Parseval's identity shows that total energy is preserved between the spatial and frequency domains.

Working in the frequency domain is valuable for many imaging tasks. Low-pass filtering can suppress high-frequency noise while retaining coarse structures, whereas high-pass filtering emphasizes edges. Periodic disturbances such as grid patterns or ripple noise can be removed using narrowband filters. Texture and repeated patterns manifest as localized peaks in the spectrum, with their positions encoding both spatial frequency and orientation. Large-kernel filtering or correlation becomes computationally feasible because convolution can be performed efficiently using the FFT.

In two-dimensional images, low frequencies near the origin of the spectrum correspond to smooth, slowly varying structures such as illumination and overall shapes. High frequencies represent fine details such as edges and noise. Directional or oriented patterns generate anisotropic structures in the spectrum, often visible as lines or peaks aligned perpendicularly to the direction of the image pattern.

A simple illustration is a horizontal sinusoidal stripe pattern. Its 2D Fourier spectrum contains two symmetric peaks positioned on the vertical axis. The distance of the peaks from the center encodes the spatial frequency of the stripes, and their placement along the axis reveals the orientation of the underlying pattern.

3 From the Continuous Fourier Transform to the Discrete Case

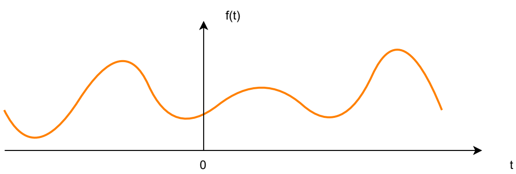
In image processing we always work with discrete digital data, so the continuous Fourier transform must be adapted to signals that are sampled in space or time. The continuous Fourier transform of a function $f(x)$ is defined as

$$F(u) = \int_{-\infty}^{\infty} f(x)\, e^{-j2\pi ux}\, dx, \qquad f(x) = \int_{-\infty}^{\infty} F(u)\, e^{j2\pi ux}\, du.$$

Both variables x and u are continuous, and the signal extends over the entire real axis. To move toward a discrete representation, the continuous function must be sampled and later quantized.

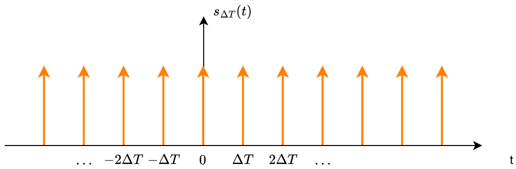
3.1 Sampling of a Continuous Signal

Sampling converts a continuous function $f(t)$ into a sequence of discrete values taken at regularly spaced intervals ΔT.



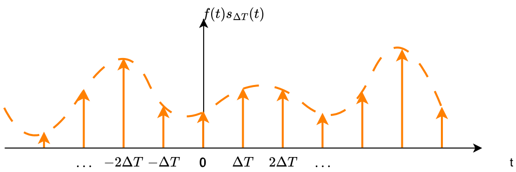
Conceptually, sampling is equivalent to multiplying $f(t)$ by a sampling function consisting of impulses spaced ΔT apart,

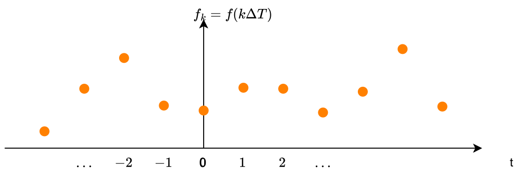
$$s_{\Delta T}(t) = \sum_{k=-\infty}^{\infty} \delta(t - k\Delta T),$$



so that the sampled signal becomes

$$\tilde{f}(t) = f(t)\, s_{\Delta T}(t).$$





This process extracts the value of $f(t)$ at every sampling instant because the Dirac delta has the sifting property

$$\int_{-\infty}^{\infty} f(t)\, \delta(t)\, dt = f(0), \qquad \int_{-\infty}^{\infty} f(t)\, \delta(t - t_0)\, dt = f(t_0).$$

The sampling period is $T = \Delta T$, and the sampling frequency is $f_s = 1/T$. Sampling turns the continuous function into a discrete sequence, while its effect in the frequency domain follows directly from the convolution theorem. If $F(u)$ is the Fourier transform of $f(t)$, then

$$\tilde{F}(u) = F(u) * S(u),$$

where the transform of the impulse train is another impulse train,

$$S(u) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} \delta\left(u - \frac{n}{\Delta T}\right).$$

3.2 Fourier Transform of the Sampling Function

Because $s_{\Delta T}(t)$ is periodic with period ΔT, it can be written as a Fourier series,

$$s_{\Delta T}(t) = \sum_{k=-\infty}^{\infty} c_k \, e^{j2\pi \frac{k}{\Delta T} t},$$

with coefficients

$$c_k = \frac{1}{\Delta T} \int_{-\Delta T/2}^{\Delta T/2} s_{\Delta T}(t) \, e^{-j2\pi \frac{k}{\Delta T} t} \, dt.$$

On this interval the sampling function contains a single impulse at $t = 0$, so all coefficients become

$$c_k = \frac{1}{\Delta T}.$$

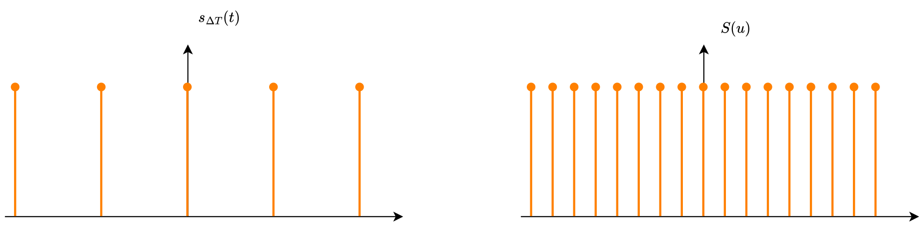
Hence the series reduces to

$$s_{\Delta T}(t) = \frac{1}{\Delta T} \sum_{k=-\infty}^{\infty} e^{j2\pi \frac{k}{\Delta T} t}.$$

Using the fact that the Fourier transform of $e^{j2\pi \frac{k}{\Delta T} t}$ is $\delta(u - \frac{k}{\Delta T})$, we obtain

$$S(u) = \frac{1}{\Delta T} \sum_{k=-\infty}^{\infty} \delta\left(u - \frac{k}{\Delta T}\right),$$

which is an impulse train whose spacing is $1/\Delta T = f_s$.



3.3 Effect of Sampling in the Frequency Domain

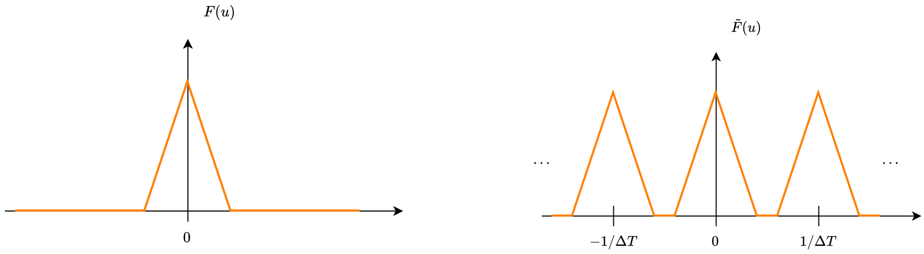
Applying the convolution theorem,

$$\tilde{F}(u) = F(u) * S(u),$$

and substituting the expression for $S(u)$ gives

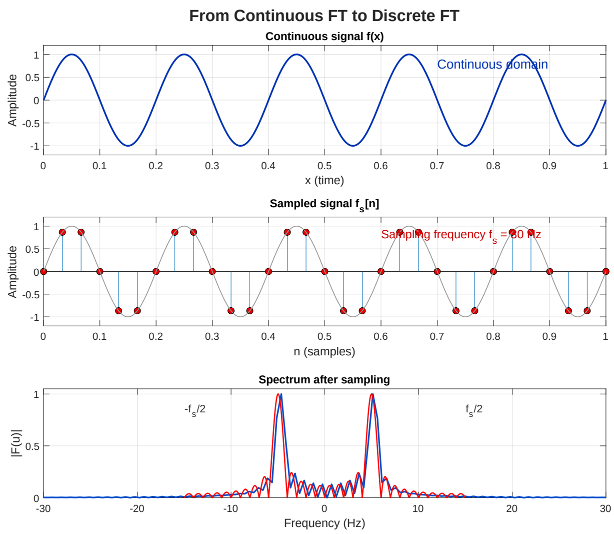
$$\tilde{F}(u) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} F\left(u - \frac{n}{\Delta T}\right).$$

Thus the spectrum of the sampled signal consists of infinitely many shifted copies of the original continuous spectrum. The spacing between these copies is exactly $1/\Delta T = f_s$.



3.4 Illustrations

Sampling a sinusoidal signal produces a discrete sequence, and its frequency spectrum becomes periodic. A schematic example shows a continuous signal, its sampled version, and the resulting periodic spectrum. The periodicity is determined by the sampling frequency f_s, and the replicated spectra clearly illustrate how sampling repeats $F(u)$ at intervals of f_s.

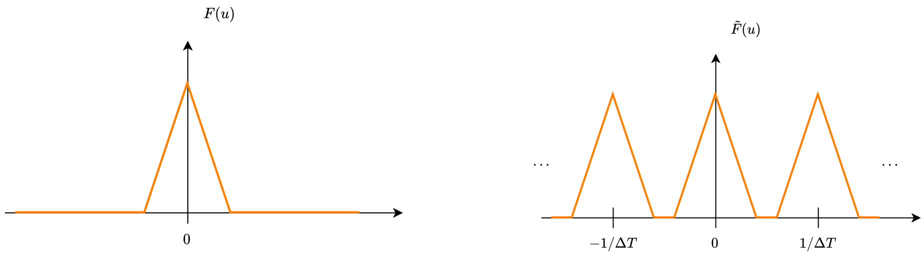


The figure provides a compact illustration of how a continuous signal becomes discrete through sampling and how this affects its Fourier spectrum. The top plot shows a continuous sinusoidal function $f(x)$, defined for all real values of x. The middle plot displays the sampled version of the same signal, denoted $f_s[n]$, where only values at equally spaced sampling instants are retained. The bottom plot shows the resulting periodic frequency spectrum that arises after sampling. The blue curve represents the original continuous spectrum, while the red impulses correspond to the sampled spectrum, demonstrating how sampling produces periodic repetitions of the original Fourier transform.

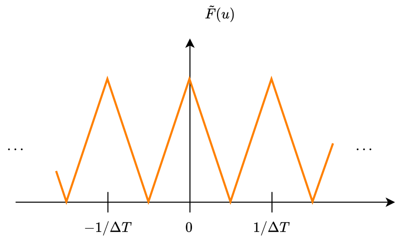
4 Aliasing and the Sampling Theorem

When a continuous signal is sampled, its Fourier transform becomes an infinite, periodic sequence of shifted copies of the original spectrum. The distance between these copies is $1/\Delta T$, which is equal to the sampling rate. If the sampling rate is sufficiently high, the copies remain well separated and the signal can be reconstructed without distortion. If the sampling rate is reduced, the spacing between spectral copies decreases and the spectra may overlap.

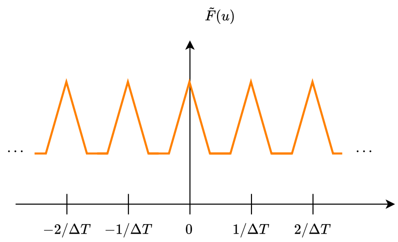
When the sampling rate is high, we obtain clear separation of all spectral copies, corresponding to an over-sampled signal.



If the sampling frequency is just sufficient to keep the copies touching but not overlapping, we speak of critical sampling.



Once the sampling rate falls below this minimum value, the spectral copies overlap and under-sampling occurs.



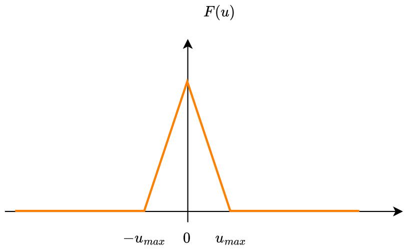
When under-sampling occurs, the overlapping spectral components produce false low-frequency components. This phenomenon is known as aliasing, meaning that distinct high-frequency signals take on a "false identity" after sampling.

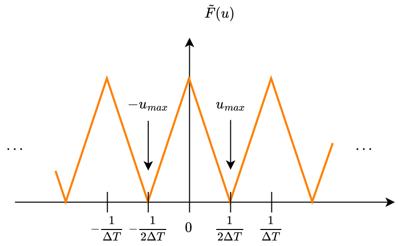
4.1 Nyquist–Shannon Sampling Theorem

A continuous signal can be perfectly reconstructed from its samples if the sampling frequency satisfies

$$f_{\max} < \frac{f_s}{2},$$

where $f_{\max}$ is the highest frequency present in the signal. The value $f_s/2$ is called the Nyquist frequency. A function whose Fourier transform is zero outside some finite interval $[-u_{\max}, u_{\max}]$ is called a band-limited function.

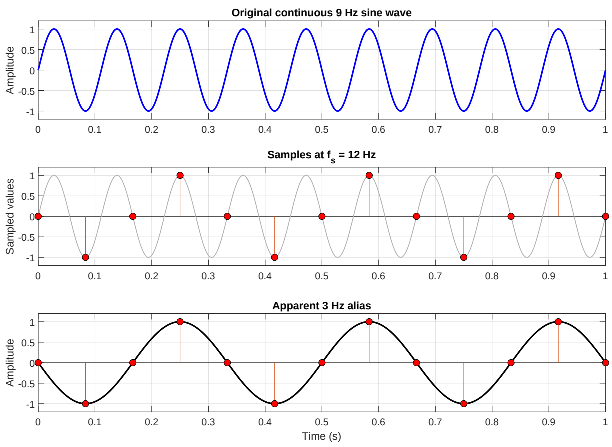




4.2 Concept of Aliasing

Aliasing arises when a signal is sampled below the Nyquist rate. In the frequency domain, the spectrum of a sampled signal is periodic with period f_s. If $f_s < 2f_{\max}$, the repeated spectra overlap, causing high frequencies to fold back into the baseband and appear as false, lower frequencies.

A typical example is a 9 Hz sine wave sampled at $f_s = 12$ Hz: the sampled data appear to correspond to a 3 Hz sine wave, which is an alias.



The top plot shows the original 9 Hz continuous sine wave, the middle plot shows the samples taken at 12 Hz, and the bottom plot illustrates the apparent 3 Hz reconstructed wave. These two different signals coincide at the sampling instants, and without additional information, we have no way of knowing how the function behaves between samples; such a pair of functions is called an aliased pair.

4.3 Avoiding Aliasing

To prevent aliasing, high-frequency components must be removed before sampling. This is achieved by an anti-aliasing filter, which is a low-pass filter applied to the signal so that no frequency above $f_s/2$ remains. The trade-off is that filtering prevents aliasing but also reduces image sharpness and fine detail.

5 Discrete Fourier Transform

Sampling alone is not sufficient, because a computer can only process a finite number of samples. If we restrict the signal to N samples and also sample its spectrum, we obtain the Discrete Fourier Transform (DFT). For a discrete signal $f(n)$, the DFT is defined as

$$F(u) = \sum_{n=0}^{N-1} f(n)\, e^{-j2\pi\frac{un}{N}}, \qquad u = 0, 1, \ldots, N-1,$$

and the inverse transform reconstructs the signal through

$$f(n) = \frac{1}{N}\sum_{u=0}^{N-1} F(u)\, e^{j2\pi\frac{un}{N}}, \qquad n = 0, 1, \ldots, N-1.$$

In this formulation, $f(n)$ represents a finite discrete signal, while $F(u)$ is a discrete spectrum that is periodic with period N. The transform is built on complex exponential basis functions of the form $e^{-j2\pi kn/N}$.

5.1 From the sampled continuous transform to the DFT

To understand how the DFT arises, consider the sampled function

$$\tilde{f}(t) = \sum_{k=-\infty}^{\infty} f(t)\,\delta(t - k\Delta T).$$

Its Fourier transform is given by

$$\tilde{F}(u) = \int_{-\infty}^{\infty} \tilde{f}(t)\, e^{-j2\pi ut}\, dt.$$

Substituting the expression for the sampled function yields

$$\tilde{F}(u) = \int_{-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f(t)\,\delta(t - k\Delta T)\, e^{-j2\pi ut}\, dt = \sum_{k=-\infty}^{\infty} \int_{-\infty}^{\infty} f(t)\,\delta(t - k\Delta T)\, e^{-j2\pi ut}\, dt.$$

Using the sifting property of the Dirac delta function, this reduces to

$$\tilde{F}(u) = \sum_{k=-\infty}^{\infty} f(k)\, e^{-j2\pi uk\Delta T}.$$

Thus, although the sequence $f(k)$ is discrete, its continuous Fourier transform $\tilde{F}(u)$ remains a continuous function and is periodic with period $1/\Delta T$.

5.2 Sampling one period of the continuous spectrum

Since $\tilde{F}(u)$ is continuous and infinitely periodic, we must sample one period of this spectrum. We choose N equally spaced sample points over the interval from $u = 0$ to $u = 1/\Delta T$. To obtain exactly N samples, we set

$$u = \frac{n}{N}, \qquad n = 0, 1, \ldots, N-1.$$

By inserting these sampling points into the expression for $\tilde{F}(u)$, we arrive at the discrete expression

$$F(u) = \sum_{n=0}^{N-1} f(n)\, e^{-j2\pi\frac{un}{N}}, \qquad u = 0, 1, \ldots, N-1,$$

which is precisely the Discrete Fourier Transform.

5.3 Truncation and Leakage

Aliasing showed that sampling produces a periodic spectrum and that coarse sampling causes overlap of spectral copies. In practice, however, we face a related difficulty: we never analyze an infinite or perfectly periodic signal. Instead, we work with a finite segment of data, effectively multiplying the signal by a rectangular window. This truncation introduces discontinuities at the boundaries, creating spectral leakage. Even a pure sinusoid generates energy in neighboring frequency bins when it does not fit exactly into the analysis window.

The problem arises because the DFT assumes the input signal is periodic across the analysis interval. If an integer number of periods does not fit inside the window, the artificial discontinuity introduces additional frequency components. Windowing addresses this by reducing the signal smoothly to zero at the edges. Multiplying by a window function softens the transitions and reduces leakage.

5.3.1 Illustration of Spectral Leakage

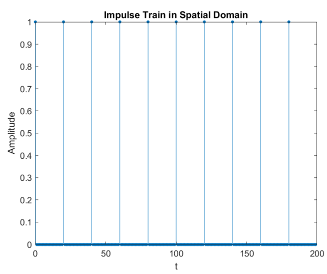
For an ideal discrete impulse train, the Fourier transform again forms a periodic train of impulses,

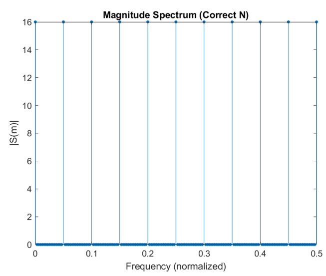
$$S(u) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} \delta\left(u - \frac{n}{\Delta T}\right),$$

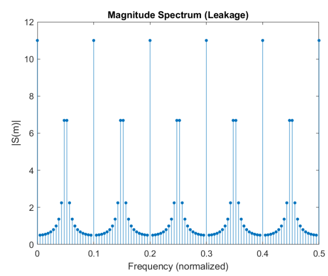
so the spectral peaks occur at integer multiples of $1/\Delta T$. Whether these impulses align with the discrete DFT frequency bins depends entirely on the chosen data length N.

In the first case, the length N is chosen such that $N = k\Delta T$, meaning that an integer number of periods of the impulse train fits exactly into the analysis window. Under this condition, the spectral impulses fall precisely on the DFT frequency bins $\omega = 2\pi m/N$. The resulting spectrum therefore contains clean, sharp peaks with no energy spilling into neighboring frequencies.

In the second case, when N is not an integer multiple of ΔT, the periodic impulse structure does not fit into the DFT window. As a consequence, the spectral impulses do not fall on the DFT bin centers. Instead, energy spreads into adjacent bins, producing spectral leakage.

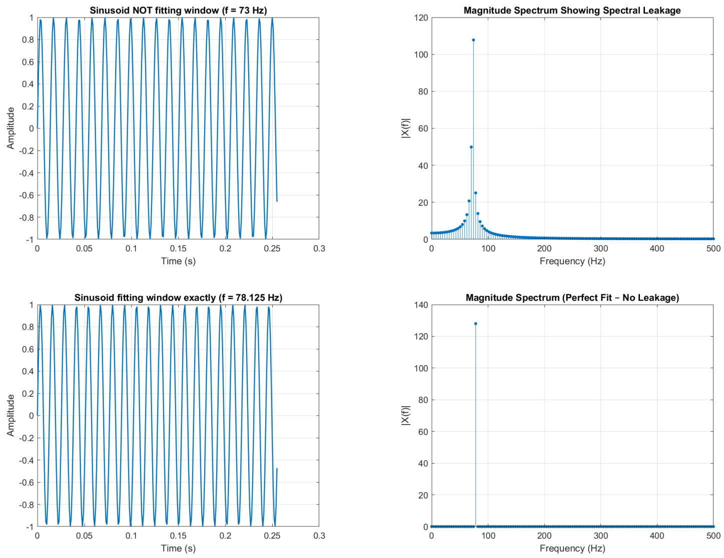






The first image shows the original impulse train in the time domain. The second image depicts the corresponding spectrum for the correct-length case, where the impulses align perfectly with the DFT frequency bins. The third image illustrates the incorrect-length case, where misalignment causes energy to spread and results in noticeable spectral leakage.

A second example of spectral leakage can be observed when the signal is a pure sinusoid that does not fit perfectly into the analysis window. Because the DFT implicitly assumes periodicity over the chosen interval, any mismatch between the sinusoid's period and the window length produces artificial discontinuities at the boundaries. When such a truncated sinusoid is transformed using the DFT, these discontinuities cause energy to spread into neighboring frequency bins. Instead of a single clean spectral peak, the resulting spectrum contains several adjacent components, revealing the presence of leakage.



The first image shows a sinusoid whose period does not match the analysis window, and the second image displays the corresponding DFT spectrum, where the energy is distributed across several neighboring frequencies. The third image illustrates a sinusoid that fits the window exactly, and the fourth image shows its DFT spectrum, which now contains a single sharp frequency component with no leakage.

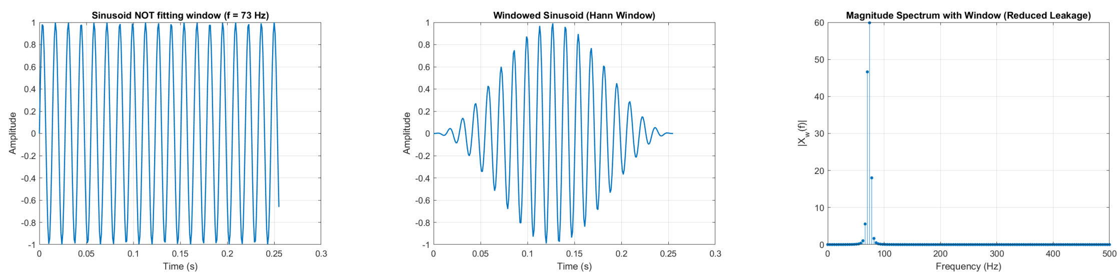
5.4 Effect of Windowing on Spectral Leakage

A sinusoidal signal that does not fit exactly into the analysis window produces discontinuities at the boundaries, and these discontinuities manifest in the frequency domain as spectral leakage. One way to reduce this effect is to multiply the signal by a window function such as the Hann window,

$$x_w(n) = x(n) \cdot w(n),$$

which gradually suppresses the signal at the edges. By smoothing the transitions to zero, the window function reduces the abrupt jumps that cause leakage.

As a result of windowing, the overall leakage level decreases significantly. The main spectral peak becomes wider but also much cleaner, and the side lobes that previously introduced energy into neighboring frequency bins are strongly reduced.



The first image shows a sinusoid that does not fit the window and therefore creates discontinuities. The second image displays the signal after windowing, and the third image shows how the spectrum becomes cleaner with reduced side lobes.

5.5 Window Functions

Window functions are expressly designed to reduce these discontinuities and to limit spectral leakage. The simplest choice is the rectangular window,

$$w(n) = 1, \qquad 0 \leq n < N,$$

which does not attenuate the signal at the edges and therefore produces strong leakage.

Smoother windows improve performance. The Hann window is defined as

$$w(n) = 0.5\left(1 - \cos\frac{2\pi n}{N-1}\right),$$

the Hamming window as

$$w(n) = 0.54 - 0.46\cos\frac{2\pi n}{N-1},$$

and the Blackman window as

$$w(n) = 0.42 - 0.5\cos\frac{2\pi n}{N-1} + 0.08\cos\frac{4\pi n}{N-1}.$$

Although their shapes differ, they all share the purpose of smoothing the edges and reducing leakage.

5.6 Time and Frequency Domain View

Windowing a signal corresponds to multiplying the time-domain sequence by a window function,

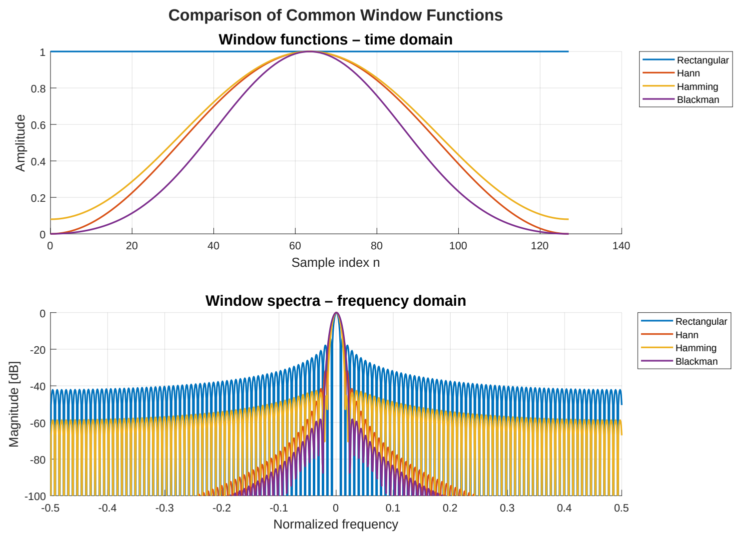
$$x_w(n) = x(n) \cdot w(n).$$

In the frequency domain, this multiplication becomes a convolution,

$$X_w(k) = X(k) * W(k),$$

which explains why the spectral peak becomes broader yet smoother. The window spreads energy from a single sharp frequency into a controlled pattern determined by the frequency response of $W(k)$.

This leads to a fundamental trade-off: a narrow main lobe yields good frequency resolution, whereas low side lobes reduce leakage. No single window provides optimal performance for all applications, so the choice of window depends on the specific task.



The figure compares several window functions by showing both their shapes in the time domain (top) and the magnitude of their frequency responses (bottom). As the windows become smoother, the main lobe widens and the side lobes decrease, illustrating the balance between spectral resolution and leakage reduction.

6 2D Discrete Fourier Transform

The Fourier Transform is not restricted to one-dimensional signals. In image processing we work with two-dimensional data $f(x, y)$, where the variables x and y represent spatial coordinates and $f(x, y)$ denotes the corresponding intensity values. The *2D Discrete Fourier Transform (2D DFT)* expresses an image in terms of its spatial frequency content. Just as in the one-dimensional case, the transform decomposes the signal into a sum of complex exponentials, but now the decomposition occurs simultaneously in two orthogonal spatial directions.

The 2D DFT describes how strongly each horizontal and vertical frequency component is present in the image.

6.1 Definition

For an image $f(x, y)$ of size $M \times N$, the 2D DFT is defined by

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

The inverse transform reconstructs the spatial-domain image via

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

6.2 Relation to the 1D DFT

The 2D DFT can be computed by repeated application of the 1D DFT. First, a 1D DFT is applied to every row of the image. Then, a 1D DFT is applied to each column of the intermediate result:

$$F(u, v) = \mathrm{DFT}_x\big(\mathrm{DFT}_y(f(x, y))\big).$$

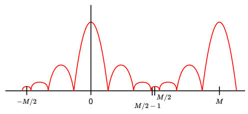
This separability property greatly simplifies both theoretical analysis and practical implementation.

6.3 Basic Properties

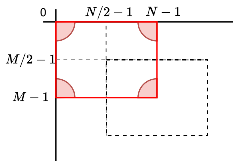
The 2D DFT inherits several key properties from its 1D counterpart. First, $F(u, v)$ is periodic in both frequency directions. Second, for real-valued images the transform exhibits conjugate symmetry:

$$F(M - u,\, N - v) = F^*(u, v),$$

which implies that the magnitude spectrum is symmetric about the origin.



$F(u)$



$F(u, v)$

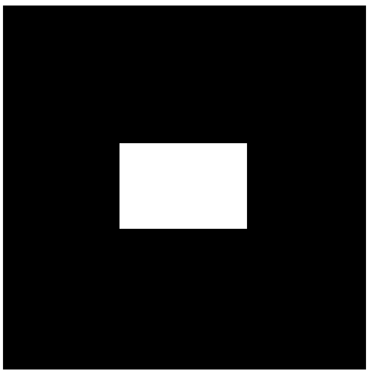
It is often advantageous to re-center the spectrum by shifting the zero frequency component to the middle of the matrix. This is achieved by the mapping

$$F_{\text{center}}(u, v) = F\Big((u + \tfrac{M}{2}) \bmod M,\ (v + \tfrac{N}{2}) \bmod N\Big)\,.$$

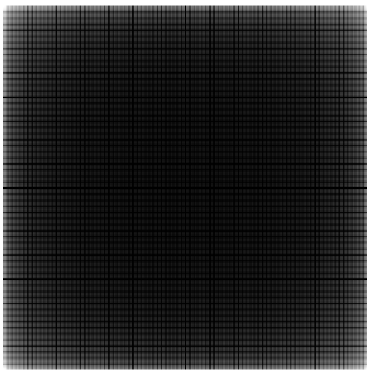
Re-centering the spectrum corresponds to multiplying the spatial image by a sign-alternating pattern:

$$f_{\text{shift}}(x, y) = f(x, y)\,(-1)^{x+y}.$$

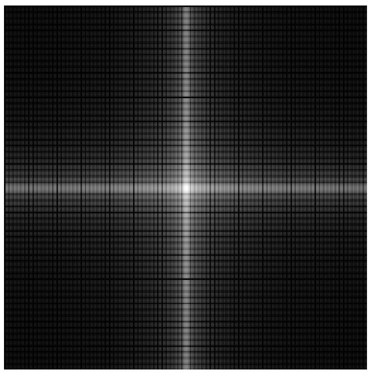
This phase modulation effectively relocates the origin of the DFT to the center of the image.



Spatial domain



Spectrum



Centered spectrum

Several further properties play an important role in image analysis. The transform is linear:

$$\mathrm{DFT}(af_1 + bf_2) = a\,\mathrm{DFT}(f_1) + b\,\mathrm{DFT}(f_2).$$

A spatial translation results in a corresponding phase modification in the frequency domain. Finally, the scaling property indicates that narrow spatial features produce broad frequency spectra, whereas broad features give rise to narrow spectra.

7 Interpretation of the 2D Spectrum

The two–dimensional Discrete Fourier Transform generally yields complex values and can therefore be expressed in polar form as

$$F(u, v) = R(u, v) + jI(u, v) = |F(u, v)|e^{j\phi(u,v)}.$$

The quantity $|F(u, v)|$ is called the Fourier (frequency) spectrum, while the phase angle

$$\phi(u, v) = \arctan \frac{I(u, v)}{R(u, v)}$$

defines the phase spectrum. The power spectrum is given by $P(u, v) = |F(u, v)|^2$. All of these quantities—the magnitude, the phase, and the power—form arrays of size $M \times N$.

The magnitude spectrum is an even function with respect to the origin,

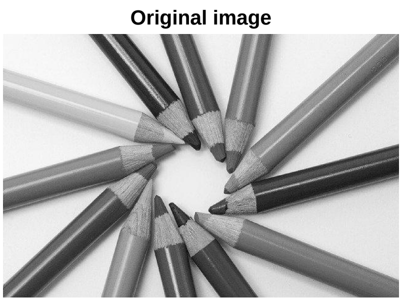
$$|F(u, v)| = |F(-u, -v)|,$$

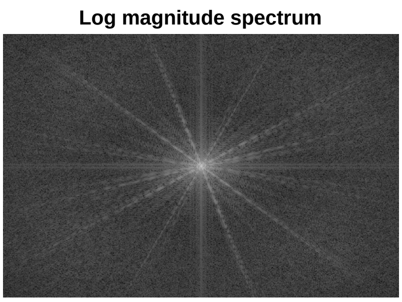
whereas the phase spectrum is odd,

$$\phi(u, v) = -\phi(-u, -v).$$

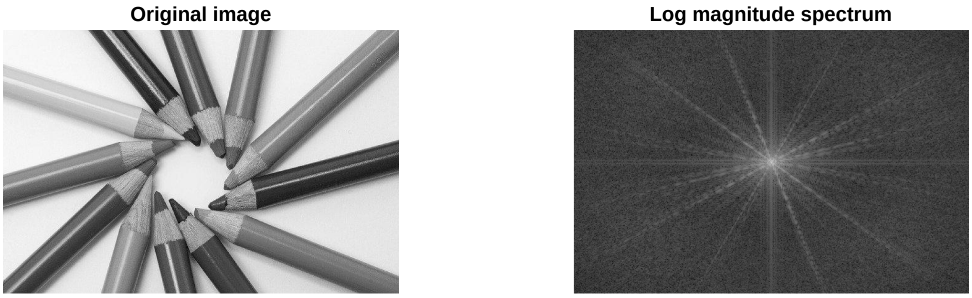
These symmetry properties follow directly from the conjugate symmetry of the Fourier transform of real-valued images.

The point $(u = 0, v = 0)$, often referred to as the origin of the spectrum, corresponds to the DC component, that is, the overall average brightness of the image. Higher spatial frequencies represent rapid changes in intensity and therefore correspond to edges and textures. The orientation of a frequency component reflects the orientation of structures in the image: horizontal structures give rise to vertical frequency components and vice versa. The magnitude spectrum reveals the strength of the spatial frequencies present in the image, while the phase spectrum governs the geometric arrangement of structures and is essential for meaningful reconstruction.





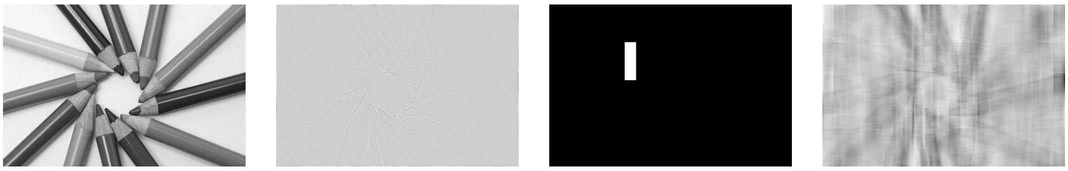
When visualising a 2D spectrum, the central bright region represents low frequencies associated with smooth areas of the image, whereas the outer regions correspond to high frequencies that encode edges and fine details. Oriented or directional patterns in the image appear as lines or elongated structures in the frequency domain. Because the magnitude values may span several orders of magnitude, a logarithmic display is typically used to make the full range visible.



The Fourier transform of an image separates its structure into magnitude and phase components according to

$$F(u,v) = |F(u,v)|\, e^{j\phi(u,v)}.$$

The magnitude describes the distribution of energy among the frequencies, while the phase encodes the spatial structure. This distinction becomes clear when comparing different reconstructions. If the magnitude is discarded and set to one while the phase remains unchanged, the resulting image still retains the essential visual structure, although not the correct brightness. Similarly, replacing the magnitude with that of a simple rectangle while preserving the original phase still produces a recognisable image. These examples show that phase carries the fundamental structural information, while magnitude determines the intensity distribution.



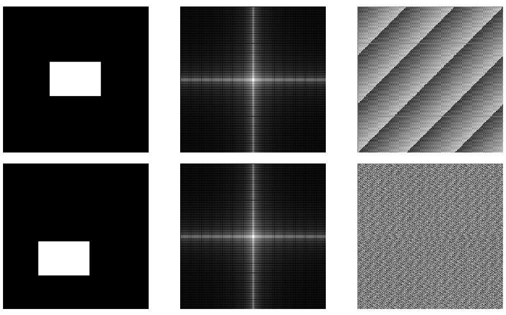
Spatial transformations also manifest themselves clearly in the spectrum. If an object in the image is shifted, the magnitude of its Fourier transform remains unchanged, whereas the phase acquires a linear term. For example, consider two images that contain the same rectangle, but at different positions. The spectra of these two images have identical magnitudes, but their phases differ according to the amount of translation.

A spatial shift does not affect the magnitude spectrum,

$$|F(u,v)| = \text{constant under translation},$$

but modifies the phase through

$$f(x - x_0, y - y_0) \xrightarrow{\mathcal{F}} F(u,v)e^{-j2\pi\left(\frac{ux_0}{M} + \frac{vy_0}{N}\right)}.$$

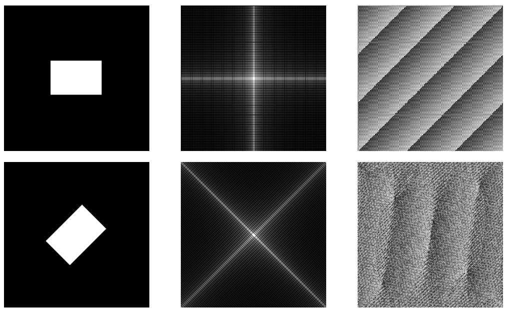


Rotations behave differently. If an image is rotated in the spatial domain, the Fourier transform undergoes the same rotation. For example, consider two images containing the same rectangle, one of them rotated. The magnitude spectrum of the rotated image is simply a rotated version of the original spectrum, and the same relation holds for the phase. The shape of the magnitude does not change; only its orientation does.

Formally, if R denotes the rotation matrix, then

$$f_{\text{rot}}(x, y) \xrightarrow{\mathcal{F}} F_{\text{rot}}(u, v) = F(R^{-1}(u, v)).$$

Thus the magnitude spectrum is preserved up to rotation, and the phase is rotated accordingly.

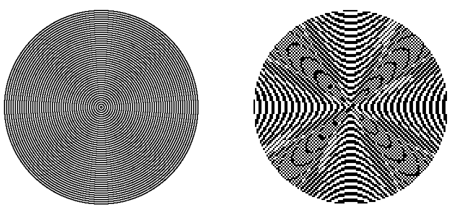


8 Aliasing in Images

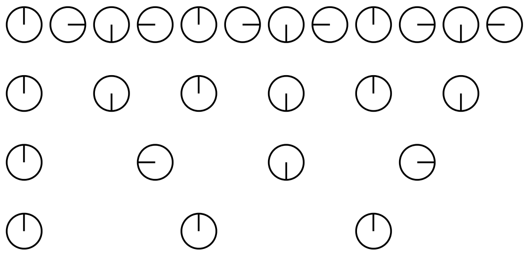
Aliasing is not limited to one–dimensional signals; it also occurs in two–dimensional images. The phenomenon arises whenever fine spatial details exceed the sampling frequency of the pixel grid. In such cases, the sampling process is unable to represent the original high–frequency structures correctly, and false patterns may be introduced into the image.

Two forms of aliasing are typically distinguished. Spatial aliasing appears when an image is under-sampled; this occurs most visibly in scenes containing repetitive patterns, where the sampling grid interferes with the underlying structure. Temporal aliasing, on the other hand, arises in sequences of images when the time interval between frames is too large to capture the motion faithfully.

The visual manifestations of aliasing include artificial repetitive patterns such as moiré, beating effects between textures and the pixel grid, and loss of high-frequency detail. Common examples include photographs of striped fabrics, tiled roofs, or grids that produce interference patterns.



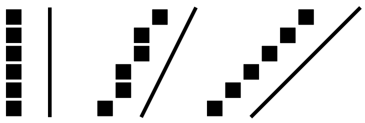
Temporal aliasing appears in dynamic scenes when the sampling rate in time is insufficient. Objects moving quickly between frames may seem to jump or reverse direction, or they may produce other misleading motion cues.



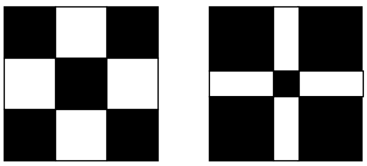
8.1 Examples of Spatial Aliasing

Spatial aliasing takes several characteristic forms. The moiré effect creates false repetitive patterns, often produced when two high-frequency grids interfere.

Another artifact is the appearance of jagged, staircase-like edges, commonly referred to as jaggies, which arise when smooth diagonal edges are sampled on a coarse pixel grid.



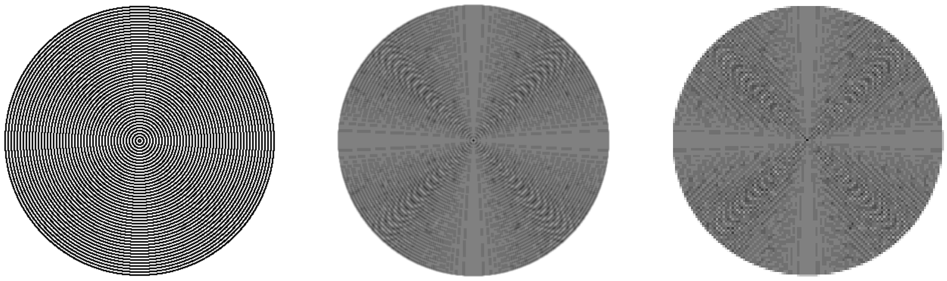
A further consequence of aliasing is the loss of fine detail. When the sampling grid cannot represent the spatial frequencies present in the scene, high-frequency information is irrecoverably lost.



8.2 Reducing Aliasing

Aliasing occurs whenever an image contains spatial frequencies higher than half the sampling rate, that is, above the Nyquist limit. Reducing aliasing therefore requires methods that either limit the presence of such frequencies or increase the sampling capability.

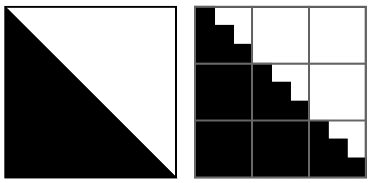
A standard approach is to apply a low-pass filter before sampling. By suppressing high-frequency content, the image becomes compatible with the sampling grid. Another option is to increase the sampling resolution, since a denser grid raises the Nyquist limit. Supersampling and multisampling constitute practical strategies frequently used in rendering: the image is generated at a higher resolution and subsequently downscaled, producing smoother edges and fewer artifacts.



8.2.1 Supersampling

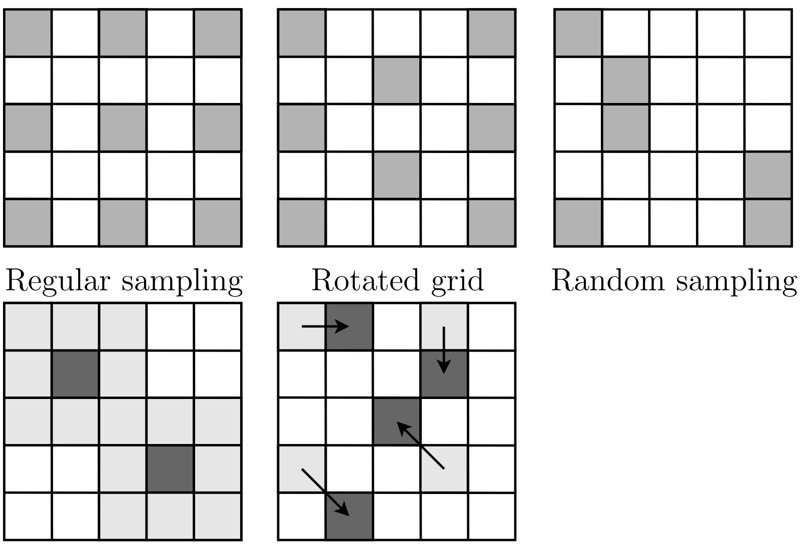
Supersampling increases the sampling frequency by subdividing each pixel into a grid of $n_x \times n_y$ high-resolution samples. These sub-samples, referred to as superpixels, provide multiple measurements within the area of a single output pixel. From each superpixel, a selection of m samples is used, where m may be smaller than the full oversampling count. Let the selected samples be denoted by s_i with weights w_i. The final pixel value is obtained by downsampling:

$$\sum_{i=0}^{m-1} w_i \, s_i.$$



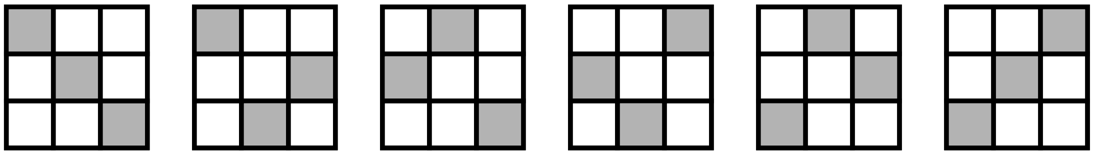
Different strategies may be used to choose the sampling points within each superpixel. Regular sampling places the samples at uniform positions within the grid, while rotated-grid sampling shifts this pattern by a rotation, reducing directional aliasing. Random sampling avoids regularity altogether, and Poisson disk sampling ensures that points are well-distributed while maintaining a minimum separation.

Jittered sampling introduces controlled randomness by perturbing regular grid positions. The N-rooks algorithm distributes samples such that no two samples share the same row or column, ensuring uniform coverage.



Regular sampling Rotated grid Random sampling

Jittered sampling Poisson disk



N-rooks algorithm

9 Windowing in 2D Fourier Analysis

In one–dimensional Fourier analysis, windowing is used to reduce spectral leakage by smoothing the boundaries of the analysis window. The same principle applies in two dimensions. When we extract a finite spatial region from an image, the abrupt boundaries of this region behave like strong discontinuities. These discontinuities introduce spurious frequency components in the two–dimensional spectrum, in direct analogy with the one–dimensional case.

To suppress this effect, a two–dimensional window function is applied before computing the local Fourier transform. A 2D window is typically constructed as the separable product

$$w(x, y) = w_x(x)\, w_y(y),$$

where each of the one–dimensional window functions may be chosen as Hann, Hamming, Blackman, Gaussian, or a similar smooth function. Applying such a window reduces the sharpness of the block boundaries and limits the resulting spectral leakage.

Windowing is generally unnecessary when computing the Fourier transform of an entire image. In this case the image is treated as one complete two–dimensional discrete signal, and applying a window would only distort the signal. For instance, a window would artificially darken the image near the borders and alter its true spectrum. Therefore, global 2D FFT computations are normally performed without windowing.

The need for windowing arises when computing the Fourier transform of a local region or image block. Extracting a block such as a 32×32 patch introduces artificial step discontinuities at the block boundaries. These discontinuities significantly affect the resulting 2D spectrum and produce leakage that obscures the true frequency content of the underlying image structure.

Applying a 2D window prior to the local Fourier transform smooths these boundary transitions. By reducing the discontinuities, the windowed block yields a more accurate estimate of local spatial frequencies.

9.1 Example: Block-Based FFT

Let $f(x, y)$ denote an image and let $B(x, y)$ be a rectangular block extracted from it. This block can be expressed as

$$B(x, y) = f(x, y)\, r(x, y),$$

where $r(x, y)$ is a rectangular window. The rectangular window introduces sharp edges at the block perimeter. In the frequency domain, these edges cause the block spectrum to be equal to a convolution

$$\mathcal{F}\{B\} = F * R,$$

where R is the sinc–like transform of the rectangular window. This convolution broadens the spectral components and leads to strong leakage.

To avoid this, the block may be multiplied by a smooth window $w(x, y)$, resulting in

$$B_w(x, y) = f(x, y)\, w(x, y),$$

which suppresses boundary artifacts and reduces the spectral spread in the frequency domain.

9.2 Windowing in Texture and Frequency Analysis

Local frequency methods such as Gabor filtering or the short–time Fourier transform naturally rely on window functions. A Gabor filter, for example, is defined as the product of a Gaussian window and a complex sinusoid. The Gaussian window provides spatial localization, controlled bandwidth, and minimal leakage, making it well suited for the analysis of textures, orientations, and spatial–frequency patterns present in images. Windowing is therefore fundamental for any method that attempts to characterize local frequency content.

9.3 Practical Use in Image Processing

In practical applications of 2D Fourier analysis, windowing is applied before computing the 2D DFT of an image block. The main purpose is to reduce edge discontinuities and to improve the quality of spectral estimation in localized image regions. This is essential in block-based texture descriptors, in frequency-domain filtering of non-periodic images, and in preprocessing steps designed to suppress ringing artifacts in FFT-based processing.

10 Applications of the 2D DFT

The two–dimensional Fourier transform has numerous applications in image processing. In filtering, low-pass filters are used for smoothing and noise reduction, while high-pass filters enhance edges and fine details. In pattern and texture analysis, periodic textures produce characteristic peaks in the frequency domain, and the orientation of these peaks reveals the dominant texture direction.

Image restoration also benefits from Fourier methods. Periodic noise can be removed by attenuating specific frequency components, while deblurring may be achieved using inverse filtering or related spectral techniques. In all these tasks, the 2D DFT provides a powerful tool for examining and manipulating the frequency content of images.

11 Motivation for the Fast Fourier Transform

The next topic builds on the properties of the Discrete Fourier Transform and addresses its major practical limitation: computational cost. Although the DFT is mathematically elegant and fundamental to frequency analysis, its direct computation is extremely demanding. For a one–dimensional signal of length N, evaluating the DFT requires on the order of $O(N^2)$ arithmetic operations. For two–dimensional data, such as an $N \times N$ image, the situation becomes even more severe, since a straightforward implementation involves approximately $O(N^4)$ operations. Such complexity quickly becomes prohibitive as the signal size grows.

To make Fourier analysis feasible in real applications, a more efficient computational method is needed — one that preserves the exact output of the DFT but dramatically reduces the amount of required computation. The breakthrough comes from exploiting the mathematical structure of the Fourier matrix, which allows the original quadratic or quartic complexity to be reduced to $O(N \log N)$.

The algorithm achieving this reduction is known as the *Fast Fourier Transform (FFT).* It produces exactly the same result as the DFT, but with a computational cost several orders of magnitude lower. Because of this efficiency, the FFT is considered one of the most significant numerical algorithms ever developed and forms the foundation of modern signal and image processing.