Discrete Fourier Transform

Scientific computing

Mgr. Markéta Trnečková, Ph.D.

Fourier Transform in Image Processing

- **Goal of the lectures**
  - understand how frequency representation helps to analyze and process images
  - learn the principles of the **Discrete Fourier Transform (DFT)** and **Fast Fourier Transform (FFT)**
  - explore applications of FFT in **image filtering**, **enhancement**, and **restoration**
- **Structure**
  - Lecture 1: Discrete Fourier Transform – sampling, aliasing, windowing
  - Lecture 2: Fast Fourier Transform – algorithm and efficiency
  - Lecture 3: Applications of FFT in image processing
- **Practical sessions**
  - DFT and aliasing in 1D and 2D
  - Implementing and timing FFT
  - Frequency-domain filtering of images

Motivation – Why the Fourier Transform?

- **Core idea**
  - Any (well-behaved) signal or image can be represented as a **superposition of sinusoids** (sine and cosine waves) of different frequencies, amplitudes and phases.
  - Frequency representation reveals **periodicity, orientation and scale** that are hard to see in the spatial/temporal domain.

- **Mathematical intuition (continuous case)**

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

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

  - **Euler formula**:

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

  - F is called **frequency domain**
  - units of frequency domain are cycles per unit of independent variable of input function

Motivation – Why the Fourier Transform?

Key properties that make FT indispensable

- **Linearity**: transform of a sum = sum of transforms.
- **Convolution theorem**: convolution in space/time $\longleftrightarrow$ multiplication in frequency,

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

  This is the basis for fast filtering (use FFT to compute large convolutions efficiently).
- **Energy preservation (Parseval)**: total signal energy is same in both domains.

Why frequency domain helps in image processing

- **Noise removal / smoothing**: low-pass filtering removes high-frequency noise while preserving coarse structures.
- **Edge enhancement**: high-pass filters emphasize edges (high-frequency components).
- **Periodic artifact removal**: narrowband (notch) filters can remove periodic interference (e.g. grid or ripple).
- **Texture and pattern analysis**: repeated patterns appear as localized peaks in the spectrum; orientation is encoded by location of those peaks.
- **Efficient convolution / correlation**: large-kernel filtering or correlation is faster in frequency domain via FFT.

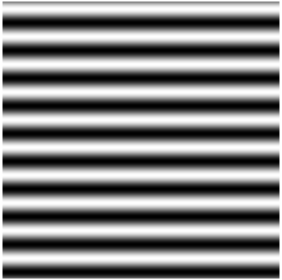
Motivation – Why the Fourier Transform?

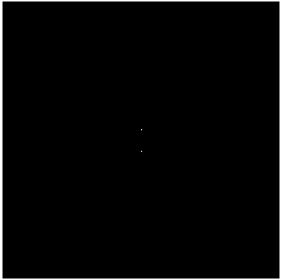
Intuition in 2D images

- Low frequencies (near DC) $\Rightarrow$ smooth, slowly varying components (overall illumination, coarse shapes).
- High frequencies (far from center after `fftshift`) $\Rightarrow$ fine details, edges, noise.
- Directional structures produce **anisotropic patterns** in the spectrum (peaks aligned with the perpendicular direction).

Short illustrative example

- A horizontal sinusoidal stripe pattern in the image domain produces two symmetric peaks on the vertical axis of the 2D spectrum (frequency location encodes spatial frequency; axis encodes orientation).





From Continuous FT to Discrete FT

- **Motivation**
  - In image processing, we always work with **discrete digital data**.
  - The continuous Fourier Transform (FT) must therefore be adapted for signals sampled in both **space** and **time**.

- **Continuous Fourier Transform (CFT)**

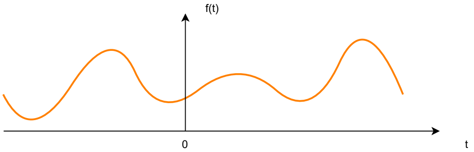
$$F(u) = \int_{-\infty}^{\infty} f(x)\, e^{-j2\pi ux}\, dx$$

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

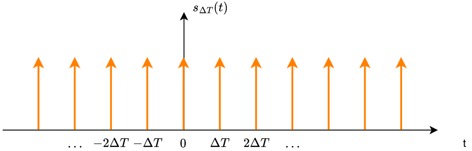
  - Continuous variables x, u
  - Infinite range and continuous signal $f(x)$

Sampling the signal

- we need to convert continuous function to discrete values – sampling and quantization
- continuous function $f(t)$ – sampling at uniform intervals ΔT



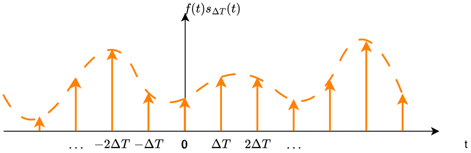
- sampling = multiplying $f(t)$ by sampling function (train of impulses ΔT units apart $s_{\Delta T}(t) = \sum_{k=-\infty}^{\infty} \delta(t - k\Delta T))$



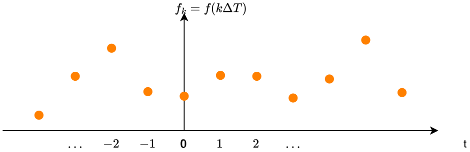
$$\delta(t) = \begin{cases} \infty, & \text{if } t = 0, \\ 0, & \text{if } t \neq 0. \end{cases}$$

Sampling the signal

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



- **Sampled function**



Sampling the signal

- T is the **sampling period**, and $f_s = 1/T$ the sampling frequency.
- Sampling turns the continuous signal into a discrete sequence of values.
- Let $F(u)$ denote Fourier transform of continuous $f(t)$
- $\tilde{f}(t) = f(t)s_{\Delta T}(t)$
- form convolution theorem $f(t) \cdot g(t) \ \rightarrow F(u) * G(u)$
  $\tilde{F}(u) = F(u) * S(u)$
- $S(u) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} \delta(u - \frac{n}{\Delta T})$

Sampling the signal

Example

Assume $s_{\Delta T}(t)$. Derive the expression for its Fourier transform $S(u)$
Sketch both $s_{\Delta T}(t)$ and $S(u)$. Explain how the ΔT affects the resulted $S(u)$.

Sampling the signal

- assume $f = \delta(t)$
- $\int_{-\infty}^{\infty} \delta(t)\, dt = 1$ (Acts as an infinitesimally narrow pulse of unit area)
- for any continuous function $f(t)$,

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

called the **sifting property**.

The sifting property of the Dirac delta function states that the delta function "picks out" the value of a function at the point where its argument is zero. Because the delta function is zero everywhere except at $t = 0$ and has a total area of one, the integral of $f(t)\delta(t)$ over all t yields exactly the value $f(0)$.

Sampling the signal

- assume $f = \delta(t - t_0)$ (pulse is located at general point)
- for any continuous function $f(t)$,

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

- we obtain

$$F(u) = \int_{-\infty}^{\infty} \delta(t - t_0)\,e^{-j2\pi ut}\,dt$$

- which is equal to

$$F(u) = e^{-j2\pi ut_0}$$

Sampling the signal

- assume $s_{\Delta T}(t)$ – periodic function with period ΔT
- Fourier series

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

  where

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

- when we consider interval $[-\Delta T/2, \Delta T/2]$, we have only one impulse located at origin, therefore

$$c_k = \frac{1}{\Delta T} \int_{-\Delta T/2}^{\Delta T/2} \delta(t) \, e^{-j2\pi \frac{k}{\Delta T} t} \, dt = \frac{1}{\Delta T} e^{-j2\pi \frac{k}{\Delta T} \cdot 0} = \frac{1}{\Delta T} e^{0} = \frac{1}{\Delta T}.$$

- Fourier series than becomes

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

Sampling the signal

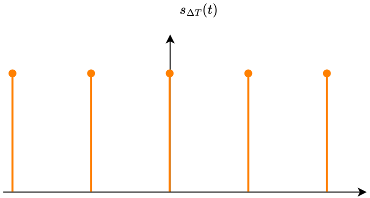
- $f = e^{j2\pi \frac{k}{\Delta T} t}$ generates one impulse

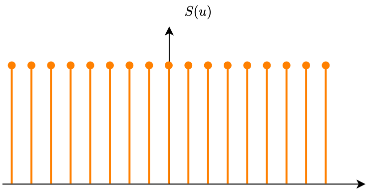
$$F(u) = \delta(u - \frac{k}{\Delta T})$$

- so

$$S(u) = \mathcal{F}\{s_{\Delta T}(t)\} = \frac{1}{\Delta T} \sum_{k=-\infty}^{\infty} \mathcal{F}\left\{e^{j2\pi \frac{k}{\Delta T} t}\right\} = \frac{1}{\Delta T} \sum_{k=-\infty}^{\infty} \delta\left(u - \frac{k}{\Delta T}\right)$$

- result is also impulse train, whose period is $1/\Delta T$





Sampling the signal

- form convolution theorem $f(t) \cdot g(t) \ \rightarrow F(u) * G(u)$
  $\tilde{F}(u) = F(u) * S(u)$
- $S(u) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} \delta(u - \frac{n}{\Delta T})$
- definition of convolution

$$(f * h)(t) = \int_{-\infty}^{\infty} f(\tau)h(t-\tau)\, d\tau$$

- i.e

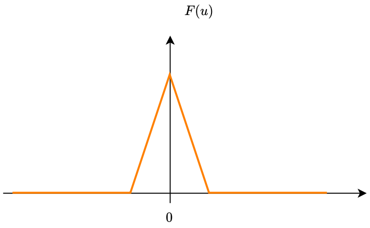
$$\tilde{F}(u) = F(u) * S(u) = \int_{-\infty}^{\infty} F(\tau)S(t-\tau)\, d\tau =$$

$$= \frac{1}{\Delta T} \int_{-\infty}^{\infty} F(\tau) \sum_{n=-\infty}^{\infty} \delta(u - \tau - \frac{n}{\Delta T})\, d\tau = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} F(u - \frac{n}{\Delta T})$$

- **result** is an infinite, periodic sequence of copies of transform of the original continuous function
- separation between copies of $F(u)$ is determined by value $1/\Delta T$

Sampling the signal

- **result** is an infinite, periodic sequence of copies of transform of the original continuous function
- separation between copies of $F(u)$ is determined by value $1/\Delta T$



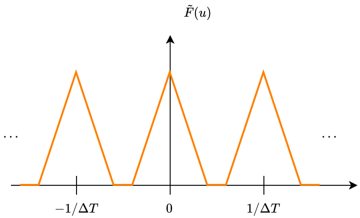
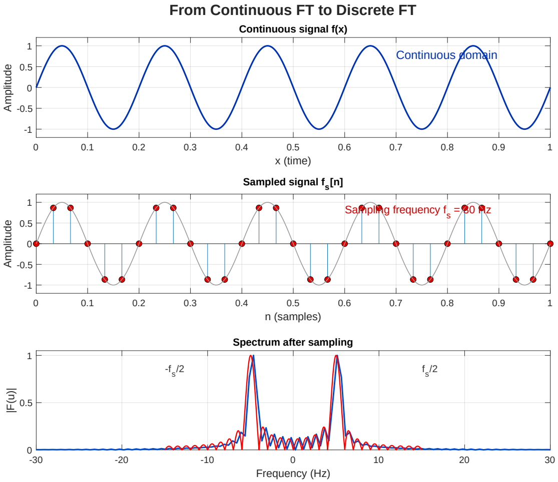
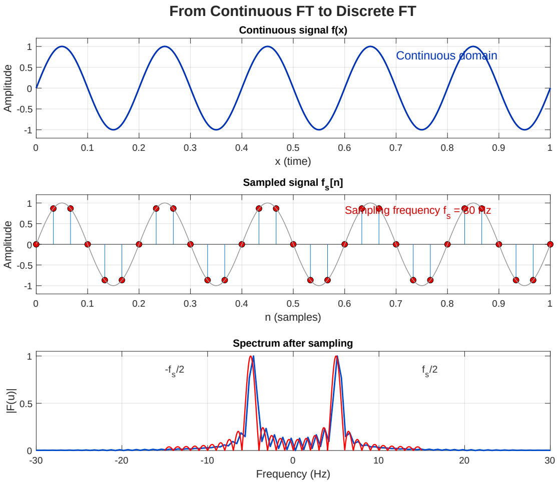


Illustration – From Continuous to Discrete FT



- **Top:** continuous sinusoidal signal $f(x)$
- **Middle:** sampled discrete signal $f_s[n]$
- **Bottom:** periodic frequency spectrum after sampling (blue continuous, red sampled)

Illustration

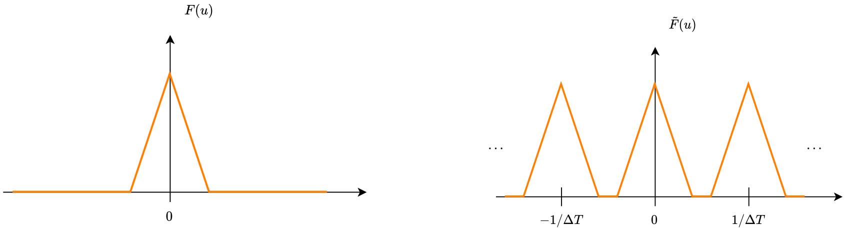


Example

Observe how the spectrum repeats with period f_s.

Aliasing and the Sampling Theorem

- **result** is an infinite, periodic sequence of copies of transform of the original continuous function
- separation between copies of $F(u)$ is determined by value $1/\Delta T$
- $1/\Delta T =$ **sampling rate**

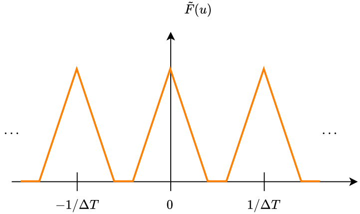


Example

What happens when the sampling rate is reduced?

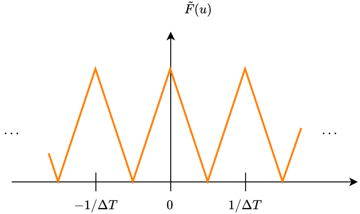
Aliasing and the Sampling Theorem

- when sampling rate is high enough, we obtain good separation for all samples



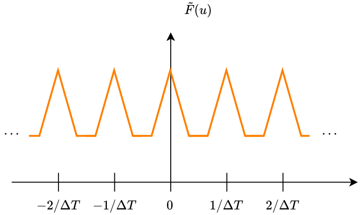
over-sampled signal

- here is sampling rate just enough to preserve integrity = **critically sampling**



Aliasing and the Sampling Theorem

- the sampling rate is below the minimum required to maintain distinct copies of $F(u)$



under-sampling

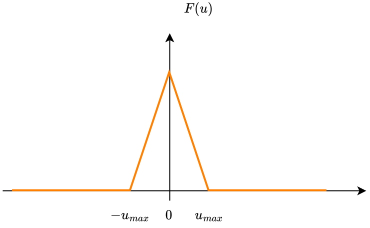
- **Alias** occurs
- alias = "false identity"

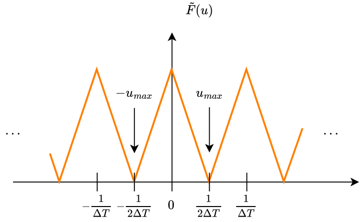
Aliasing and the Sampling Theorem

- **Sampling Theorem (Nyquist–Shannon)**
  - A continuous-time signal can be perfectly reconstructed from its samples if:

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

    where $f_{\max}$ is the highest frequency present in the signal, and f_s is the sampling frequency.
  - The value $f_s/2$ is called the **Nyquist frequency**.





- function $f(t)$ whose Fourier transform is zero for values of frequencies outside a finite interval (band) $[-u_{max}, u_{max}]$ is called a **band-limited function**

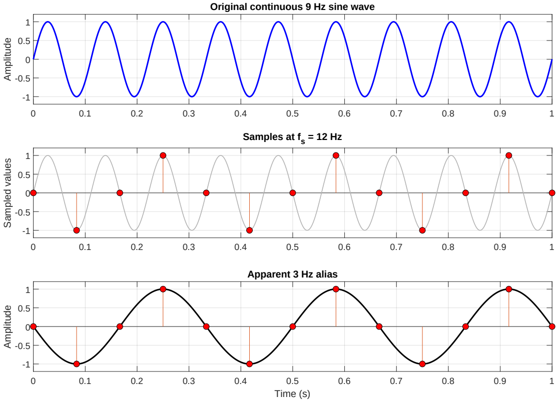
Aliasing – Concept

- **Aliasing** occurs when a signal is sampled below the Nyquist rate.
- High frequencies "fold back" into the baseband, appearing as false low frequencies.
- In frequency domain terms:
  - The spectrum of a sampled signal is periodic with period f_s.
  - When $f_s < 2f_{\mathrm{max}}$, the repeated spectra overlap.

Example

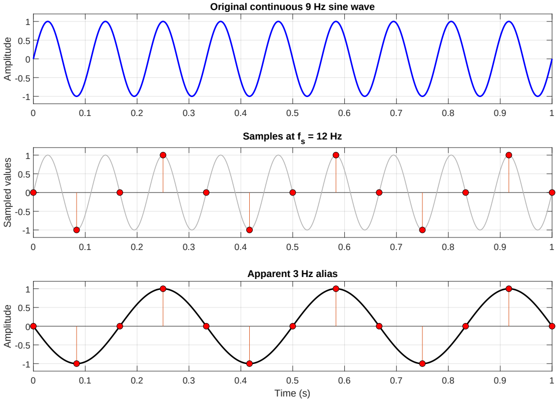
When sampling a sine wave of 9 Hz with $f_s = 12$ Hz, it appears as a 3 Hz signal.

Aliasing in 1D – Example



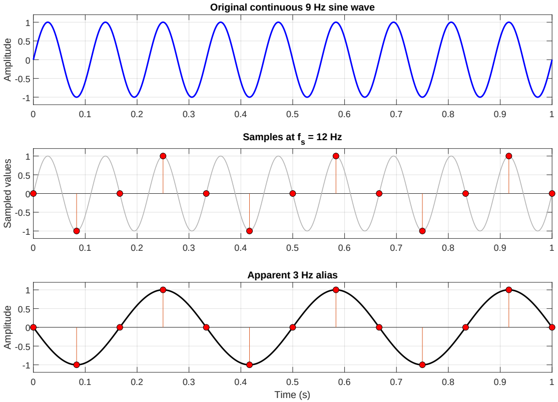
- **Top:** Original continuous 9 Hz sine wave (blue)
- **Middle:** Samples taken at $f_s = 12$ Hz (red dots)
- **Bottom:** Apparent reconstructed wave of 3 Hz (alias)

Aliasing in 1D – Example



- these two different functions coincide at the values of their respective samples
- we do not know, how functions look like between them
- we called this functions **aliased pair**

Aliasing in 1D – Example



Example

Think about how the result would look for other sampling frequencies.

- What happens if the sampling frequency f_s increases/decreases?

Avoiding Aliasing

- To prevent aliasing, we must eliminate high-frequency components before sampling.
- **Anti-aliasing filter:**
  - A **low-pass filter** applied before sampling.
  - Ensures that no frequency higher than $f_s/2$ remains.
- **Trade-off:** Filtering prevents aliasing, but also reduces sharpness and detail.

Summary – Sampling and Aliasing

- Sampling converts a continuous signal into discrete samples.
- The Fourier spectrum of the sampled signal is periodic with period f_s.
- If $f_s < 2f_{\text{max}}$, spectra overlap $\rightarrow$ **aliasing**.
- Nyquist frequency $f_s/2$ defines the highest safely representable frequency.

Discrete Fourier Transform (DFT)

- Sampling is not enough – computers can only handle a **finite number of samples**.
- When we limit the signal to N samples and sample its spectrum too, we obtain the **Discrete Fourier Transform** (DFT):

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

Inverse transform:

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

- **Interpretation:**
  - $f(n)$ – finite, discrete signal
  - $F(u)$ – discrete spectrum (periodic with period N)
  - Basis functions – complex exponentials $e^{-j2\pi kn/N}$

Discrete Fourier Transform (DFT)

- let us have $\tilde{f}(t)$, Fourier transform:

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

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

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

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

- although $f(n)$ is discrete, $\tilde{F}(u)$ is continuous and infinitely periodic with period $1/\Delta T$

Discrete Fourier Transform (DFT)

- although f_n is discrete, $\tilde{F}(u)$ is continuous and infinitely periodic with period $1/\Delta T$
- we need to sample one period
- we want to obtain N equally spaced samples of $\tilde{F}(u)$ over one period from $u = 0$ to $u = 1/\Delta T$
- so we want $u = \frac{n}{N}$, where $n = 0, 1, \ldots, N - 1$
- it concludes to

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

What next?

- **Aliasing** reminded us that the Fourier Transform has strict assumptions:
  - Sampling creates a **periodic spectrum**.
  - If we sample too coarsely, spectral copies overlap.
- Now we face a different, but related problem:
  - In practice, we never analyze an **infinite** or perfectly periodic signal.
  - We always take a **finite segment** (e.g., a short time window).
- Truncating a signal in time is equivalent to multiplying it by a **rectangular window**.
  - This operation causes **spectral leakage**.
  - Even a pure tone produces energy at neighboring frequencies.

Why Windowing?

- **Problem:** The Discrete Fourier Transform (DFT) assumes that the input signal is **periodic** in the analysis window.
- If the signal segment does not exactly contain an integer number of periods:
  - Discontinuities appear at the boundaries.
  - These discontinuities introduce **spectral leakage**.
- **Spectral leakage** means that energy from one frequency "leaks" into neighboring frequencies.
- **Solution:** Multiply the signal by a smooth **window function** that reduces the edges to zero.

Think of windowing as "fading in and out" of the signal before computing the DFT.

Spectral Leakage Illustration

Example

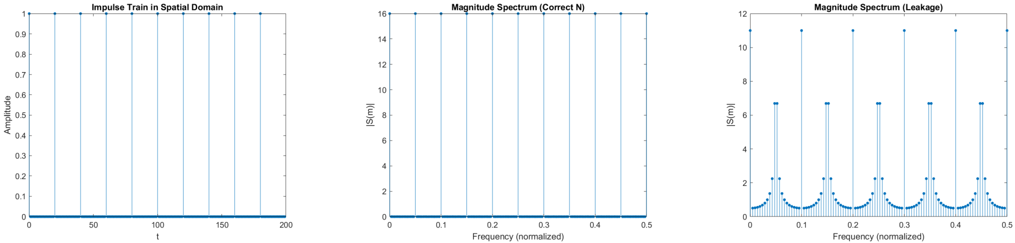
Let us consider the discrete impulse train

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

What should the Fourier transform look like for this signal?

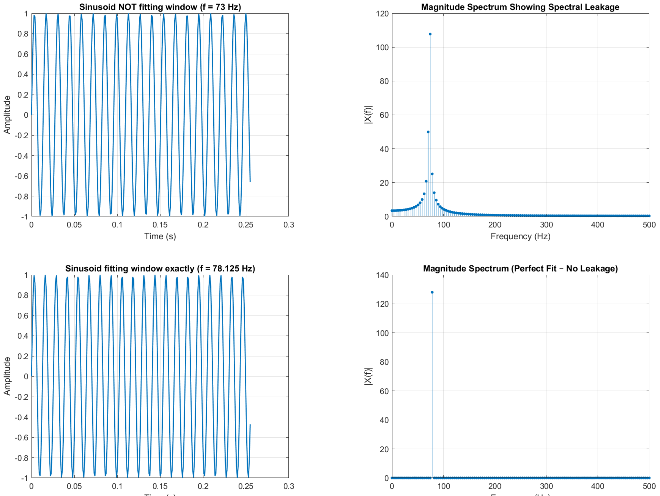
Spectral Leakage Illustration

- Its ideal Fourier transform is again a train of impulses: $S(u) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} \delta(u - \frac{n}{\Delta T})$
- Peaks occur at multiples of $\frac{1}{\Delta T}$.
- **Case 1: correct length** If N is chosen so that $N = k\Delta T$ (an integer number of periods fits), then the spectral impulses fall exactly on DFT frequency bins $\omega = 2\pi m/N$.
  - Result: clean, sharp spectral peaks.
- **Case 2: incorrect length** If N is not a multiple of ΔT, the periodic structure does not fit into the DFT window.
  - The frequency impulses do **not** land on DFT bin centers.
  - Energy spreads into neighboring bins.
  - This produces **spectral leakage**.



Spectral Leakage Illustration

- Consider a pure sinusoid that does **not** fit perfectly into the analysis window.
- When transformed by DFT:
  - Instead of a single peak, we observe multiple neighboring components.
  - This is due to sharp discontinuities at the window borders.

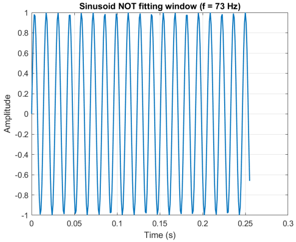


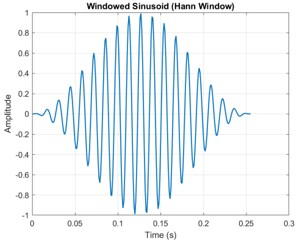
Effect of Windowing on Spectral Leakage

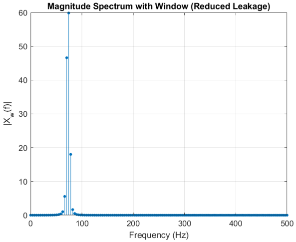
- A sinusoid that does **not** fit into the analysis window introduces discontinuities at the boundaries, causing spectral leakage.
- Multiplying the signal by a **window function** (e.g. Hann window):

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

  smooths the transitions at the edges.
- As a result:
  - leakage is significantly reduced,
  - frequency peak becomes wider but much cleaner,
  - side lobes are suppressed.







Window Functions

- Applying a **window function** reduces these discontinuities and limits leakage.
- **Rectangular window:**

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

  - No attenuation at the edges.
  - Causes strong spectral leakage.
- **Hann (Hanning) window:**

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

- **Hamming window:**

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

- **Blackman window:**

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

Time and Frequency Domain View

- Windowing is a **multiplication** in the time domain:

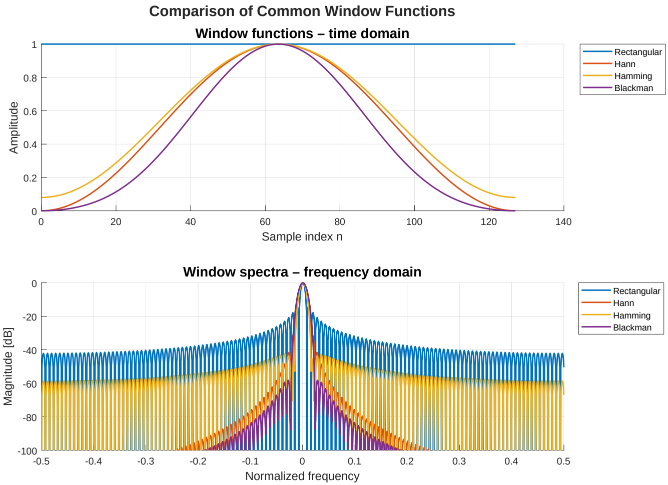
$$x_w(n) = x(n) \cdot w(n)$$

- In the frequency domain, this becomes a **convolution**:

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

- This explains why windowing smooths the spectrum but also broadens the main lobe.
- **Trade-off:**
  - Narrow main lobe $\rightarrow$ good frequency resolution.
  - Low side lobes $\rightarrow$ low leakage.
  - No single window is best for all purposes.

Comparison of Common Windows



- The time-domain shape (top) and magnitude spectrum (bottom)
- Note how the main lobe width increases and side lobes decrease with smoother windows.

Motivation for 2D DFT

- The Fourier Transform is not limited to one-dimensional signals.
- In image processing, we work with 2D data $f(x, y)$:
  - spatial coordinates: x, y
  - intensity values: $f(x, y)$
- The **2D Discrete Fourier Transform (2D DFT)** represents the image in terms of spatial frequencies.
- Just as in 1D, it decomposes the signal into a sum of complex exponentials – now in two directions.

Example

The 2D DFT tells us "how much" of each frequency component exists in the horizontal and vertical directions of the image.

2D Discrete Fourier Transform – Definition

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

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

- The inverse transform:

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

Relation to 1D DFT

- The 2D DFT is simply the 1D DFT applied twice:
  - First, compute DFT for each row.
  - Then, compute DFT for each column of the result.

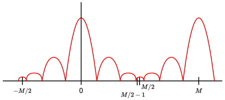
$$F(u, v) = \mathrm{DFT}_x \left(\mathrm{DFT}_y(f(x, y))\right)$$

Properties of the 2D DFT

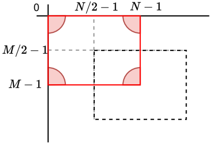
- **Periodicity:** $F(u, v)$ is periodic in both frequency directions.
- **Conjugate symmetry:** For real-valued images,

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

$\Rightarrow$ the magnitude spectrum is symmetric about the origin.



$F(u)$



$F(u, v)$

- **Re-centering the spectrum:** For visualization and frequency-domain filtering it is convenient to shift the periodic spectrum so that the zero frequency lies in the center:

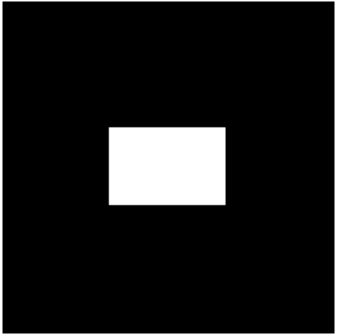
$$F_{\text{center}}(u, v) = F\Big(\big(u + \tfrac{M}{2}\big) \bmod M,\ \big(v + \tfrac{N}{2}\big) \bmod N\Big)$$

Properties of the 2D DFT

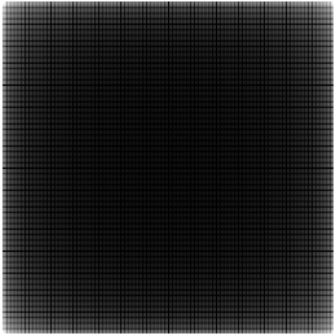
- Shifting the spectrum corresponds to multiplying the spatial image by a sign-alternating factor:

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

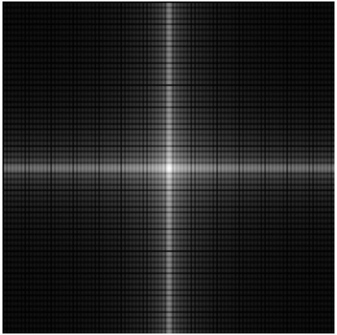
This phase term moves the origin of the DFT to the image center.



Spatial domain



Spectrum



Centered spectrum

Properties of the 2D DFT

- **Linearity:** $\mathrm{DFT}(af_1 + bf_2) = a\,\mathrm{DFT}(f_1) + b\,\mathrm{DFT}(f_2)$
- **Translation property:** A spatial shift causes a phase change in the frequency domain.
- **Scaling property:** Narrow features $\rightarrow$ broad spectra, and vice versa.

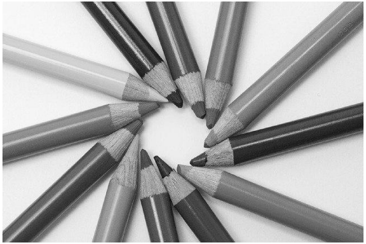
Interpretation of 2D Spectrum

- 2D DFT is complex in general, it can be expressed in polar form $F(u,v) = R(u,v) + jI(u,v) = |F(u,v)|e^{j\phi(u,v)}$
- magnitude $|F(u,v)|$ = **Fourier (frequency) spectrum**
- $\phi(u,v) = arctan\frac{I(u,v)}{R(u,v)}$ = **phase angle** or **phase spectrum**
- $P(u,v) = |F(u,v)|^2$ = **Power spectrum**
- $|F(u,v)|$, $\phi(u,v)$ and $P(u,v)$ are arrays of size $M \times N$
- spectrum has even symmetry about origin – $|F(u,v)| = |F(-u,-v)|$
- phase has odd symmetry about origin – $\phi(u,v) = -\phi(-u,-v)$

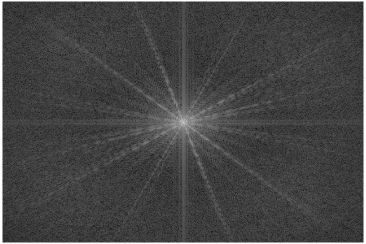
Interpretation of 2D Spectrum

- The **origin (u=0, v=0)** corresponds to the average (DC component) – overall brightness.
- High frequencies represent rapid intensity changes (edges, textures).
- The direction of a frequency component corresponds to orientation in the image:
  - Horizontal structures $\rightarrow$ vertical frequencies.
  - Vertical structures $\rightarrow$ horizontal frequencies.
- **Magnitude spectrum** shows the strength of spatial frequencies.
- **Phase spectrum** carries structural information – necessary to reconstruct the image.

Original image

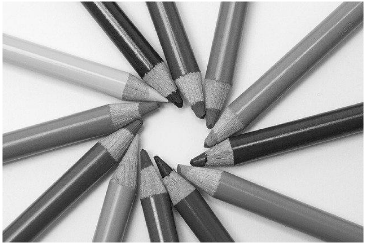


Log magnitude spectrum

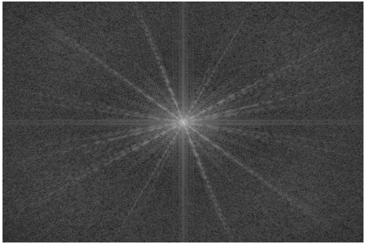


Visualizing the 2D DFT

Original image

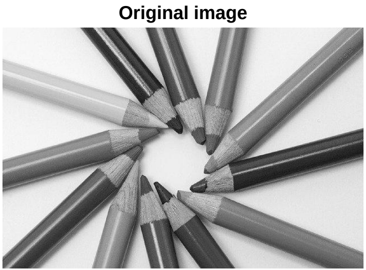


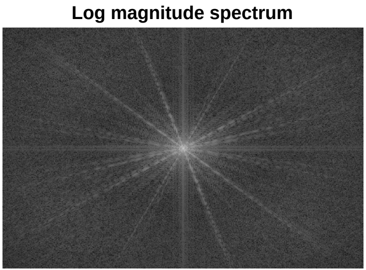
Log magnitude spectrum



- **Interpretation:**
  - The bright center $\rightarrow$ low frequencies (smooth regions).
  - The outer parts $\rightarrow$ high frequencies (edges, details).
  - Diagonal or oriented patterns correspond to lines in the spectrum.
- The logarithmic display helps compress large dynamic ranges.

Visualizing the 2D DFT





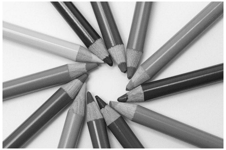
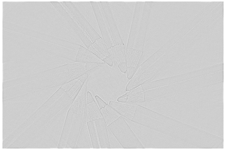
- **Interpretation:**
  - The bright center $\rightarrow$ low frequencies (smooth regions).
  - The outer parts $\rightarrow$ high frequencies (edges, details).
  - Diagonal or oriented patterns correspond to lines in the spectrum.
- The logarithmic display helps compress large dynamic ranges.

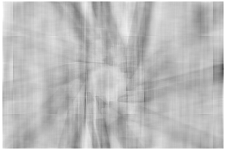
Phase interpretation

- The 2D Fourier transform decomposes an image into:

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

  where – $|F(u, v)|$ is the magnitude (energy distribution), – $\phi(u, v)$ is the phase (structural information).
- The following example illustrates three reconstructions:
  - Original image
  - Phase-only reconstruction – magnitude set to 1, phase kept unchanged
  - Rectangle magnitude + original phase – magnitude replaced by spectrum of a simple rectangle
- The results show that the image remains recognisable even when magnitude is removed or replaced, confirming that phase carries essential visual structure
- However, phase alone does not carry intensity information brightness and contrast come solely from the magnitude spectrum.

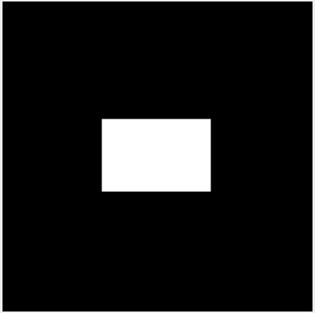



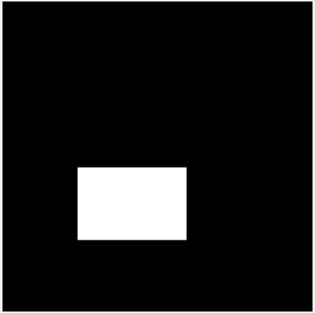


Effect of Translation on Spectrum and Phase

Example

Suppose we have two images, each containing the same rectangle, but one image has the rectangle shifted to a different location.





How will the magnitude spectrum and the phase change?

Translation: Spectrum vs. Phase

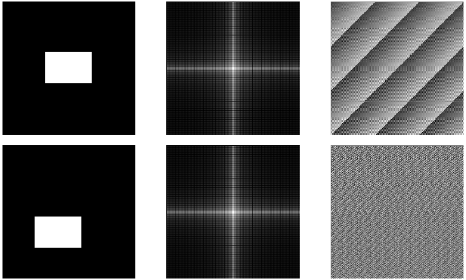
- A spatial shift does not change the magnitude of the Fourier spectrum:

$$|F(u,v)| \text{ remains unchanged.}$$

- But translation introduces a linear phase shift:

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

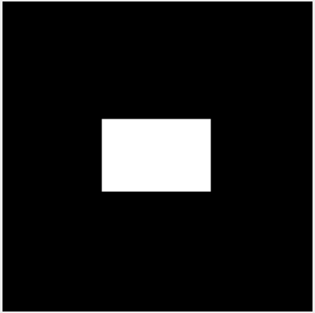
- Therefore:
  - The magnitude spectrum stays the same.
  - The phase changes in a predictable, linear way.

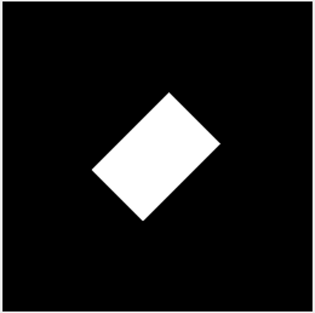


Effect on Magnitude and Phase

Example

Now consider two images, each containing the same rectangle,but in the second image the rectangle is rotated.





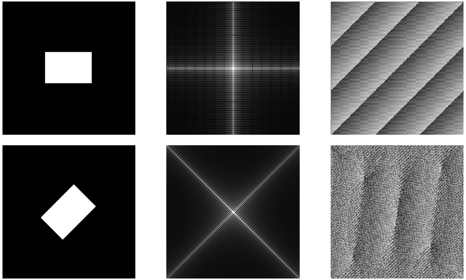
How will the magnitude spectrum and the phase change?

Rotation: Effect on Magnitude and Phase

- Rotation in the spatial domain results in an identical rotation in the Fourier domain:

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

  where R is the rotation matrix.
- Therefore:
  - The shape of the magnitude spectrum is preserved.
  - The spectrum is simply rotated by the same angle.
  - The phase also rotates accordingly.

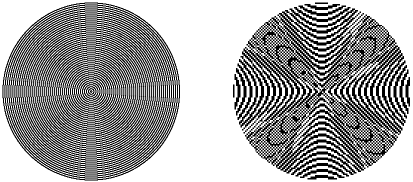


Aliasing in Images

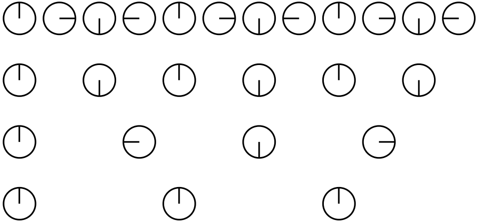
- Aliasing is not only a 1D phenomenon – it also appears in 2D images.
- It occurs when fine spatial details exceed the pixel sampling frequency.
- Two types of alias in images:
  - spatial aliasing – caused by under-sampling (more visible in images with repetitive patterns)
  - temporal aliasing – time intervals between images of a sequence of dynamic images
- **Visual effects:**
  - False repetitive patterns (moiré)
  - "Beating" between fine textures and pixel grid
  - Loss of high-frequency detail
- **Examples:**
  - Photograph of a striped shirt on a digital camera
  - Fine roof tiles or grids producing interference patterns
  - Downscaling an image without proper low-pass filtering

Aliasing in Images

- Aliasing is not only a 1D phenomenon – it also appears in 2D images.
- It occurs when fine spatial details exceed the pixel sampling frequency.
- Two **types of alias** in images:
  - spatial aliasing – caused by under-sampling (more visible in images with repetitive patterns)



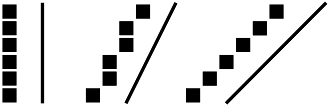
  - temporal aliasing – time intervals between images of a sequence of dynamic images



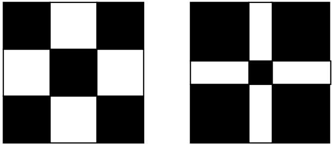
Spatial Alias

- **Moiré** – False repetitive patterns (`https://www.mightool.com/scanimation`)

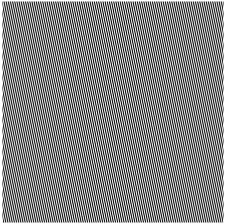
- **Jaggies** – staircase-like edge artifacts



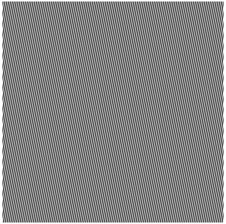
- **Loss of details** – sampling grid cannot represent the original spatial frequencies correctly



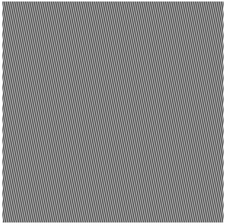
Aliasing in Images



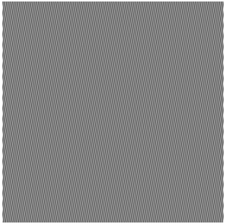
Original size



1024 × 1024



512 × 512



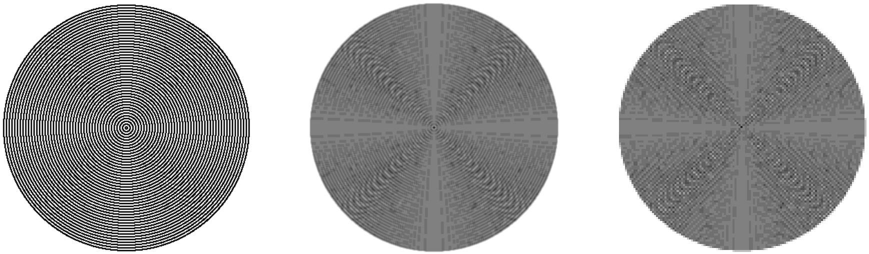
256 × 256

128 × 128

64 × 64

Reducing Aliasing in Images

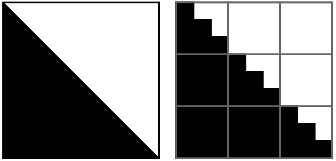
- Aliasing appears when an image contains spatial details with frequencies higher than half the sampling rate (Nyquist limit).
- **Methods to reduce aliasing**:
  - Low-pass filtering before sampling – Removes high-frequency content that cannot be represented.
  - Increasing the sampling resolution – Higher pixel density raises the Nyquist limit.
  - Supersampling / multisampling – Render at a higher resolution and downscale for smoother edges.



Aliasing Reduction – Supersampling

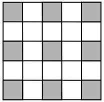
- **supersampling**: increasing the sampling frequency – $n_x \times$ in the x-direction and $n_y \times$ in the y-direction.
- **superpixels**: blocks of $n_x \times n_y$ high-resolution samples.
- Select m samples from each superpixel (if $m = n_x \cdot n_y$, this corresponds to full oversampling).
- Selected samples: $s_i, \quad i \in \{0, \ldots, m-1\}$
- Weights for reconstruction: $w_i, \quad i \in \{0, \ldots, m-1\}$
- **downsampling**: each superpixel is replaced by a single output pixel

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

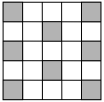


Sample Selection Within a Superpixel

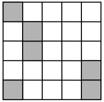
- **Regular sampling**



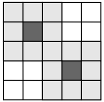
- **Rotated grid**



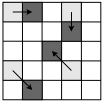
- **Random sampling**



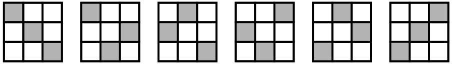
- **Poisson disk**



- **Jittering** – jittered sampling



- **N-rooks algorithm**



Windowing in 2D Fourier Analysis

- Windowing reduces spectral leakage in 1D signals by smoothing the boundaries of the analysis window.
- The same principle applies to 2D images: sharp boundaries in a spatial block produce spurious frequencies.
- A 2D window (Hann, Hamming, Blackman, Gaussian) is defined as

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

and is applied before computing the local 2D DFT.

Global 2D FFT of an Entire Image

- When computing the Fourier transform of the **entire image**, windowing is typically not used.
- The image is treated as one complete 2D discrete signal.
- Applying a window would:
  - artificially darken the borders,
  - distort the spectrum of the image itself.

Local Fourier Analysis and Windowing

- Windowing becomes essential when computing the FFT of a local region or image block.
- Extracting a block (e.g. 32×32) introduces sharp edges at the block boundary.
- These edges behave like discontinuities and cause 2D spectral leakage.
- Applying a 2D window suppresses boundary discontinuities and reduces leakage.

Example: Block-Based FFT

- Let $f(x, y)$ be an image and $B(x, y)$ a rectangular block cut from it.
- The block has discontinuities at its edges:

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

  where r is a rectangular window.
- In the frequency domain:

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

  convolution with the sinc-like transform of the rectangle $\Rightarrow$ strong leakage.
- Using a 2D window $w(x, y)$:

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

  smooths the boundaries and reduces spectral spread.

Windowing in Texture and Frequency Analysis

- Local frequency analysis (e.g. Gabor filters, STFT for images) inherently uses window functions.
- A Gabor filter is the product of:

$$\text{Gaussian window} \times \text{complex sinusoid.}$$

- The Gaussian window ensures:
  - spatial localization,
  - controlled bandwidth,
  - minimal leakage.
- Windowing is thus fundamental for analyzing textures, orientations, and spatial-frequency patterns.

Practical Use in Image Processing

- In 2D image analysis, windowing is applied before performing the 2D DFT.
- Purpose:
  - Reduce edge discontinuities in image blocks.
  - Improve spectral estimation for local features.
- Commonly used in:
  - Block-based spectral analysis (e.g., texture descriptors)
  - Frequency-domain filtering of non-periodic images
  - Preprocessing before FFT to reduce ringing artifacts

Applications of 2D DFT in Image Processing

- **Filtering:**
  - Low-pass filtering $\rightarrow$ smoothing, noise reduction.
  - High-pass filtering $\rightarrow$ edge enhancement.
- **Pattern and texture analysis:**
  - Periodic textures show peaks in the spectrum.
  - Orientation of peaks corresponds to texture direction.
- **Image restoration:**
  - Removing periodic noise in the frequency domain.
  - Deblurring by inverse filtering.

Motivation for the Next Lecture – Fast Fourier Transform (FFT)

- In the next lecture, we will see how mathematical structure can turn millions of operations into thousands — without changing the result.
- The Discrete Fourier Transform (DFT) is fundamental, but **computationally expensive**.
  - For an $N \times N$ image, direct computation requires $O(N^4)$ operations.
  - Even for 1D signals, DFT scales as $O(N^2)$.
- To make Fourier analysis practical, we need a faster algorithm that:
  - gives **exactly the same result** as DFT,
  - but reduces computation to $O(N \log N)$.

- This algorithm is called the **Fast Fourier Transform (FFT)** — one of the most important numerical algorithms ever developed.