Fourier Transform
Lecture 3: 2D Image Transforms

Image Processing Course

November 10, 2025

Contents

1 The Core Idea: Image vs. Frequency Domain

This lecture introduces the 2D Fourier Transform. The main idea is that an image can be represented in two different "bases" or "domains."

- **Image (Spatial) Domain:** The standard image, $f(x, y)$. Each value describes the intensity at a **local** (x, y) pixel coordinate.
- **Frequency Domain:** The Fourier image, $F(u, v)$. Each value describes the amplitude and phase of a **global** 2D sine/cosine wave that makes up the entire image.

Key Concept 1.1 (Change of Basis)**:** *The **2D Fourier Transform** is the mathematical tool that changes the basis of an image from its pixel representation ($f(x, y)$) to its frequency representation ($F(u, v)$). It provides the "recipe" of 2D waves needed to build the image.*

2 The 2D Discrete Fourier Transform (DFT)

For a digital, $N \times N$ image, we use the 2D DFT.

2.1 The Formulas

- **Forward DFT (Analysis):** $f(x, y) \to F(u, v)$
  Analyzes the image to find its frequency recipe.

$$F(u, v) = \frac{1}{N} \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} f(x, y) e^{\frac{-2\pi i(ux+vy)}{N}} \tag{1}$$

- **Inverse DFT (Synthesis):** $F(u, v) \to f(x, y)$
  Reconstructs the image from its frequency recipe.

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) e^{\frac{2\pi i(ux+vy)}{N}} \tag{2}$$

Key Concept 2.1 (The DC Component: F(0,0))**:** *The coefficient $F(0, 0)$ (where $u = 0, v = 0$) has a special meaning. The e^0 term becomes 1, leaving:*

$$F(0, 0) = \frac{1}{N} \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} f(x, y)$$

*This is N times the **average intensity** of the entire image ($F(0, 0) = N \cdot \overline{f}$). For real images (where intensities are positive), this is almost always the brightest, highest-value point in the entire spectrum.*

3 Visualizing the Fourier Spectrum

The raw Fourier coefficients $F(u, v)$ are complex numbers. To visualize them, we usually plot the **Spectrum** (or Magnitude).

Definition 3.1 (Spectrum and Phase)**:** *A complex coefficient $F(u) = R(u) + iI(u)$ can be represented in polar form:*

- ***Fourier Spectrum (Magnitude):*** $|F(u)| = \sqrt{R^2(u) + I^2(u)}$*. This is the "strength" of the frequency.*
- ***Fourier Phase (Angle):*** $\theta(u) = \tan^{-1}(I(u)/R(u))$*. This is the "shift" of the wave.*

3.1 Displaying the Spectrum

The raw spectrum $|F(u, v)|$ is hard to view because the $F(0, 0)$ value is so large it makes everything else look black. We follow a 3-step process to display it:

1. **Log Transform:** Compute $\log(1 + |F(u, v)|)$. This compresses the huge range of values, making dimmer frequencies visible.
2. **Scale:** Scale the log-transformed values to the display range (e.g., 0-255).
3. **Shift:** Shift the $F(0, 0)$ point from the top-left corner to the center of the image. This gives an intuitive view with low frequencies in the center and high frequencies at the edges.

4 Properties of the 2D DFT

- **Linearity:** The transform of a sum of images is the sum of their transforms.
- **Separability:** A 2D FFT can be computed by first running a 1D FFT on all the columns, then running a 1D FFT on all the rows of the result.
- **Symmetry:** For real images, the spectrum is symmetric: $|F(u, v)| = |F(-u, -v)|$.
- **Translation:** If an image is shifted, $f(x - x_0, y - y_0)$, its spectrum $|F(u, v)|$ **does not change**. Only the phase is altered.

Key Concept 4.1 (Phase vs. Magnitude)**:** *The **Phase** contains the essential structural information (the "what" and "where" of objects). The **Magnitude** (Spectrum) contains the "style" or "energy" of the frequencies. If you combine the Phase of a building with the Magnitude of a Godzilla, the resulting image will look like the building.*

5 Applications of the 2D DFT

5.1 Calculating Derivatives

Taking a derivative in the spatial domain is equivalent to a simple multiplication in the frequency domain. To find the partial derivative with respect to x:

1. Compute $F(u, v)$.
2. Multiply: $F_{x-deriv}(u, v) = F(u, v) \cdot \left(\frac{2\pi i}{N} u\right)$.
3. Compute the Inverse Transform of $F_{x-deriv}(u, v)$.

Note: *This operation is a **high-pass filter**. It amplifies high frequencies (large u) and cancels the DC component (u = 0). Because noise is primarily high-frequency, **derivatives dramatically amplify noise**.*

5.2 Frequency Domain Filtering

You can filter an image by multiplying its spectrum $F(u, v)$ by a filter mask $H(u, v)$ and then taking the inverse transform.

- **Low-Pass Filter (LPF):** $H(u, v)$ keeps the center frequencies and blocks the outer ones. This **blurs** the image.
- **High-Pass Filter (HPF):** $H(u, v)$ blocks the center and keeps the outer ones. This detects **edges**.

Note: *A sharp "Box Filter" (a hard-edged square) will create "ringing" artifacts. A smooth "Gaussian Filter" (a soft, bell-shaped blur) is preferred because its transform is also smooth and does not cause ringing.*

6 Sampling and Aliasing

Definition 6.1 (Aliasing)**:** *Aliasing occurs when you "re-sample" (shrink) an image incorrectly. It creates **new, false patterns** (like Moiré patterns) that were not in the original.*

The **Nyquist-Shannon Sampling Theorem** states that to avoid aliasing, your sampling distance must be **at most half the wavelength** of the highest frequency (smallest detail) in the image.

Example 6.1 (Breaking Nyquist)**:** *If an image is "black, white, black, white..." (a 2-pixel wavelength), the Nyquist limit is half of that, or 1 pixel. If you "sample every 2nd pixel" (a distance of 2), you break the rule and will see a solid black or solid white (aliased) result.*

Key Concept 6.1 (The Golden Rule of Sampling)**:** *This leads to the most important rule for resizing and sampling signals:*

BLUR BEFORE YOU SAMPLE

*To shrink an image correctly, you must first **blur it (low-pass filter)** to remove the high frequencies that would cause aliasing, then you can safely throw away the extra pixels.*

7 Image Resizing with the Fourier Transform

The DFT provides an "ideal" method for resizing, as it allows for perfect frequency filtering.

7.1 Image Reduction (Shrinking)

1. Compute the FFT of the $N \times N$ image and shift the center.
2. **Crop** the central $M \times M$ region of the spectrum (where $M < N$). This is a perfect low-pass filter.
3. Compute the Inverse FFT on this new, smaller $M \times M$ spectrum. The result is a perfectly anti-aliased $M \times M$ image.

7.2 Image Expansion (Enlarging)

1. Compute the FFT of the $N \times N$ image and shift the center.
2. **Pad** the spectrum with zeros to create a larger $M \times M$ spectrum (where $M > N$).
3. Compute the Inverse FFT on this new, larger $M \times M$ spectrum.