Introduction to Computer Vision

Appendix for lab 2: Frequency Domain Filtering & Hybrid Images

ESIN, UIR — Spring 2026

Ilias TOUGUI

Contents

1 The Convolution Theorem

1.1 2D Convolution: A Reminder

Given an image $f(x, y)$ and a filter kernel $h(x, y)$, their **2D convolution** is:

$$(f * h)(x, y) = \sum_m \sum_n f(m,\, n)\; h(x - m,\; y - n) \tag{1}$$

For an image of size $N \times N$ and a kernel of size $K \times K$, computing equation (1) directly requires $\mathcal{O}(N^2 K^2)$ multiplications. For a 1000×1000 image and a 101×101 kernel, that is over **10 billion operations**.

1.2 The 2D Fourier Transform

The **2D Discrete Fourier Transform (DFT)** of an image $f(x, y)$ of size $M \times N$ is:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)\; e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \tag{2}$$

where u and v are the **spatial frequencies** in the horizontal and vertical directions respectively. The inverse transform recovers $f(x, y)$:

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v)\; e^{j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \tag{3}$$

The **Fast Fourier Transform (FFT)** computes equation (2) in only $\mathcal{O}(N^2 \log N)$ operations — a dramatic improvement over the naive $\mathcal{O}(N^4)$ direct computation.

1.3 The Convolution Theorem

Convolution Theorem (2D)

Let $f(x, y)$ and $h(x, y)$ be two 2D signals with Fourier transforms $F(u, v)$ and $H(u, v)$ respectively. Then:

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

That is, **convolution in the spatial domain is equivalent to pointwise multiplication in the frequency domain**.

This means that instead of sliding the kernel over every pixel, we can:

1. Compute $F = \mathcal{F}\{f\}$ and $H = \mathcal{F}\{h\}$ (two FFTs)
2. Multiply pointwise: $G = F \cdot H$ ($\mathcal{O}(N^2)$)
3. Compute $g = \mathcal{F}^{-1}\{G\}$ (one inverse FFT)

The total cost is $\mathcal{O}(N^2 \log N)$, **independent of the kernel size** K.

Computational Comparison

Method	**Complexity**	**Depends on kernel size?**
Spatial convolution	$\mathcal{O}(N^2K^2)$	Yes
FFT-based convolution	$\mathcal{O}(N^2 \log N)$	No

For large kernels ($K \gg 1$), FFT convolution is significantly faster.

2 Frequency Domain Filtering

2.1 Reading the Magnitude Spectrum

The magnitude spectrum $|F(u, v)|$ tells us which frequencies are present in an image and how strong they are. After shifting the zero-frequency DC component to the center, we can interpret it as follows:

- **Center** (low frequencies): slow spatial variations — smooth regions, overall shapes, background.
- **Outer edges** (high frequencies): rapid spatial variations — sharp edges, fine textures, noise.

Most of the energy in natural images is concentrated in the low frequencies (the center of the spectrum), which is why a heavily blurred image still looks recognizable.

2.2 Ideal Filters

2.2.1 Ideal Low-Pass Filter (ILPF)

An Ideal Low-Pass Filter keeps all frequencies within a circle of radius r_c (the *cutoff frequency*) and completely zeroes out everything outside:

$$H_{\text{LP}}(u, v) = \begin{cases} 1 & \text{if } \sqrt{u^2 + v^2} \leq r_c \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Effect: Removing high frequencies blurs the image — only the smooth, slowly-varying parts survive.

2.2.2 Ideal High-Pass Filter (IHPF)

An Ideal High-Pass Filter is the complement of the ILPF:

$$H_{\mathrm{HP}}(u,v) = 1 - H_{\mathrm{LP}}(u,v) = \begin{cases} 0 & \text{if } \sqrt{u^2+v^2} \le r_c \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

Effect: Removing low frequencies leaves only sharp edges and fine texture details.

2.2.3 The Gibbs Phenomenon

Ringing Artifacts

The ideal circular mask creates a **sharp, abrupt cut** in the frequency domain. This hard edge is mathematically equivalent to a sinc-like function in the spatial domain — one that oscillates and never fully dies out. When convolved with the image, these oscillations appear as visible **ripple rings around sharp edges**, known as the **Gibbs Phenomenon**.

Think of it like hitting a table sharply: the surface vibrates before settling. A Gaussian filter is a gentle press — no sudden shock, no ringing.

Filter	Frequency domain	Spatial domain
Ideal LP/HP	Sharp circular edge	Oscillating sinc → **ringing**
Gaussian LP/HP	Smooth Gaussian taper	Gaussian → **no ringing**

The key takeaway: any filter with a *sharp, abrupt boundary* in the frequency domain will produce ringing in the spatial domain. The Gaussian avoids this entirely because it has *no sharp boundary* — it fades out gradually in both domains.

2.3 Gaussian Filters

The solution to ringing is to use a filter that **gradually tapers** to zero instead of cutting off abruptly. The most natural choice is the **Gaussian filter**.

2.3.1 Gaussian Low-Pass Filter

A 2D Gaussian kernel with standard deviation σ is:

$$h_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{6}$$

Its Fourier Transform is also a Gaussian:

$$H_\sigma(u, v) = \exp\left(-\frac{2\pi^2\sigma^2(u^2 + v^2)}{N^2}\right) \tag{7}$$

The Gaussian is Self-Dual

The Fourier Transform of a Gaussian is a Gaussian. This unique property means the filter is **smooth in both domains simultaneously** — no sharp transitions, no ringing artifacts.

2.3.2 Gaussian High-Pass Filter

A Gaussian high-pass filter is defined as:

$$\mathrm{HP}(f) = f - \mathrm{LP}_\sigma(f) \tag{8}$$

That is, we subtract the blurred version from the original. This naturally leaves only the high-frequency components (edges and textures).

2.3.3 The Uncertainty Principle

There is a fundamental trade-off between spatial and frequency resolution:

Heisenberg Uncertainty Principle (Signal Processing)

A signal cannot be simultaneously compact in both the spatial and frequency domains. Formally:

$$\Delta x \cdot \Delta u \;\geq\; \frac{1}{4\pi}$$

For Gaussian filters this means:

- **Large** σ: wide kernel in space $\rightarrow$ narrow Gaussian in frequency $\rightarrow$ **more blur** (fewer frequencies pass)
- **Small** σ: narrow kernel in space $\rightarrow$ wide Gaussian in frequency $\rightarrow$ **less blur** (more frequencies pass)

3 Hybrid Images

3.1 Concept

Hybrid images were introduced by Oliva, Torralba, and Schyns in their **SIGGRAPH 2006** paper *"Hybrid Images"*. The key observation is that human visual perception behaves like a low-pass filter at large viewing distances: fine details (high frequencies) are lost, and only the overall shape (low frequencies) remains visible.

Hybrid Image

A hybrid image H is formed by combining:

- The **low-pass** filtered version of image A (perceived from far)
- The **high-pass** filtered version of image B (perceived up close)

$$H = \mathrm{LP}_{\sigma_1}(A) + \alpha \cdot \mathrm{HP}_{\sigma_2}(B)$$

where $\alpha \in [0.5, 1.5]$ controls the blending weight of the high-frequency component.

3.2 Why it Works: Human Visual Perception

When viewing a hybrid image:

- **Up close**: the visual system resolves fine details. The high-frequency component of B dominates — you see image B.
- **From far away**: the visual system acts as a low-pass filter, averaging out fine details. Only the low-frequency component of A survives — you see image A.

3.3 Requirements for a Good Hybrid Image

For the illusion to work well, the two images must satisfy:

1. **Spatial alignment**: Key features (eyes, nose, dominant shapes) must be roughly aligned between the two images. Misalignment breaks the perceptual grouping.

2. **Similar framing**: Both images should have the same general composition (e.g., both are close-up portraits, or both are full-body shots). A face and a landscape rarely combine well.

3. **Complementary frequency content**: Image A (low-pass) should be clearly recognizable even when heavily blurred. Image B (high-pass) should have strong, distinctive high-frequency content (edges, textures).

4. **Similar luminance**: The two images should have comparable brightness and contrast. If one image is much darker, it will be overpowered by the other. Histogram matching can correct this.

3.4 Parameter Selection

The quality of the hybrid depends critically on three parameters:

Parameter	Effect	Recommendation
σ_{low} (far image blur)	Larger $\rightarrow$ more aggressive blur	$\sigma \in [5, 15]$
σ_{high} (close image edges)	Smaller $\rightarrow$ sharper edges	$\sigma \in [2, 8]$
α (HP weight)	Higher $\rightarrow$ stronger close image	$\alpha \in [0.7, 1.2]$

Quick Test for a Good Image Pair

Before combining two images, apply a strong blur ($\sigma = 15$) to both:

- Can you still recognize image A after blurring? $\rightarrow$ **Good low-freq source.**
- Does image B become unrecognizable after blurring? $\rightarrow$ **Good high-freq source.**

If both conditions hold, the pair will produce a convincing hybrid.

References

1. Oliva, A., Torralba, A., & Schyns, P.G. (2006). *Hybrid Images.* ACM SIGGRAPH 2006.

2. Gonzalez, R.C., & Woods, R.E. (2018). *Digital Image Processing* (4th ed.). Pearson. Chapters 4 (Frequency Domain) & 5 (Filtering).

3. Oppenheim, A.V., & Schafer, R.W. (2010). *Discrete-Time Signal Processing* (3rd ed.). Pearson.

4. NumPy Documentation: `numpy.fft.fft2`
   https://numpy.org/doc/stable/reference/routines.fft.html

5. SciPy Documentation: `scipy.signal.fftconvolve`
   https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.fftconvolve.html