Central Topic

Filter

Purpose of Image Filtering

Used for denoising images, handling low-light photography, dead pixels, and interference.

Variations of filter: one example

Box Filter: The Simplest Average Filter

$$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad F(di, dj) = \frac{1}{(2r+1)^2}$$

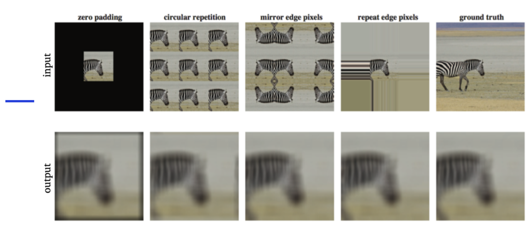
$$J(i,j) = \sum_{di=-r}^{r} \sum_{dj=-r}^{r} F(di,dj) \cdot I(i+di, j+dj)$$

filter radius $\quad r = 1 \Rightarrow 3 \times 3$ box filter

filter weight $\quad F(di,dj) = \frac{1}{9}$

Ways to Handle boundaries

Handling Boundaries



Convolution, Cross-, Auto- Correlation

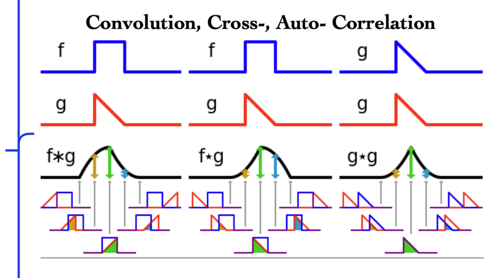
Convolution vs. Cross-Correlation

Convolution $\quad x(t) * h(t) = \int_{-\infty}^{+\infty} x(\tau) \cdot h(t-\tau) d\tau$

Cross-Correlation $\quad x(t) \star h(t) = \int_{-\infty}^{+\infty} x(\tau) \cdot h(t+\tau) d\tau$

Auto-Correlation $\quad x(t) \star x(t) = \int_{-\infty}^{+\infty} x(\tau) \cdot x(t+\tau) d\tau$

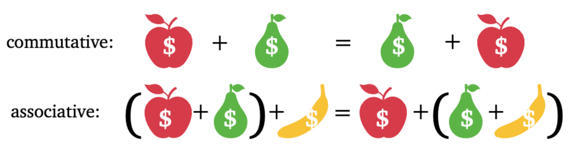
Convolution, Cross-, Auto- Correlation



Properties of convolution and correlation

Convolution vs. Correlation: Key Differences

property	$f * g = f \star g(-t)$	convolution $*$	correlation $\star$
commutative	$f \cdot g = g \cdot f$	✔	✘
associative	$(f \cdot g) \cdot h = f \cdot (g \cdot h)$	✔	✘



Linear Shift-Invariant System (LSI)

Definition: A system that is both Linear and Shift-Invariant

Linear

1. Linearity

A system F is **linear** if it satisfies the following two conditions for any two input signals $x_1(t)$ and $x_2(t)$, and any two scalar constants a_1, a_2:

$$F(a_1 x_1(t) + a_2 x_2(t)) = a_1 F(x_1(t)) + a_2 F(x_2(t))$$

This means that:

- **Scaling Property**: If you scale the input by a, the output is scaled by a as well.
- **Additivity (Superposition Property):** The response to a sum of inputs is the sum of the responses to each input separately.

🔷 **Example**:

- The system $F(x) = 2x$ is linear because:

$$F(a_1x_1 + a_2x_2) = 2(a_1x_1 + a_2x_2) = a_1(2x_1) + a_2(2x_2) = a_1F(x_1) + a_2F(x_2)$$

- The system $F(x) = x^2$ is **not linear** because:

$$F(a_1x_1 + a_2x_2) = (a_1x_1 + a_2x_2)^2 \neq a_1x_1^2 + a_2x_2^2$$

Shift-Invariant

2. Shift-Invariance

A system F is **shift-invariant** if shifting the input by t_0 results in shifting the output by t_0, without changing its form:

$$F(x(t-t_0)) = y(t-t_0)$$

for all t_0.

This means that:

- The system does **not change** its behavior over time (or space in image processing).
- If an input $x(t)$ is delayed, the output is delayed by the same amount.

🔷 **Example**:

- The system $F(x) = x(t) + 5$ **is shift-invariant** because:

$$F(x(t-t_0)) = x(t-t_0) + 5 = y(t-t_0)$$

- The system $F(x) = x(2t)$ **is not shift-invariant** because shifting $x(t)$ by t_0 results in:

$$F(x(t-t_0)) = x(2(t-t_0)) = x(2t - 2t_0)$$

which is **not just a shift of $F(x)$, but also a scaling in time.**

LSI can be write a a convolution with its Impulse Response Function (One dimensional filtering can be represented as convolution !)

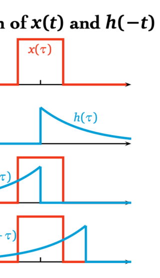
Convolution: A Sliding Product Sum of $x(t)$ and $h(-t)$

$$y(t) = x(t) * h(t) = \int_{-\infty}^{+\infty} x(\tau) \cdot h(t-\tau) d\tau$$

Let's look at two specific values of $y(t)$. Each is a product sum.

$$y(0) = \int_{-\infty}^{+\infty} x(\tau) \cdot h(-\tau) d\tau$$

$$y(1) = \int_{-\infty}^{+\infty} x(\tau) \cdot h(1-\tau) d\tau$$



Here, $h(t)$ is the **impulse response function**, which describes how the system responds to a single impulse input (a Dirac delta function $\delta(t)$). The convolution integral:

$$y(t) = x(t) * h(t) = \int_{-\infty}^{+\infty} x(\tau) \cdot h(t-\tau) d\tau$$

computes the weighted sum of $x(\tau)$ scaled by the shifted impulse response $h(t-\tau)$. This operation characterizes how an input $x(t)$ is transformed by the system.

Explanation of the picture above

Transition from Filtering to mathematical operation

We can write Difference Filtering as A Toeplitz Transformation (Two dimensional filtering can be represented as matrix-vector multiplications !)

Difference Filtering as A Toeplitz Transformation

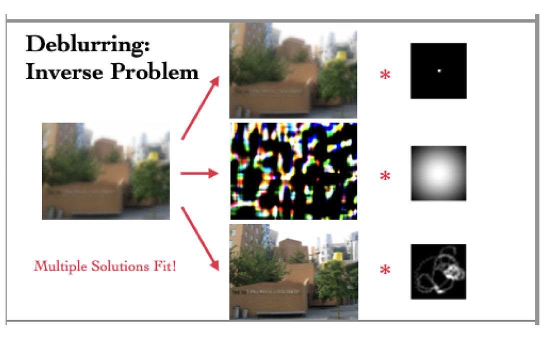
$$\mathbf{I}_x = \frac{\partial \mathbf{I}}{\partial x} = \mathbf{I} * F \qquad F = [1 \quad -1]$$

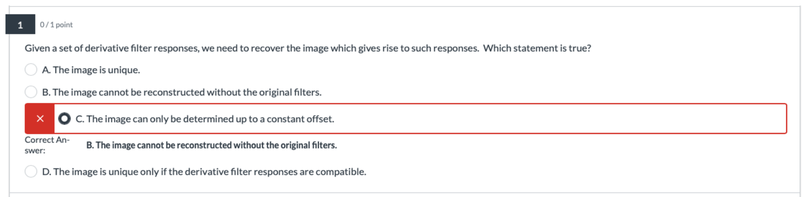
$$\begin{bmatrix} \vdots \\ \mathbf{I}_x[t] \\ \mathbf{I}_x[t+1] \\ \vdots \end{bmatrix} = \begin{bmatrix} & & & \vdots & & & \\ \cdots & 0 & -1 & 1 & 0 & \cdots & \\ & \cdots & 0 & -1 & 1 & 0 & \cdots \\ & & & \vdots & & & \end{bmatrix} \begin{bmatrix} \vdots \\ \mathbf{I}[t-1] \\ \mathbf{I}[t] \\ \mathbf{I}[t+1] \\ \vdots \end{bmatrix}$$

$$\vec{\mathbf{I}}_x = \mathrm{Toeplitz}(F) \cdot \vec{\mathbf{I}}$$

Pyramid

Blurring Problem

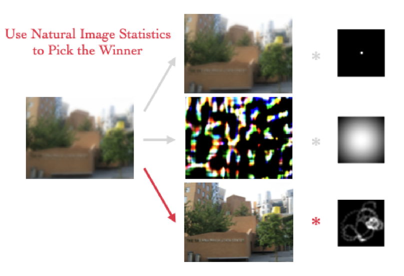




MCQ Question (got me):The image cannot be reconstructed without the original filters.

Natural Imgae

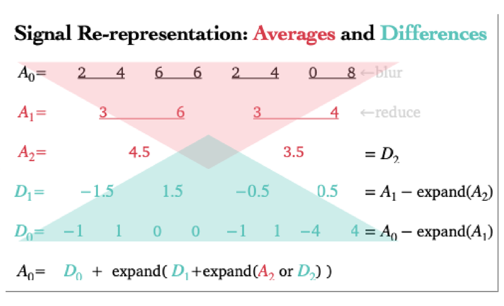
Application: Using natural image statistics to recover sharp images from blurred ones.



Observation: The distribution of pixel intensities follows a heavy-tailed distribution.

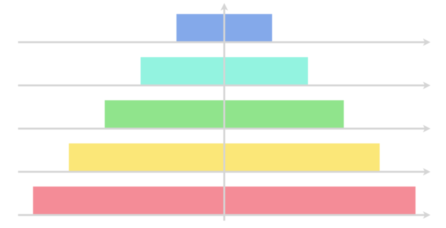
Pyramid Representations: Gaussian and Laplacian Pyramids

Signal Re-representation: Averages and Differences



Guassian Pyramid and Laplacian Pyramid

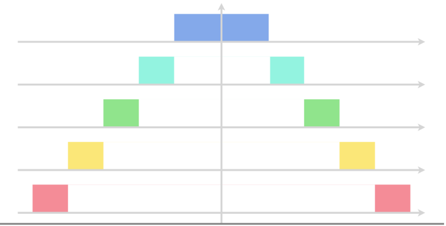
Gaussian Pyramid: Low-Pass Frequency Composition



Gaussian Filter is a low-pass filtering.

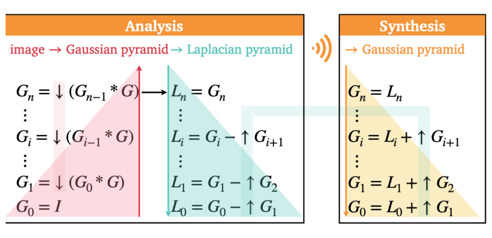
✅ Key Idea: High-frequency components (sharp details) are smoothed out, leaving a "blurred" image.

Laplacian Pyramid: Band-Pass Frequency Composition



A Laplacian pyramid is designed to capture band-pass frequency information

✅ Key Idea: A Laplacian pyramid preserves edges and fine details while eliminating large-scale smooth regions.



Frequency

Fourier Transformation: decomposes an image into frequency components

- Fourier transform: signal analysis into frequency components

$$F(\omega) \Leftarrow \int_{-\infty}^{+\infty} S(x)\, e^{-j\omega x}\, dx$$

- Inverse Fourier: signal synthesis from frequency components

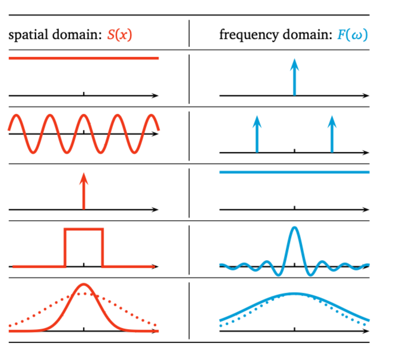
$$S(x) \Leftarrow \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega)\, e^{j\omega x}\, dx$$

- Complex exponentials $\{e^{-j\omega x} : \omega\}$ are orthogonal bases (inverse: $\{e^{j\omega x} : \omega\}$)

$\{\ldots, \cos(\omega x) - j\sin(\omega x), \ldots, 1, \ldots, \cos(\omega x) + j\sin(\omega x), \ldots\}$

Orthogonal Basis of fourier transformation

Examples



spatial domain: $S(x)$	frequency domain: $F(\omega)$
$S(x) \Leftarrow \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega) e^{j\omega x} d\omega$	$F(\omega) \Leftarrow \int_{-\infty}^{+\infty} S(x) e^{-j\omega x} dx$
$S(x) = 1$	$F(\omega) = 2\pi\delta(\omega)$
$S(x) = e^{j\omega_0 x}$	$F(\omega) = 2\pi\delta(\omega - \omega_0)$
$S(x) = \delta(x)$	$F(\omega) = 1$
$S(x) = \begin{cases} 1, & \lvert x\rvert \le \frac{T}{2} \\ 0, & \text{otherwise} \end{cases}$	$F(\omega) = T\,\mathrm{sinc}(\frac{T\omega}{2})$
$S(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$	$F(\omega) = e^{-\frac{\omega^2}{2(\sigma^{-1})^2}}$

Properties

Fourier Transform Properties

$$S(x) \Leftrightarrow F(\omega)$$

Linear addition: $\quad a_1S_1(x) + a_2S_2(x) \Leftrightarrow a_1F_1(\omega) + a_2F_2(\omega)$

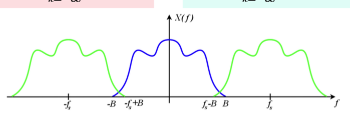
Scaling: $\quad S(a \cdot x) \Leftrightarrow \frac{1}{a} \cdot F(\frac{1}{a} \cdot \omega)$

Spatial shift $\Leftrightarrow$ linear phase shift: $\quad S(x - a) \Leftrightarrow e^{j\omega a} F(\omega)$

Spatial Sampling = Periodic Frequency Repetition

$$\underbrace{S_1(x) \times S_2(x)}_{\text{space multiplication}} \Leftrightarrow \underbrace{F_1(\omega) * F_2(\omega)}_{\text{frequency convolution}}$$

$$x_s(n) = x(n) \cdot \sum_{k=-\infty}^{\infty} \delta(n - kT) \qquad X_s(f) = \sum_{k=-\infty}^{\infty} X(f - kf_s)$$



Convolution Theorem

Convolution Theorem

$$\underbrace{S_1(x) * S_2(x)}_{\text{space convolution}} \Leftrightarrow \underbrace{F_1(\omega) \times F_2(\omega)}_{\text{frequency multiplication}}$$

$$\underbrace{O(KN)}_{\text{sliding product sums}} \text{ v.s. } \underbrace{O(N)}_{\text{frequency multiplication}} + \underbrace{O(N \log N)}_{\text{Fourier transform}}$$

K = convolutional kernel size
N = image size

✅ Key advantage: Frequency domain filtering is computationally more efficient than direct convolution when logN < K (when kernel is large)

Main Topic 4