# Announcement

2/4/2025 Tuesday

- Patch-level processing (filtering)
  - Same filter applied to sub-regions/patches

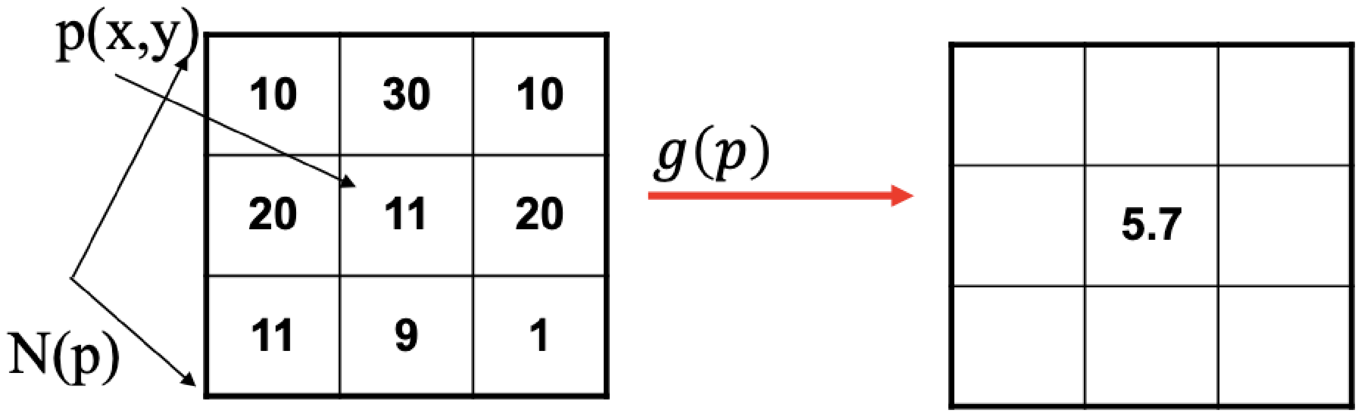
- HW01 out
  - Due 2/18 Tue, 5 PM CST
  - 100 pts = 90 pts for coding + 10 pts for written question
  - Zip and rename your solutions

- Final project

# Fundamentals of Spatial Filtering

- $g(p)$
  - Linear function
    - Correlation
    - Convolution

  - Non-linear function
    - Order statistics (e.g., median)



# Order-statistic filter

- Take the pixel values in a neighbourhood
- Sort (order) the values
  - Output one of the ranks
    - Max
    - Min
    - Median
  - Delete the two extremes and average the rest
    - Alpha-trimmed-mean filter

# Case 1: image segmentation

- Before: pixel-level thresholding

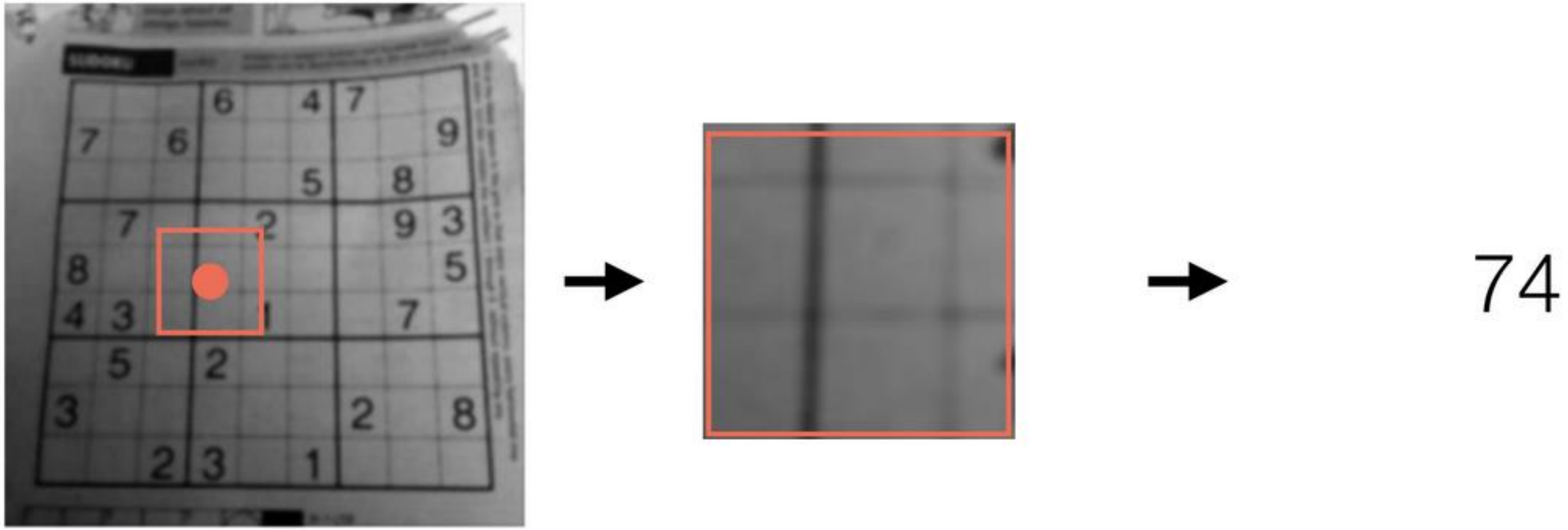
Image

Image>127

Image>60

# Case 1: image segmentation

- Patch-level: estimate the brightness for each pixel

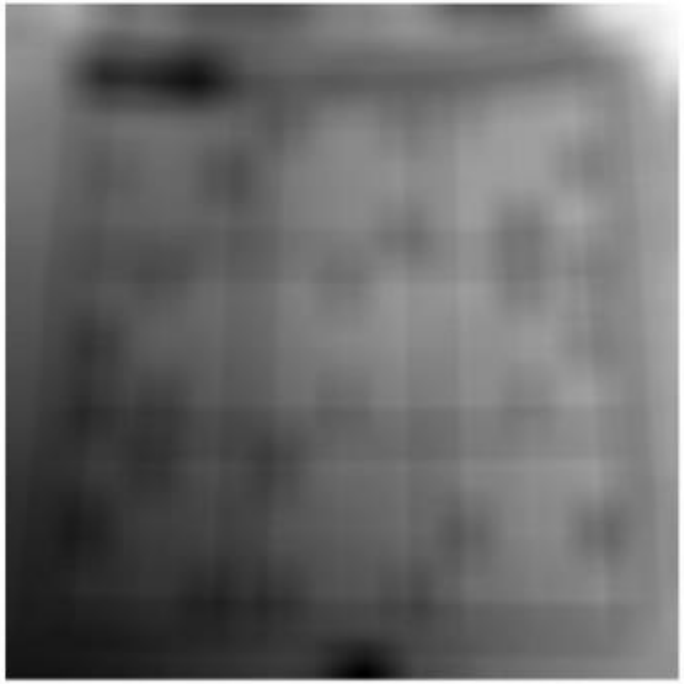



For every pixel | Surrounding patch | Estimated brightness (Mean or Gaussian filter result)

# Case 1: image segmentation

- Adaptive thresholding filter = (Impulse - Blur_big) > k

Image



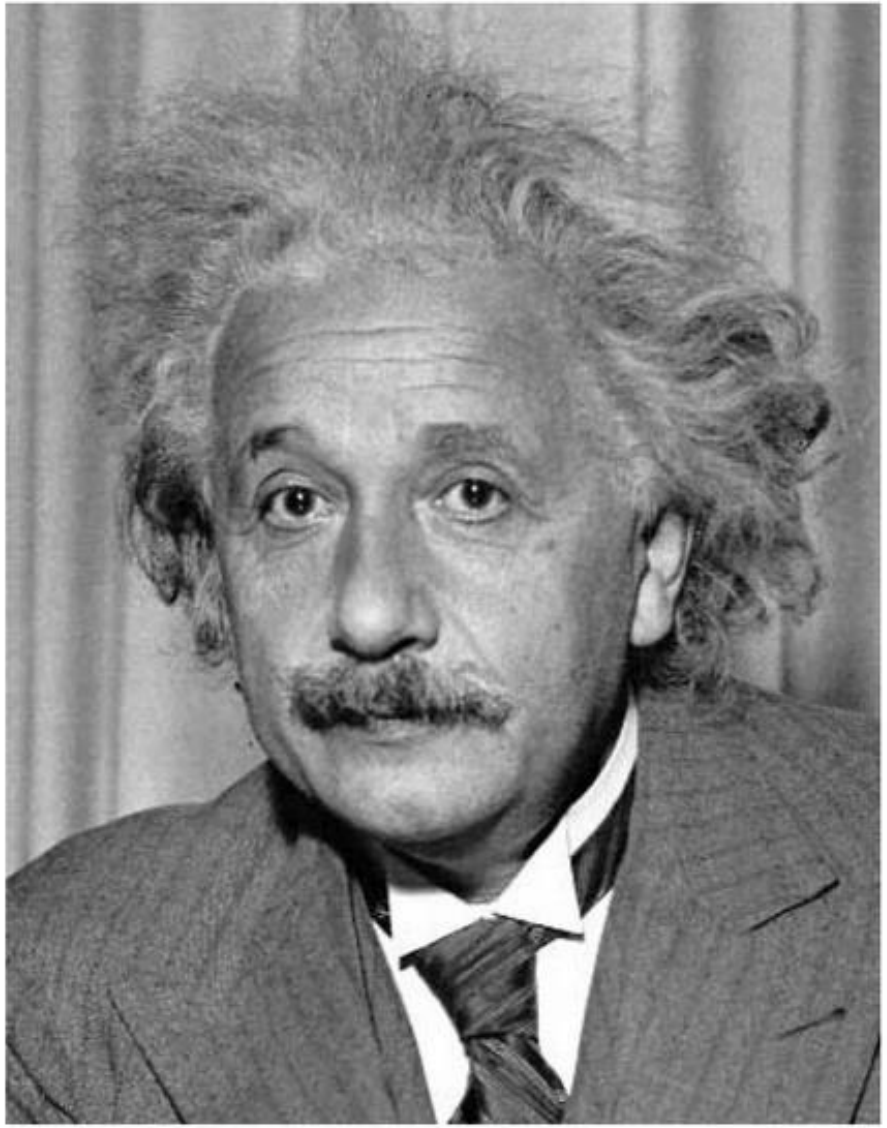
Estimated brightness
(Gaussian filter result)

Image-brightness > -5

# Case 2: object detection

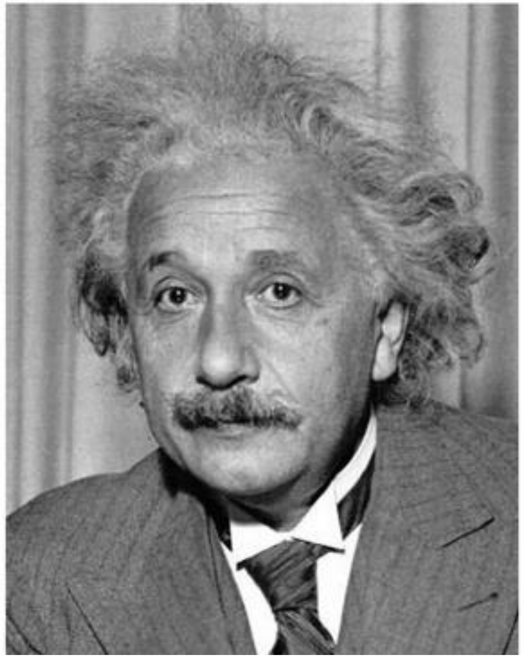
- Goal: find  in an image

- What's a good similarity/distance measure between two patches?



# Case 2: object detection

- Method 1: Filter the image with 
- $g(x,y) = \sum_{s,t} w(s,t) f(x+s, y+t)$

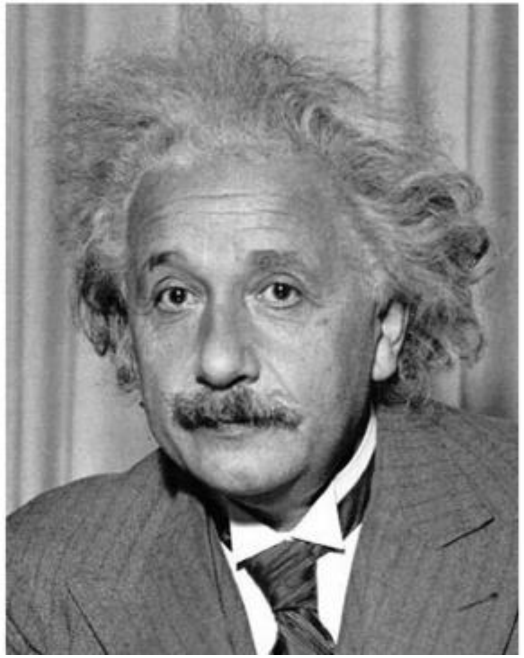


Input image

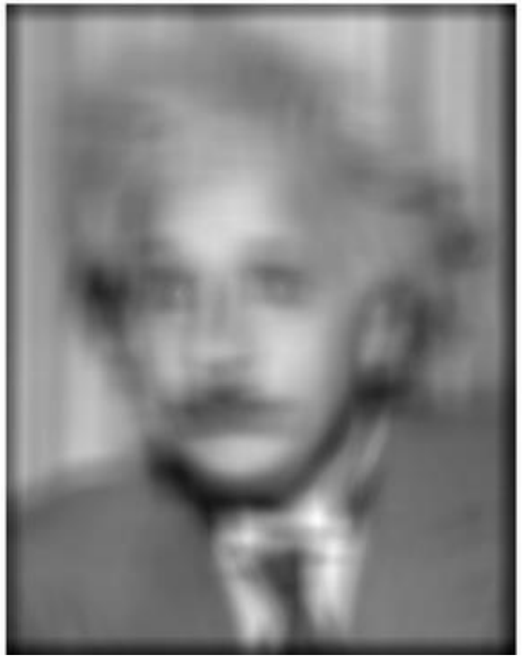
What went wrong?

# Case 2: object detection

- Method 1: Filter the image with 
- $g(x,y) = \sum_{s,t} w(s,t) f(x+s, y+t)$



Input image

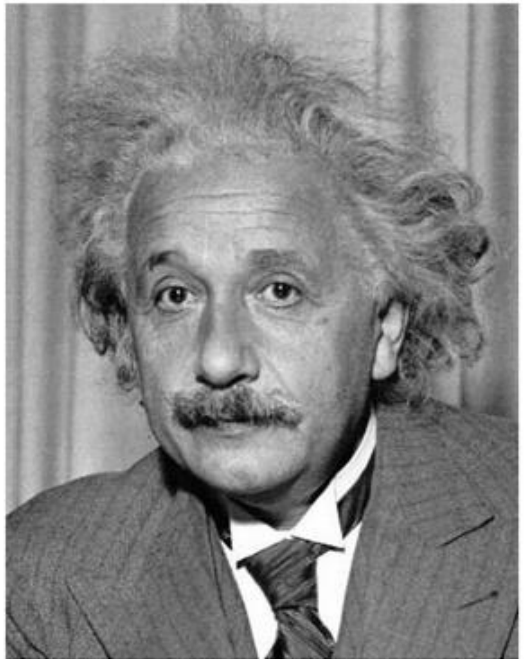


Filtered image

What went wrong?

# Case 2: object detection

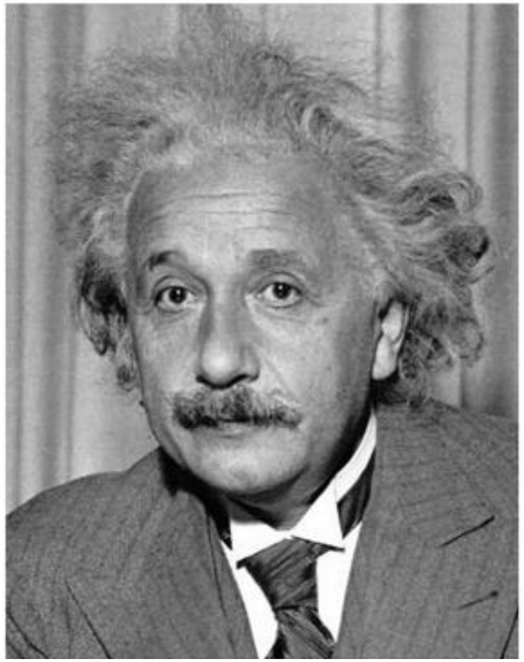
- Method 2: Filter the image with zero-mean eye
- $g(x, y) = \sum_{s,t}(w(s, t) - \overline{w})f(x + s, y + t)$



Input image

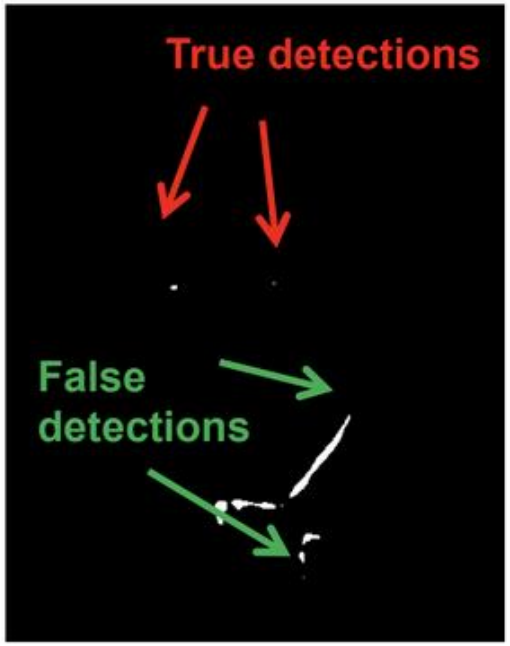
# Case 2: object detection

- Method 2: Filter the image with zero-mean eye
- $g(x,y) = \sum_{s,t} (w(s,t) - \overline{w}) f(x+s, y+t)$



Input image

Filtered image (scaled)




Threshold image

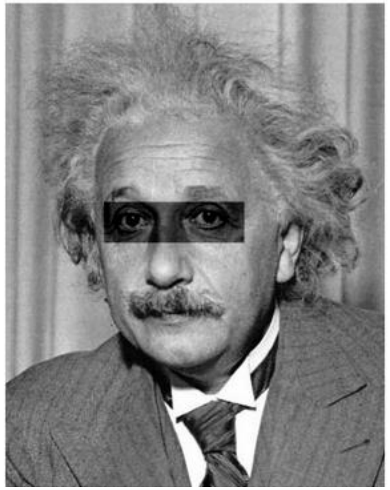
# Case 2: object detection

- Method 3: Normalized cross-correlation
- Divide by standard deviation of both patches, so they are unit vectors

Mean template

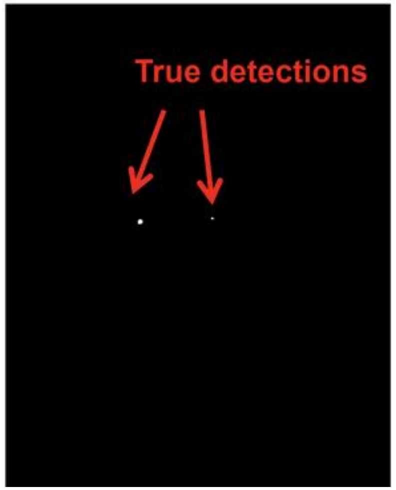
Mean image patch

$$g(x,y) = \frac{\sum_{s,t}(w(s,t)-\overline{w})\left(f(x+s,y+t)-\overline{f}_{x,y}\right)}{\sqrt{\sum_{s,t}(w(s,t)-\overline{w})^2 \sum_{s,t}\left(f(x+s,y+t)-\overline{f}_{x,y}\right)^2}}$$



Input image

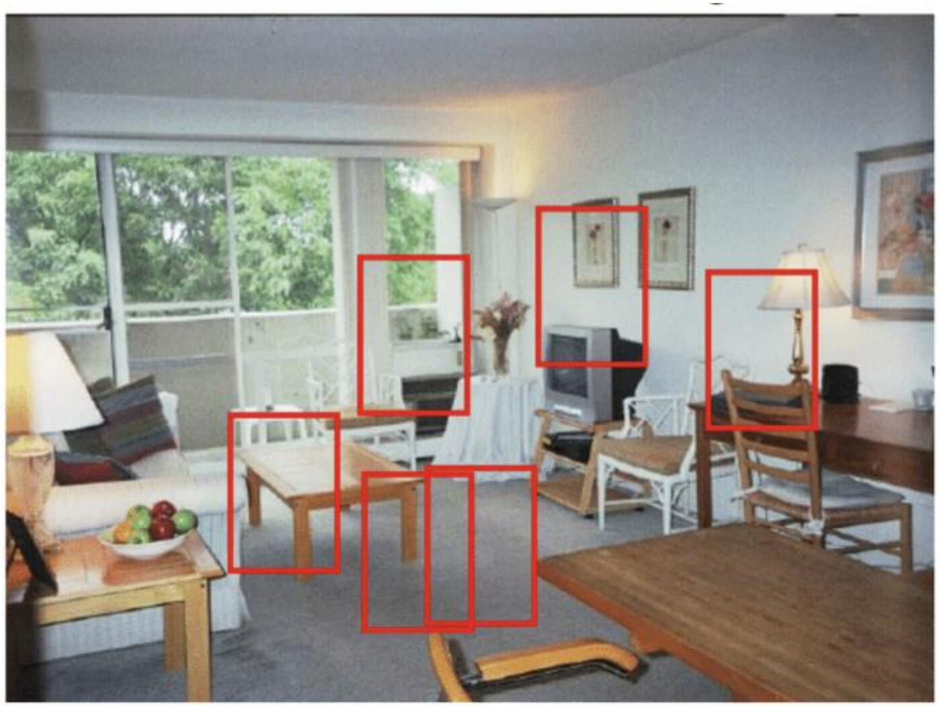
Normalized X-correlation

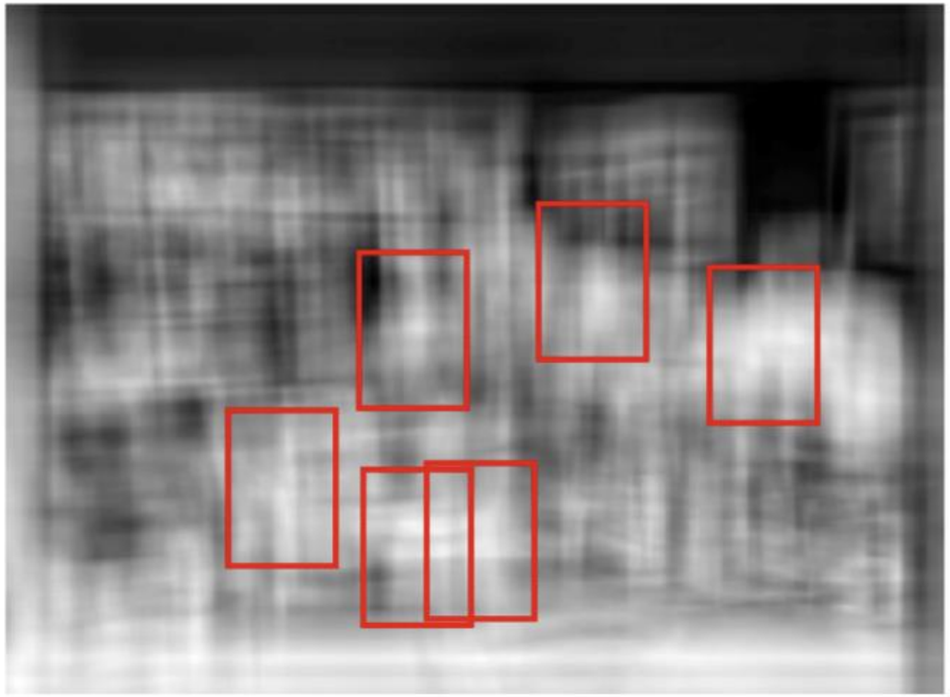



Threshold image

# Recognizing objects: is it really so hard?

- Find the chair in this image



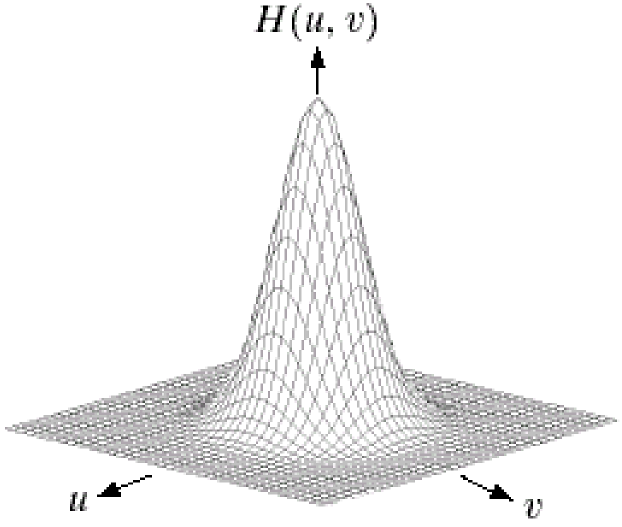


Not so great!

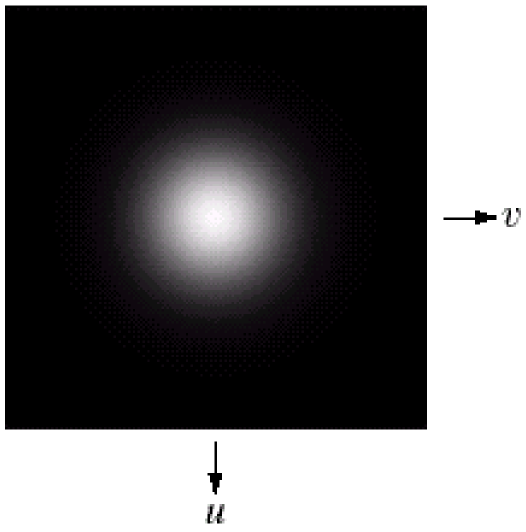
# Case 3: image enhancement

- Different $\sigma$ values result in different degrees of smoothing

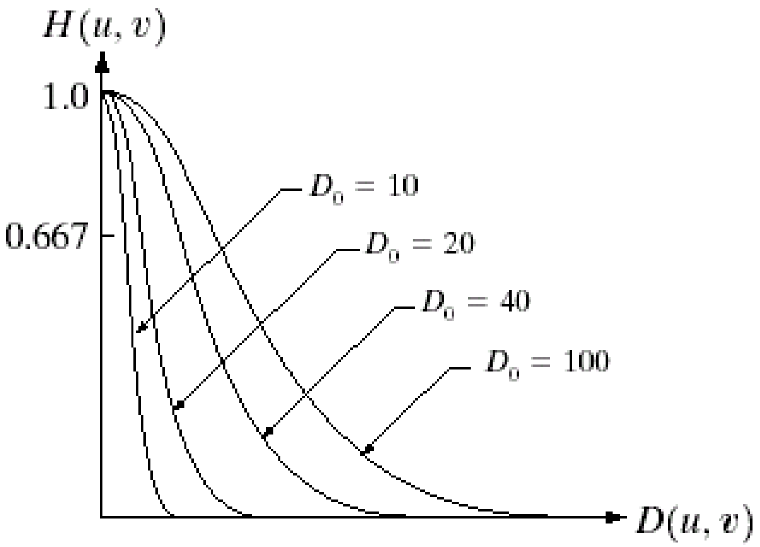
Gaussian filter




Filter seen as an image




Cross sections for different $\sigma = D_0$



# Case 3: image enhancement

- Different $\sigma$ values result in different degrees of smoothing

$\sigma = 4$ $\sigma = 8$ $\sigma = 16$ $\sigma = 32$

Image filters aren’t “aware” of edges

# Case 3: image enhancement

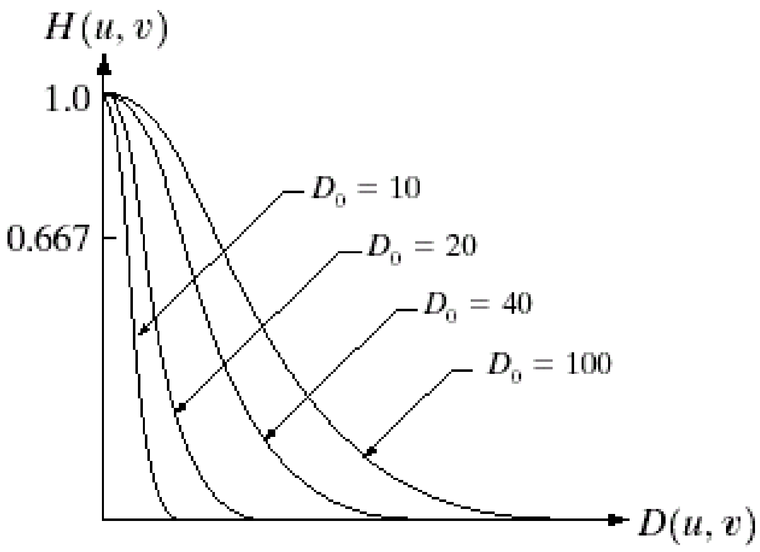
- Why is this happening?


Gaussian filter can be written:

$$GB[\mathbf{p}] = \sum_{\mathbf{q} \in \mathcal{N}} G_\sigma(||\mathbf{p} - \mathbf{q}||) I[\mathbf{q}]$$

Other pixels

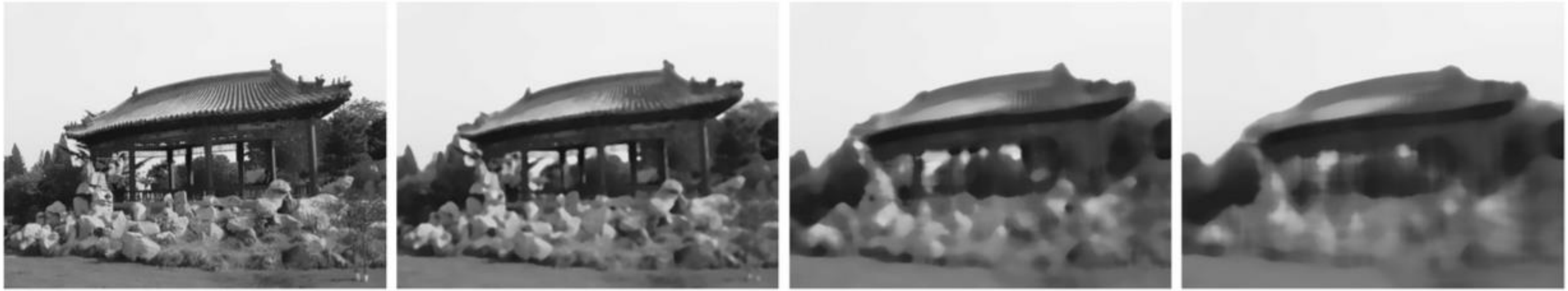
Blurred intensity Gaussian density Intensity



# Case 3: image enhancement

- Median filtering window sizes

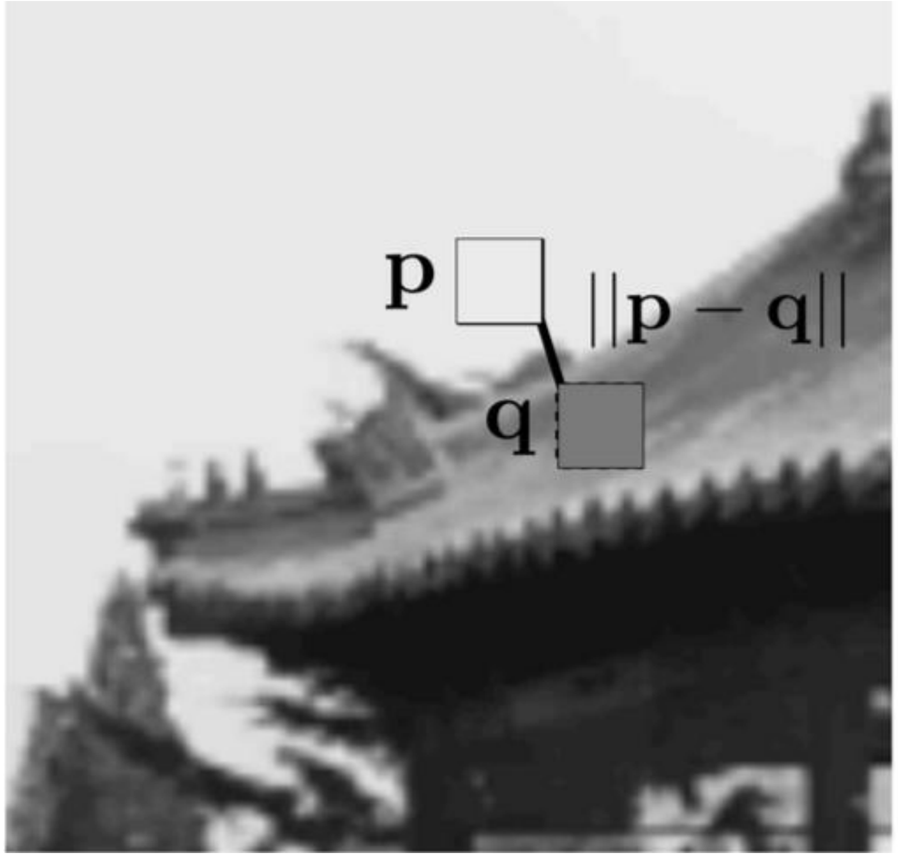
$$MB[\mathbf{p}] = \underset{\mathbf{q} \in \mathcal{N}}{\mathrm{median}}\ I[\mathbf{q}]$$



Original w = 5 w = 13 w = 21

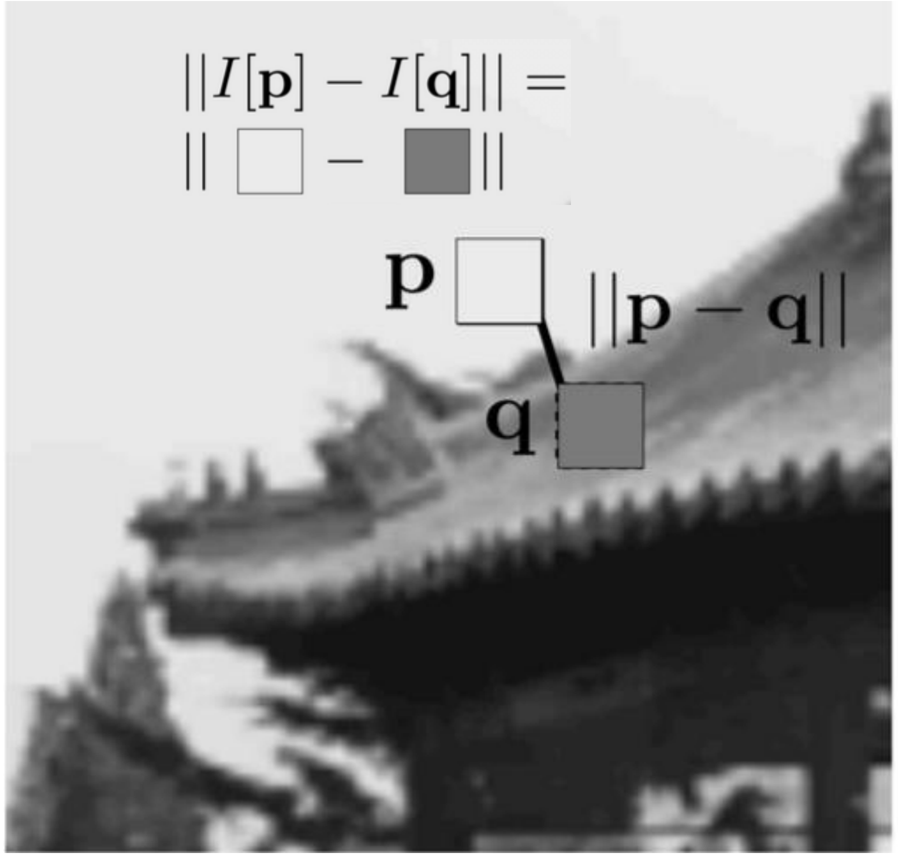
# Case 3: image enhancement

- Bilateral filtering: What if we weight by **appearance**?



# Case 3: image enhancement

- Bilateral filtering: What if we weight by **appearance**?




**Gaussian filter:**

$$GB[\mathbf{p}] = \sum_{\mathbf{q} \in \mathcal{N}} G_\sigma(||\mathbf{p} - \mathbf{q}||) I[\mathbf{q}]$$

**Bilateral filter:**

$$BB[\mathbf{p}] = \frac{1}{W_p} \sum_{\mathbf{q} \in \mathcal{N}} G_{\sigma_s}(||\mathbf{p} - \mathbf{q}||) G_{\sigma_r}(||I[\mathbf{p}] - I[\mathbf{q}]||) I[\mathbf{q}]$$

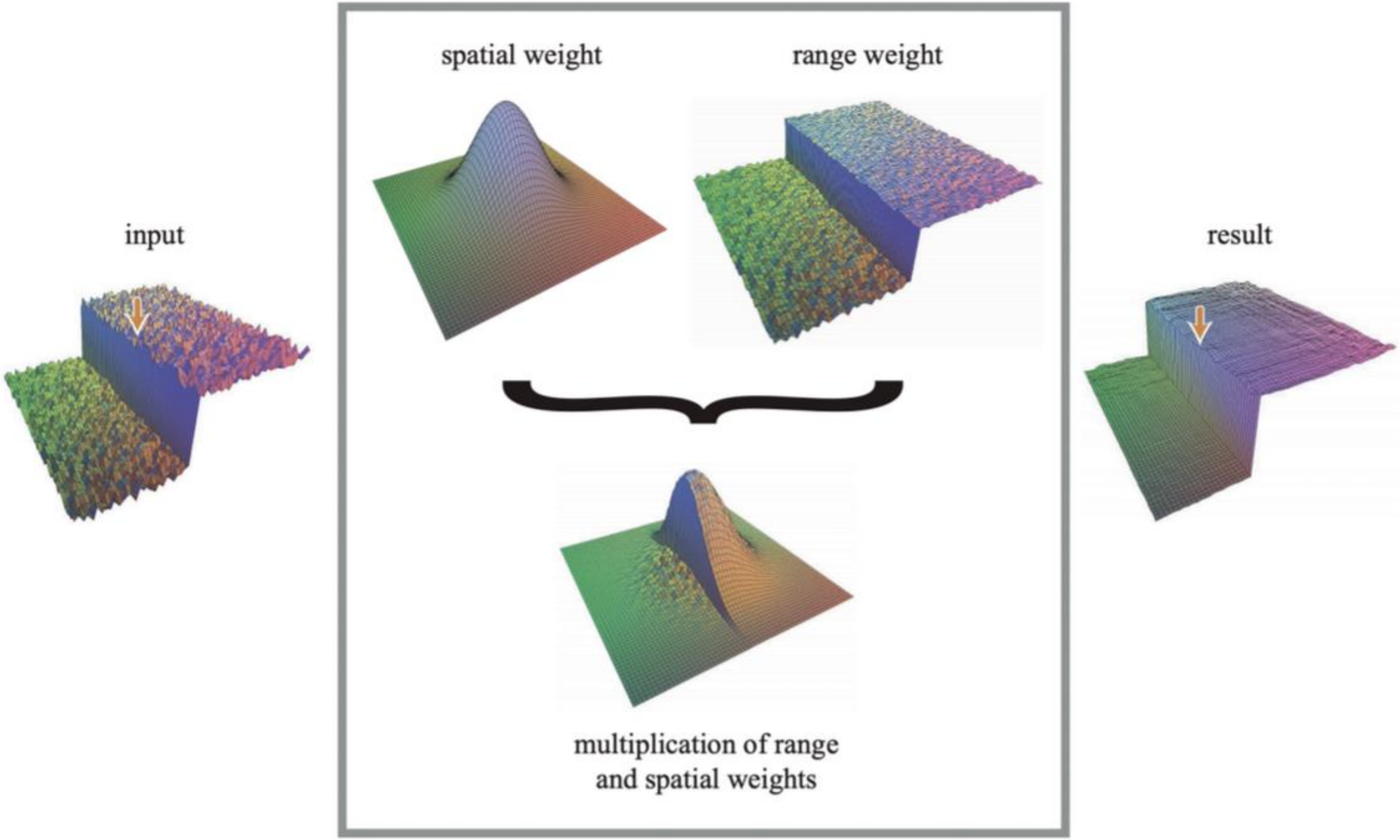
Normalization constant (to make weights sum to 1)

Spatial Gaussian

Range Gaussian

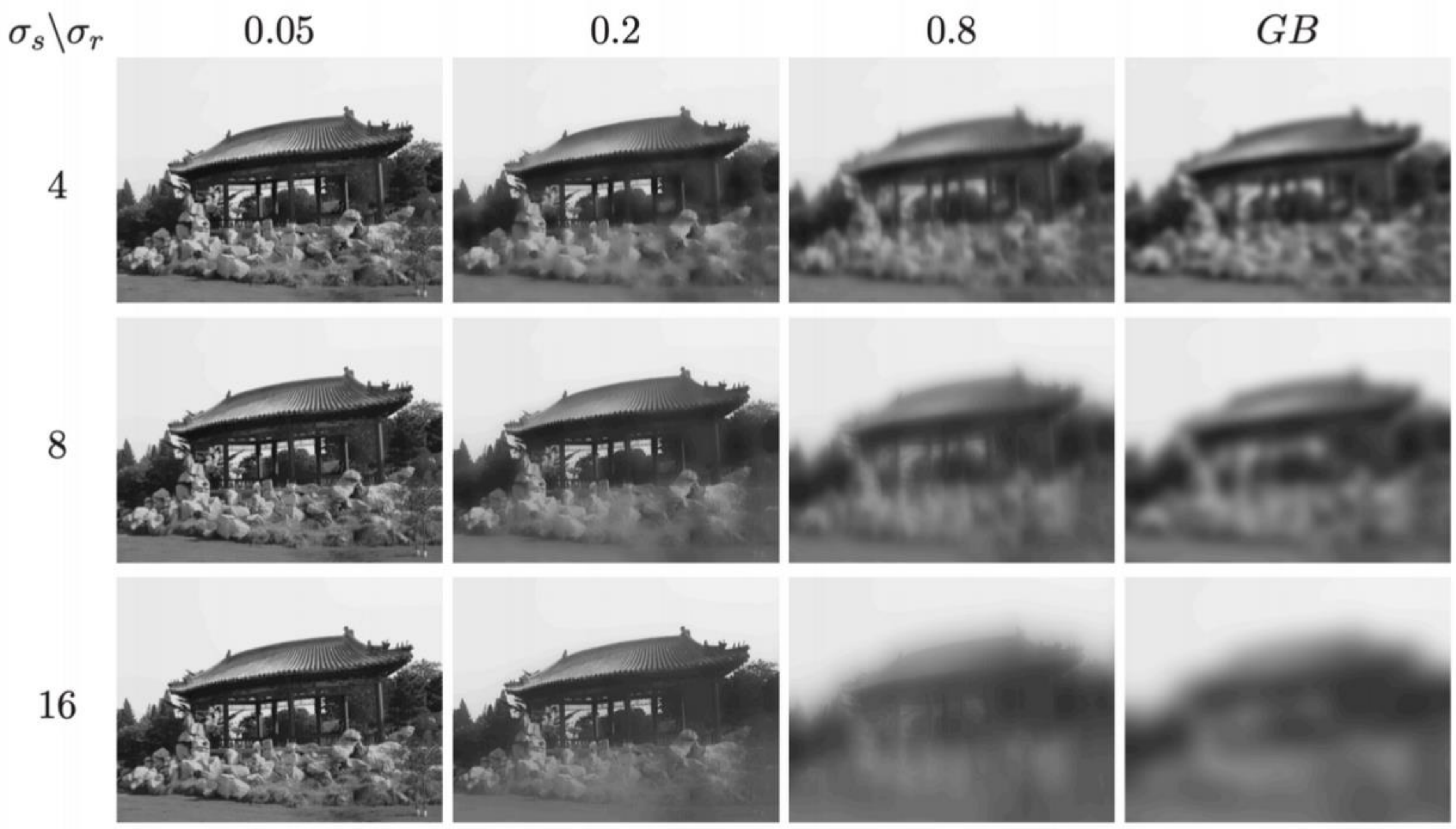
# Case 3: image enhancement

- Bilateral filter



# Case 3: image enhancement

- Bilateral filter



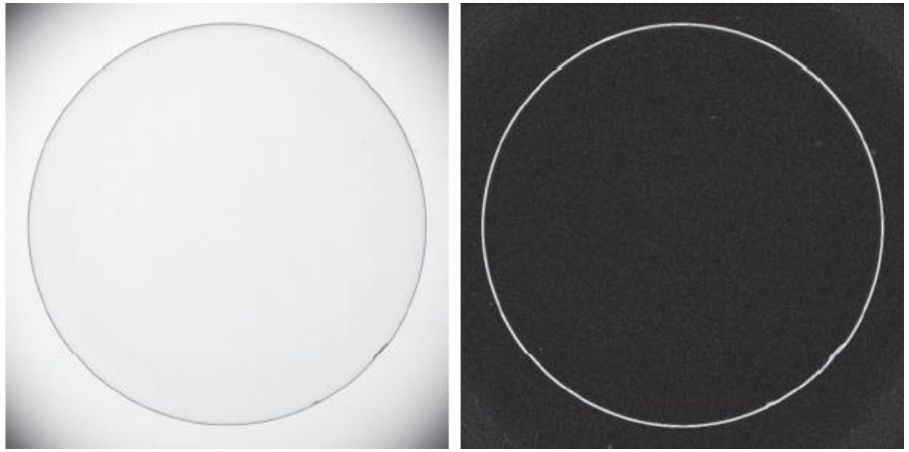
# Summary

- Linear filtering
  - Correlation
  - Convolution
  - Example:
    - Box, Gaussian, edge detection

- Non-linear filtering
  - Order statistics (e.g., median)
  - Example:
    - Max, Min, Median, Alpha-trimmed-mean filter
    - Match filter, bilateral edge detection

# What does filtering do in frequency domain?

- Low pass
  - Average
  - Basic overall shape of the function
- High pass
  - Differences
  - Details of the function



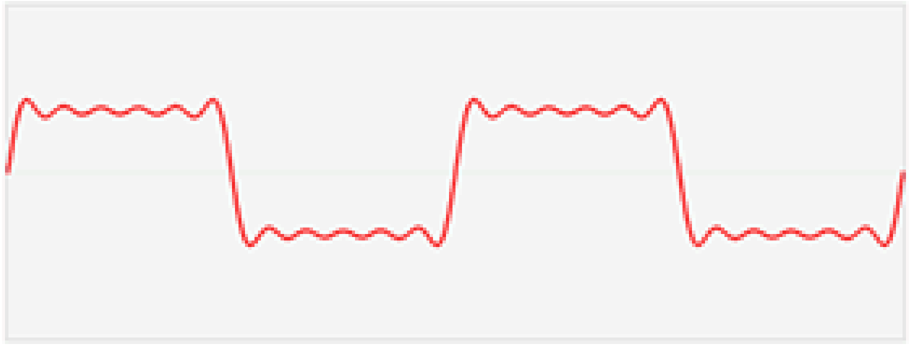
Sobel gradient

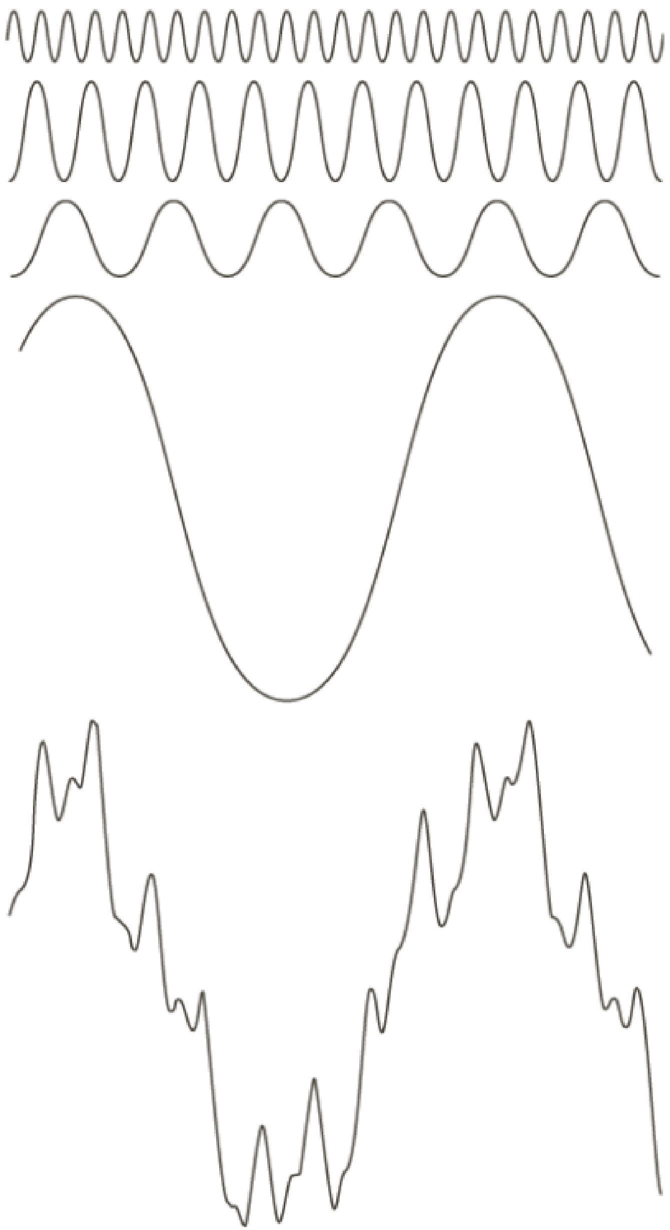
$\sigma = 4$ $\sigma = 8$ $\sigma = 16$ $\sigma = 32$

Gaussian filtering

# Fourier series & transforms

- **Fourier series**: any periodic function can be represented by a discrete weighted sum of sines and cosines



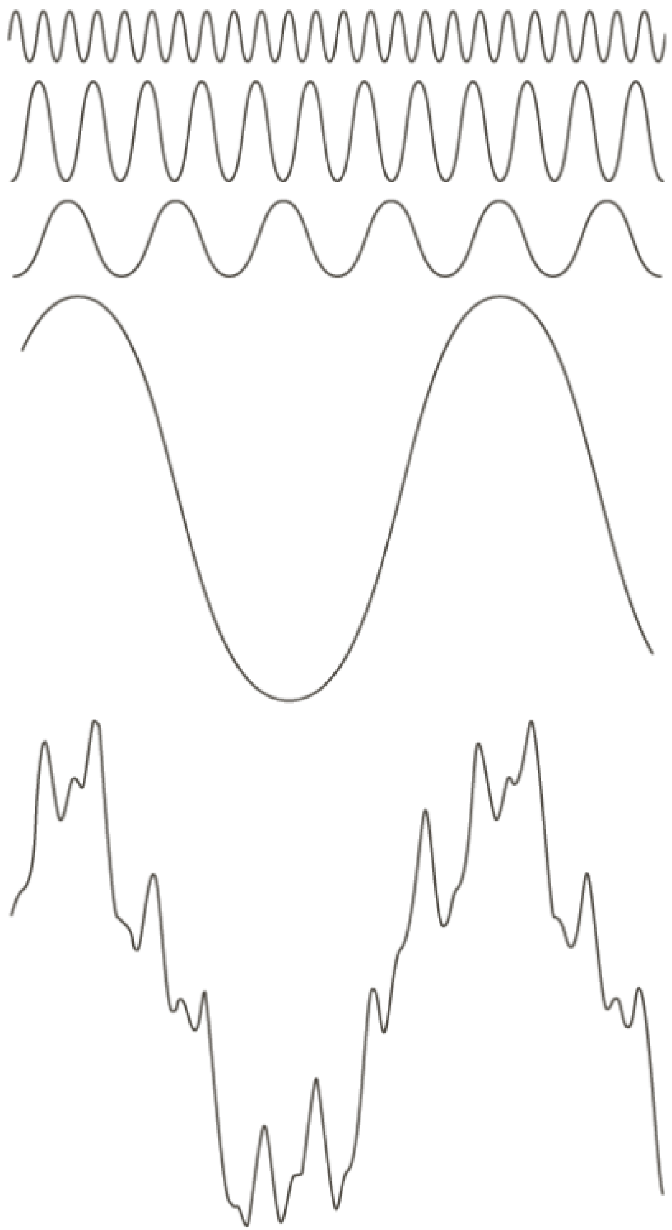


https://en.wikipedia.org/wiki/Fourier_transform#/media/File:Fourier_transform_time_and_frequency_domains_(small).gif

# Fourier series & transforms

- **Fourier series**: any periodic function can be represented by a discrete weighted sum of sines and cosines

- **Fourier transform**: an arbitrary function with finite duration (non-periodic function) can be expressed by a weighted integrals of sines and cosines



# Fourier series & transforms

- $f(t)$ is a continuous function with period $T$, we have:

$$f(t) = \sum_{n=-\infty}^{+\infty} c_n e^{\frac{j2\pi nt}{T}}$$

Coefficient

Discrete frequency

$$c_n = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)e^{-\frac{j2\pi nt}{T}}dt\,,$$

$$n = 0, \pm 1, \pm 2, \ldots$$

- $f(t)$ is an arbitrary non-periodic function, we have:

$$f(t) = \int_{-\infty}^{\infty} F(u)e^{j2\pi\mu t}d\mu$$

Coefficient

Discrete frequency

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

# FT of simple functions

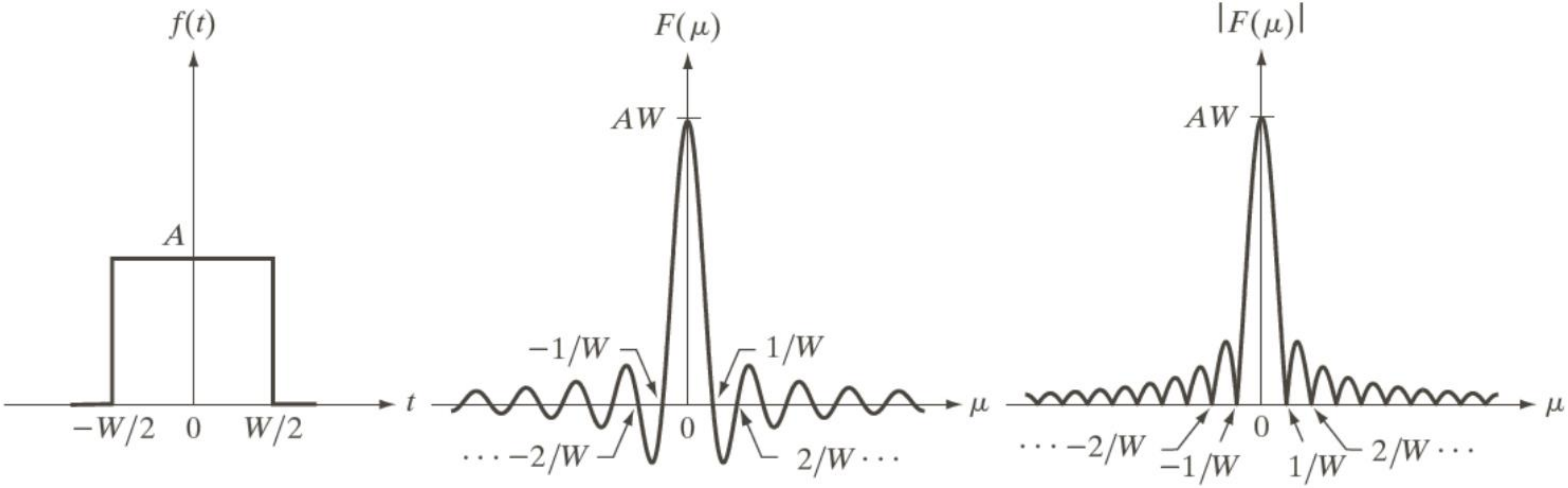
- Rectangle function

$$f(t) = \begin{cases} A & -\dfrac{w}{2} \le t \le \dfrac{w}{2} \\ 0 & otherwise \end{cases}$$

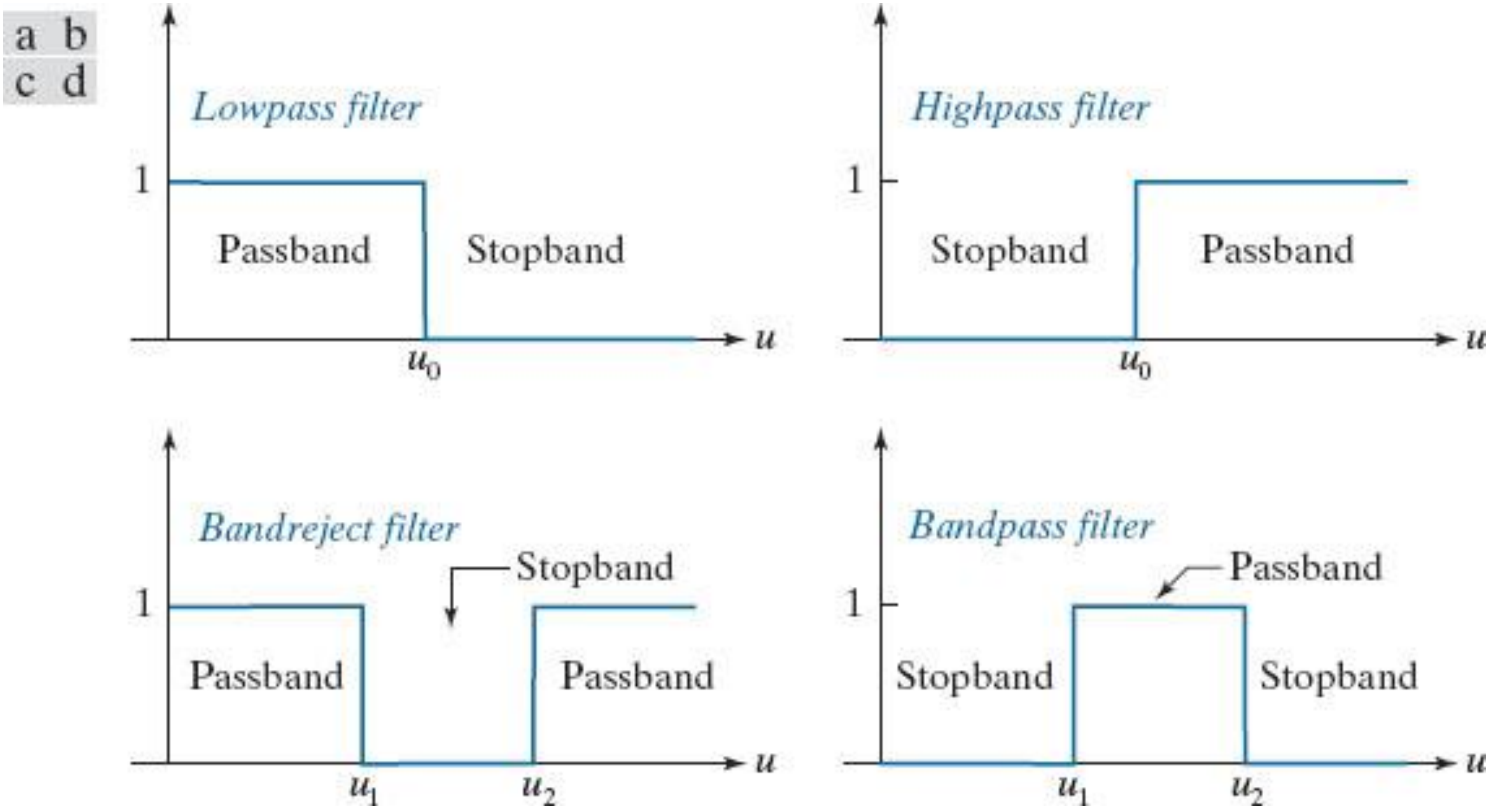
$$F(\mu) = \frac{A}{\pi\mu} \sin \pi w\mu = Aw \frac{\sin \pi w\mu}{\pi w\mu} = Aw\ sinc(\pi w\mu)$$

# FT of simple functions

- Rectangle function



# Lowpass & highpass filter

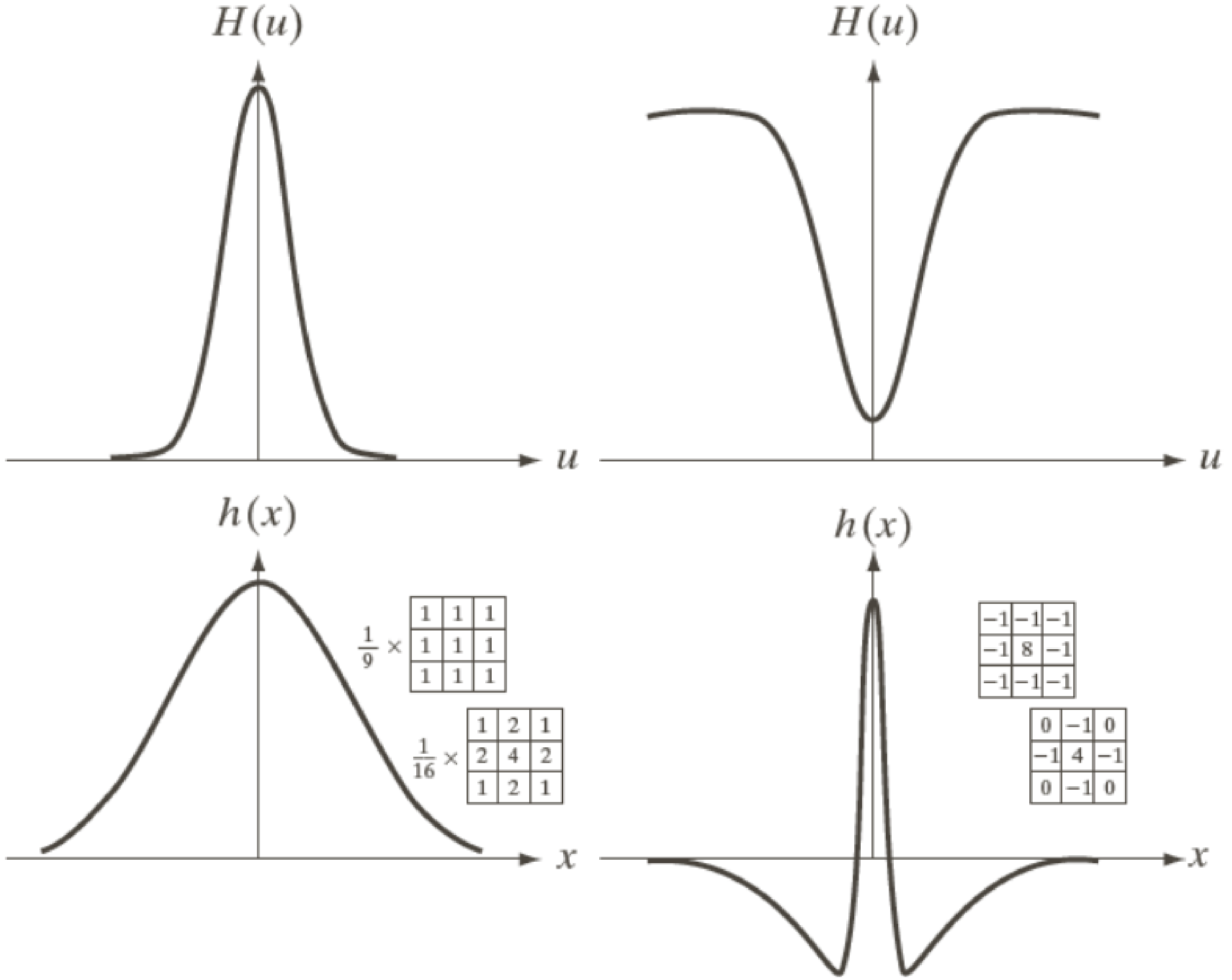


# Correspondence to the spatial domain filter

- The FT of a Gaussian function is still a Gaussian function

1D Gaussian lowpass filter (frequency domain)

1D Gaussian lowpass filter (spatial domain)




1D Gaussian highpass filter (frequency domain)

1D Gaussian highpass filter (spatial domain)

# A 2D example

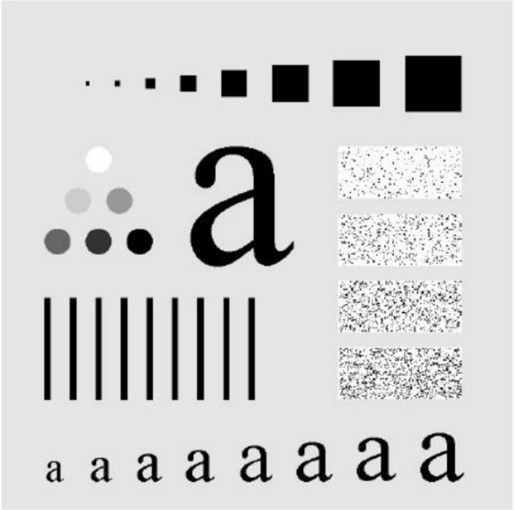
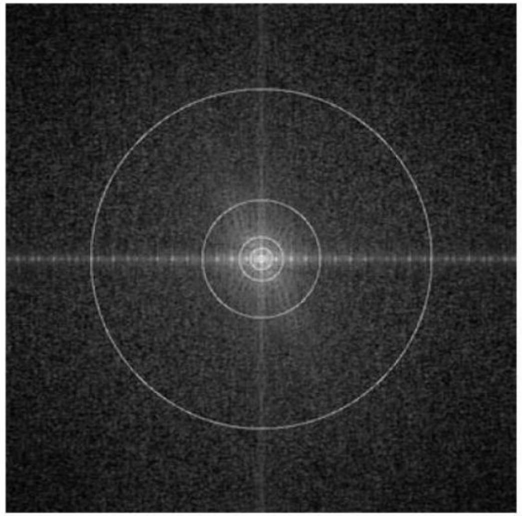
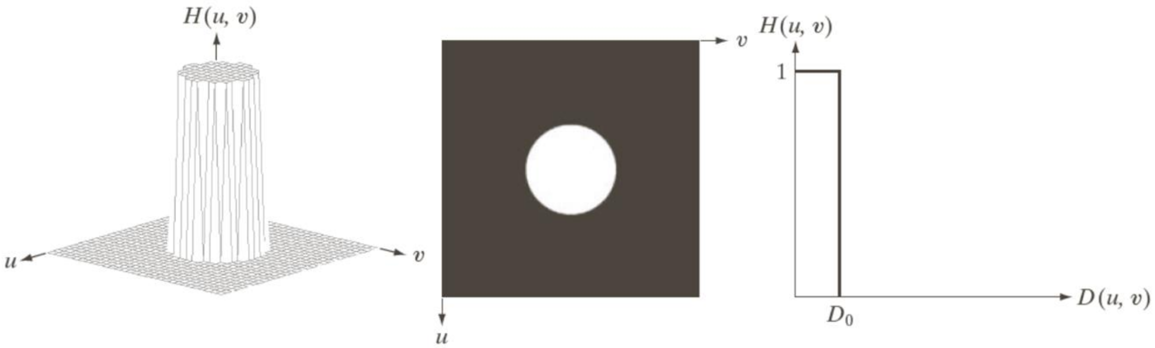


Image (668 × 668)



Fourier spectrum




Ideal Lowpass Filter

Original

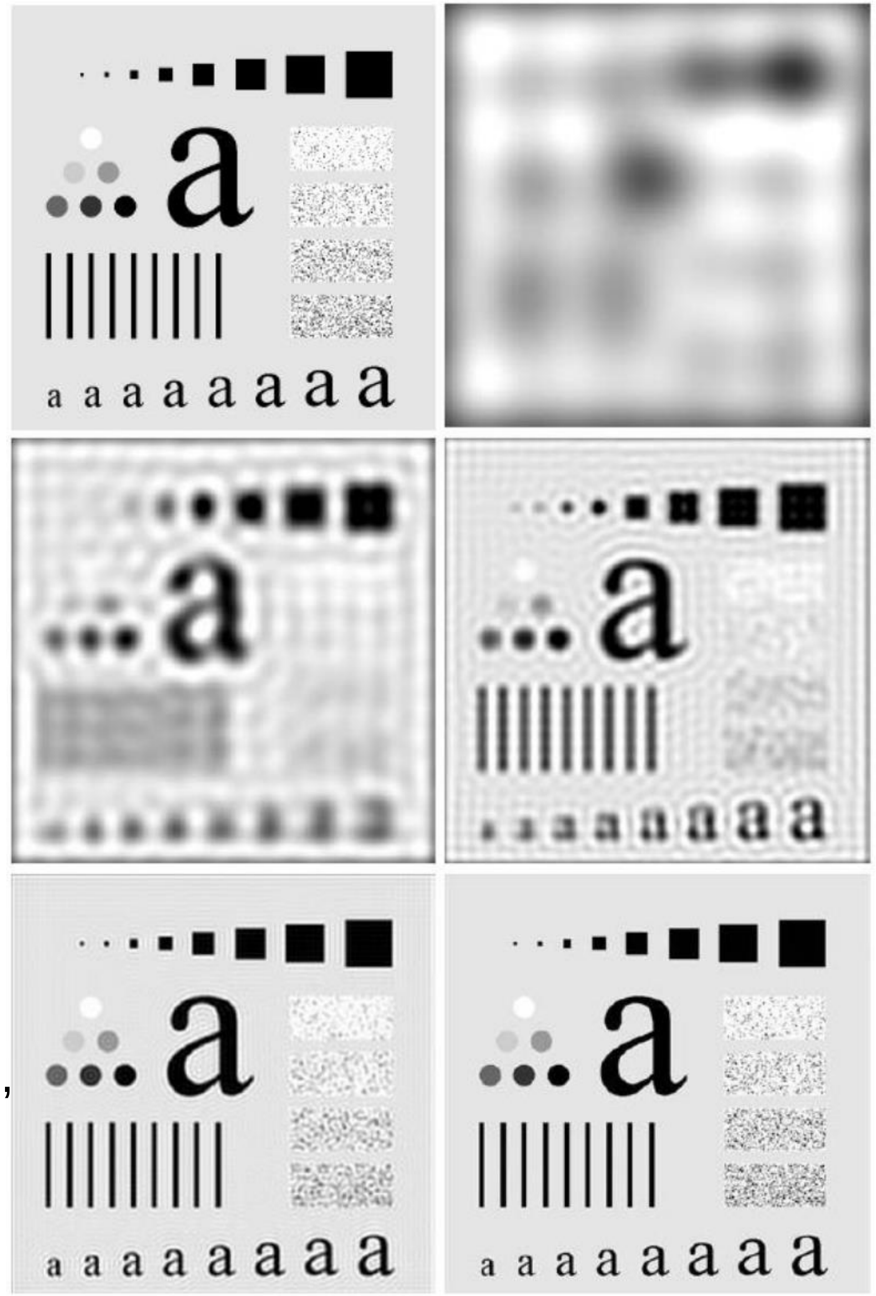
ILPF, cutoff 10, Energy 87%

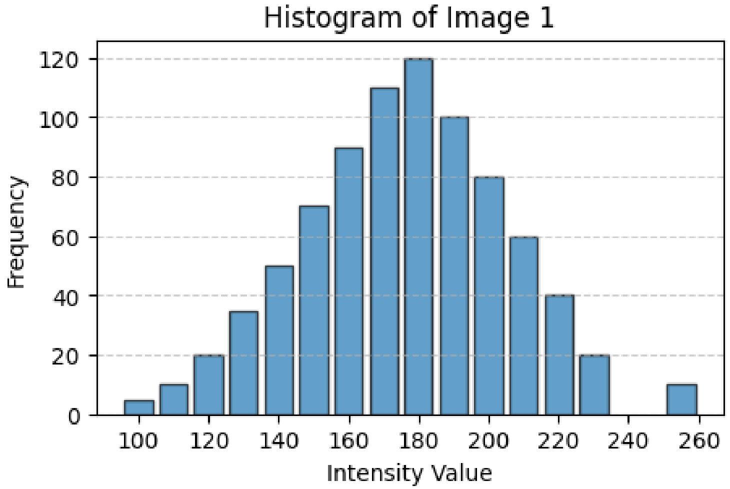
ILPF, cutoff 30, Energy 93.1%

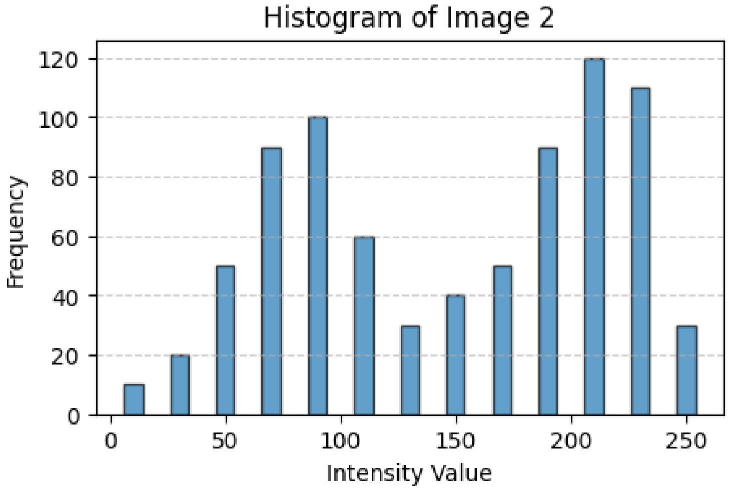
ILPF, cutoff 60, Energy 95.7%

ILPF, cutoff 160, Energy 97.8%

ILPF, cutoff 460, Energy 99.2%






| Intensity | # of pixels |
|---|---|
| 100 | 5 |
| 110 | 10 |
| 120 | 20 |
| 130 | 35 |
| 140 | 50 |
| 150 | 70 |
| 160 | 90 |
| 170 | 110 |
| 180 | 120 |
| 190 | 100 |
| 200 | 80 |
| 210 | 60 |
| 220 | 40 |

| Intensity | # of pixels |
|---|---|
| 10 | 10 |
| 30 | 20 |
| 50 | 50 |
| 70 | 90 |
| 90 | 100 |
| 110 | 60 |
| 130 | 30 |
| 150 | 40 |
| 170 | 50 |
| 190 | 90 |
| 210 | 120 |
| 230 | 110 |
| 250 | 30 |