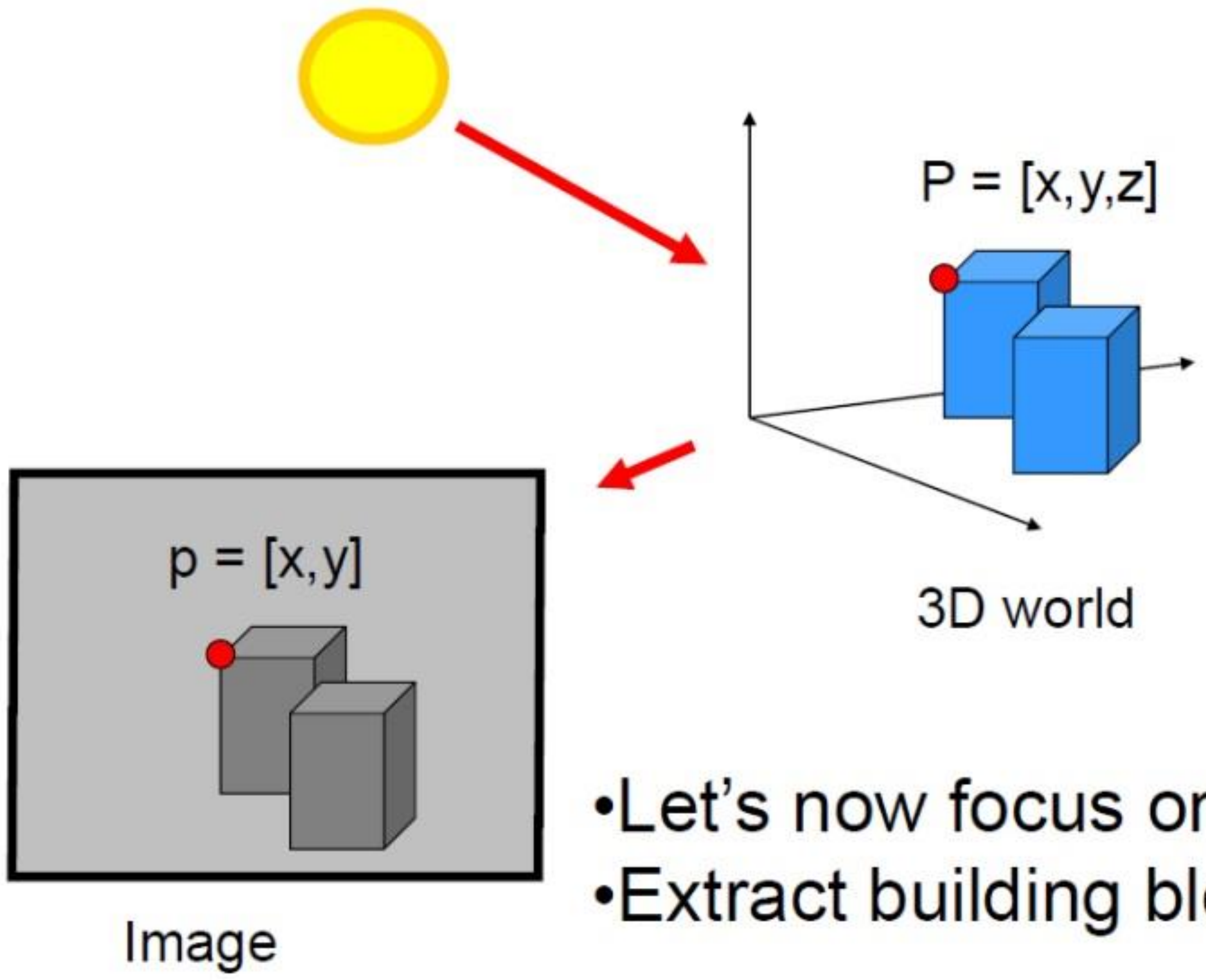
University at Buffalo
The State University of New York

Recap



- Let’s now focus on 2D
- Extract building blocks

Slide credit Fei Fei Li

Image filtering

- Image filtering: compute function of local neighborhood at each position

- Really important!
  - Enhance images
    - Denoise, resize, increase contrast, etc.
  - Extract information from images
    - Texture, edges, distinctive points, etc.
  - Detect patterns
    - Template matching

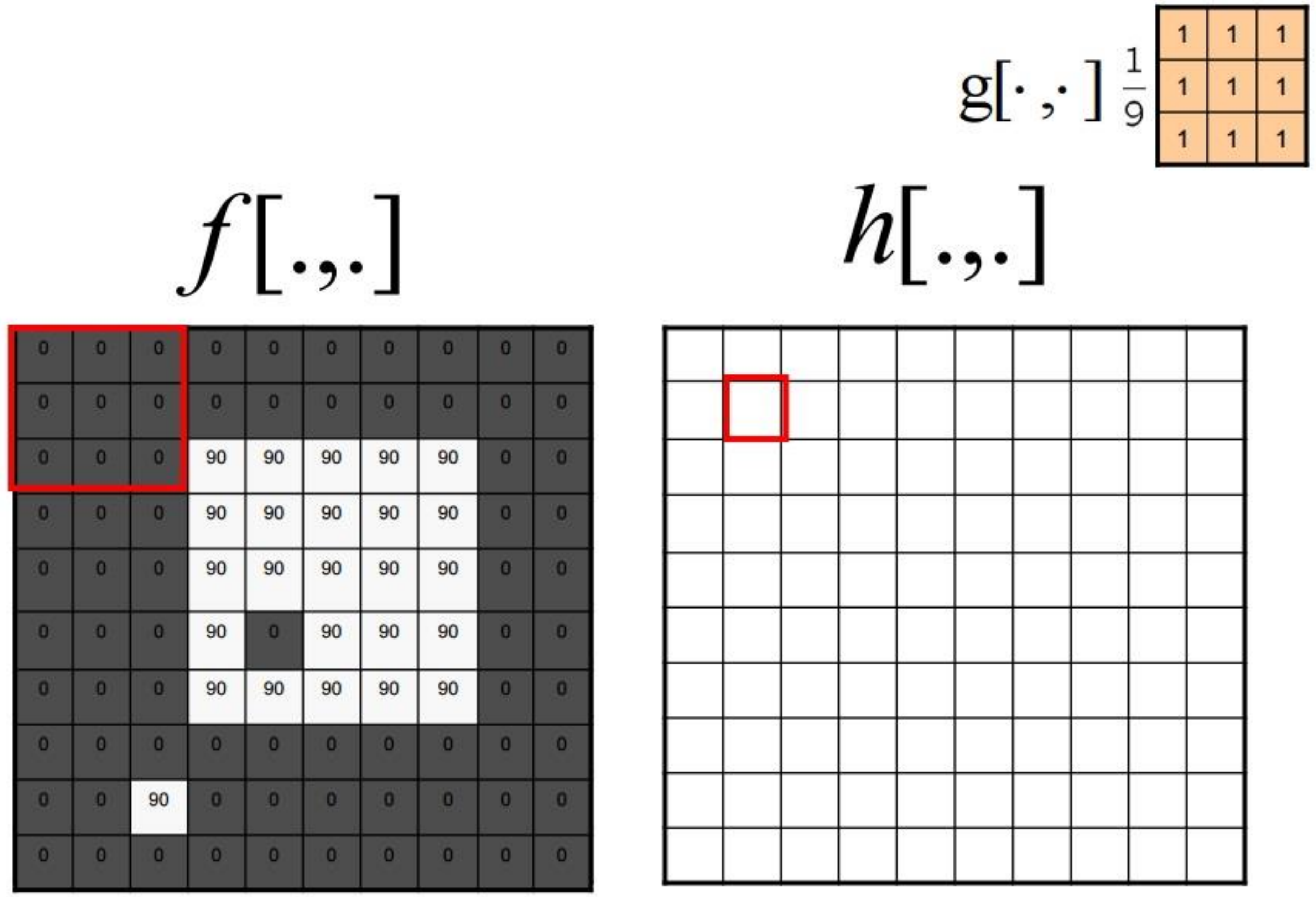
Image filtering - Example

$$g[\cdot,\cdot]$$

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

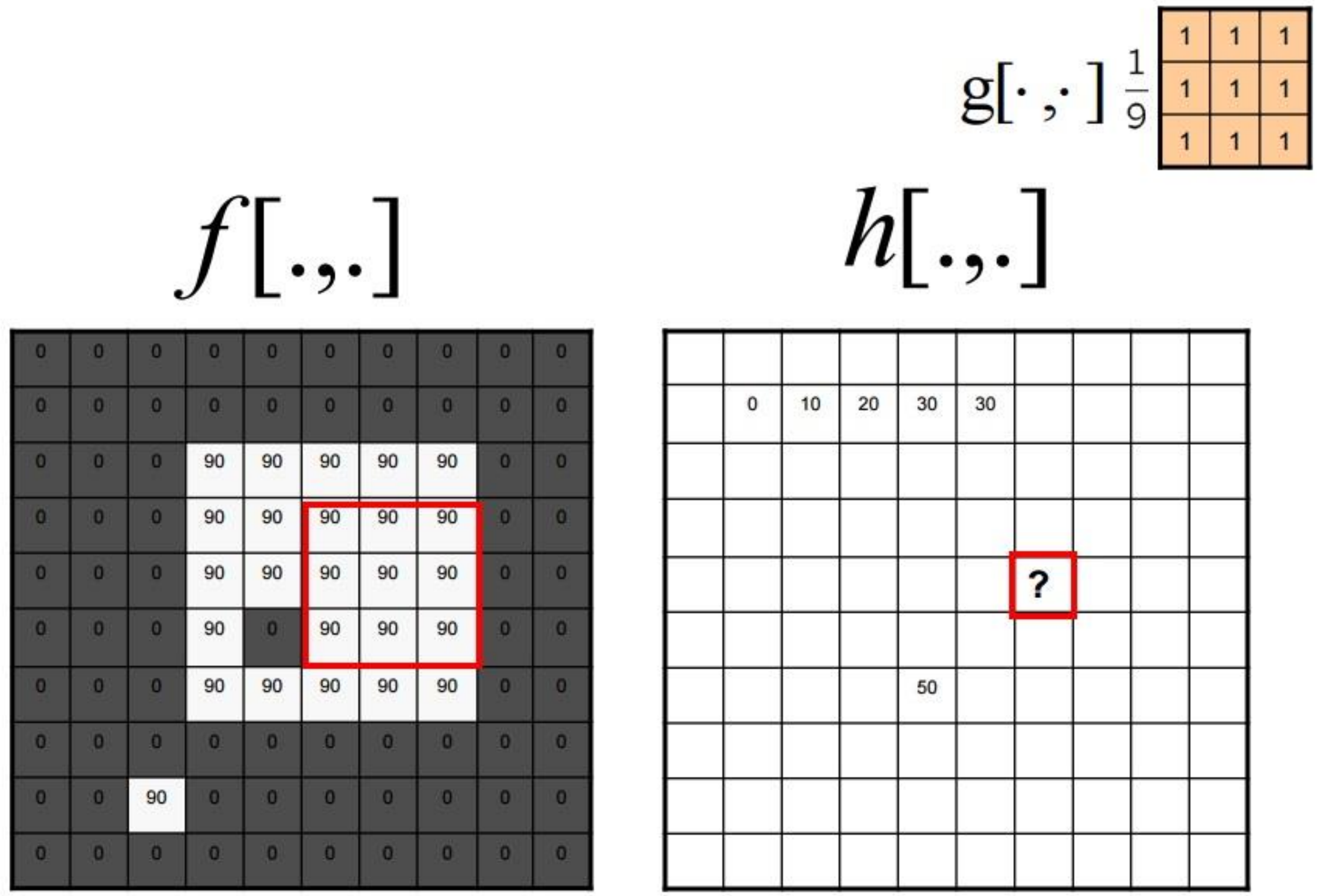
Slide credit: David Lowe (UBC)

Image filtering - Example



Credit: S. Seitz

Image filtering - Example



Credit: S. Seitz

University at Buffalo
The State University of New York

Image filtering – Correlation Formula

$$g[\cdot,\cdot] \quad \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

Credit: S. Seitz

Image filtering - Example

What does it do?

- Replaces each pixel with an average of its neighborhood
- Achieve smoothing effect (remove sharp features)

$$g[\cdot\,,\cdot\,]$$

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

Slide credit: David Lowe (UBC)

Image filtering - Example

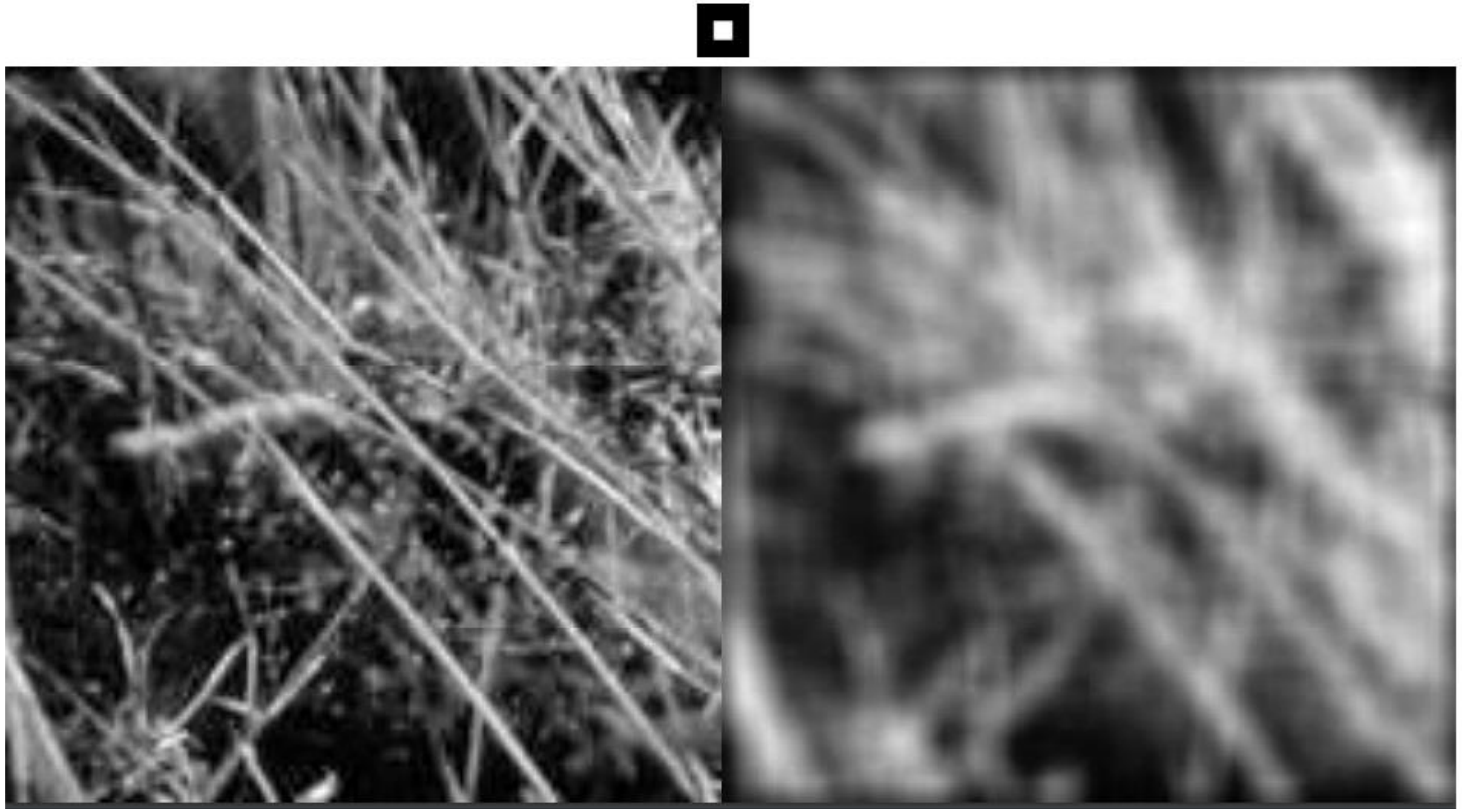


Image filtering – Using Correlation Formula

Original

0	0	0
0	1	0
0	0	0

?

Source: D. Lowe

Image filtering - Questions

Original

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Filtered
(no change)

Image filtering - Questions

Original

0	0	0
0	0	1
0	0	0

?

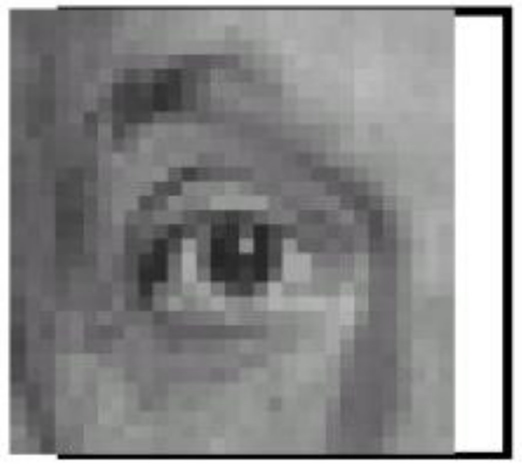
Source: D. Lowe

Image filtering - Questions

Correlation formula leads to counter intuitive image filtering

original

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$



shifted

Image filtering - Convolution

- Same as cross-correlation, except that the kernel is "flipped" (horizontally and vertically)

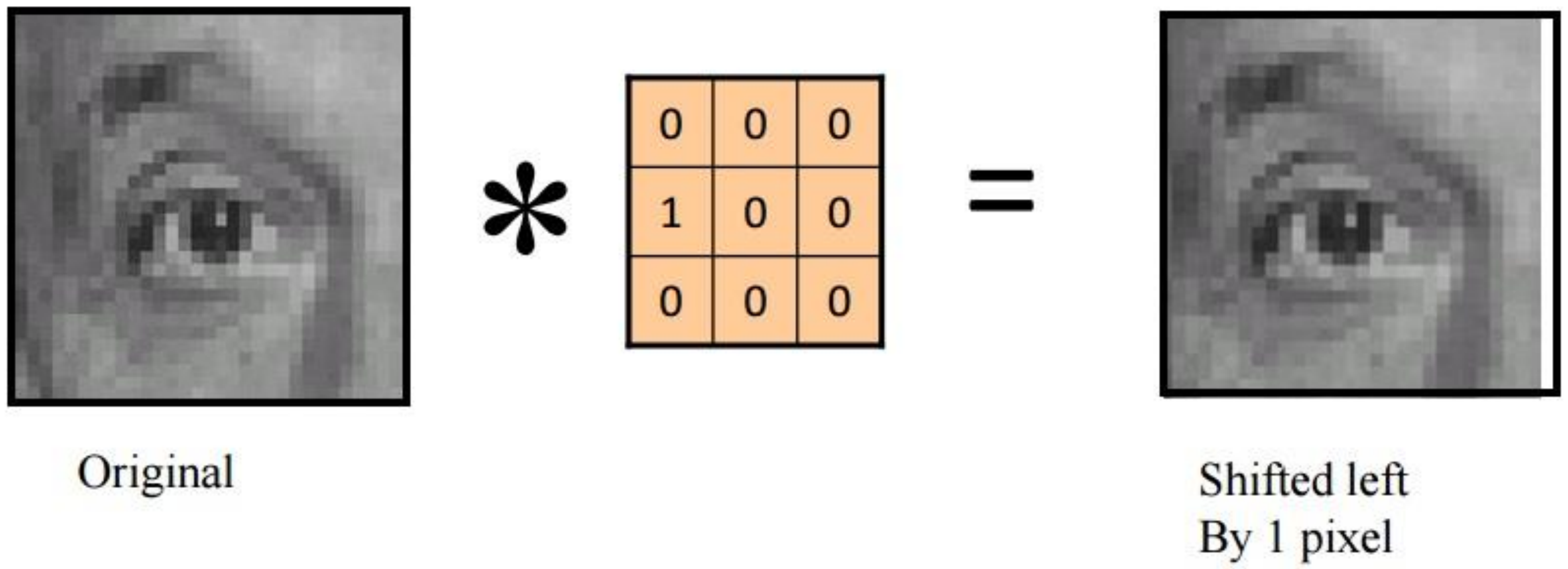
$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u,j-v]$$

This is called a **convolution** operation:

$$G = H * F$$

- Convolution is **commutative** and **associative**

Image filtering - Convolution



Source: D. Lowe

Image filtering - Sharpening

Original

$$\begin{bmatrix}0&0&0\\0&2&0\\0&0&0\end{bmatrix} - \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

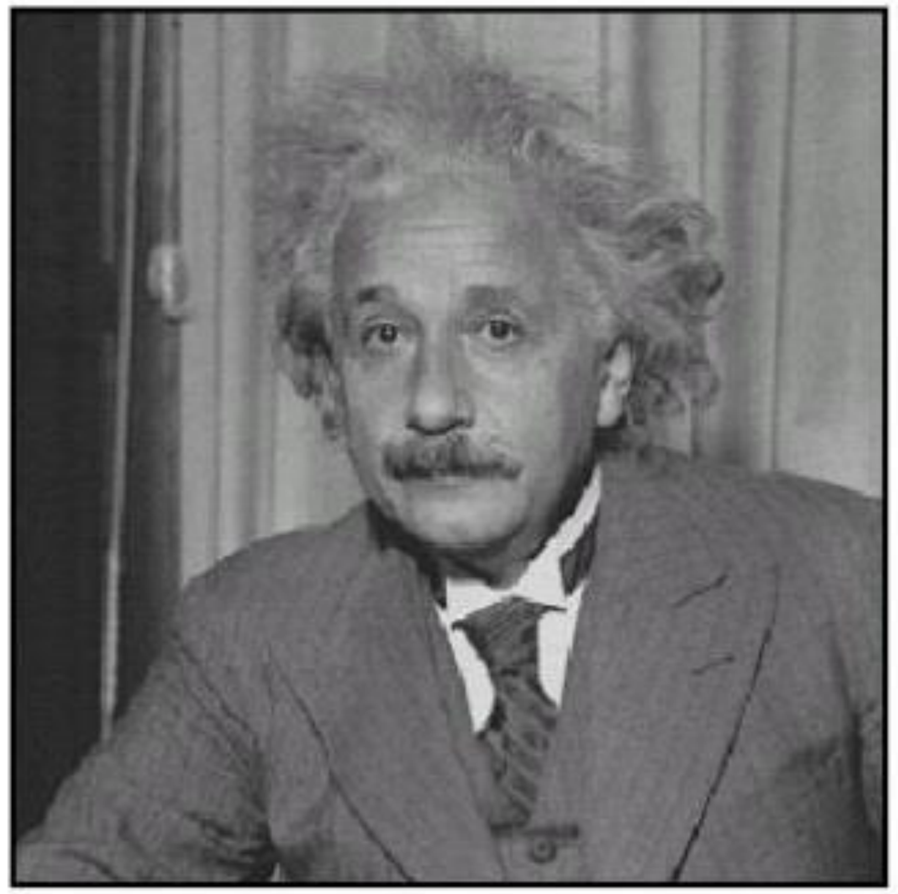
Sharpening filter

- Accentuates differences with local average

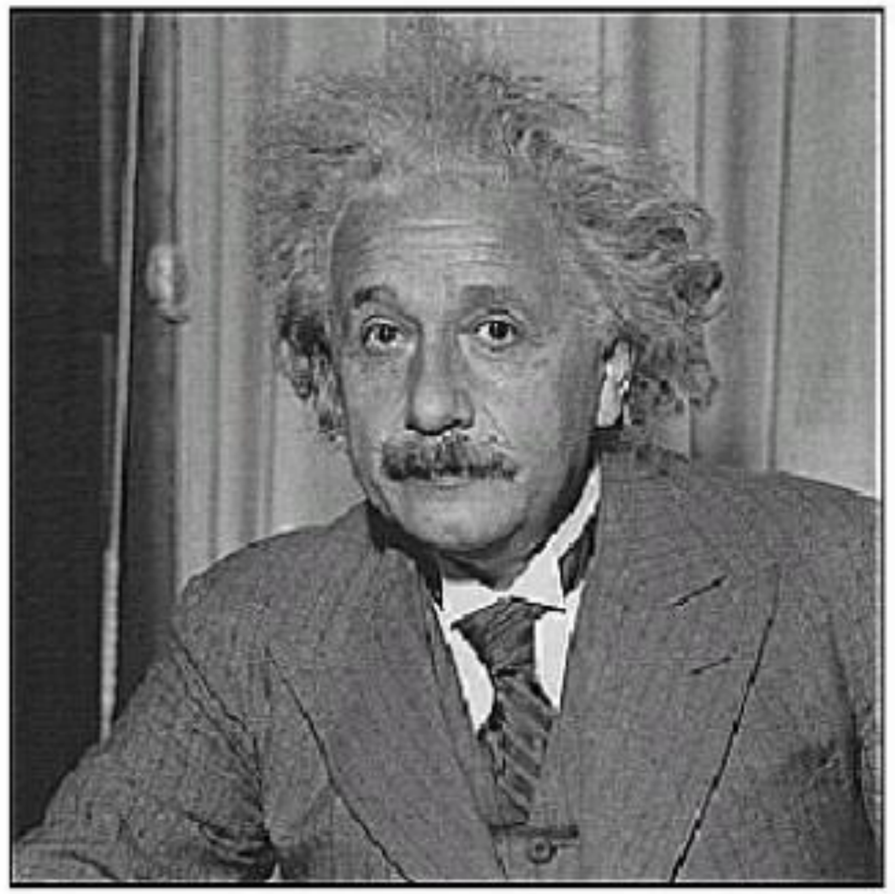
Source: D. Lowe

University at Buffalo
The State University of New York

Image filtering - Sharpening



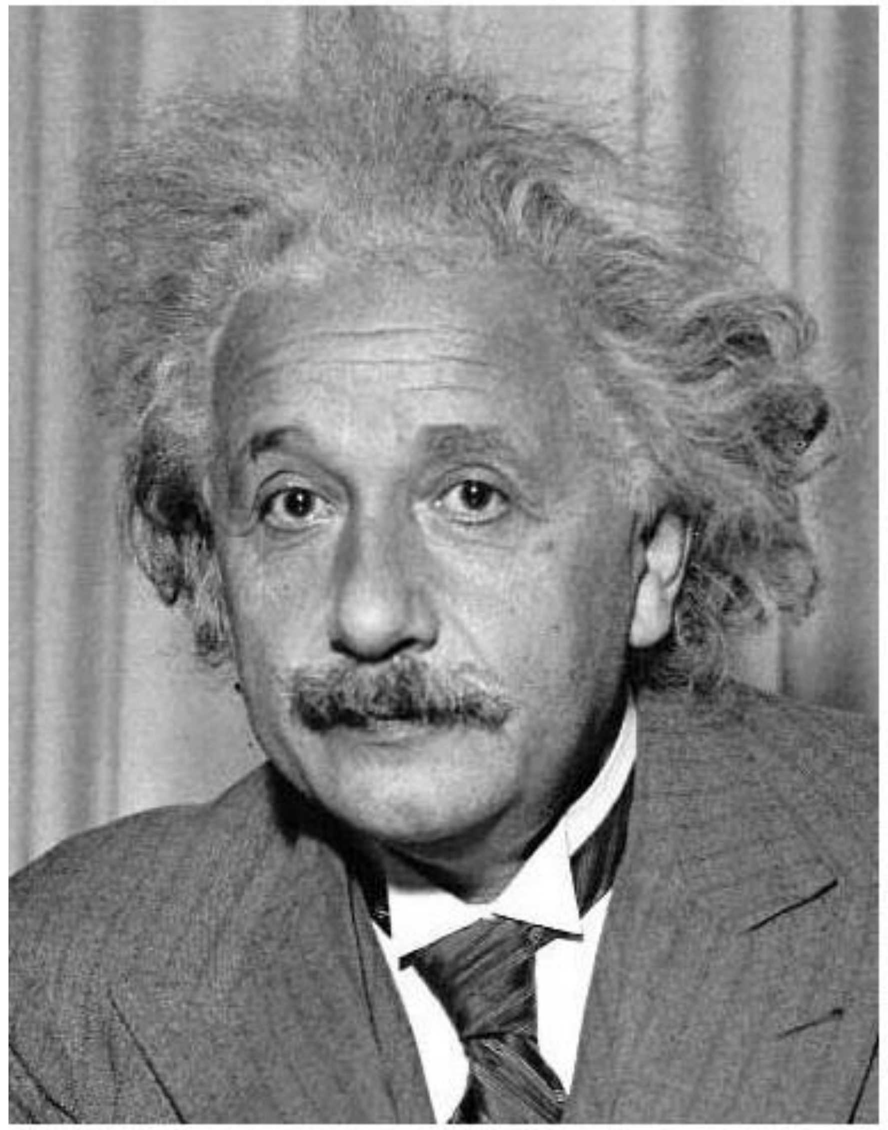
before



after

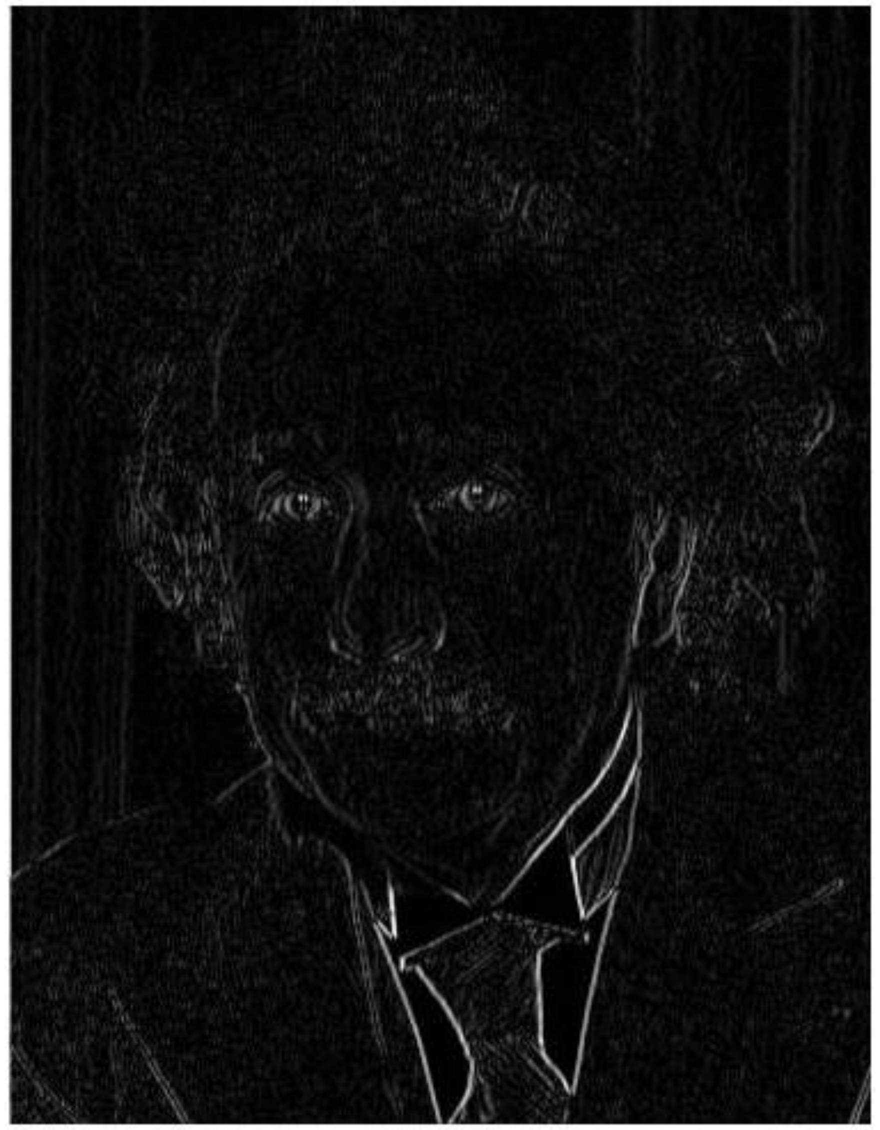
Source: D. Lowe

Image filtering - Edges



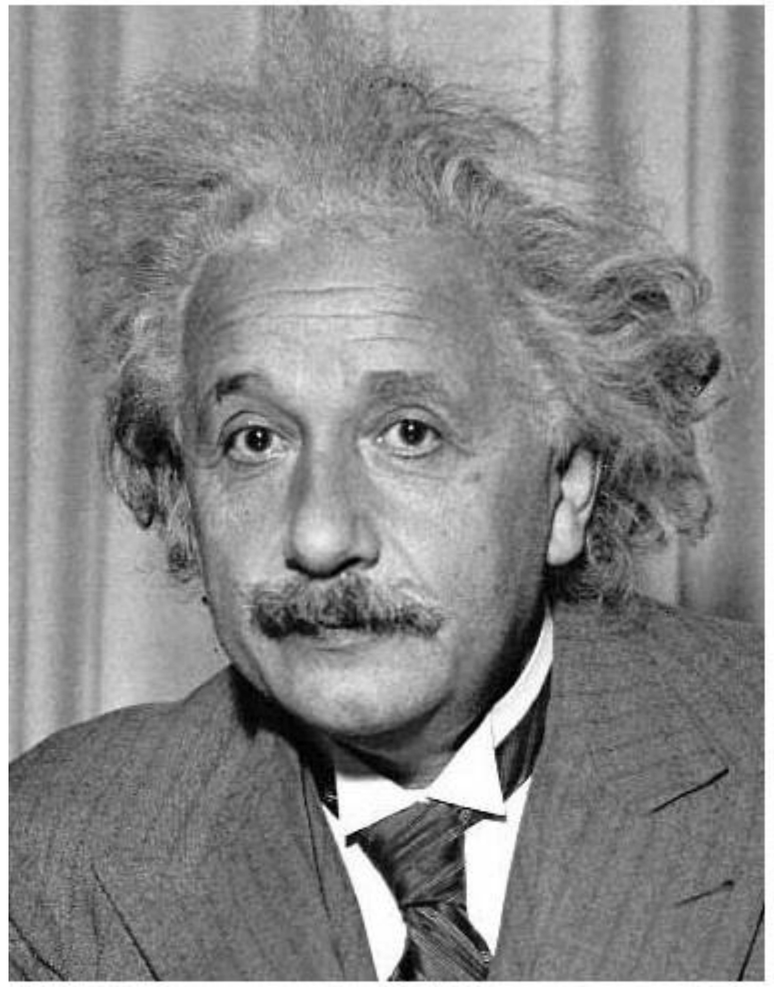
1	0	-1
2	0	-2
1	0	-1

Sobel



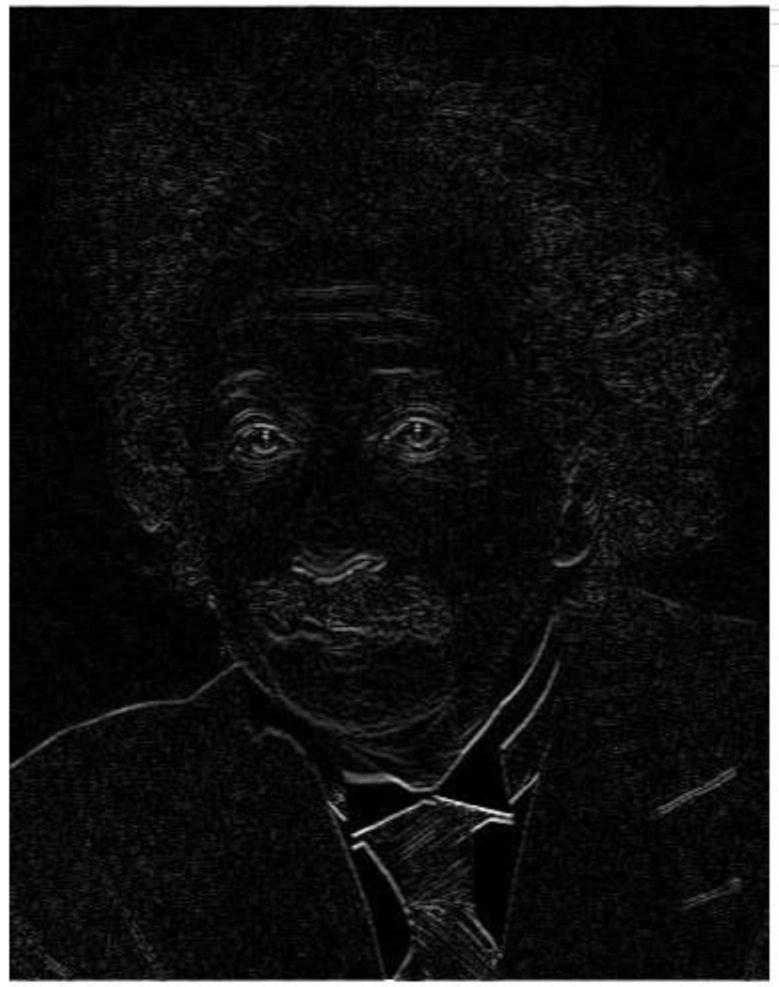
Vertical Edge
(absolute value)

Image filtering - Edges



1	2	1
0	0	0
-1	-2	-1

Sobel



Horizontal Edge
(absolute value)

Image filters - Separability

Are you familiar with Matric outer product?

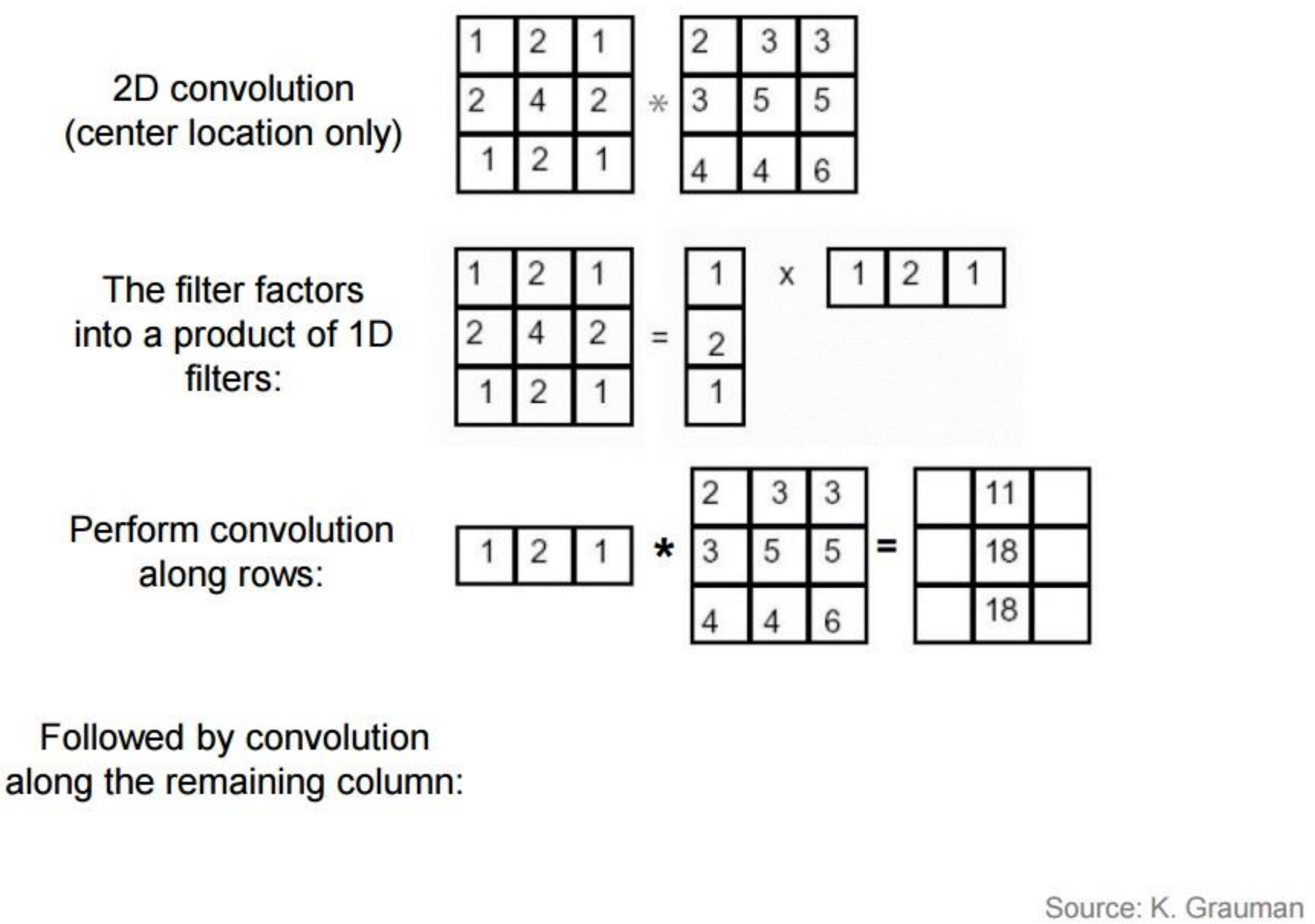
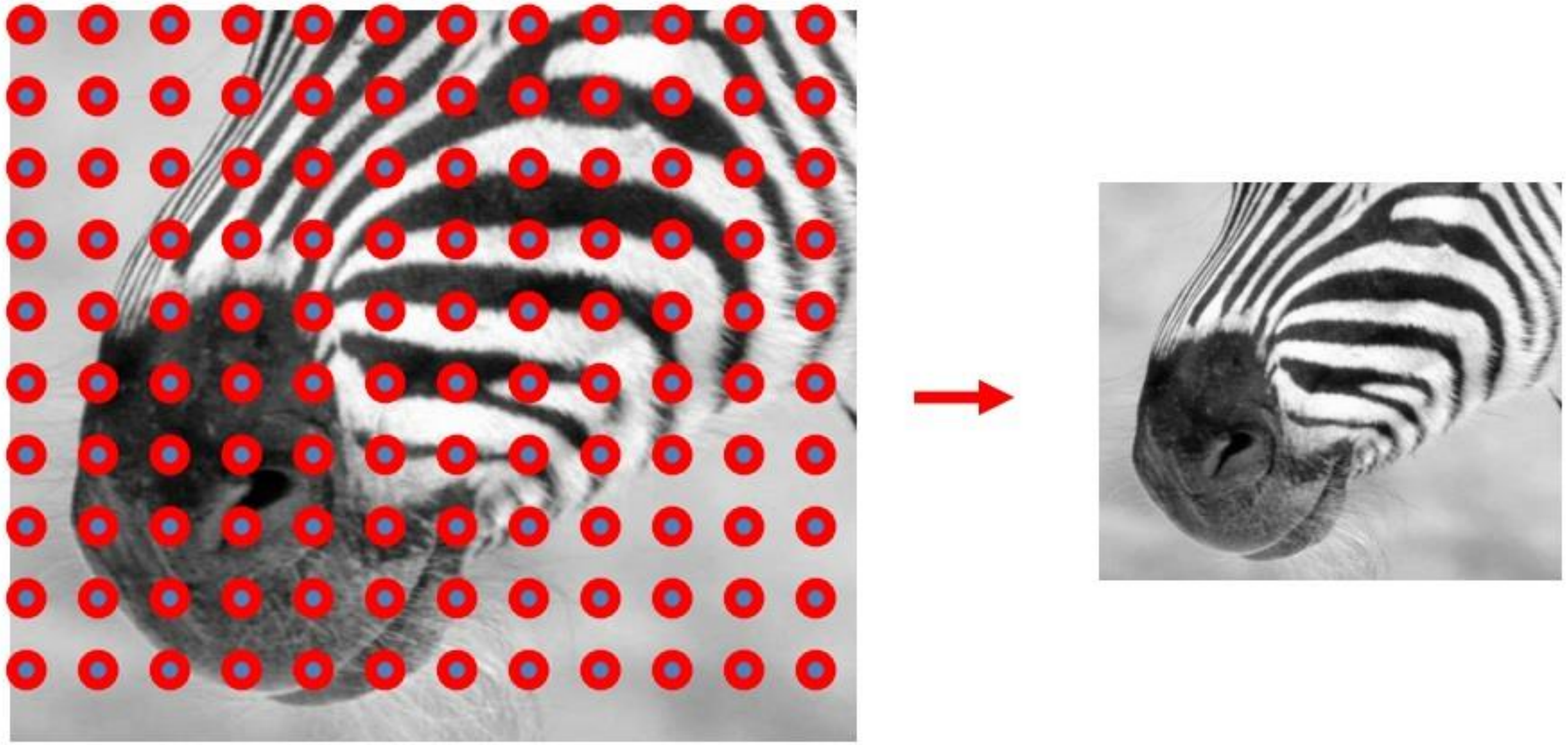


Image Resizing

Why does a lower resolution image still make sense to us? What do we lose?

Image Resizing – Basic Approach

Do you see any potential Issues?

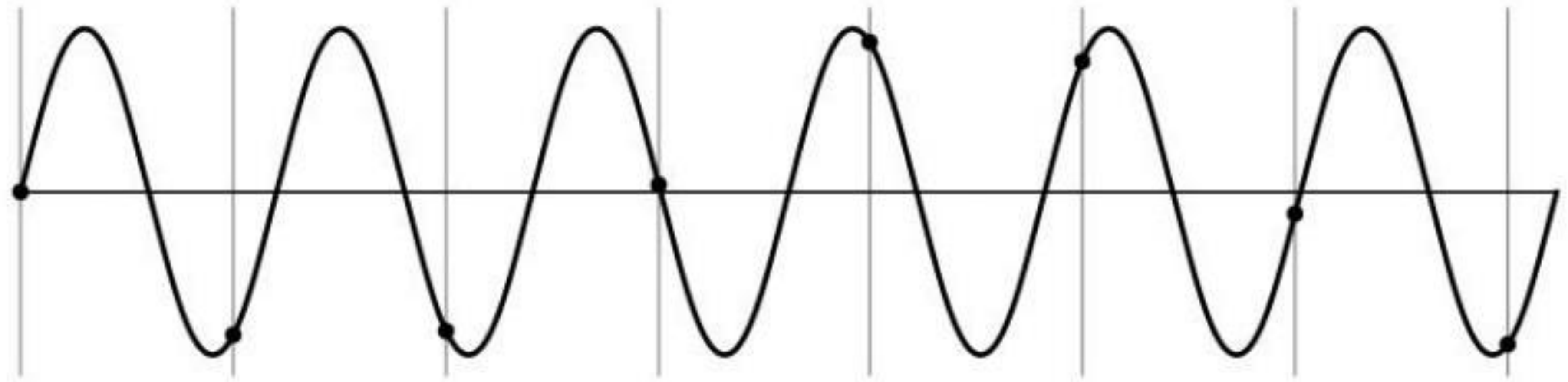


Throw away every other row and column to create a 1/2 size image

University at Buffalo
The State University of New York

Image Resizing – Aliasing

- 1D example (sinewave):

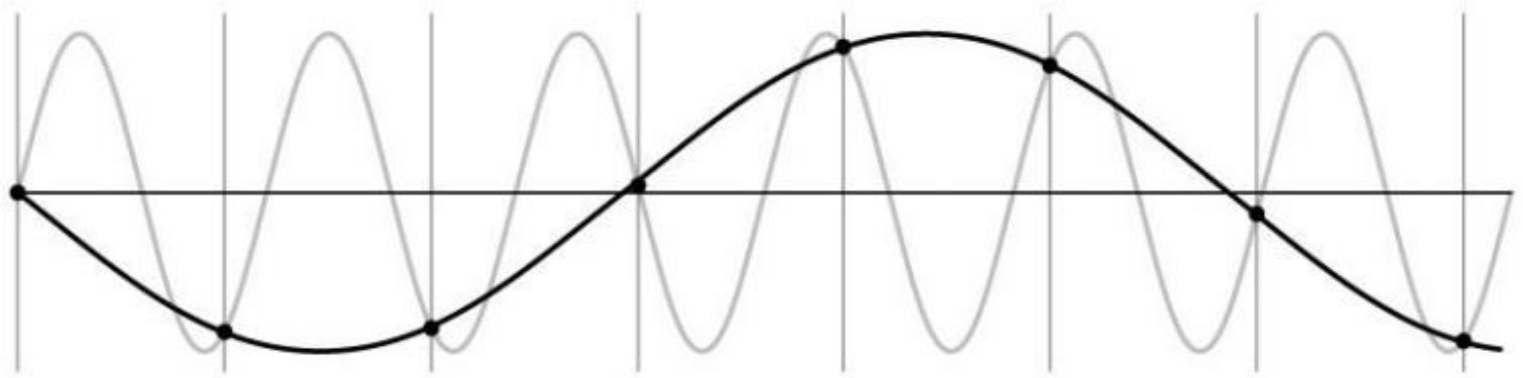


Source: S. Marschner

Image Resizing – Aliasing

Reconstructed signal not the same as original!

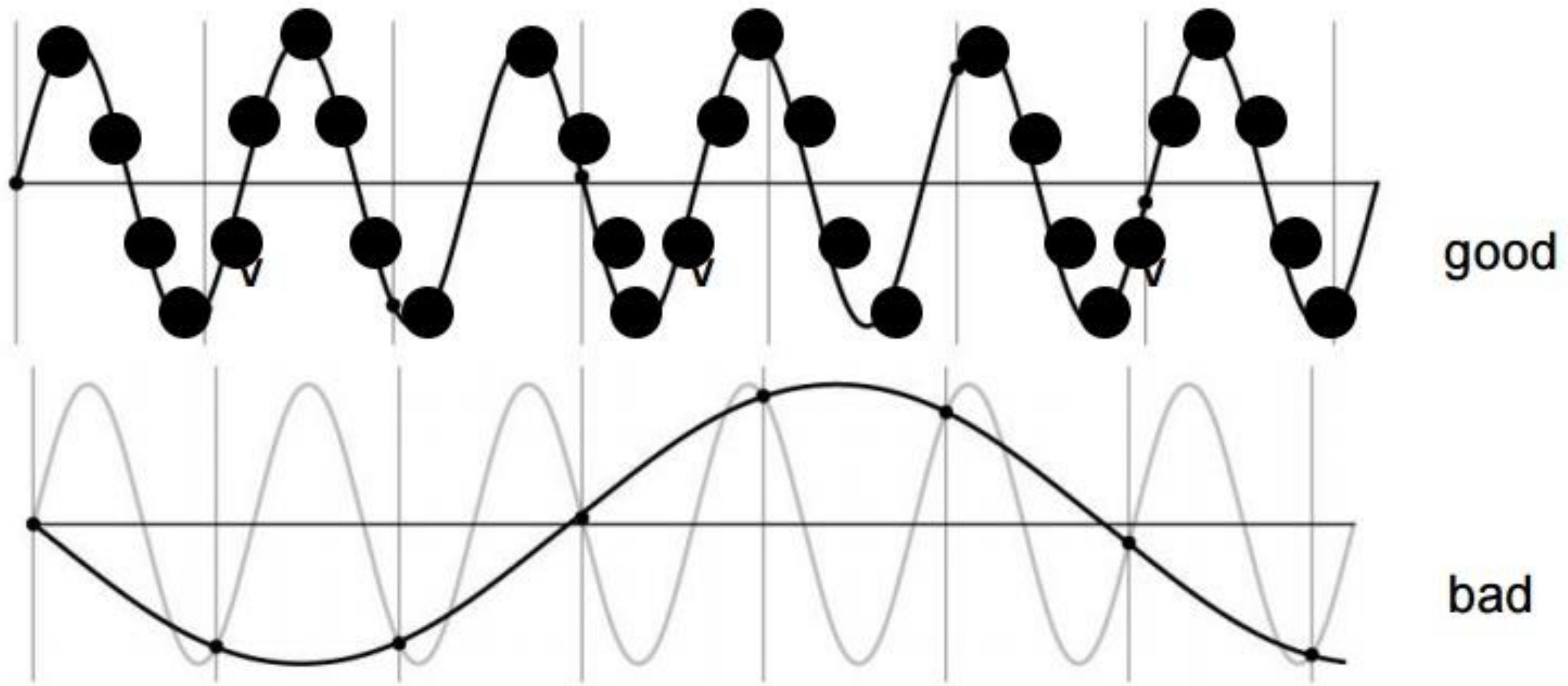
- 1D example (sinewave):



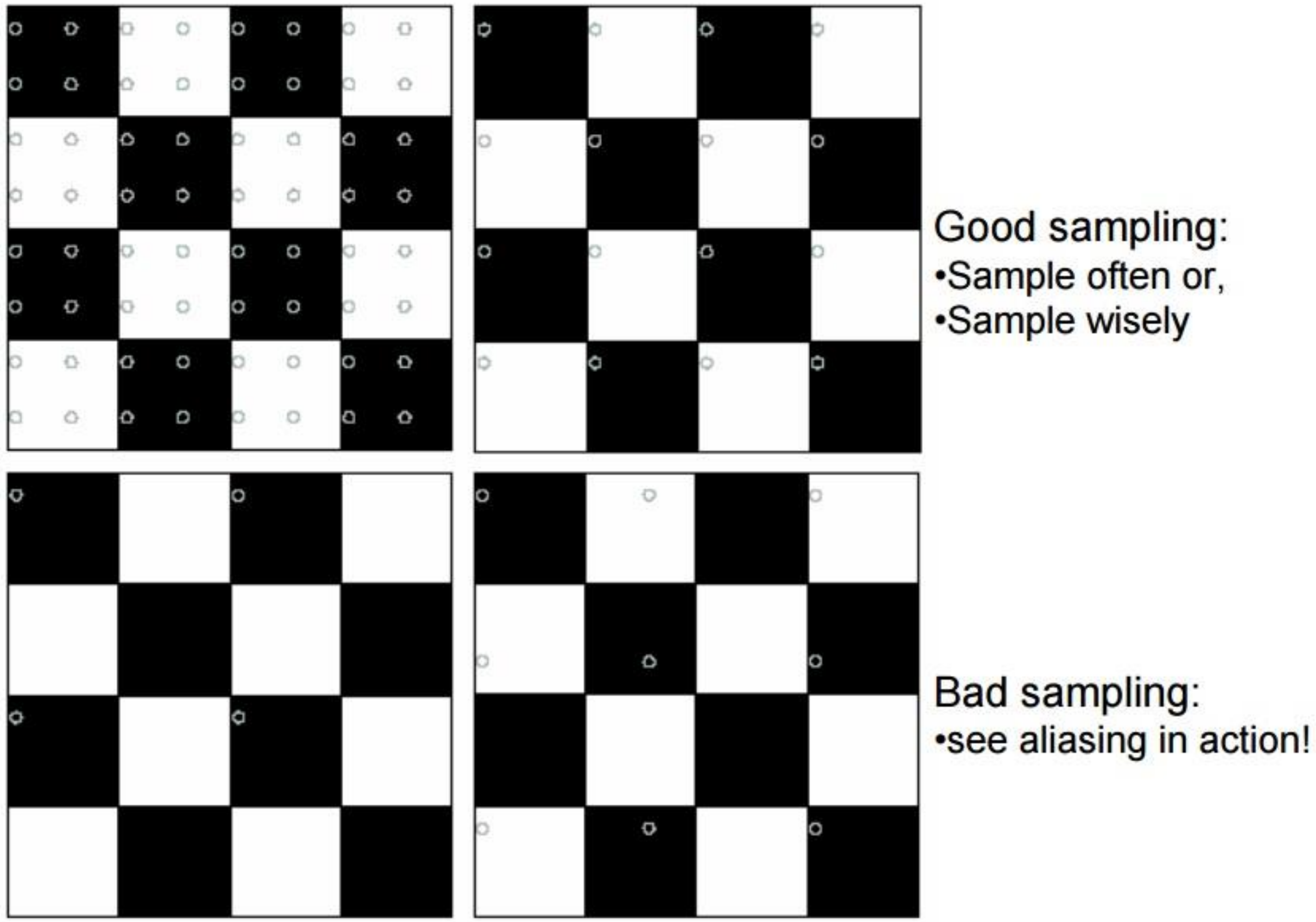
Source: S. Marschner

Image Resizing – Aliasing

- When sampling a signal at discrete intervals, the sampling frequency must be $\geq 2 \times f_{max}$
- f_{max} = max frequency of the input signal
- This will allows to reconstruct the original perfectly from the sampled version



Sampling


Good sampling:
•Sample often or,
•Sample wisely
Bad sampling:
•see aliasing in action!

How to Subsample Images

Solutions:

- Sample more often

- Get rid of all frequencies that are greater than half the new sampling frequency
  - Will lose information
  - But it's better than aliasing
  - Apply a smoothing filter

Image Sub-Sampling Pipeline

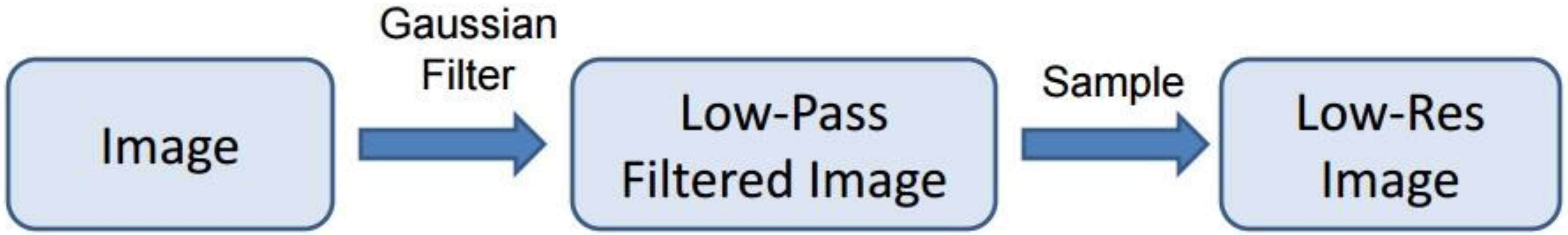


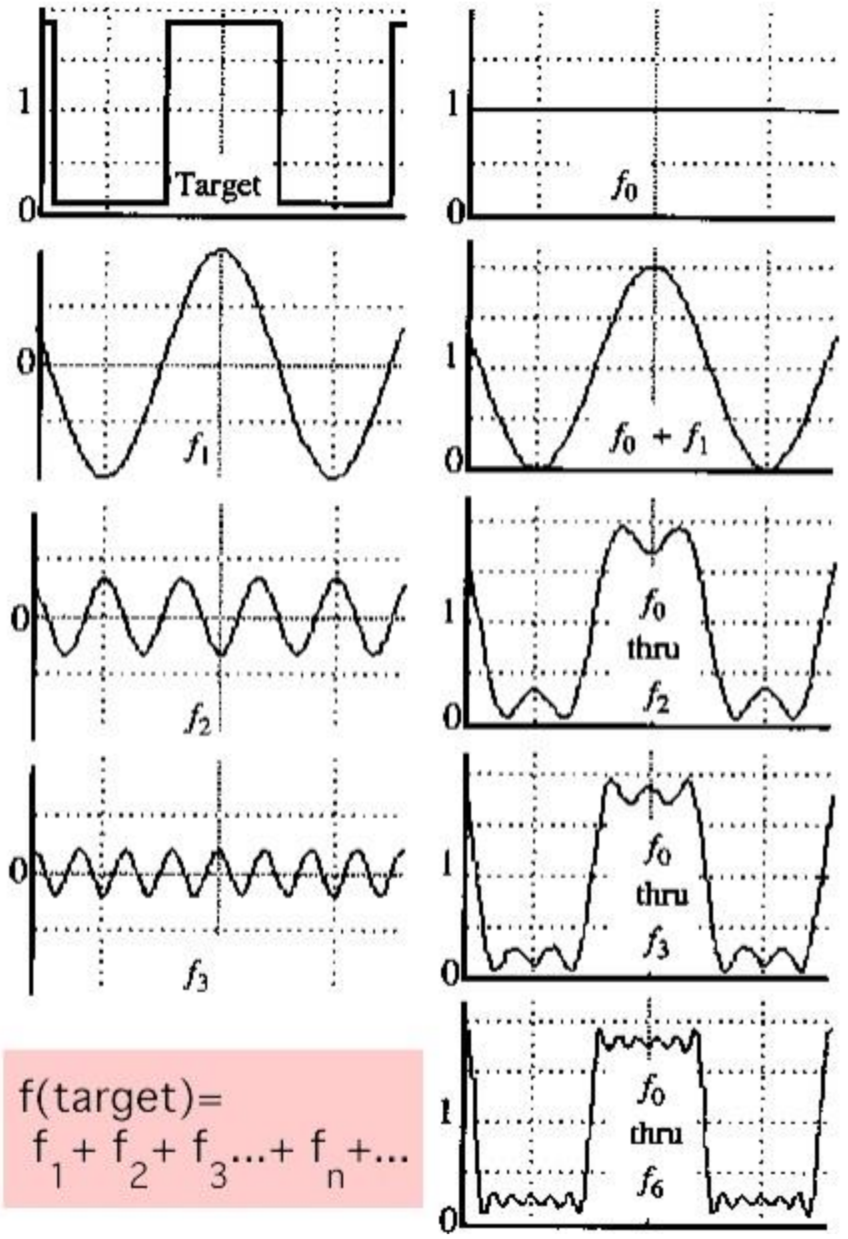
Image Fourier Analysis

Any signal can be approximate by a summation of sinusoids!

Our building block:

$$A \sin(\omega x + \phi)$$

Add enough of them to get any signal *g(x)* you want!

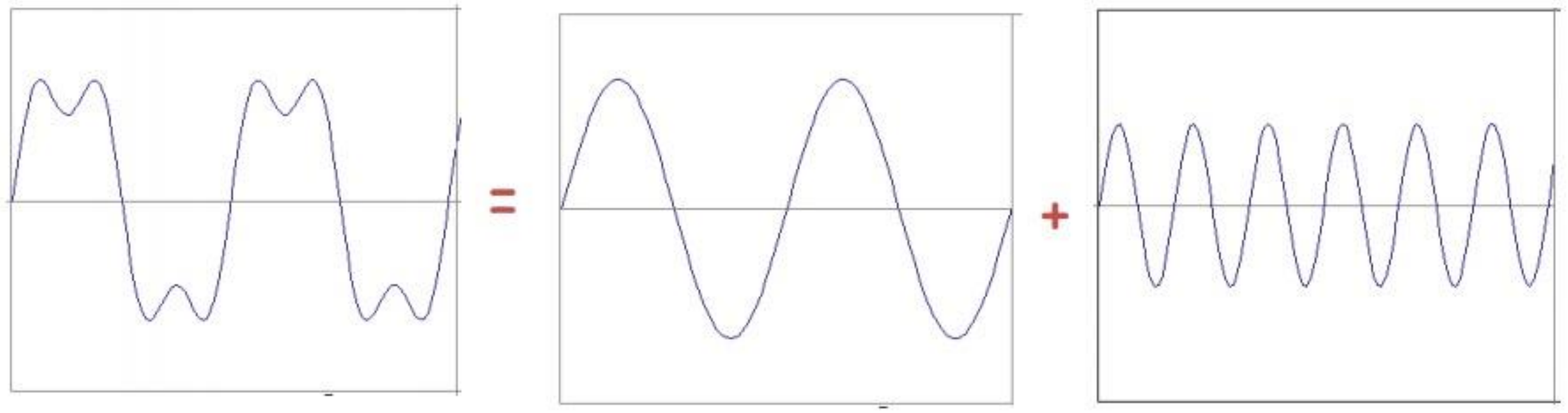


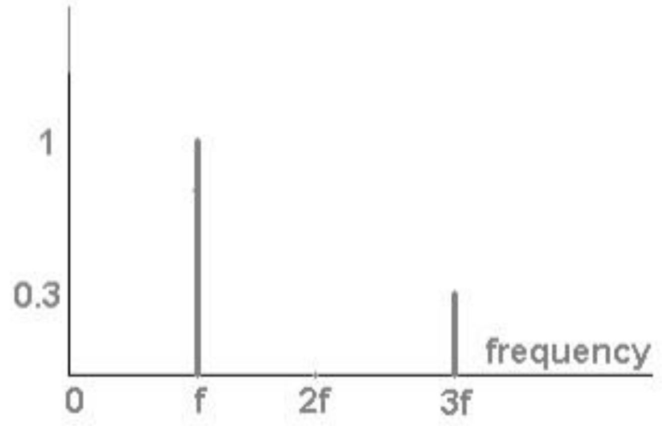
f(target)=
$f_1 + f_2 + f_3 ... + f_n + ...$

University at Buffalo
The State University of New York

Image Fourier Analysis

- example : $g(t) = \sin(2\pi f\, t) + (1/3)\sin(2\pi(3f)\, t)$





Slides: Efros

Image Fourier Analysis

Example: Music

- We think of music in terms of frequencies at different magnitudes

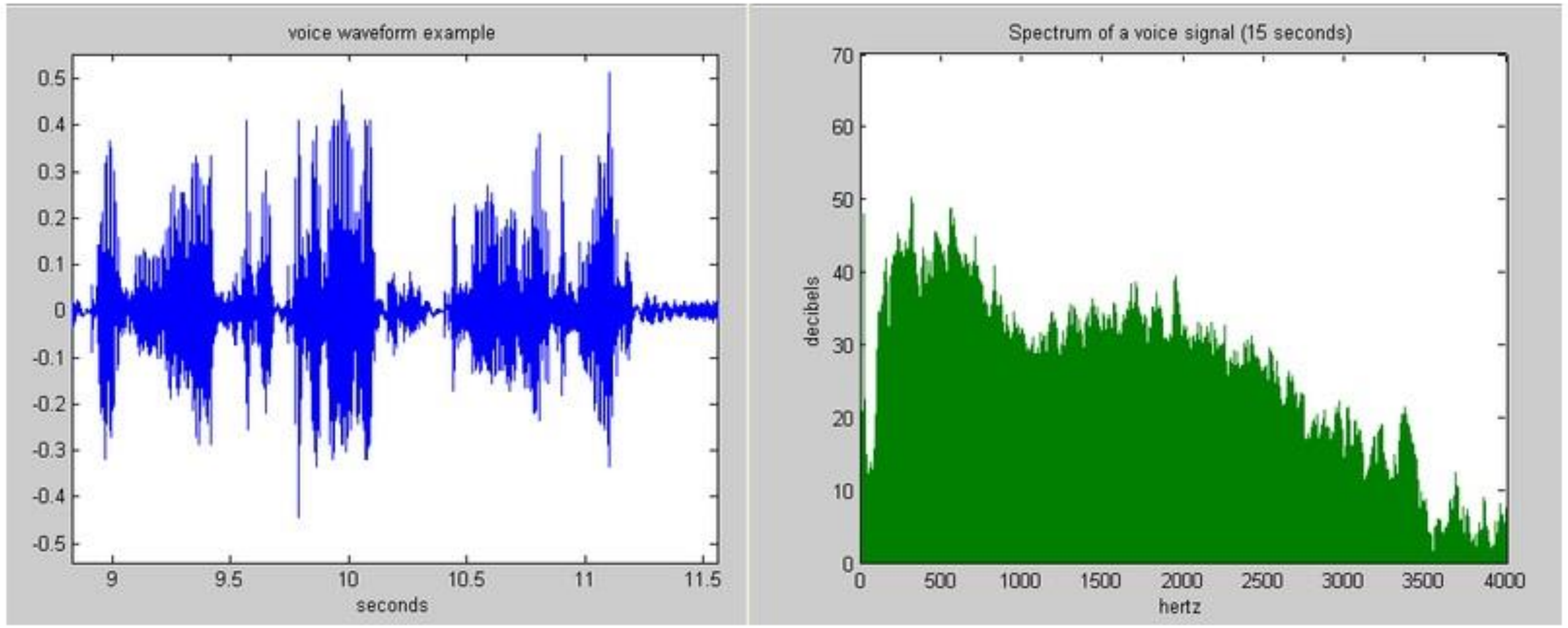
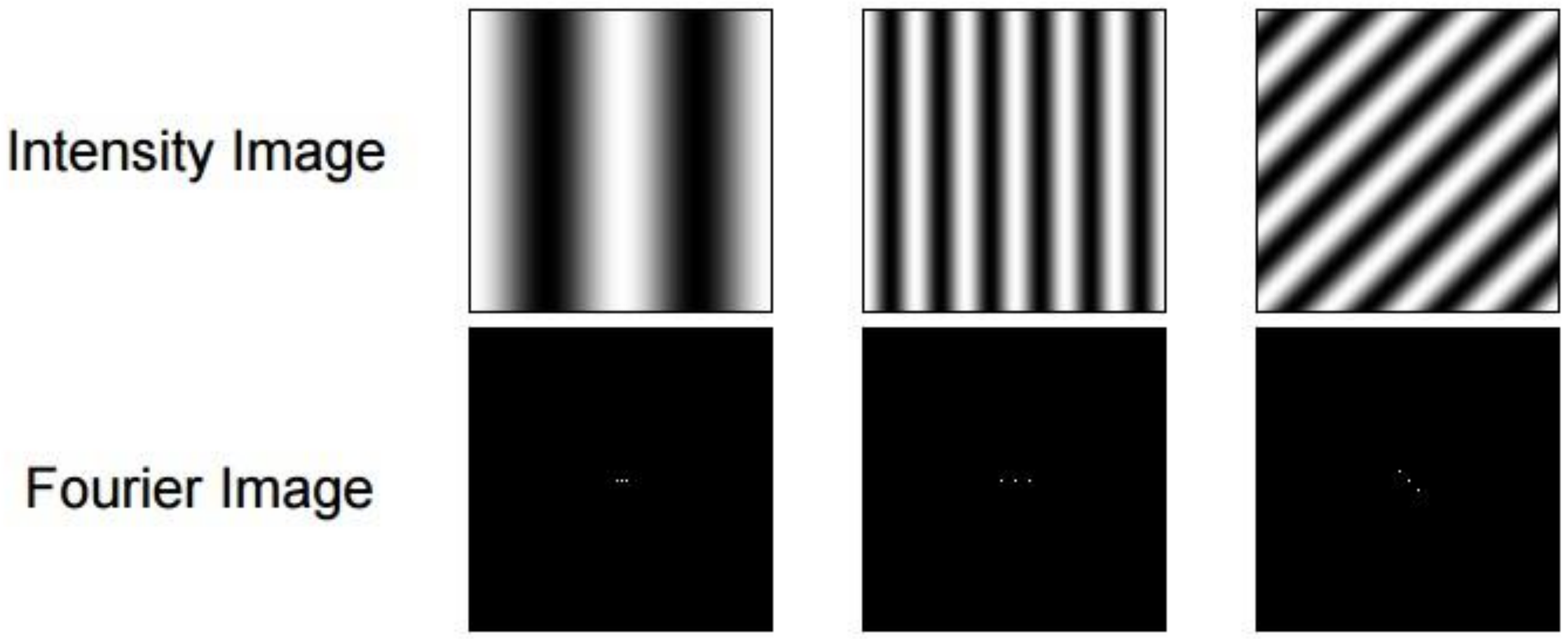


Image Fourier Analysis



http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering

Image Fourier Analysis - Convolution

- The Fourier transform of the convolution of two functions is the product of their Fourier transforms

$$F[g * h] = F[g]F[h]$$

- **Convolution** in spatial domain is equivalent to **multiplication** in frequency domain!

$$g * h = F^{-1}[F[g]F[h]]$$

Image Fourier Analysis - Advantage

Too many spatial filter operations! What is the computational complexity of convolution?

Filtering in spatial domain

1	0	-1
2	0	-2
1	0	-1

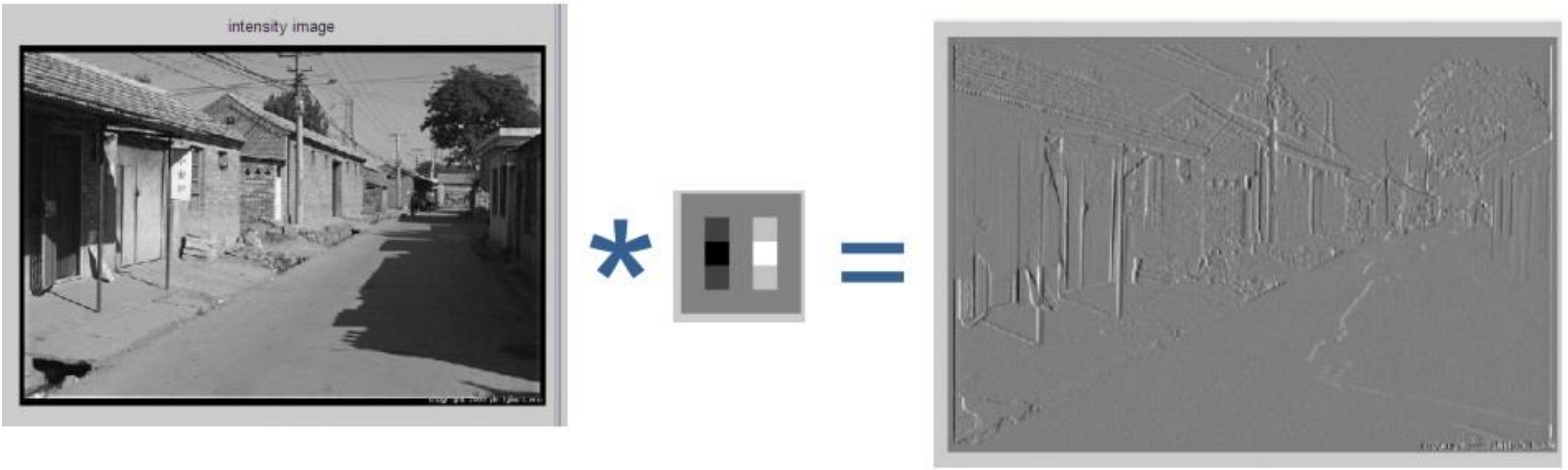


Image Fourier Analysis - Advantage

In frequency domain – Convolution becomes multiplication

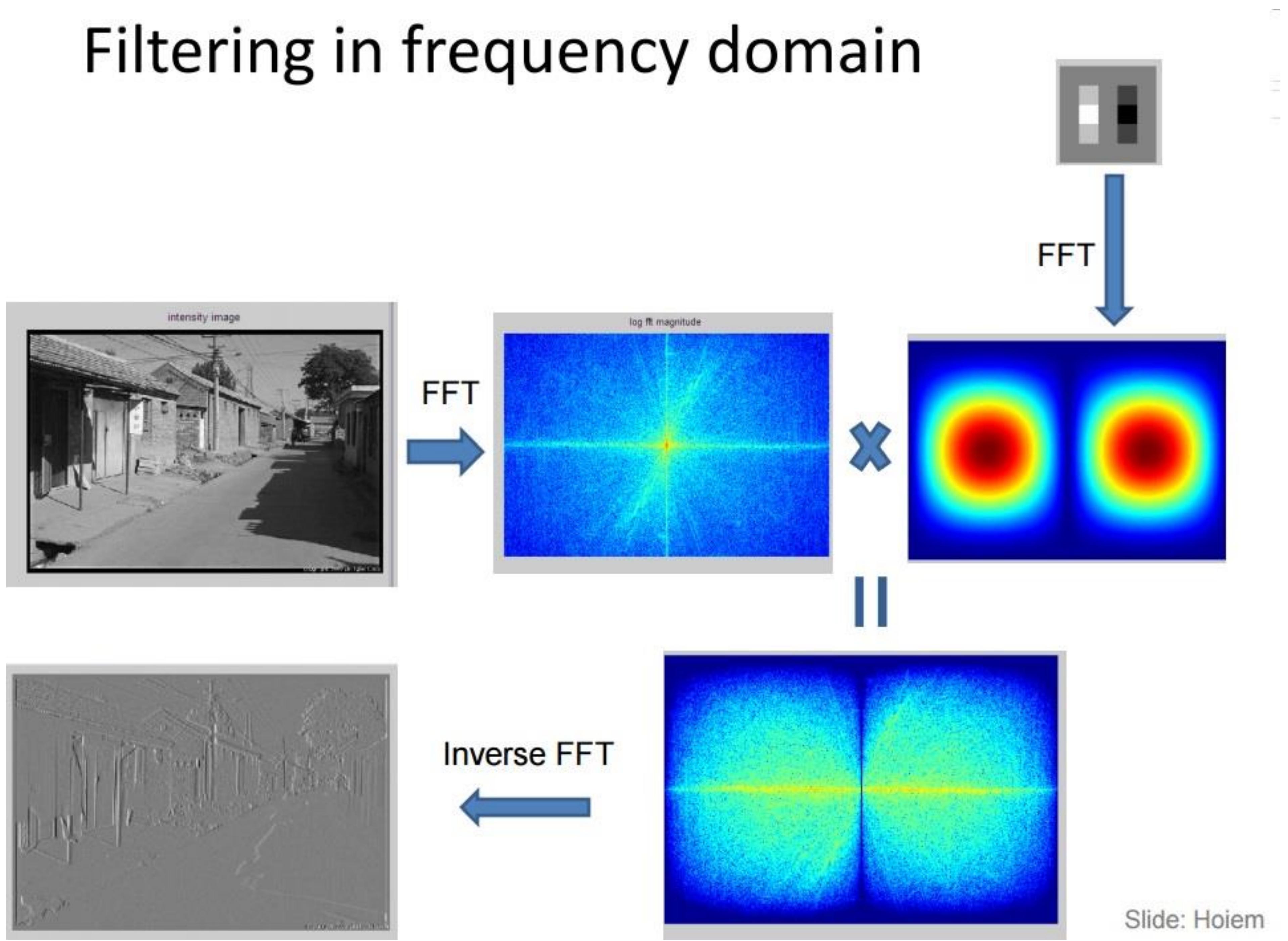
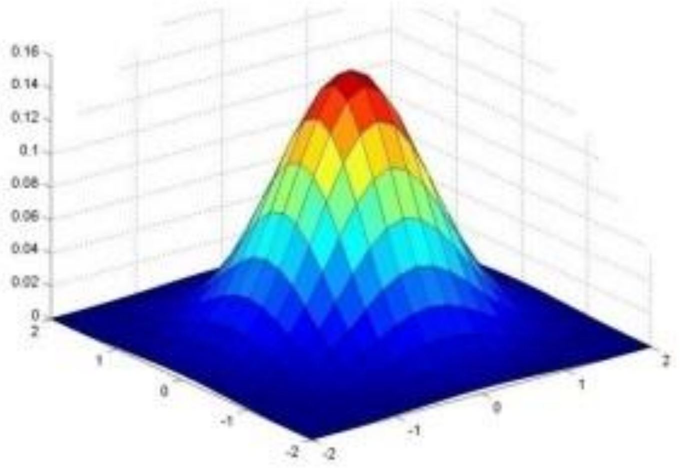
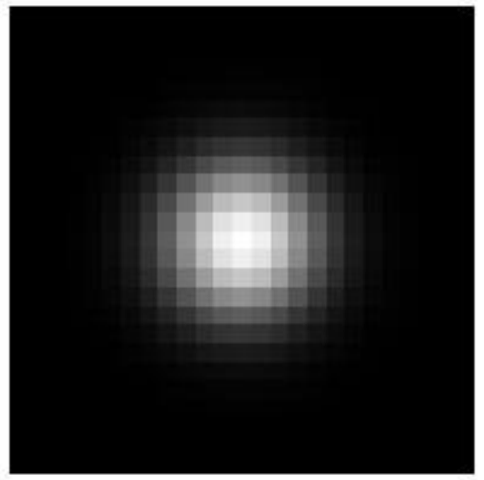


Image Smoothing – Gaussian Kernel





$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Source: C. Rasmussen

Image Smoothing – Gaussian Kernel

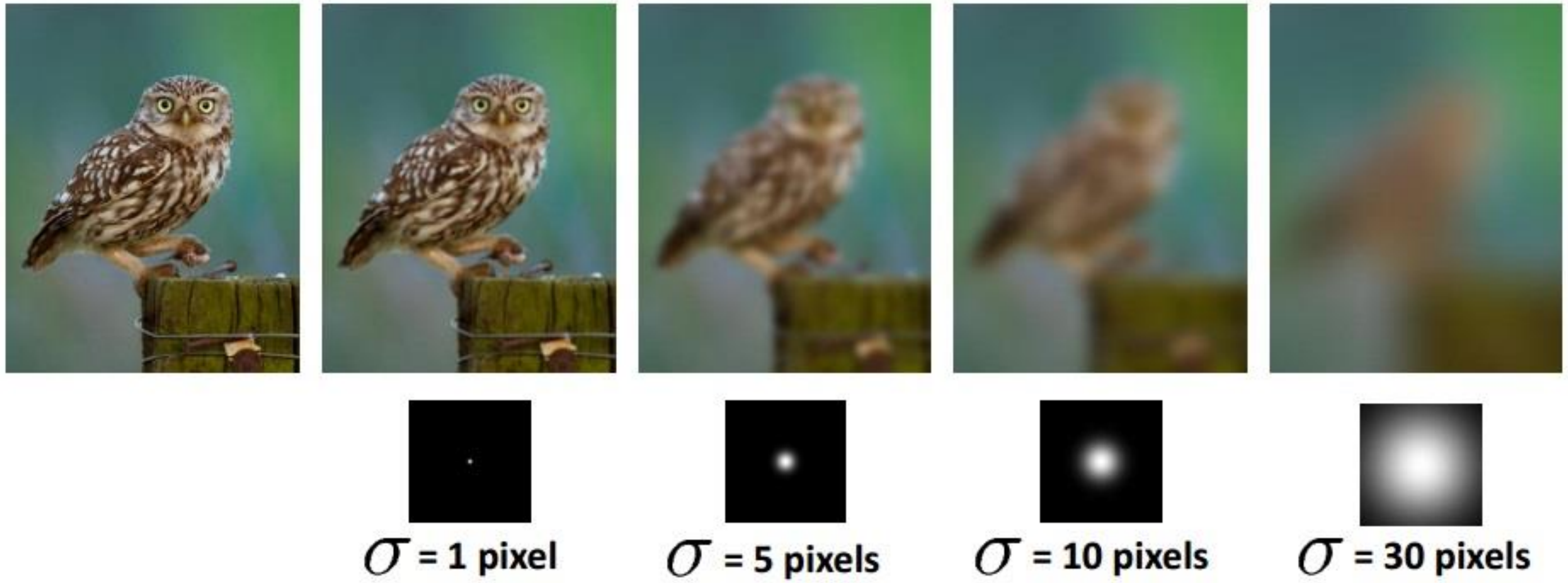


Image Smoothing – Gaussian Kernel

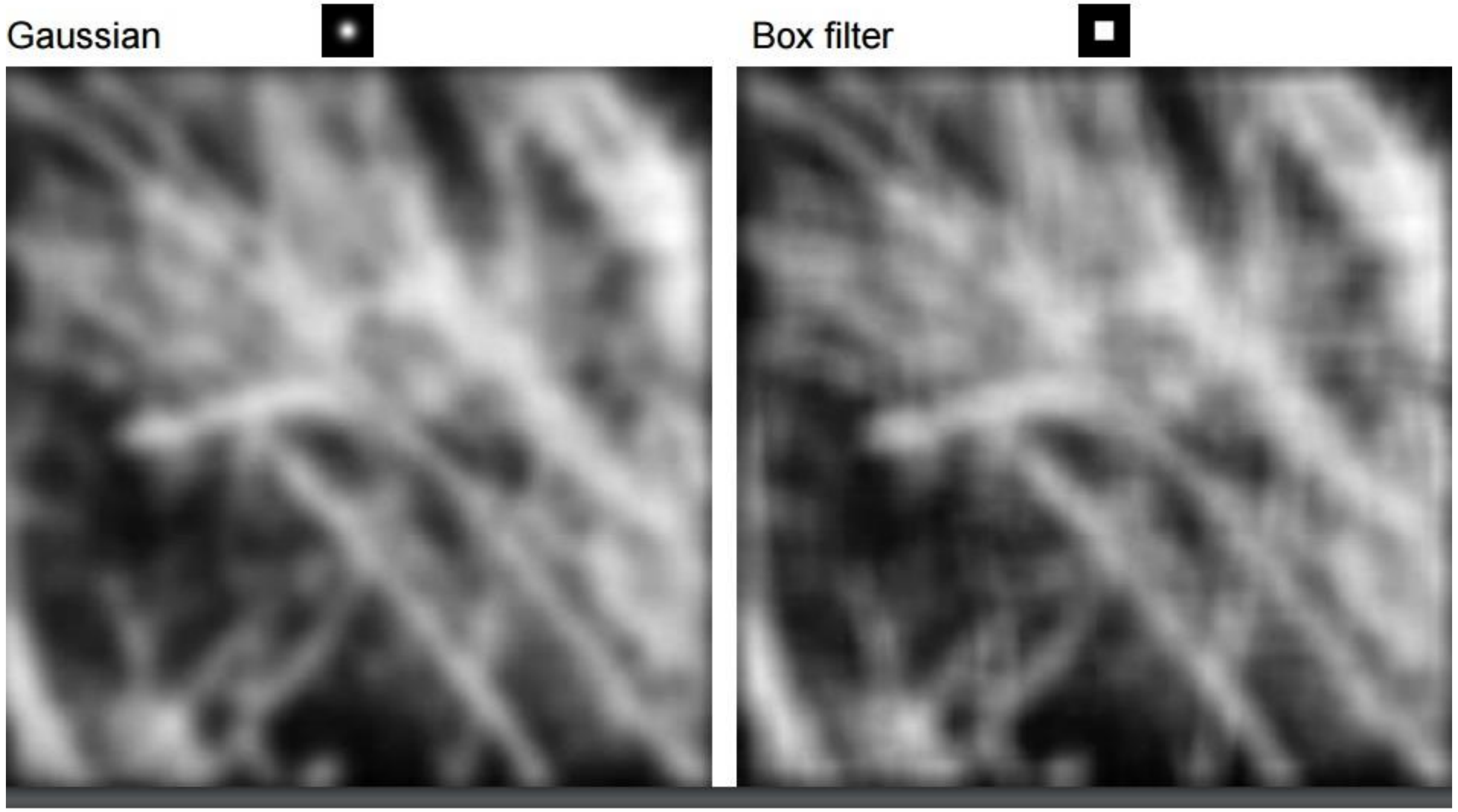
- Removes "high-frequency" components from the image (low-pass filter)
- Convolution with self is another Gaussian

  - Convolving two times with Gaussian kernel of width σ = convolving once with kernel of width $\sigma\sqrt{2}$

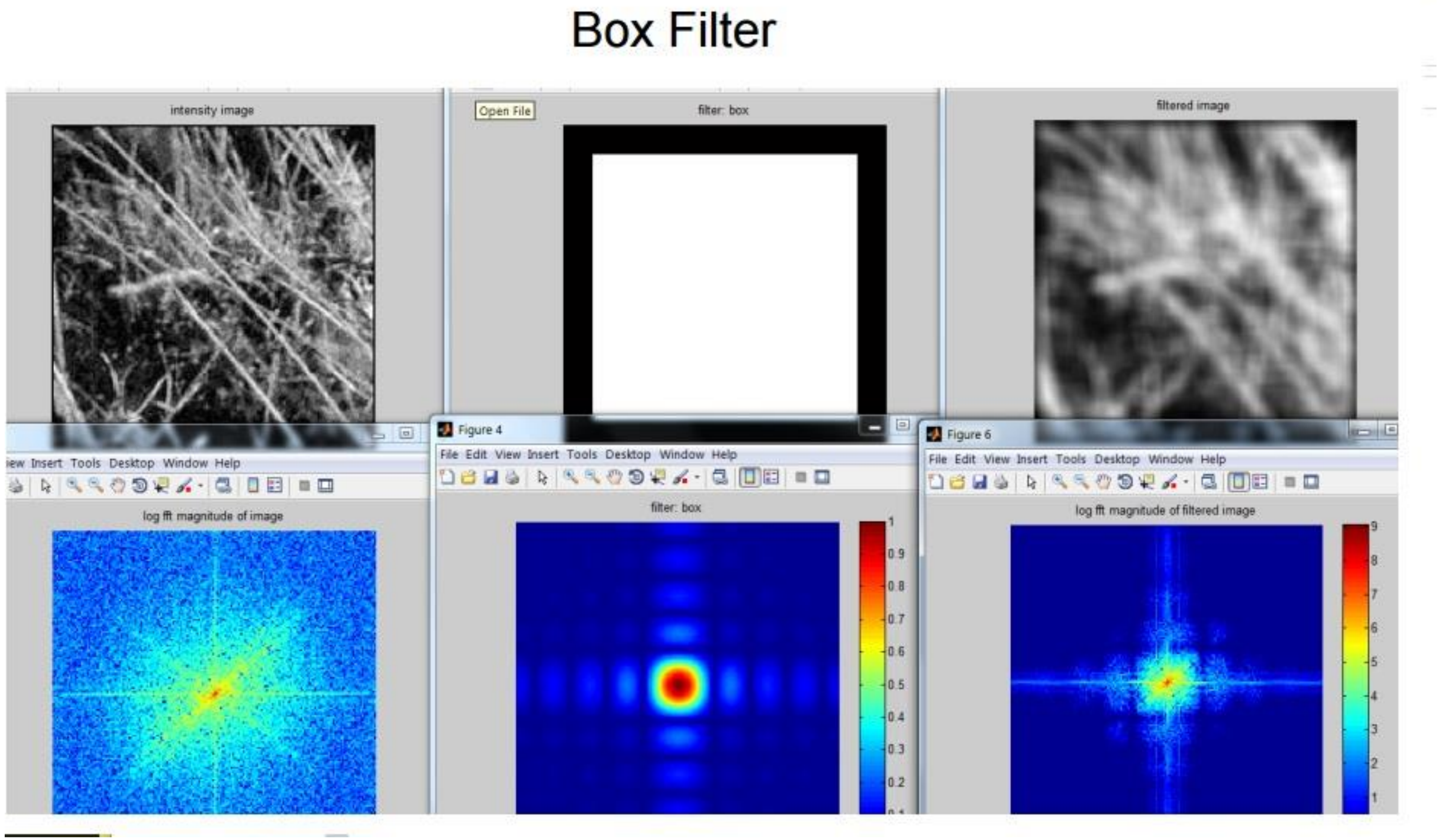
Source: K. Grauman

Box Filter Vs Gaussian Filter

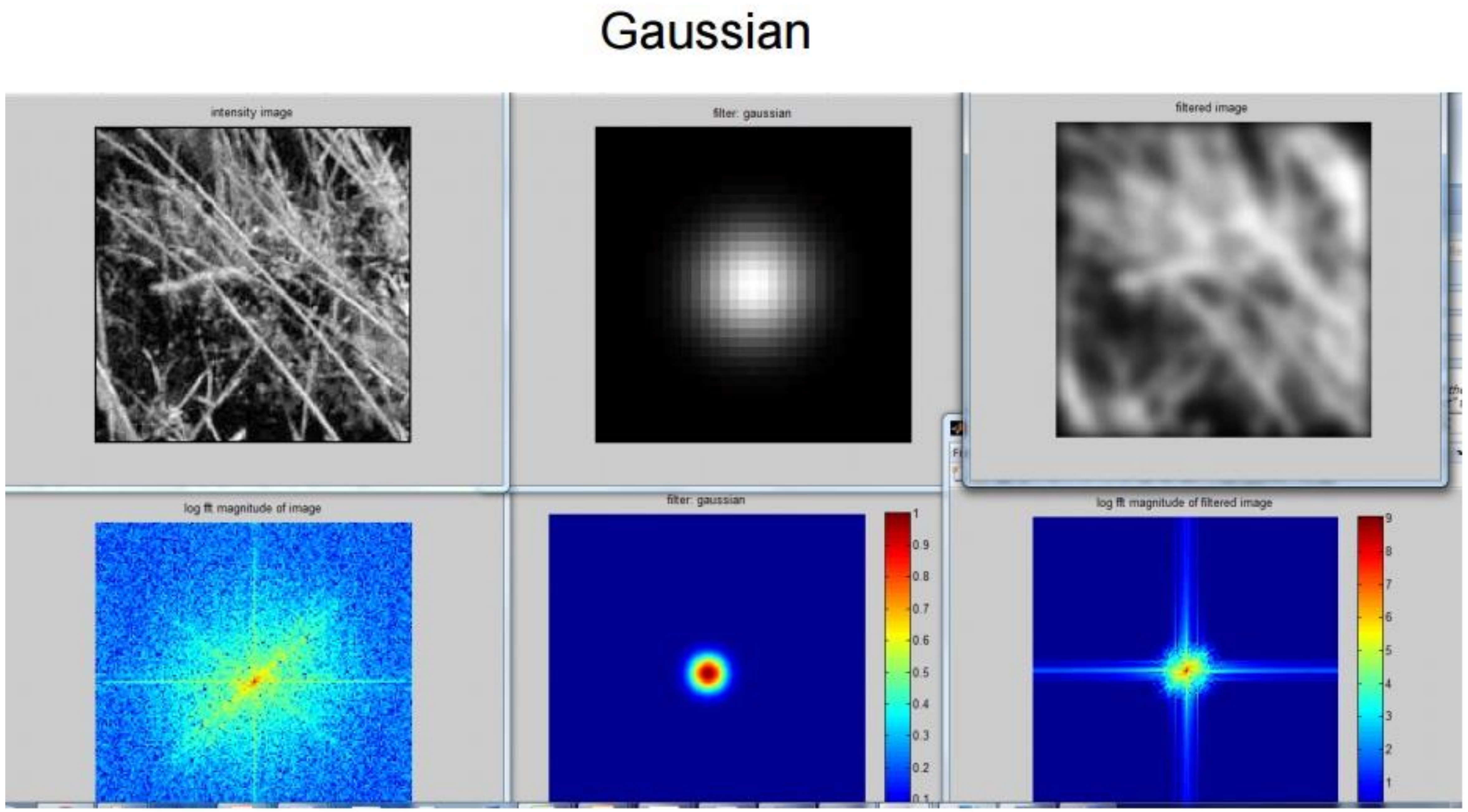
Why does the Gaussian give a nice smooth image, but the square filter give edgy artifacts?



Fourier Analysis – Box Filter

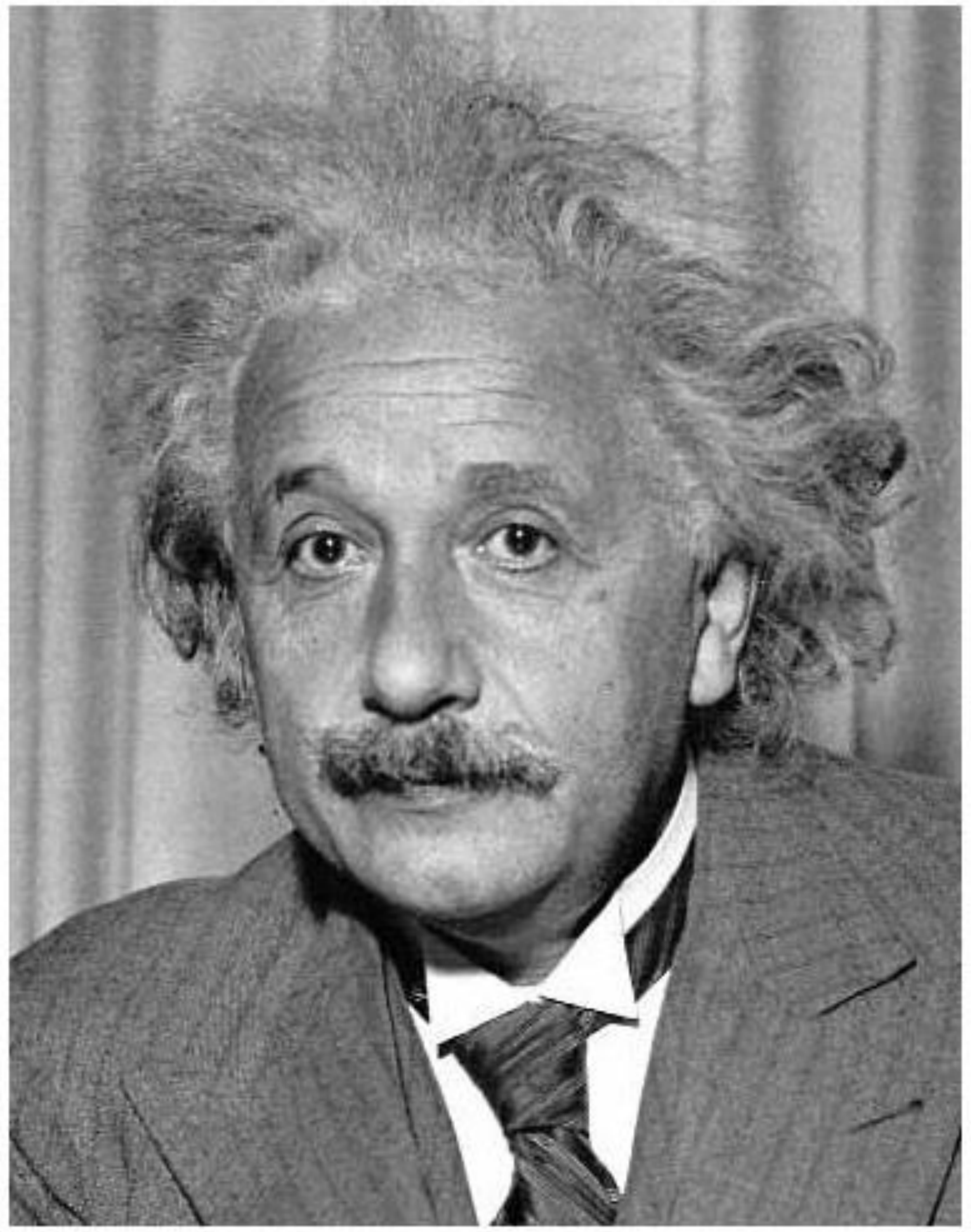


Fourier Analysis – Gaussian Filter



Template Matching

- Goal: find  in image
- Main challenge: What is a good similarity or distance measure between two patches?
  - Correlation
  - Zero-mean correlation
  - Sum Square Difference
  - Normalized Cross Correlation



Sum of Squared Differences (SSD)

- Goal: find  in image
- Method 2: SSD

$$h[m,n] = \sum_{k,l} (g[k,l] - f[m+k, n+l])^2$$

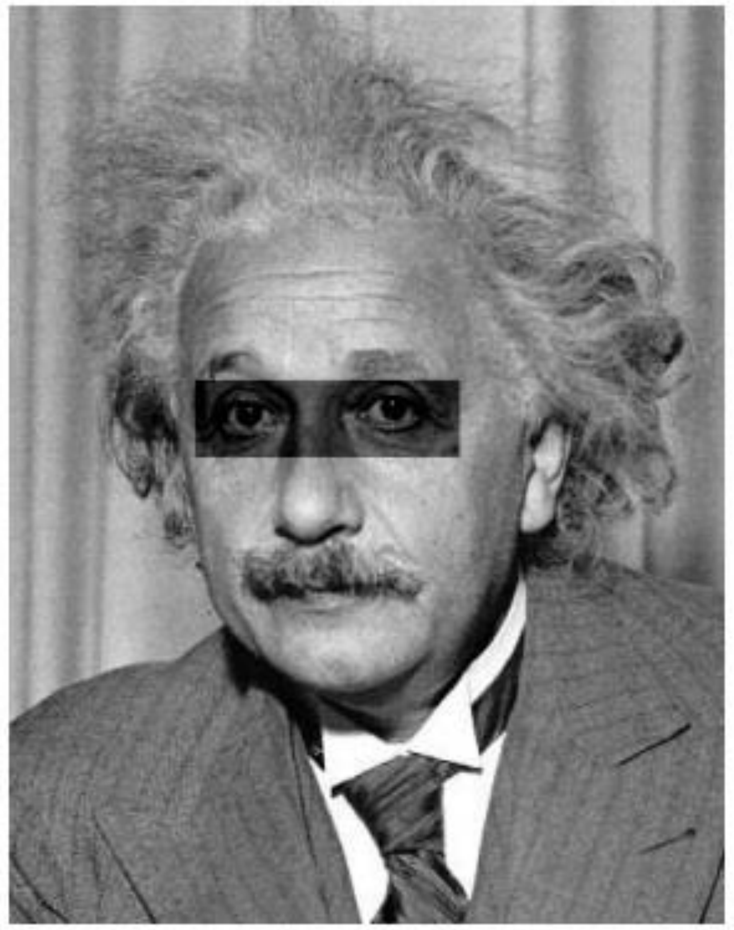
Input | 1- sqrt(SSD) | Thresholded Image

Sum of Squared Differences (SSD)

- Goal: find  in image
- Method 2: SSD

What's the potential downside of SSD?

$$h[m,n]=\sum_{k,l}(g[k,l]-f[m+k,n+l])^2$$



Input

1- sqrt(SSD)

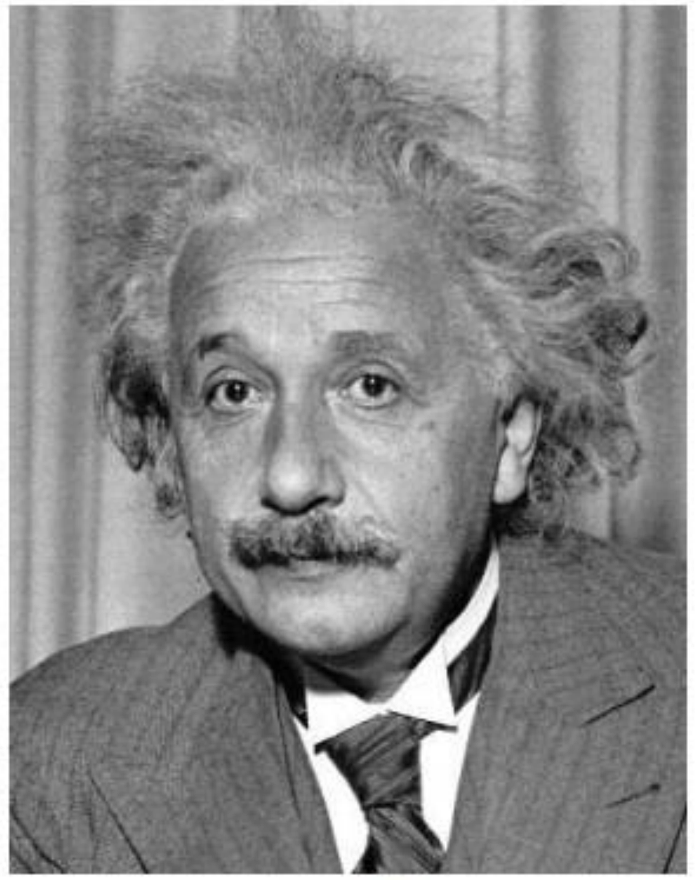
Normalized Cross Correlation

- Goal: find  in image
- Method 3: Normalized cross-correlation

$$h[m,n]=\frac{\sum_{k,l}(g[k,l]-\overset{\text{mean template}}{\bar{g}})(f[m-k,n-l]-\overset{\text{mean image patch}}{\bar{f}_{m,n}})}{\left(\sum_{k,l}(g[k,l]-\bar{g})^2\sum_{k,l}(f[m-k,n-l]-\bar{f}_{m,n})^2\right)^{0.5}}$$

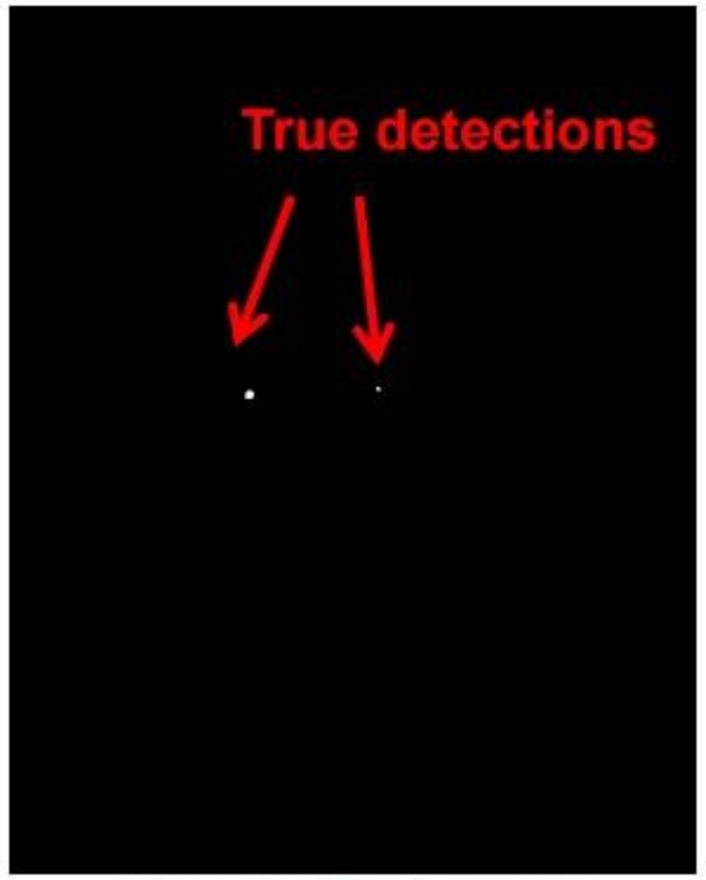
Normalized Cross Correlation

- Goal: find  in image
- Method 3: Normalized cross-correlation



Input

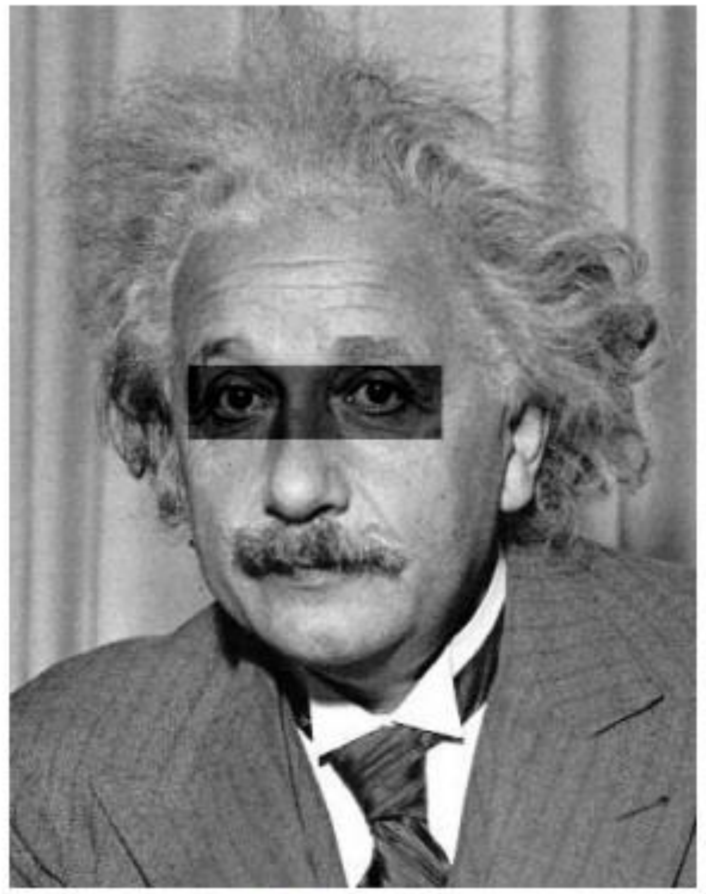
Normalized X-Correlation



Thresholded Image

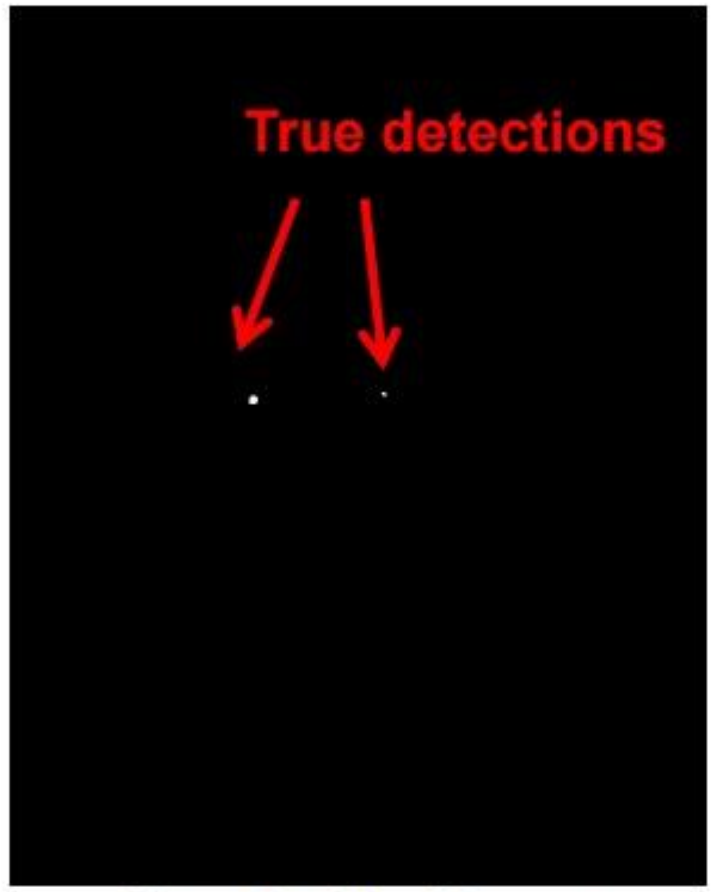
Normalized Cross Correlation

- Goal: find  in image
- Method 3: Normalized cross-correlation



Input

Normalized X-Correlation



Thresholded Image

SSD vs NCC – What to use?

A: Depends

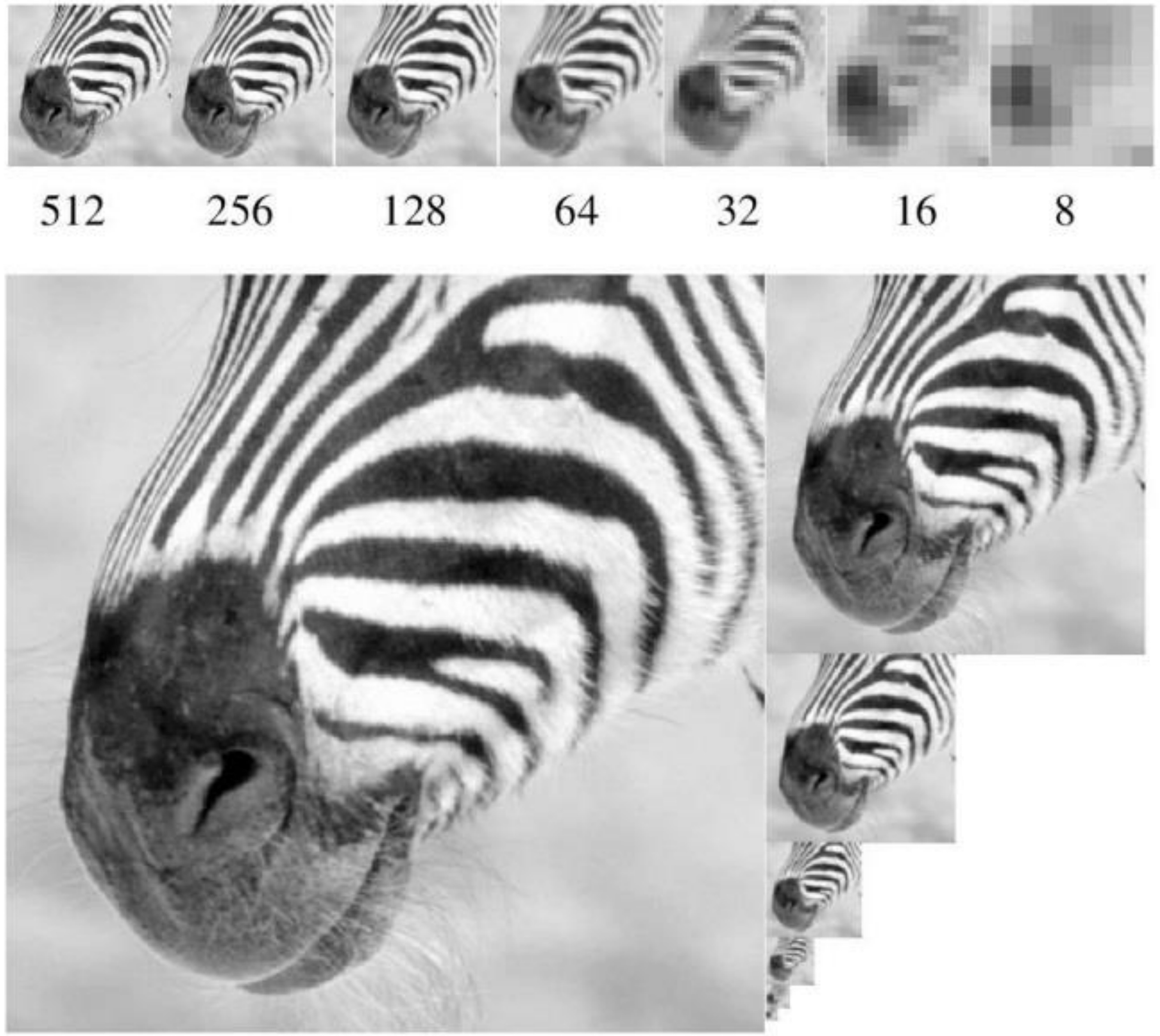
- SSD: faster, sensitive to overall intensity
- Normalized cross-correlation: slower, invariant to local average intensity and contrast

Image Pyramids

Q: What if we want to find larger or smaller eyes?

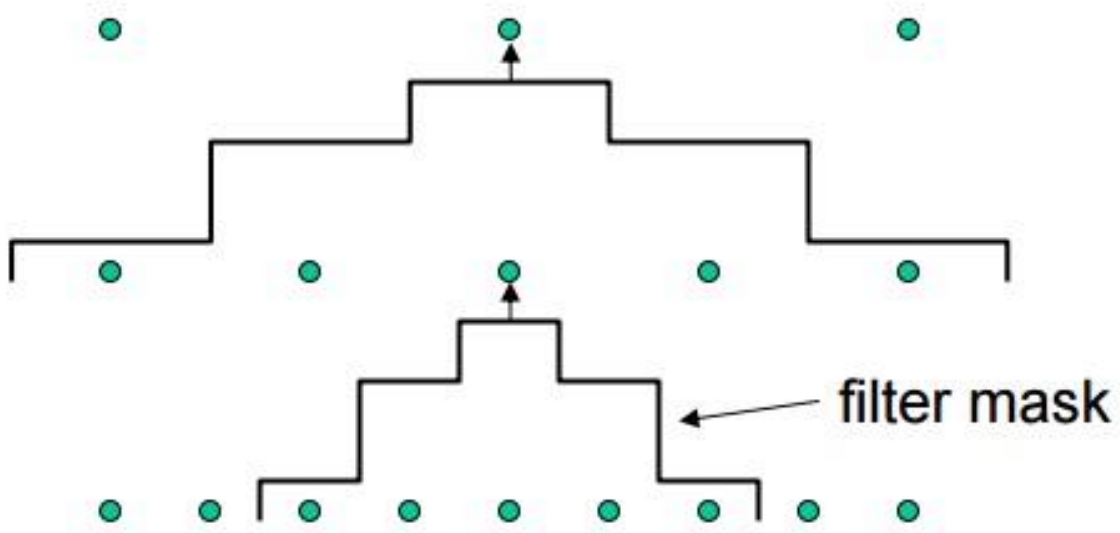
A: Image Pyramid

Image Pyramids



Source: Forsyth

Image Pyramids



Repeat

- Filter
- Subsample

Until minimum resolution reached

- can specify desired number of levels (e.g., 3-level pyramid)

The whole pyramid is only 4/3 the size of the original image!

Gaussian Smoothing

What does blurring take away?

original

Gaussian Smoothing

What does blurring take away?

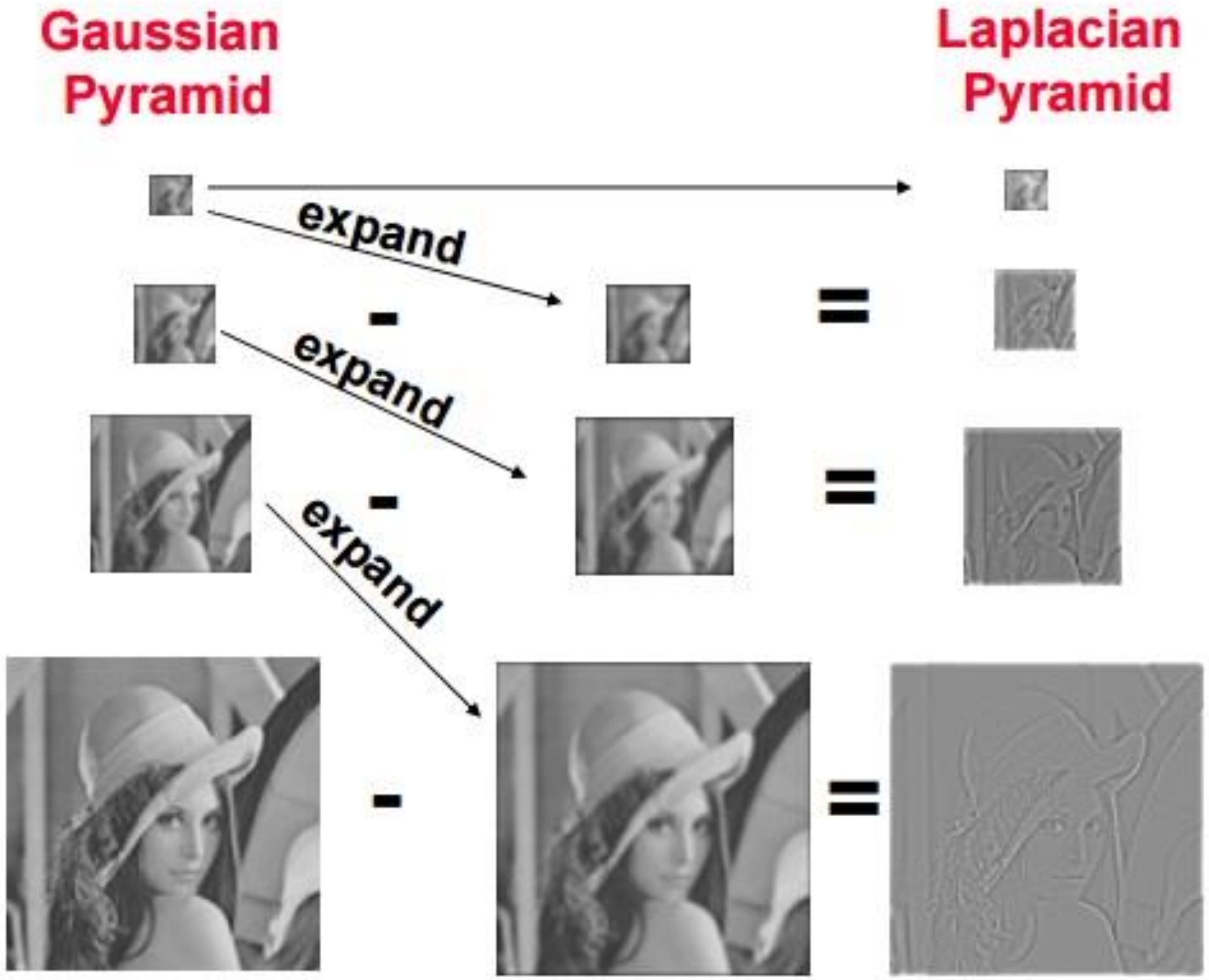
smoothed (5x5 Gaussian)

Gaussian Smoothing

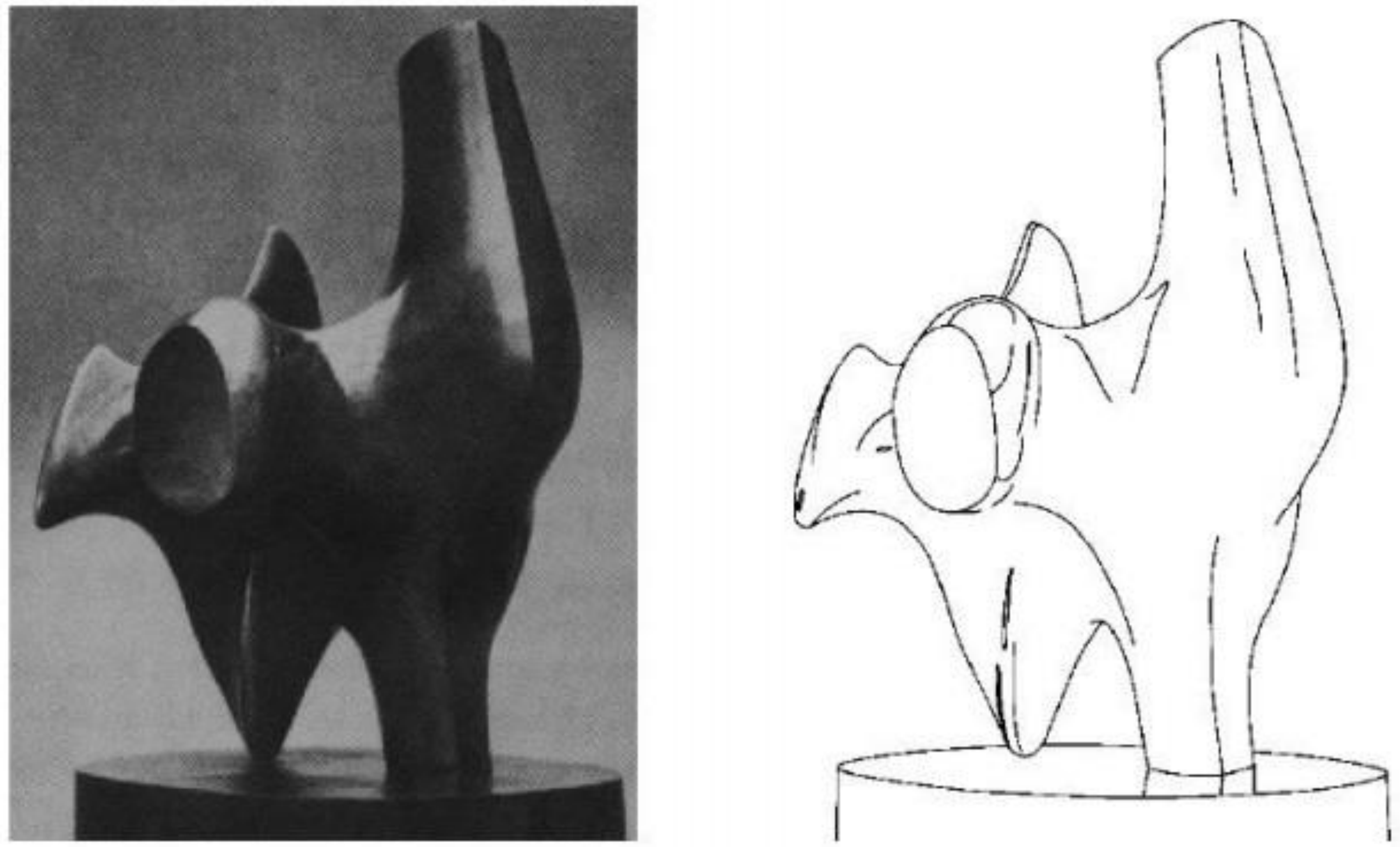
High-Pass filter

smoothed – original

Laplacian Pyramid

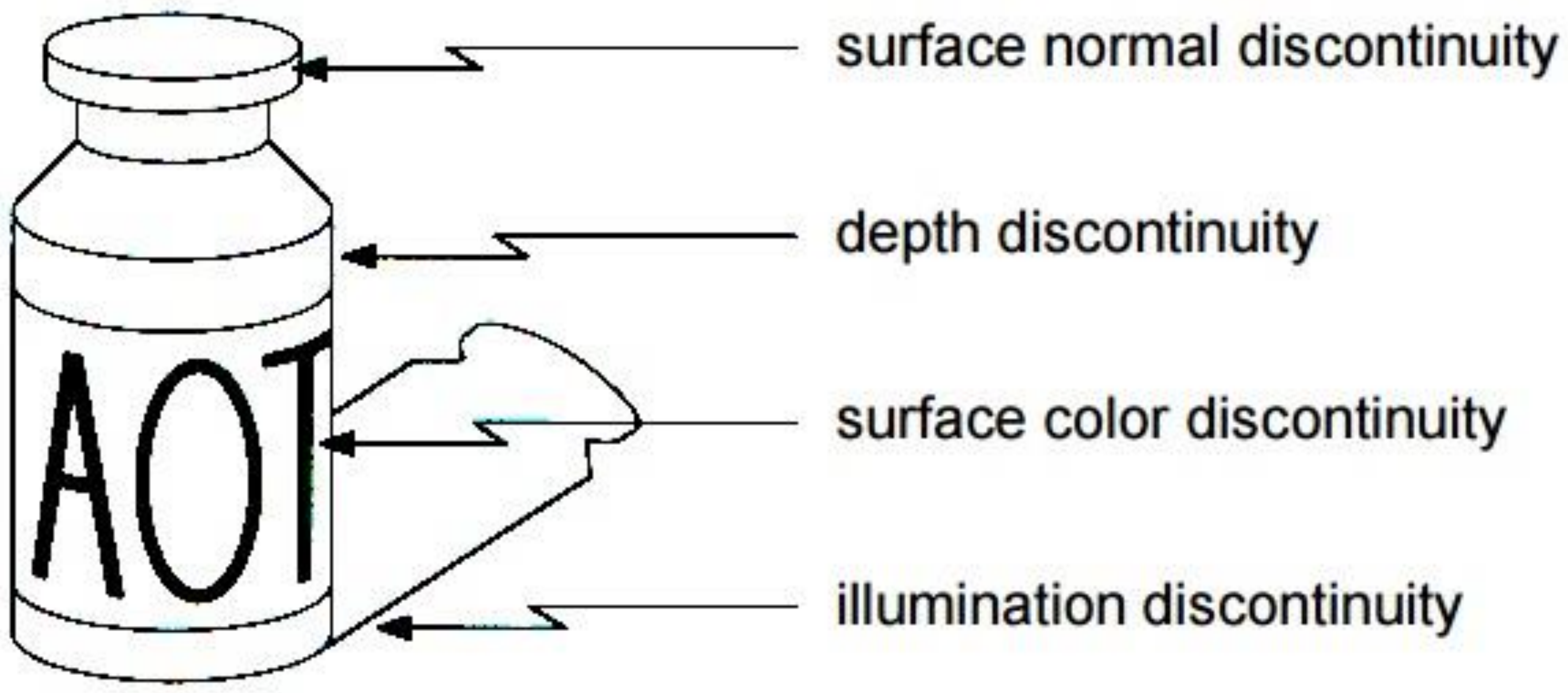


Edges in Depth



- Convert a 2D image into a set of curves
  - Extracts salient features of the scene
  - More compact than pixels

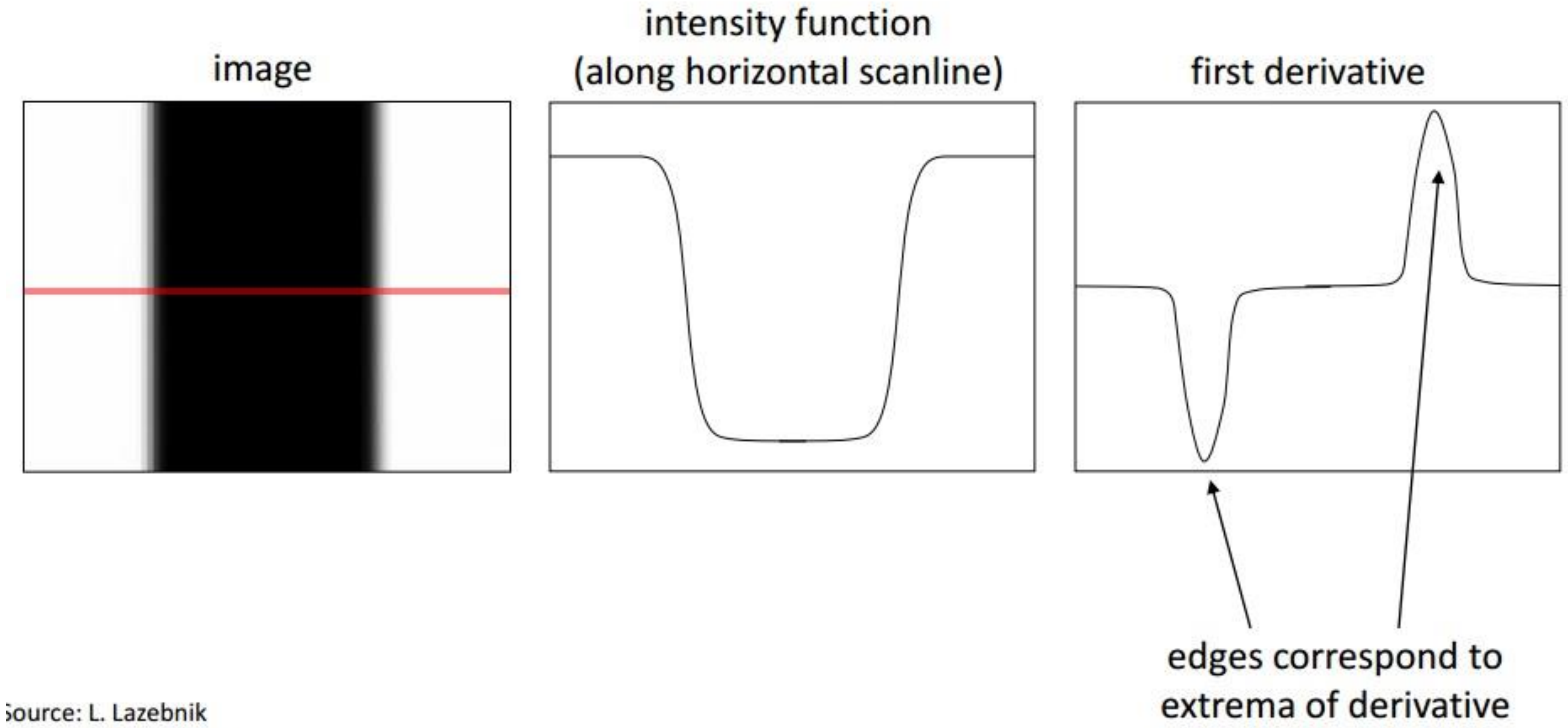
Edges in Depth



- Edges are caused by a variety of factors

Finding Edges

- An edge is a place of rapid change in the image intensity function

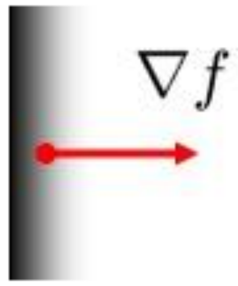


Source: L. Lazebnik

Edges as Gradient

- The *gradient* of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

The gradient points in the direction of most rapid increase in intensity



$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$

$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$

$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

The *edge strength* is given by the gradient magnitude:

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

The gradient direction is given by:

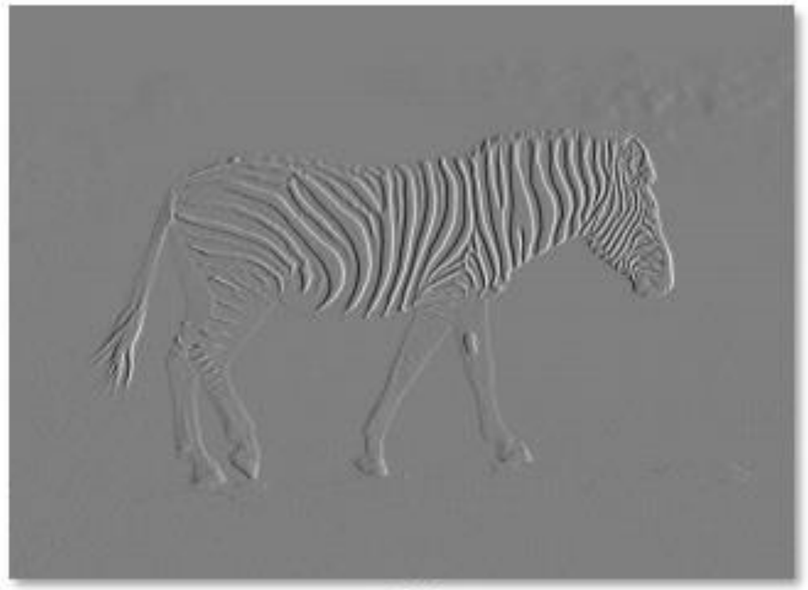
$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x}\right)$$

- how does this relate to the direction of the edge?

Source: Steve Seitz

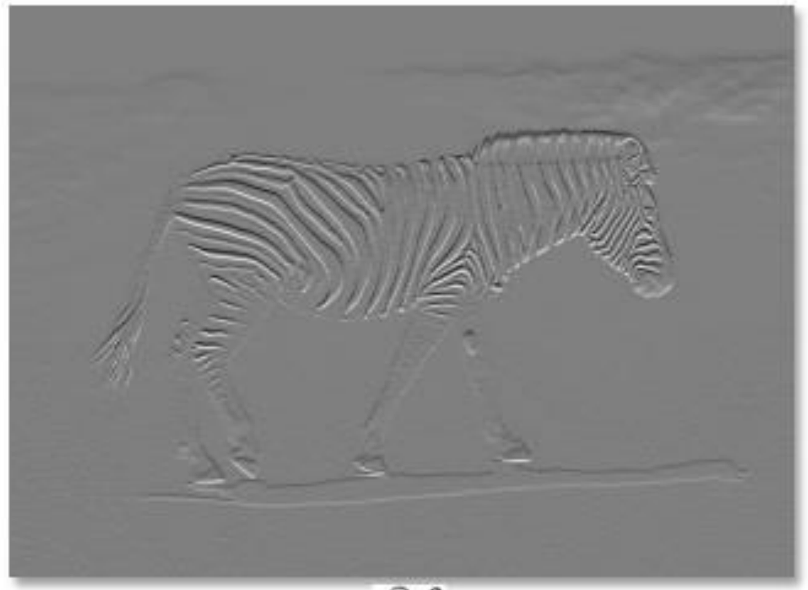
Edges as Gradient

f



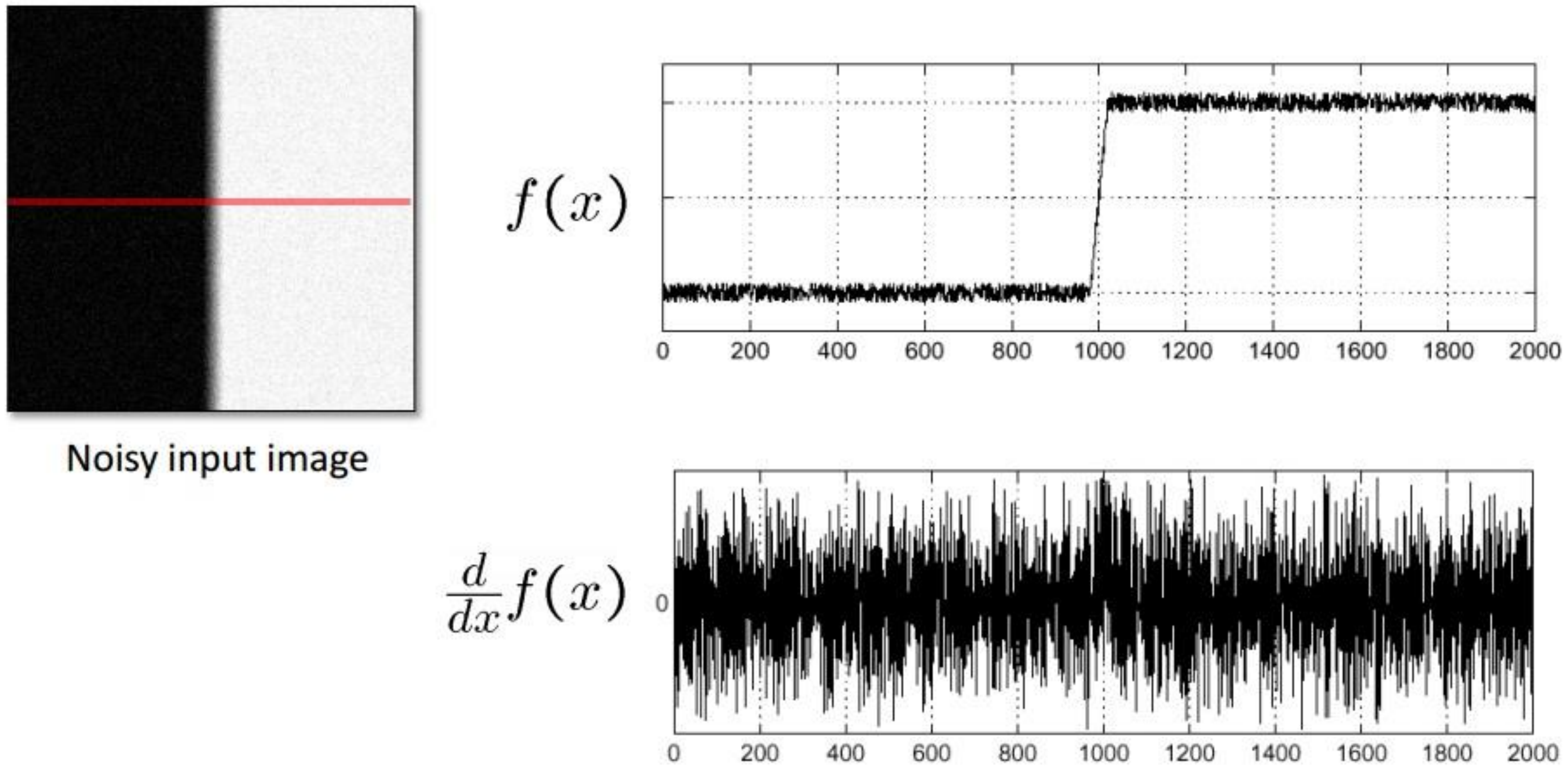
$\frac{\partial f}{\partial x}$

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$



$\frac{\partial f}{\partial y}$

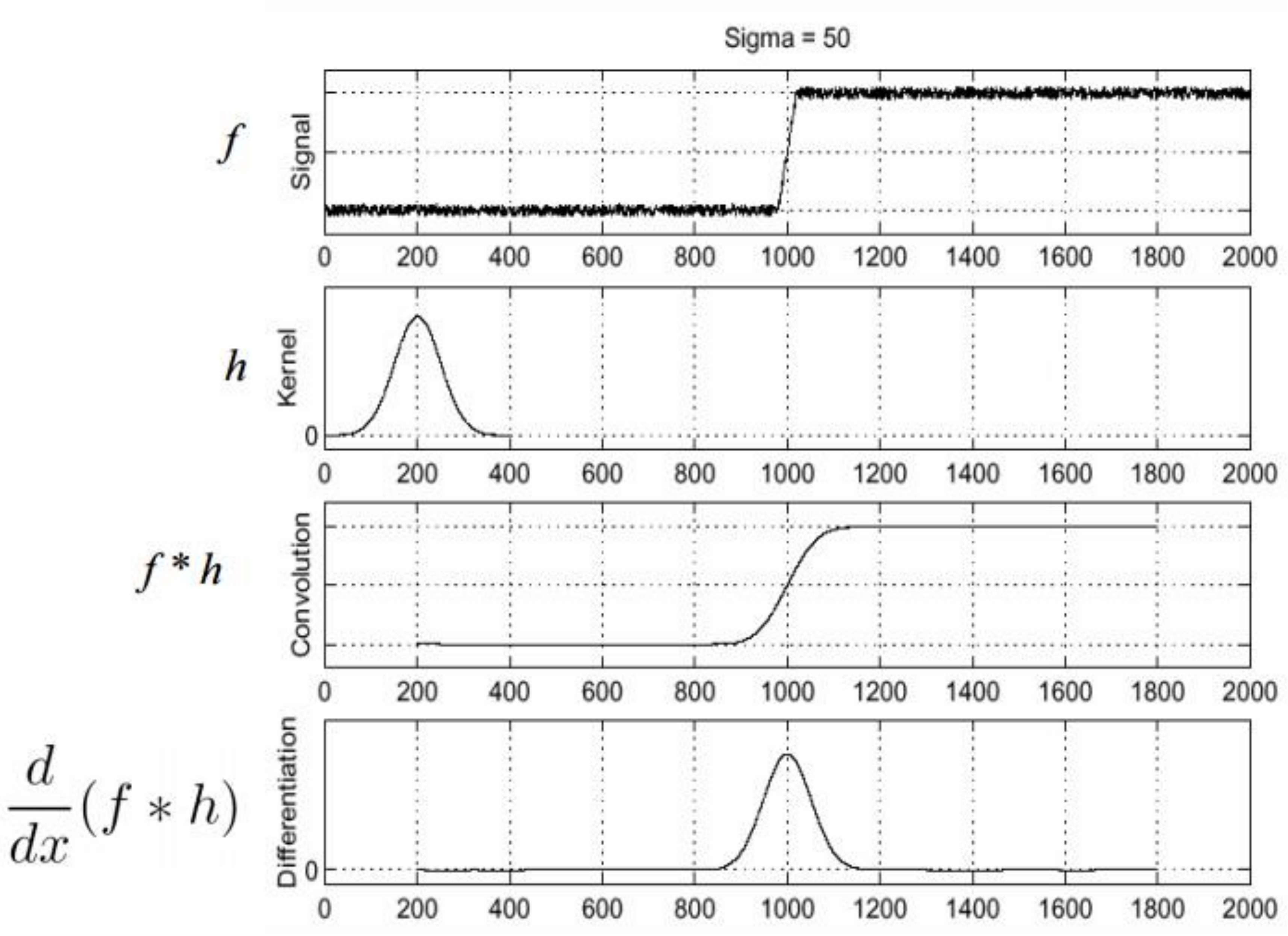
Edges for Noisy Image!



Where is the edge?

Source: S. Seitz

Finding Edges in Noisy Images



To find edges, look for peaks in $\frac{d}{dx}(f * h)$

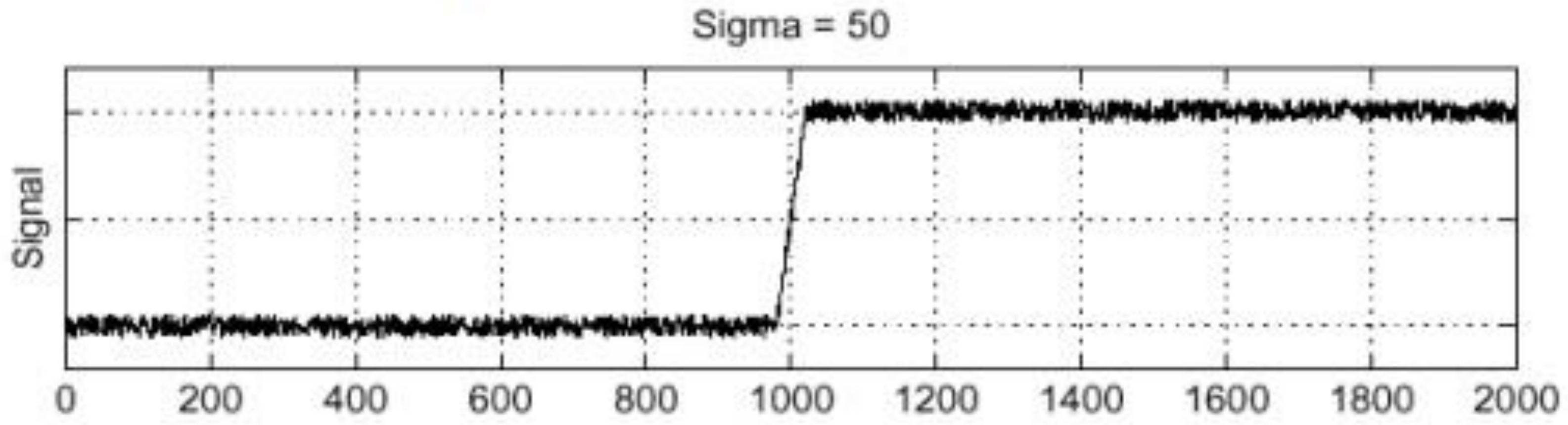
Source: S. Seitz

Differential and Associative Properties

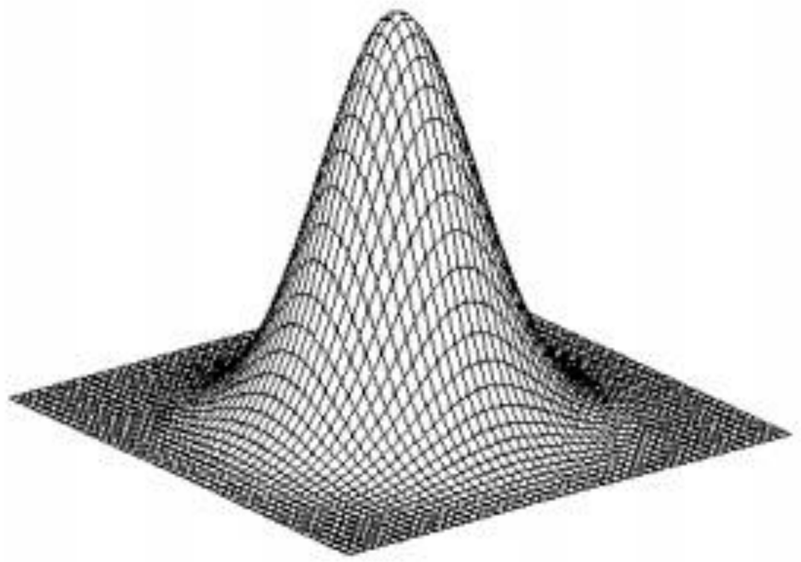
$$\frac{d}{dx}(f * h) = f * \frac{d}{dx}h$$

- This saves us one operation:

f

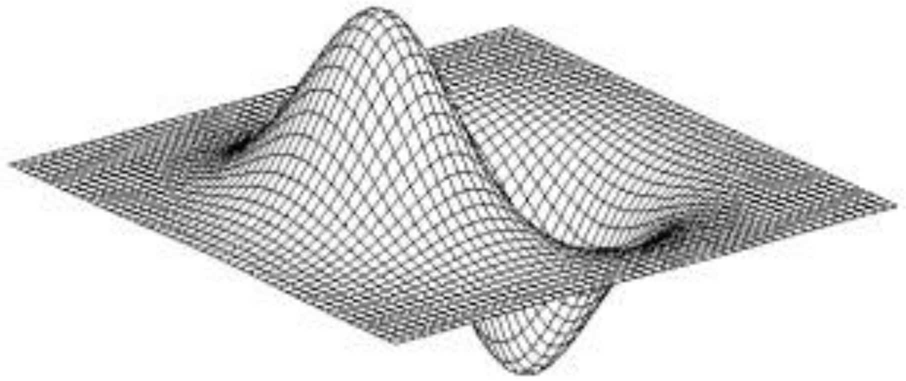


Derivative of Gaussian



Gaussian

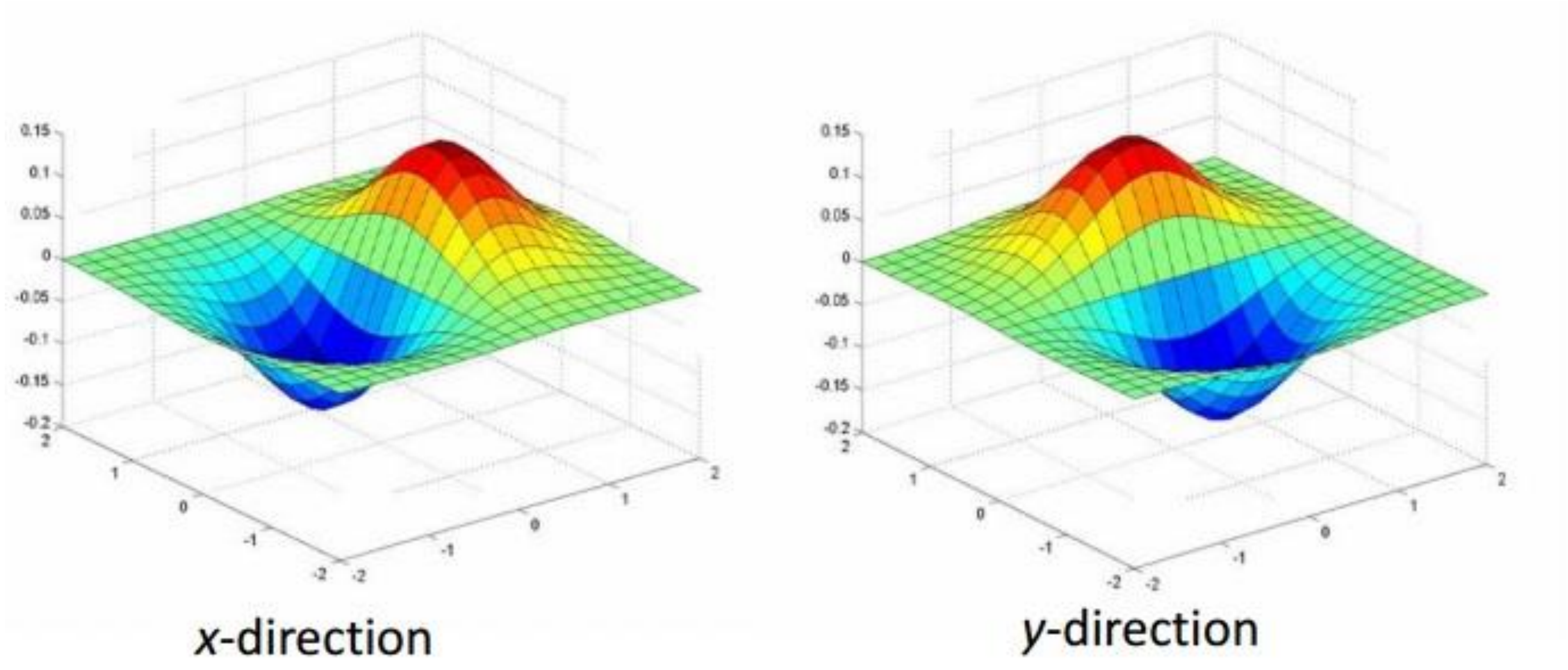
$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

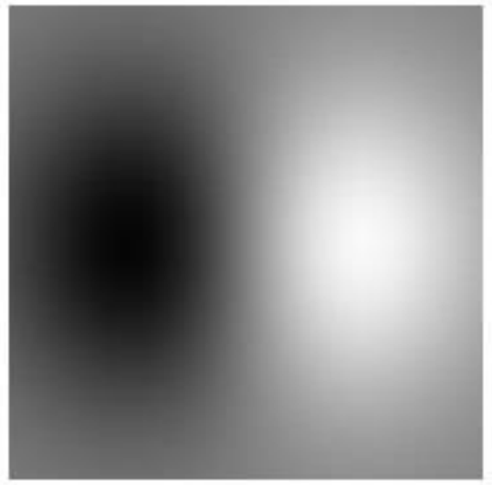


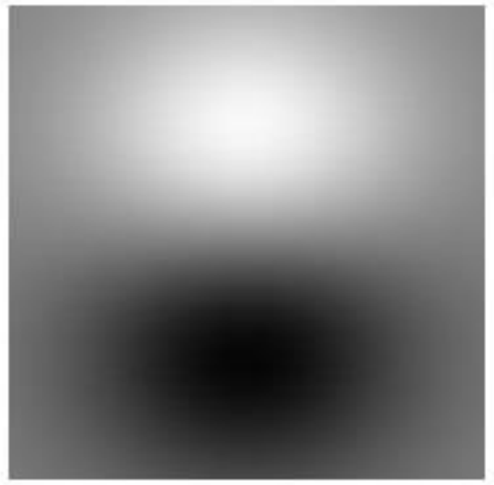
derivative of Gaussian (x)

$$\frac{\partial}{\partial x} h_\sigma(u,v)$$

Derivative of Gaussian







Sobel Filters

- Common approximation of derivative of Gaussian

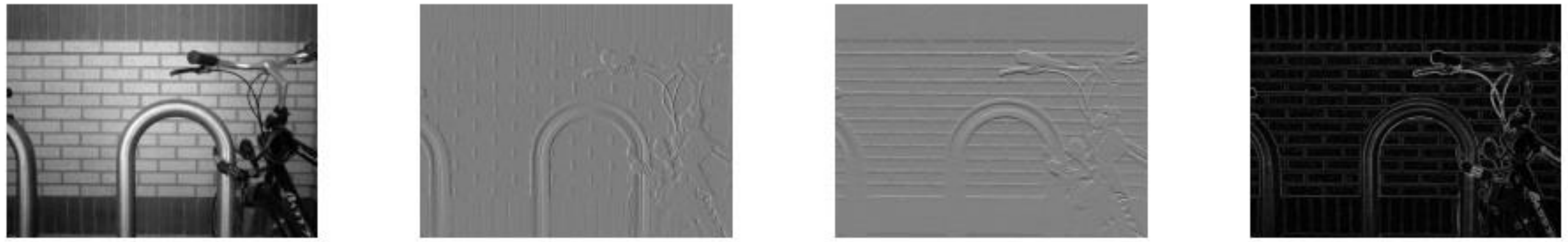
-1	0	1
-2	0	2
-1	0	1

s_x

1	2	1
0	0	0
-1	-2	-1

s_y

Sobel Filter - Example

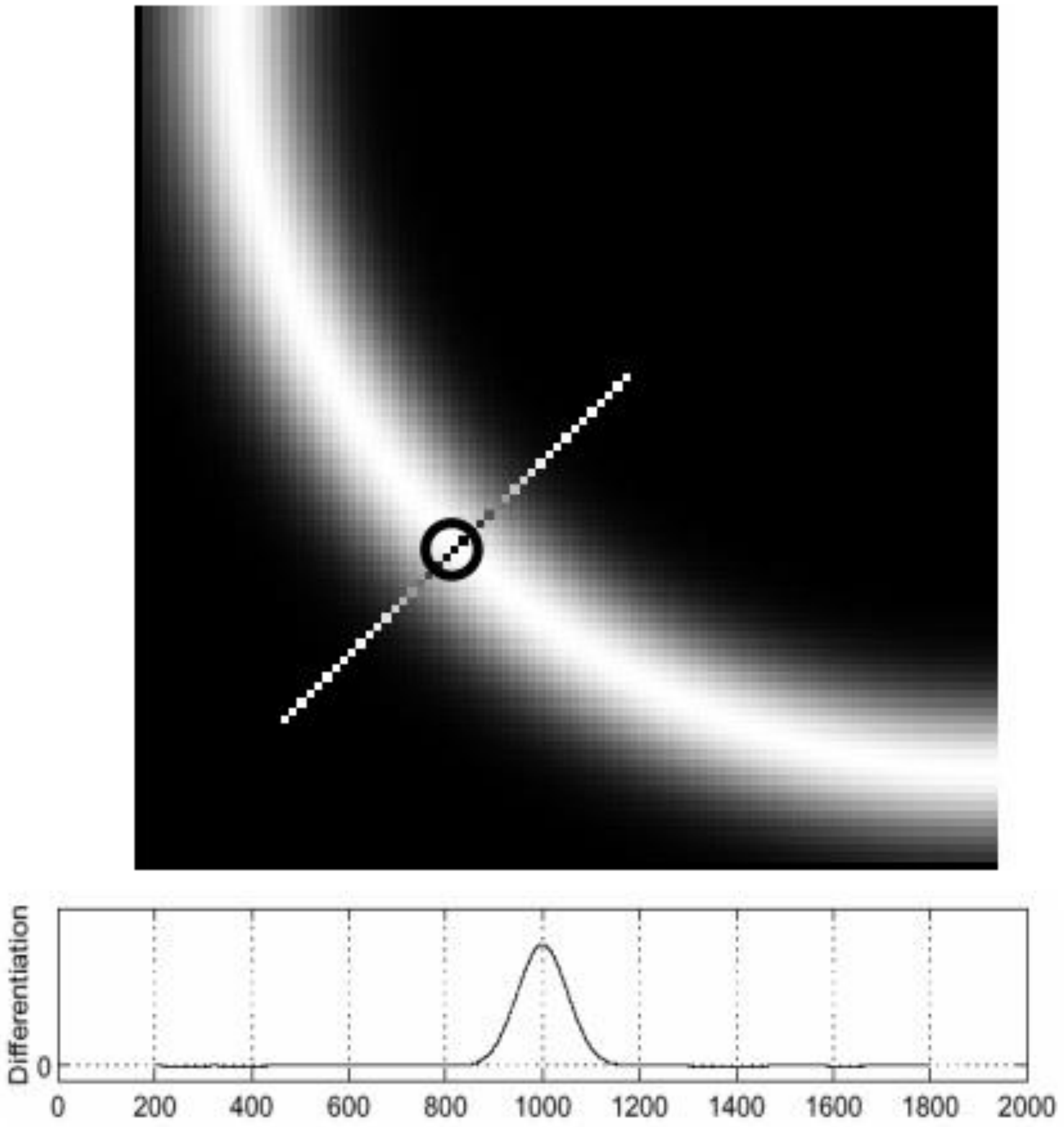


Source: Wikipedia

Sobel Filter - Example

thresholding

Non-Maxima Suppression



- Check if pixel is local maximum along gradient direction

Edge Thinning

thinning

(non-maximum suppression)