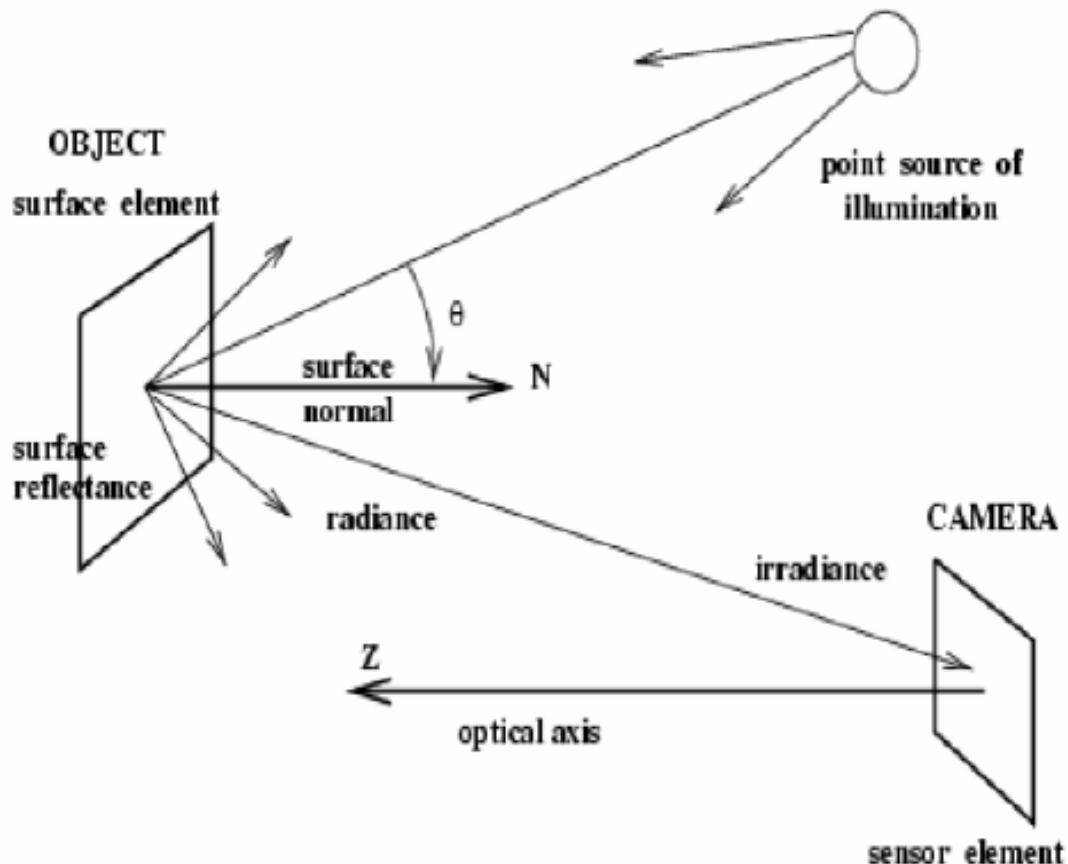
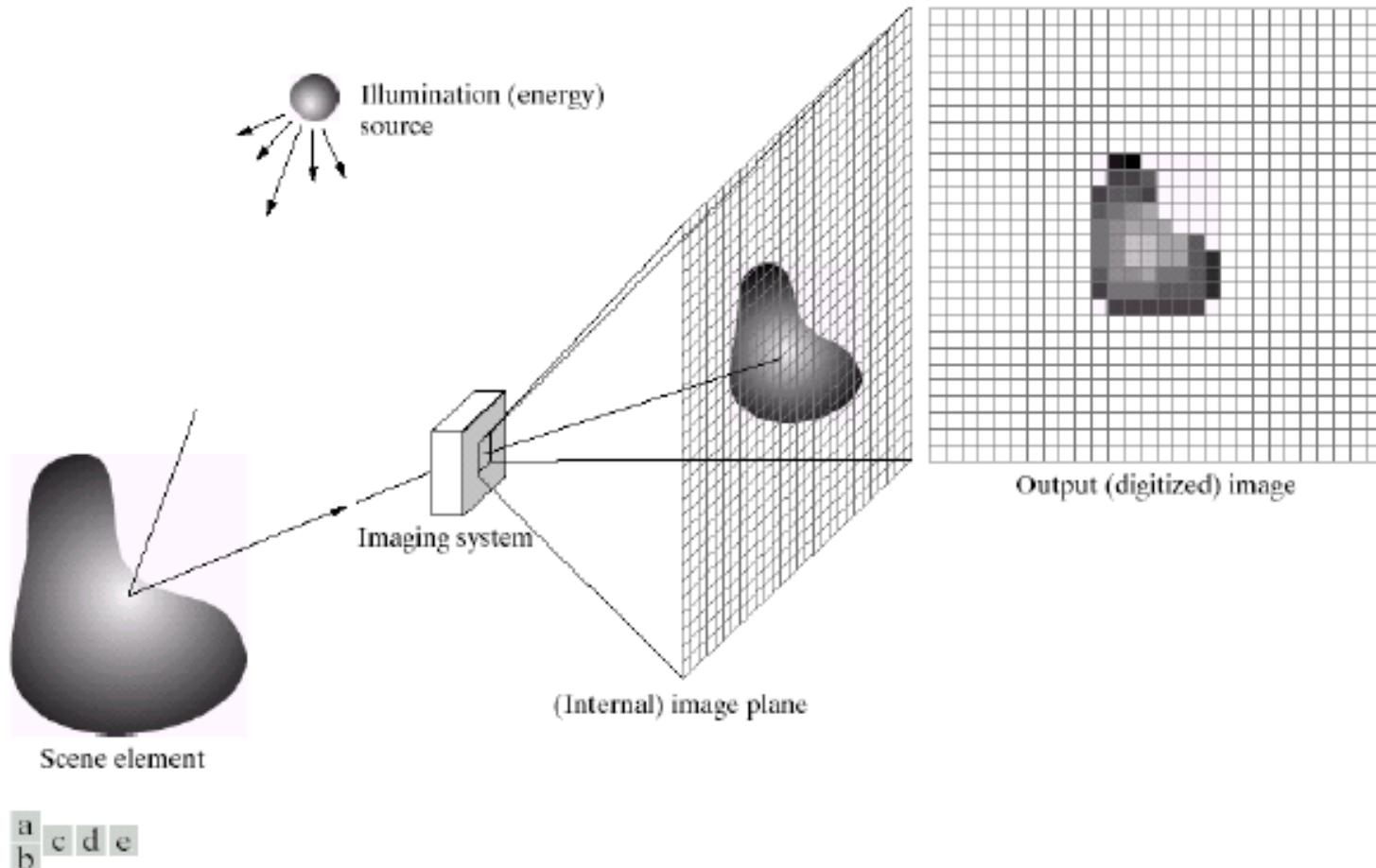


# Elements of Image

- Light reaches surfaces in 3D.
- Surfaces reflect.
- Sensor element receives light energy.
- Intensity is important.
- Angles are important.
- Material is important.

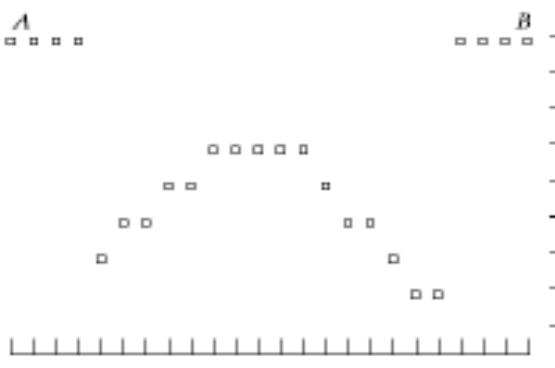
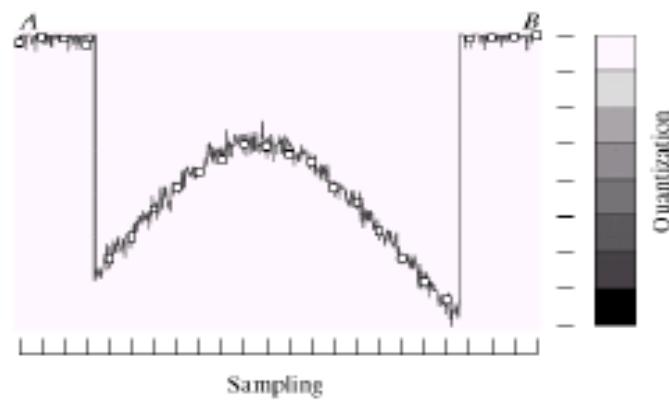
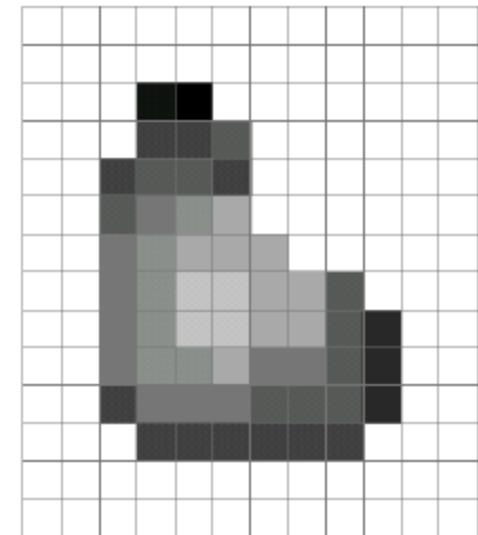
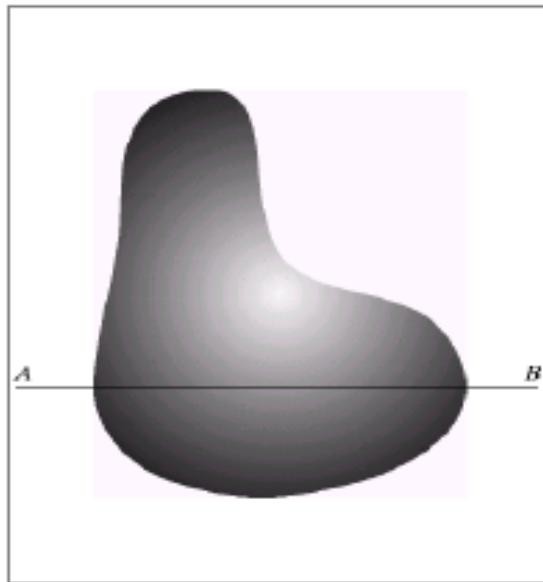


# Image Formation Process



**FIGURE 2.15** An example of the digital image acquisition process. (a) Energy (“illumination”) source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

# Sampling and Quantization



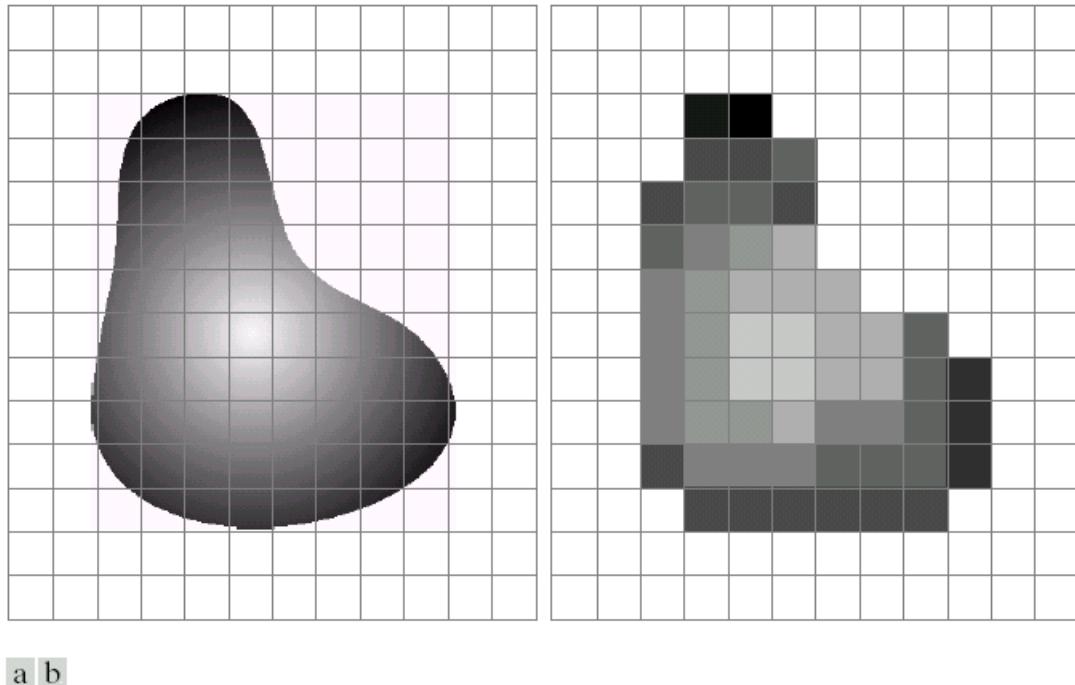
# Digital camera



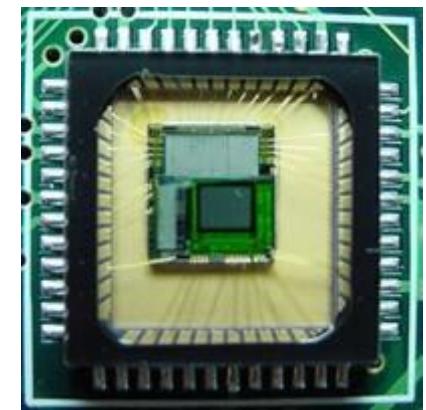
A digital camera replaces film with a sensor array

- Each cell in the array is light-sensitive diode that converts photons to electrons
- Two common types: Charge Coupled Device (CCD) and CMOS
- <http://electronics.howstuffworks.com/digital-camera.htm>

# Sensor Array

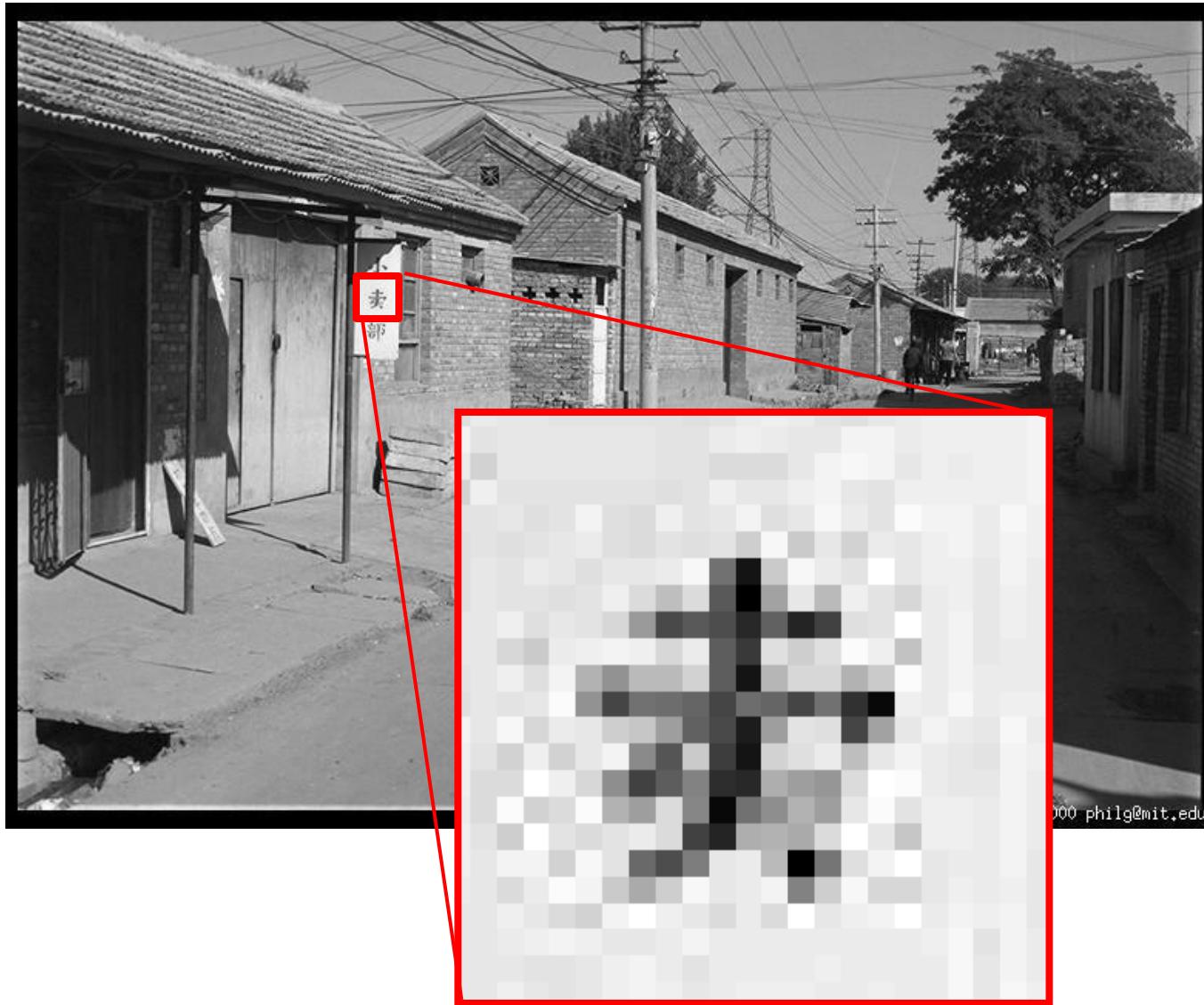


**FIGURE 2.17** (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

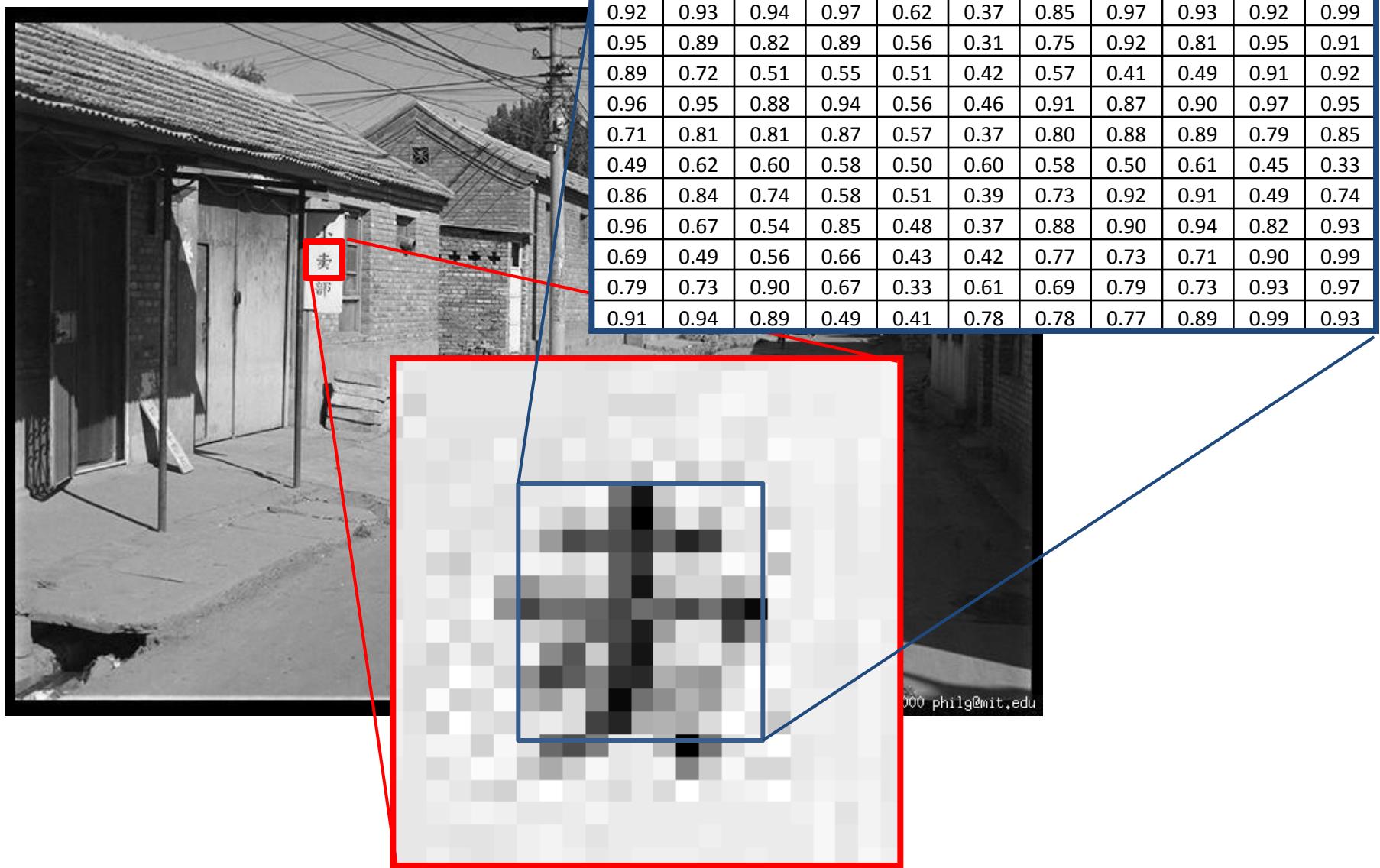


CMOS sensor

# The raster image (pixel matrix)

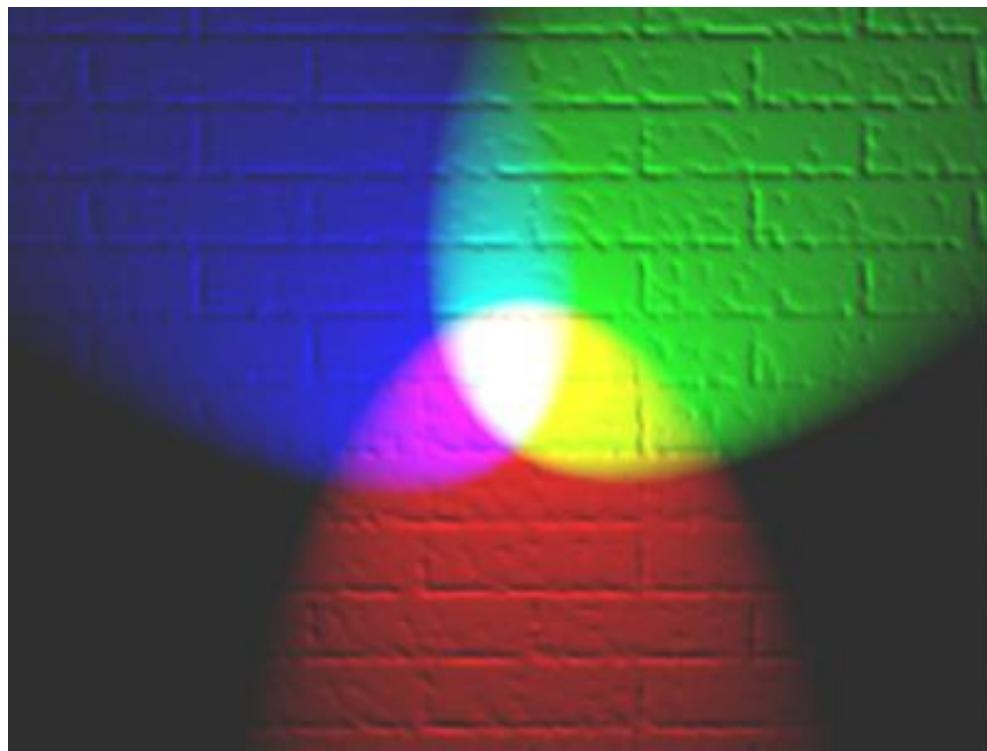


# The raster image (pixel matrix)



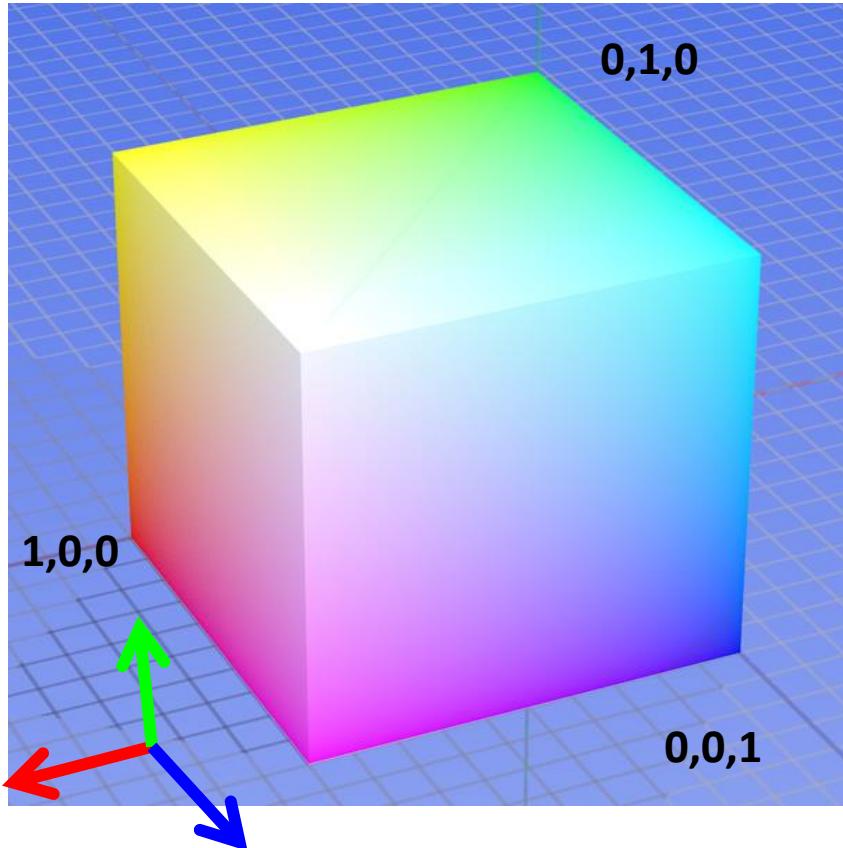
# Color spaces

- How can we represent color? Or color image?



# Color spaces: RGB

Default color space



**R**  
 $(G=0, B=0)$



**G**  
 $(R=0, B=0)$



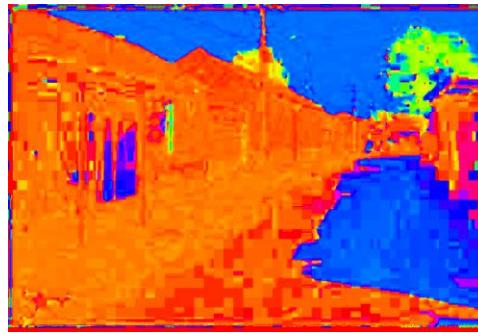
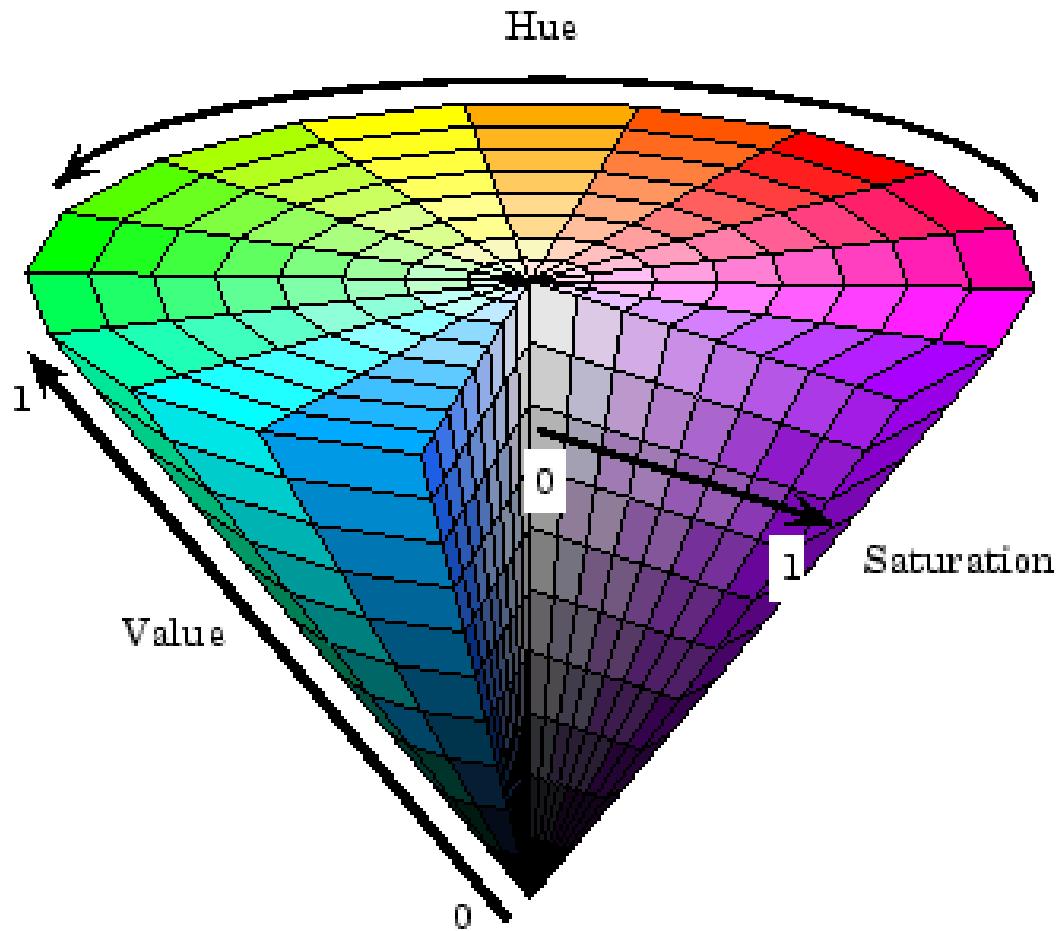
**B**  
 $(R=0, G=0)$

Some drawbacks

- Strongly correlated channels
- Non-perceptual

# Color spaces: HSV

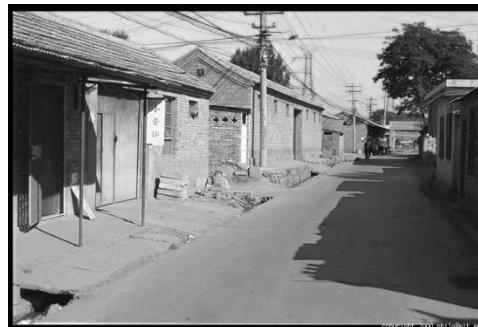
Intuitive color space



**H**  
( $S=1, V=1$ )



**S**  
( $H=1, V=1$ )



**V**  
( $H=1, S=0$ )

# Most information in intensity



Only color shown – constant intensity

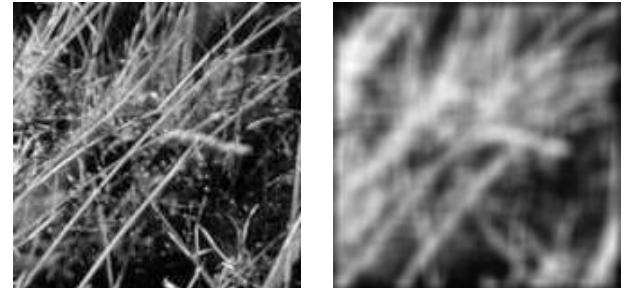
# Most information in intensity



Only intensity shown – constant color

# 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



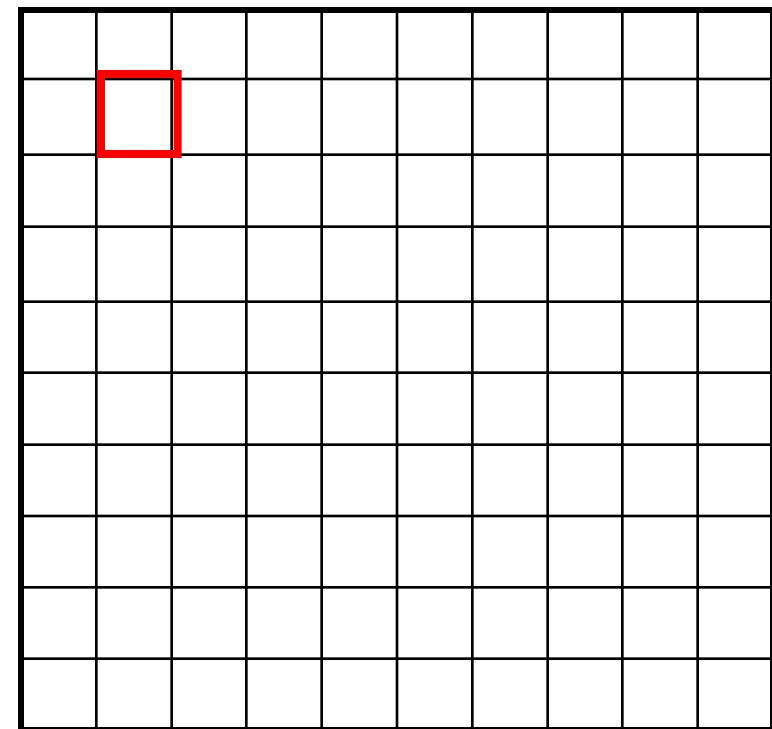
# Image filtering – box filter

$$g[\cdot, \cdot] \frac{1}{9}$$

1	1	1
1	1	1
1	1	1

 $f[.,.]$  $h[.,.]$ 

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



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

Credit: S. Seitz

# Image filtering

$$g[\cdot, \cdot] \frac{1}{9}$$

1	1	1
1	1	1
1	1	1

$$f[.,.]$$

$$h[.,.]$$

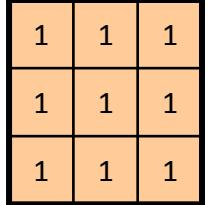
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

			0	10						

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

Credit: S. Seitz

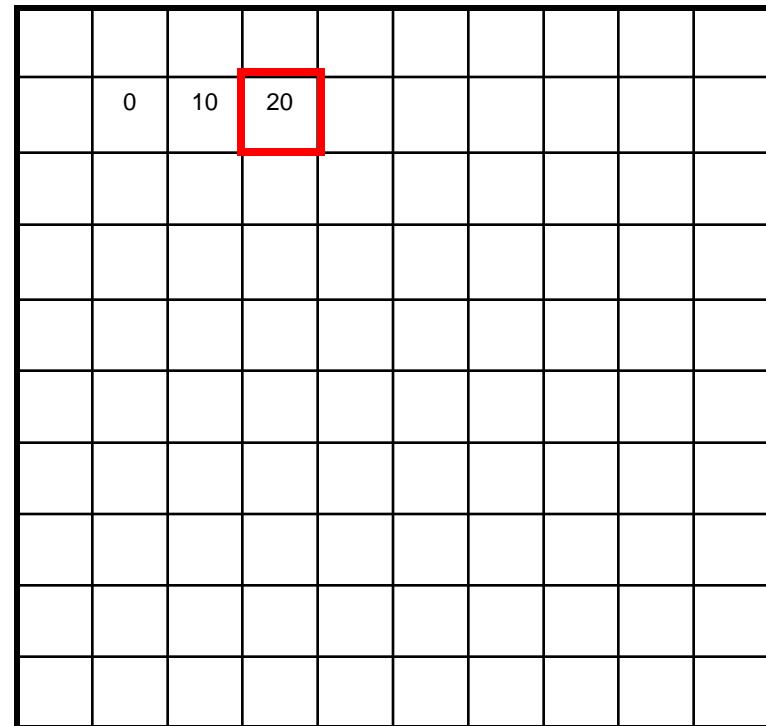
# Image filtering

$$g[\cdot, \cdot] \frac{1}{9}$$


$$f[.,.]$$

$$h[.,.]$$

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



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

Credit: S. Seitz

# Image filtering

$$g[\cdot, \cdot] \frac{1}{9}$$

1	1	1
1	1	1
1	1	1

$$f[.,.]$$

$$h[.,.]$$

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


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

Credit: S. Seitz

# Image filtering

$$g[\cdot, \cdot] \frac{1}{9}$$

1	1	1
1	1	1
1	1	1

$$f[.,.]$$

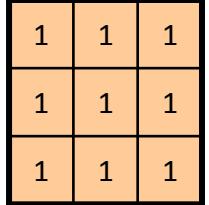
$$h[.,.]$$

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


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

Credit: S. Seitz

# Image filtering

$$g[\cdot, \cdot] \frac{1}{9}$$


$$f[., .]$$

$$h[., .]$$

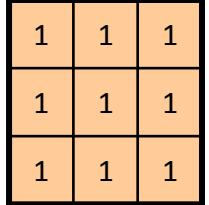
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

			0	10	20	30	30		

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

Credit: S. Seitz

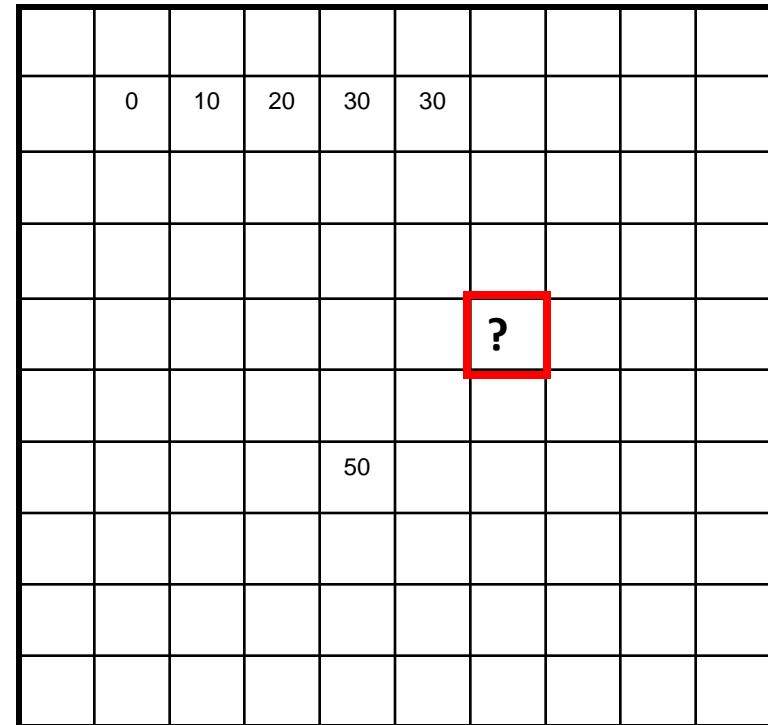
# Image filtering

$$g[\cdot, \cdot] \frac{1}{9}$$


$$f[.,.]$$

$$h[.,.]$$

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	90	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



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

Credit: S. Seitz

# Image filtering

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

$$f[., .]$$

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

# Box Filter

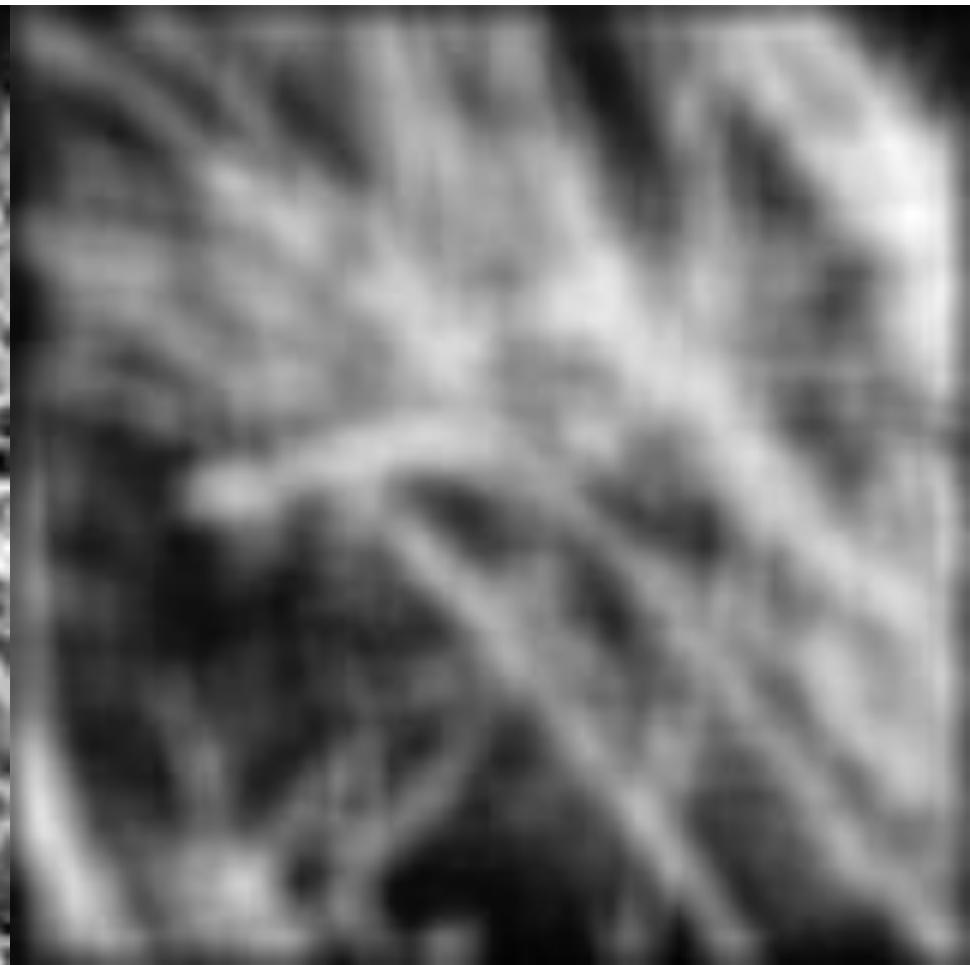
What does it do?

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

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

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

# Smoothing with box filter



# Practice with linear filters



0	0	0
0	1	0
0	0	0

?

# Practice with linear filters



Original

0	0	0
0	1	0
0	0	0



Filtered  
(no change)

# Practice with linear filters



0	0	0
0	0	1
0	0	0

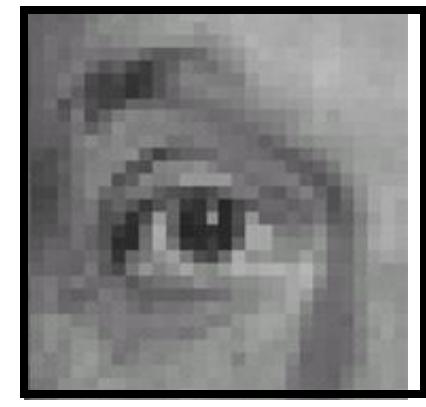
?

# Practice with linear filters



Original

0	0	0
0	0	1
0	0	0



Shifted left  
By 1 pixel

# Practice with linear filters



0	0	0
0	2	0
0	0	0

-

$\frac{1}{9}$	1	1	1
1	1	1	1
1	1	1	1

?

(Note that filter sums to 1)

Original

# Practice with linear filters



Original

0	0	0
0	2	0
0	0	0

-

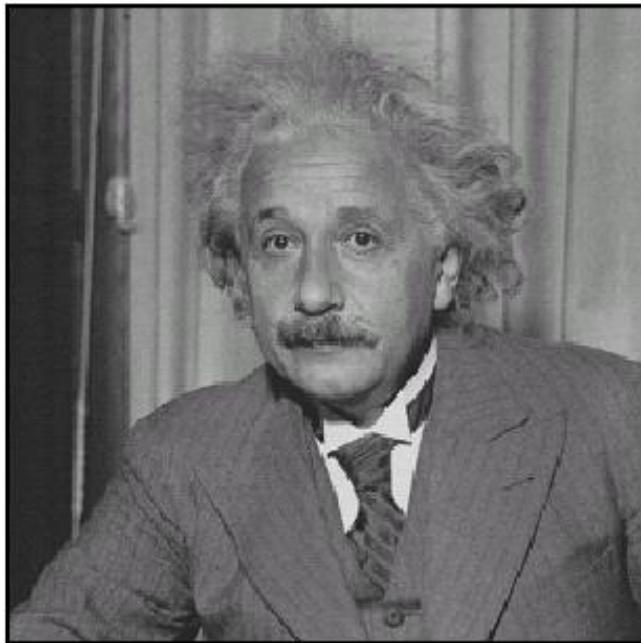
$\frac{1}{9}$	1	1	1
1	1	1	1
1	1	1	1



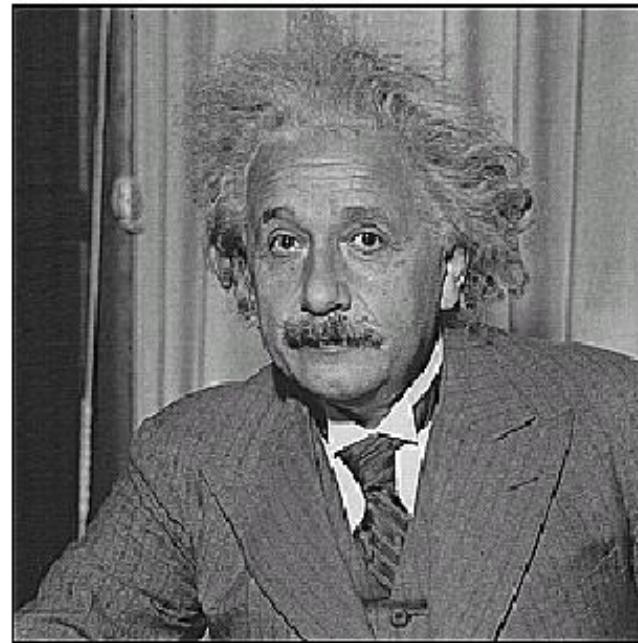
## Sharpening filter

- Accentuates differences with local average

# Sharpening

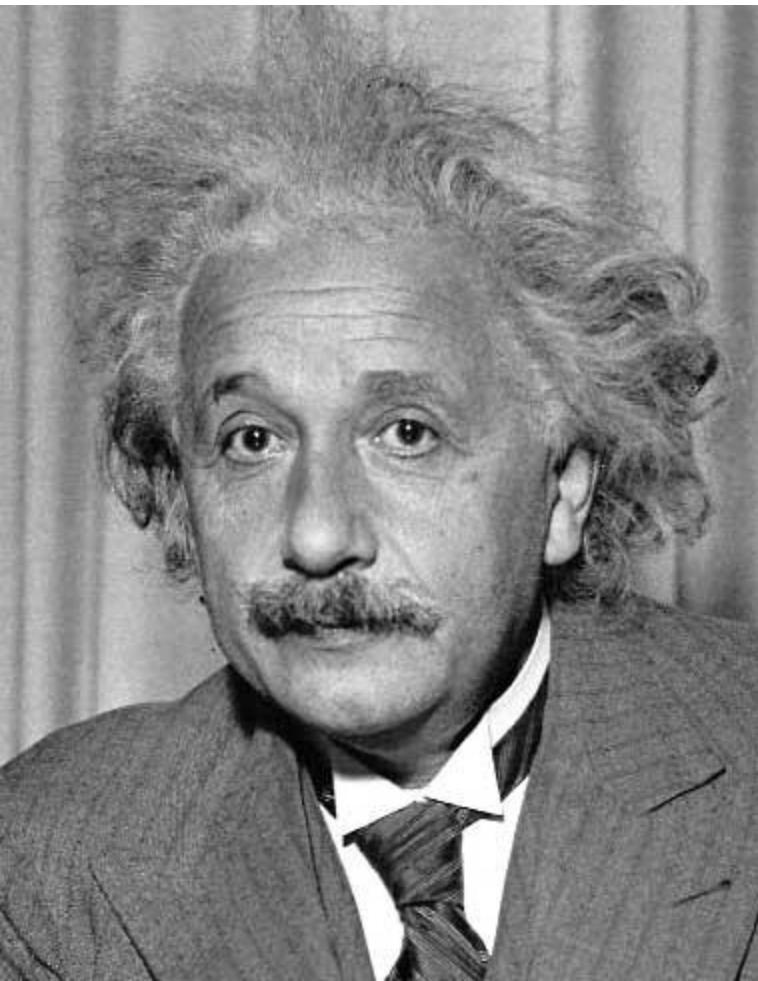


**before**



**after**

# Other filters



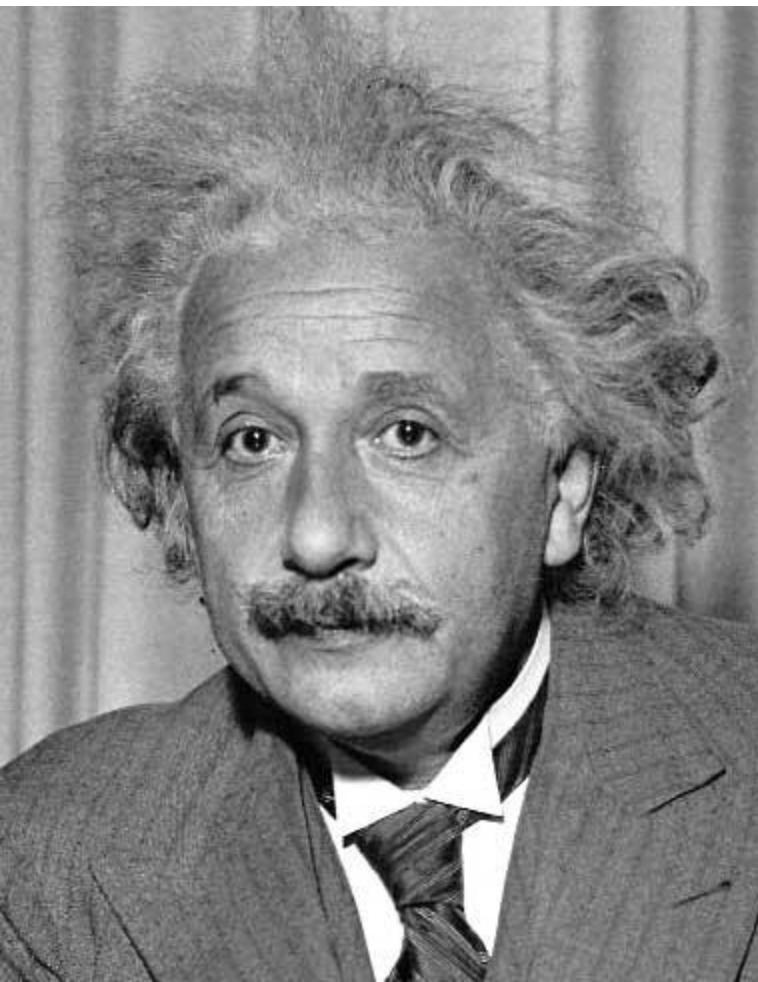
1	0	-1
2	0	-2
1	0	-1

Sobel



Vertical Edge  
(absolute value)

# Other filters



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

Sobel



Horizontal Edge  
(absolute value)

# Key properties of linear filters

## Linearity:

$$\text{filter}(f_1 + f_2) = \text{filter}(f_1) + \text{filter}(f_2)$$

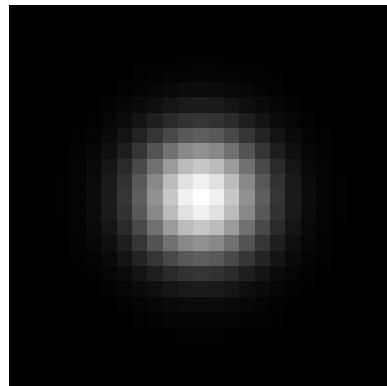
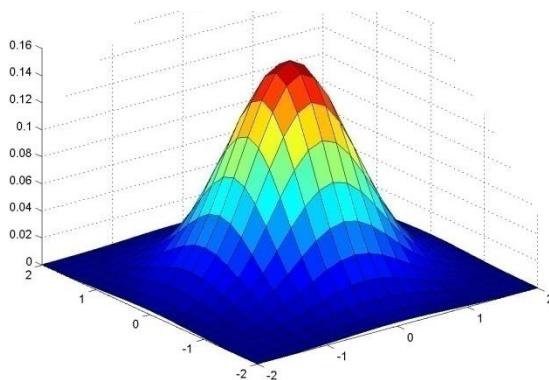
**Shift invariance:** same behavior regardless of pixel location

$$\text{filter}(\text{shift}(f)) = \text{shift}(\text{filter}(f))$$

Any linear, shift-invariant operator can be represented as a convolution

# Important filter: Gaussian

- Weight contributions of neighboring pixels by nearness

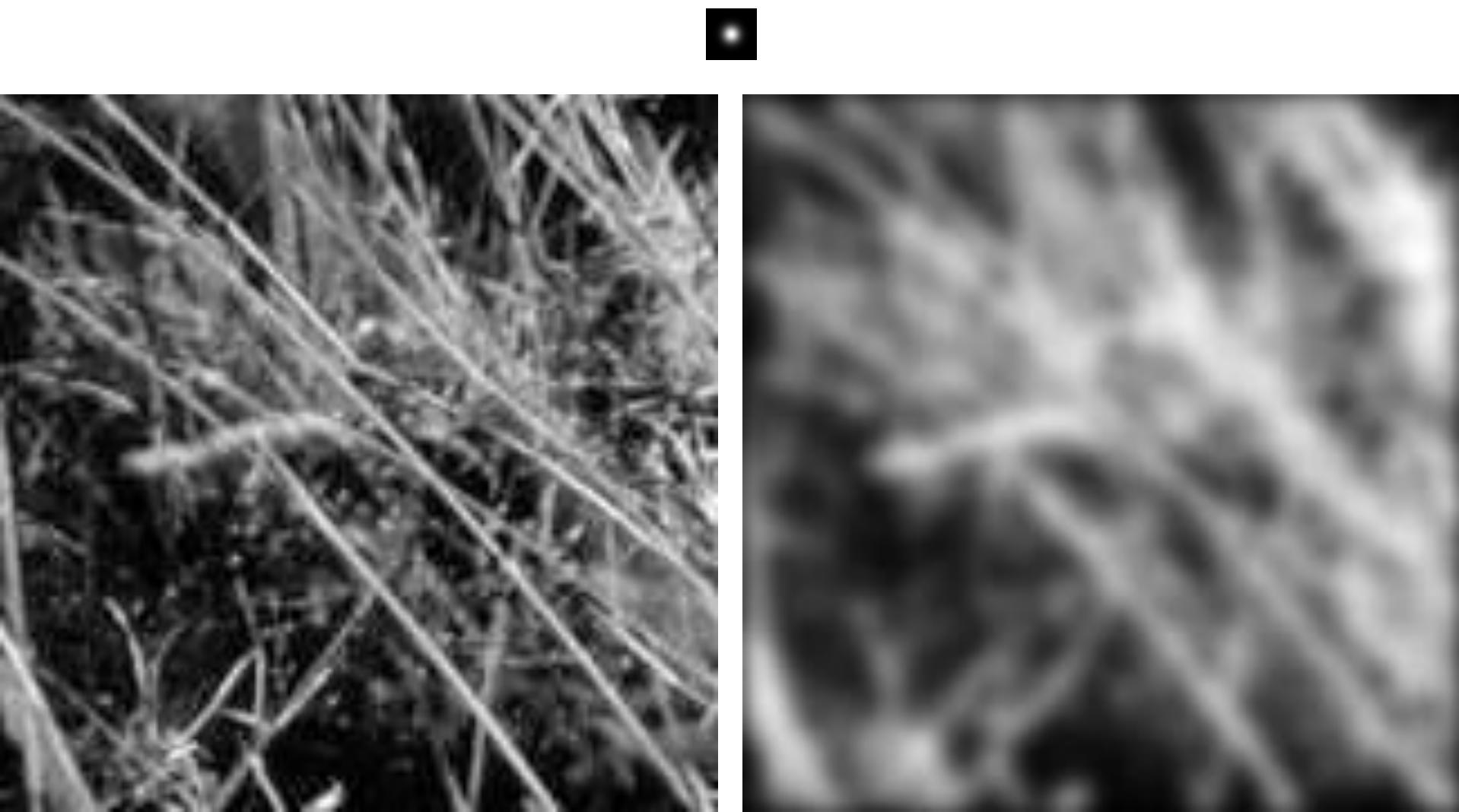


0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

$5 \times 5, \sigma = 1$

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

# Smoothing with Gaussian filter



# Smoothing with box filter

