Image Processing I

Computer Vision
Fall 2018
Columbia University

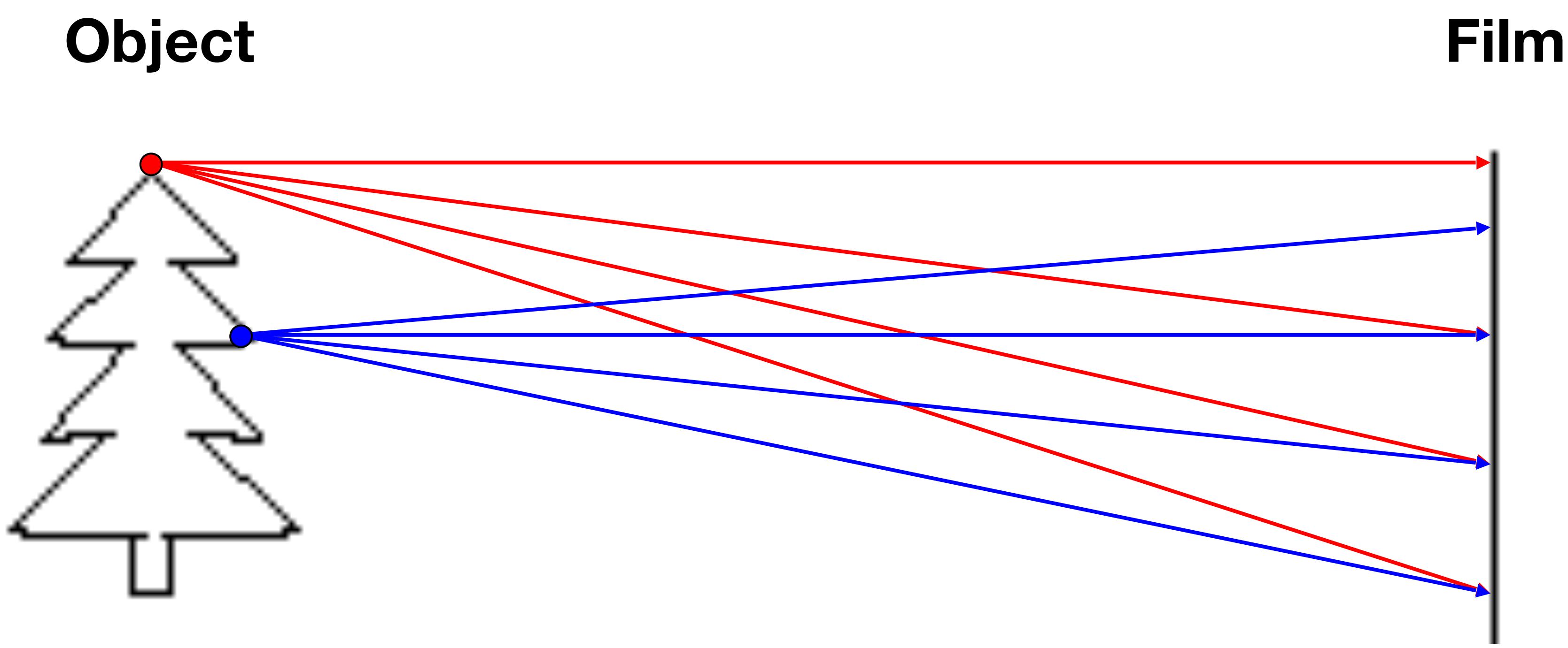
Homework 1

- Posted online today
- **Due September 24** before class starts
- Turn in PDF and your code online

Office Hours

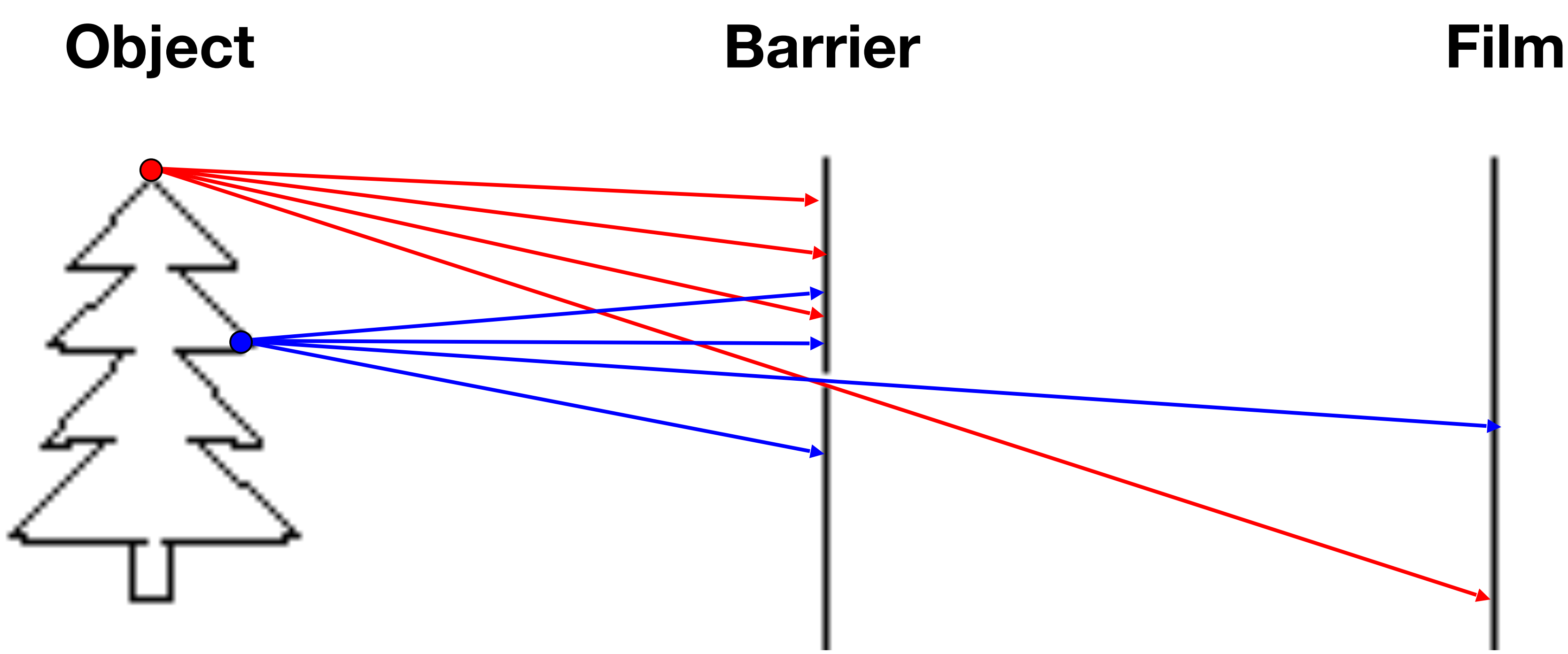
- **Carl:** Monday 4:30pm to 5:30pm, CSB 502
- **Oscar:** Thursday 3-4pm, Mudd 500
- **Xiaoning:** Monday, 5-6pm, CS TA Room
- **Bo:** Tuesday, 3-4pm, CS TA Room
- **James:** Thursday 12-1pm, CS TA Room
- **Luc:** Tuesday 4-5pm, CS TA Room

Image Formation



Slide credit: Steve Seitz

Image Formation



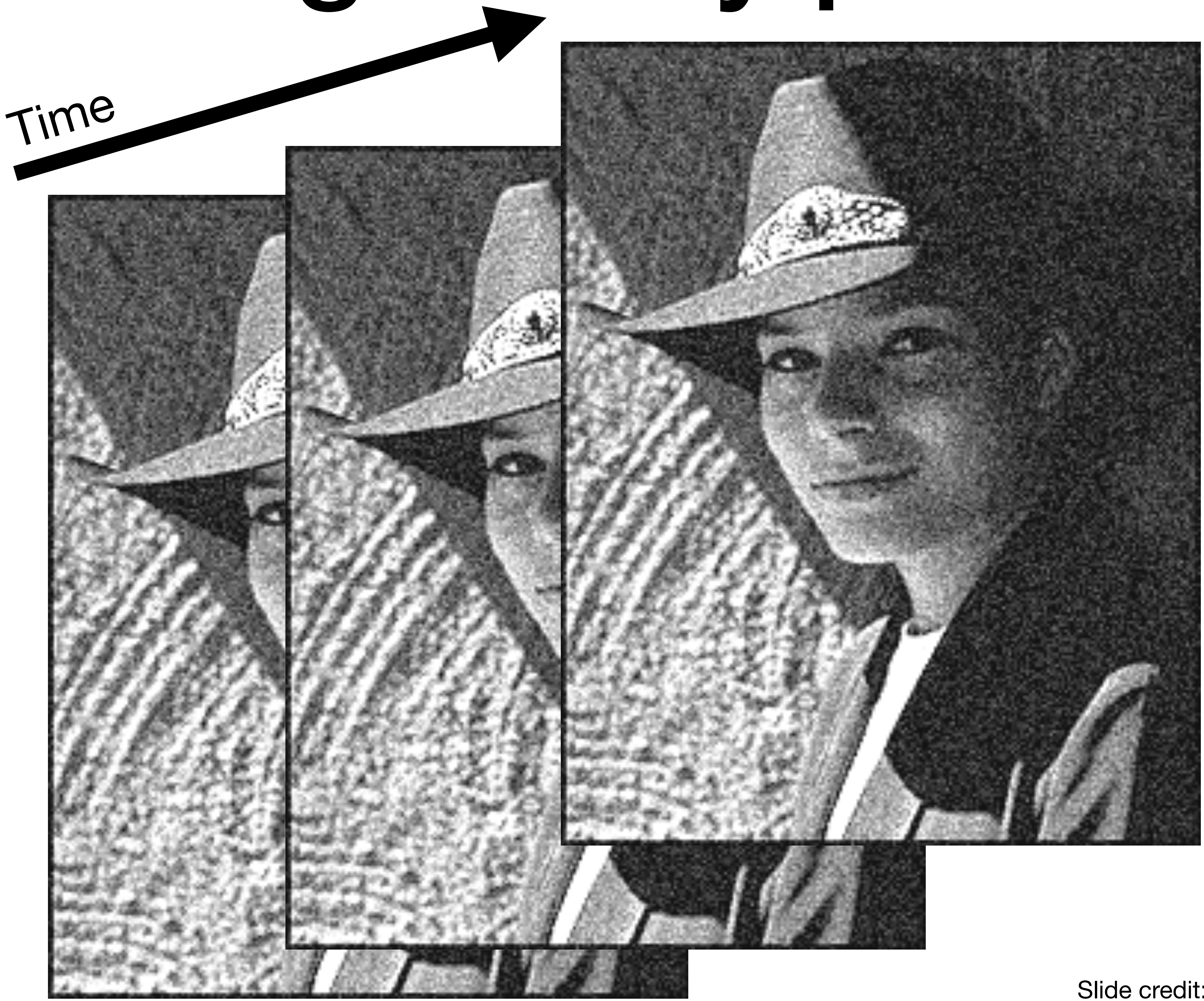
Add a barrier to block off most of the rays

Slide credit: Steve Seitz

Image denoising

Slide credit: S. Lazebnik

Average many photos!

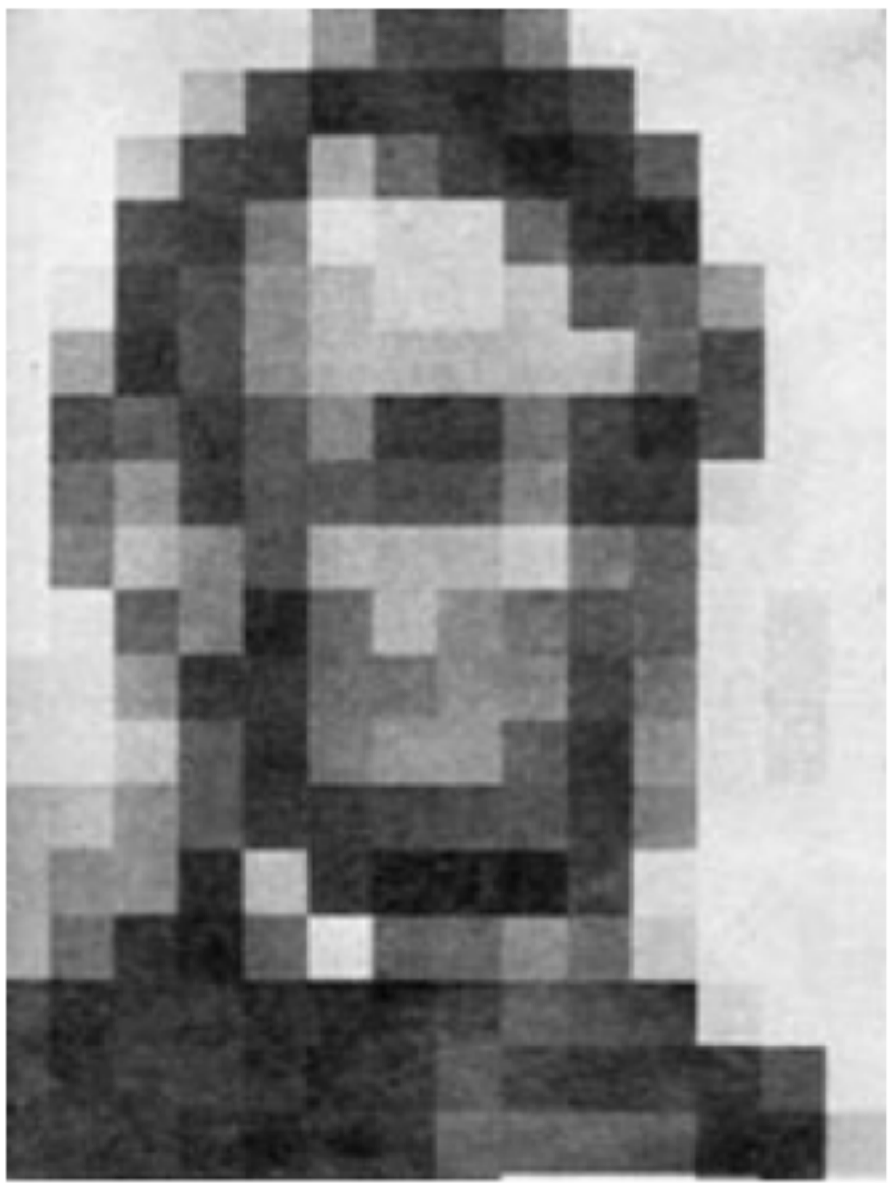


Slide credit: S. Lazebnik

What if just one?

Slide credit: S. Lazebnik

Reminder:
Images as Functions



234	7	89	7	98	98	7	9	7	5
43	7	0	123	4	13	454	23	5	87
67	5	76	4	3	56	67	87	65	45
97	0	6	3	6	25	7	3	587	8
78	5	54	7	876	71	54	76	9	75
45	81	67	78	78	5	4	75	86	8
5	4	3	35	8	256	6	4	3	36
7	6	64	3	4	7	77	76	4	54
64	35	46	46	64	56	7	56	4	7
75	464	576	75	75	75	57	64	75	75

$$F[x, y]$$

Moving Average

$$F[x, y]$$

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

Source: S. Seitz

Moving Average

$F[x, y]$

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								

Source: S. Seitz

Moving Average

$$F[x, y]$$

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							

Source: S. Seitz

Moving Average

$F[x, y]$

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						

Source: S. Seitz

Moving Average

$F[x, y]$

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					

Source: S. Seitz

Moving Average

$F[x, y]$

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					
				?					

Source: S. Seitz

Moving Average

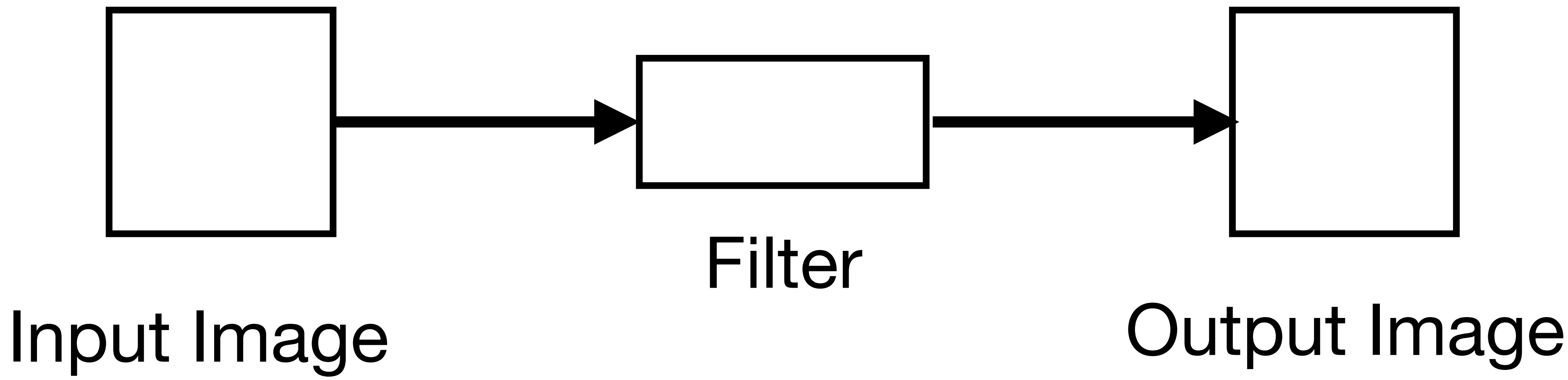
$$F[x, y]$$

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	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	

Source: S. Seitz

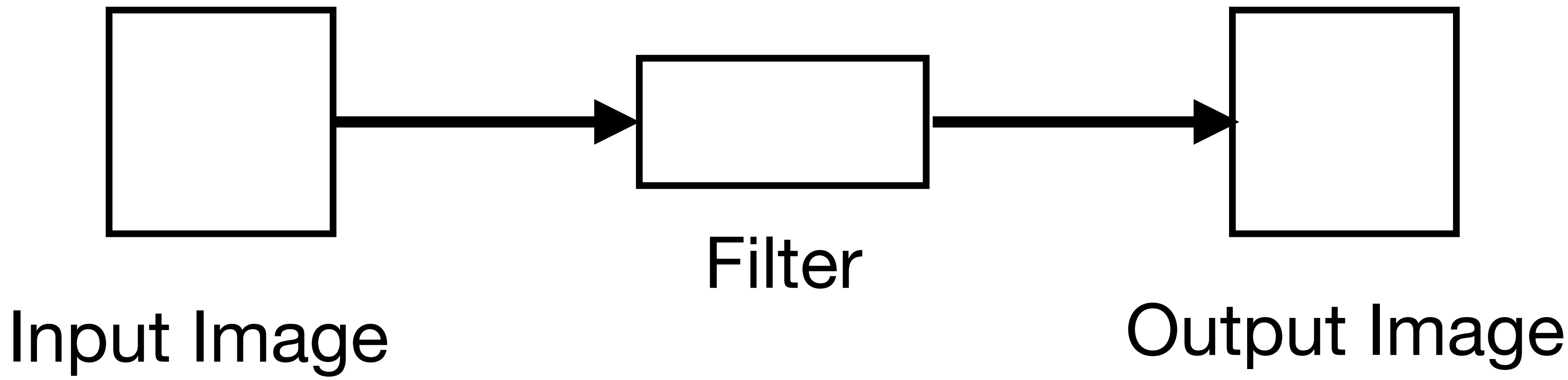
Filtering



We want to remove unwanted sources of variation, and keep the information relevant for whatever task we need to solve

Source: Bill Freeman

Linear Filtering



For a filter to be linear, it must satisfy two properties:

- filter(im, f1 + f2) = filter(im, f1) + filter(im, f2)
- C * filter(im, f1) = filter(im, C * f1)

Source: Bill Freeman

Convolution

$F[x, y]$

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

$G[x, y]$

$* \; \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$=$

	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	

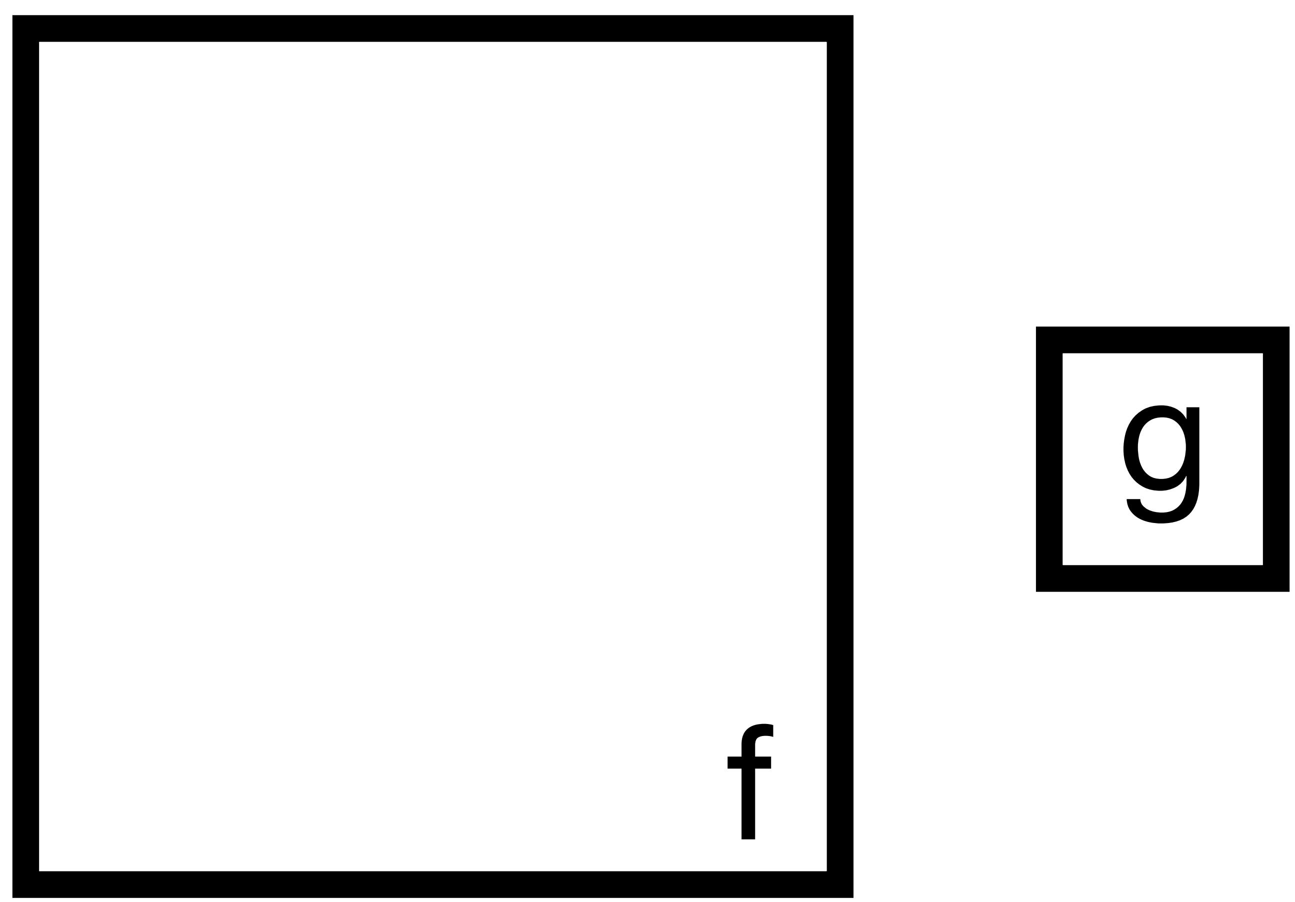
Convolution

- Let f be an image/function, and g is the kernel/filter
- The convolution is defined as:

$$(f * g)[x, y] = \sum_{i,j} f[x - i, y - j]g[i, j]$$

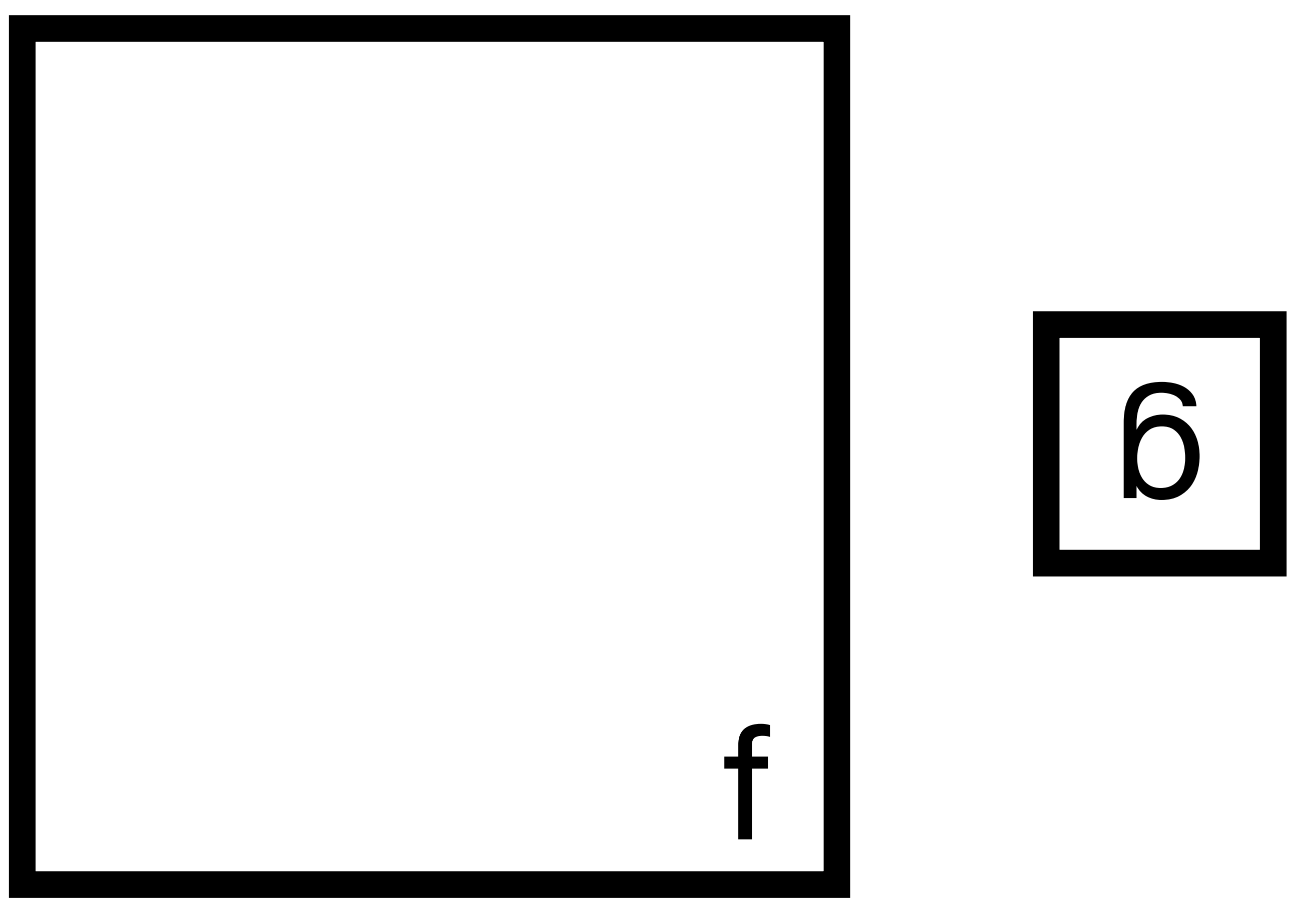
Convolution

$$(f * g)[x, y] = \sum_{i,j} f[x - i, y - j]g[i, j]$$



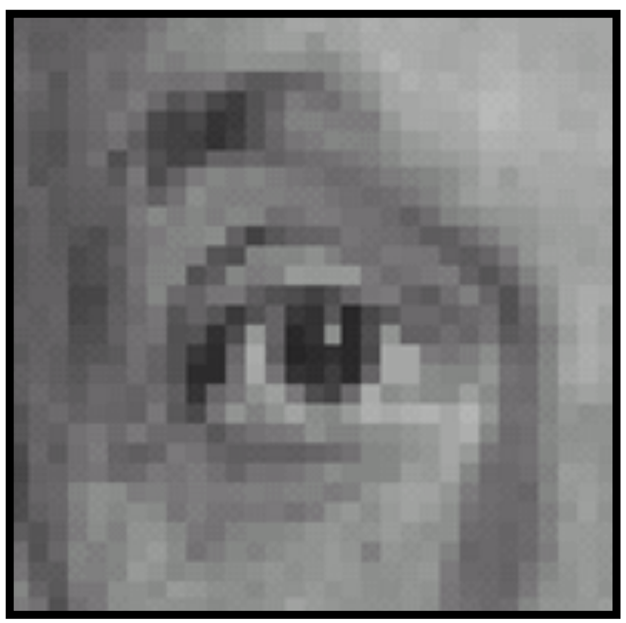
Convolution

$$(f * g)[x, y] = \sum_{i,j} f[x - i, y - j]g[i, j]$$



Flip LR, UD

Convolution Practice

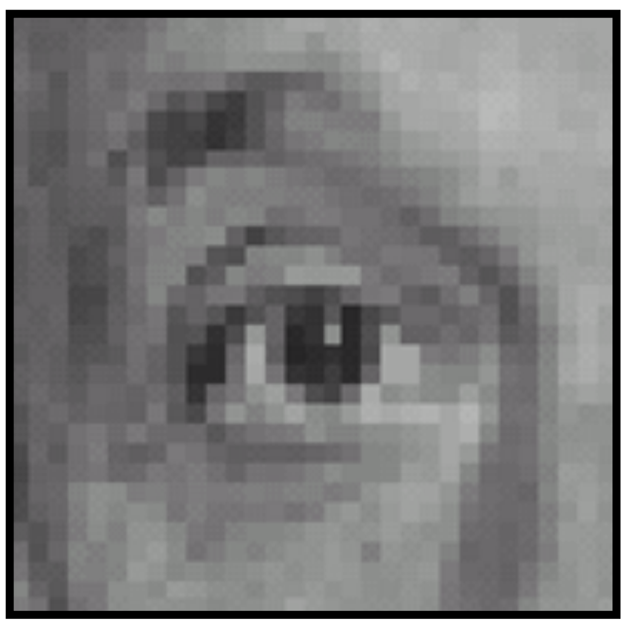


$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

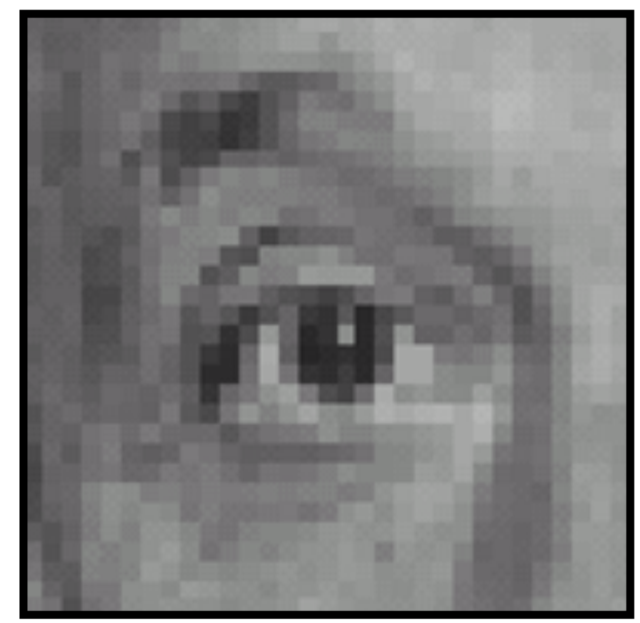
?

Source: D. Lowe

Convolution Practice

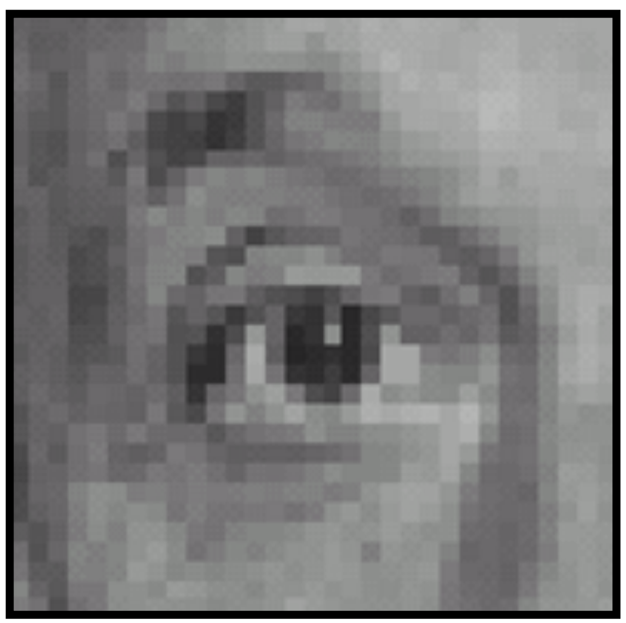


0	0	0
0	1	0
0	0	0



Source: D. Lowe

Convolution Practice



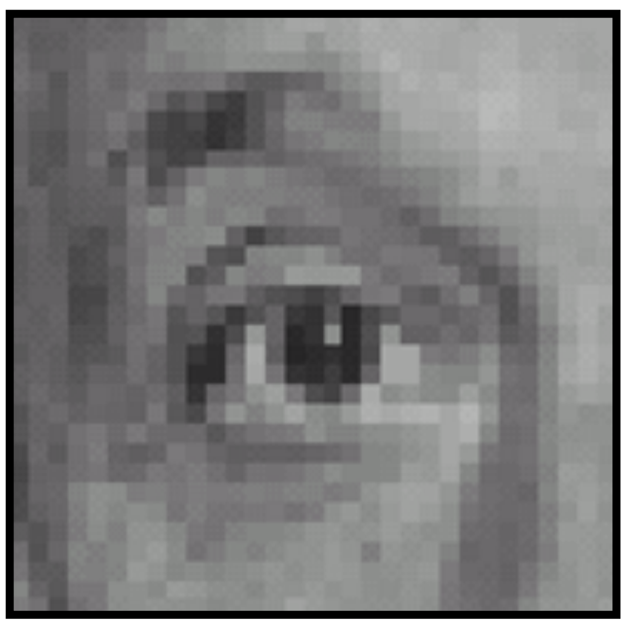
0	0	0
0	0	1
0	0	0

?

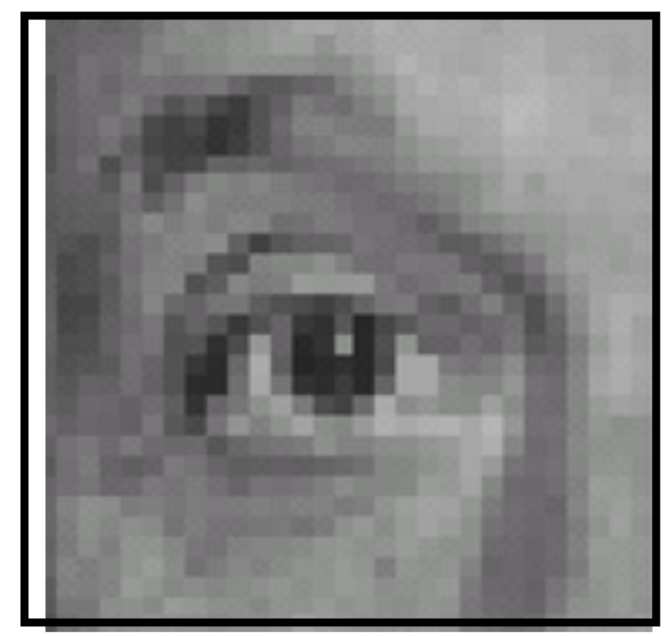
Source: D. Lowe

Convolution Practice

Translation Filter

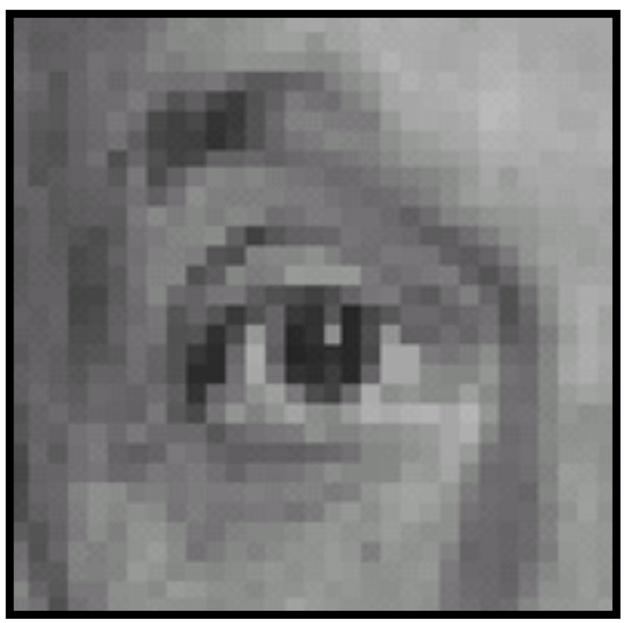


0	0	0
0	0	1
0	0	0



Source: D. Lowe

Convolution Practice



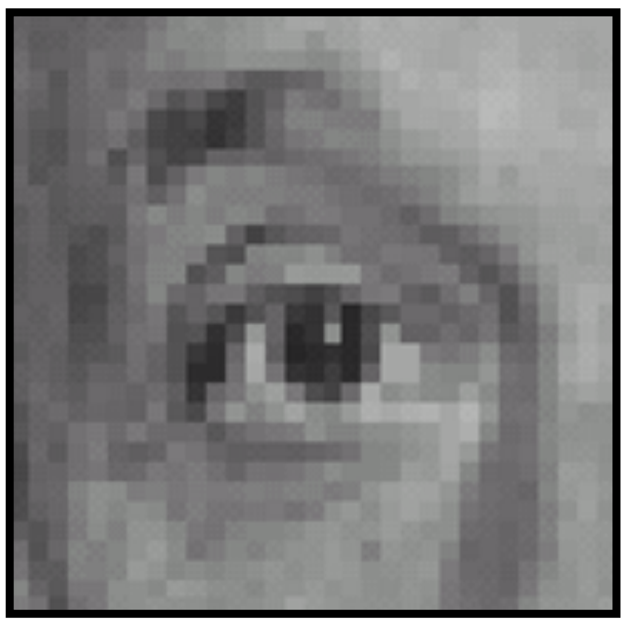
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

?

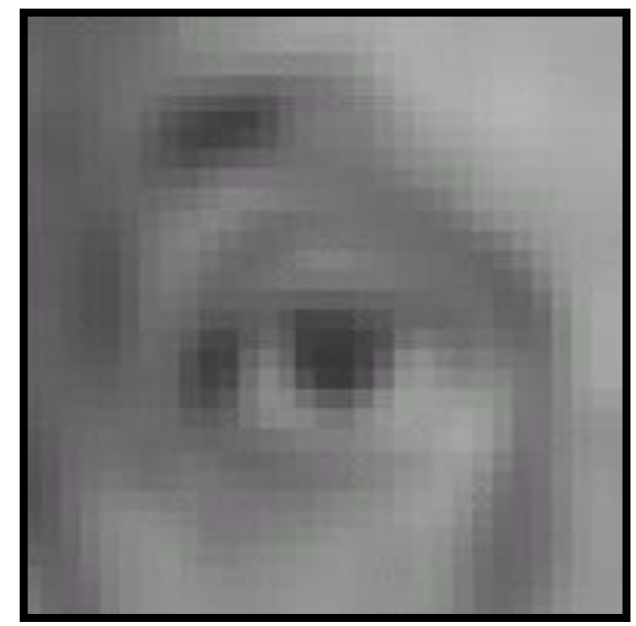
Source: D. Lowe

Convolution Practice

Blur Filter

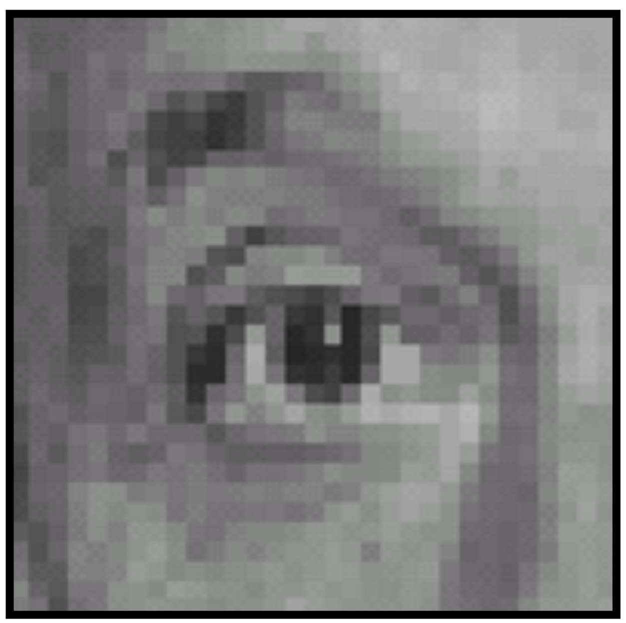


$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

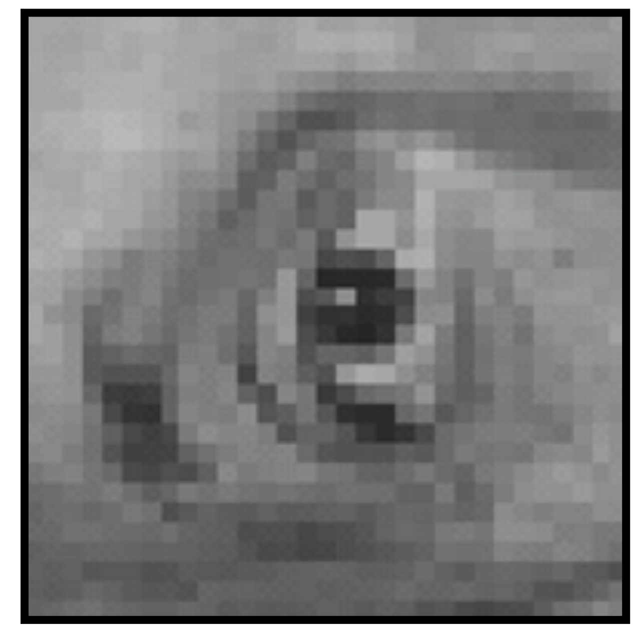


Source: D. Lowe

Convolution Practice

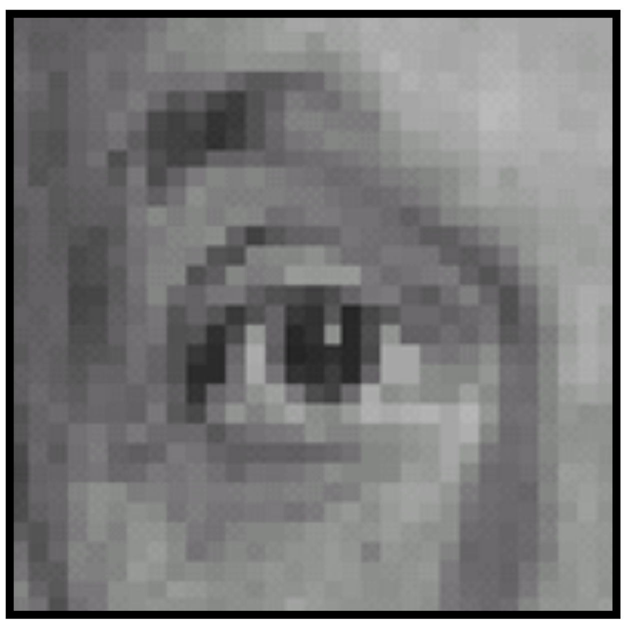


?



Source: D. Lowe

Convolution Practice



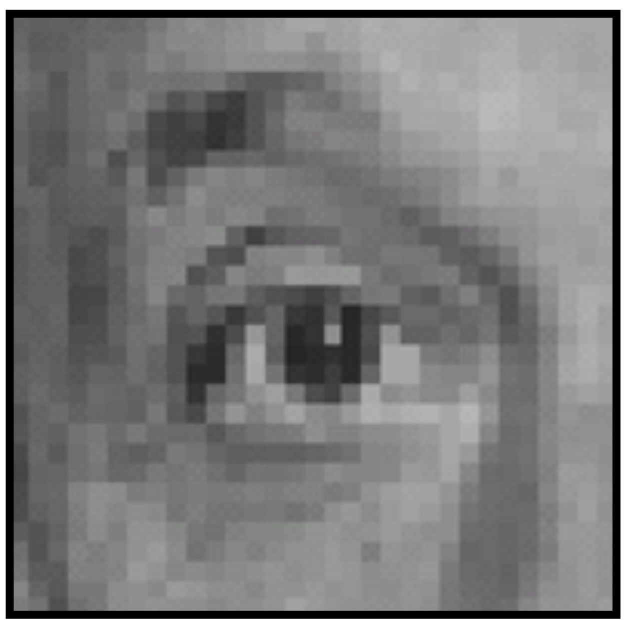
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

?

Source: D. Lowe

Convolution Practice

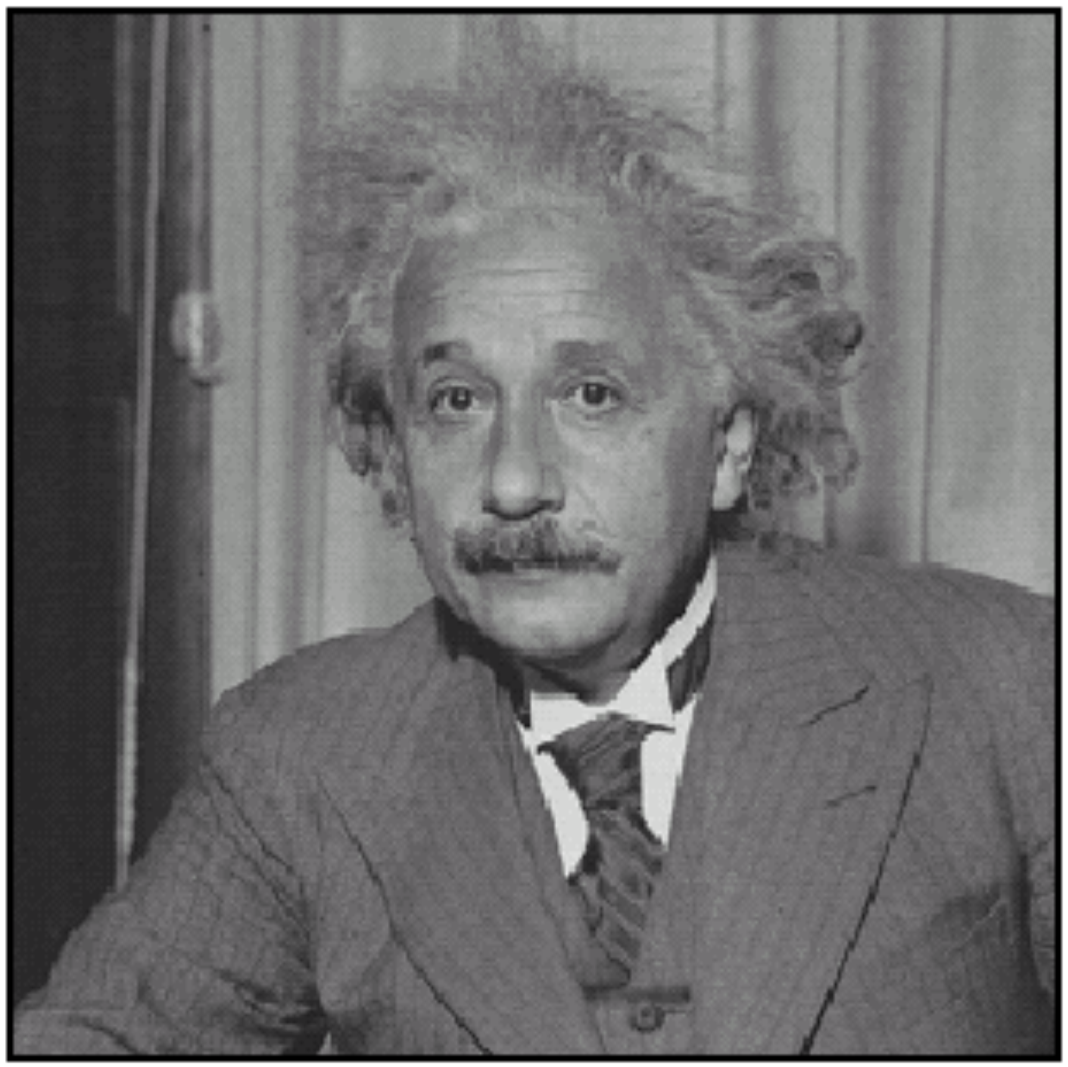
Sharpening Filter



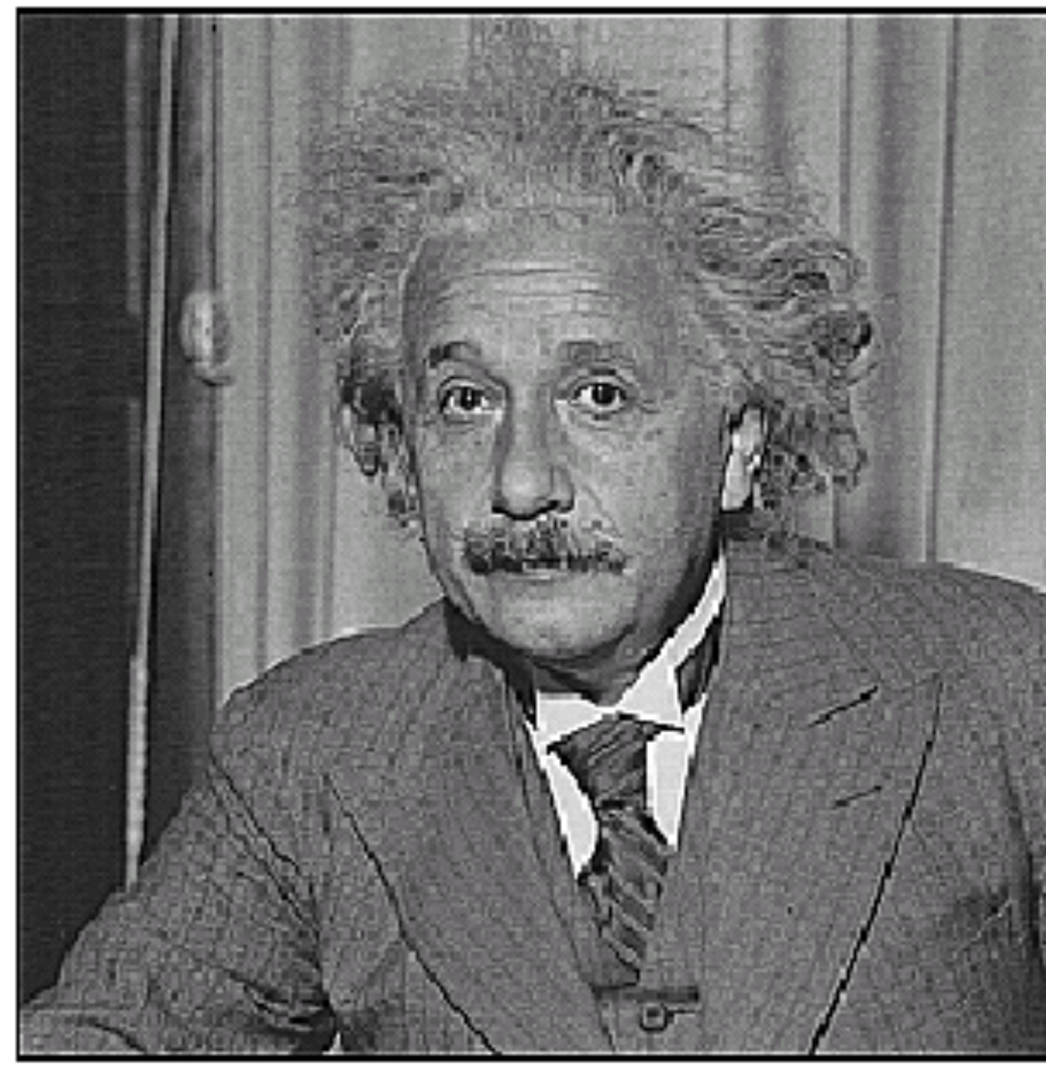
$$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} - \frac{1}{9} \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Source: D. Lowe

Sharpening



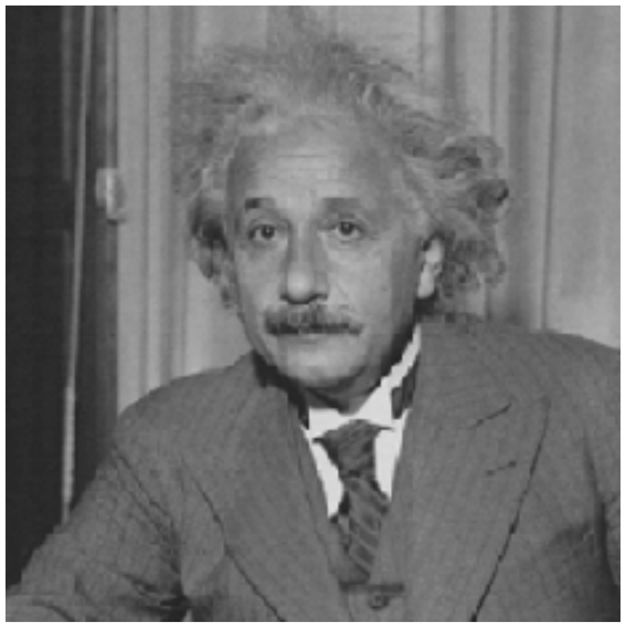
before



after

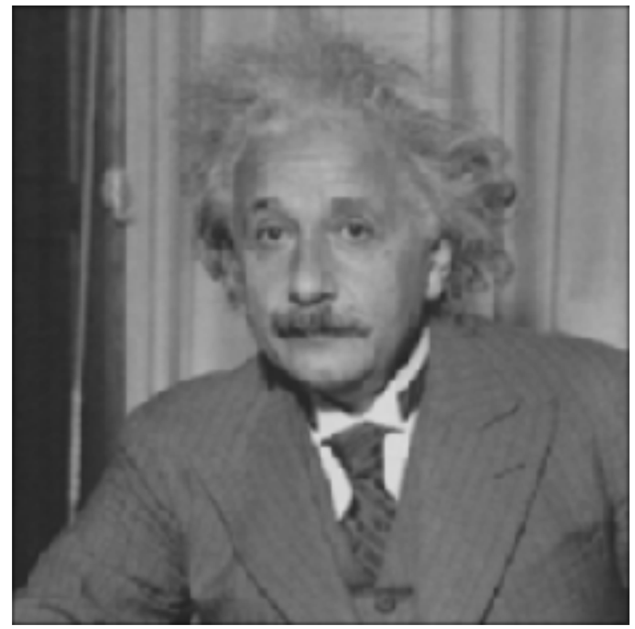
Source: D. Lowe

Sharpening



Image

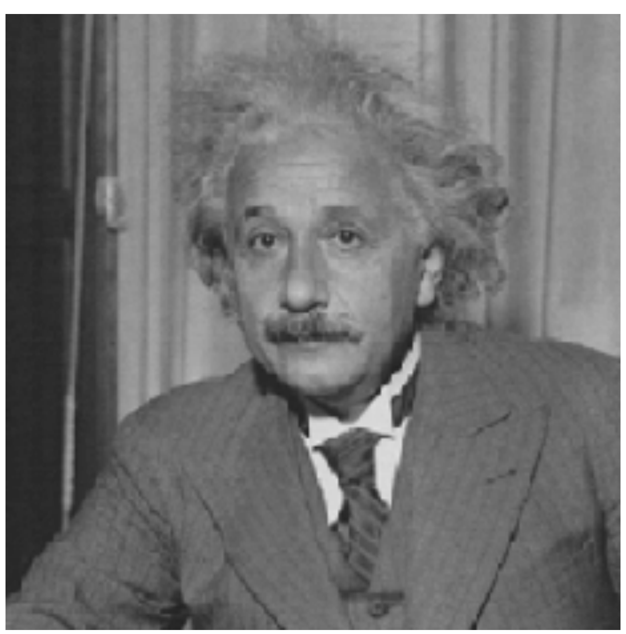
-



Blurred

=

Detail

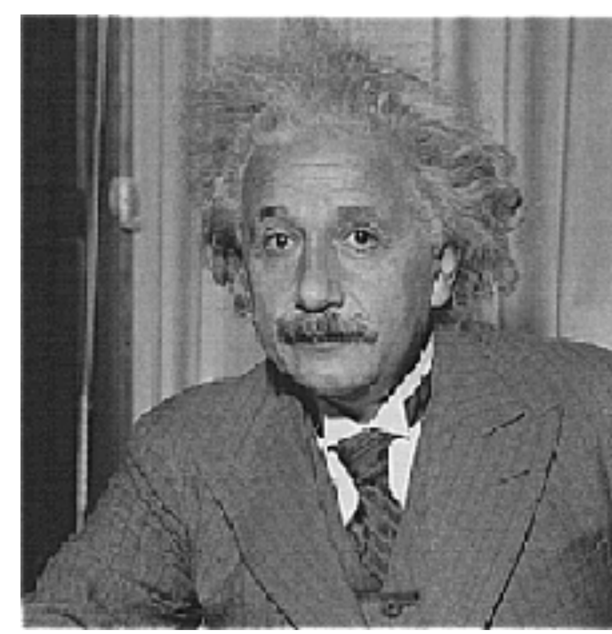


Image

+

Detail

=



Sharpened

Convolution Properties

Commutative:

$$F * H = H * F$$

Associative:

$$(F * H) * G = F * (H * G)$$

Distributive:

$$(F * G) + (H * G) = (F + H) * G$$

Convolution Properties

Scale:

filter(A * f) = A * filter(f)

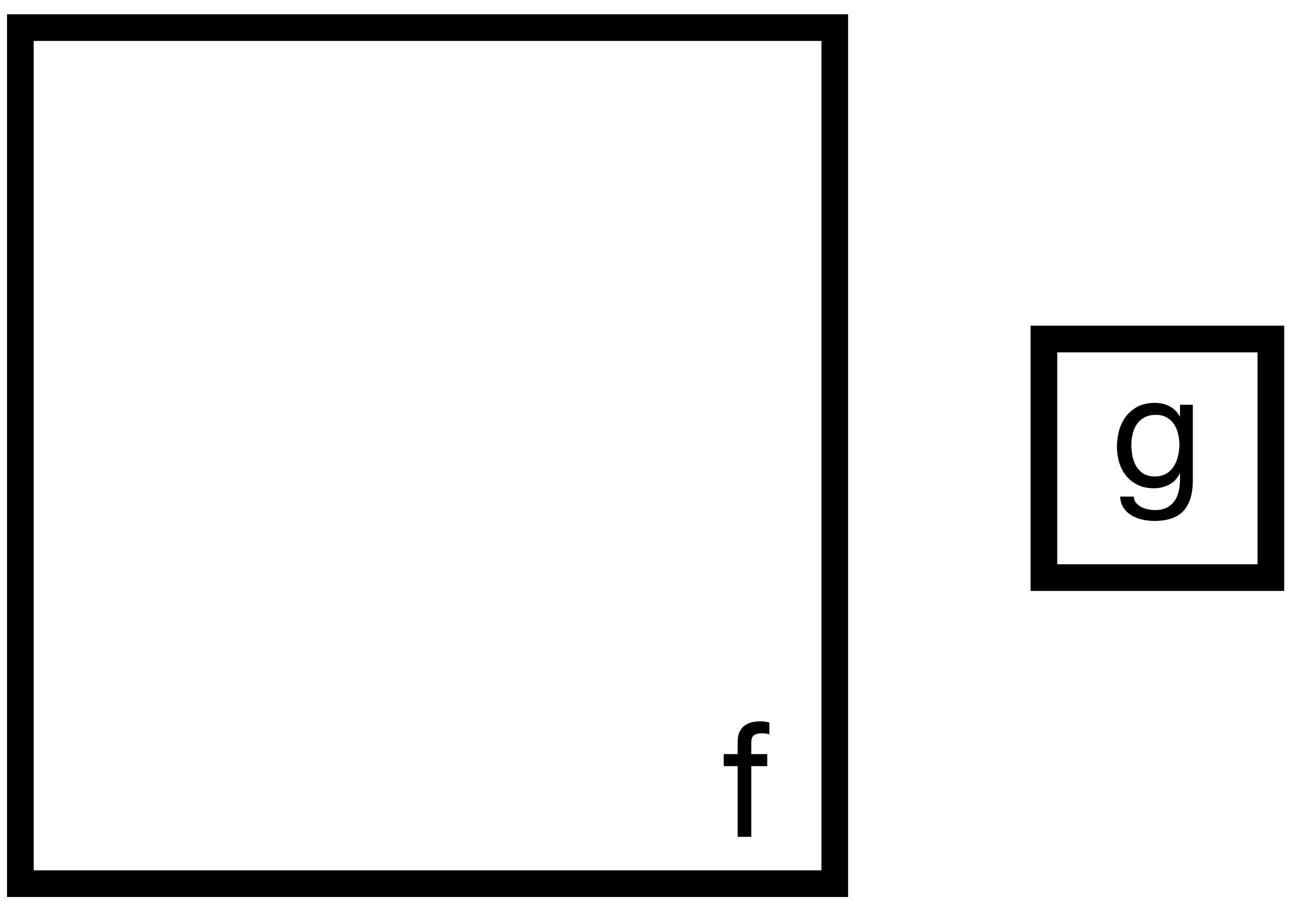
Shift Invariance:

filter(shift(f)) = shift(filter(f))

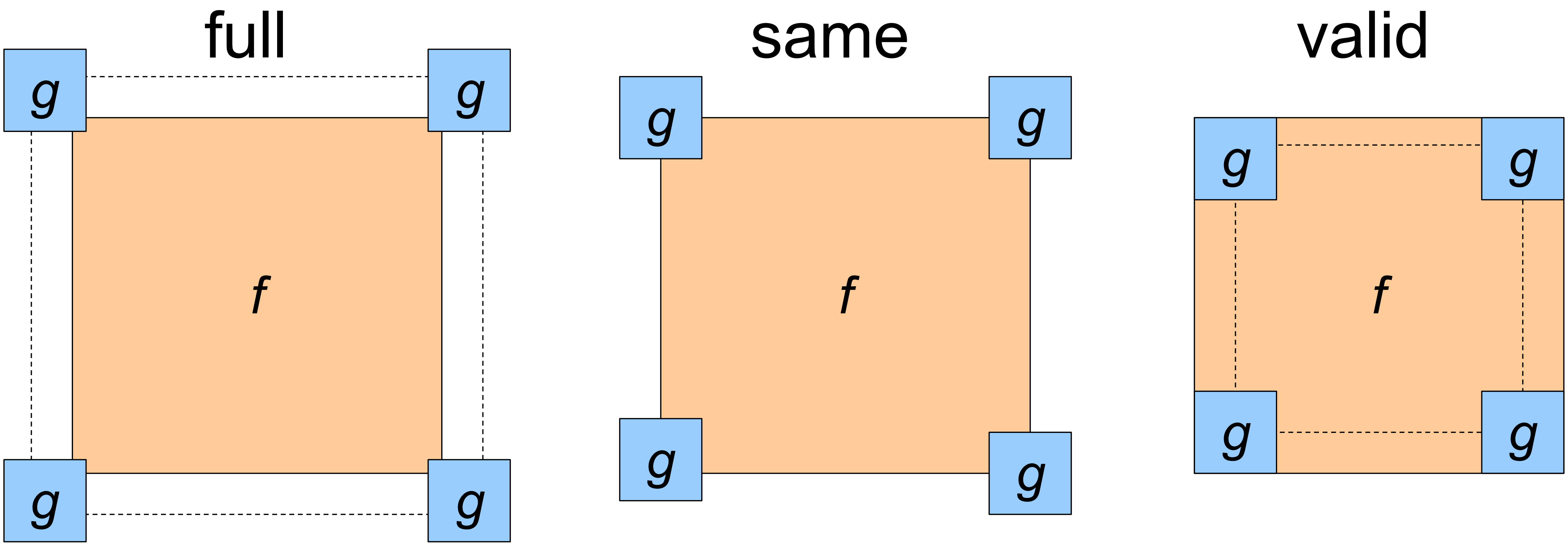
Cross-Correlation

- Conceptually simpler, but not as nice properties:

$$(f * g)[x, y] = \sum_{i,j} f[x+i, y+j]g[i,j]$$



Boundary Issues



Slide credit: S. Lazebnik

Border Padding

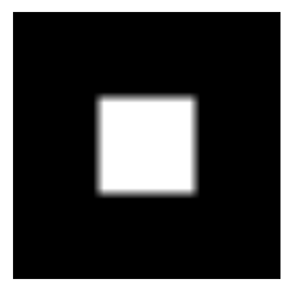
Zero Pad

Circular

Replicate

Symmetric

Box Filter

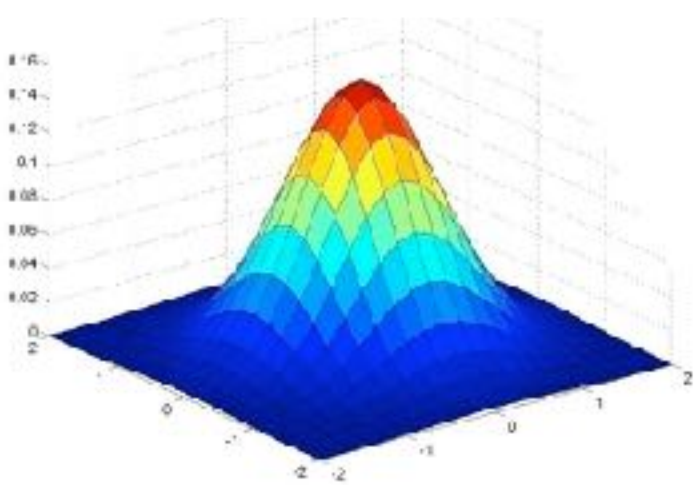


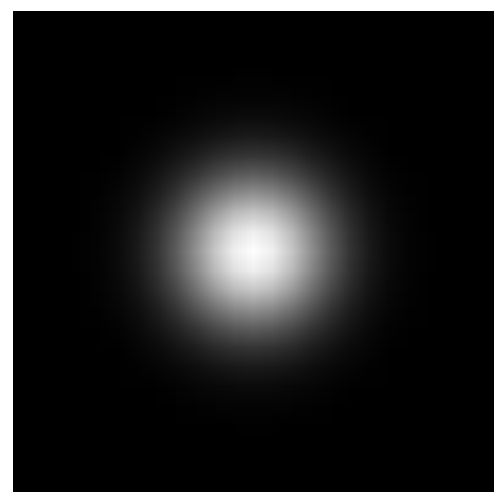
Gaussian Filter



Gaussian Filter

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





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

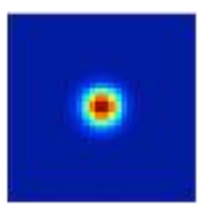
5 x 5, σ = 1

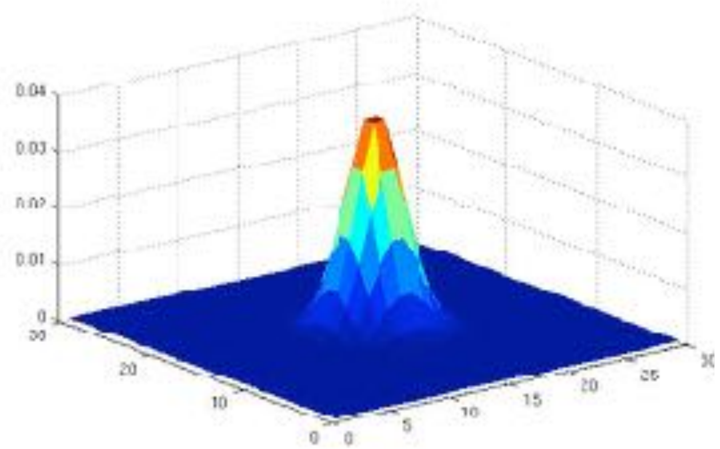
Constant factor at front makes volume sum to unity

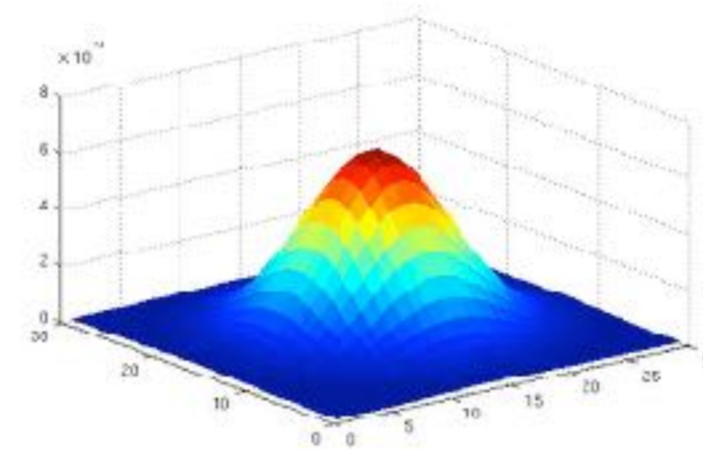
Source: C. Rasmussen

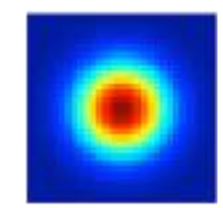
Standard Deviation

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









σ = 2 with
30 x 30
kernel

σ = 5 with
30 x 30
kernel

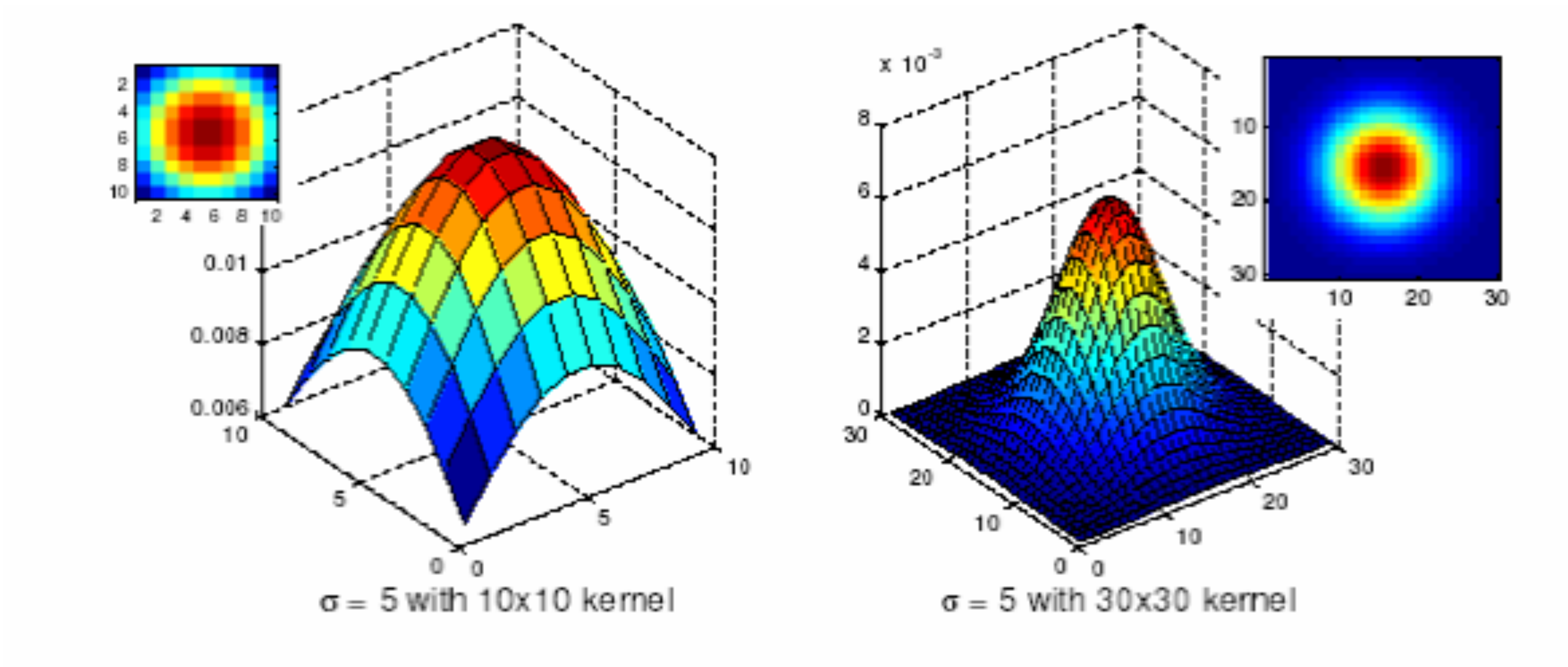
Standard deviation σ: determines extent of smoothing

Source: K Grauman

Changing Sigma

Kernel Width

The Gaussian function has infinite support, but discrete filters use finite kernels



Rule of thumb: set filter half-width to about 3σ

Source: K Grauman

Complexity

What is the complexity of filtering an $n \times n$ image with an $m \times m$ kernel?

Complexity

What is the complexity of filtering an $n \times n$ image with an $m \times m$ kernel?

$O(n^2 m^2)$

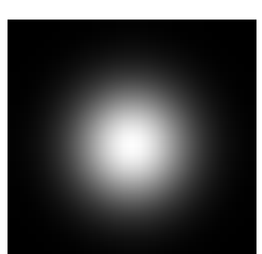
Separable Convolution

Two dimensional Gaussian is product of two Gaussians:

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

$$= \left(\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)\right)\left(\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{y^2}{2\sigma^2}\right)\right)$$

Take advantage of associativity:

$f\,*$  $= f\,*$  $*$ 

Complexity

What is the complexity of filtering an $n \times n$ image with an $m \times m$ kernel?

$O(n^2 m^2)$

What if kernel is separable?

Complexity

What is the complexity of filtering an $n \times n$ image with an $m \times m$ kernel?

$O(n^2 m^2)$

What if kernel is separable?

$O(n^2 m)$

Denoising

Additive Gaussian Noise

Gaussian Filter (sigma=1)

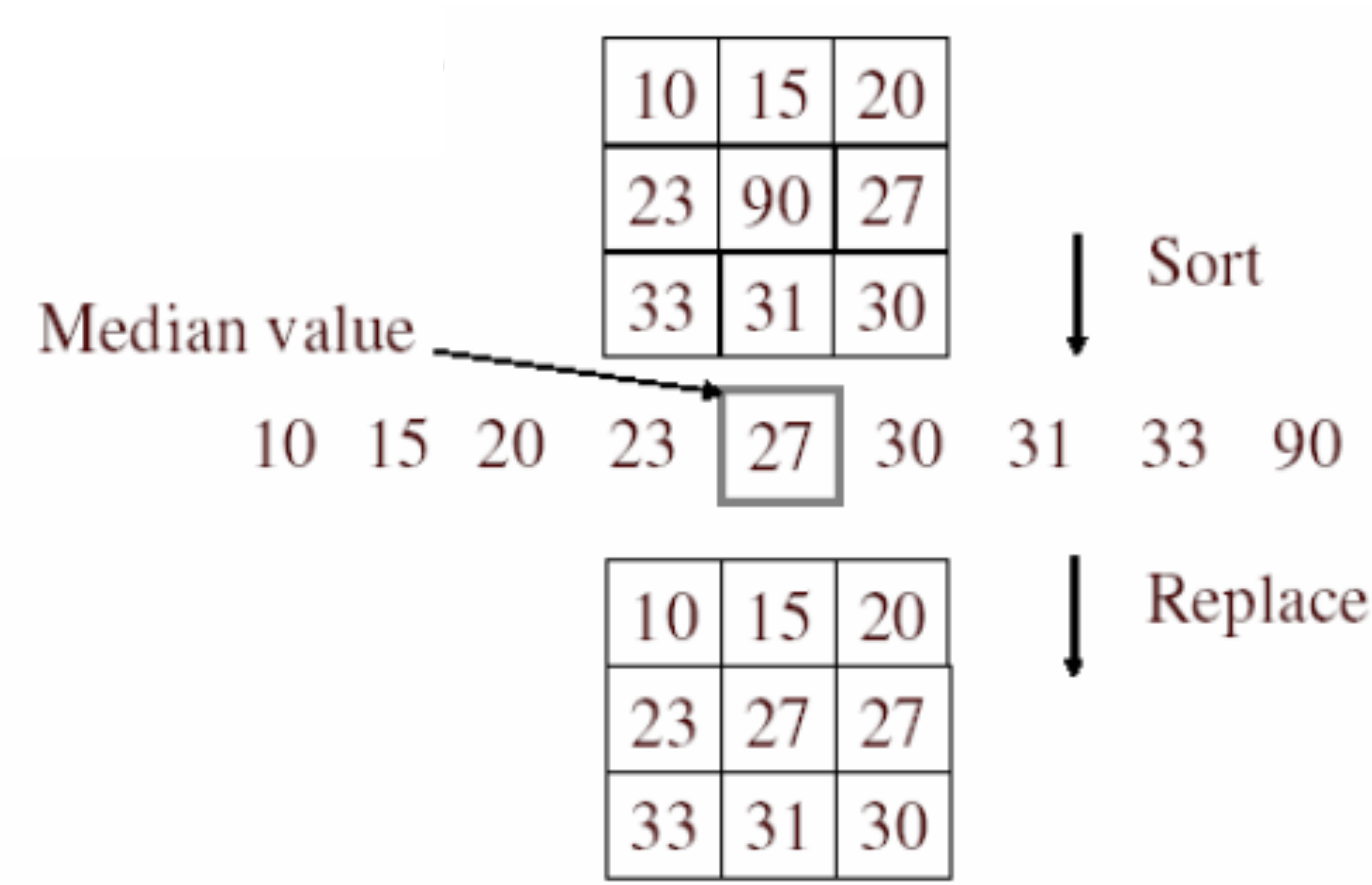
What’s wrong?

Salt and Pepper Noise

Gaussian Filter (sigma=1)

Median Filter

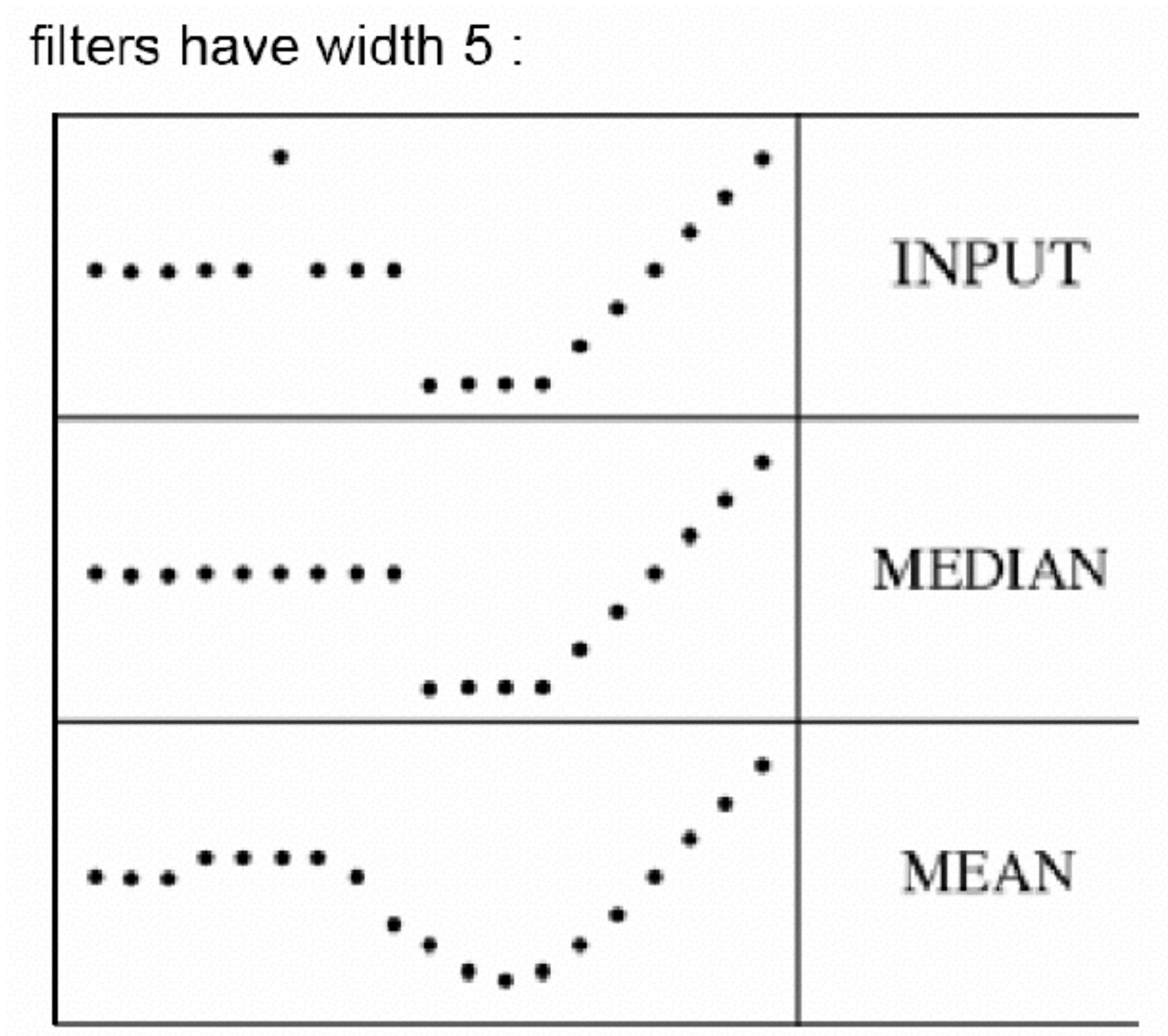
- A **median filter** operates over a window by selecting the median intensity in the window



Is median filtering linear?

Source: Kristen Grauman

Why use median?



Source: Kristen Grauman

What's wrong?

Salt and Pepper Noise

Median Filter 3x3

Median Filtering

Median 3x3

Median 5x5

Median 9x9

Image Gradients

Image Gradients

How does intensity change as you move left to right?

How do you take the ***derivative*** *of an image?*

First Derivative

$$* [-1,1] =$$

$$\frac{\partial I}{\partial x}$$

$$* [-1,1]^T =$$

$$\frac{\partial I}{\partial y}$$

Second Derivative

$$* [-1,1] =$$

$$\frac{\partial^2 I}{\partial x^2}$$

$$* [-1,1]^T =$$

$$\frac{\partial^2 I}{\partial y^2}$$

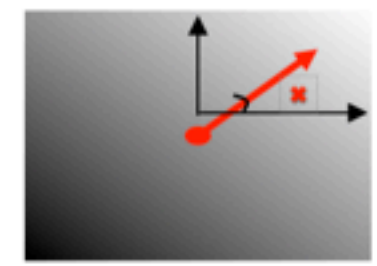
Image Gradients

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

It points in the direction of most rapid change 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 gradient direction is given by:

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

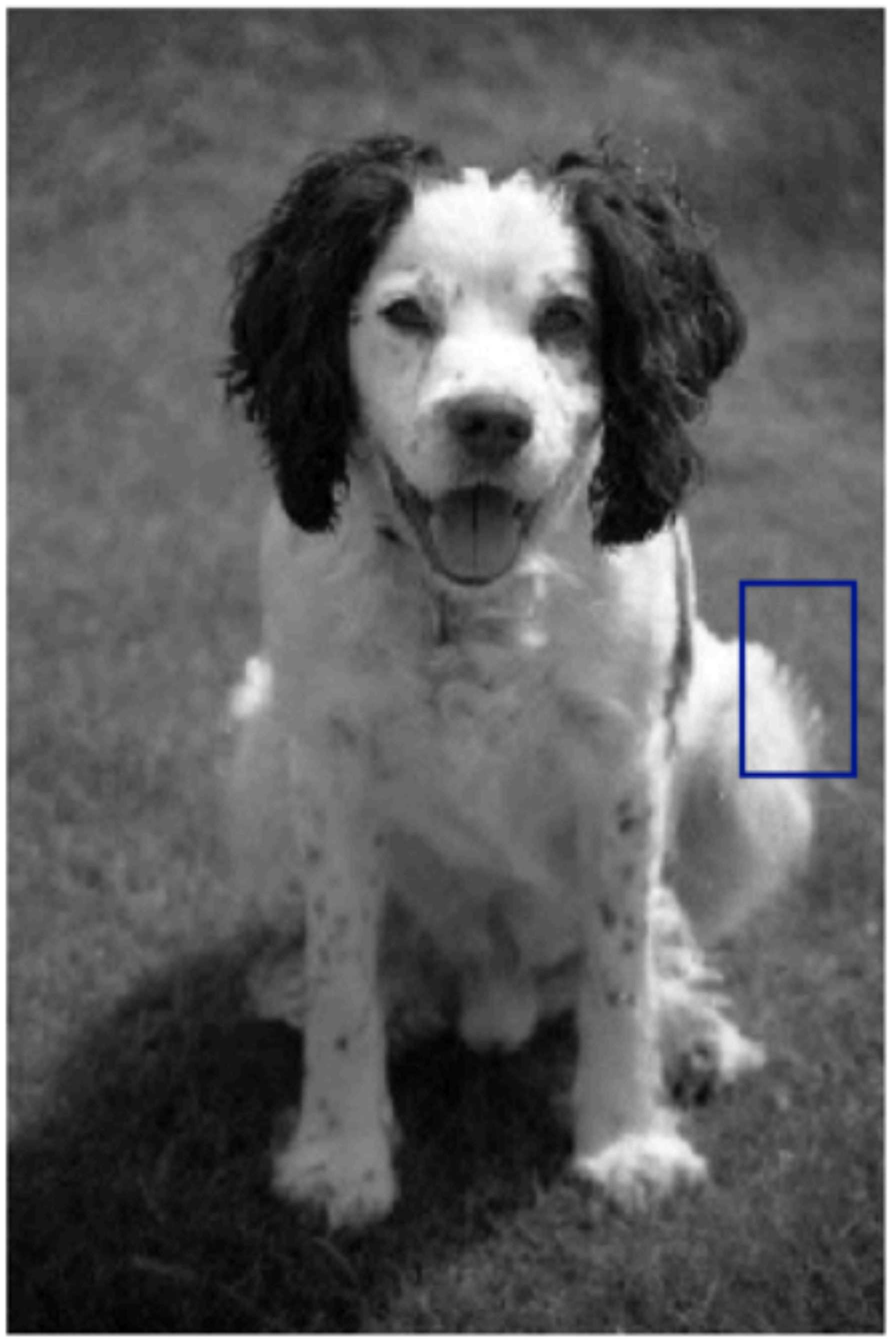
- how does this relate to the direction of the edge?

The *edge strength* is given by the gradient magnitude

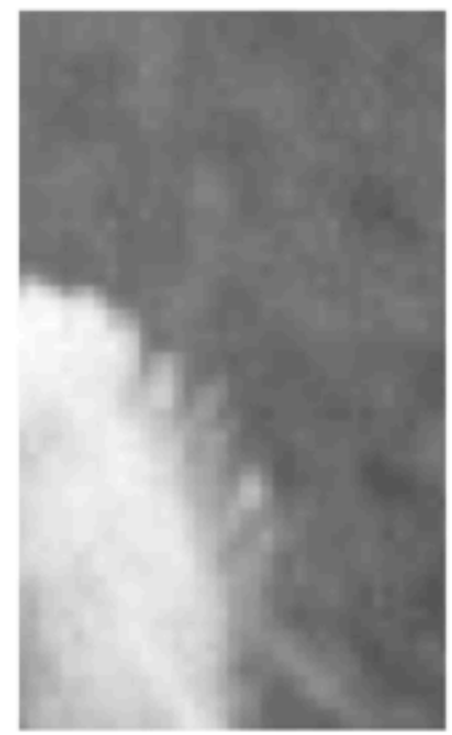
$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Source: Seitz and Szeliski

What causes an edge?



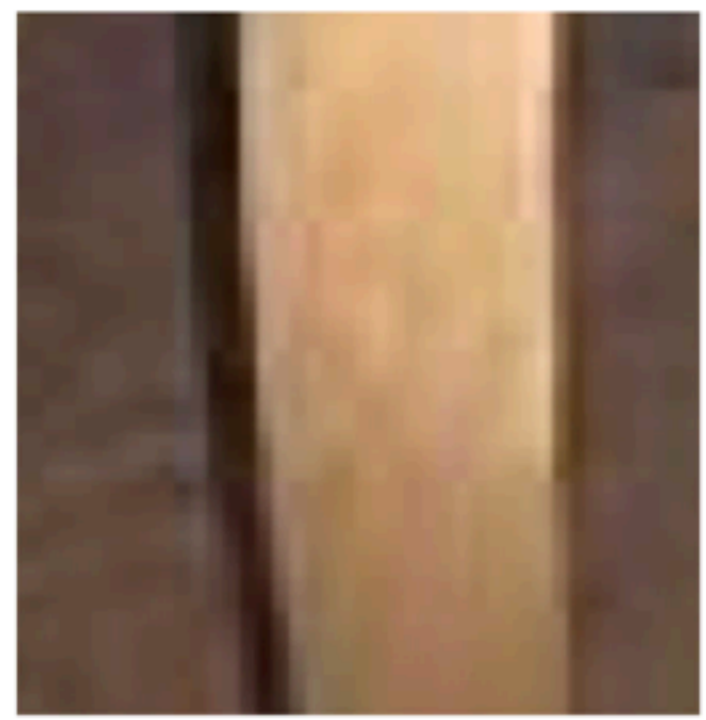
Object Boundaries



Source: G Hager

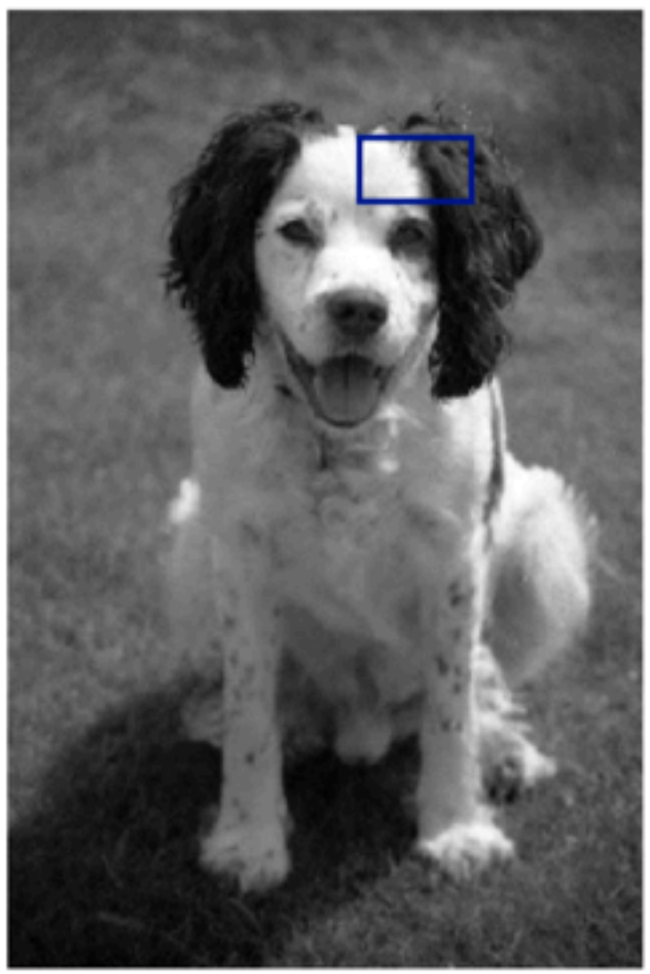
What causes an edge?

Surface normal discontinuities



Source: G Hager

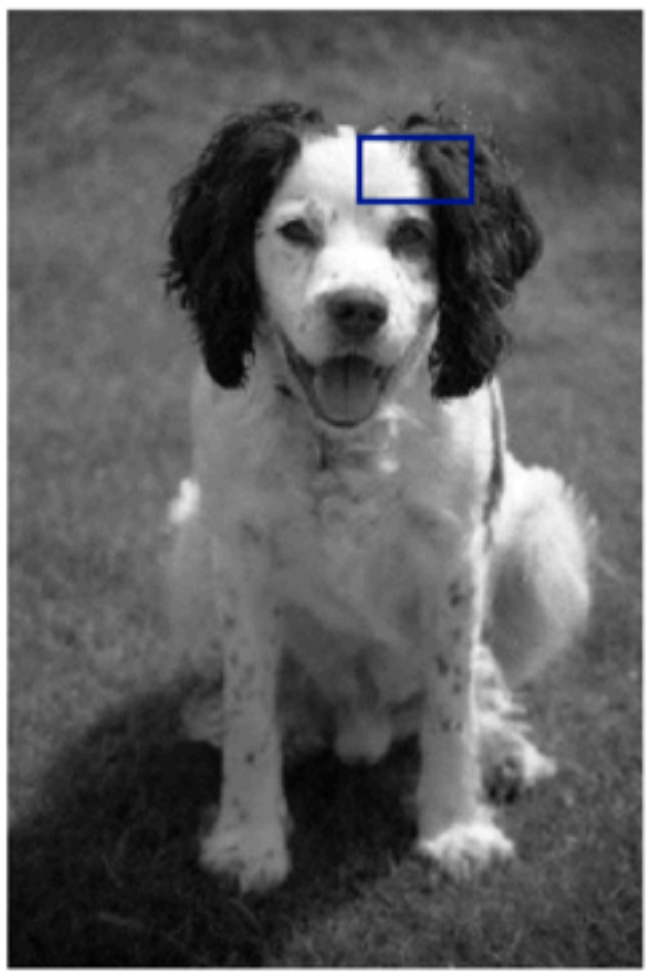
What causes an edge?



Boundaries of material properties

Source: G Hager

What causes an edge?

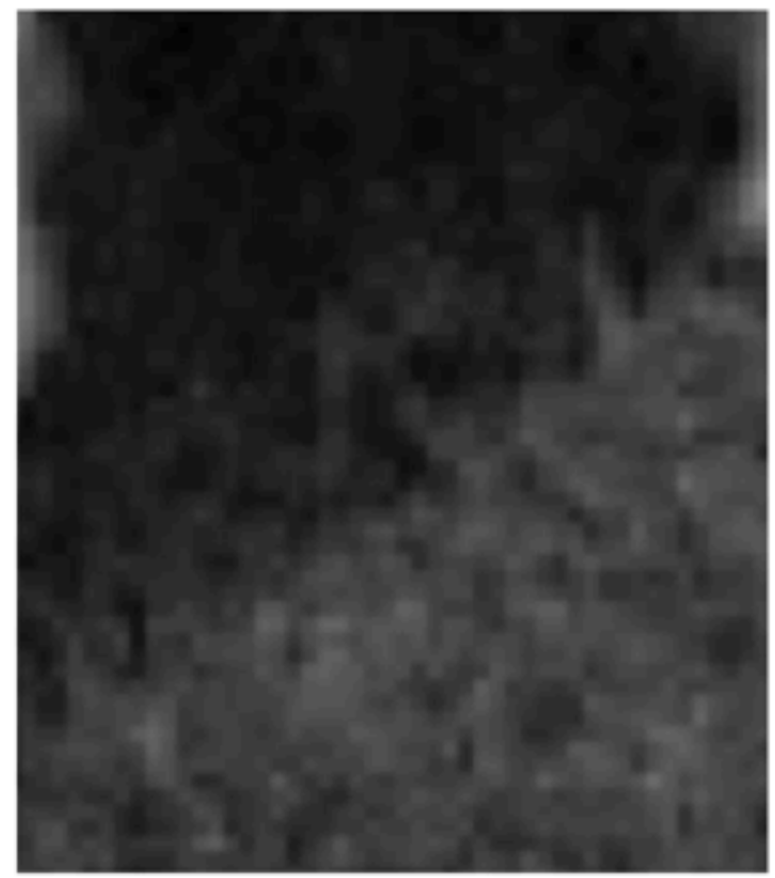


Boundaries of material properties

Source: G Hager

What causes an edge?

Boundaries of lighting



Source: G Hager

Edge Types

Step

Ridge

Which of these do you suppose a derivative filter detects best?

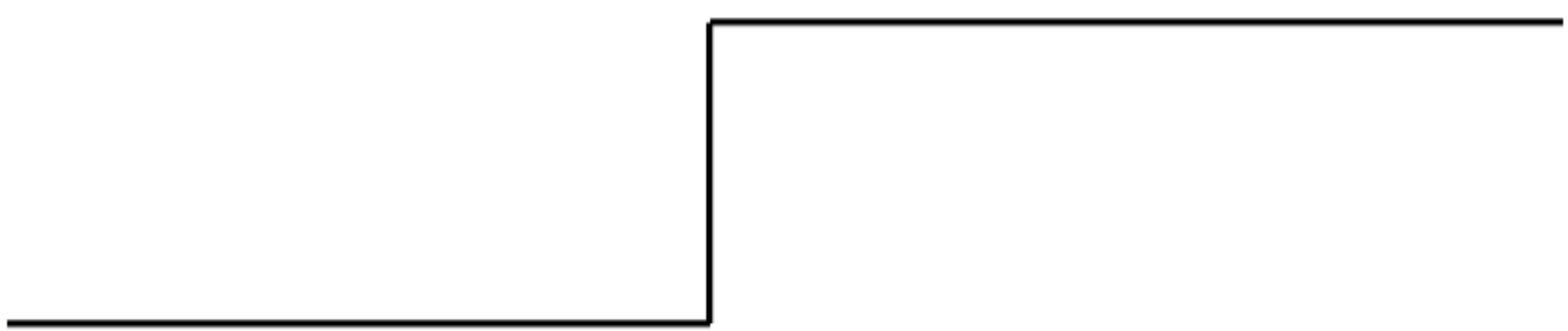
Roof

Source: G Hager

What is an edge?

Change is measured by derivative in 1D

- Biggest change, derivative has maximum magnitude
- Or 2[nd] derivative is zero.

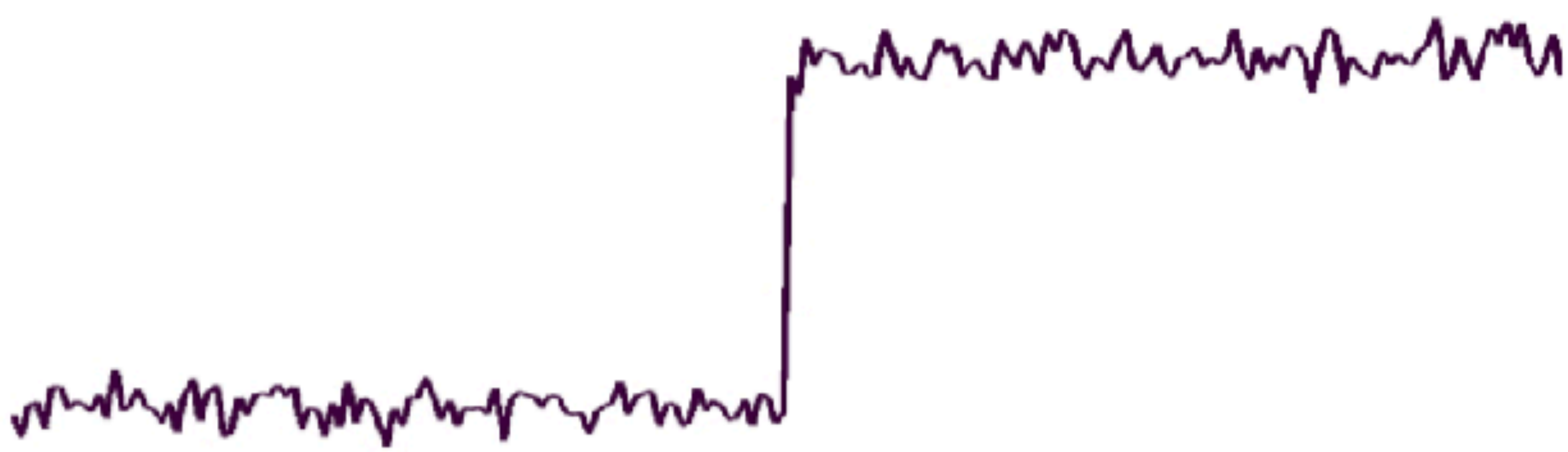


Source: G Hager

What about noise?

Derivative is high everywhere.

Must smooth before taking gradient.



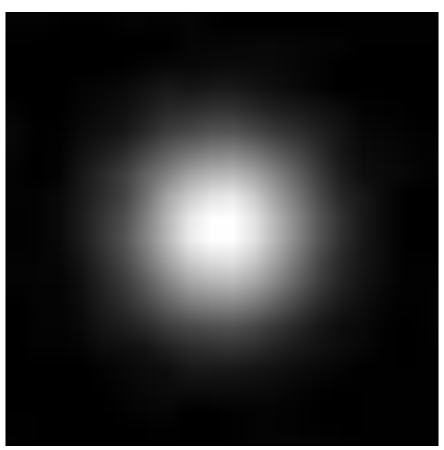
Source: G Hager

Handling Noise

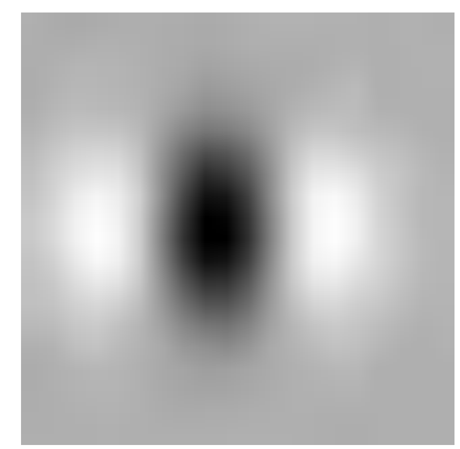
- Filter with a Gaussian to smooth, then take gradients
- But, convolution is linear

Gaussian Filter

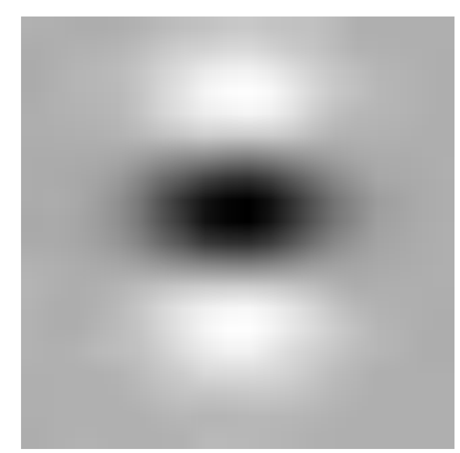
Laplacian Filter



$$* [-1,1] * [-1,1] =$$



$$* [-1,1]^T * [-1,1]^T =$$

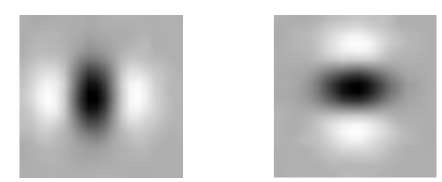


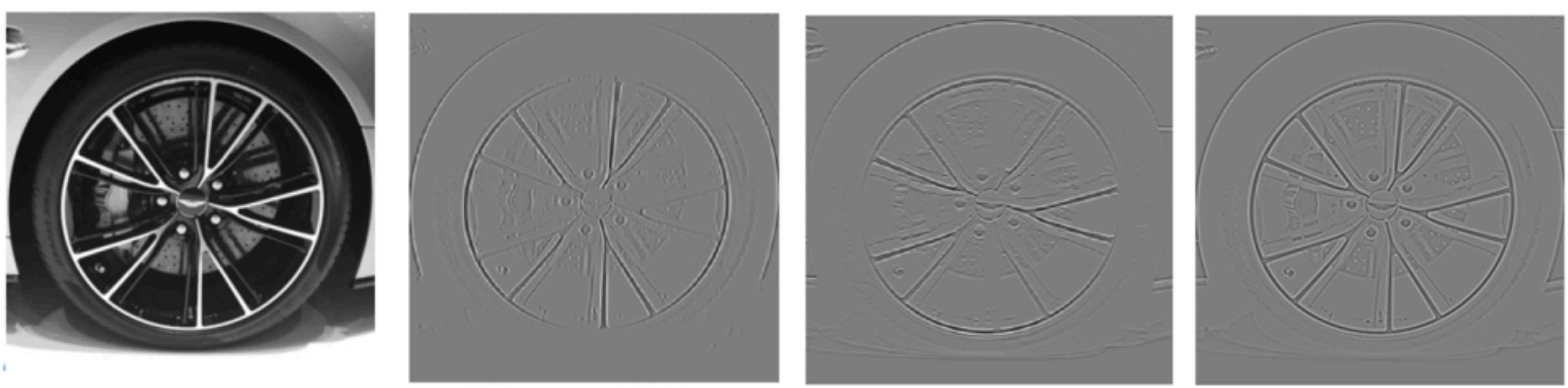
The Laplacian Filter

- Popularized by Marr and Hildreth in 1980 to locate boundaries between objects

- Defined as the sum of second order partial derivatives:

$$\nabla I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$



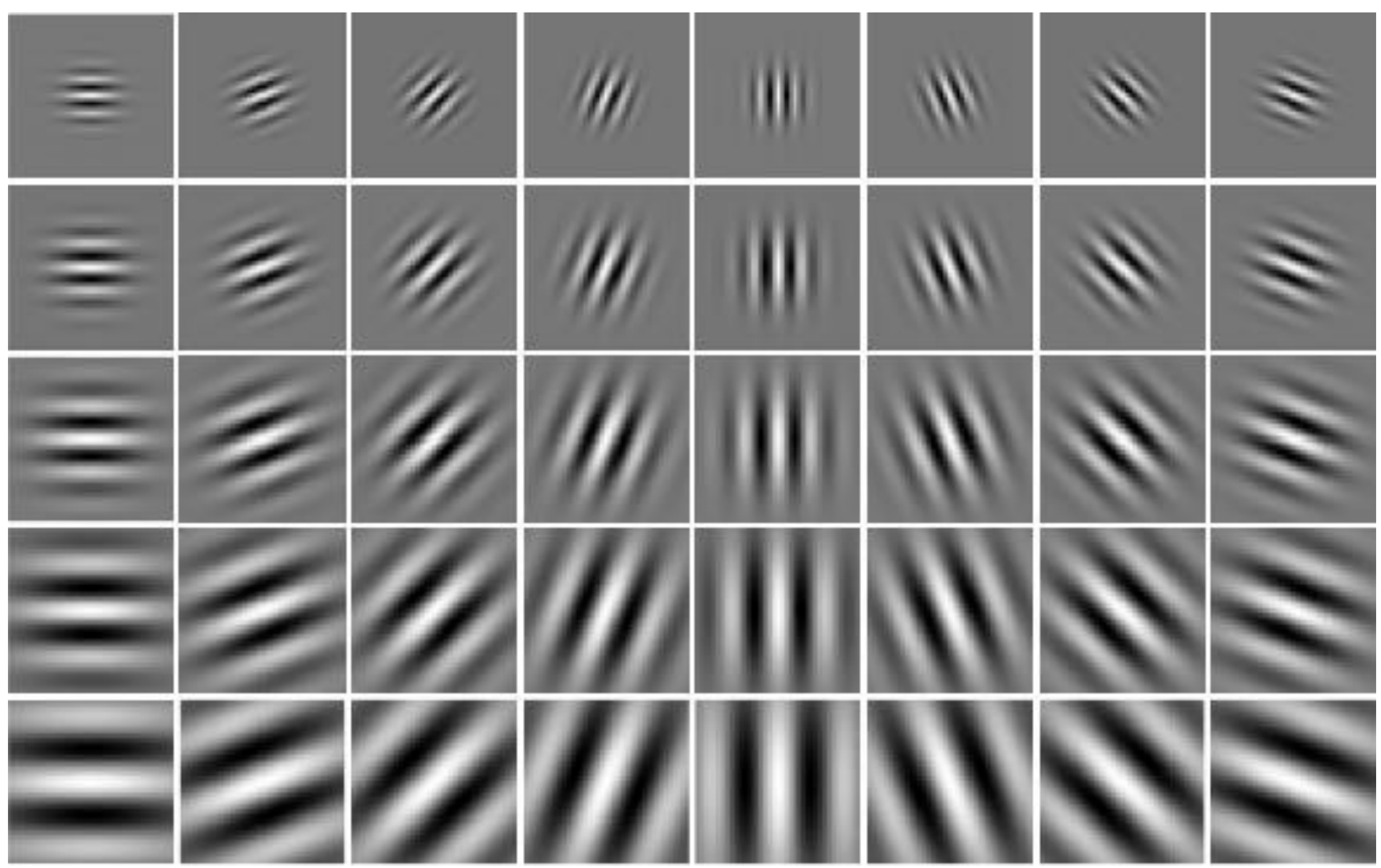


dx dy laplacian

Aside: Gabor Filters

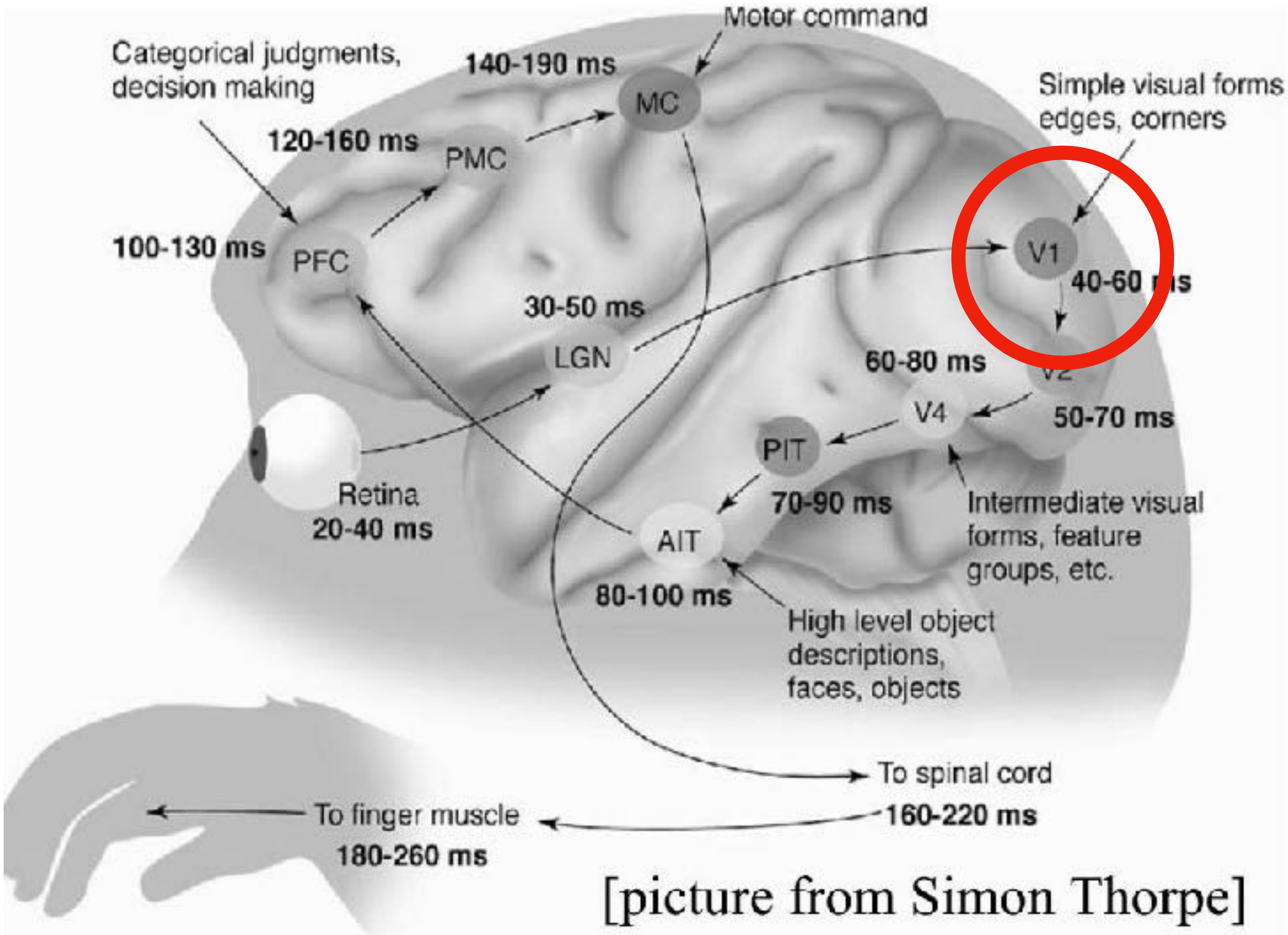
Cosine wave multiple by a Gaussian

$$\psi(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}} cos(2\pi\mu x)$$



Source: MathWorks

Aside: Human Visual System



[picture from Simon Thorpe]

Aside: Cat Visual System

IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. 36, NO. 7, JULY 1988

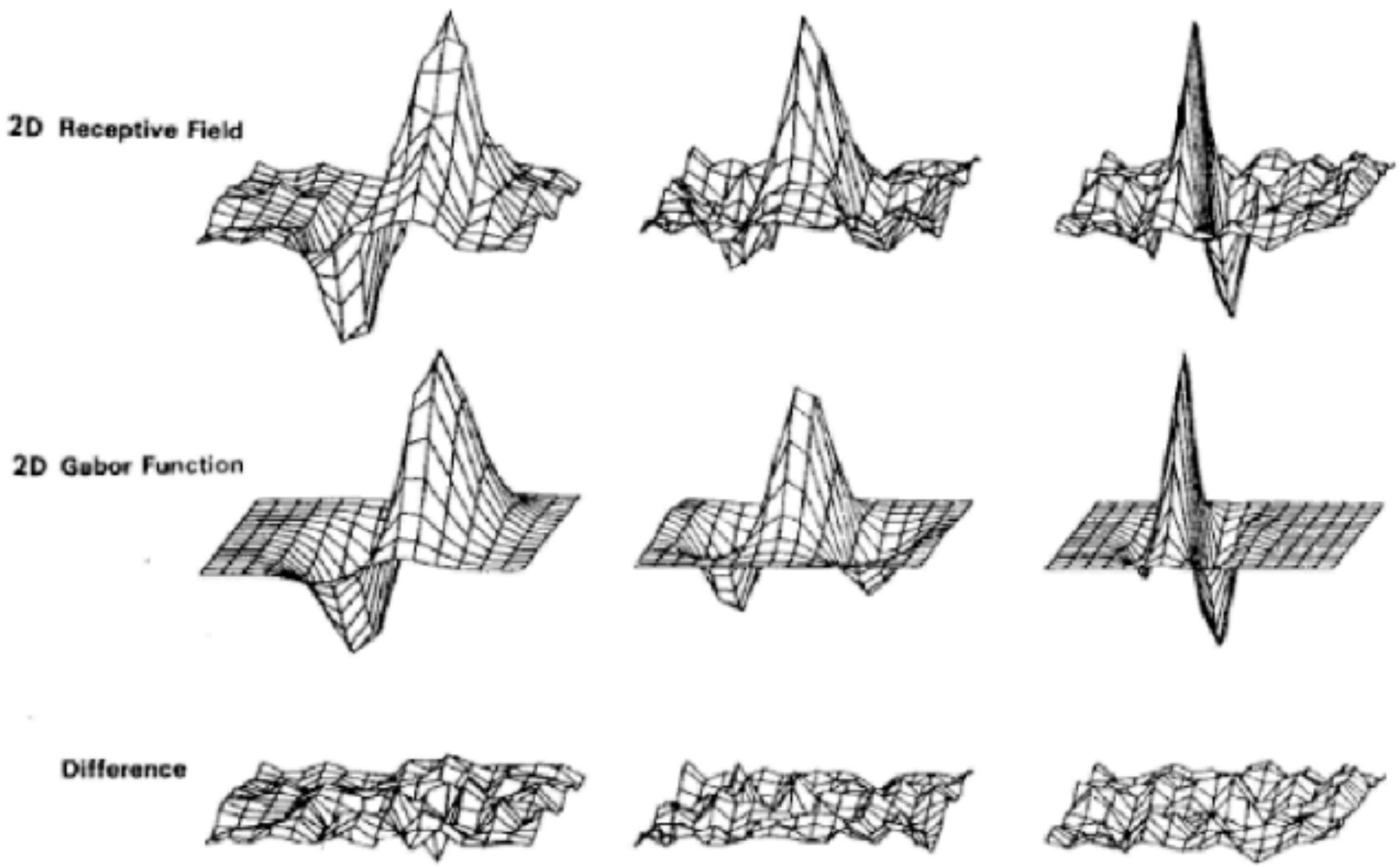


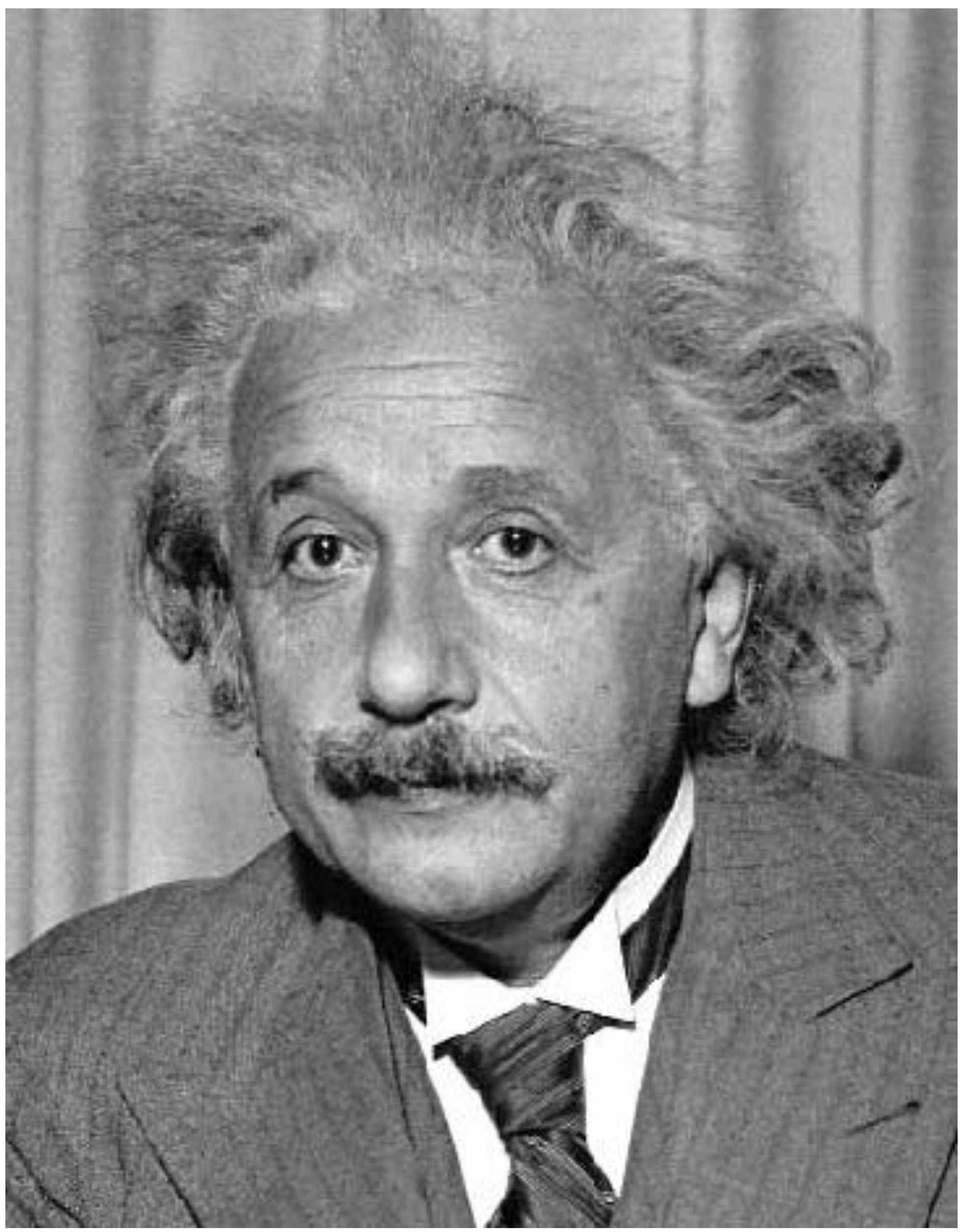
Fig. 5. Top row: illustrations of empirical 2-D receptive field profiles measured by J. P. Jones and L. A. Palmer (personal communication) in simple cells of the cat visual cortex. Middle row: best-fitting 2-D Gabor elementary function for each neuron, described by (10). Bottom row: residual error of the fit, indistinguishable from random error in the Chi-squared sense for 97 percent of the cells studied.

Source: Antonio Torralba

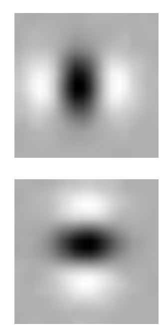
Detection

Finding Boundaries

f



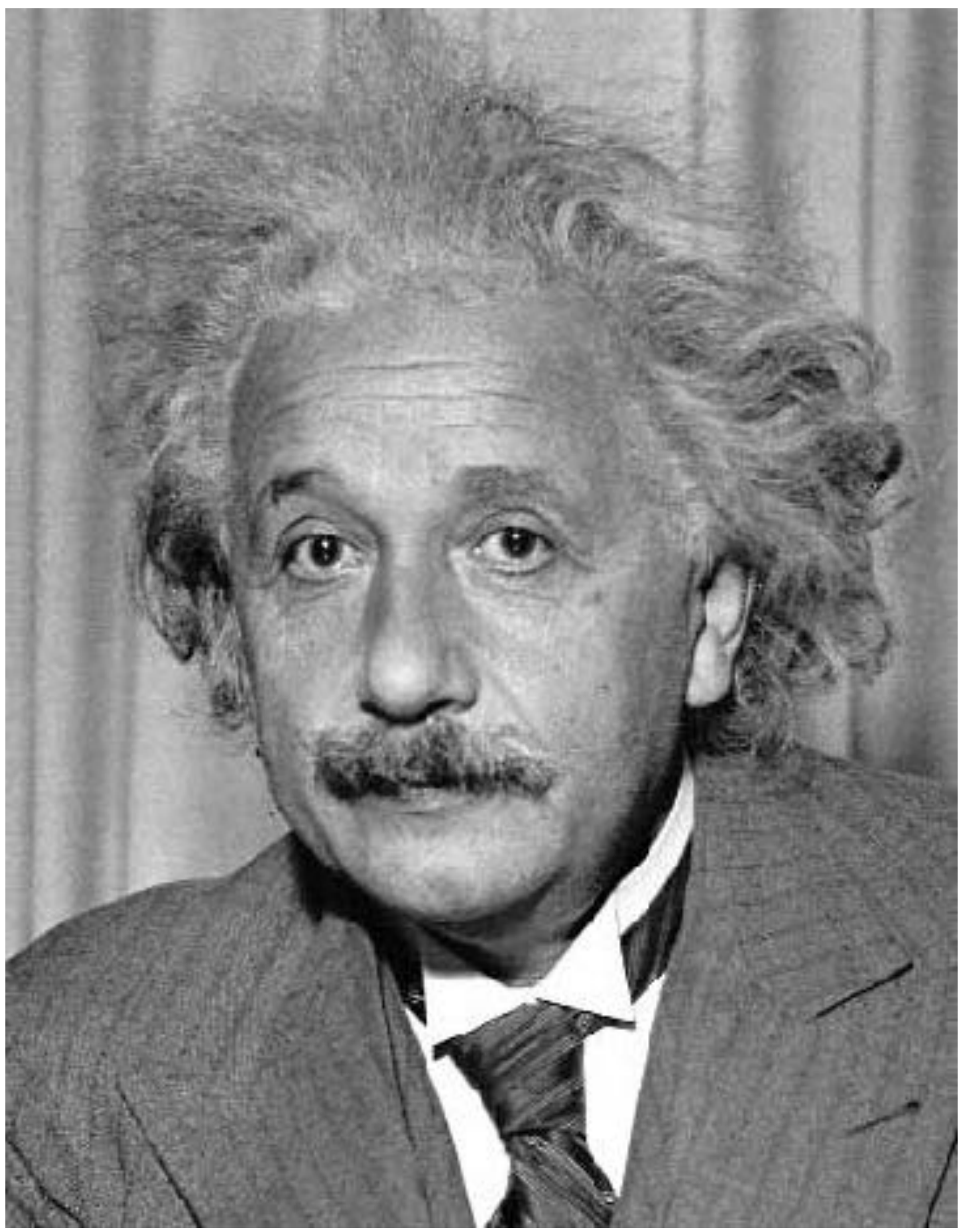
g



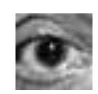
$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} > \lambda$$

Finding Things

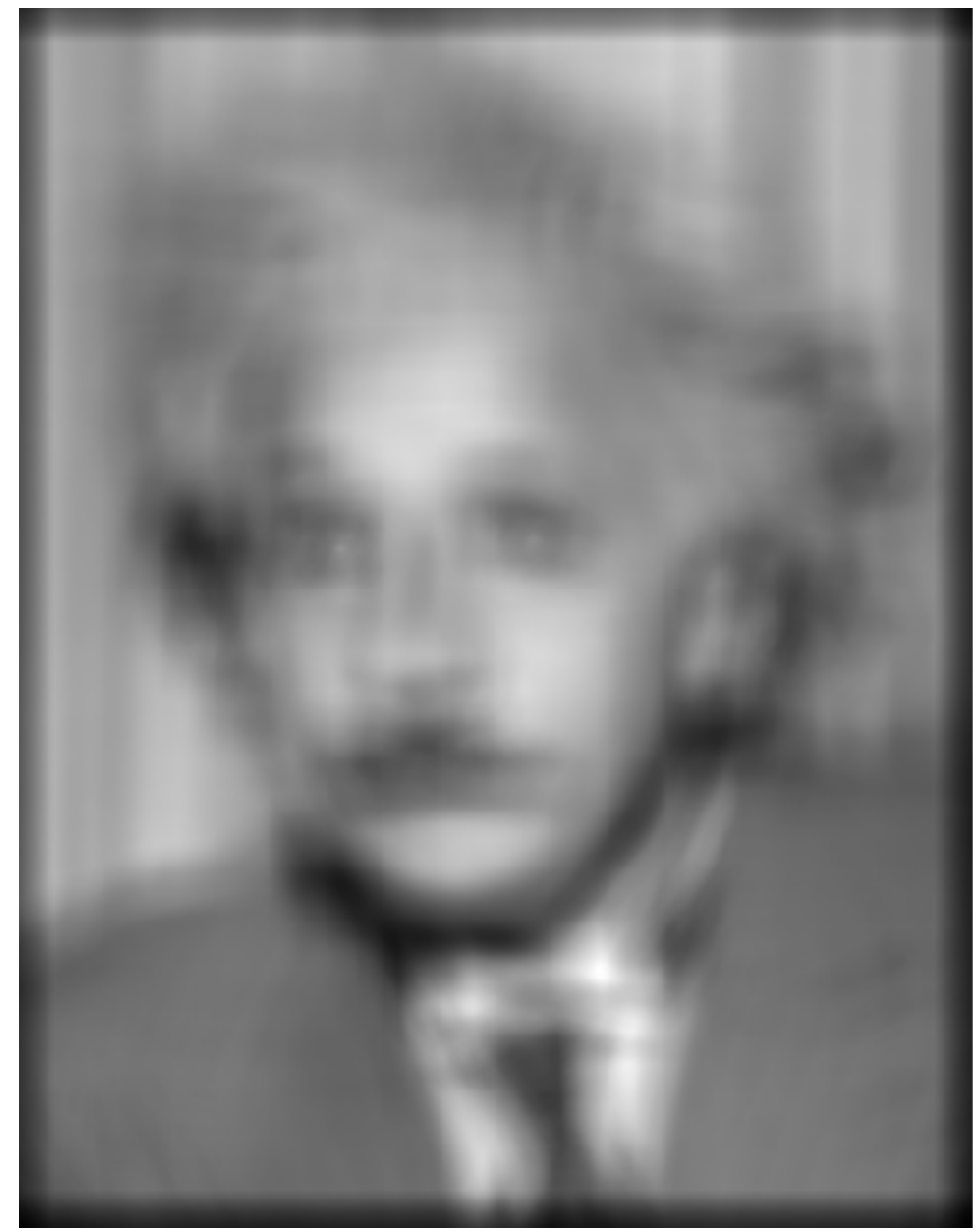
f



g



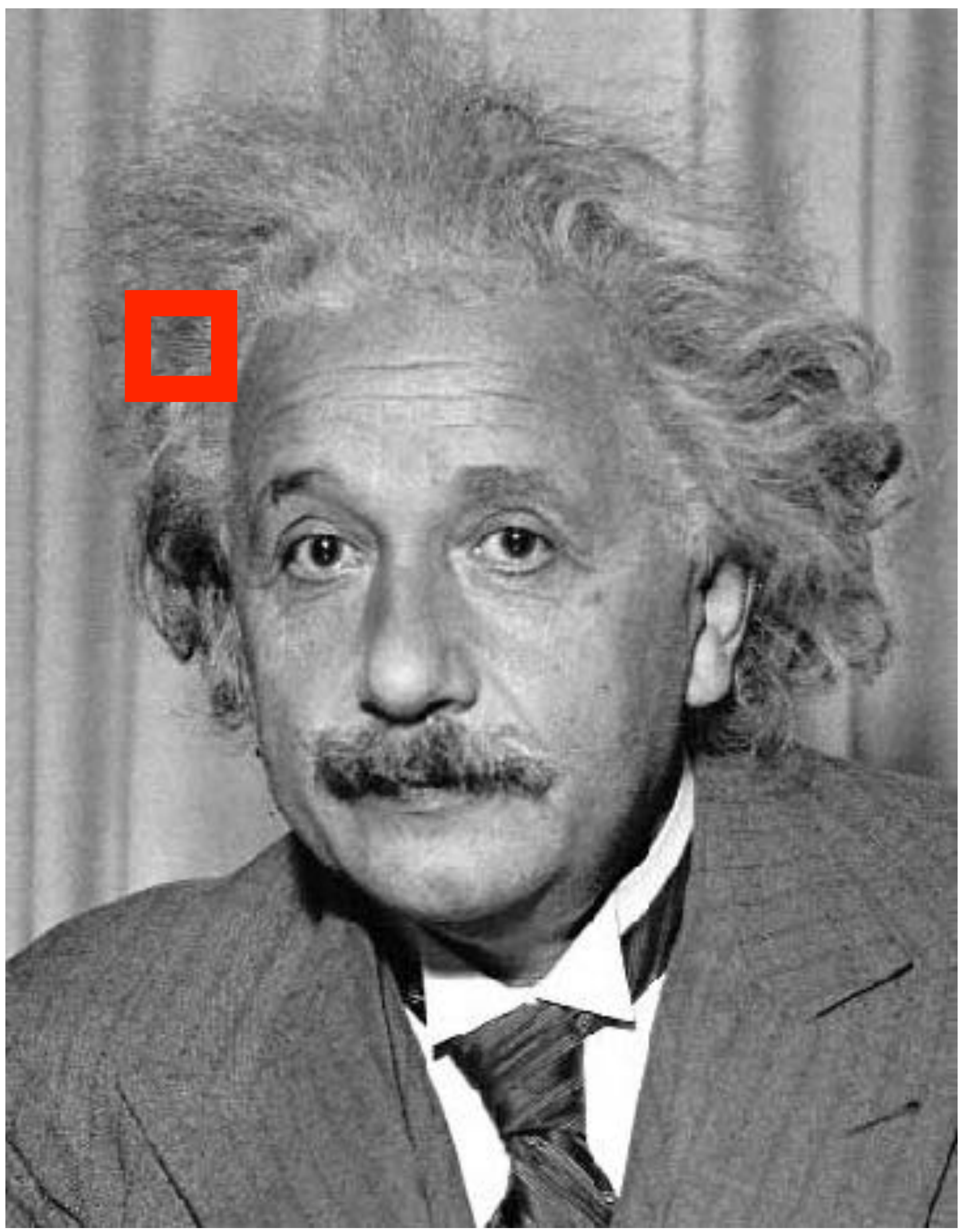
$f * g$



Source: James Hays, Deva Ramanan

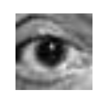
Detection by Filtering

f



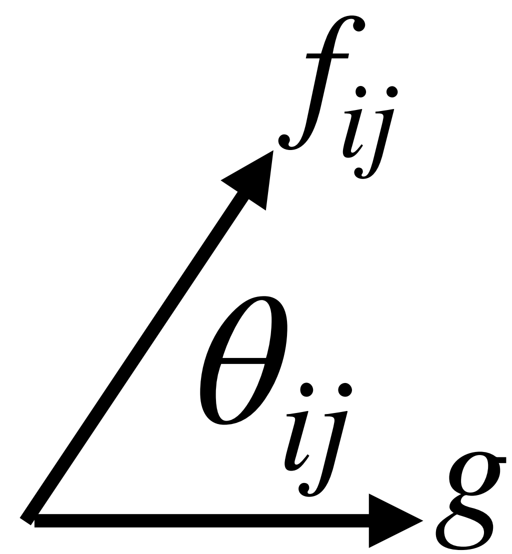
f_{ij}

g



Response for one window:

$$f_{ij}^T g = \|f_{ij}\| \, \|g\| \cos \theta_{ij}$$

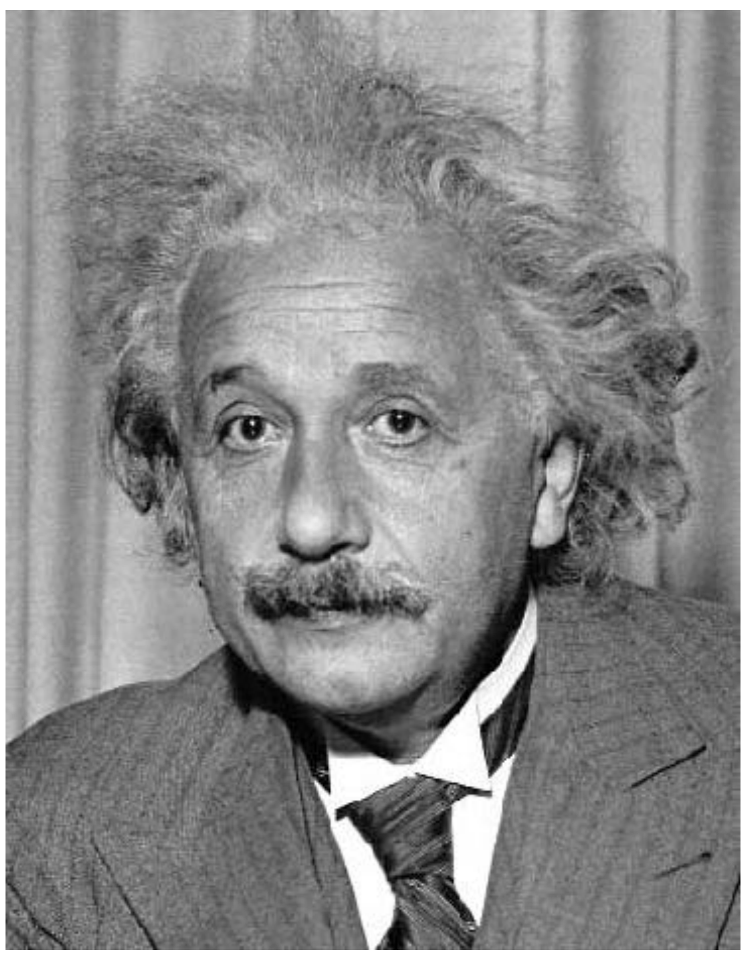


Source: James Hays, Deva Ramanan

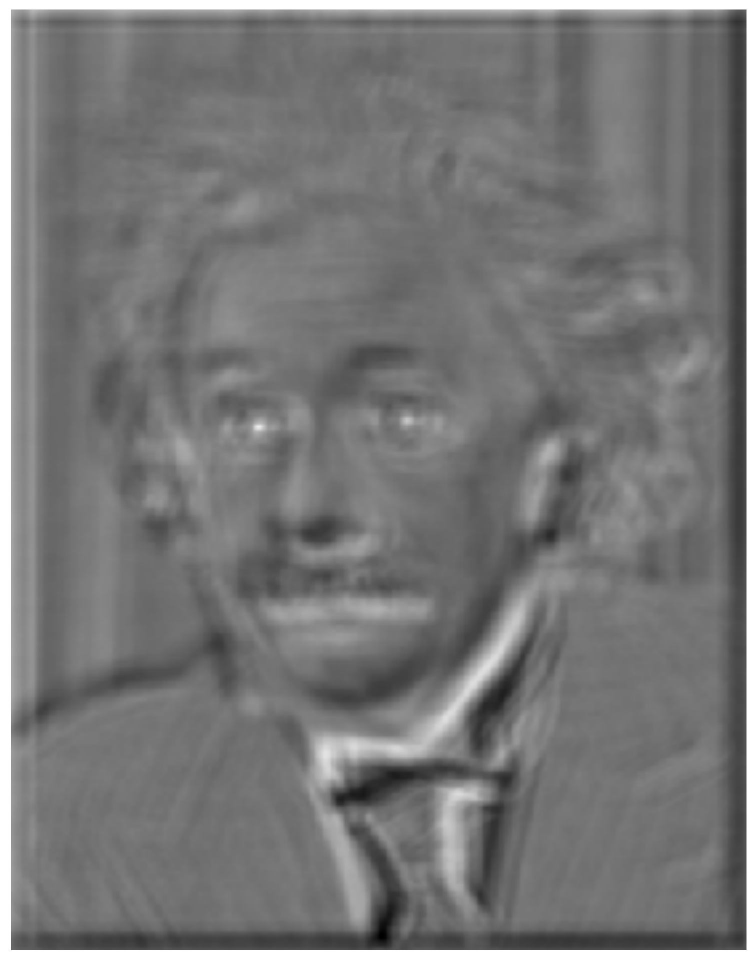
Detection by Filtering

Find the filter g 

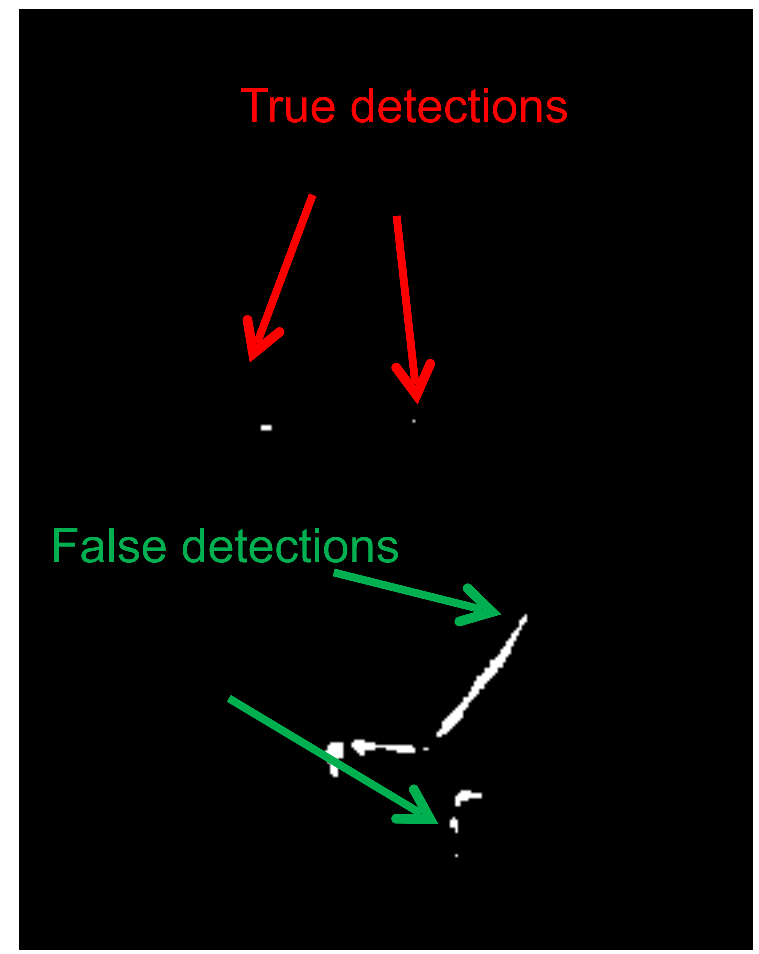
f



$f * (g - \bar{g})$



Filter Response



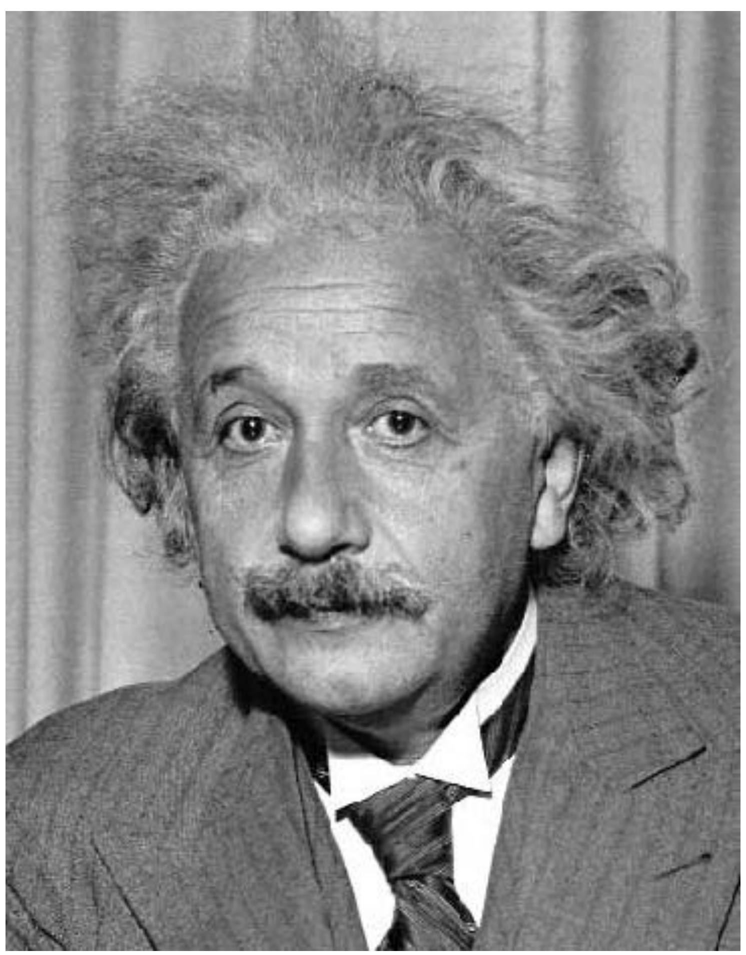
Thresholded

Source: James Hays, Deva Ramanan

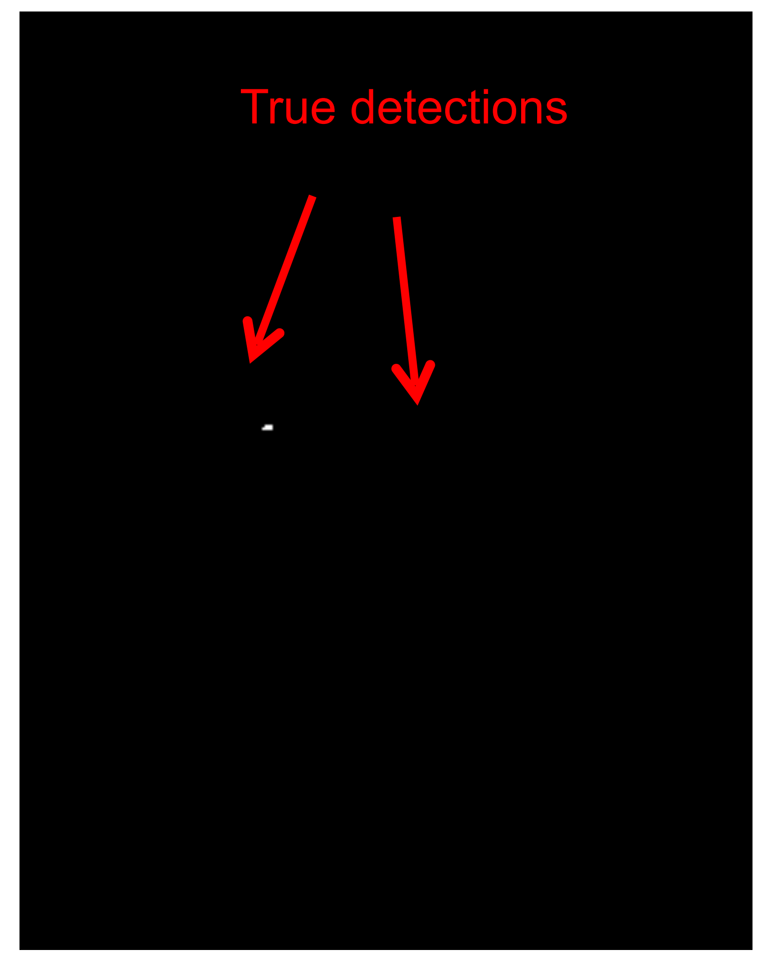
Sum of Squared Differences

$$SSD[i,j] = \|f_{ij} - h\|_2^2$$
$$= \left(f_{ij} - h\right)^T \left(f_{ij} - h\right)$$

How do you write this as a linear filter?



1-sqrt(SSD)

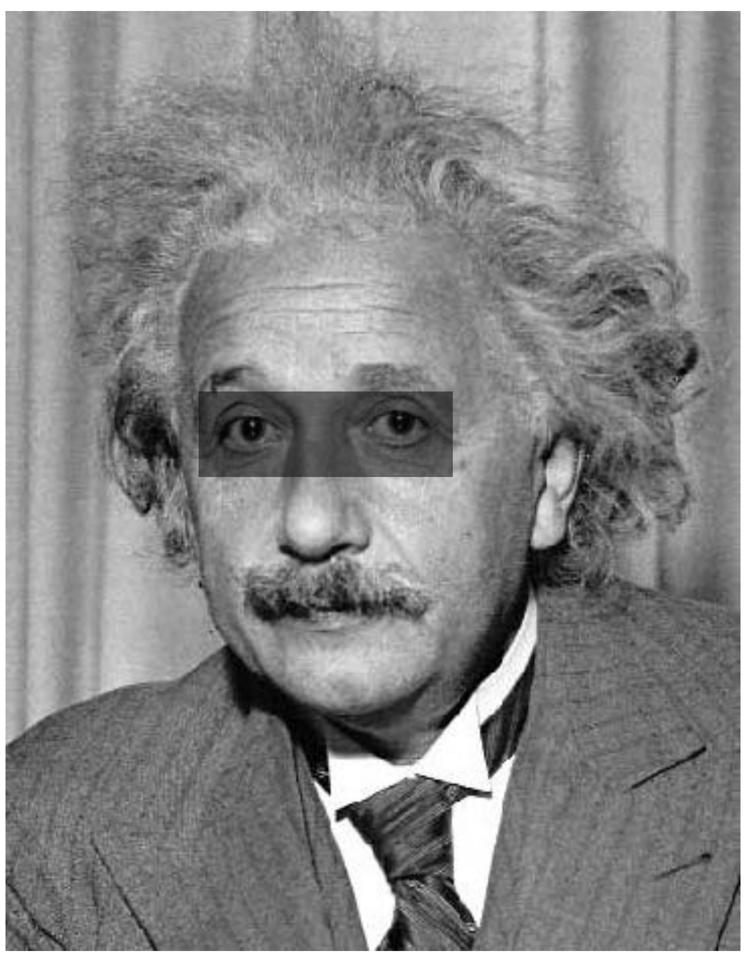


Thresholded

Source: James Hays, Deva Ramanan

Sum of Squared Differences

What does SSD do here?

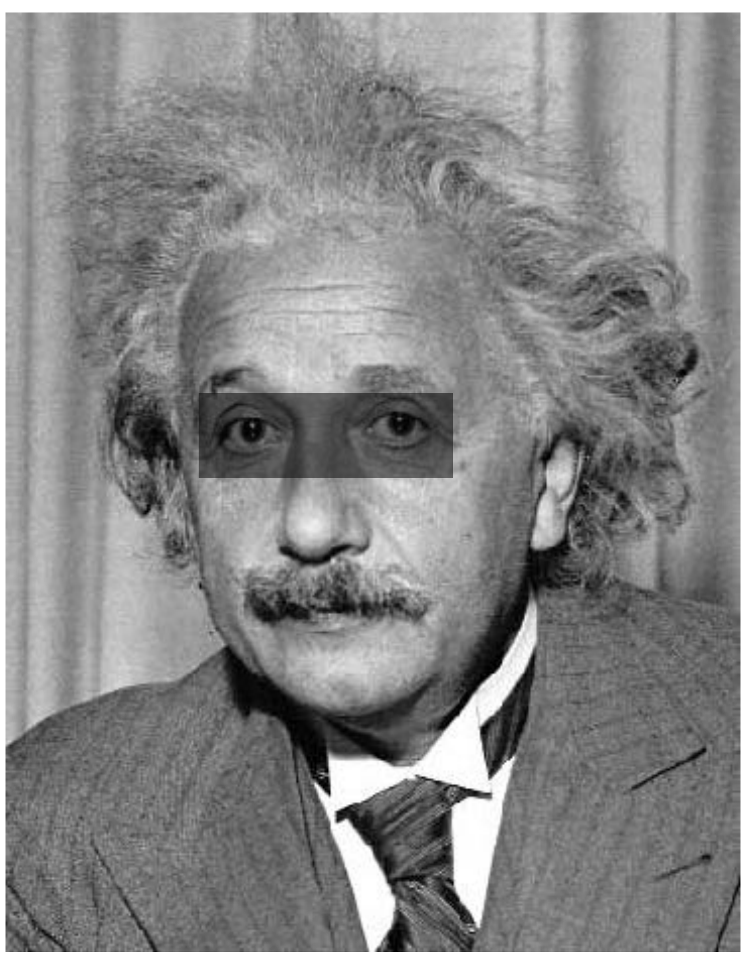


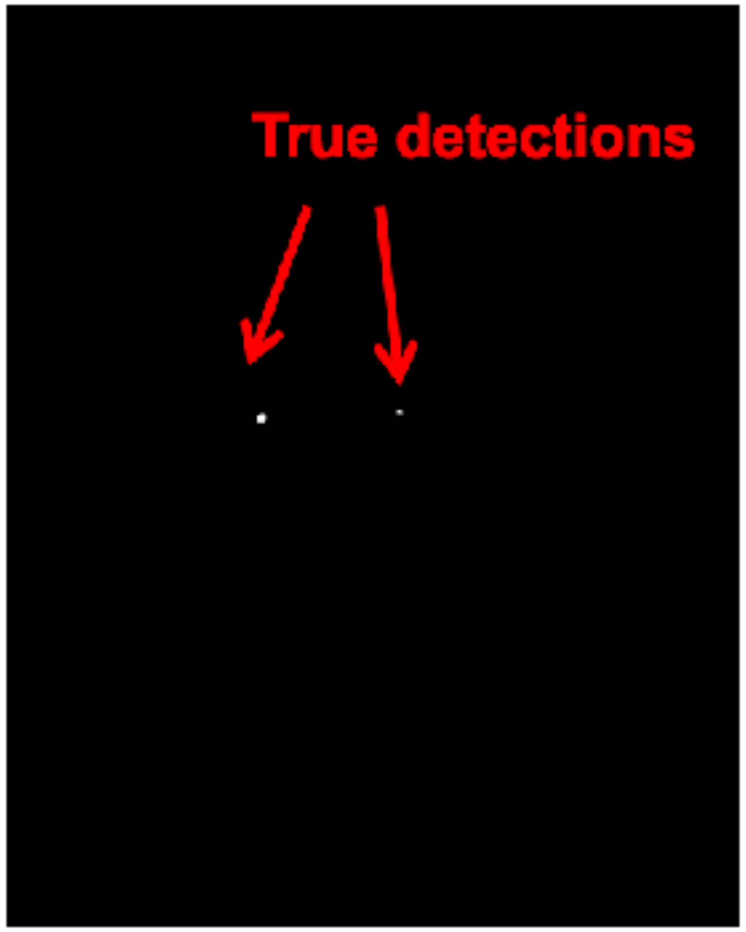
1-sqrt(SSD)

Source: Deva Ramanan

Normalized Cross Correlation

$$NCC[i, j] = \frac{f_{ij}^T h}{\|f_{ij}\| \, \|h\|} = \cos \theta_{ij}$$





Source: Deva Ramanan

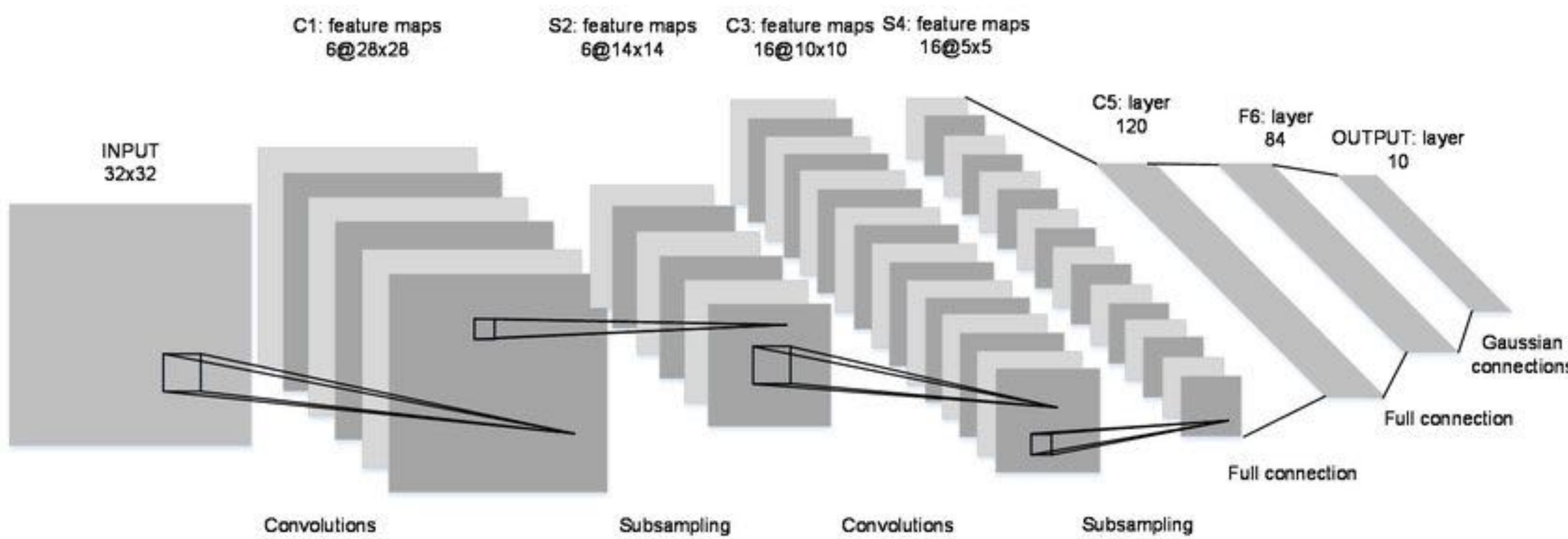
Intra-class variance






Convolutional Networks

Convolution is building blocks for modern object recognition systems

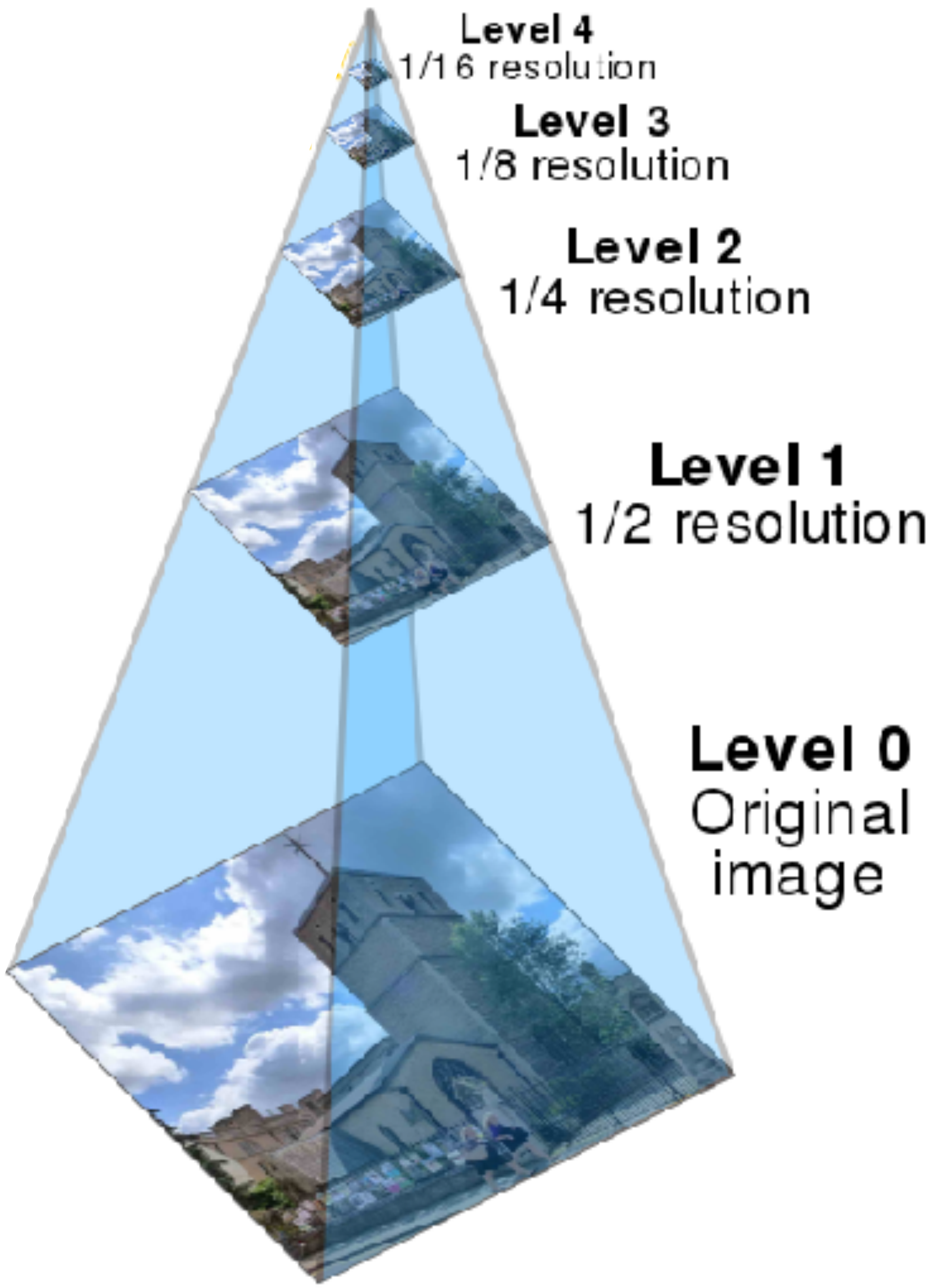


LeNet5

Pyramids


Scale

Image Pyramids



- Recursively resize image by a factor of two
- Called pyramid because it looks like a pyramid
- Invariance to scale by running operation over each level of the pyramid

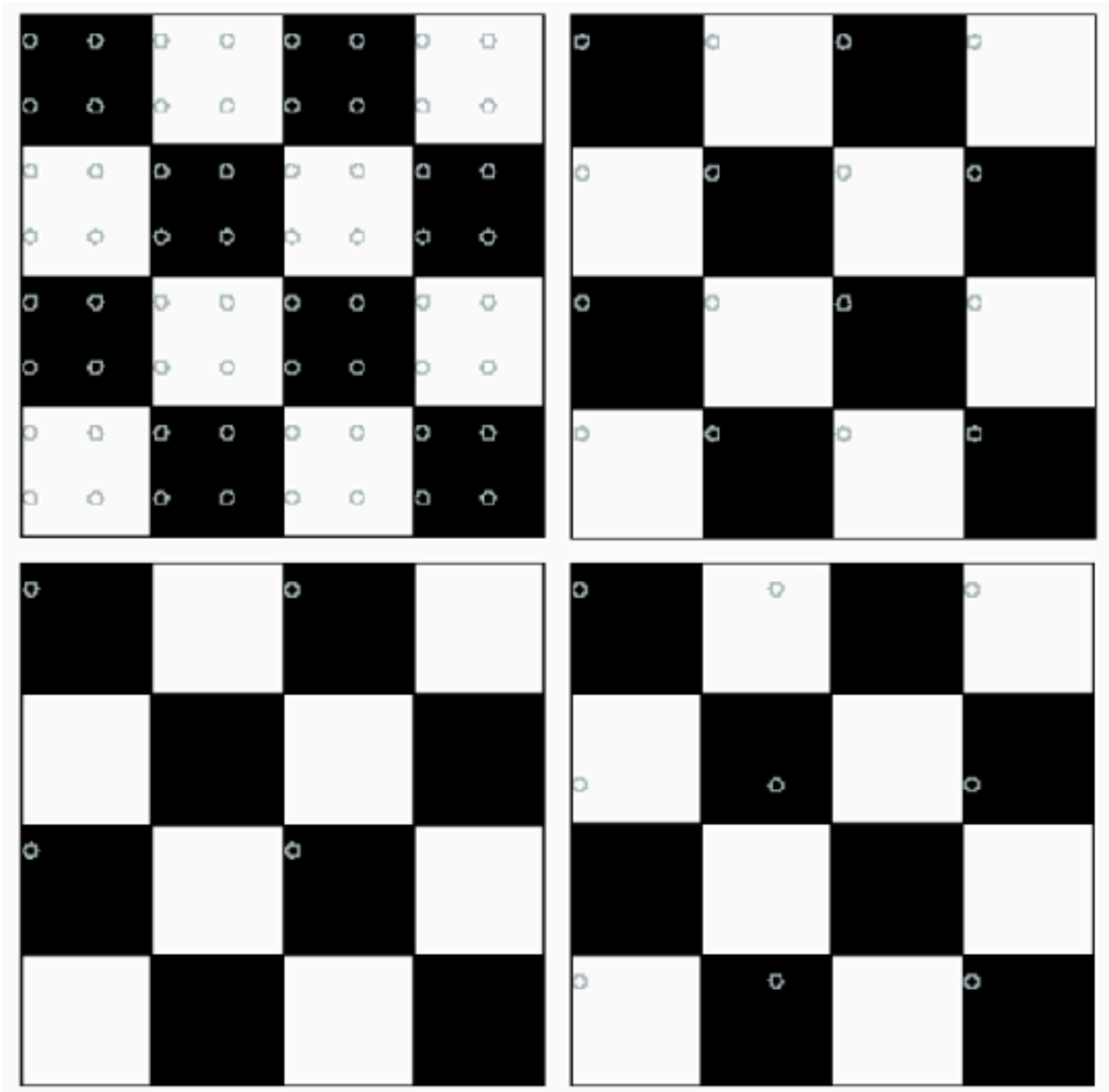
Image: Wikipedia

How to resize images?

Skip every other pixel

Why does this look bad?

Aliasing



Good sampling:
•Sample often or,
•Sample wisely

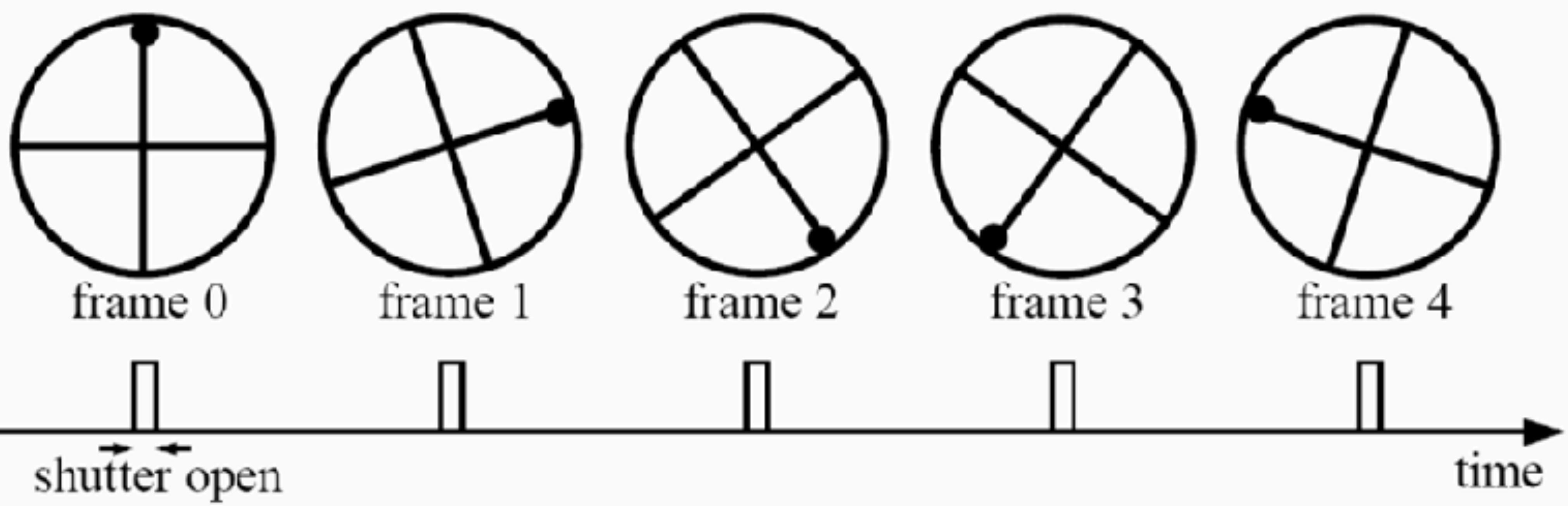
Bad sampling:
•see aliasing in action!

Source: Efros

Aliasing

Imagine a spoked wheel moving to the right (rotating clockwise). Mark wheel with dot so we can see what's happening.

If camera shutter is only open for a fraction of a frame time (frame time = 1/30 sec. for video, 1/24 sec. for film):



Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)

Source: Efros

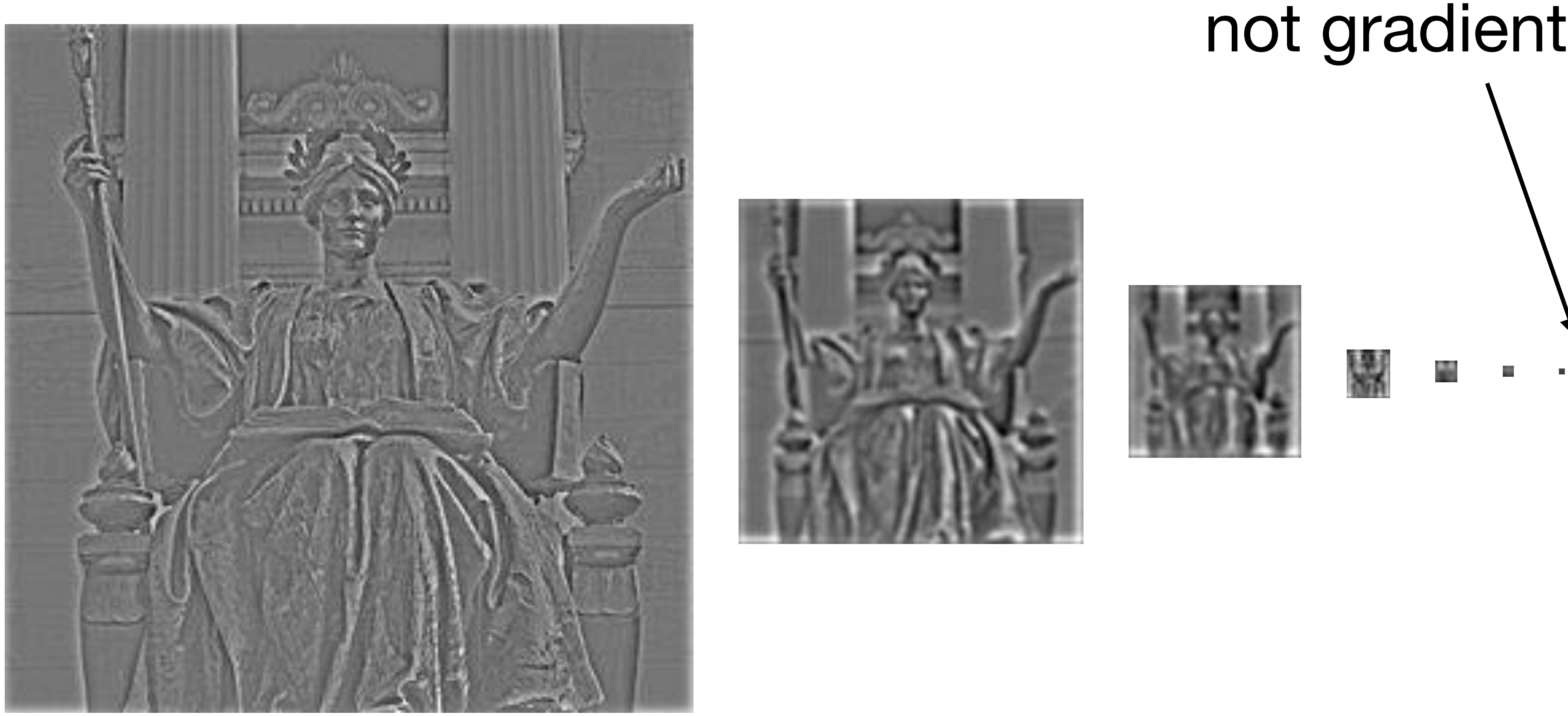
Gaussian Pyramids

1. Convolve with Gaussian filter
2. Subsample every other pixel
3. Repeat

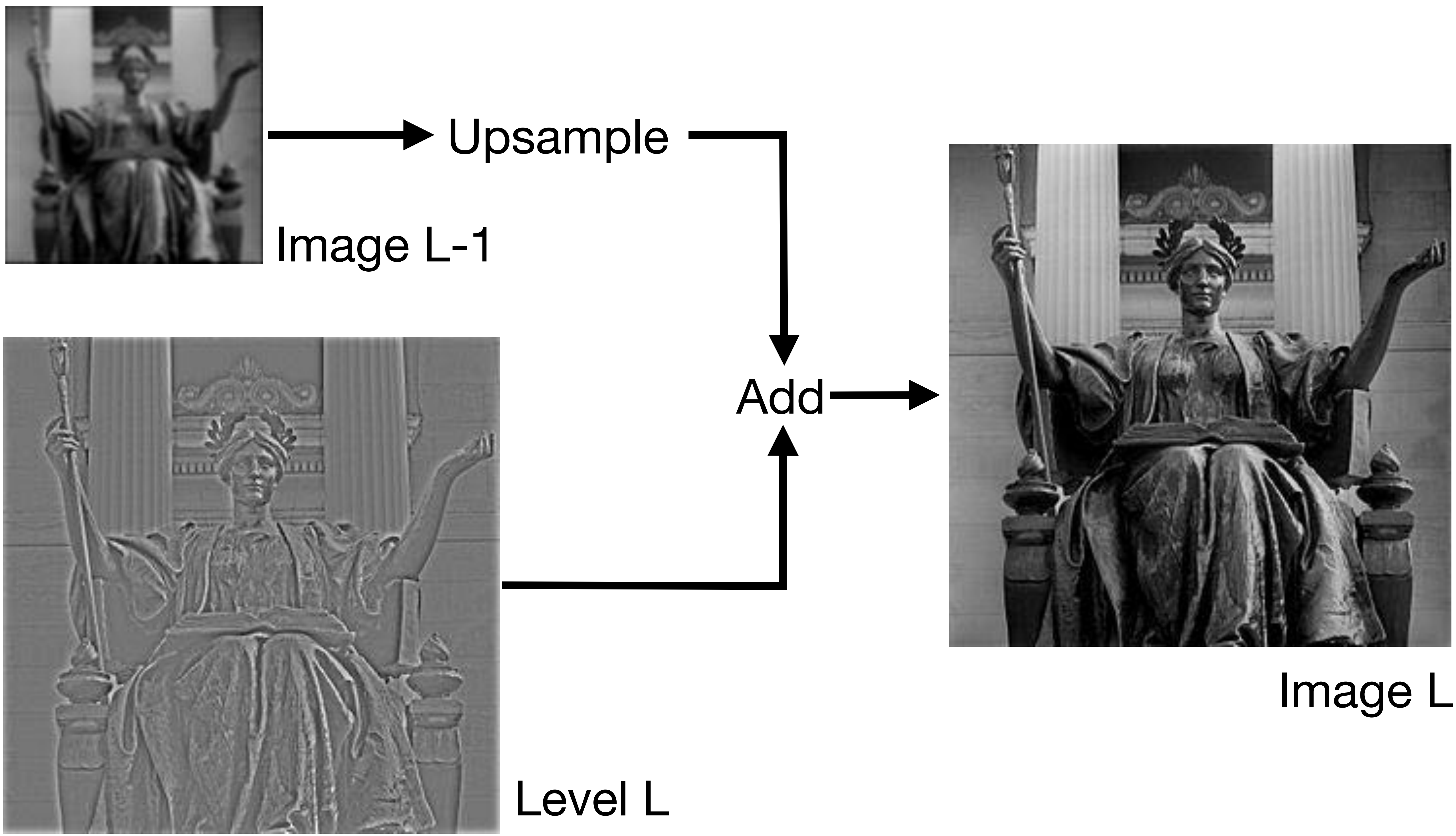
Laplacian Pyramids

1. Convolve with Laplacian filter
2. Subsample every other pixel
3. Repeat

Store downsampled image, not gradients



Recovering Image

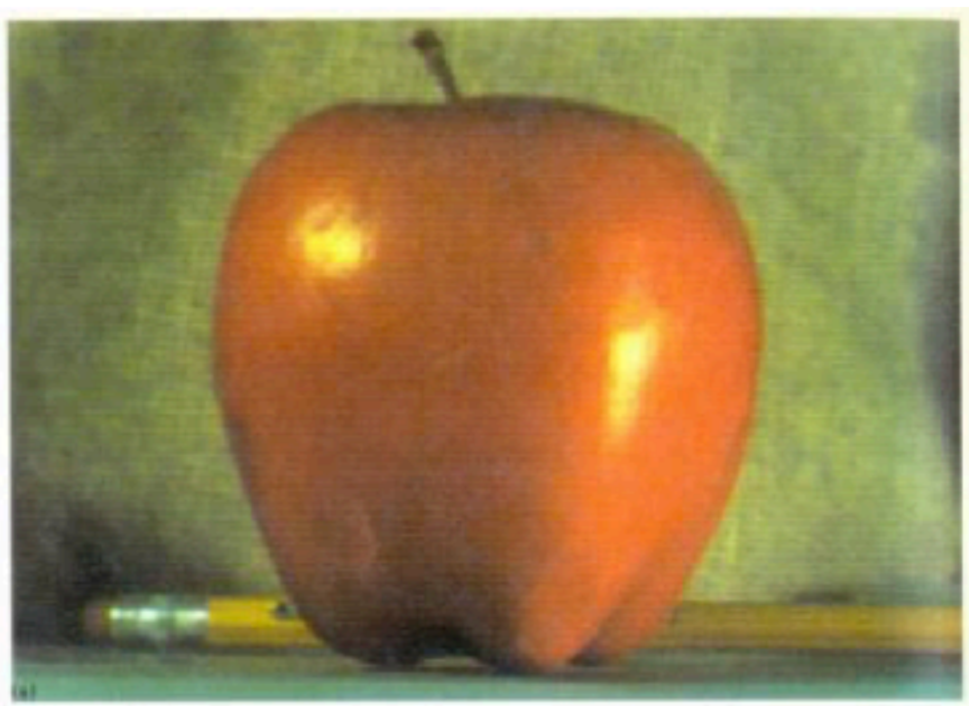


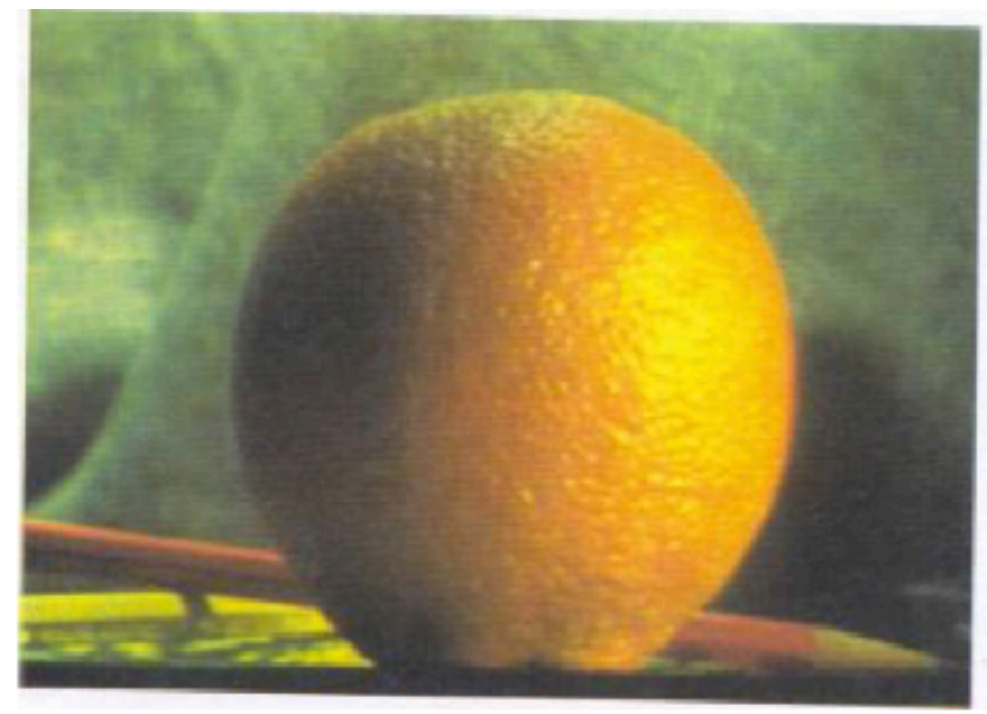
Laplacian Pyramids

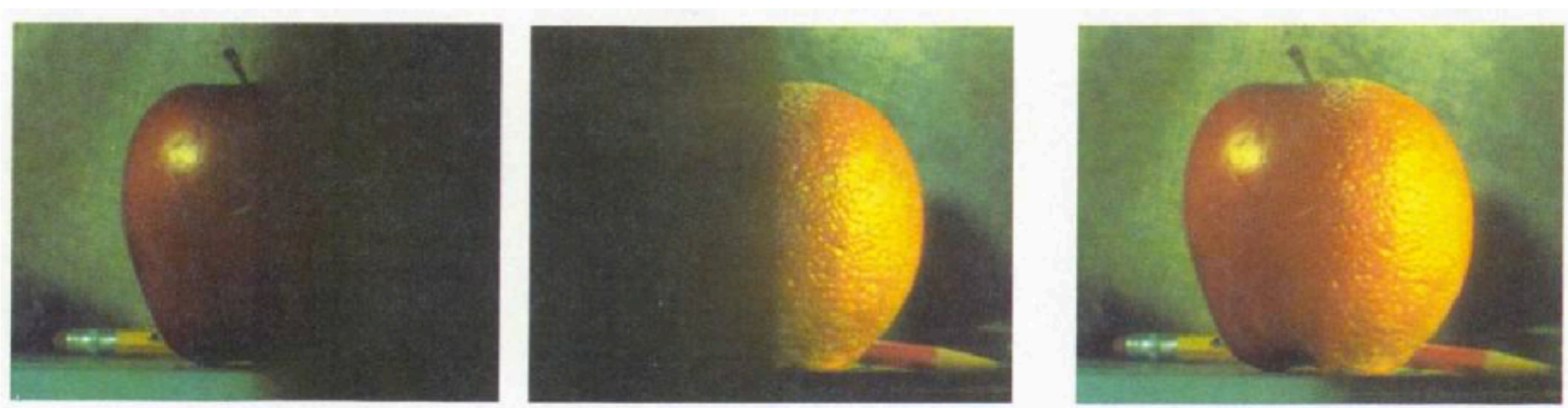
Applications:

- Compression
- Incremental transmission

Image Blending







(d) (h) (l)

Image Blending

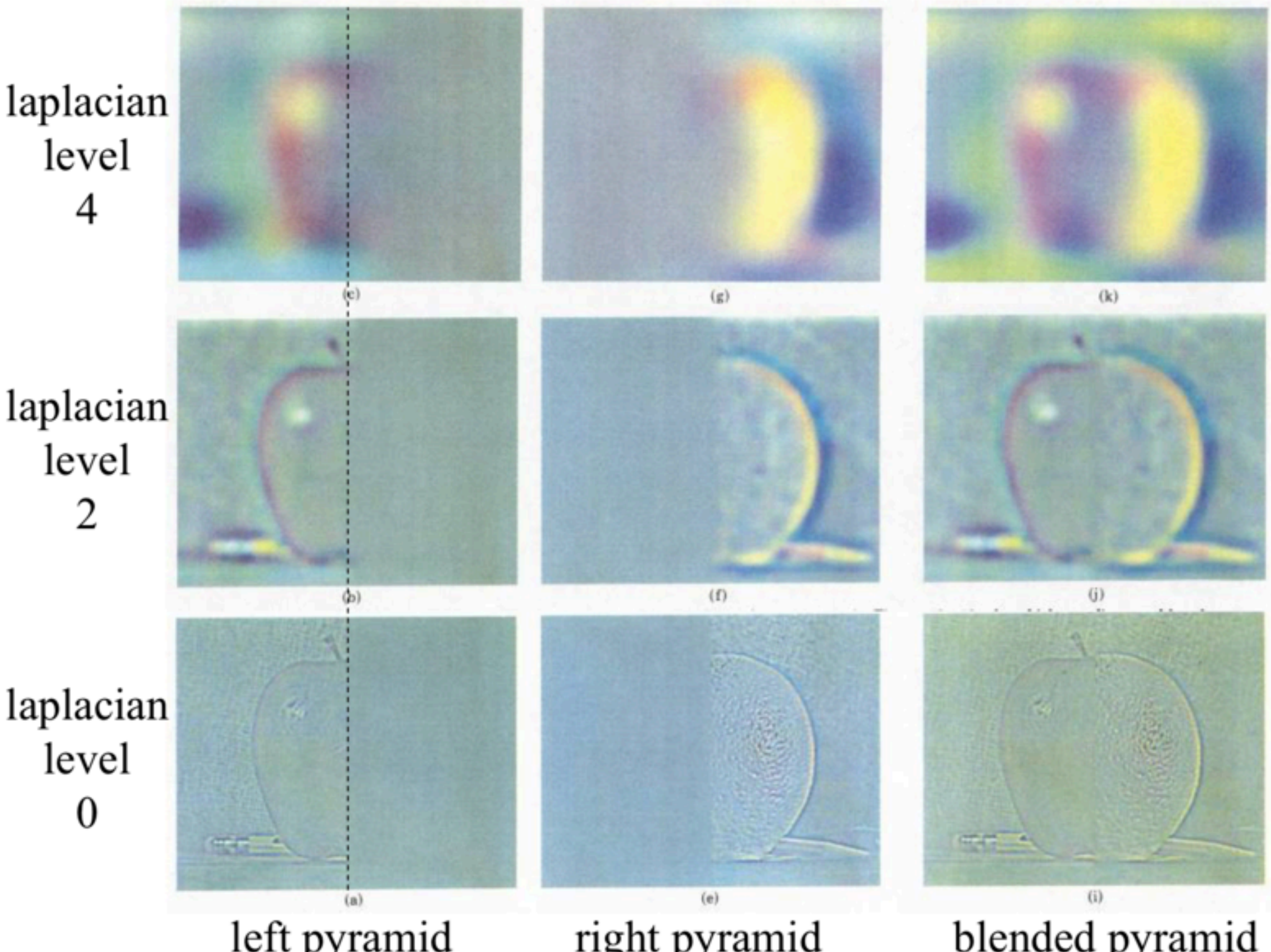


Image Blending

Image A

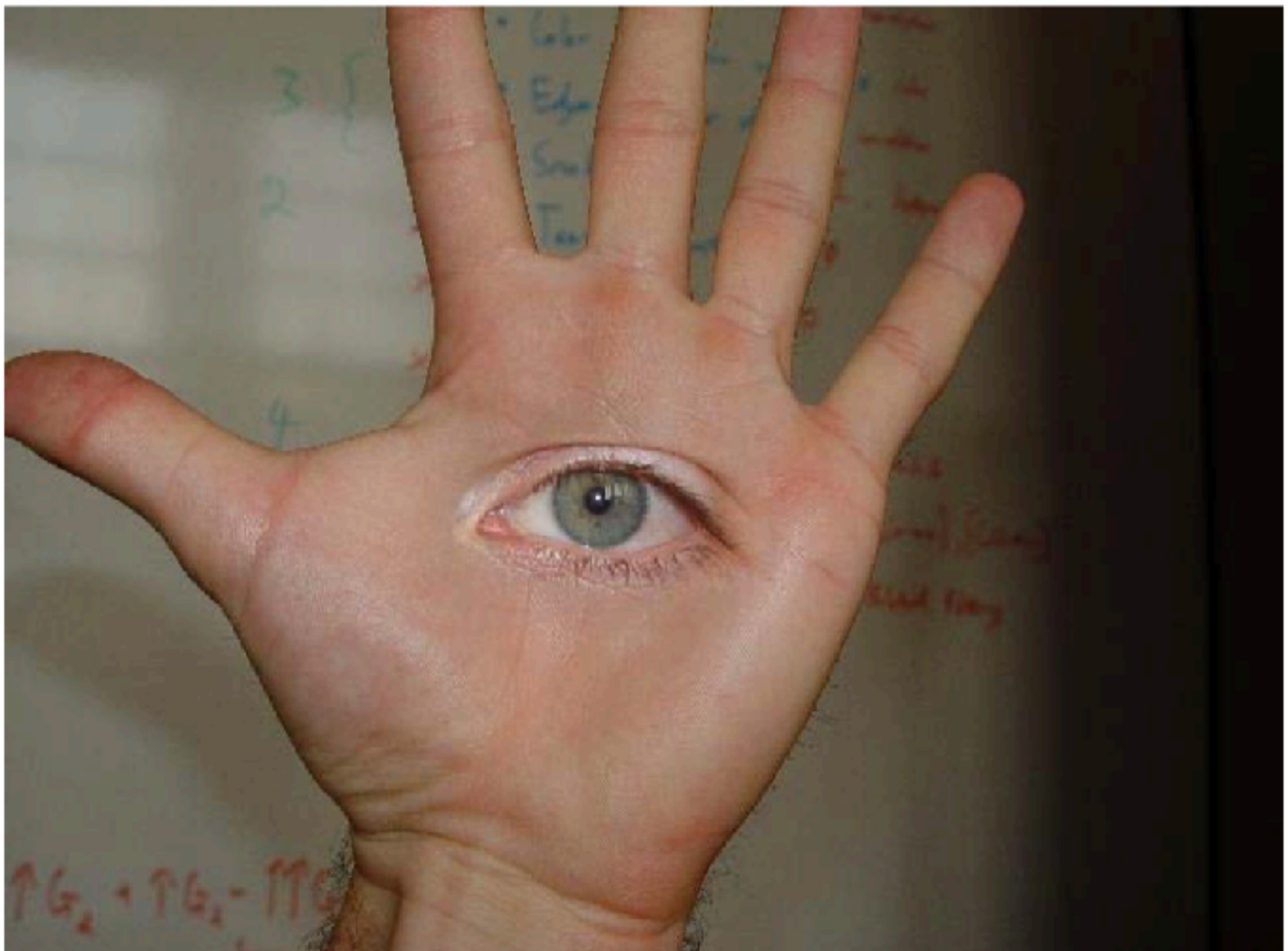
Image B

Region R

General Approach:

1. Build Laplacian pyramids *LA* and *LB* from images *A* and *B*
2. Build a Gaussian pyramid *GR* from selected region *R*
3. Form a combined pyramid *LS* from *LA* and *LB* using nodes of *GR* as weights:
   - $LS(i,j) = GR(I,j,)*LA(I,j) + (1\text{-}GR(I,j))*LB(I,j)$
4. Collapse the *LS* pyramid to get the final blended image

Image Blending



© prof. dmartin

Next Class: Repetition