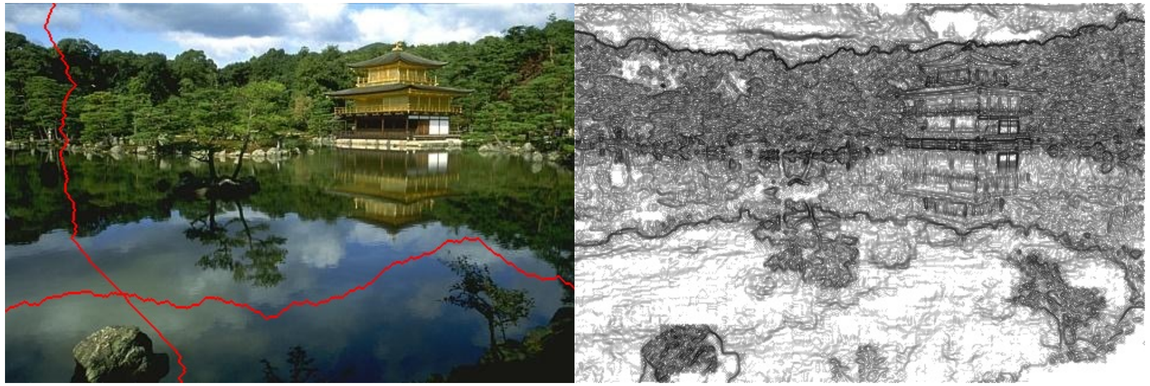
CS 4476 Computer Vision:

4. Image Gradients & Edge Detection

Humphrey Shi

8/29/2024



The follow slides are mostly based on the lecture slides of Judy Hoffman, James Hays, and Svetlana Lazebnik, with credits to Devi Parikh, Kristen Grauman, Fei-Fei Li etc.

Outline

- Logistics & Course Recap
- Image Filtering (continued)
- Image Gradients & Edge Detection

Logistics & Course Recap

Logistics

- Assignment 1 due this weekend.

Assignment 1: Image Filtering

AUG 26, 2024 8:00 AM — SEP 8, 2024 11:59 PM

Late Due Date: SEP 13, 2024 11:59 PM

8

New Multimodal LLMs from GT / NVIDIA

EAGLE: Exploring The Design Space for Multimodal LLMs with Mixture of Encoders

Min Shi[2*], Fuxiao Liu[3*], Shihao Wang[4], Shijia Liao[1], Subhashree Radhakrishnan[1],
De-An Huang[1], Hongxu Yin[1], Karan Sapra[1], Yaser Yacoob[3], Humphrey Shi[2],
Bryan Catanzaro[1], Andrew Tao[1], Jan Kautz[1], Zhiding Yu[1†], Guilin Liu[1†]

[1]NVIDIA [2]Georgia Tech [3]UMD [4]HKPU
https://github.com/NVlabs/Eagle

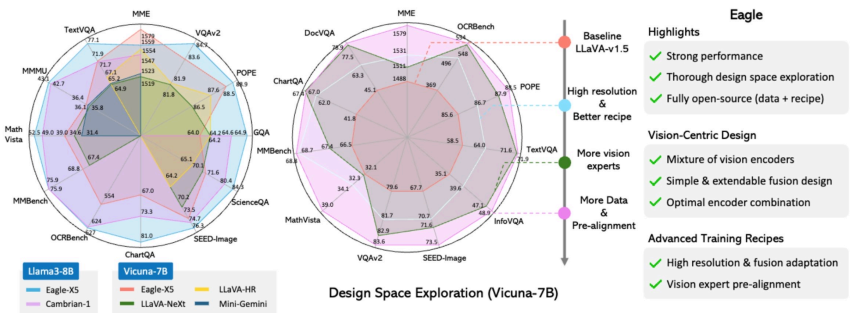
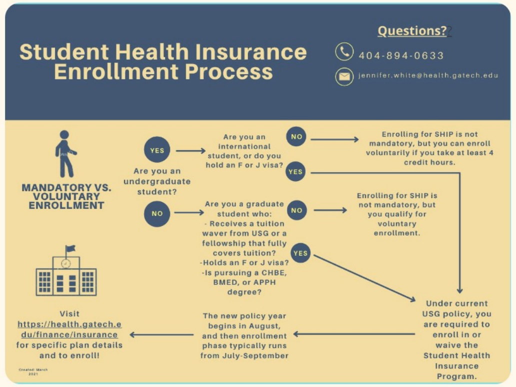


Figure 1: **Overview of *Eagle*.** *Eagle* is a family of multimodal large language models (MLLMs) with a mixture of vision encoders. Left: comparisons between *Eagle* and existing competitive MLLMs with *Vicuna-7B* [15] and *Llama3-8B* [3], with *Eagle* achieving favorable results on all 12 benchmarks. Middle: an evolutionary road map of the design space and advanced training recipes leading to consistent and significant improvements. Right: highlights and core features of *Eagle*.



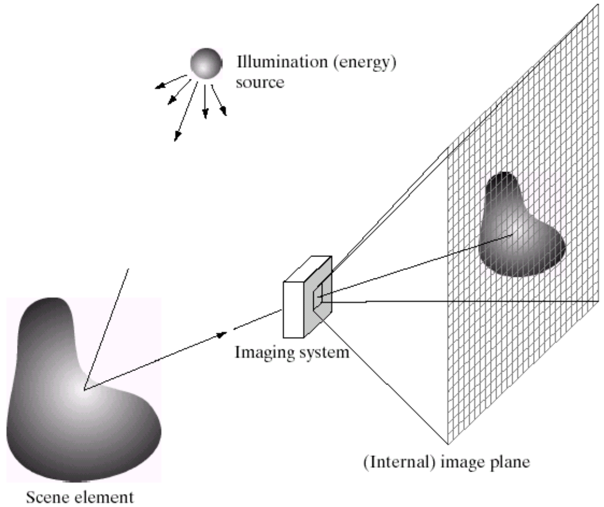
Under which circumstances do I need to be enrolled in mandatory health insurance if I am an international student?

If you are an international student and hold an F or J visa

Link

ML Student Conference @ CODA, Tomorrow

Recap: Image Filtering



Digital color image formation

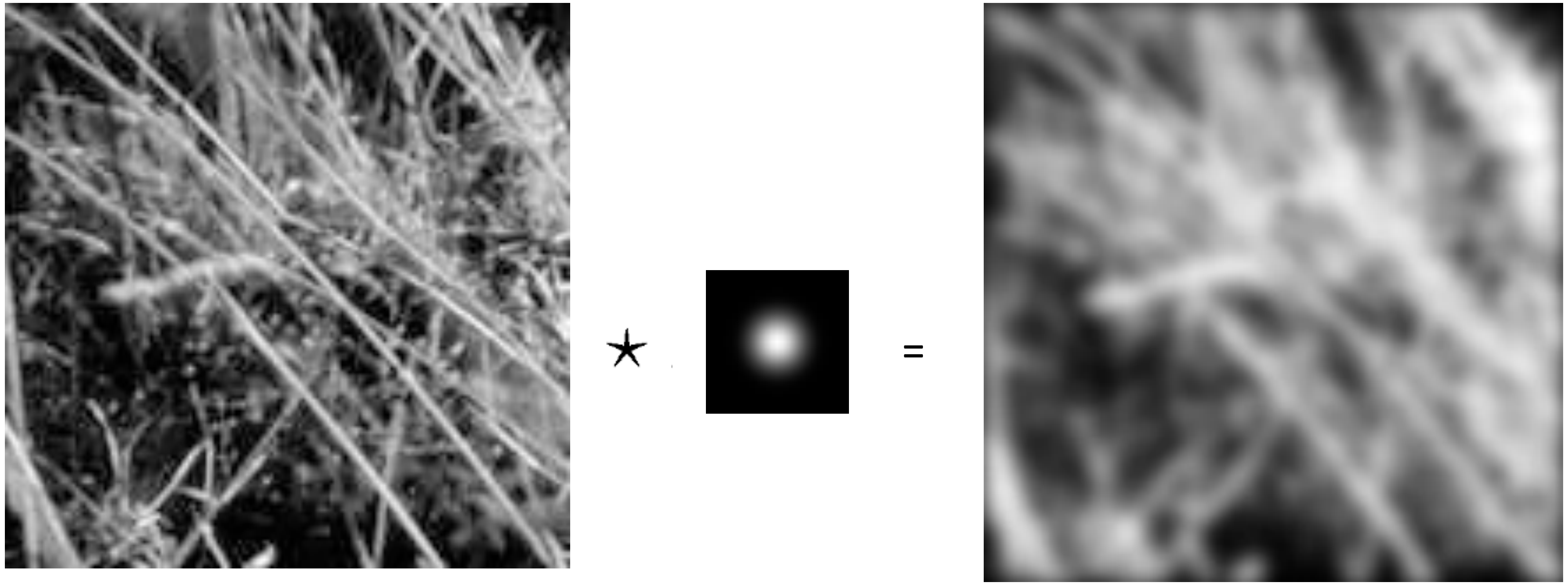
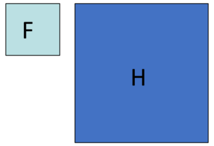


Image filtering to enhance/smooth an image

Cross-correlation

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

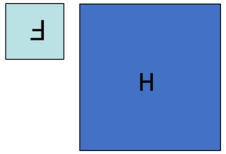
$$G = H \otimes F$$



Convolution

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

$$G = H \star F$$



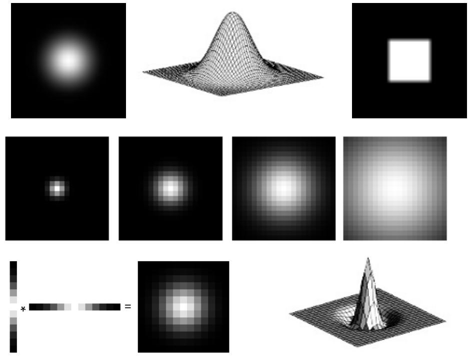
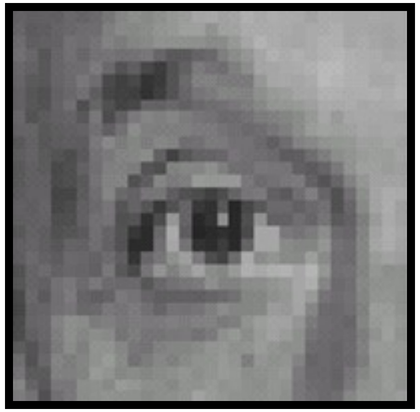


Image Filtering (continued)

Sharpening, Median Filters, and Hybrid Images

Practice with linear filters



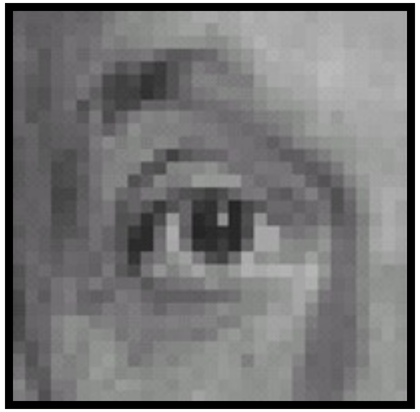
Original

0	0	0
0	1	0
0	0	0

?

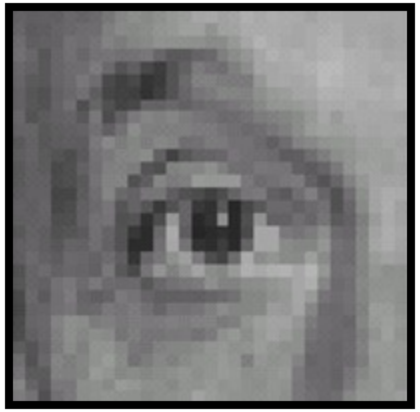
Slide credit: David Lowe

Practice with linear filters



Original

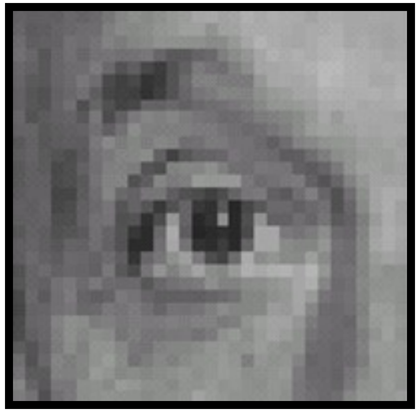
0	0	0
0	1	0
0	0	0



Filtered
(no change)

Slide credit: David Lowe

Practice with linear filters



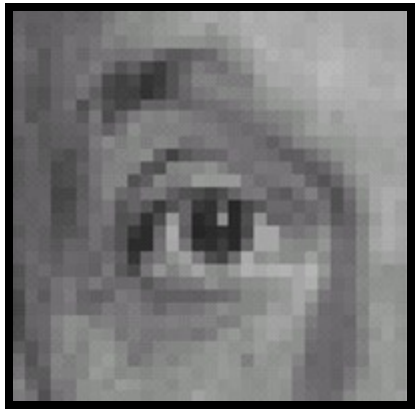
Original

0	0	0
0	0	1
0	0	0

?

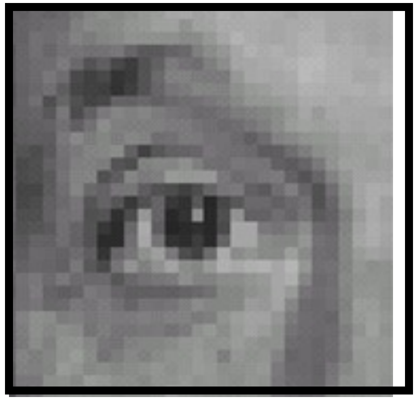
Slide credit: David Lowe

Practice with linear filters



Original

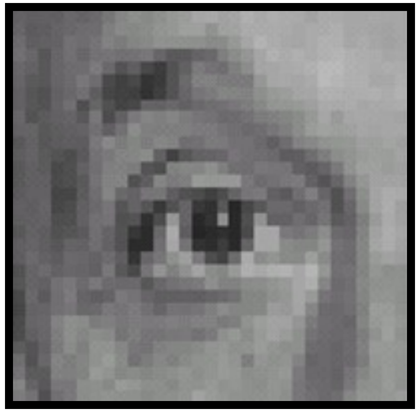
0	0	0
0	0	1
0	0	0



Shifted **LEFT**
by 1 pixel with
correlation

Slide credit: David Lowe

Practice with linear filters



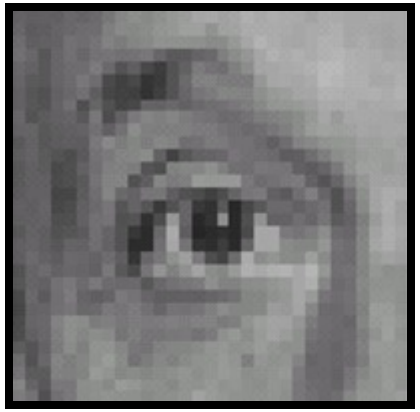
Original

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

?

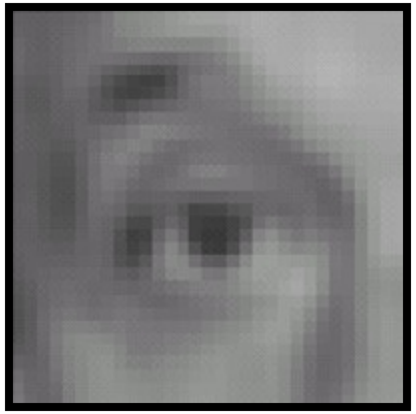
Slide credit: David Lowe

Practice with linear filters



Original

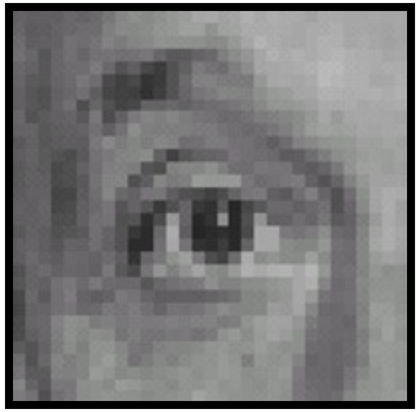
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$



Blur (with a box filter)

Slide credit: David Lowe

Practice with linear filters

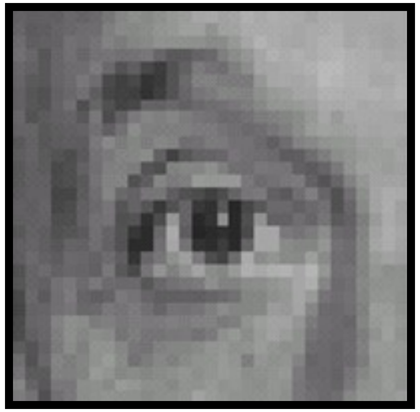


Original

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

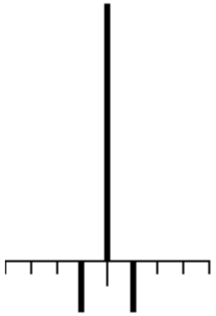
?

Practice with linear filters



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

Slide credit: David Lowe

Filtering – Sharpening

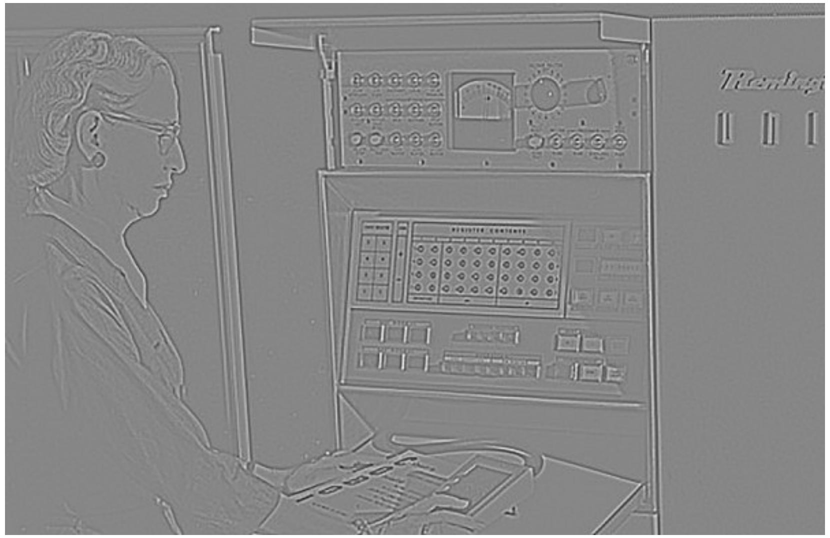
Image

-

Smoothed

Details

=



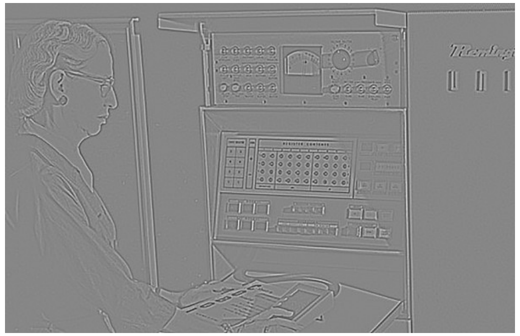
Slide credit: David Fouhey

Filtering – Sharpening

Image

+α

Details



“Sharpened” α=1

=

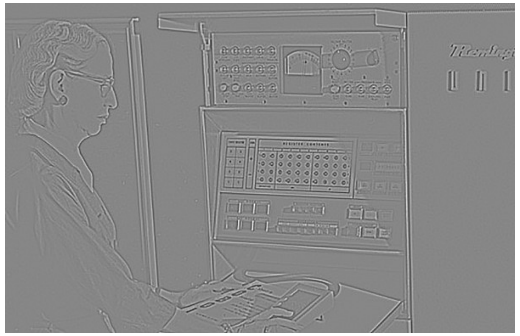
Slide credit: David Fouhey

Filtering – Sharpening

Image

+α

Details



"Sharpened" α=0

=

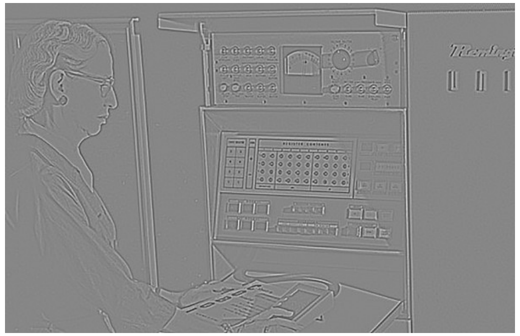
Slide credit: David Fouhey

Filtering – Sharpening

Image

+α

Details



“Sharpened” α=2

=

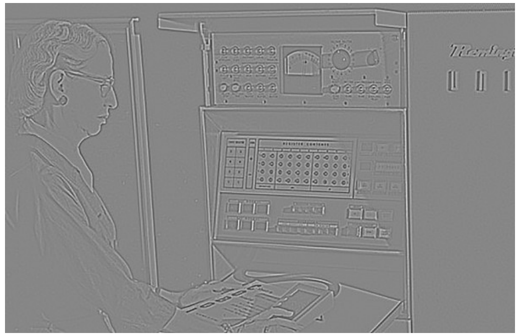
Slide credit: David Fouhey

Filtering – Sharpening

Image

+α

Details



“Sharpened” α=0

=

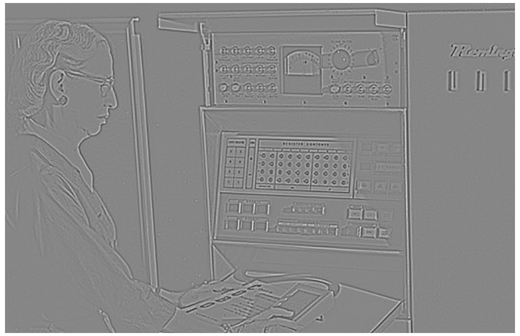
Slide credit: David Fouhey

Filtering – Extreme Sharpening

Image

+α

Details



“Sharpened” α=10

=

Slide credit: David Fouhey

Effect of smoothing filters

5x5

Additive Gaussian noise

Salt and pepper noise

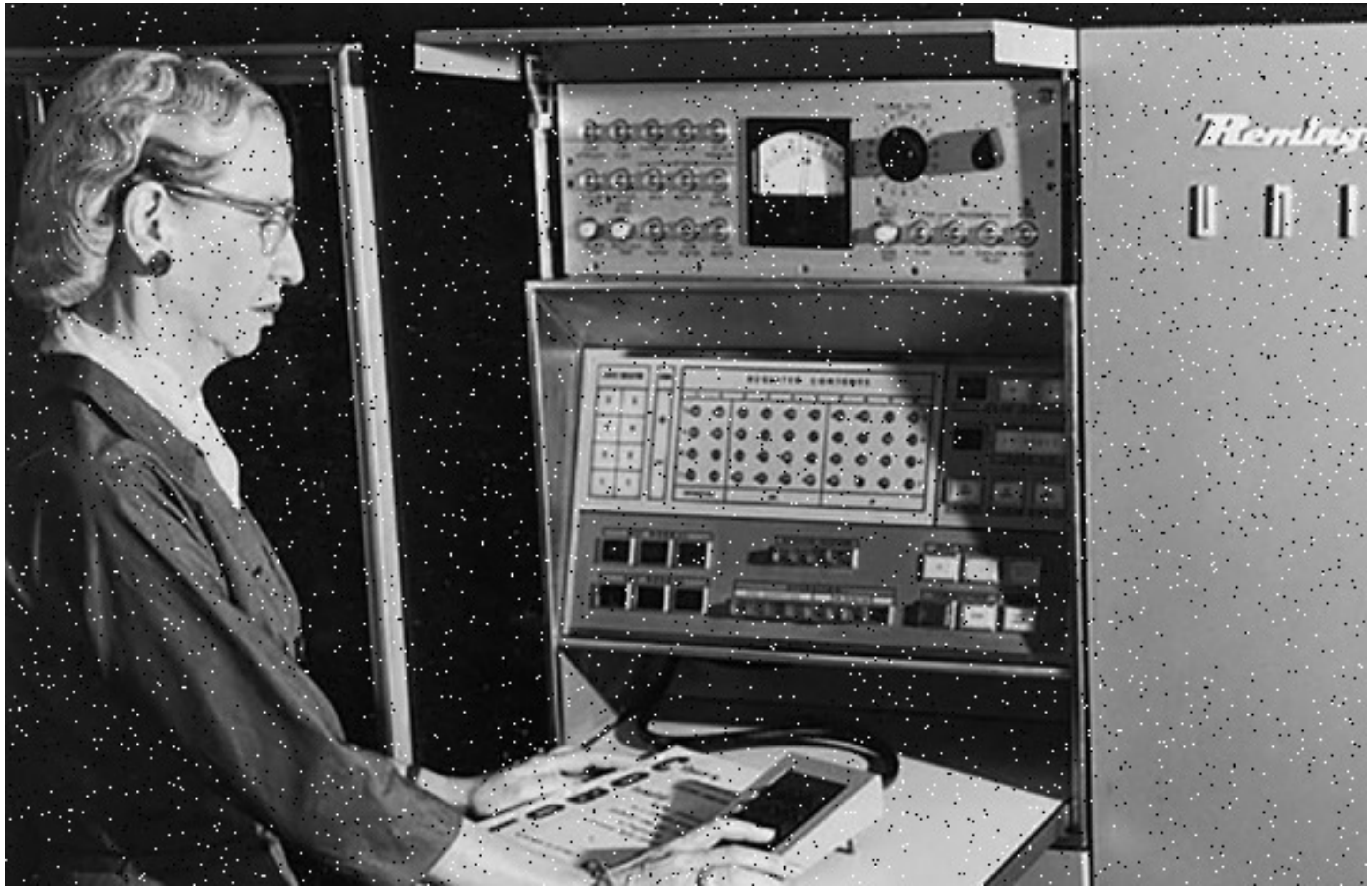
Slide credit: Kristen Grauman

Why Gaussian?

Gaussian filtering removes parts of the signal above a certain frequency. Often noise is high frequency and signal is low frequency.

Slide credit: David Fouhey

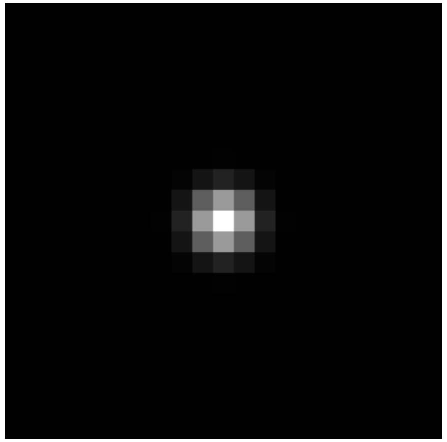
Where Gaussian Fails

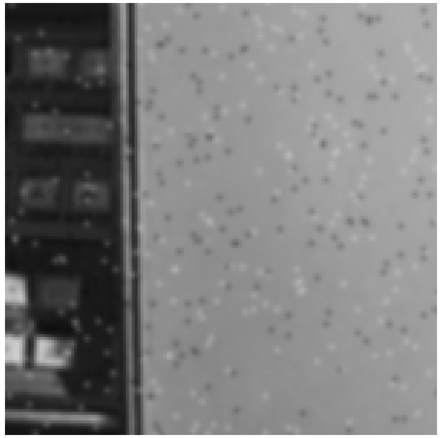


Slide credit: David Fouhey

Applying Gaussian Filters

σ = 1





Slide credit: David Fouhey

Why Does This Fail?

Means can be arbitrarily distorted by outliers

Signal	10	12	9	8	1000	11	10	12
Filter	0.1	0.8	0.1					
Output	11.5	9.2	107.3	801.9	109.8	10.3		

What else is an "average" other than a mean?

Slide credit: David Fouhey

Non-linear Filters (2D)

40	81	13	22
125	830	76	80
144	92	108	95
132	102	106	87

[040, 081, 013, 125, 830, 076, 144, 092, 108]

Sort

[013, 040, 076, 081, 092, 108, 125, 144, 830]

92

[830, 076, 080, 092, 108, 095, 102, 106, 087]

Sort

[076, 080, 087, 092, 095, 102, 106, 108, 830]

95

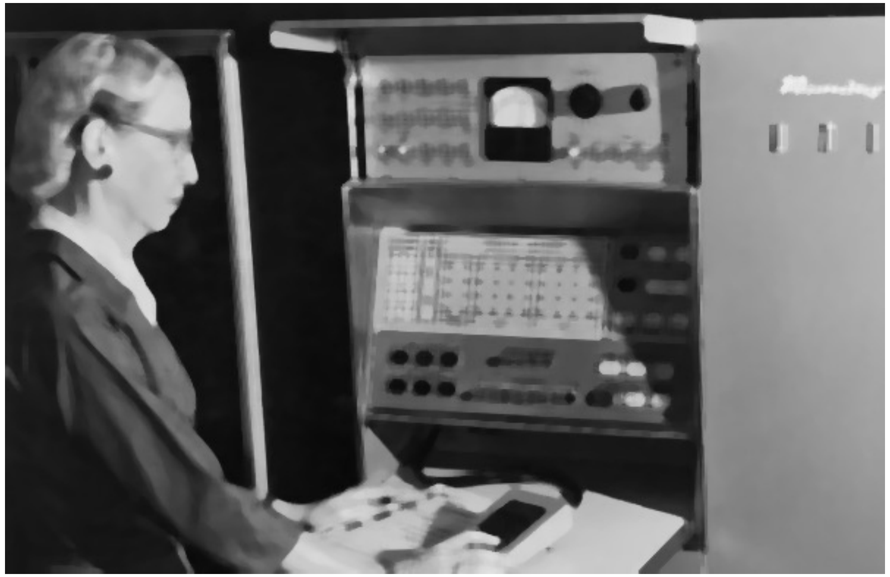
Slide credit: David Fouhey

Applying Median Filter

Median
Filter
(size=3)

Applying Median Filter

Median
Filter
(size = 7)



Slide credit: David Fouhey

Is Median Filtering Linear?

$$\begin{bmatrix}1 & 1 & 1\\1 & 1 & 2\\2 & 2 & 2\end{bmatrix} + \begin{bmatrix}0 & 0 & 0\\0 & 1 & 0\\0 & 0 & 0\end{bmatrix} = \begin{bmatrix}1 & 1 & 1\\1 & 2 & 2\\2 & 2 & 2\end{bmatrix}$$

Median Filter

$$1 \quad + \quad 0 \quad = \quad 2$$

Example from (I believe): Kristen Grauman

IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. ASSP-27, NO. 1, FEBRUARY 1979 13

A Fast Two-Dimensional Median Filtering Algorithm

THOMAS S. HUANG, SENIOR MEMBER, IEEE, GEORGE J. YANG, STUDENT MEMBER, IEEE, AND
GREGORY Y. TANG, STUDENT MEMBER, IEEE

Abstract—We present a fast algorithm for two-dimensional median filtering. It is based on storing and updating the gray level histogram of the picture elements in the window. The algorithm is much faster than conventional sorting methods. For a window size of m × n, the computer time required is 0(n).

II. A Fast Two-Dimensional Median Filtering Algorithm

In doing median filtering, we are computing *running* medians. From one output picture element to the next, the m × n window moves only one column. To get the numbers in the new

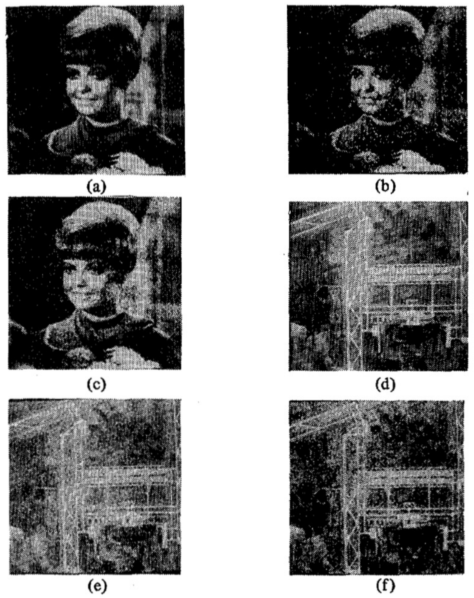


Fig. 1. Images before filtering. (a) Girl, original. (b) Girl, with BSC noise. (c) Girl, with Gaussian noise. (d) Airport, original. (e) Airport, with BSC noise. (f) Airport, with Gaussian noise.

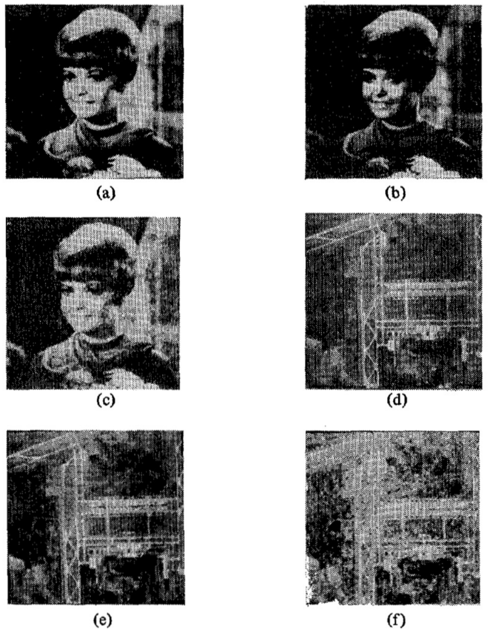


Fig. 2. The six images of Fig. 1 after 3 × 3 median filtering.

TABLE II
COMPUTER TIME IN SECONDS (GIRL—IMAGES 1-3)

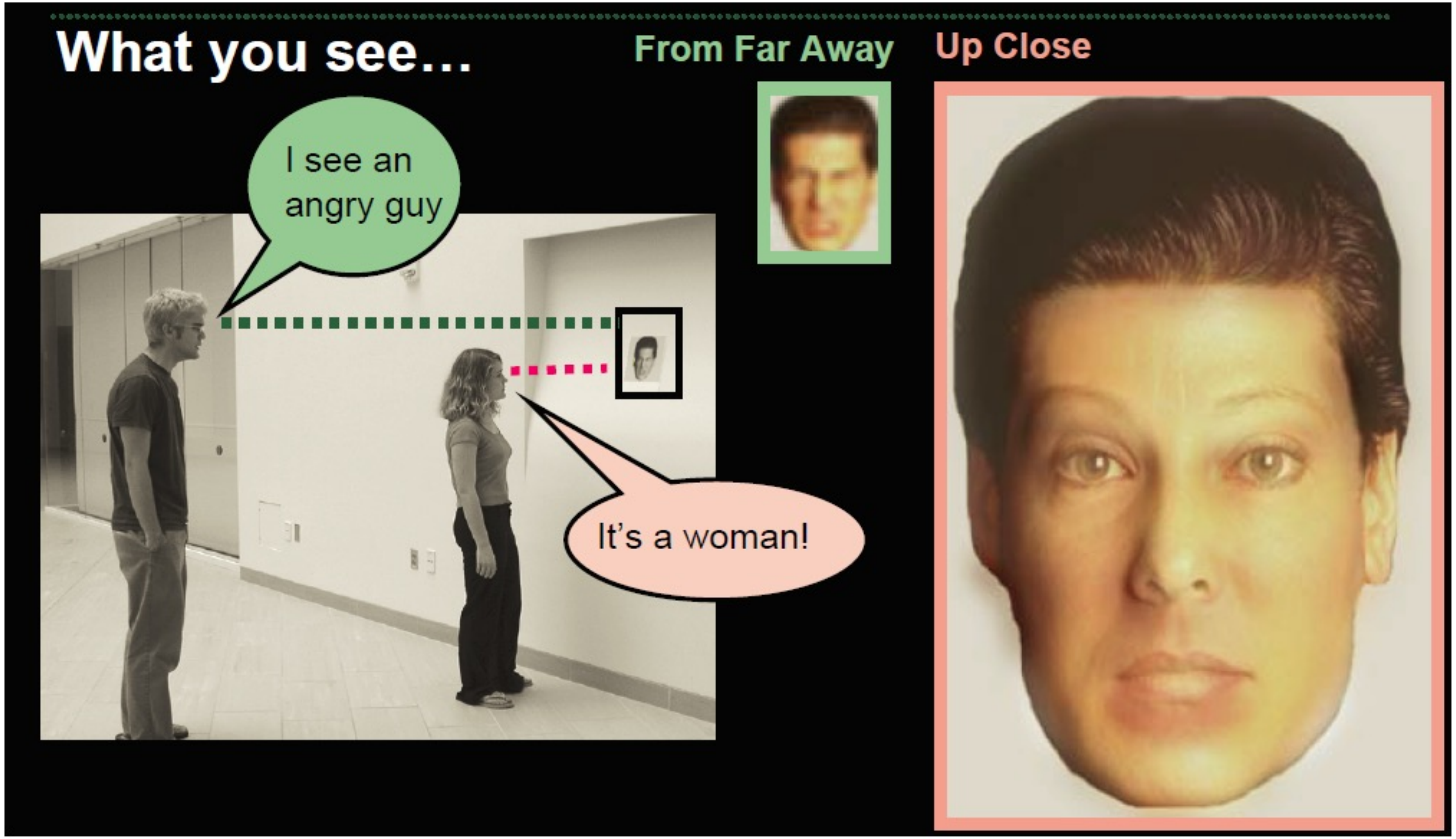
Window Size	Histogram Method			Quicksort		
	Girl	Girl BSC	Girl Gaussian	Girl	Girl BSC	Girl Gaussian
3 x 3	41.9	45.4	53.1	180.4	143.7	155.8
5 x 5	52.9	52.9	56.4	443.32	347.5	372.7
7 x 7	64.4	66.4	65.6	833.3	651.0	690.5
9 x 9	74.9	76.2	75.9	1087.4	1053.3	
11 x 11	86.2	86.9	86.5			
13 x 13	97.3	98.4	99.2			

Filtering application: Hybrid Images

Slide credit: Kristen Grauman

Aude Oliva & Antonio Torralba & Philippe G Schyns, SIGGRAPH 2006

Filtering application: Hybrid Images



Slide credit: Kristen Grauman

Aude Oliva & Antonio Torralba & Philippe G Schyns, SIGGRAPH 2006

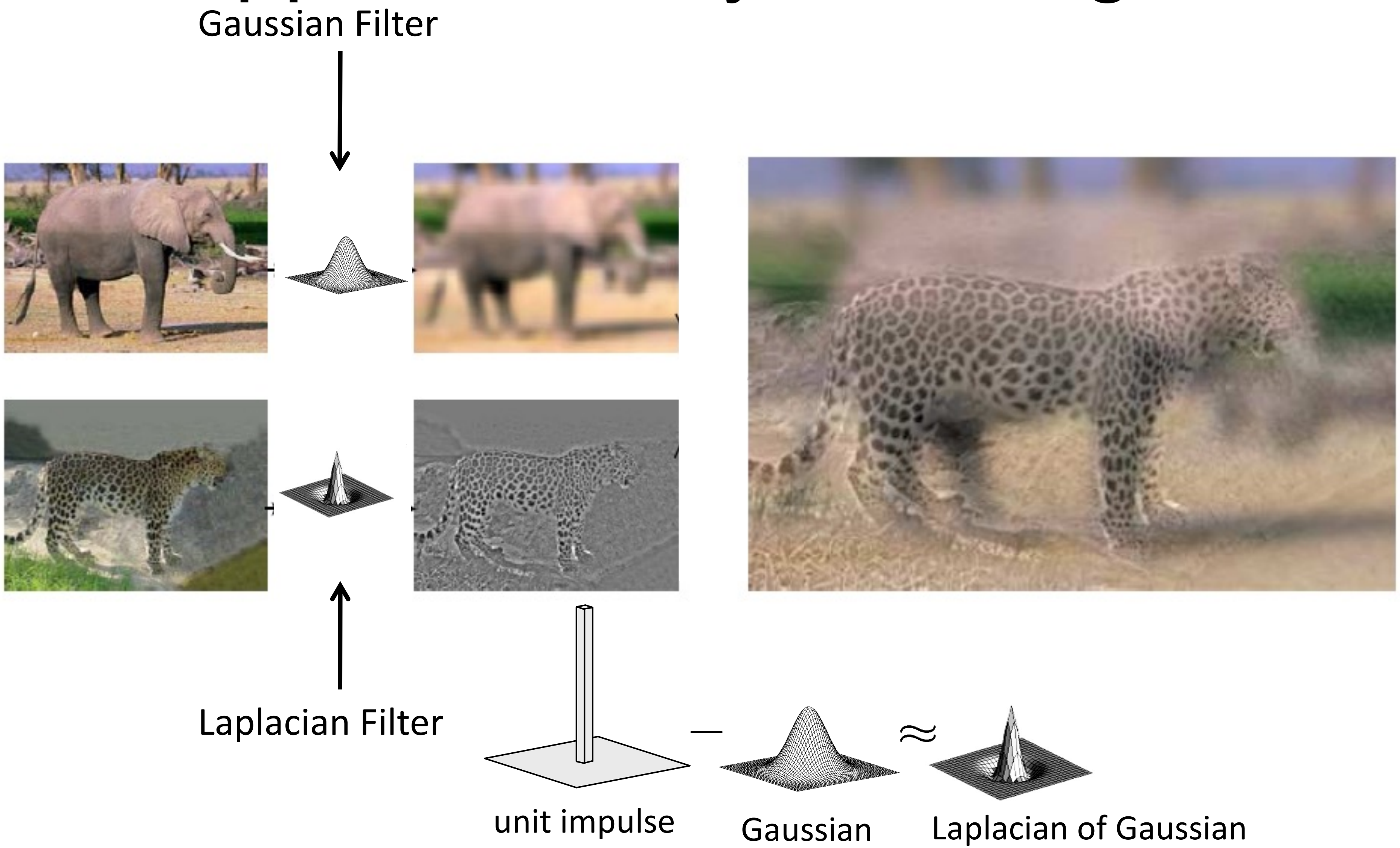
Application: Hybrid Images

unit

Slide credit: Kristen Grauman

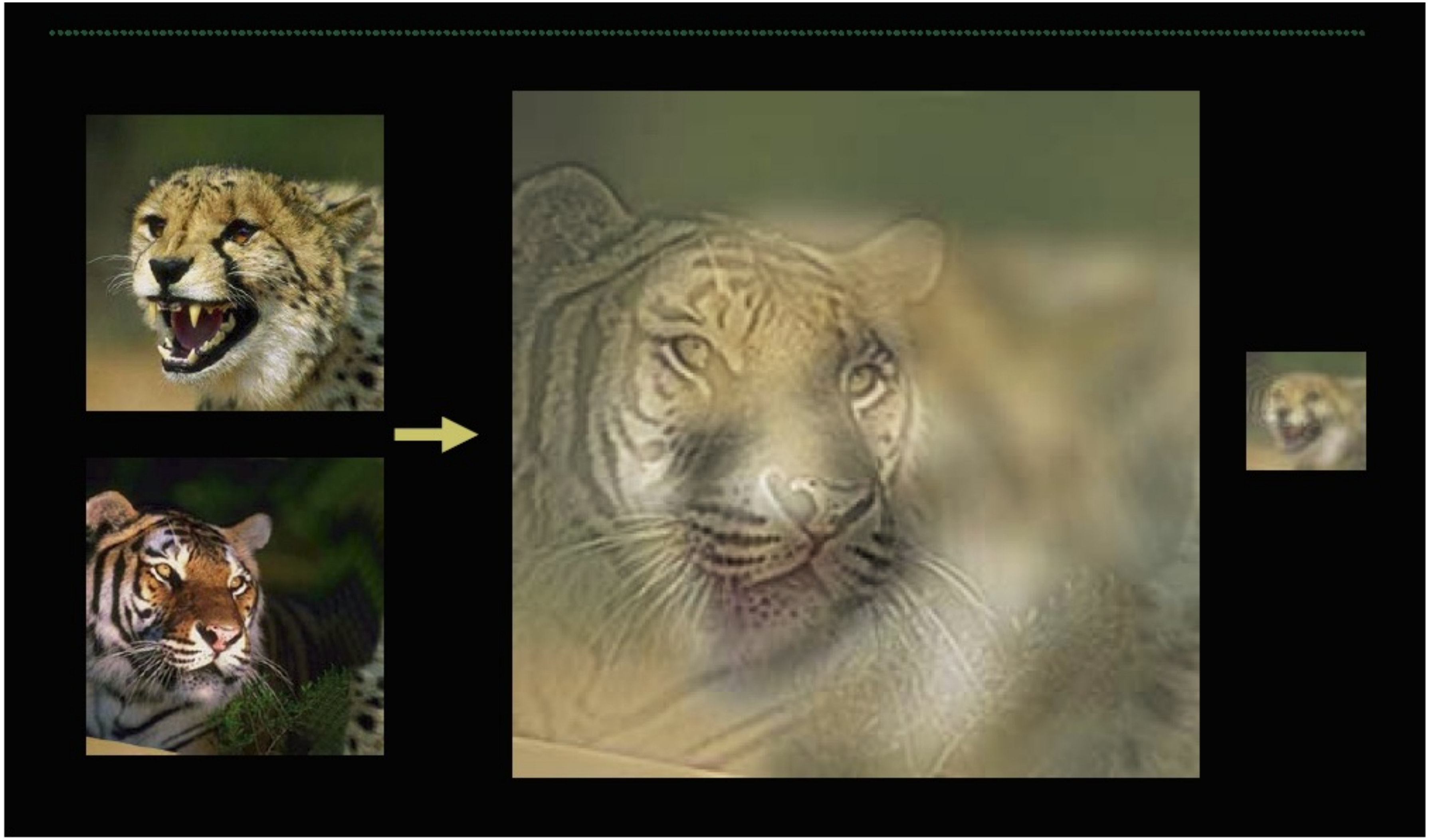
A. Oliva, A. Torralba, P.G. Schyns, "Hybrid Images," SIGGRAPH 2006

Application: Hybrid Images



A. Oliva, A. Torralba, P.G. Schyns, “Hybrid Images,” SIGGRAPH 2006

Slide credit: Kristen Grauman



Slide credit: Kristen Grauman

Aude Oliva & Antonio Torralba & Philippe G Schyns, SIGGRAPH 2006

Changing expression

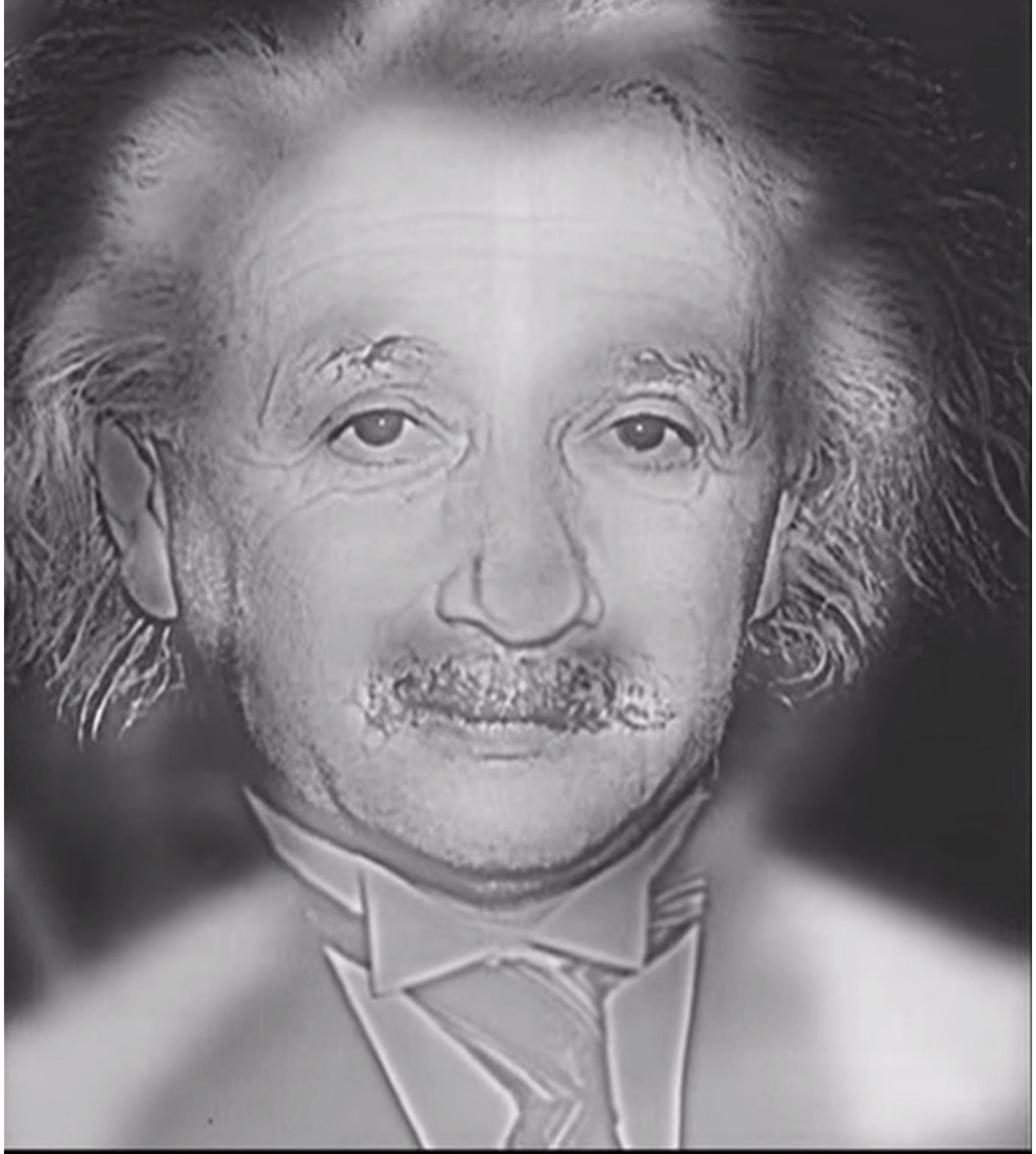
SIGGRAPH2006

Sad ⟷ Surprised

Slide credit: Kristen Grauman

Aude Oliva & Antonio Torralba & Philippe G Schyns, SIGGRAPH 2006

If you see Marilyn Monroe rather than Albert Einstein in this photo, you may need glasses.



Recall: image filtering

- Compute a function of the local neighborhood at each pixel in the image
  - Function specified by a "filter" or mask saying how to combine values from neighbors.

- Uses of filtering:
  - Enhance an image (denoise, resize, etc)
  - Extract information (texture, edges, etc)
  - Detect patterns (template matching)

Slide credit: Kristen Grauman, Adapted from Derek Hoiem

Edge Detection

Image Gradients and Edge Detection

Edge detection

- **Goal**: map image from 2d array of pixels to a set of curves or line segments or contours.
- **Why?**

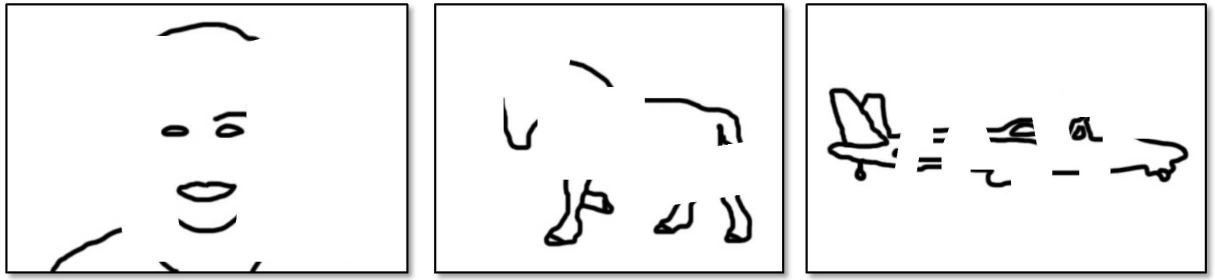


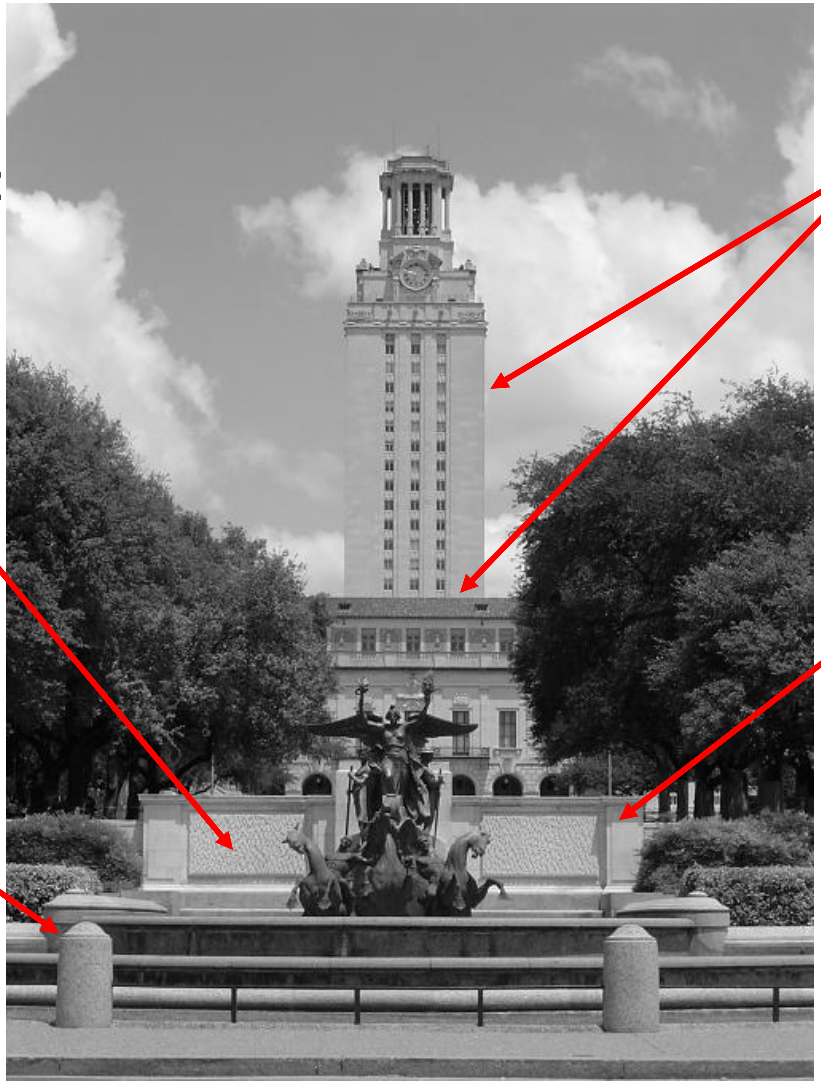
Figure from J. Shotton et al., PAMI 2007

- **Main idea**: look for strong gradients, post-process

Slide credit: Kristen Grauman

What causes an edge?

Reflectance change: appearance information, texture



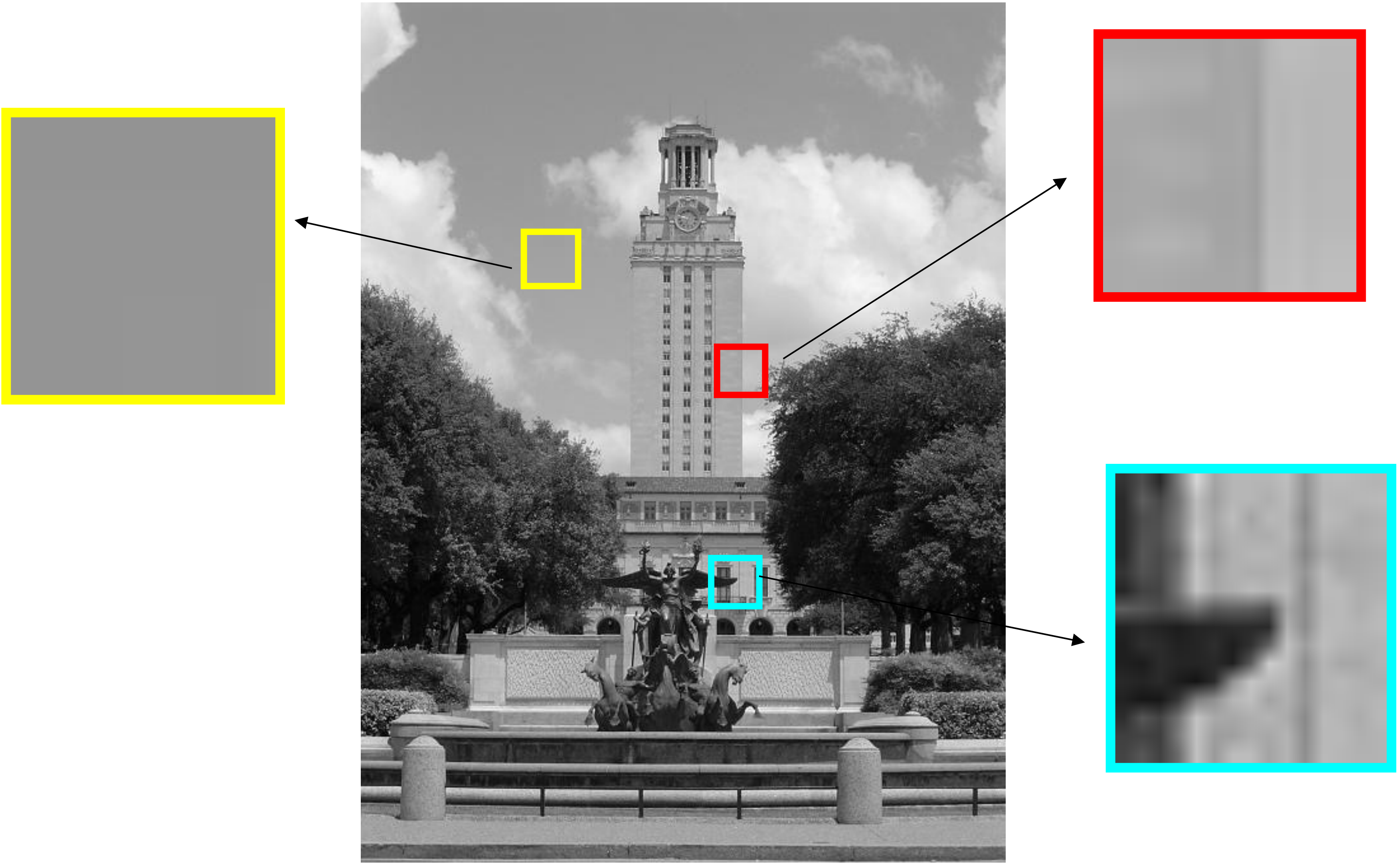
Depth discontinuity: object boundary

Cast shadows

Change in surface orientation: shape

Slide credit: Kristen Grauman

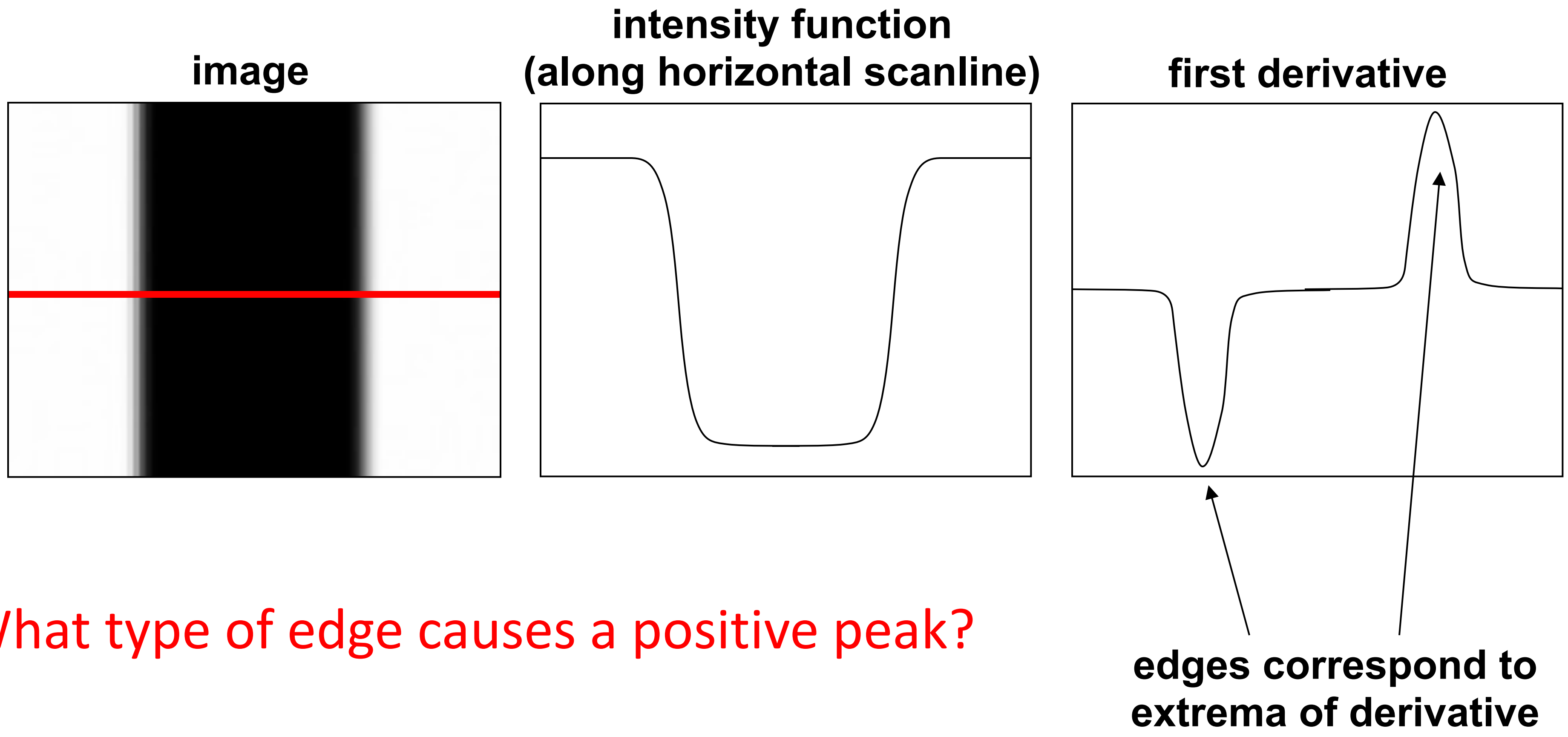
Edges/gradients and invariance



Slide credit: Kristen Grauman

Derivatives and edges

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



What type of edge causes a positive peak?

Slide credit: Svetlana Lazebnik

Derivatives with convolution

For 2D function, f(x,y), the partial derivative is:

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon, y) - f(x,y)}{\varepsilon}$$

For discrete data, we can approximate using finite differences:

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1, y) - f(x,y)}{1}$$

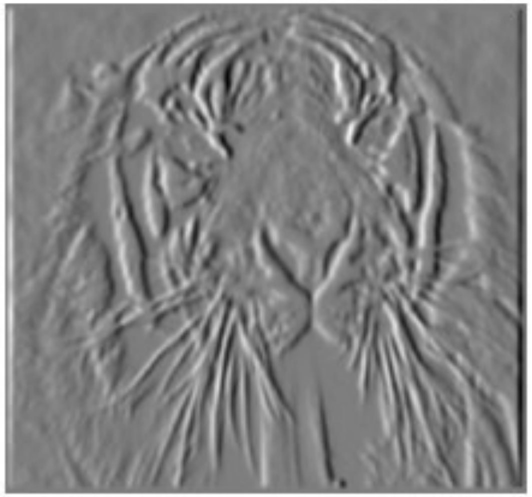
To implement above as convolution, what would be the associated filter?

-1	**1**

Slide credit: Kristen Grauman

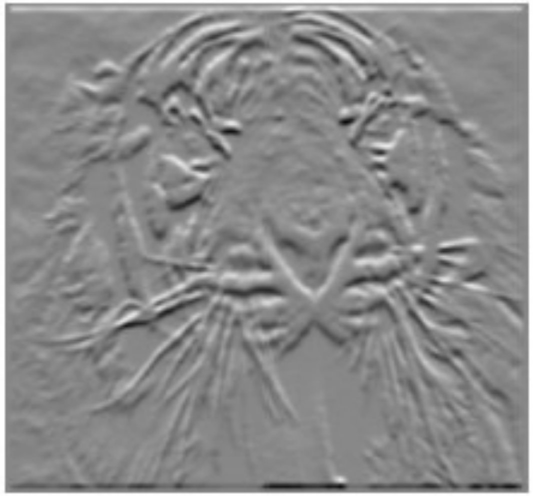
Partial derivatives of an image

$$\frac{\partial f(x,y)}{\partial x}$$



-1	1

$$\frac{\partial f(x,y)}{\partial y}$$



-1
1

?

or

1
-1

Which shows changes with respect to x?

(showing filters for correlation)

Slide credit: Kristen Grauman

Assorted finite difference filters

Prewitt

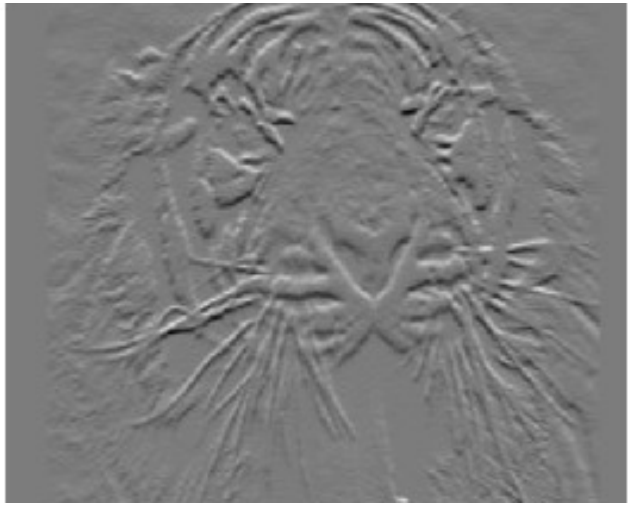
$$M_x = \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \qquad M_y = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

Roberts

$$M_x = \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array} \qquad M_y = \begin{array}{|c|c|}\hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array}$$

Sobel

$$M_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \qquad \boxed{M_y = \begin{array}{|c|c|c|}\hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array}}$$



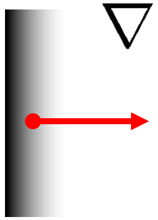
Slide credit: Judy Hoffman

Image gradient

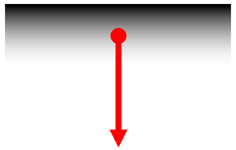
The gradient of an image:

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

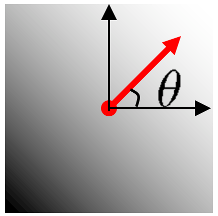
The gradient points in the direction of most rapid change in intensity



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



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



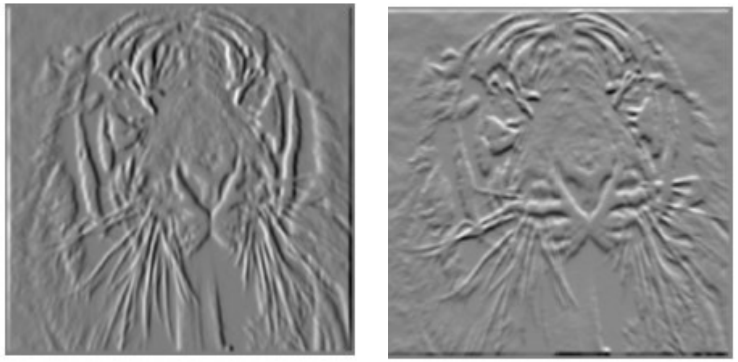
$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

The **gradient direction** (orientation of edge normal) is given by:

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

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

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

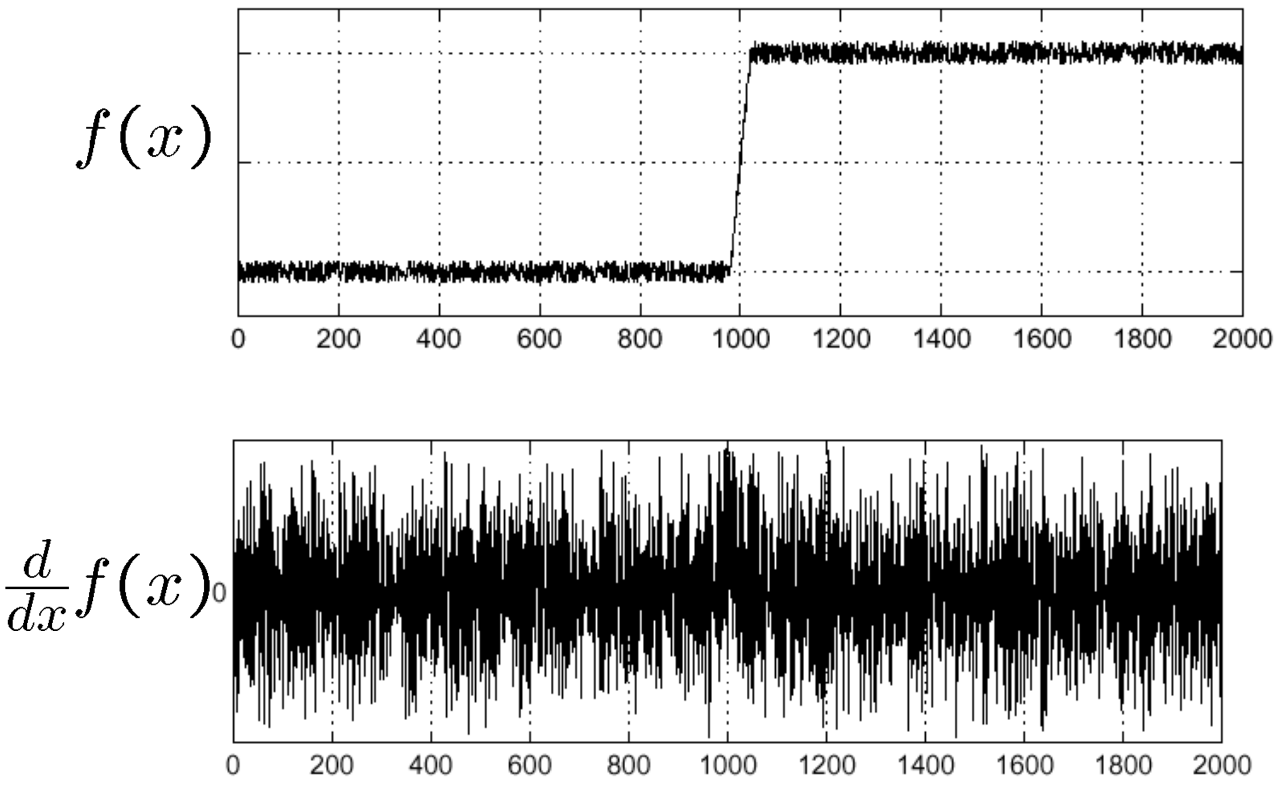


Slide credit: Steve Seitz

Effects of noise

Consider a single row or column of the image

- Plotting intensity as a function of position gives a signal



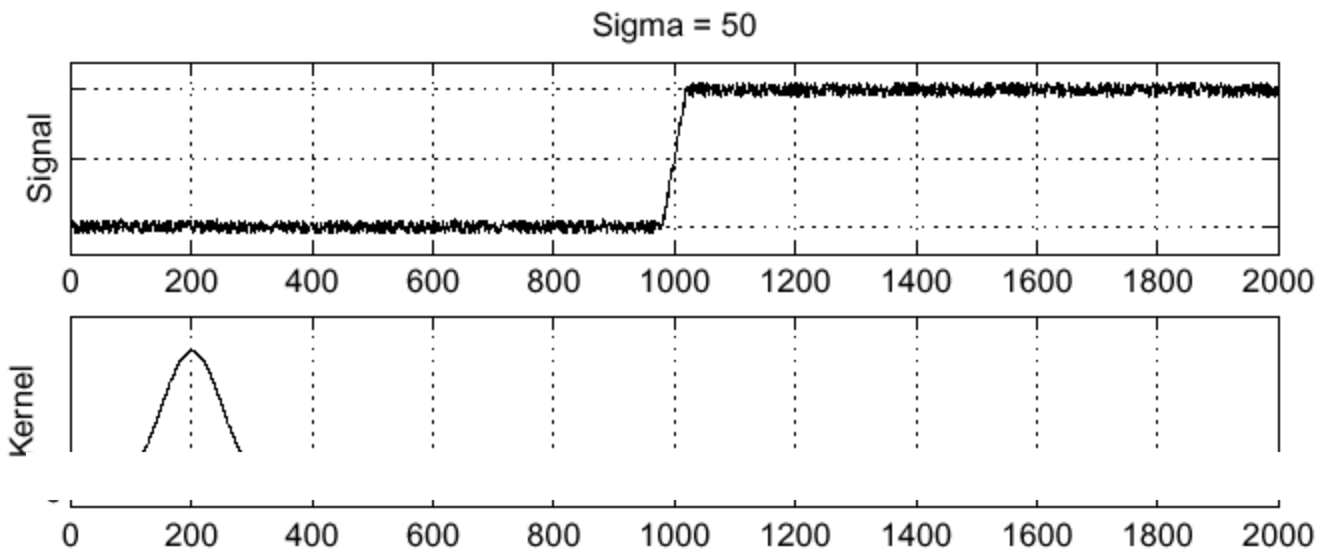
Where is the edge?

Slide credit: Steve Seitz

Solution: smooth first

f

h



Where's the edge?

Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

Slide credit: Kristen Grauman

Solution: smooth first

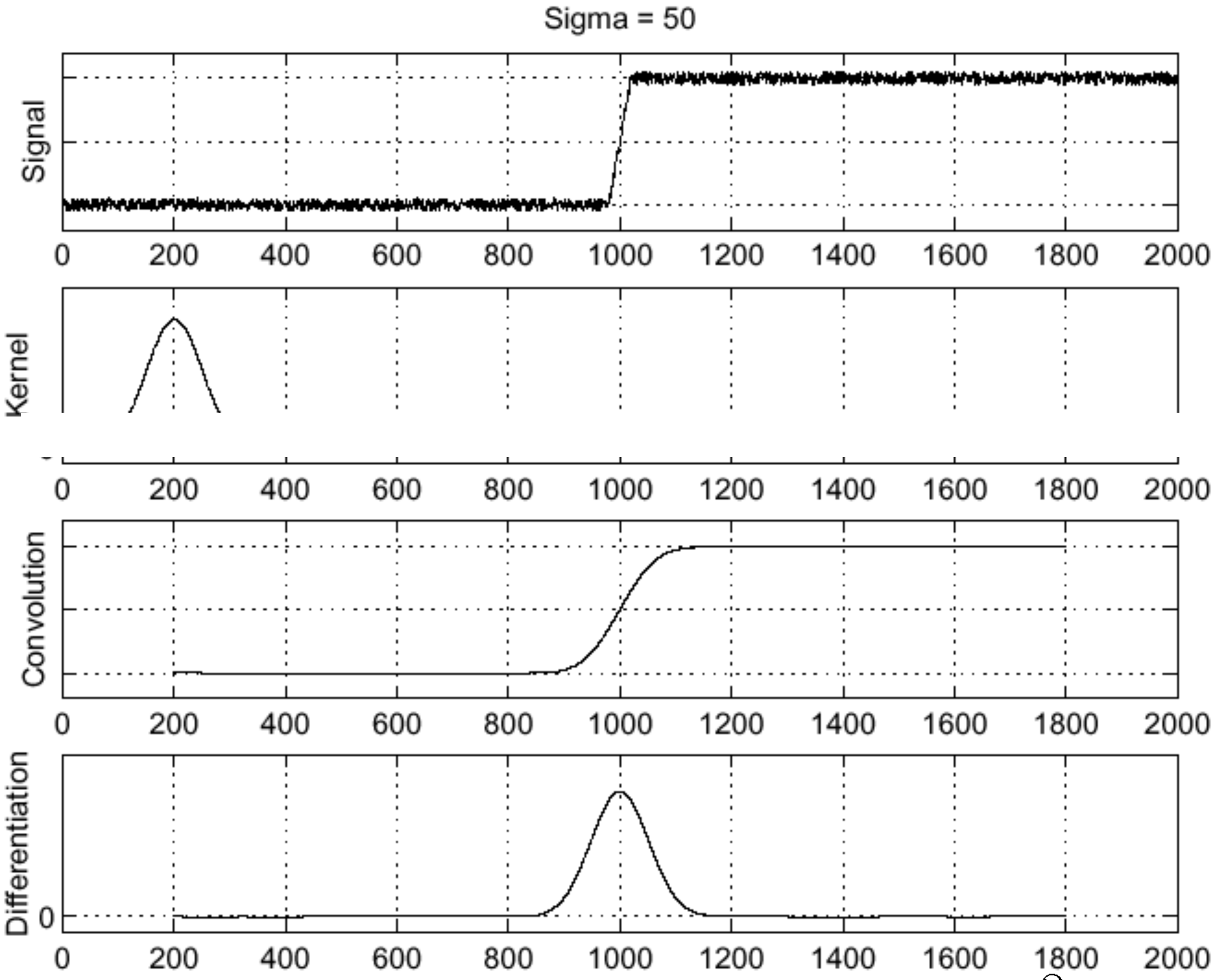
f

h

Where's the edge?

$h \star f$

$\frac{\partial}{\partial x}(h \star f)$



Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

Slide credit: Kristen Grauman

Derivative theorem of convolution

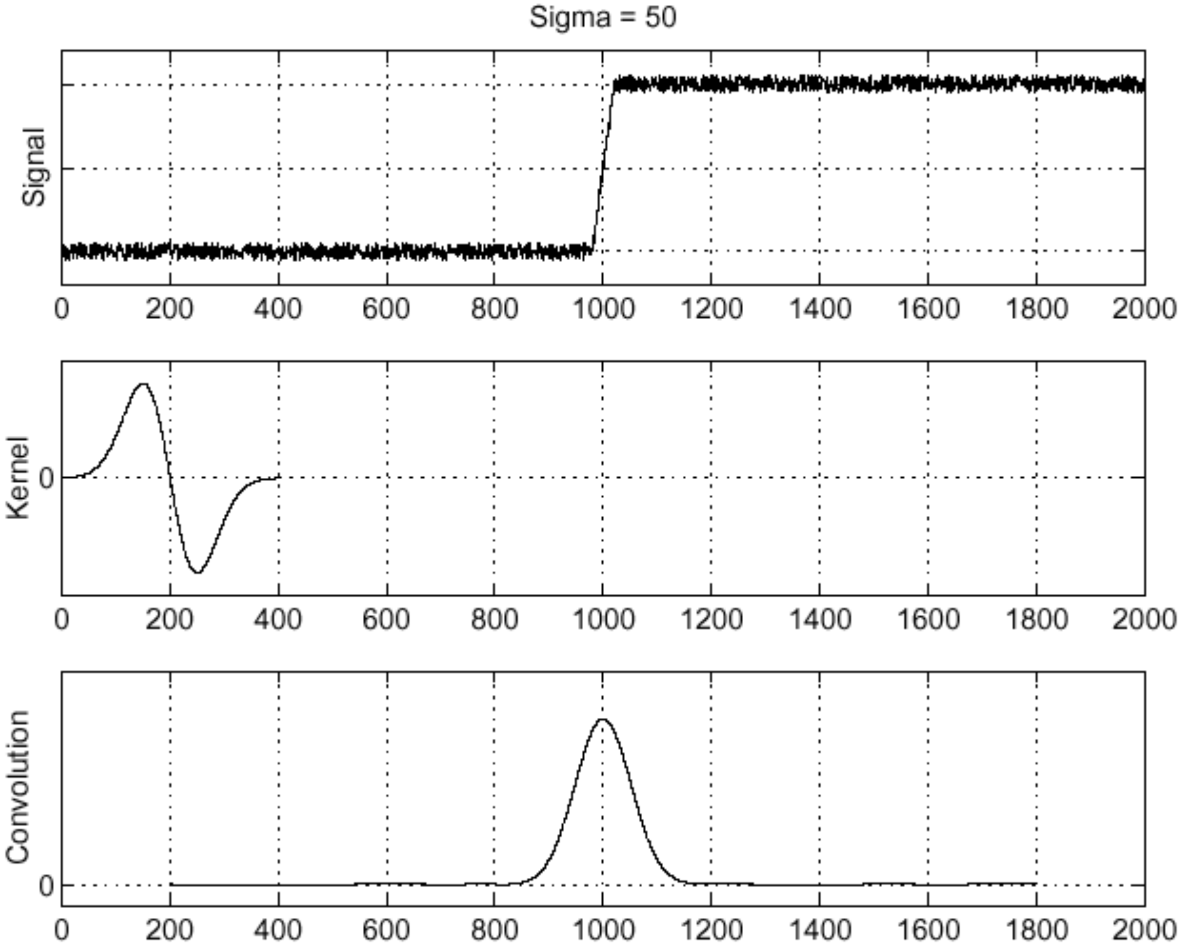
Differentiation property of convolution.

$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

f

$\frac{\partial}{\partial x}h$

$(\frac{\partial}{\partial x}h) \star f$

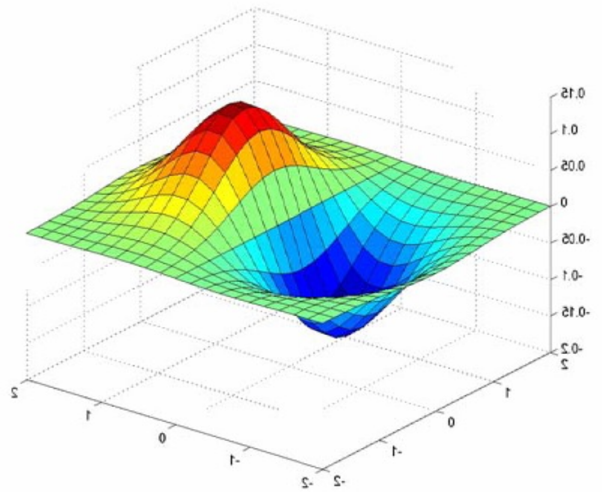


Slide credit: Steve Seitz

Derivative of Gaussian filters

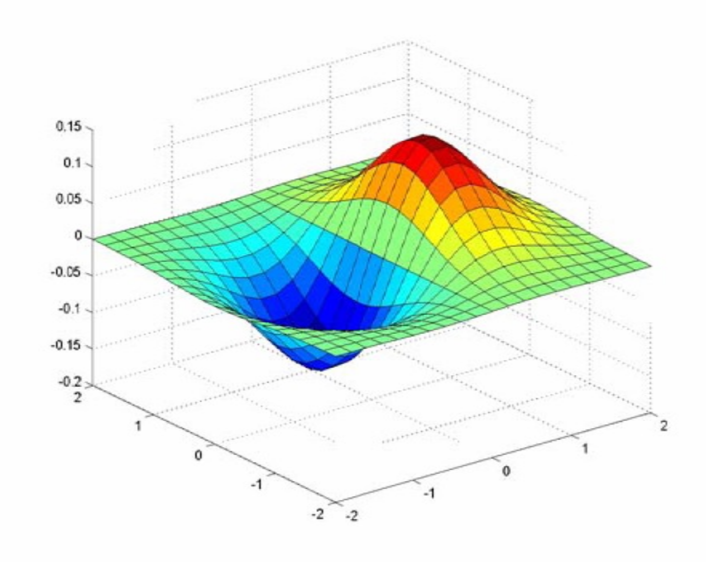
$$(I \otimes g) \otimes h = I \otimes (g \otimes h)$$

$$\begin{bmatrix} 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0219 & 0.0983 & 0.1621 & 0.0983 & 0.0219 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \end{bmatrix} \otimes \begin{bmatrix} 1 & -1 \end{bmatrix}$$

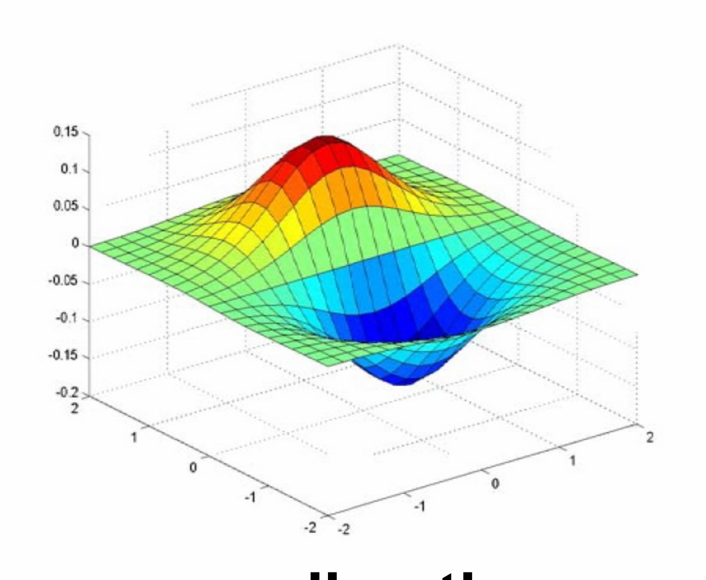


Slide credit: Kristen Grauman

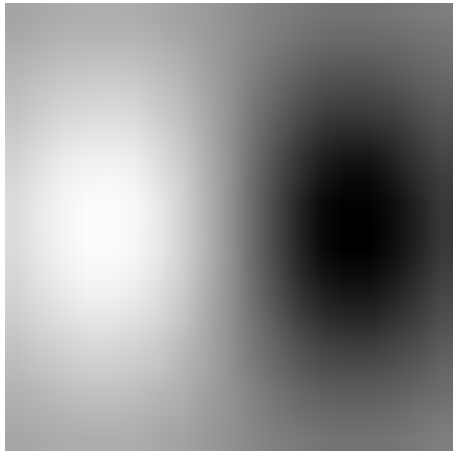
Derivative of Gaussian filters

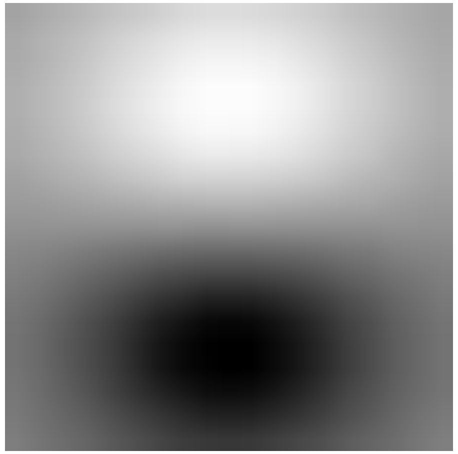


***x*-direction**



***y*-direction**





Slide credit: Svetlana Lazebnik

Derivative theorem of convolution

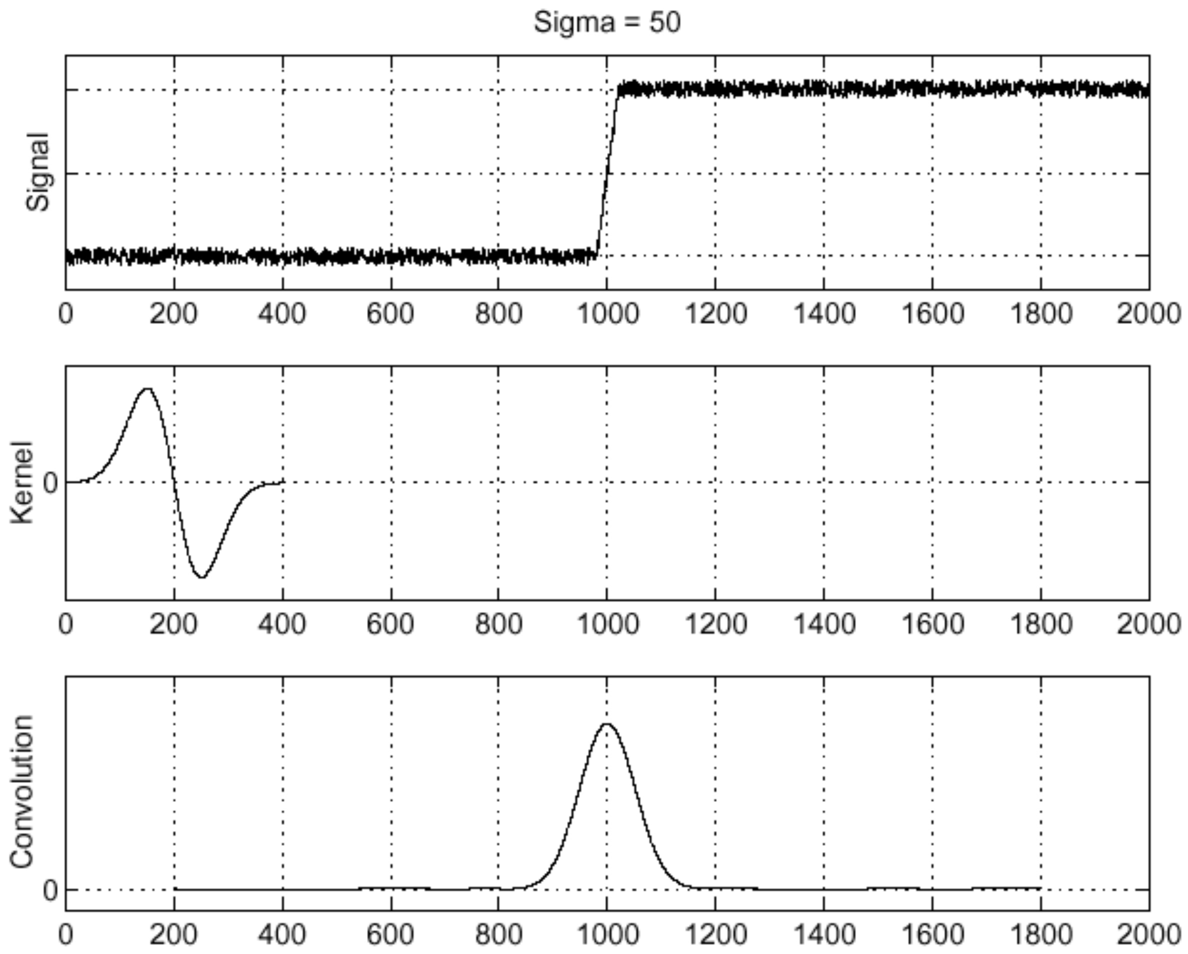
$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

How to find min/max of function?

f

$\frac{\partial}{\partial x}h$

$(\frac{\partial}{\partial x}h) \star f$



Slide credit: Steve Seitz

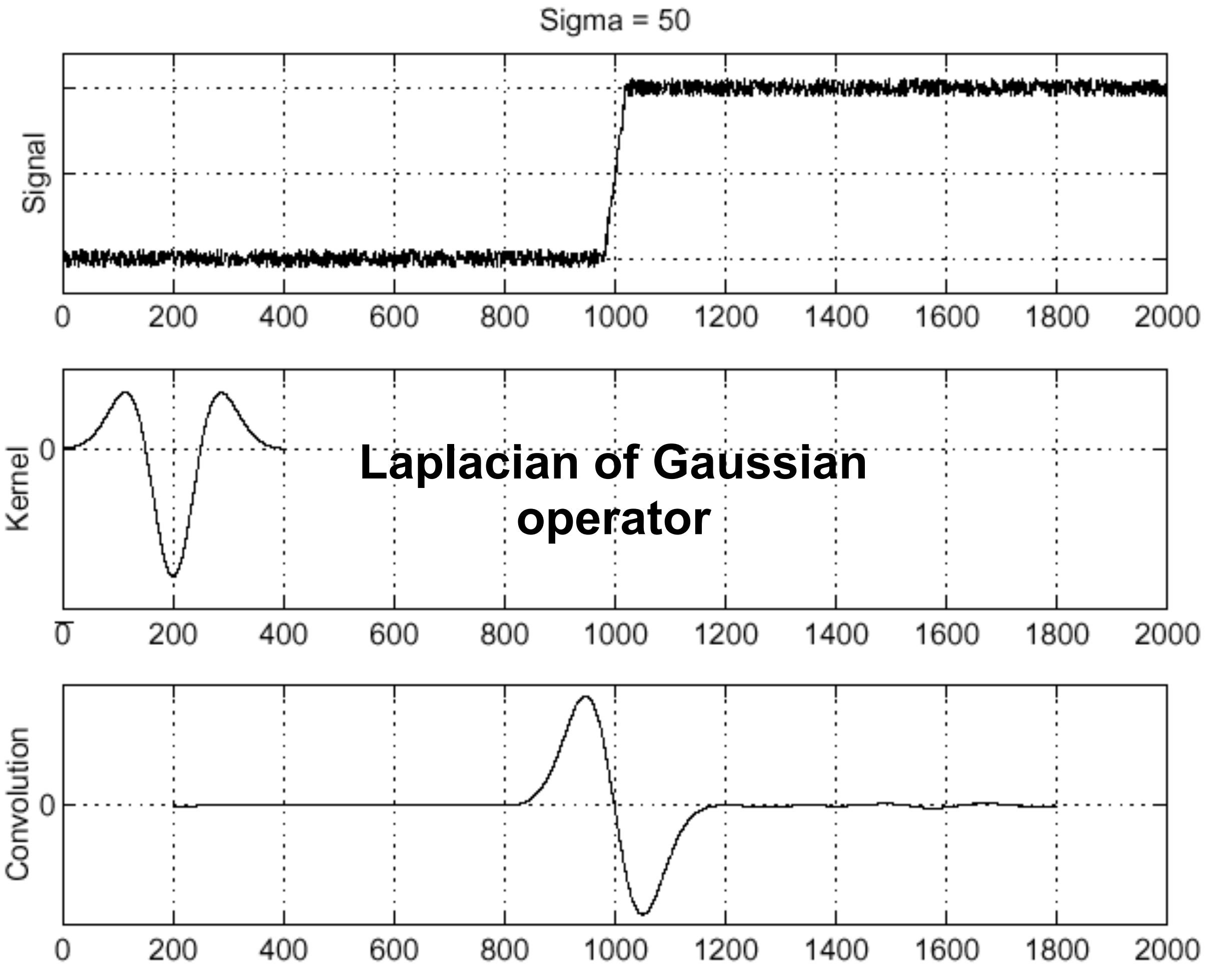
Laplacian of Gaussian

Consider $\frac{\partial^2}{\partial x^2}(h \star f)$

f

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

$(\frac{\partial^2}{\partial x^2}h) \star f$

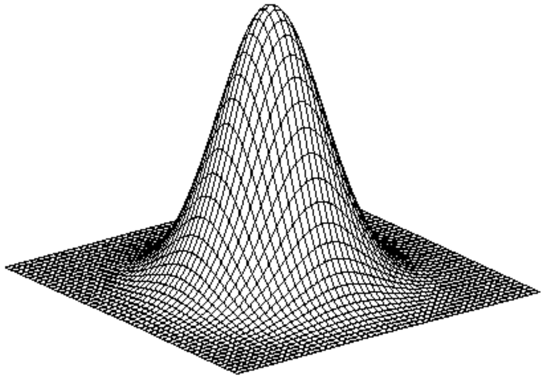


Where is the edge?

Zero-crossings of bottom graph

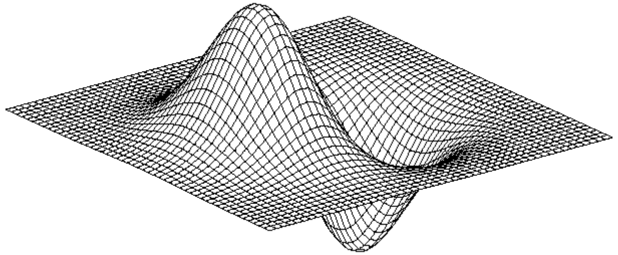
Slide credit: Steve Seitz

2D edge detection filters



Gaussian

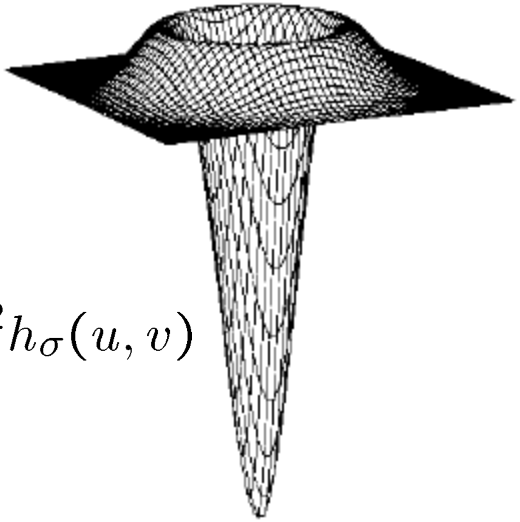
$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



derivative of Gaussian

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

Laplacian of Gaussian



$$\nabla^2 h_\sigma(u,v)$$

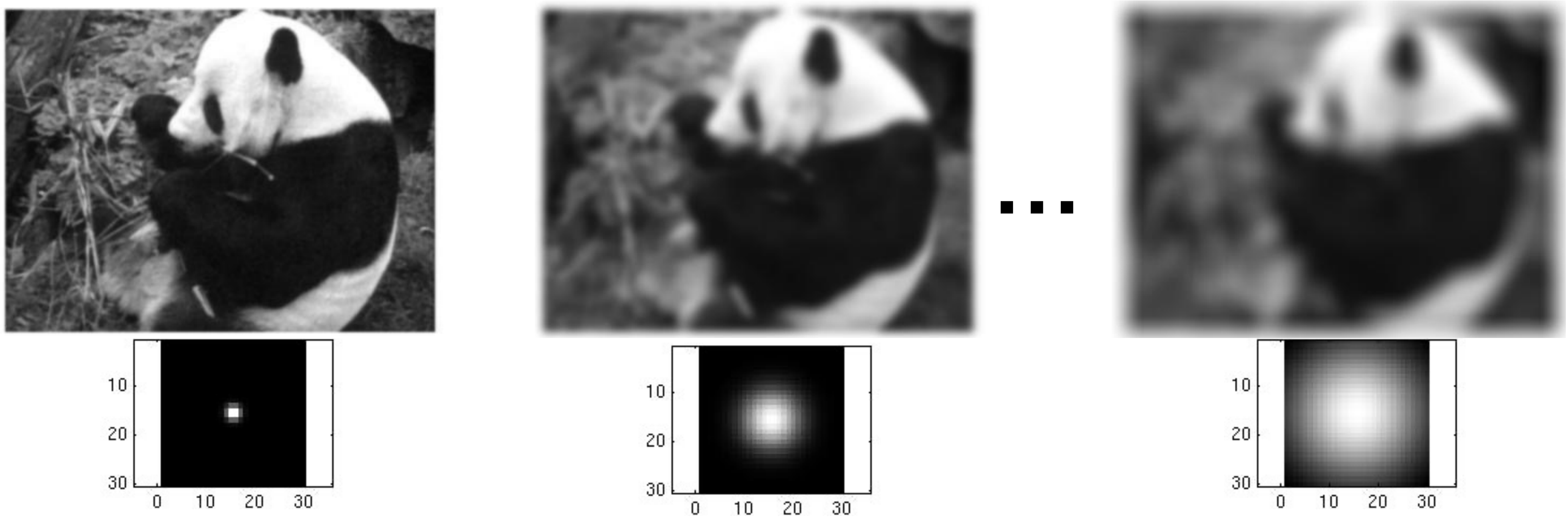
- ∇^2 is the Laplacian operator:

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

Slide credit: Steve Seitz

Smoothing with a Gaussian

Recall: parameter σ is the "scale" / "width" / "spread" of the Gaussian kernel, and controls the amount of smoothing.



Slide credit: Kristen Grauman

Effect of σ on derivatives

σ = 1 pixel

σ = 3 pixels

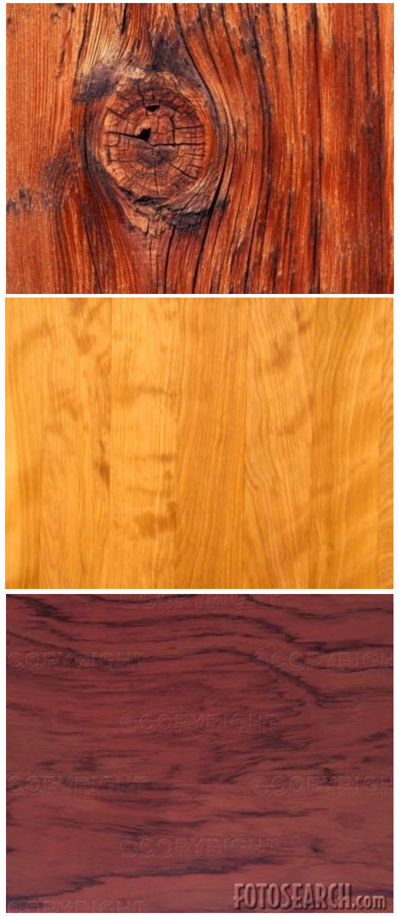
The apparent structures differ depending on Gaussian’s scale parameter.

Larger values: larger scale edges detected
Smaller values: finer features detected

Slide credit: Kristen Grauman

So, what scale to choose?

It depends what we’re looking for.



Slide credit: Kristen Grauman

Gradients → Edges

Primary edge detection steps:

1. Smoothing: suppress noise

2. Edge enhancement: filter for contrast

3. Edge localization

   Determine which local maxima from filter output are actually edges vs. noise

   - Threshold, Thin

Slide credit: Kristen Grauman

Thresholding

- Choose a threshold value t
- Set any pixels less than t to zero (off)
- Set any pixels greater than or equal to t to one (on)

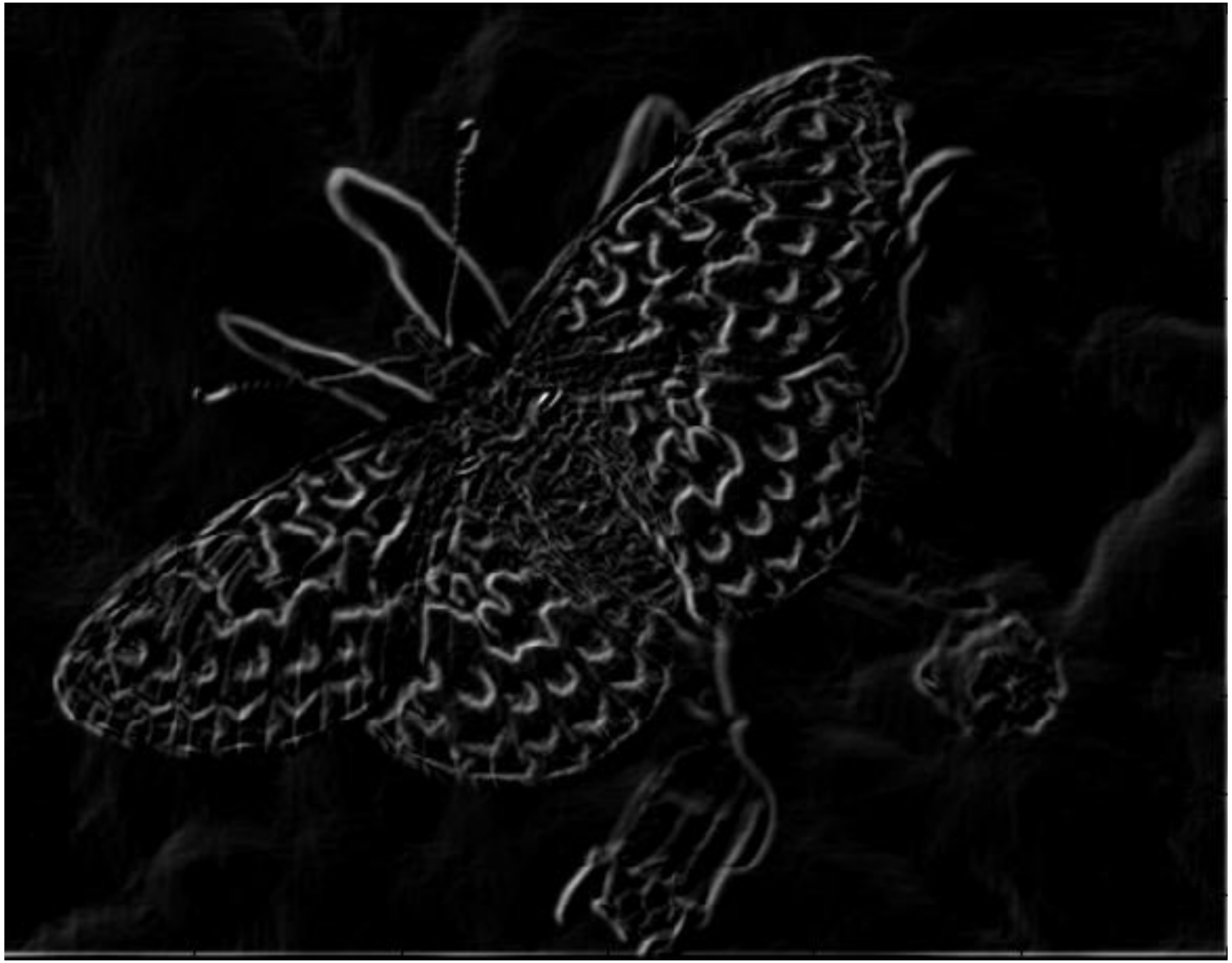
$$v = \begin{cases} 0; & p < t \\ 1; & p \geq t \end{cases}$$

Slide credit: Judy Hoffman

Original image

Slide credit: Kristen Grauman

Gradient magnitude image



Slide credit: Kristen Grauman

Thresholding gradient with a lower threshold

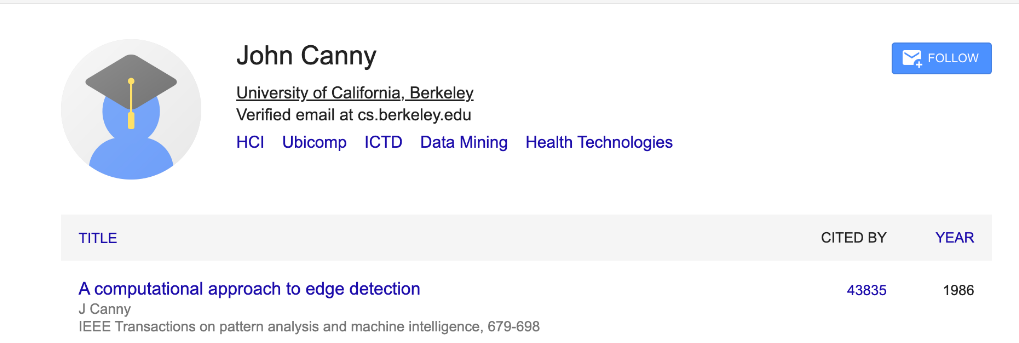
Slide credit: Kristen Grauman

Thresholding gradient with a higher threshold

Slide credit: Kristen Grauman

Canny edge detector

- Filter image with derivative of Gaussian
- Find magnitude and orientation of gradient
- **Non-maximum suppression**:
  - Thin wide "ridges" down to single pixel width
- **Linking and thresholding (hysteresis)**:
  - Define two thresholds: low and high
  - Use the high threshold to start edge curves and the low threshold to continue them



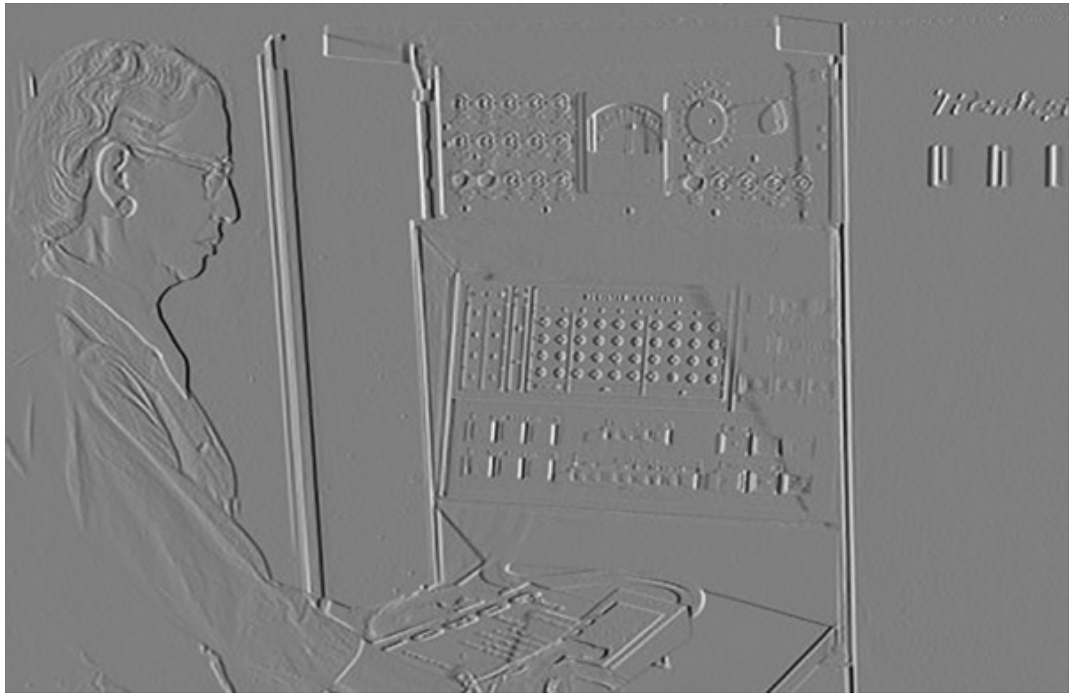
Slide credit: David Lowe, Fei-Fei Li

The Canny edge detectors

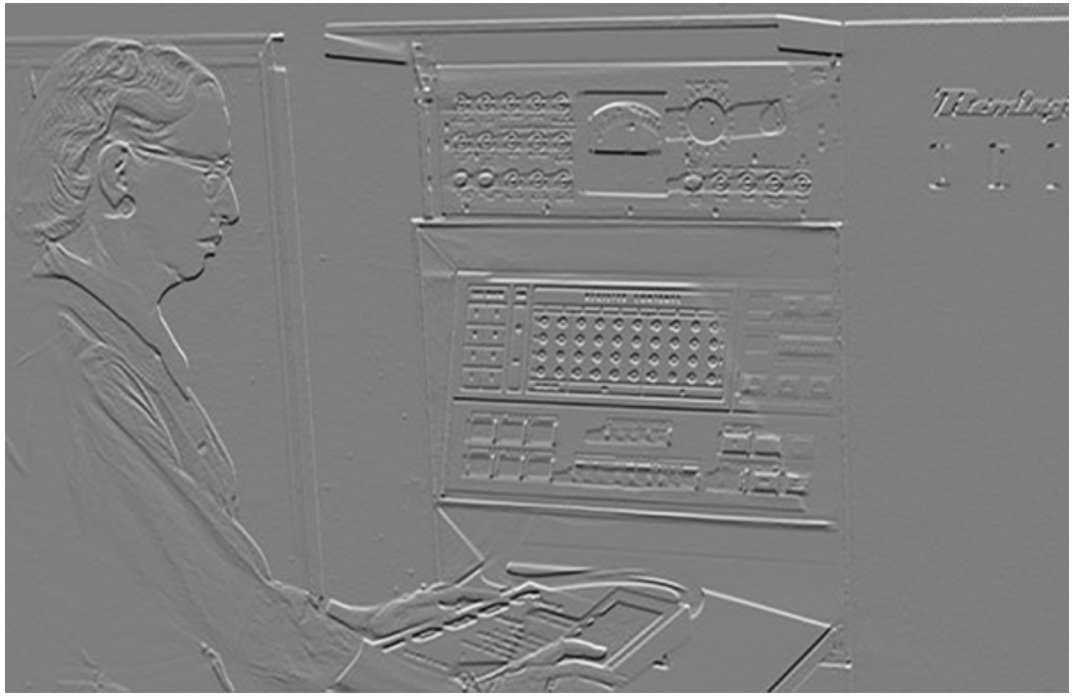
original image

Slide credit: Judy Hoffman

The Canny edge detectors



dX



dY

Gradients in X and Y

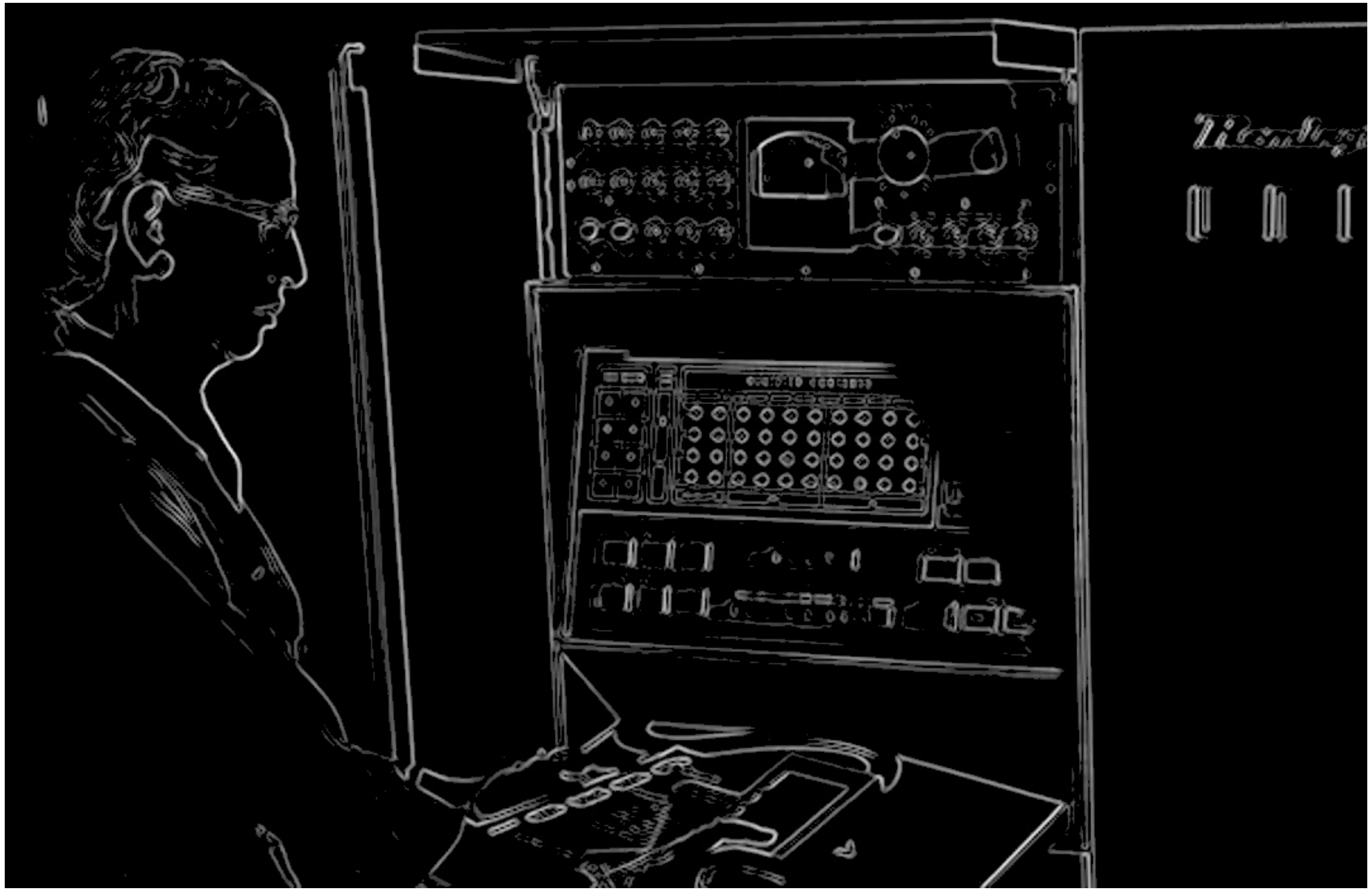
Slide credit: Judy Hoffman

The Canny edge detector

Norm of Image Gradient

Slide credit: Judy Hoffman

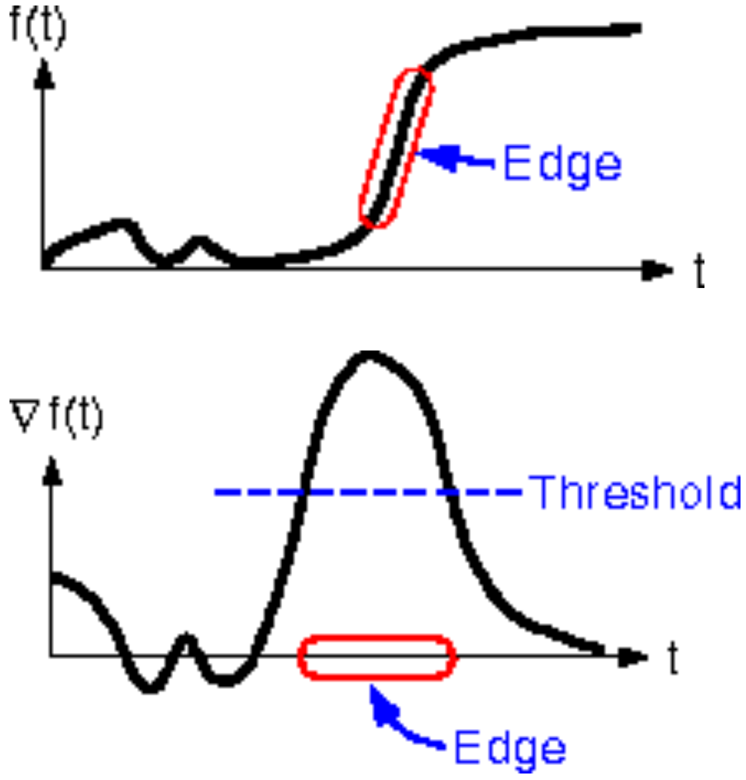
The Canny edge detector

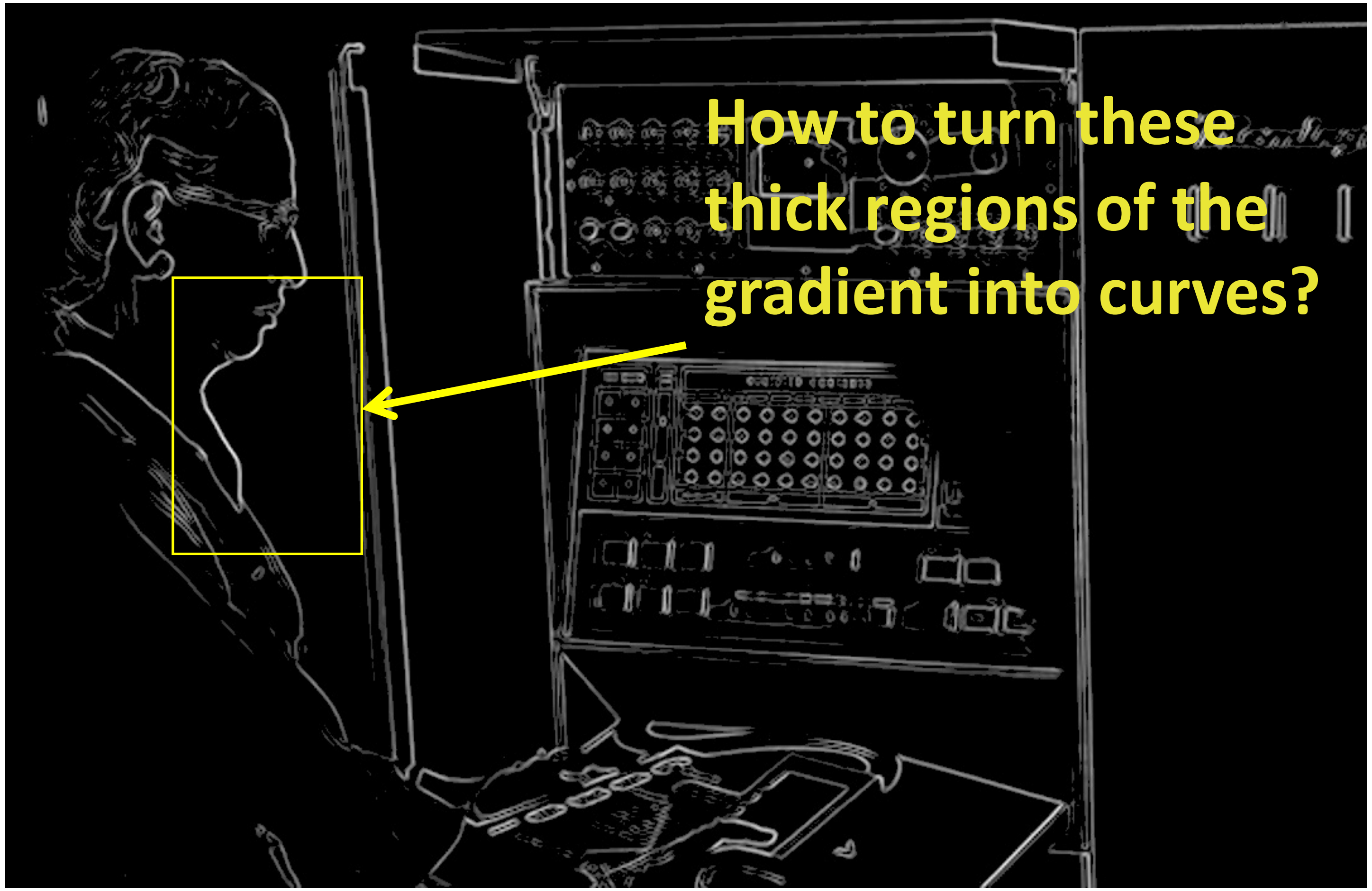


Thresholding

Slide credit: Judy Hoffman

The Canny edge detector

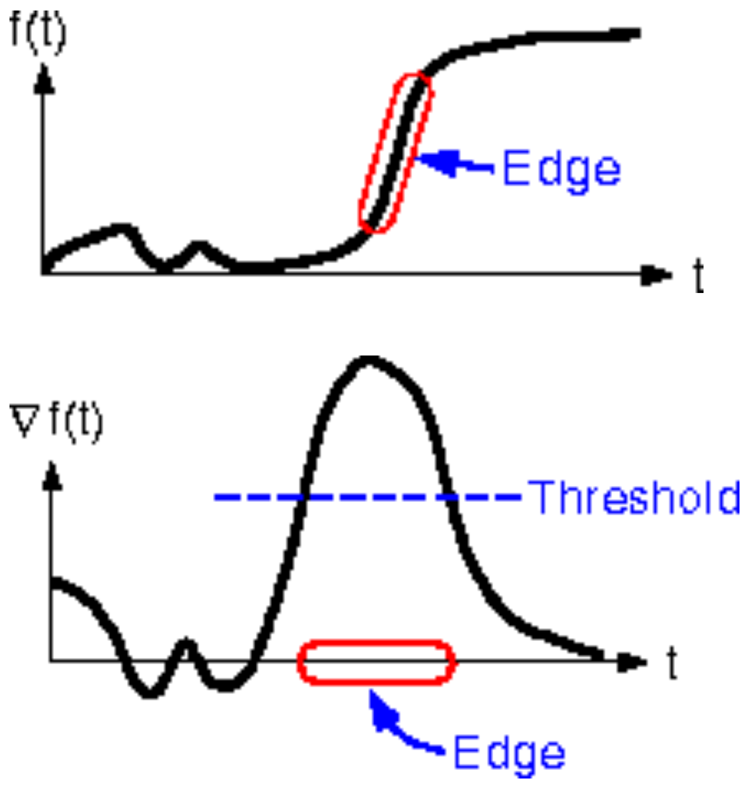




Thresholding

Slide credit: Judy Hoffman

The Canny edge detector

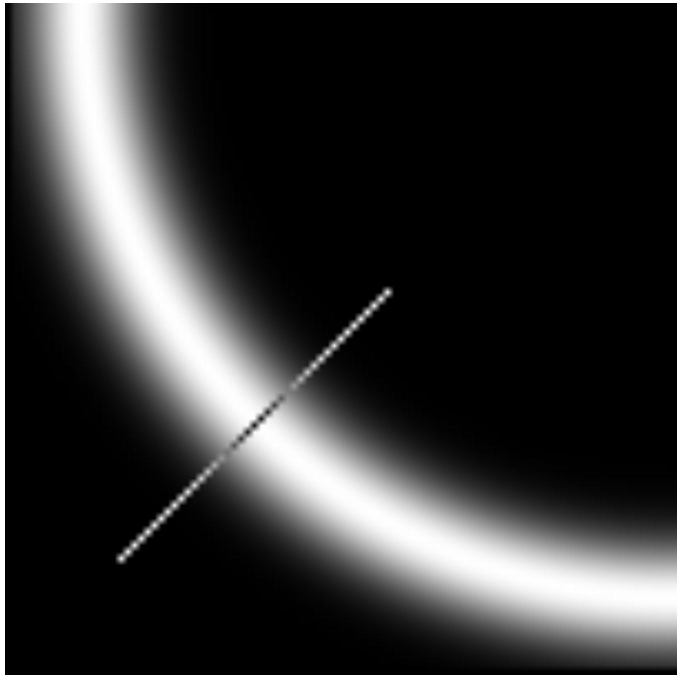


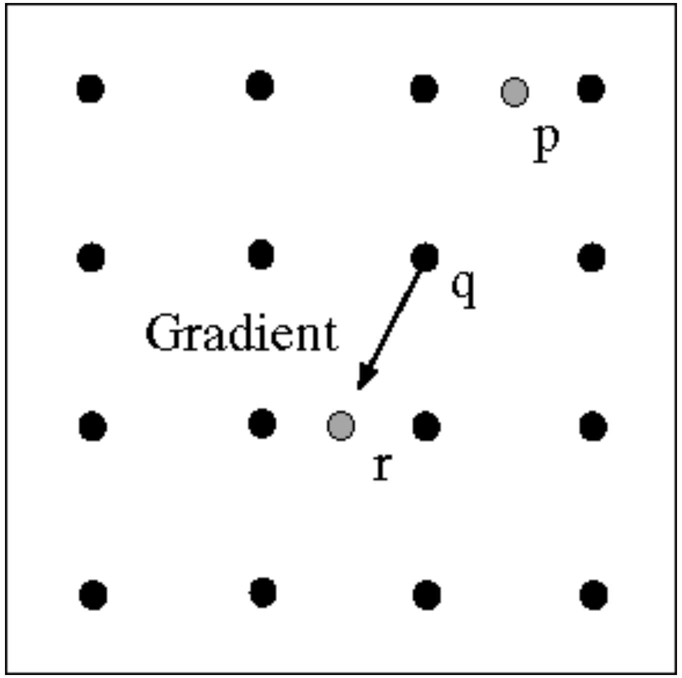
Thresholding

How to turn these thick regions of the gradient into curves?

Slide credit: Judy Hoffman

Non-maximum suppression



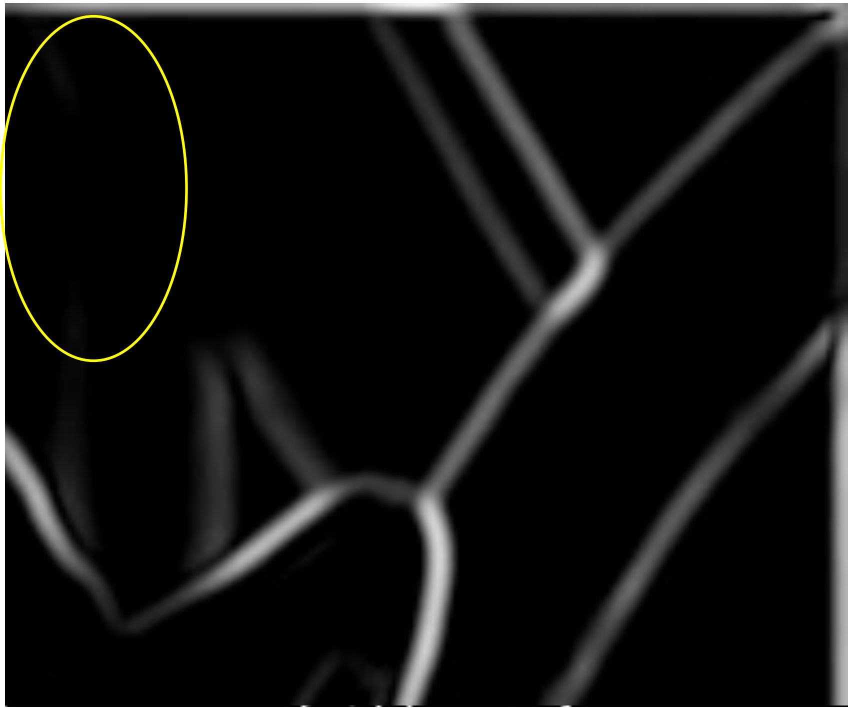


Check if pixel is local maximum along gradient direction
Select single max across width of the edge
Requires checking interpolated pixels p and r

Slide credit: Kristen Grauman

The Canny edge detector

Problem: pixels along this edge didn't survive the thresholding



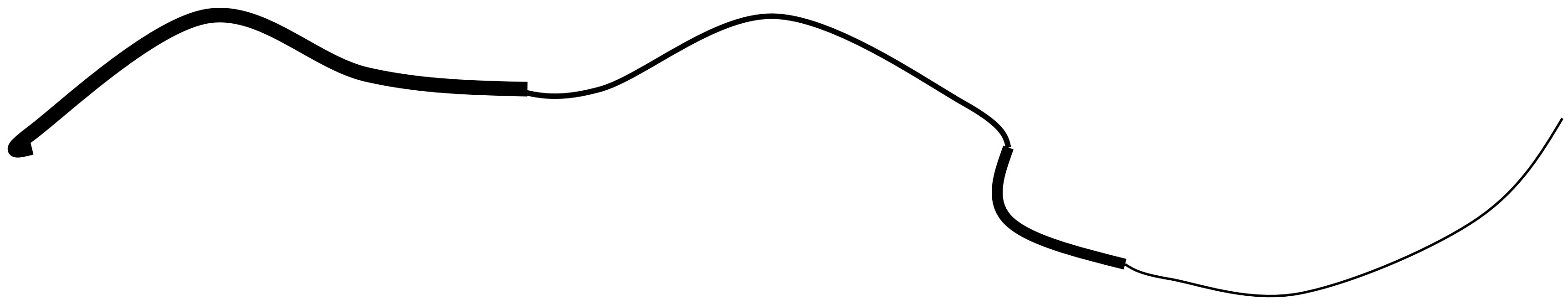
thinning

(non-maximum suppression)

Slide credit: Kristen Grauman

Hysteresis thresholding

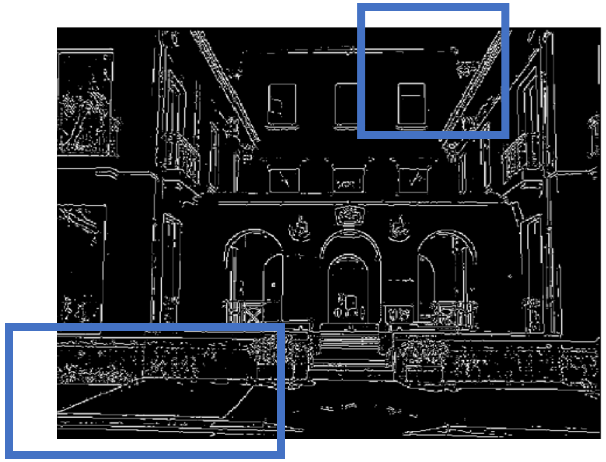
- Use a high threshold to start edge curves, and a low threshold to continue them.



Slide credit: Steve Seitz

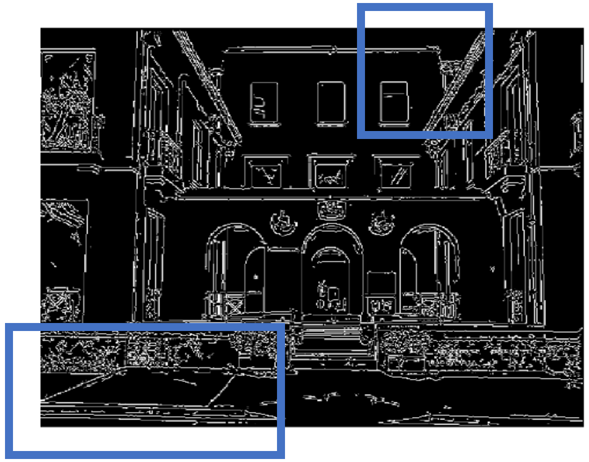
Hysteresis thresholding

original image



high threshold
(strong edges)

low threshold
(weak edges)



hysteresis threshold

Slide credit: Fei-Fei Li

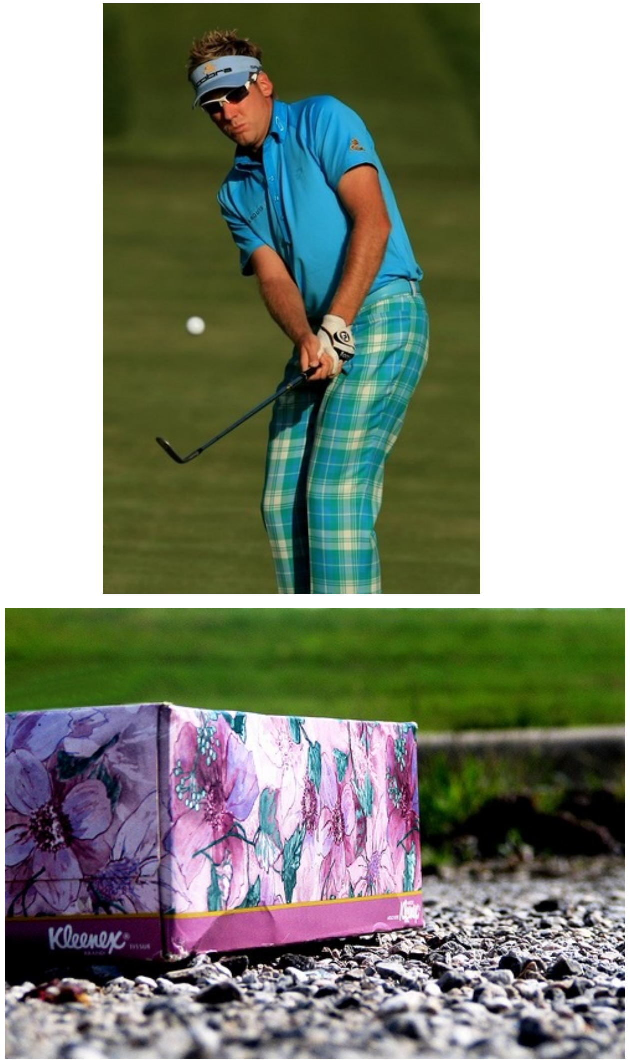
Recap: Canny edge detector

- Filter image with derivative of Gaussian
- Find magnitude and orientation of gradient
- **Non-maximum suppression**:
  - Thin wide "ridges" down to single pixel width
- **Linking and thresholding** (**hysteresis**):
  - Define two thresholds: low and high
  - Use the high threshold to start edge curves and the low threshold to continue them

Slide credit: David Lowe, Fei-Fei Li

Low-level edges vs. perceived contours

Background

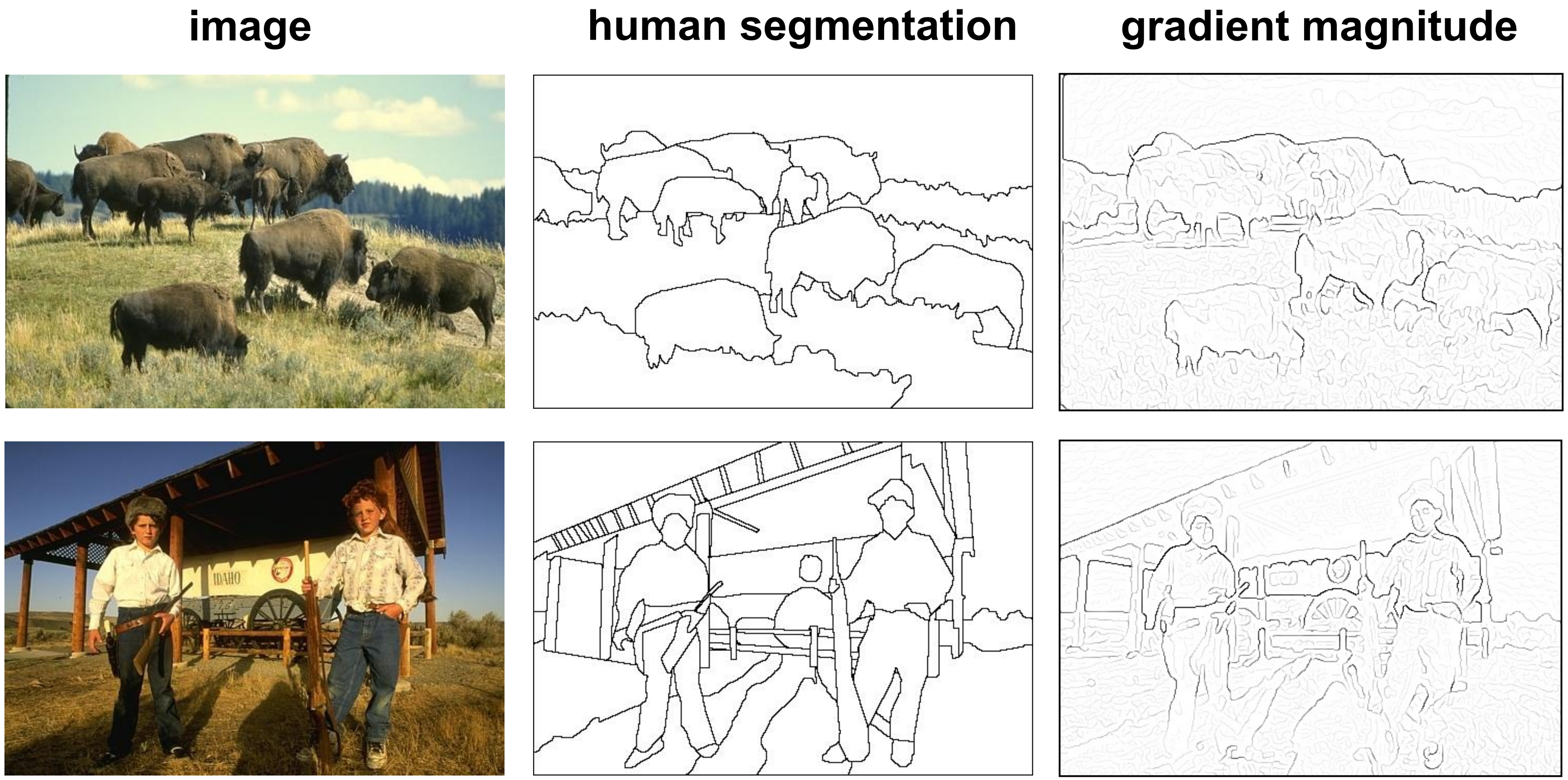


Texture

Shadows

Slide credit: Kristen Grauman

Low-level edges vs. perceived contours



- Berkeley segmentation database: http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

Slide credit: Svetlana Lazebnik

Questions?

See you next time!