COSC579: Edge Detection

Jeremy Bolton, PhD
Assistant Teaching Professor

GEORGETOWN UNIVERSITY

Outline

I. Edge Detection
   I. Gradient
   II. Laplacian
   III. Marr - Hildreth
   IV. Sobel
   V. Canny
   VI. Harris Structure
   VII. Smooth first (… ask questions later). Witkins

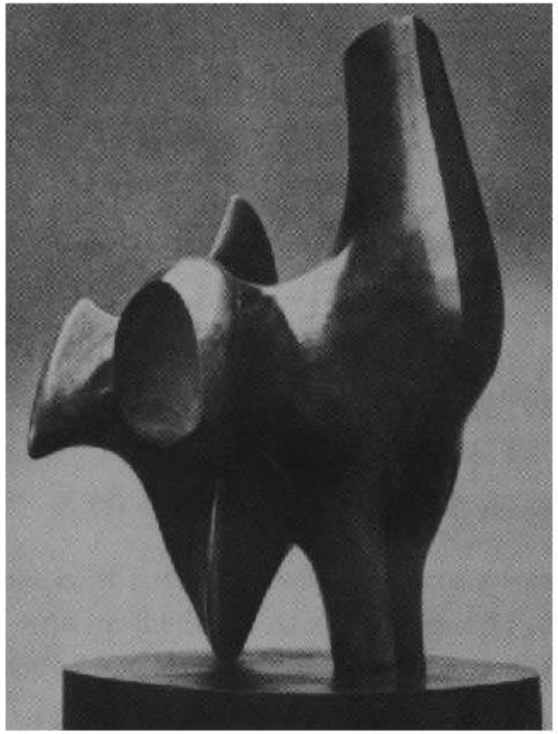
II. Filtering as Matrix Multiplication

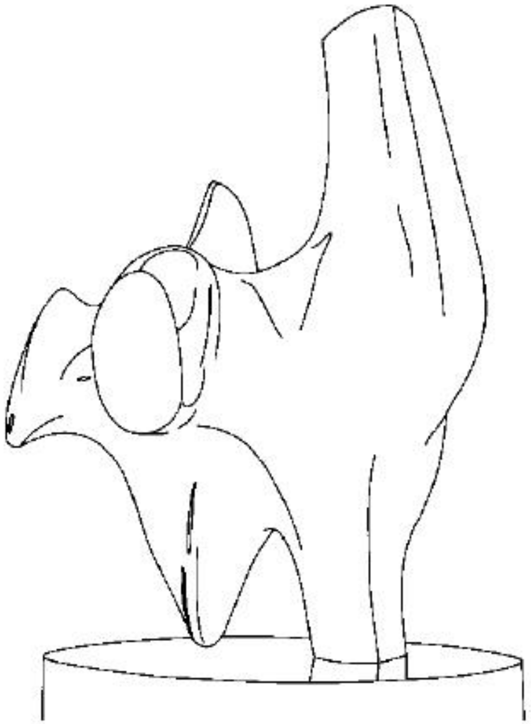
III. (Detection) Thresholding Schemes

Readings

- Read
  - Marr and Hildreth Theory of Edge Detection
  - Witkins: Gaussian Smoothing
  - Skip* Cipolla (on edge detection)
  - Szeliski 3.4.1 – 3.4.2

Edge detection

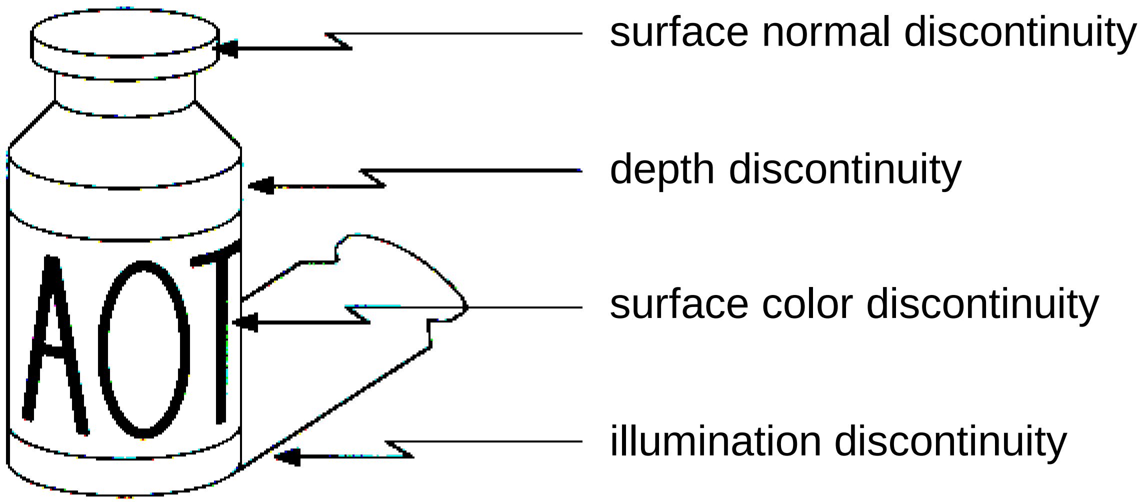




- Convert a 2D image into a set of curves
  - Extracts salient features of the scene
  - More compact than pixels

GEORGETOWN UNIVERSITY

Origin of Edges



- Edges are caused by a variety of factors

Characterizing edges

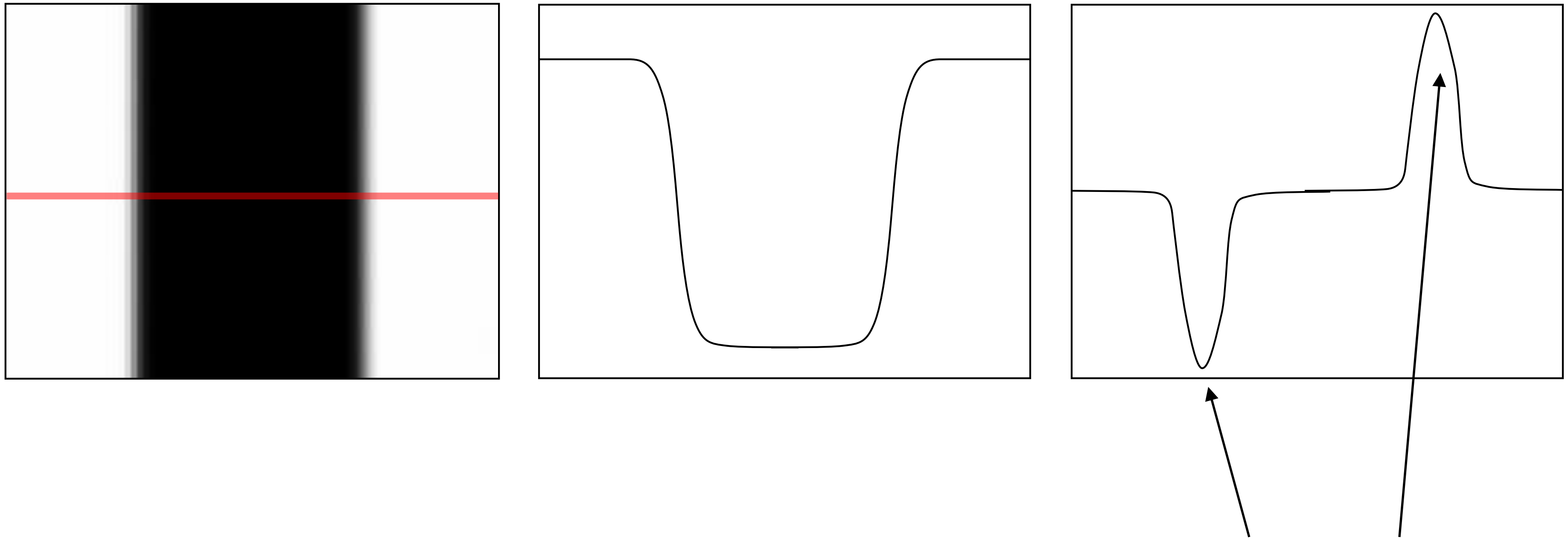
- An edge is a place of rapid change (discontinuity) in the image intensity function

Observe: Extrema of first derivative are characterized by “zero-crossings” of second derivative.

image

intensity function
(along horizontal scanline)

first derivative



edges correspond to
extrema of derivative

Source: L. Lazebnik

Image derivatives

- How can we differentiate a *digital* image F[x,y]?
  - Option 1: reconstruct a continuous image, *f,* then compute the derivative
  - Option 2: take discrete derivative (finite difference)

$$\frac{\partial f}{\partial x}[x,y] \approx F[x+1,y] - F[x,y]$$

How would you implement this as a linear filter?

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

H_x

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

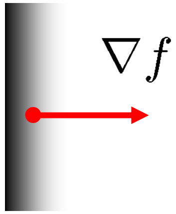
H_y

Source: S. Seitz

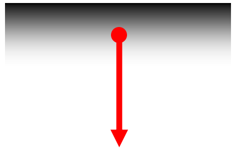
Image gradient

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

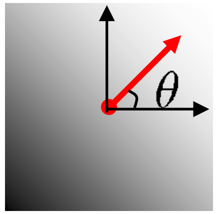
The gradient points in the direction of most rapid increase in intensity



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



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



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

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:

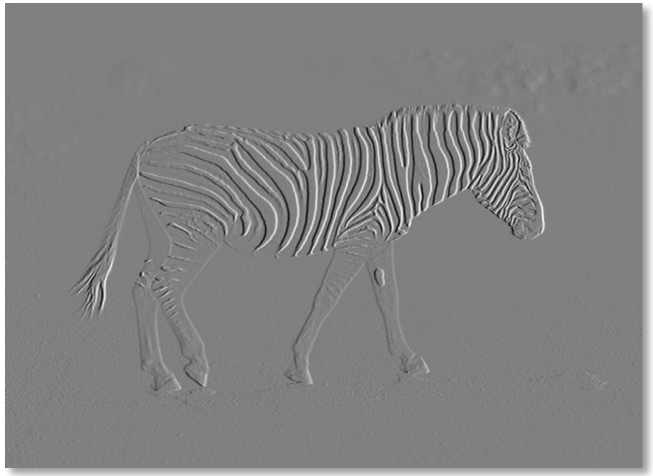
$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x}\right)$$

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

Source: Steve Seitz

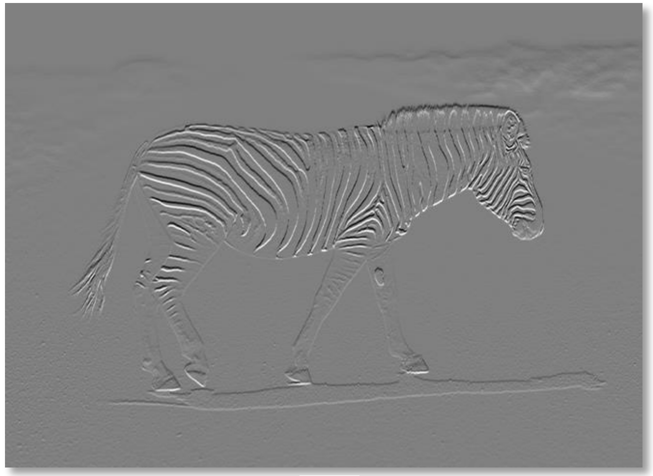
Image gradient

f



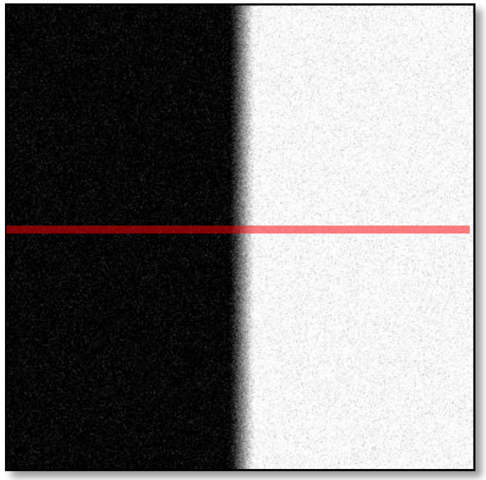
$\frac{\partial f}{\partial x}$

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

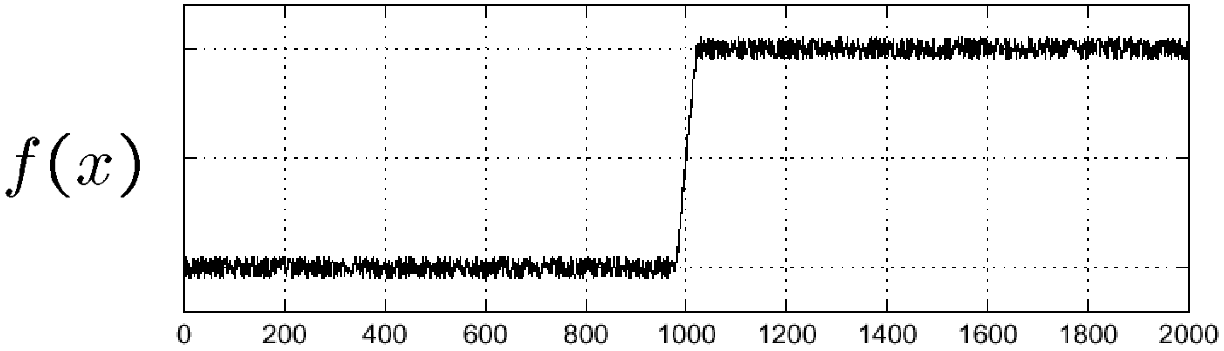


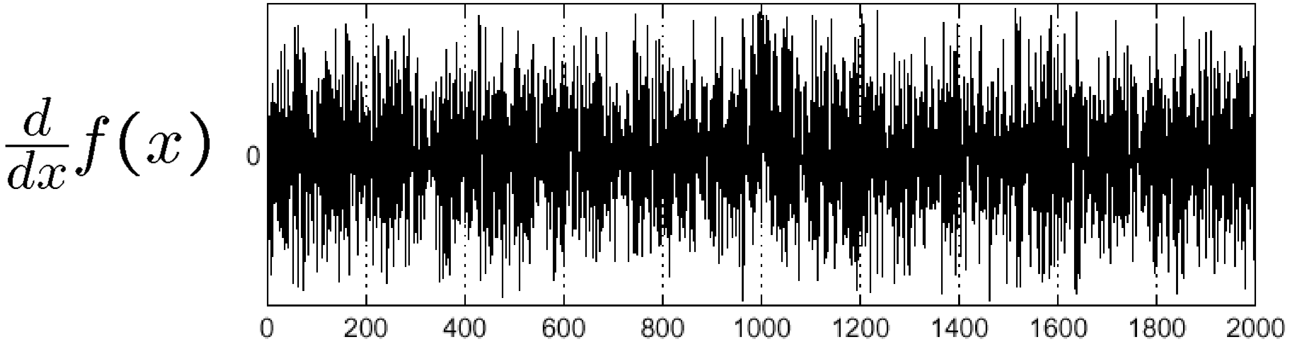
$\frac{\partial f}{\partial y}$

Effects of noise



Noisy input image



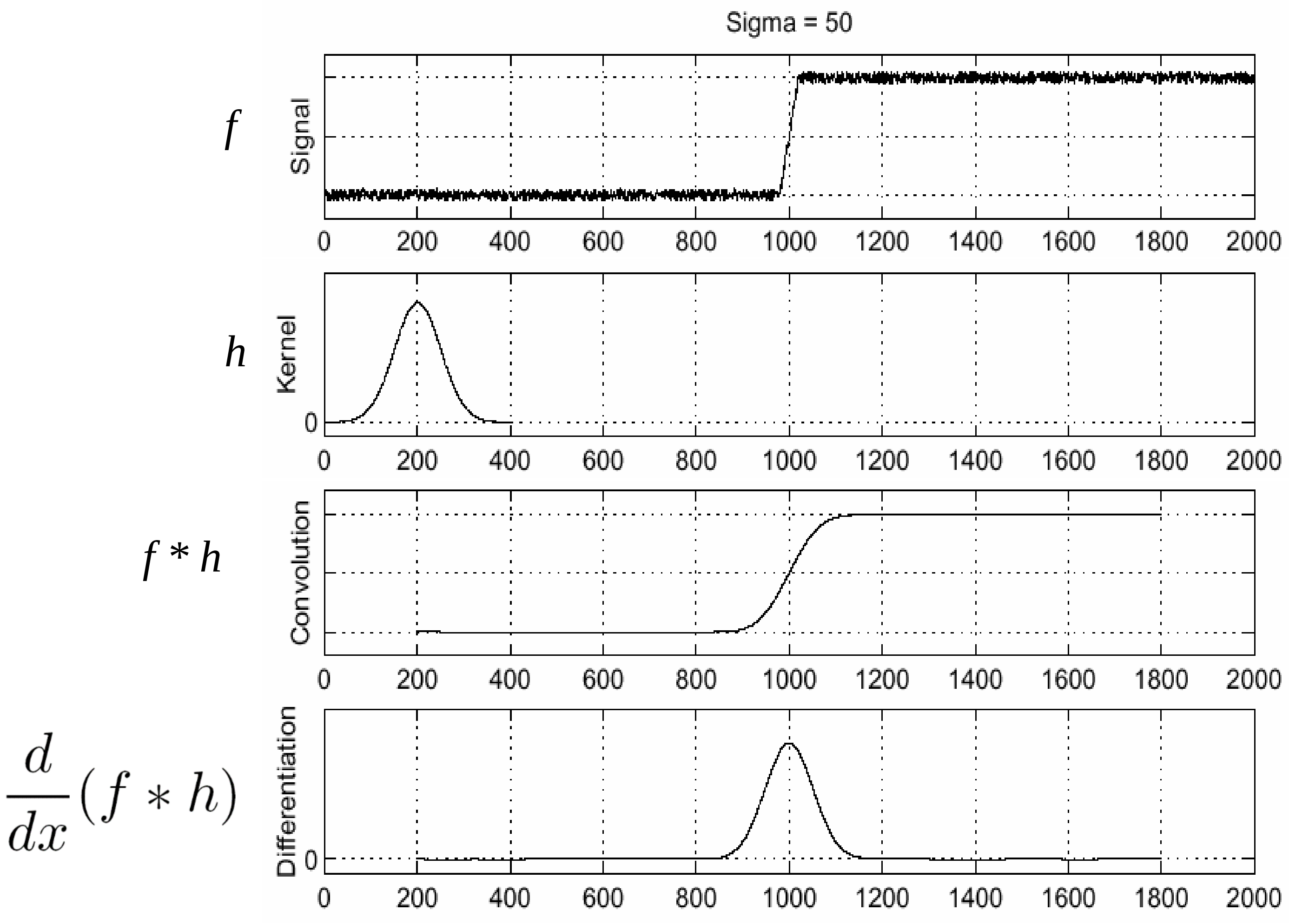


Where is the edge?

Source: S. Seitz

GEORGETOWN UNIVERSITY

Solution: smooth first



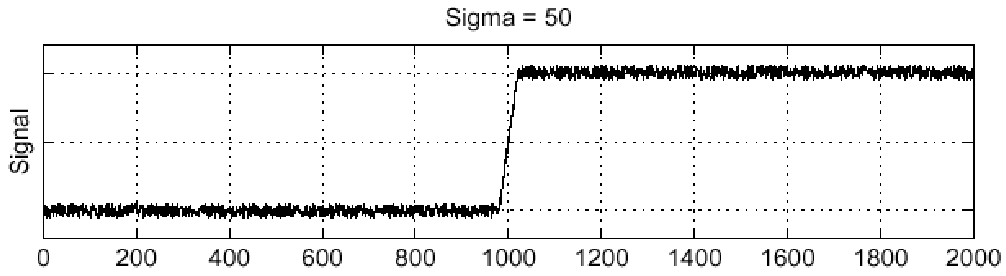
To find edges, look for peaks in $\frac{d}{dx}(f * h)$

Source: S. Seitz

Associative property of convolution

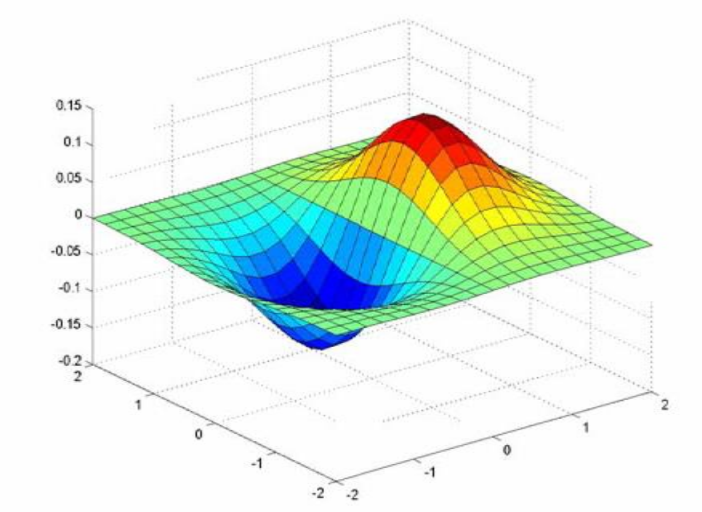
- Differentiation is convolution, and convolution is associative: $$\frac{d}{dx}(f * h) = f * \frac{d}{dx}h$$
- This saves us time:

f

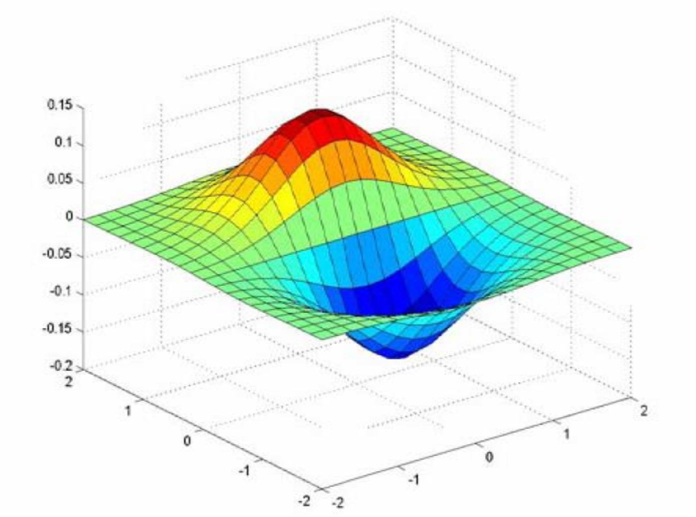


Source: S. Seitz

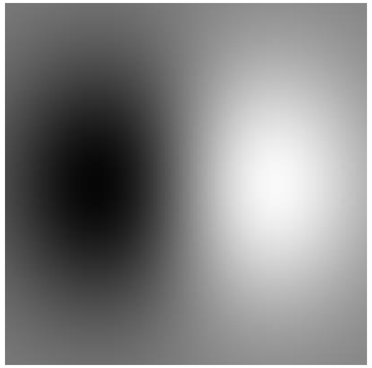
Derivative of Gaussian filter

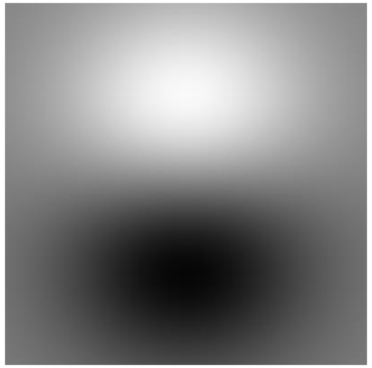


x-direction

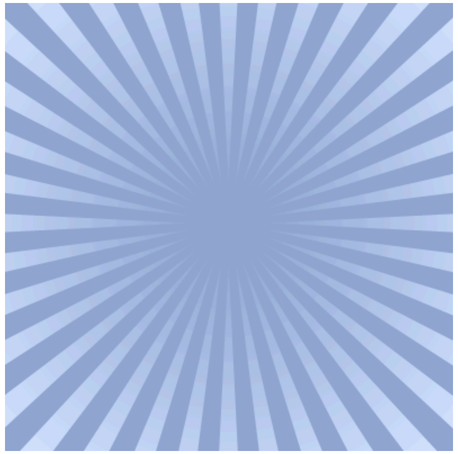


y-direction

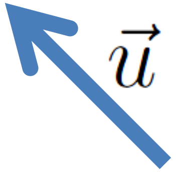




Side note: How would you compute a directional derivative?



f



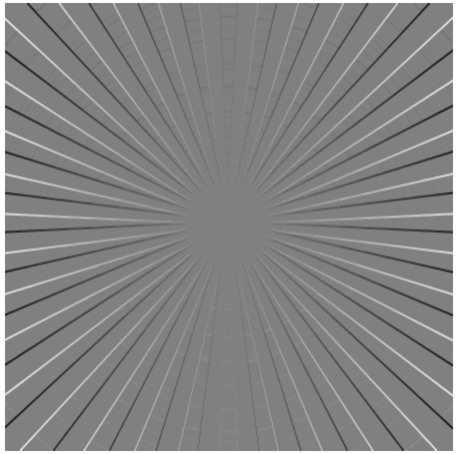
$\vec{u}$

$\nabla_{\vec{u}} f$ = ?

(From vector calculus)

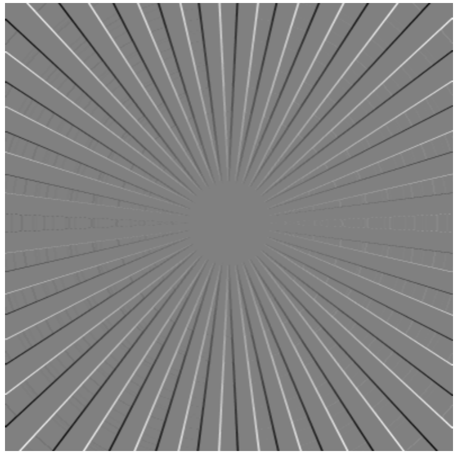
$$\nabla_{\vec{u}} f(\vec{x}) = \nabla f(\vec{x}) \cdot \vec{u}$$

Directional deriv. is a linear combination of partial derivatives



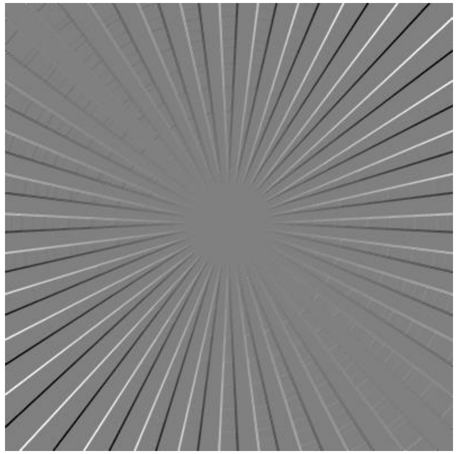
$\frac{\partial f}{\partial x} \cdot u_x$

+



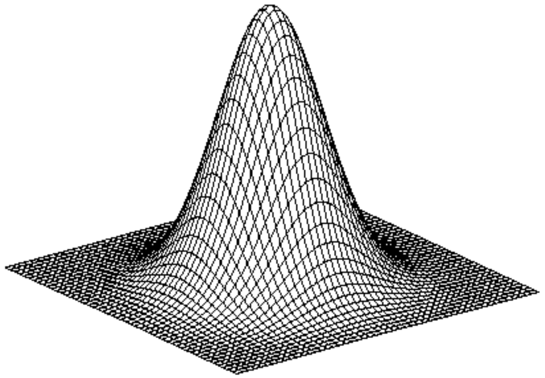
$\frac{\partial f}{\partial y} \cdot u_y$

=



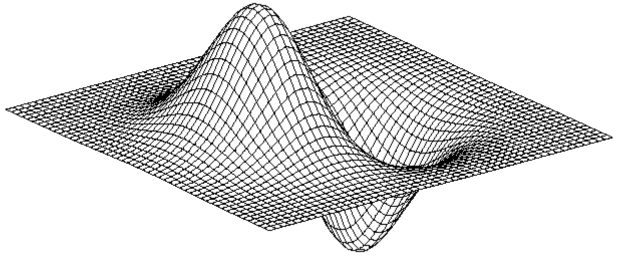
$\nabla_{\vec{u}} f$

2D edge detection filters



Gaussian

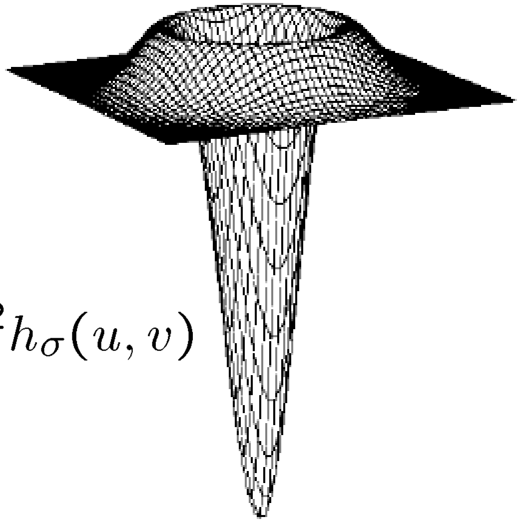
$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



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:

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

GEORGETOWN UNIVERSITY

The Sobel operator

- Common approximation of derivative of Gaussian

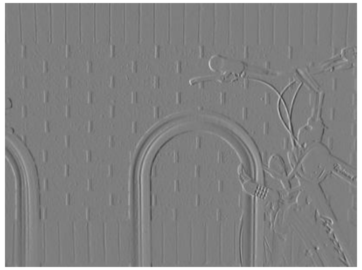
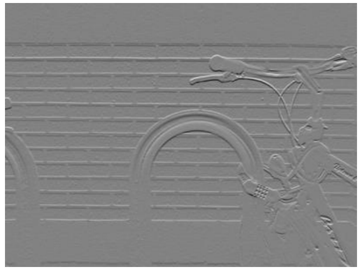
$$\frac{1}{8}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad \frac{1}{8}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$s_x \qquad\qquad\qquad s_y$$

- The standard defn. of the Sobel operator omits the 1/8 term
  - doesn't make a difference for edge detection
  - the 1/8 term **is** needed to get the right gradient value

GEORGETOWN UNIVERSITY

Sobel operator: example

Source: Wikipedia

Example

- original image (Lena)

Finding edges

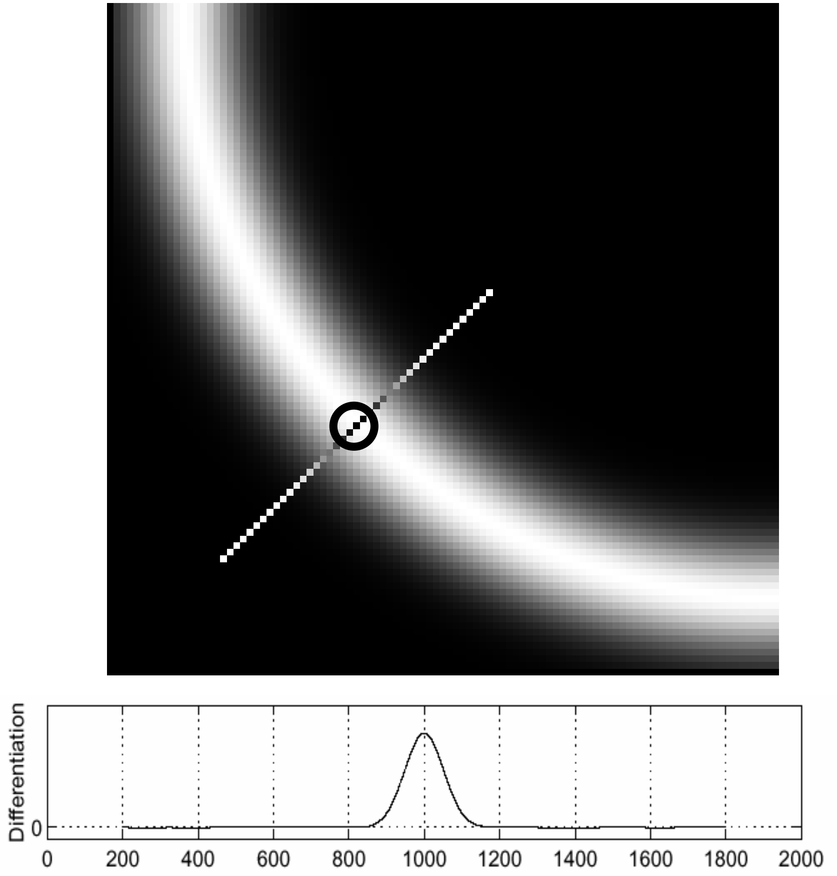
gradient magnitude

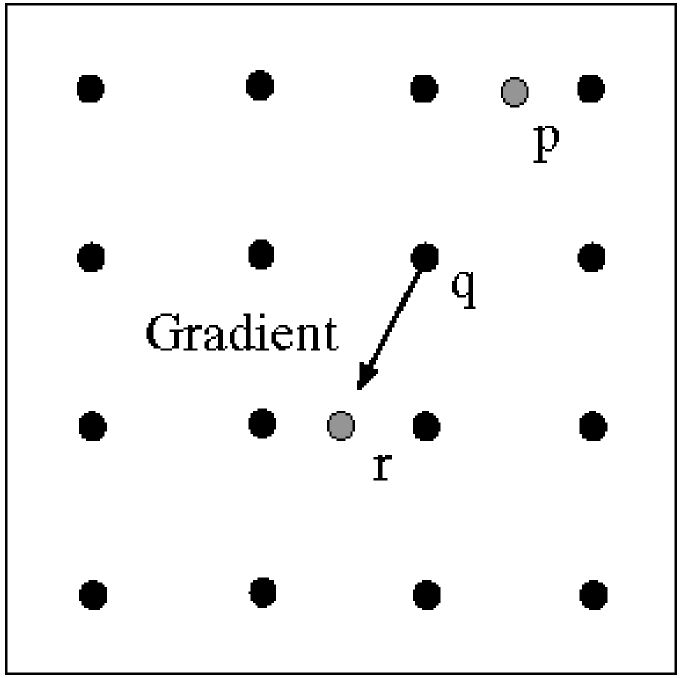
Finding edges

where is the edge?

thresholding

Non-maximum supression





- Check if pixel is local maximum along gradient direction
  - requires *interpolating* pixels p and r

Finding edges

thresholding

GEORGETOWN UNIVERSITY

Finding edges

thinning

(non-maximum suppression)

GEORGETOWN UNIVERSITY

Canny edge detector

MATLAB: `edge(image,'canny')`

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient

3. Non-maximum suppression
4. 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

Source: D. Lowe, L. Fei-Fei

Canny edge detector

- Still one of the most widely used edge detectors in computer vision

J. Canny, [***A Computational Approach To Edge Detection***], IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

- Depends on several parameters:

σ : width of the Gaussian blur

high threshold
low threshold

Canny edge detector

original | Canny with $\sigma = 1$ | Canny with $\sigma = 2$

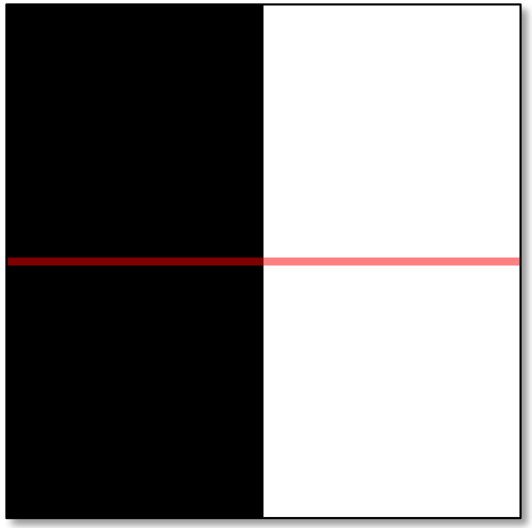
- The choice of σ depends on desired behavior
  - large σ detects "large-scale" edges
  - small σ detects fine edges

Source: S. Seitz

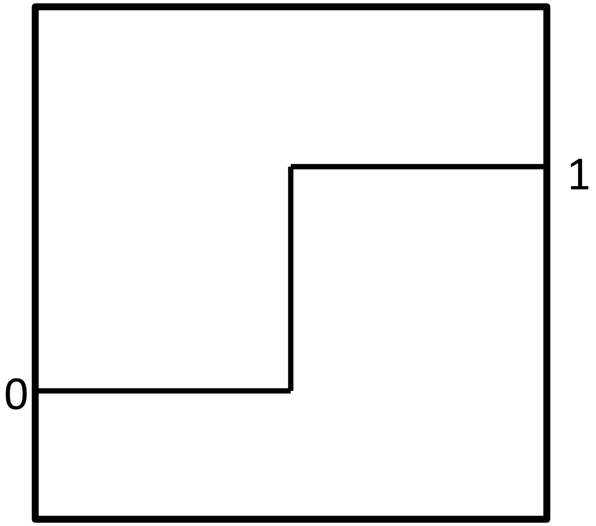
GEORGETOWN UNIVERSITY

Images as vectors

- Very important idea!



2D image



Scanline (1D signal)

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

Vector

(A 2D, *n* x *m* image can be represented by a vector of length *nm* formed by concatenating the rows)

Multiplying row and column vectors

$$\begin{bmatrix} 0 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = ?$$

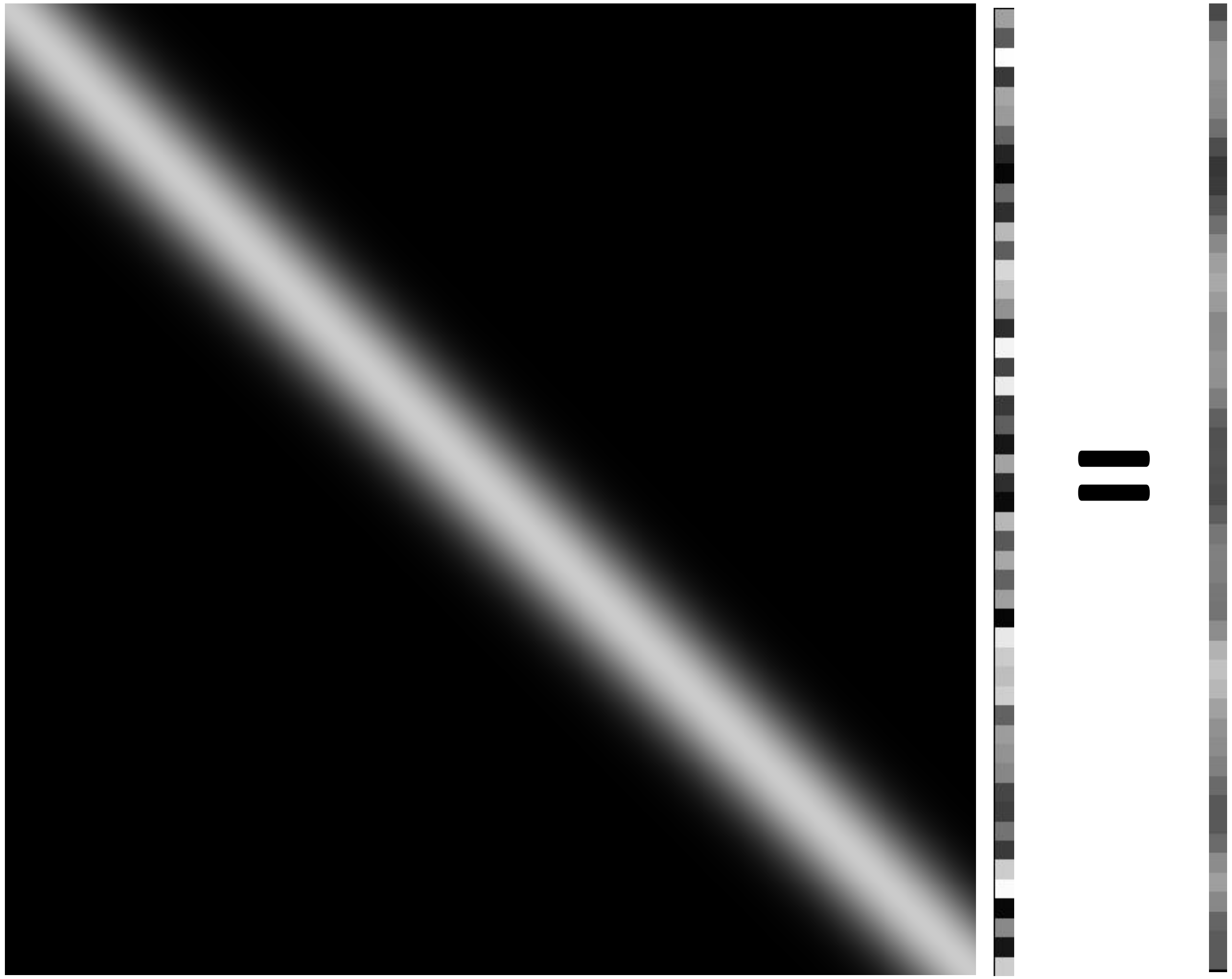
GEORGETOWN UNIVERSITY

Filtering as matrix multiplication

$$
\begin{bmatrix}
0.2 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.2 & 0.2
\end{bmatrix}
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}
$$

What kind of filter is this?

Filtering as matrix multiplication



Scale Space

- Recall: Edges can be identified by zero-crossings of first derivative.
- Observe derivative filter may be “fixed” (in size).

1	-1	

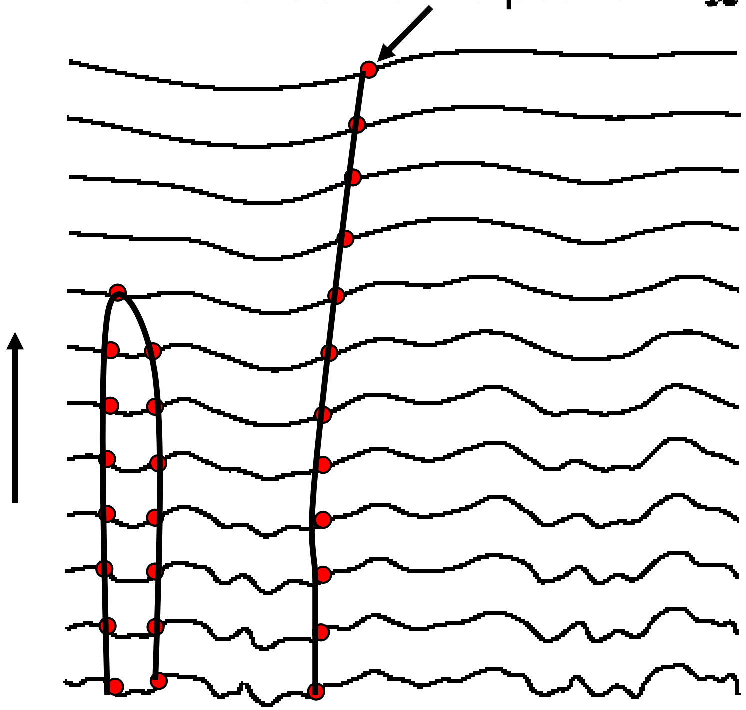
- But edges may exist at various levels of fineness.

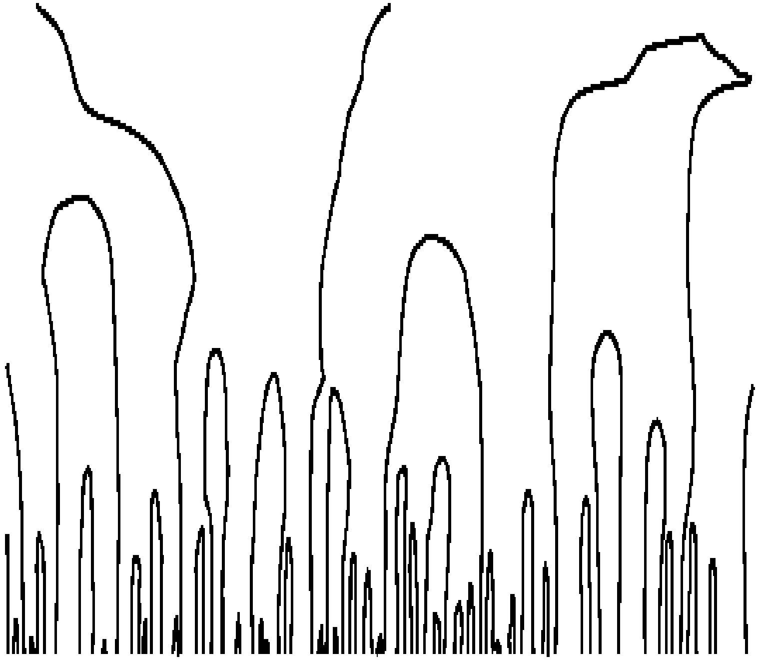
GEORGETOWN UNIVERSITY

Scale space *(Witkin 83)*

first derivative peaks $F_{xx} = 0, F_{xxx} \neq 0$

larger σ





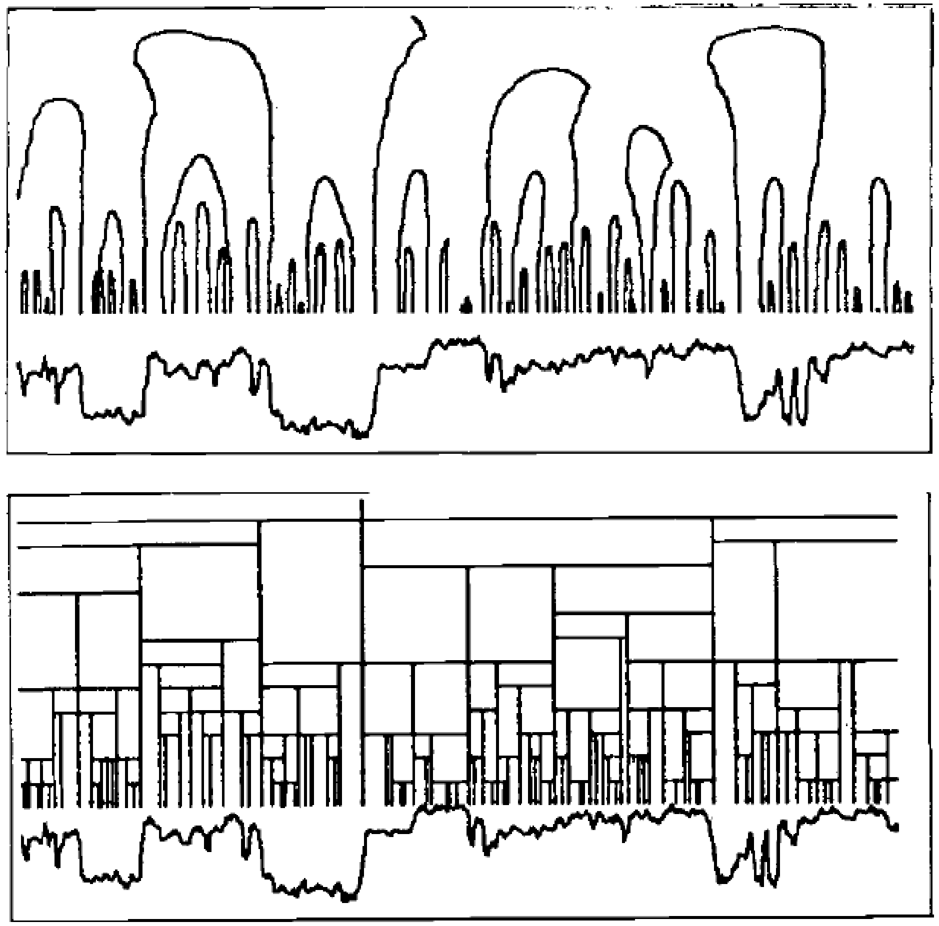
Gaussian filtered signal

- Properties of scale space (w/ Gaussian smoothing)
  - edge position may shift with increasing scale (σ)
  - two edges may merge with increasing scale
  - an edge may ***not*** split into two with increasing scale

Observe: Each "zero-contour" (set of (x,σ) pairs) characterizes an edge

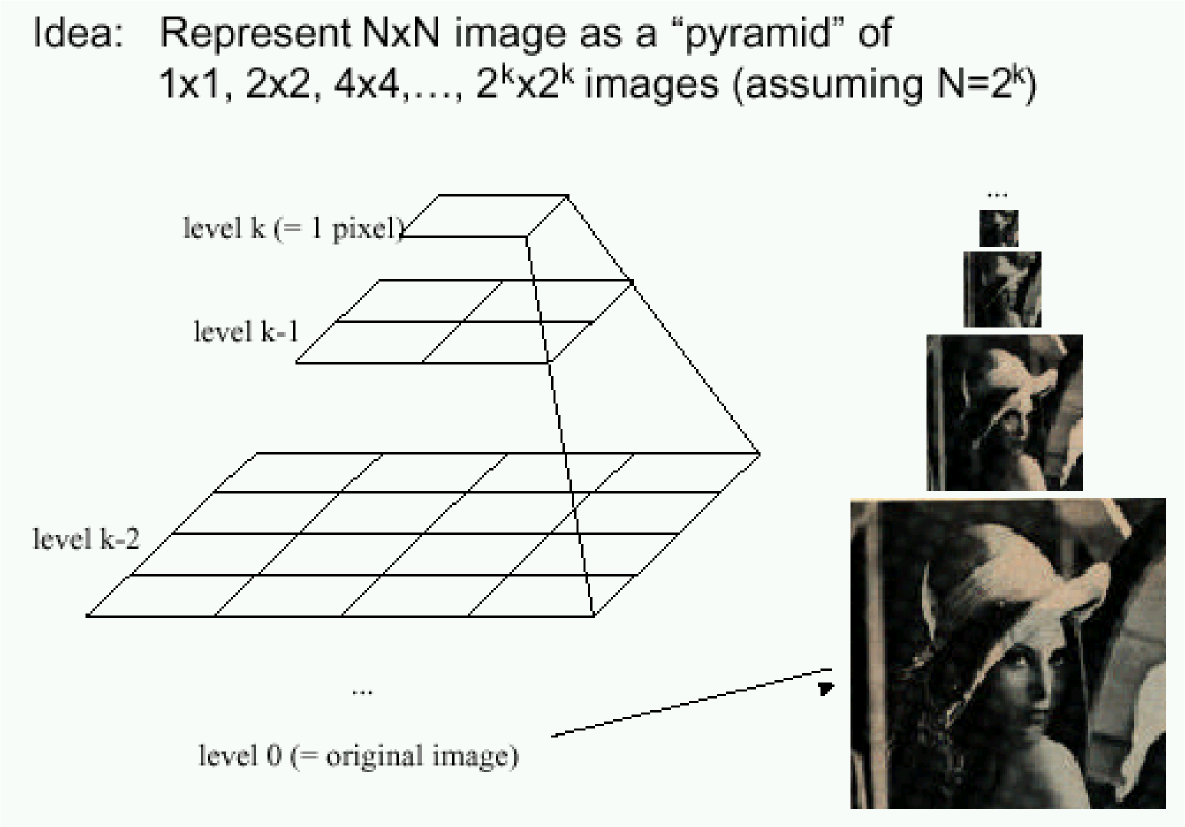
Interval Tree Representation

- Each entity or event may have “a beginning and an end”
  - Interval bounded by zero-crossings
  - The scale of the event can be determined by the zero-contour representation.
  - Zero-contours can be reduced to a hierarchical representation or a simple partition of the contour space.
  - Tree representation: each interval is simply a node in a tree.



- Uses:
  - Image characterization
  - Course to fine localization

Some times we want many resolutions

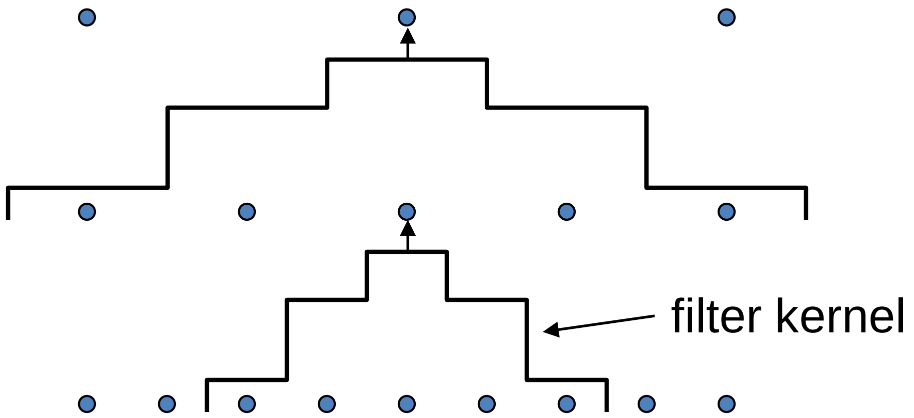


Known as a **Gaussian Pyramid** [Burt and Adelson, 1983]

- A precursor to *wavelet transform*

Gaussian Pyramids have all sorts of applications in computer vision

Gaussian pyramid construction



Repeat

- Filter
- Subsample

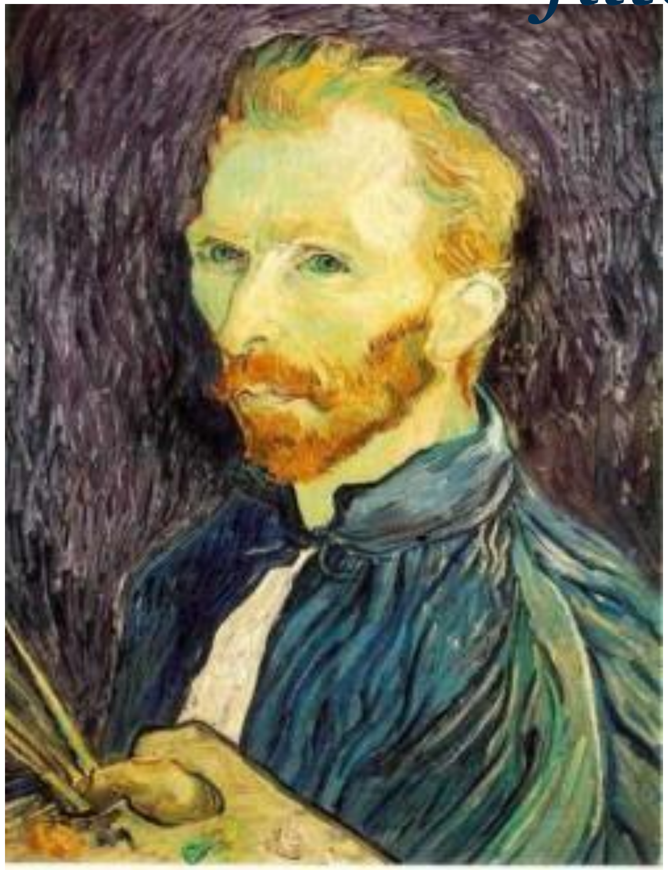
Until minimum resolution reached

- can specify desired number of levels (e.g., 3-level pyramid)
- Can find edges (or features) at **fine** and **gross** scales.
- Analyze an image at multiple scales
  - Referred to as **Scale Space** analysis

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

GEORGETOWN UNIVERSITY

Subsampling with Gaussian pre-filtering



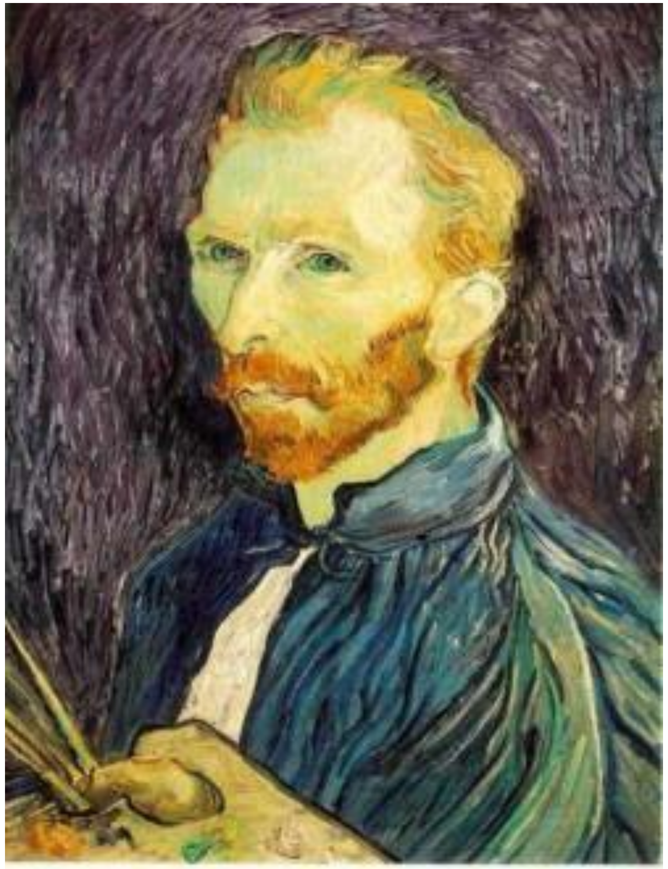
Gaussian 1/2

G 1/4

G 1/8

Filter the image, *then* subsample

Subsampling with Gaussian pre-filtering



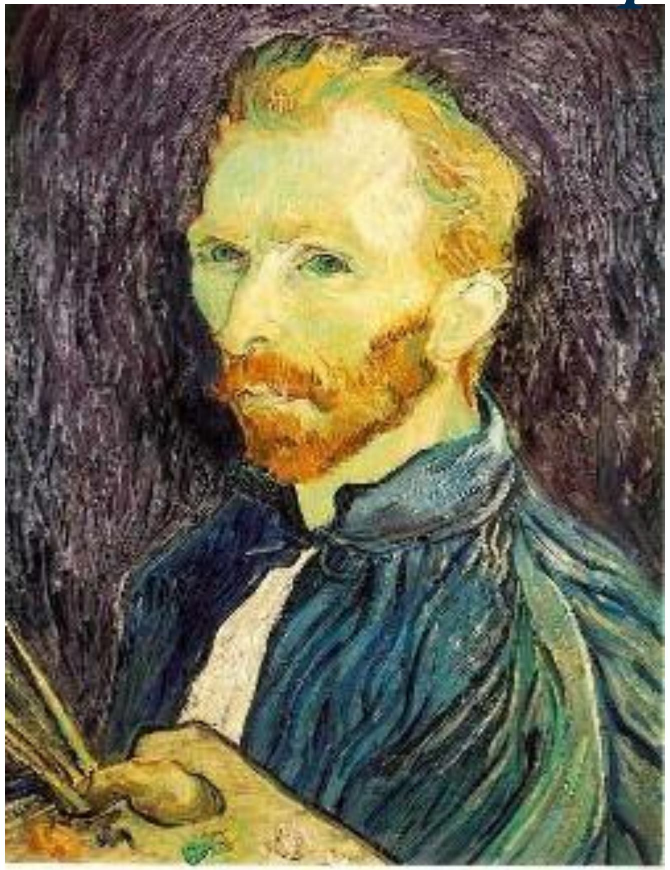
Gaussian 1/2

G 1/4

G 1/8

Filter the image, *then* subsample

Subsampling without pre-filtering



1/2

1/4 (2x zoom)

1/8 (4x zoom)

Summary