Edge detection

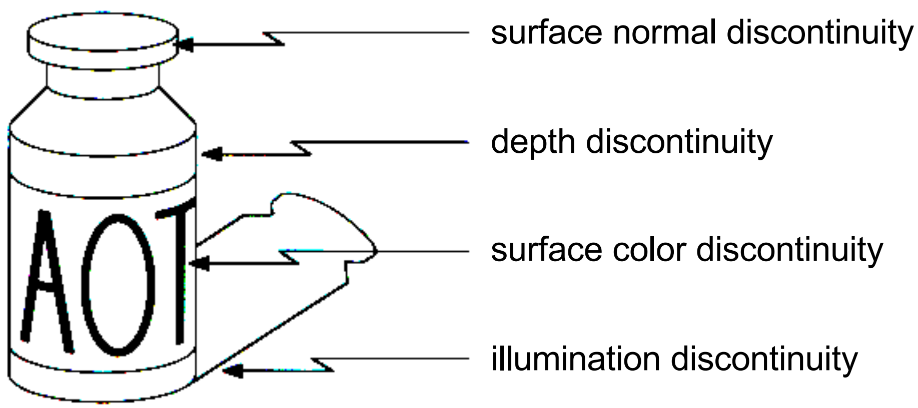
Winter in Kraków photographed by Marcin Ryczek

Overview

- Motivating edge detection
- Image gradients
- Derivative of Gaussian filters
- Canny edge detector
- Role of edge detection in image understanding

Edge detection

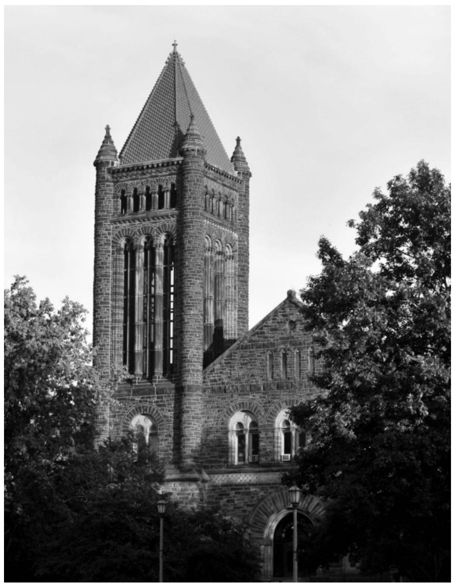
- **Goal:** Identify sudden changes (discontinuities) in an image
- Intuitively, edges carry most of the semantic and shape information from the image



Sources: D. Lowe and S. Seitz

Edge detection

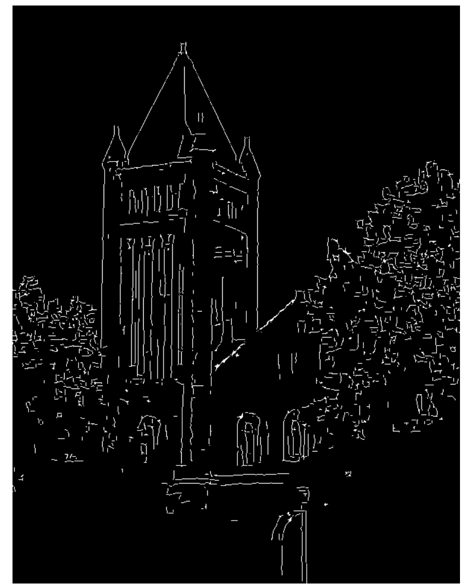
Input photo



Ideal: artist's line drawing

Image source

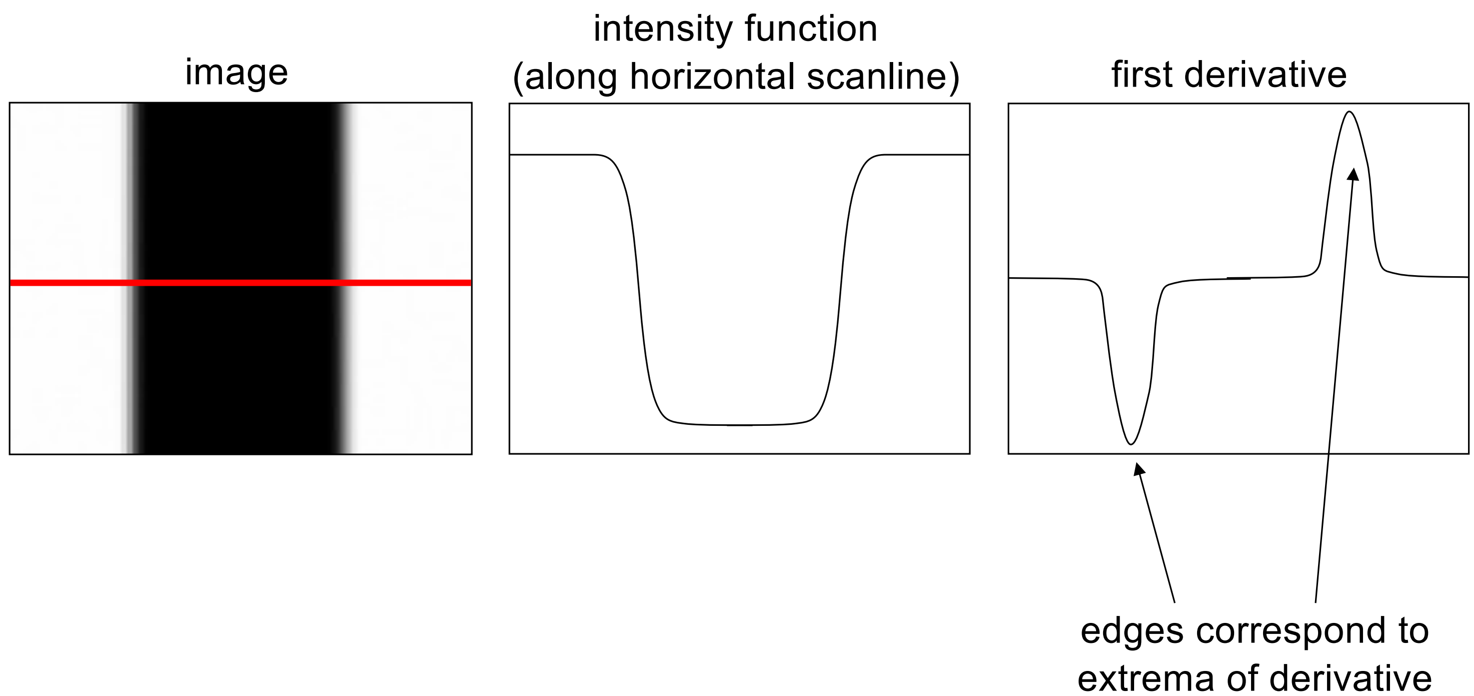
Reality



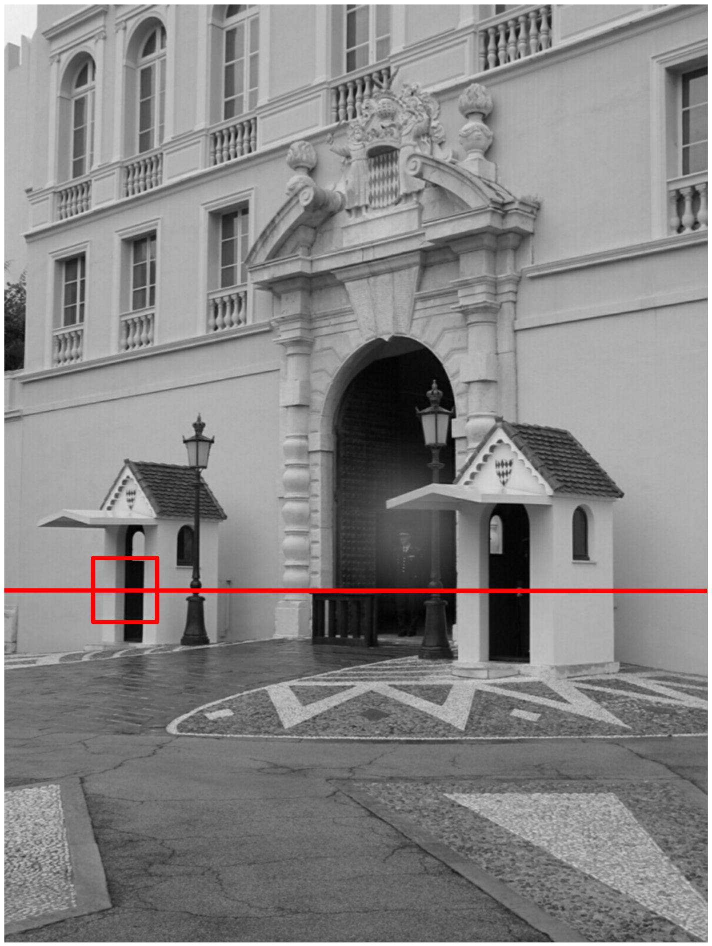
Source: S. Gupta

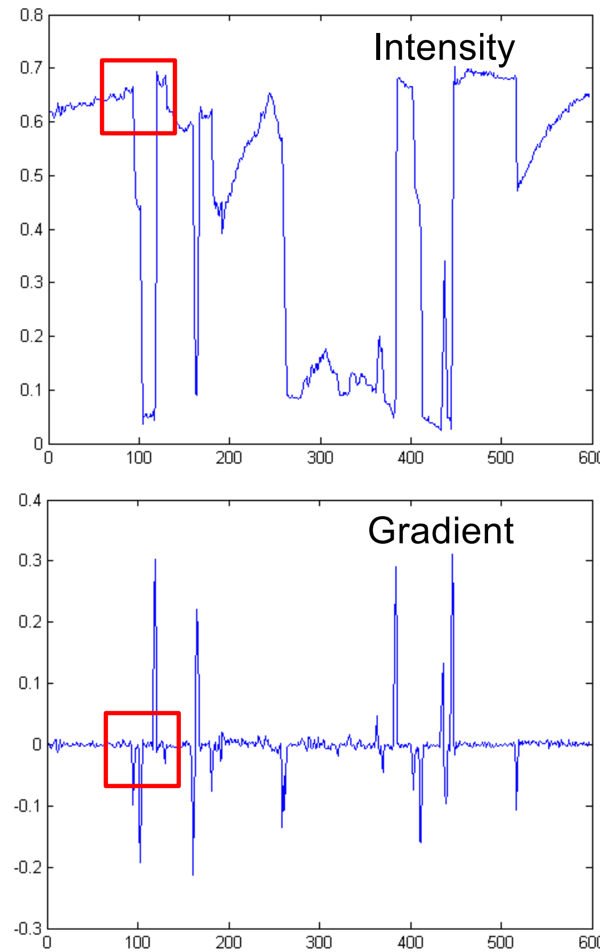
Characterizing edges

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



A realistic example





Source: D. Hoiem

Partial derivatives of an image

- For 2D function $f(x, y)$, the partial derivative w.r.t. x is

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

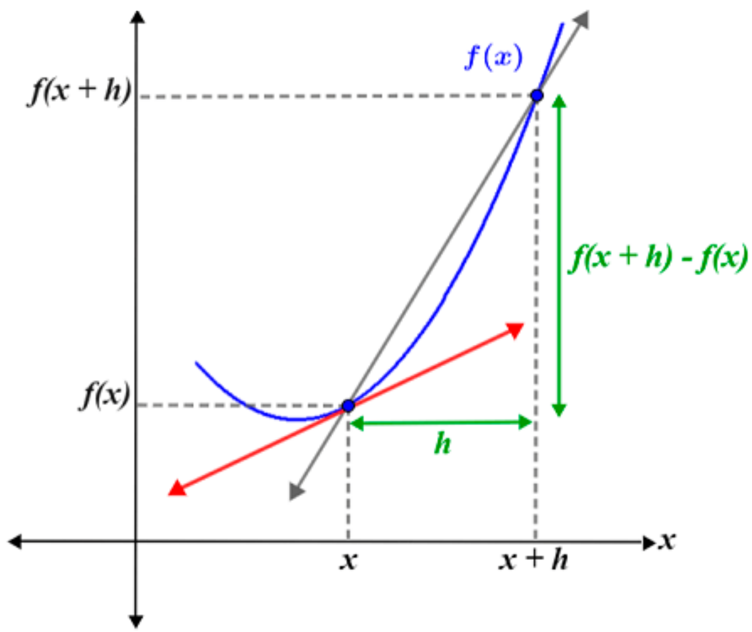


Image source

Partial derivatives of an image

- For 2D function $f(x, y)$, the partial derivative w.r.t. x is

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

- For discrete data, we can approximate using finite differences:

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

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

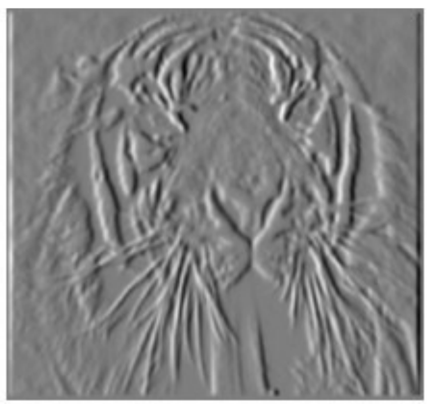
Source: K. Grauman

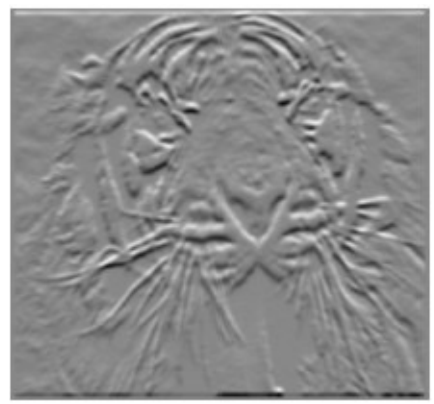
Partial derivatives of an image

$f(x, y)$

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

-1	1





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

-1
1

Finite difference filters

Other approximations of derivative filters exist:

- Prewitt

M_x

-1	0	1
-1	0	1
-1	0	1

M_y

1	1	1
0	0	0
-1	-1	-1

- Sobel

M_x

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

M_y

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

- Roberts

M_x

0	1
-1	0

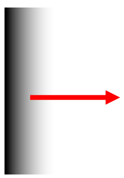
M_y

1	0
0	-1

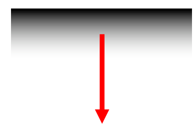
Source: K. Grauman

Image gradient

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

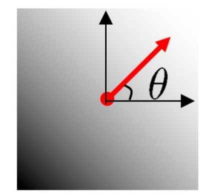


$\nabla f = \left(\frac{\partial f}{\partial x}, 0\right)$



$\nabla f = \left(0, \frac{\partial f}{\partial y}\right)$

$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$

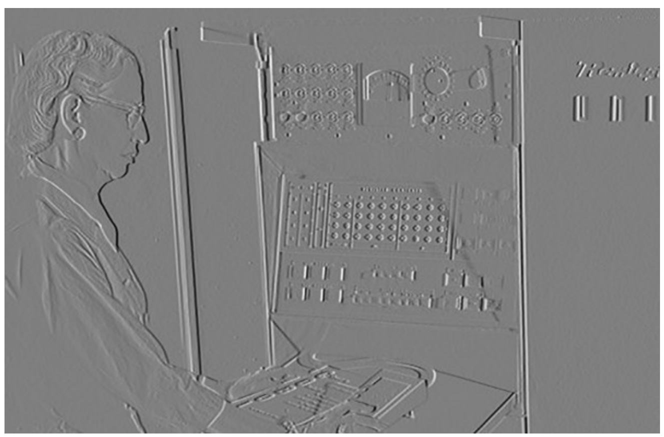


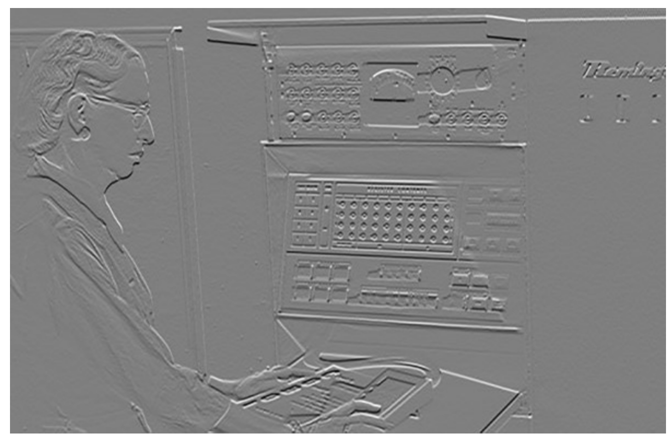
- The gradient points in the direction of the most rapid *increase* in intensity
- Gradient orientation is given by $\theta = \tan^{-1} \frac{\partial f / \partial y}{\partial f / \partial x}$
- Gradient magnitude is given by $\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$

Image gradient: Example

$f(x, y)$

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





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

Source: D. Fouhey

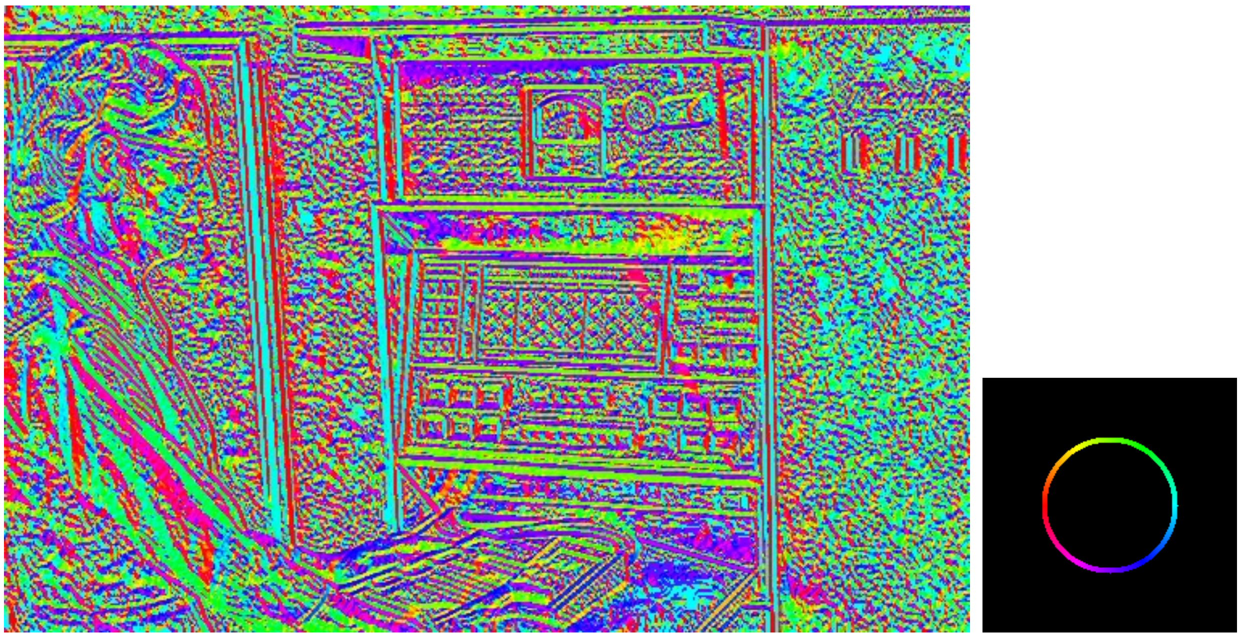
Image gradient: Example

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

Source: D. Fouhey

Image gradient: Example

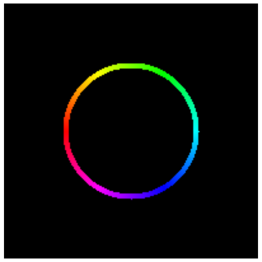
Orientation: $\tan^{-1} \frac{\partial f / \partial y}{\partial f / \partial x}$



Source: D. Fouhey

Image gradient: Example

Orientation: $\tan^{-1} \frac{\partial f / \partial y}{\partial f / \partial x}$



Make lightness equal to gradient magnitude

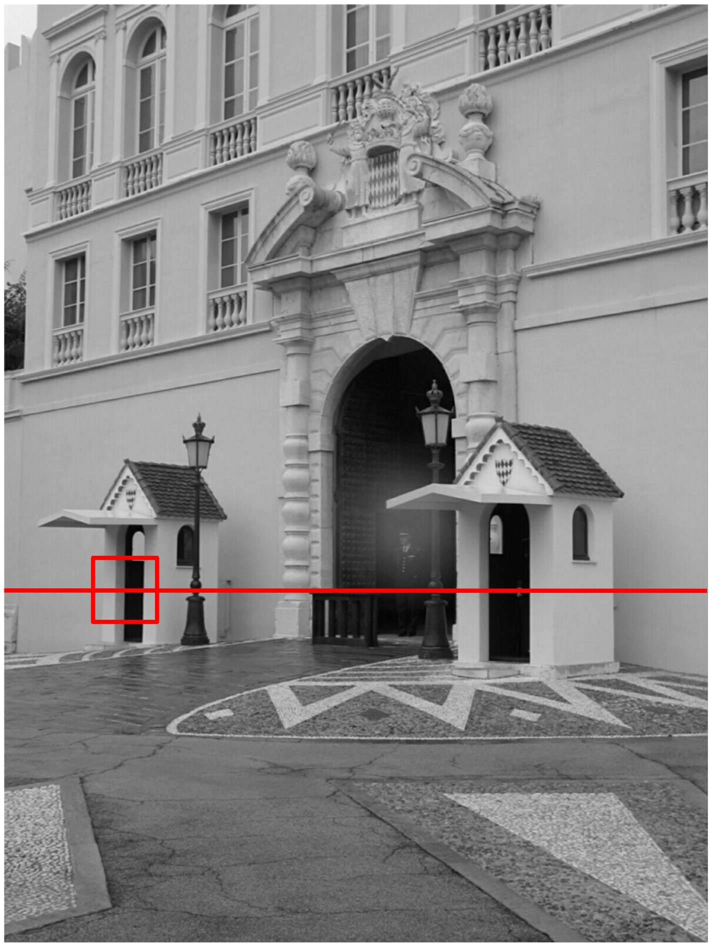
Source: D. Fouhey

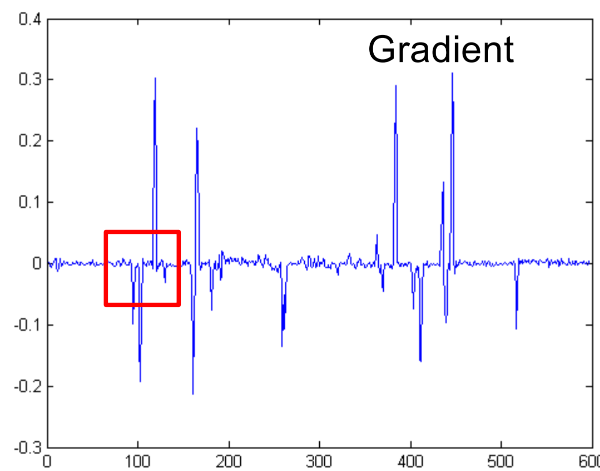
Aside: Gradient-domain image editing

- Goal: solve for pixel values in the target region to match gradients of the source region while keeping background pixels the same

P. Perez et al. Poisson Image Editing. SIGGRAPH 2003

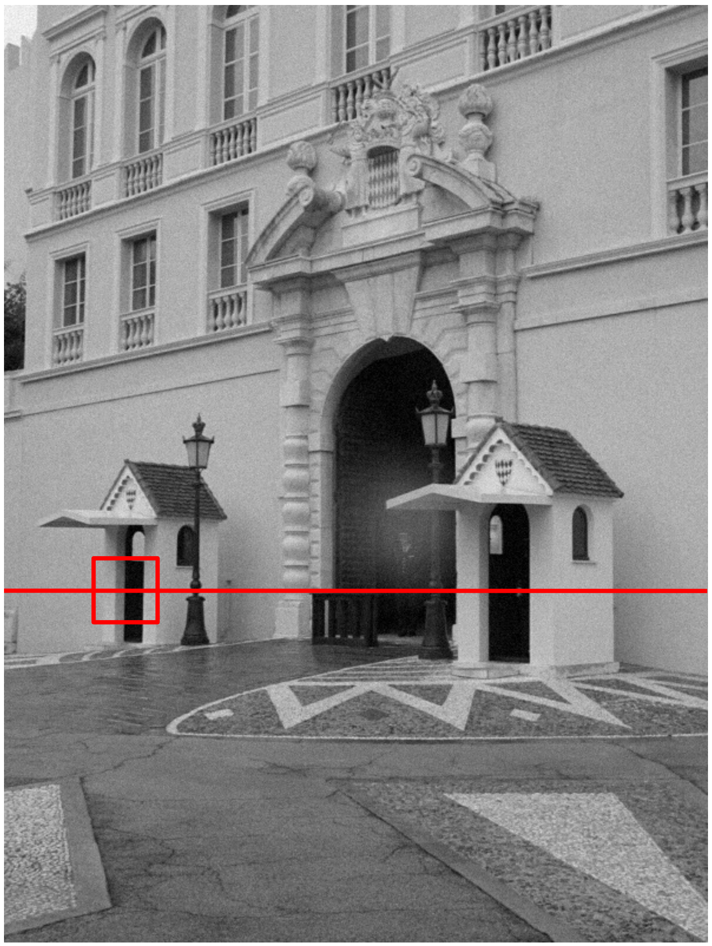
Derivatives and noise



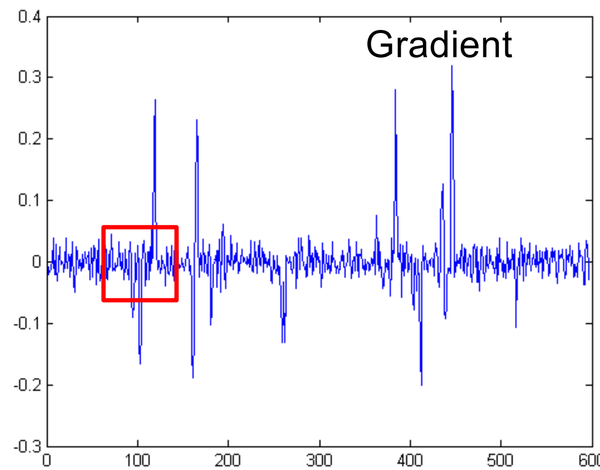


Source: D. Hoiem

Derivatives and noise



- Let's add a little Gaussian noise



Source: D. Hoiem

Derivatives and noise

- Suppose pixels of the "true" image $f_{i,j}$ are corrupted by additive Gaussian noise $\epsilon_{i,j} \sim N(0, \sigma^2)$
- What happens when we compute pixel differences?

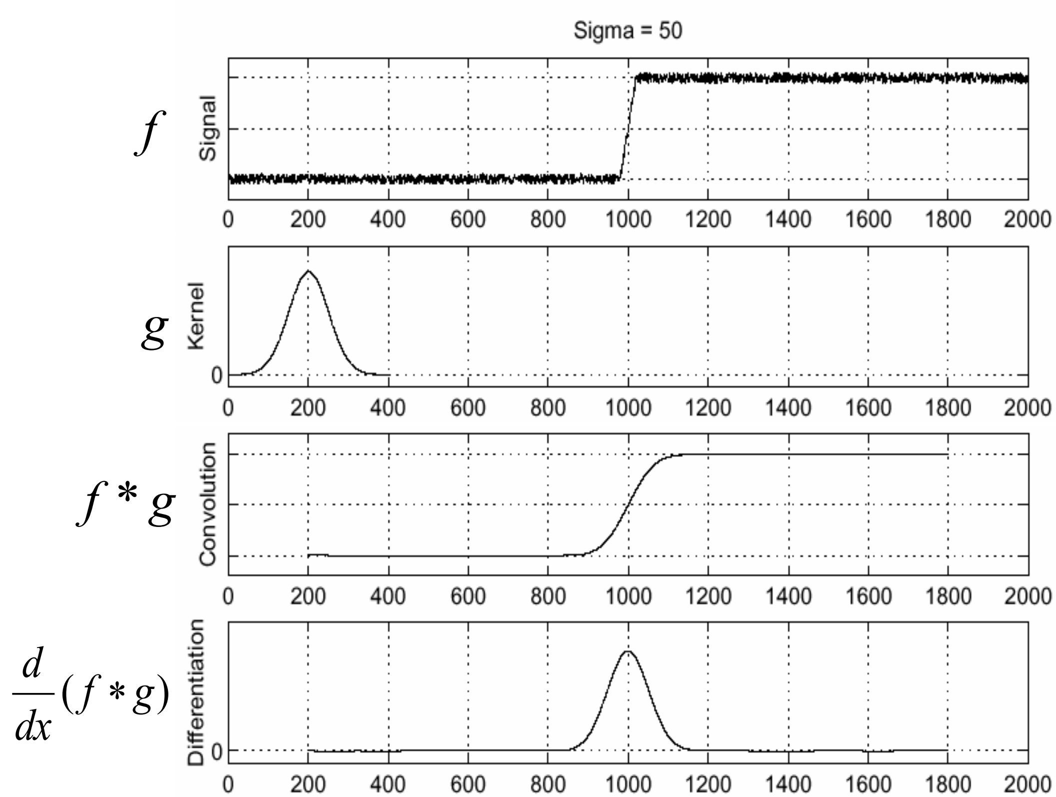
$$D_{i,j} = (f_{i,j+1} + \epsilon_{i,j+1}) - (f_{i,j} + \epsilon_{i,j})$$

$$= \underbrace{(f_{i,j+1} - f_{i,j})}_{\text{True difference}} + \underbrace{(\epsilon_{i,j+1} - \epsilon_{i,j})}_{\text{Difference of two zero-mean Gaussian random variables (same as sum)}}$$

$\epsilon_{i,j+1} - \epsilon_{i,j} \sim N(0, 2\sigma^2) \rightarrow$ Variance doubles!

Adapted from [D. Fouhey and J. Johnson]

Finding noisy edges: Smooth first



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

Source: S. Seitz

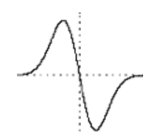
Finding noisy edges: Smooth first

- Let d denote the derivative filter, e.g., $[-1\ 0\ 1]$

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

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

Derivative of Gaussian filter

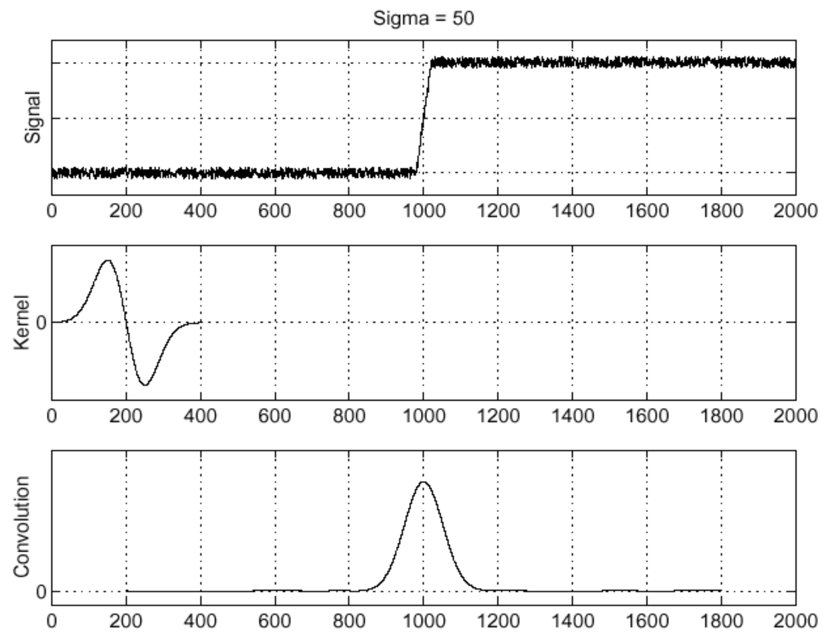


Filtering with derivative of Gaussian

f

$$\frac{d}{dx} g$$

$$f * \frac{d}{dx} g$$

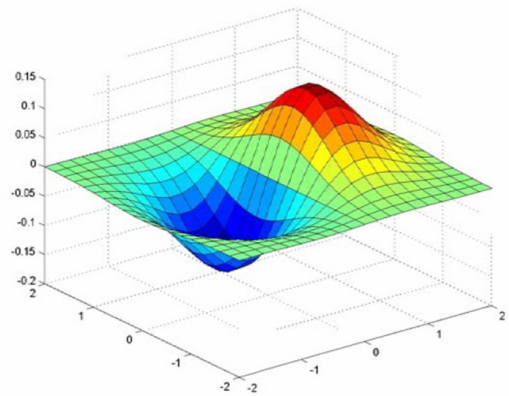


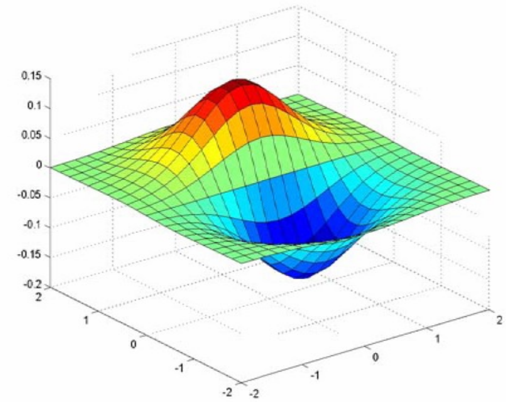
Source: S. Seitz

Overview

- Motivating edge detection
- Image gradients
- Derivative of Gaussian filters

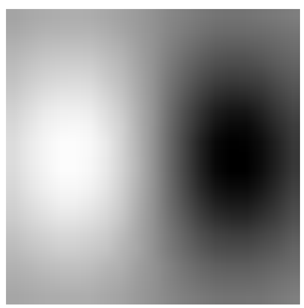
2D Derivative of Gaussian filters

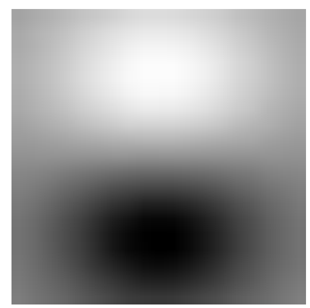




x-direction

y-direction





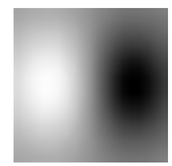
2D Derivative of Gaussian filters

- (Unnormalized) 2D Gaussian:

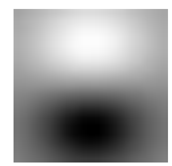
$$g(x,y) \propto \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) = \exp\left(-\frac{x^2}{2\sigma^2}\right)\exp\left(-\frac{y^2}{2\sigma^2}\right)$$

- (Unnormalized) Gaussian derivatives:

$$\frac{\partial g}{\partial x} \propto -x\exp\left(-\frac{x^2}{2\sigma^2}\right)\exp\left(-\frac{y^2}{2\sigma^2}\right)$$



$$\frac{\partial g}{\partial y} \propto -y\exp\left(-\frac{y^2}{2\sigma^2}\right)\exp\left(-\frac{x^2}{2\sigma^2}\right)$$



- These are products of a 1D Gaussian in one direction and 1D derivative of Gaussian in the other direction!

Derivative of Gaussian: Frequency domain

- If $F(u)$ is the Fourier transform of $f(t)$, then

$$\mathcal{F}\left\{\frac{d}{dt}f(t)\right\} = i2\pi u F(u)$$

- For a 1D Gaussian with $\sigma = 1$, $F(u) = g(u)$, so we have

$$\mathcal{F}\left\{\frac{d}{dt}g(t)\right\} = i2\pi u g(u) = -i2\pi \frac{d}{du}g(u)$$

- This is minus the derivative of Gaussian on the imaginary axis:

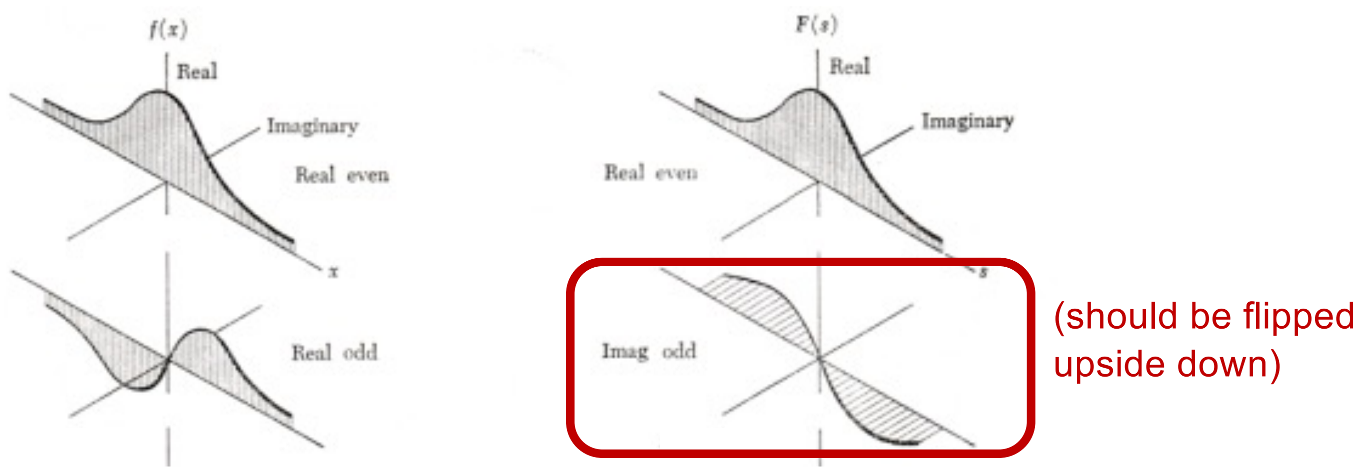


Image source

Derivative of Gaussian: Frequency domain

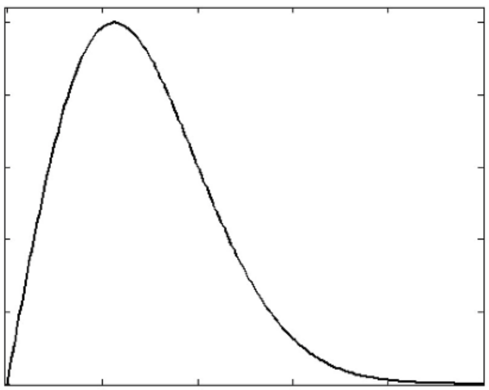
- If $F(u)$ is the Fourier transform of $f(t)$, then

$$\mathcal{F}\left\{\frac{d}{dt}f(t)\right\} = i2\pi u F(u)$$

- For a 1D Gaussian with $\sigma = 1$, $F(u) = g(u)$, so we have

$$\mathcal{F}\left\{\frac{d}{dt}g(t)\right\} = i2\pi u g(u) = -i2\pi\frac{d}{du}g(u)$$

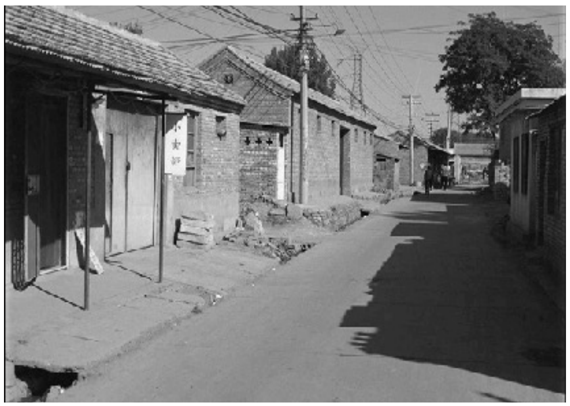
- The magnitude spectrum looks like this:

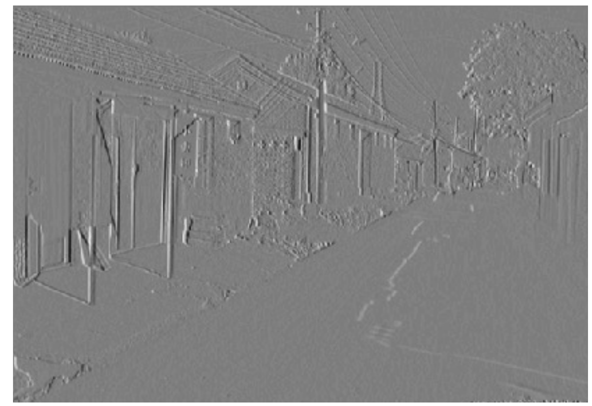
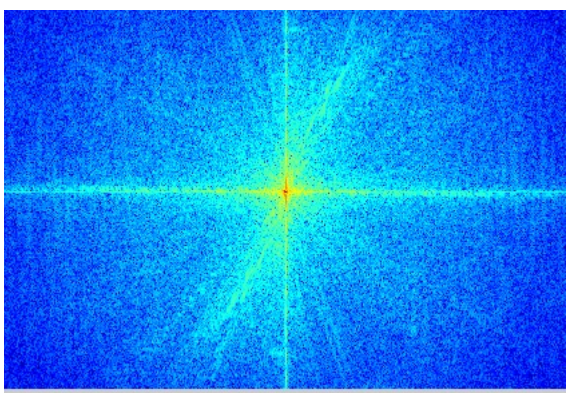
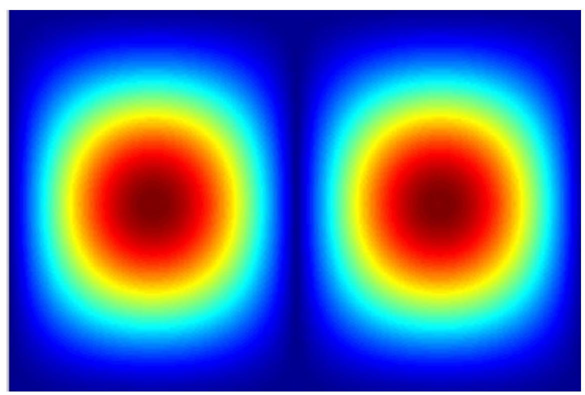
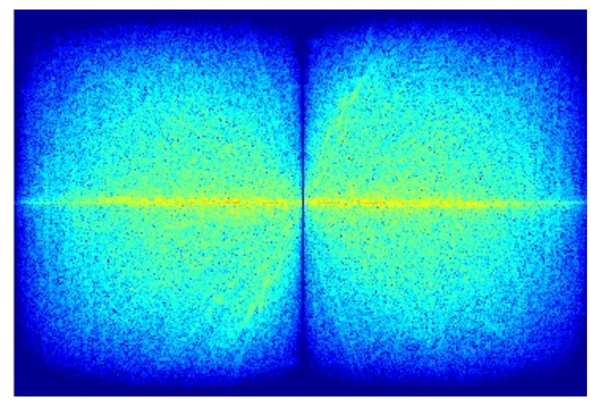


Functions as a *band-pass filter*!

Edge detection in frequency domain

Image | Filter | Filtered image

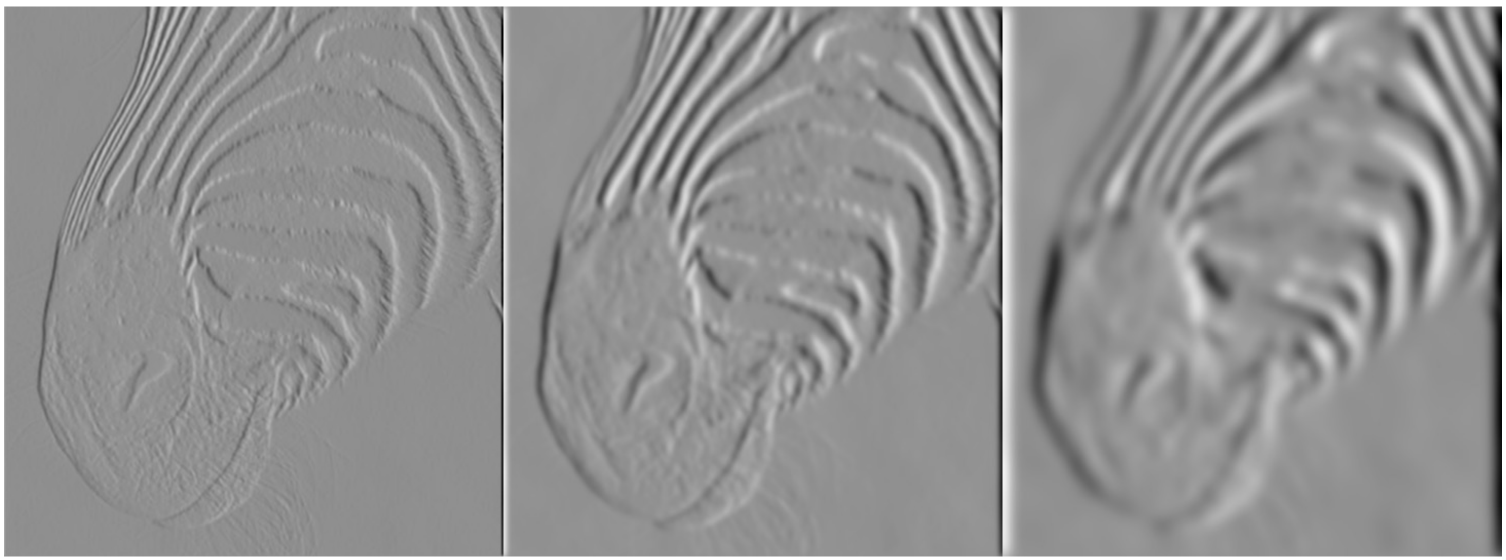


Source: D. Hoiem

Derivative of Gaussian: Scale

- Using Gaussian derivatives with different values of σ finds structures at different scales or frequencies

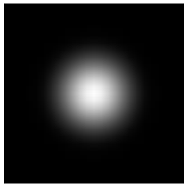


$\sigma = 1$ $\sigma = 3$ $\sigma = 7$

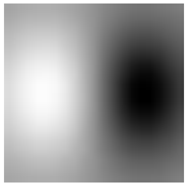
Source: D. Forsyth

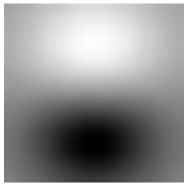
Summing up: Types of filters

- Smoothing filters
  - Gaussian: remove “high-frequency” components; “low-pass” filter
  - Can the values of a smoothing filter be negative?
  - What should the values sum to?
    - **One:** constant regions are not affected by the filter



- Derivative filters
  - Derivatives of Gaussian: compute smoothed differences; “band-pass” filters
  - Can the values of a derivative filter be negative?
  - What should the values sum to?
    - **Zero:** no response in constant regions

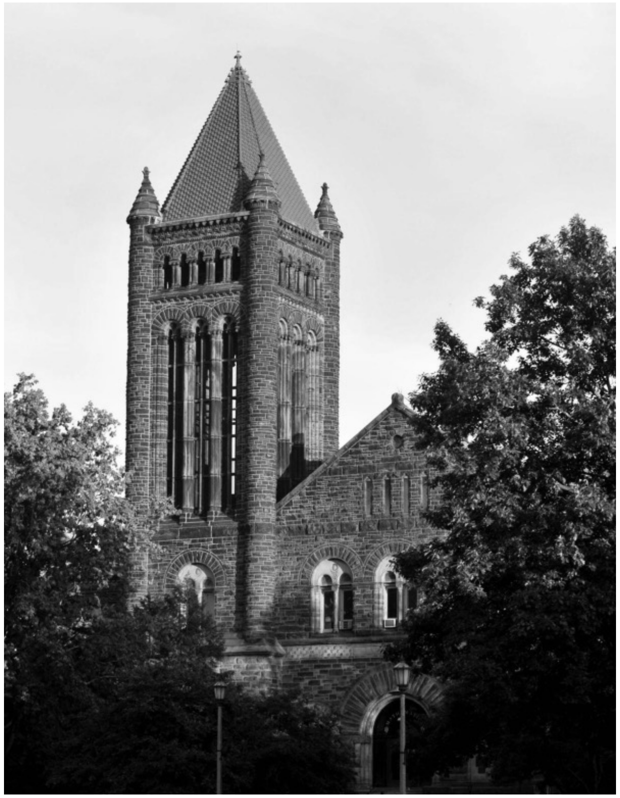




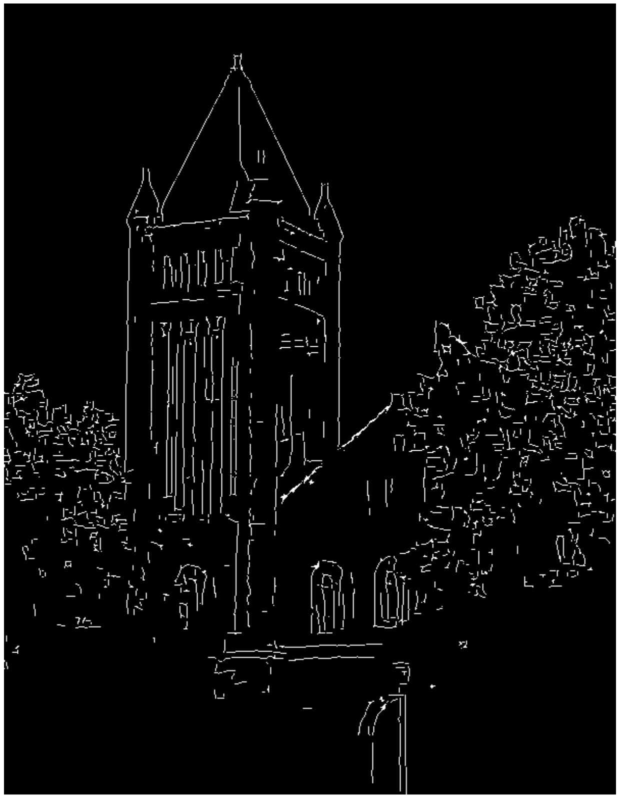
Edge detection: Overview

- Motivating edge detection
- Image gradients
- Derivative of Gaussian filters
- Canny edge detector

Building an edge detector



original image

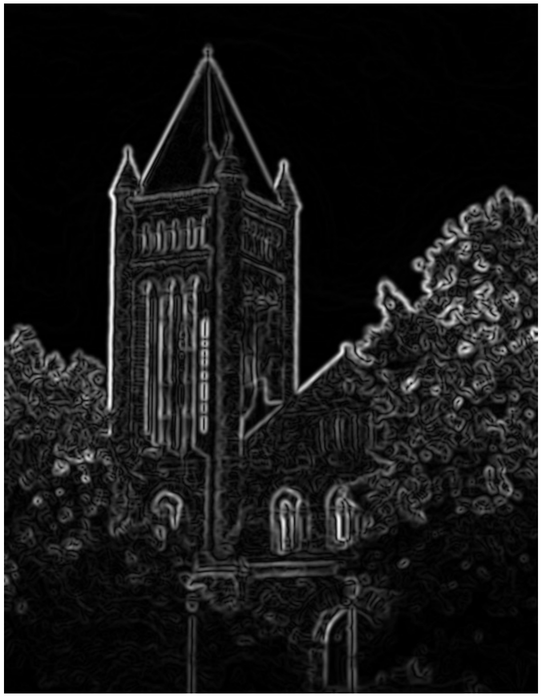


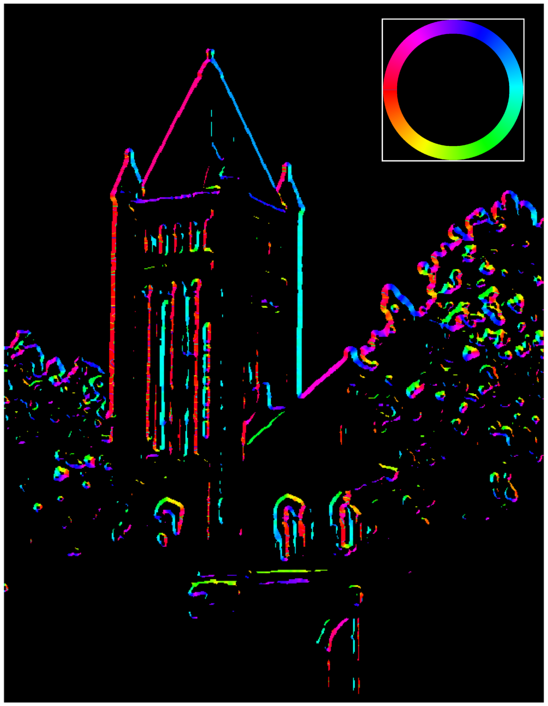
final output

J. Canny. A Computational Approach To Edge Detection. PAMI 1986

Building an edge detector

1. Compute x and y derivative images
2. Find magnitude and orientation of the gradient

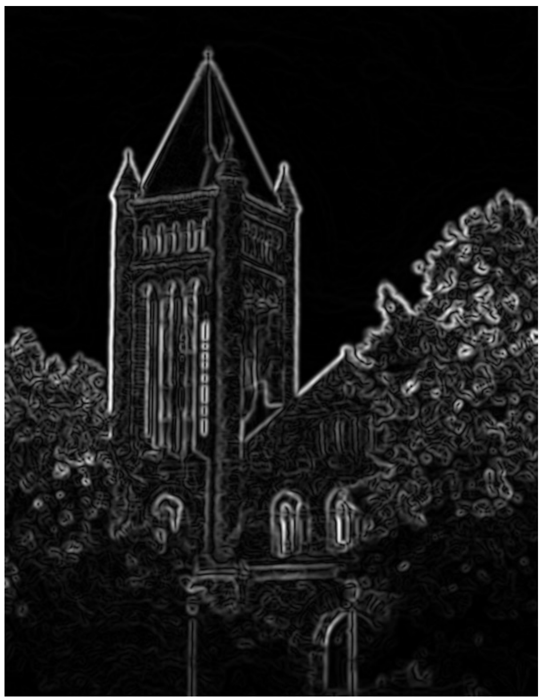




Source: S. Gupta

Building an edge detector

1. Compute x and y derivative images
2. Find magnitude and orientation of the gradient

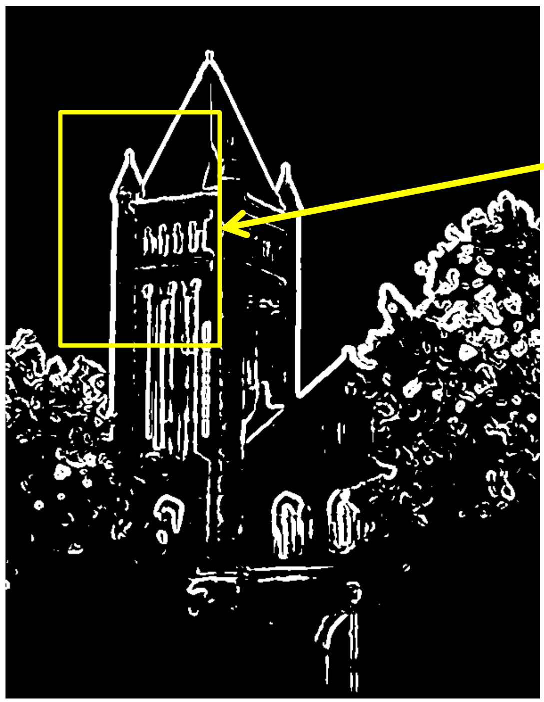


Let's threshold the gradient magnitude

Source: S. Gupta

Building an edge detector

1. Compute x and y derivative images
2. Find magnitude and orientation of the gradient

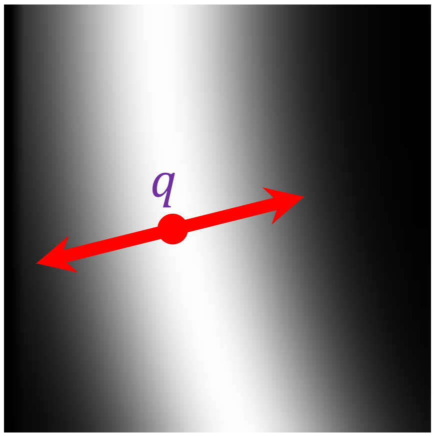


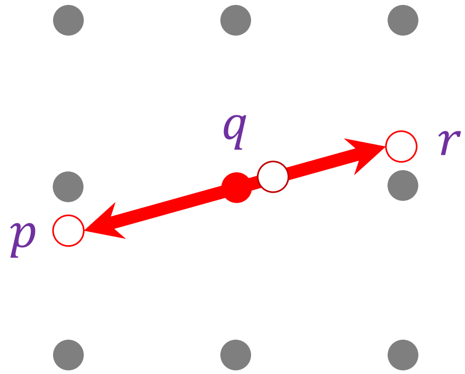
We get thick trails, not neat edge curves

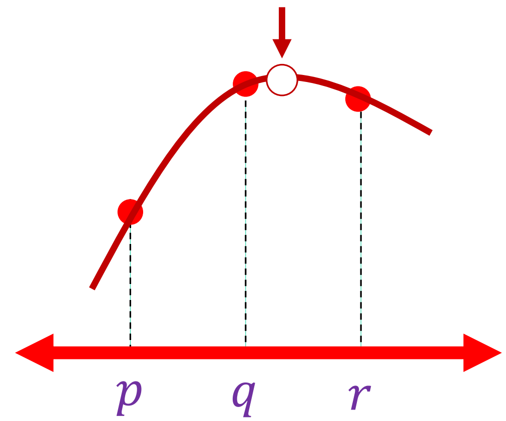
Let's threshold the gradient magnitude

Source: S. Gupta

Non-maximum suppression



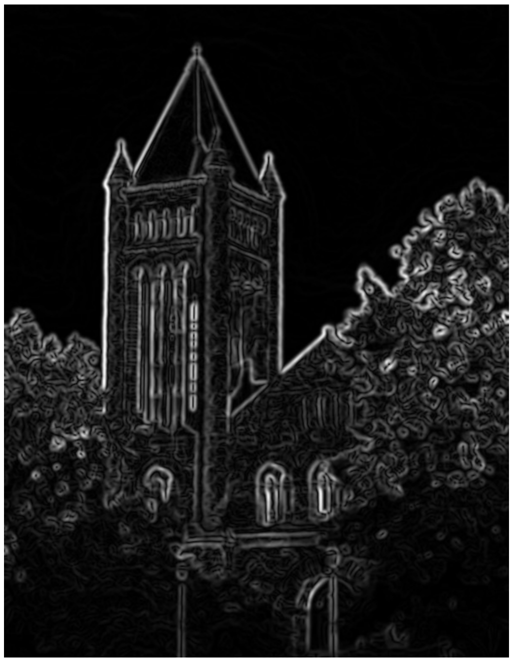


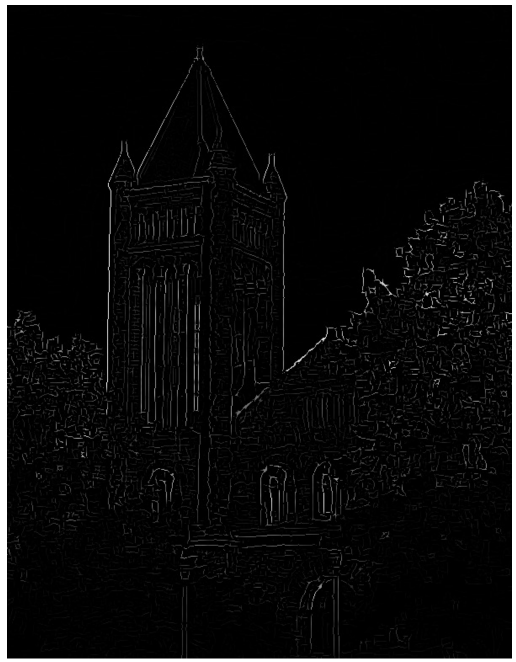


1D image "slice" normal to the edge

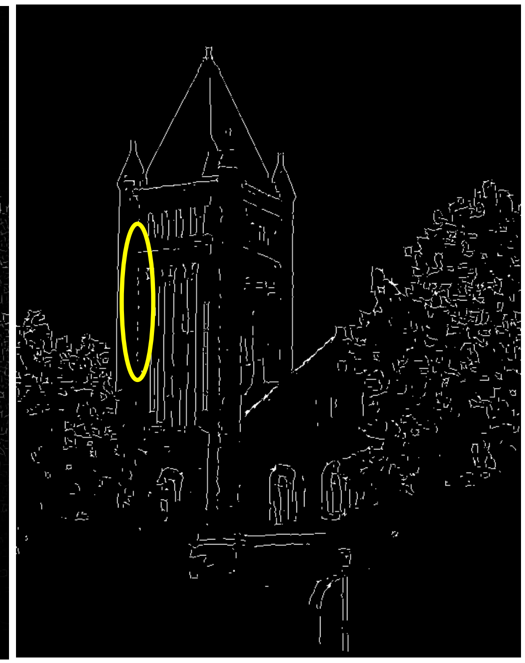
- For each location q above threshold, check that the gradient magnitude is higher than at adjacent points p and r along the direction of the gradient
  - Need to interpolate to get the gradient magnitude values at p and r
  - Can even use nonlinear interpolation to get sub-pixel edge localization!

Non-maximum suppression





NMS



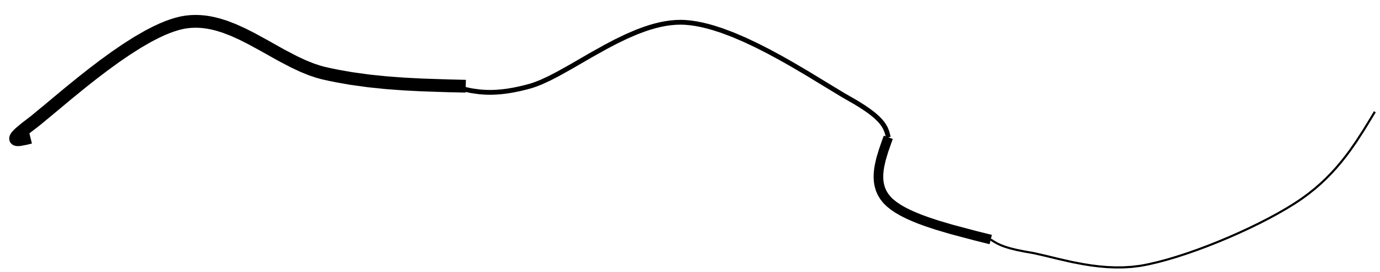
NMS > threshold

Another problem: pixels along this edge didn't survive the thresholding

Source: S. Gupta

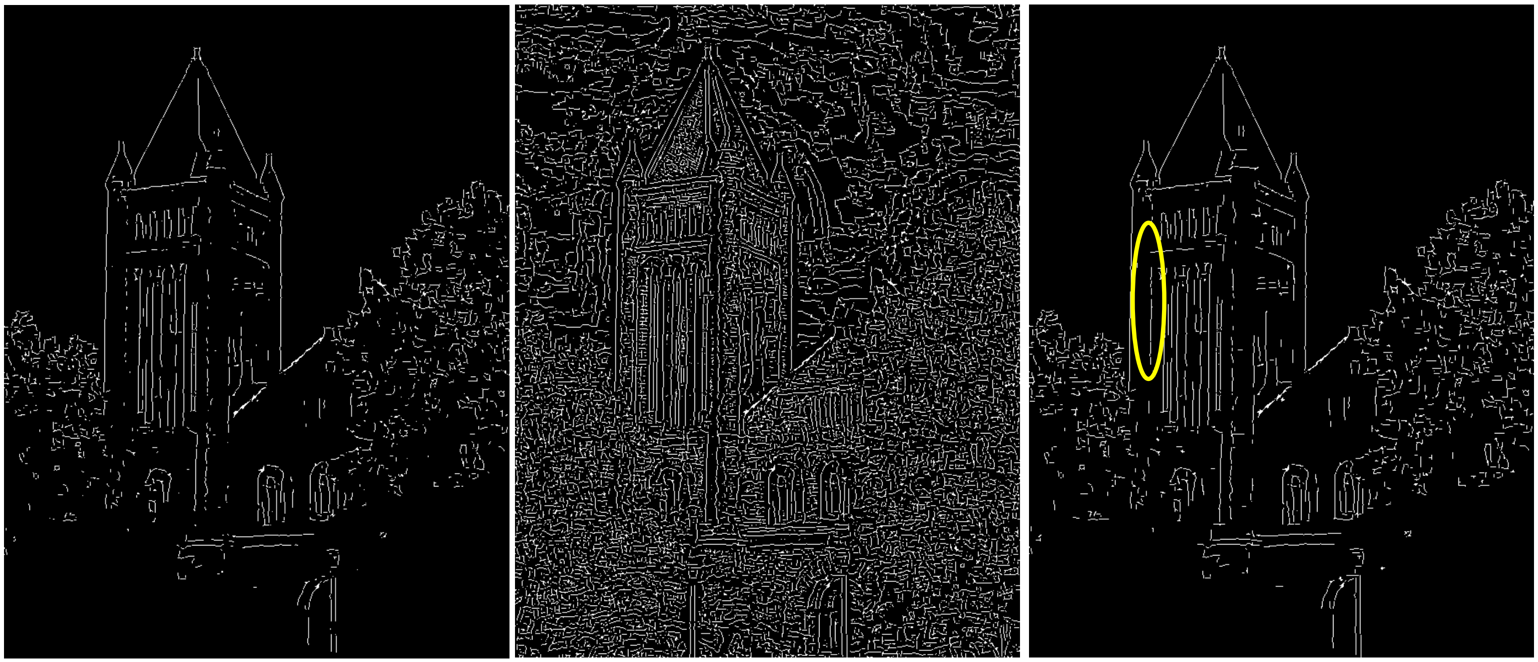
Hysteresis thresholding

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



Source: Steve Seitz

Hysteresis thresholding



high threshold
(strong edges)

low threshold
(weak edges)

hysteresis threshold

Source: S. Gupta

Recap: Canny edge detector

1. Compute x and y derivative images
2. Find magnitude and orientation of the gradient
3. Non-maximum suppression:
   - Thin wide “ridges” down to single pixel width
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

J. Canny, A Computational Approach To Edge Detection, IEEE Trans. PAMI, 8:679-714, 1986.

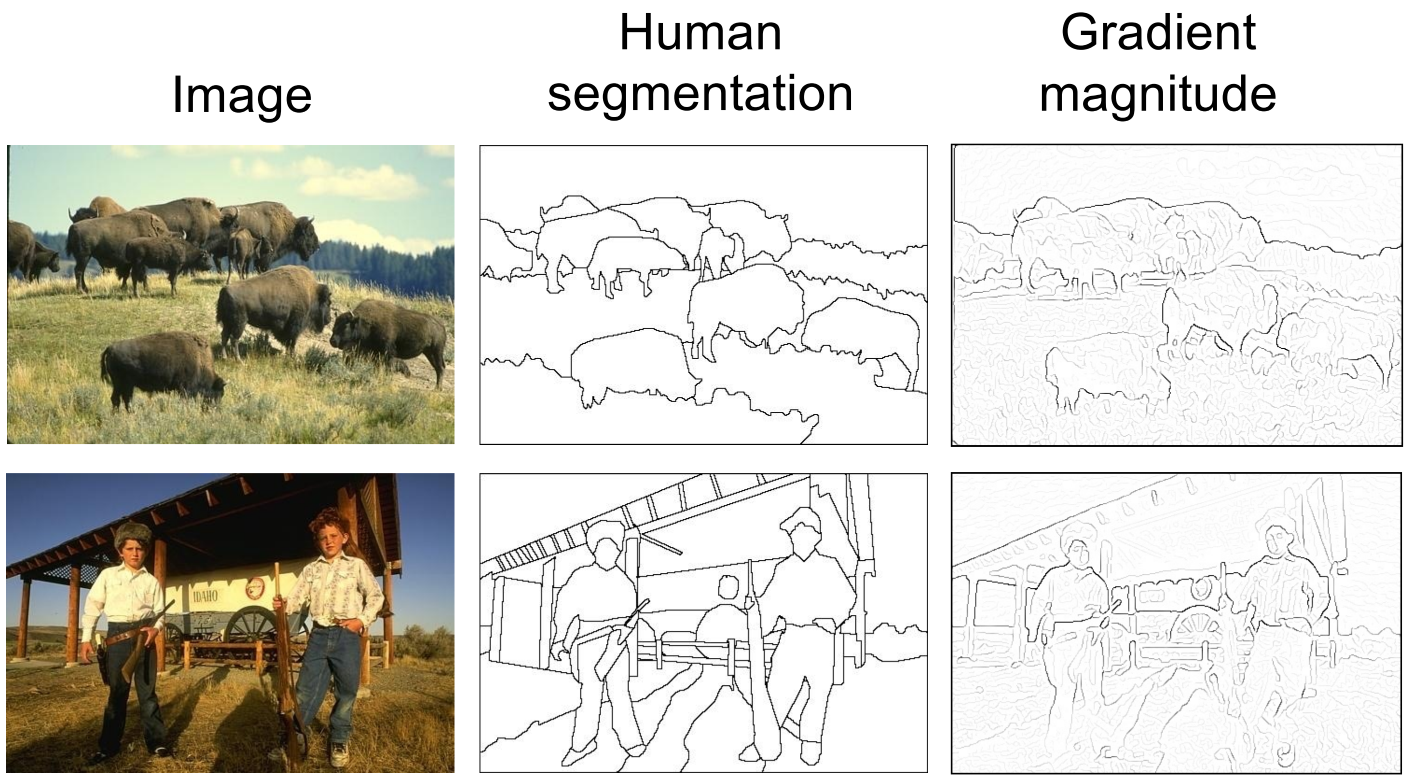
Overview

- Motivating edge detection
- Image gradients
- Derivative of Gaussian filters
- Canny edge detector
- What is the role of edge detection in image understanding?

Top-down vs. bottom-up edge detection

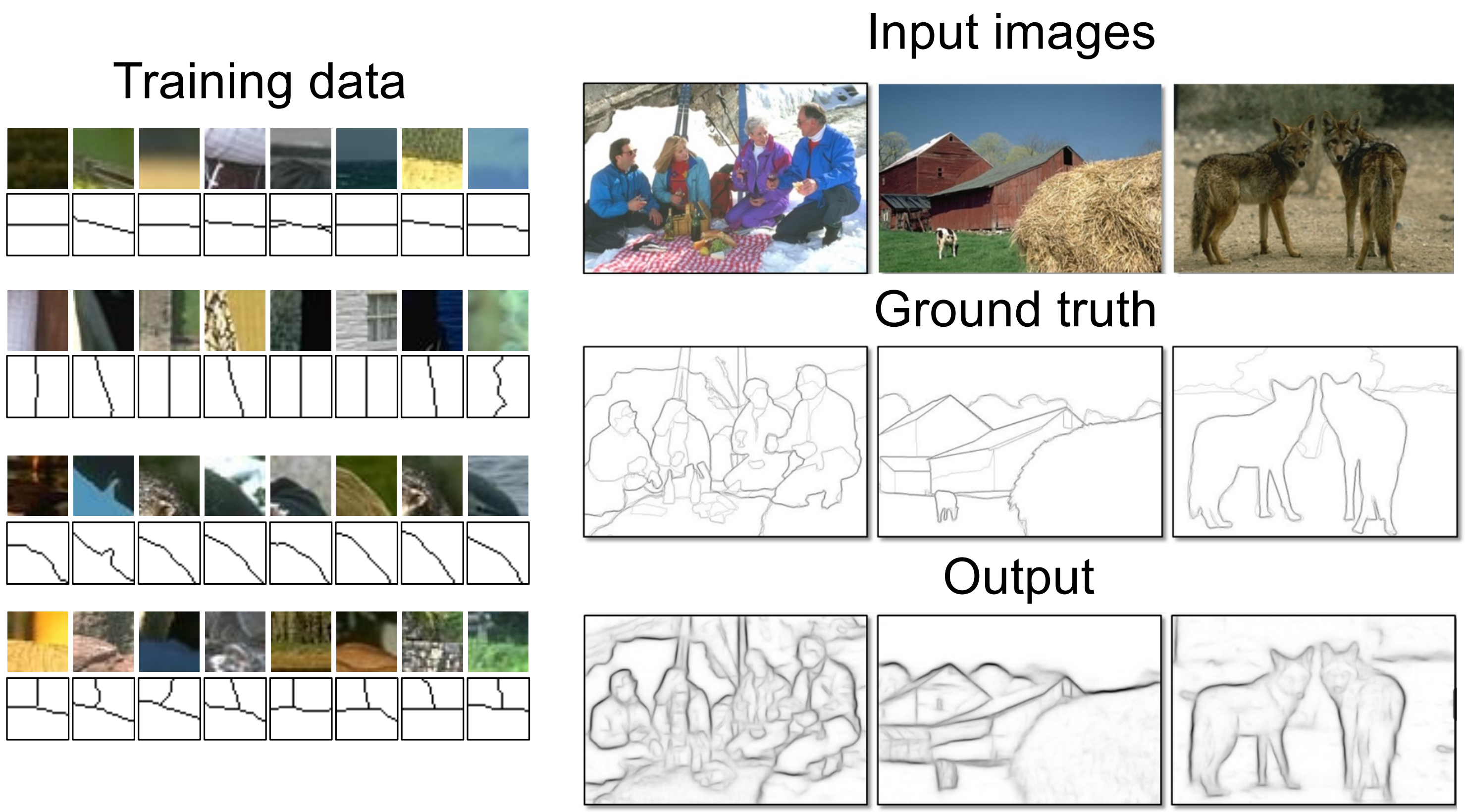
Figure from Marr (1982), attributed to R. C. James

Top-down vs. bottom-up edge detection



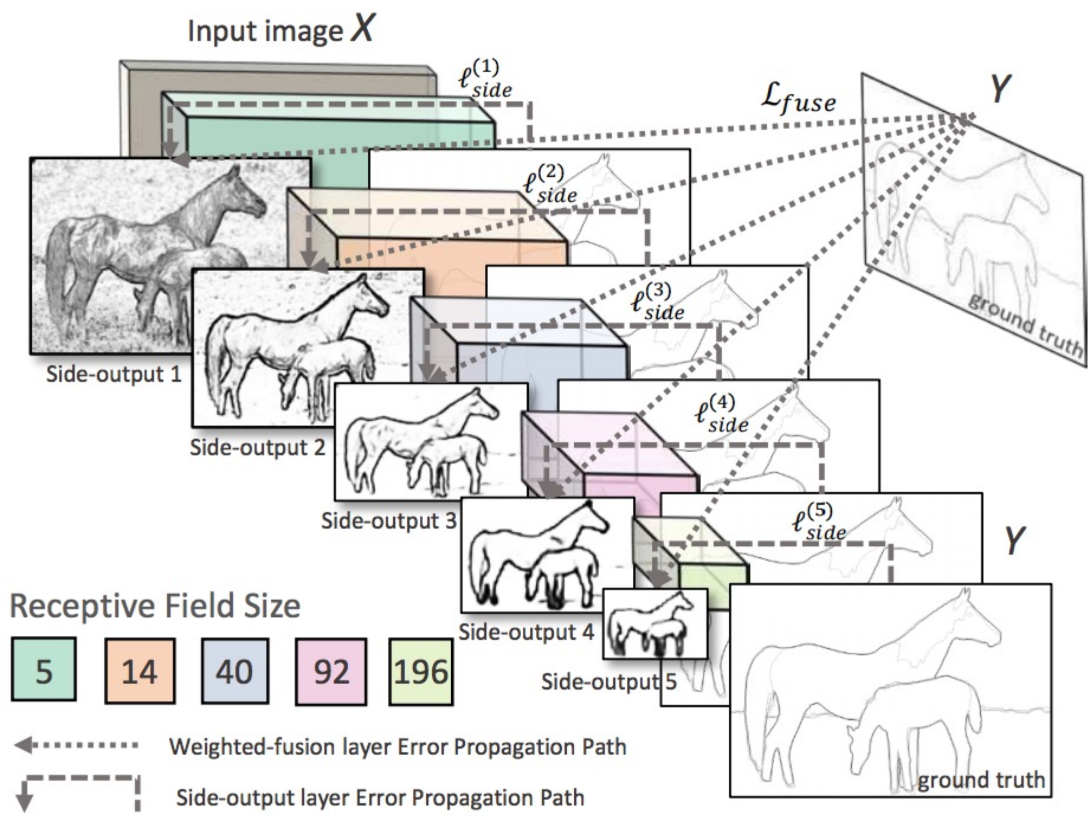
D. Martin, C. Fowlkes, D. Tal, and J. Malik. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. ICCV 2001

Trainable edge detection



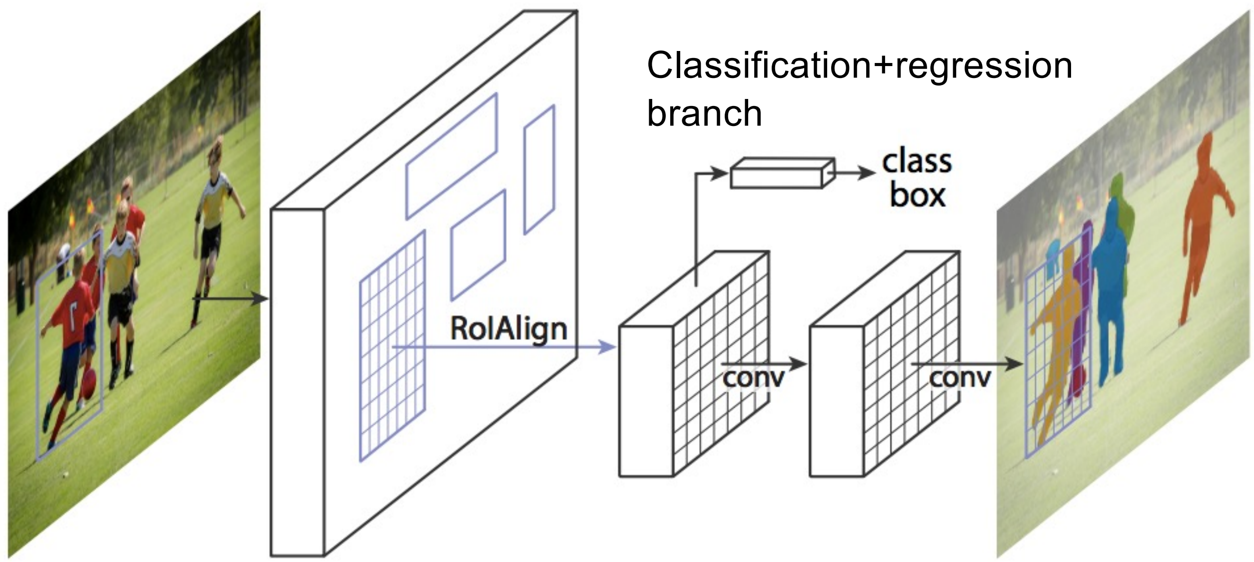
P. Dollar and L. Zitnick. [Structured forests for fast edge detection](). ICCV 2013

Trainable edge detection



S. Xie and Z. Tu. [Holistically-nested edge detection](). ICCV 2015

Today: Segmentation instead of edge detection



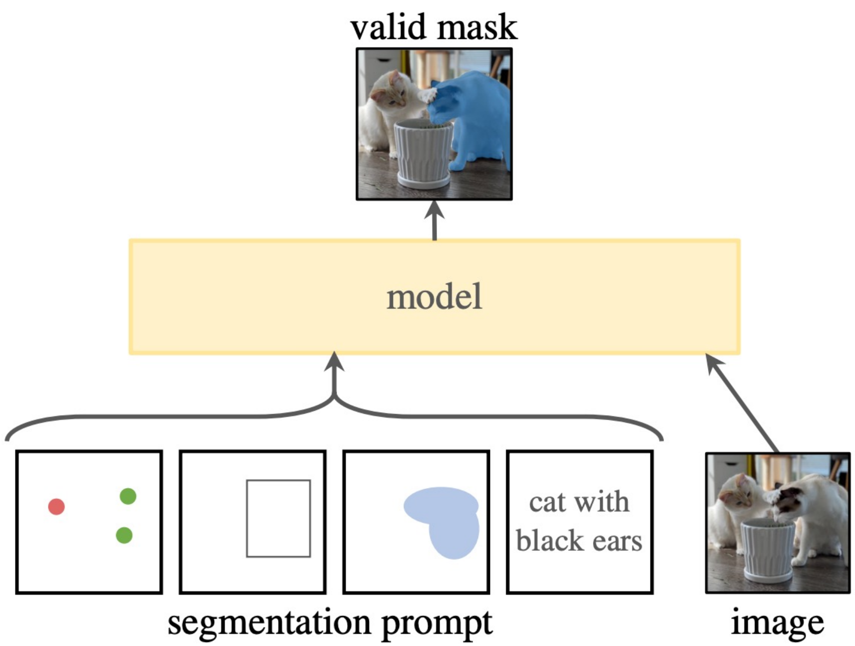
Classification+regression branch

Mask branch: output binary segmentation for each instance

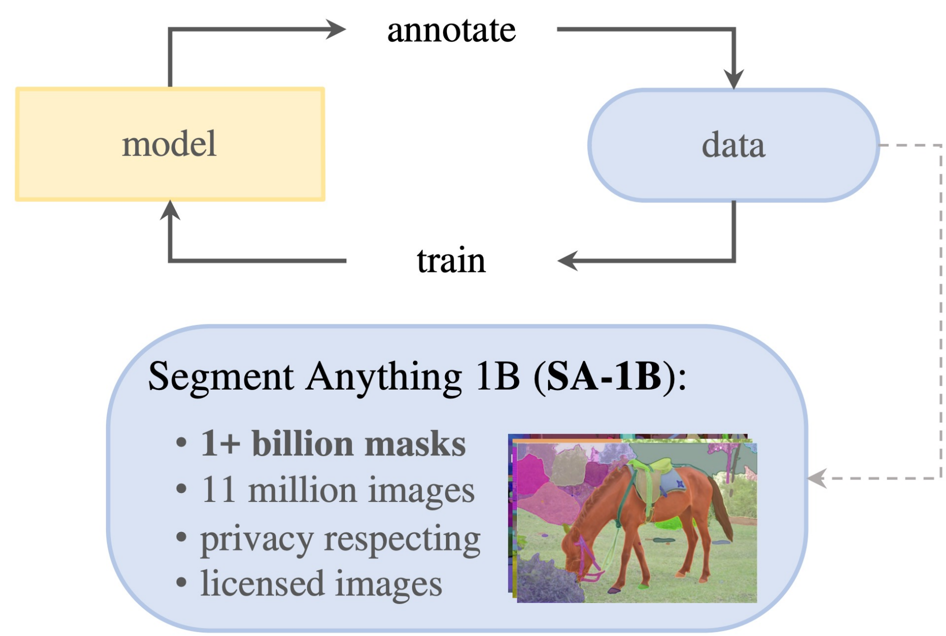
K. He et al. [Mask R-CNN](). ICCV 2017

Today: Massively trained promptable segmentation

Task: Promptable segmentation



Data engine



A. Kirilllov et al. [Segment anything](). ICCV 2023