Edges, Contours and Lines

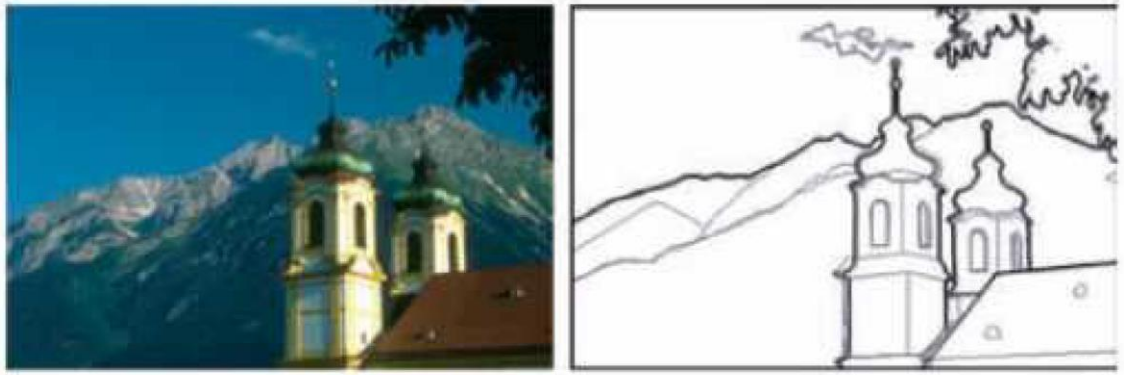
CS 6384 Computer Vision

Professor Yu Xiang

The University of Texas at Dallas

Some slides of this lecture are courtesy Robert Collins (PSU)

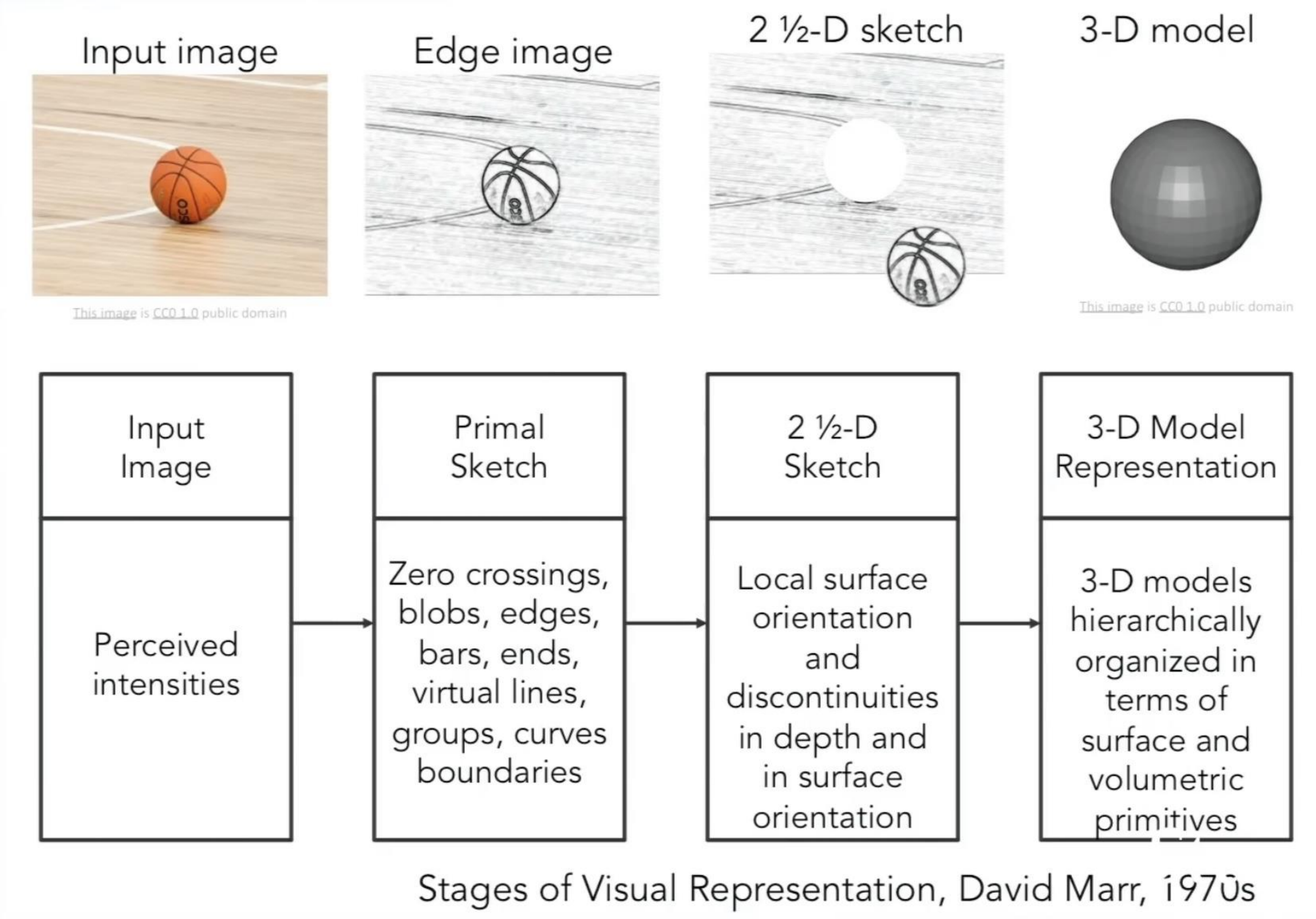
Keypoint Features vs. Edge Points



Keypoints
- Good for feature matching
- Less or no semantic meaning

Edges
- Not robust for feature matching
- With semantic meanings (object boundaries, occlusion boundaries, shadows, etc.)

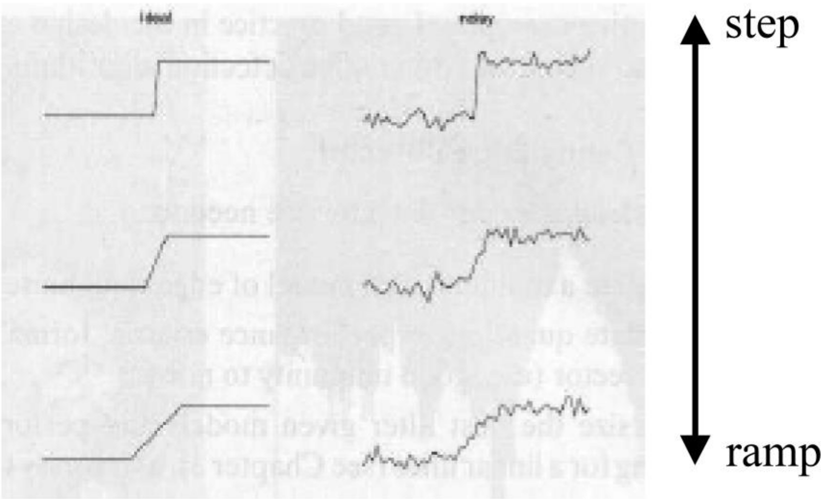
David Marr’s Theory of Vision (Neuroscientist)



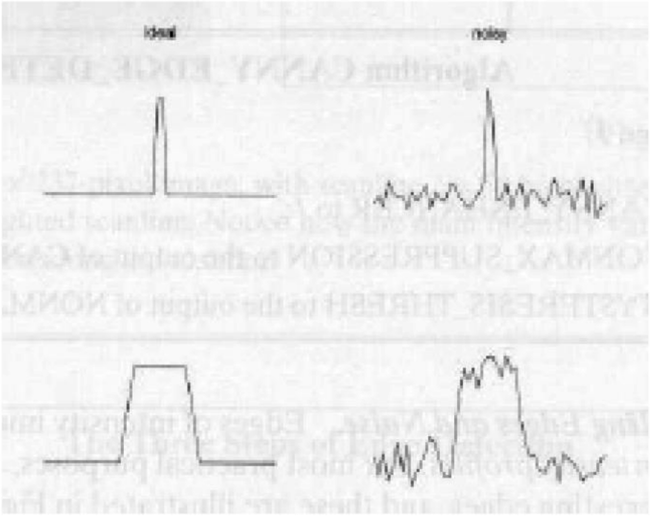
D. Marr. Vision. W. H. Freeman and Co., 1982.

Edges

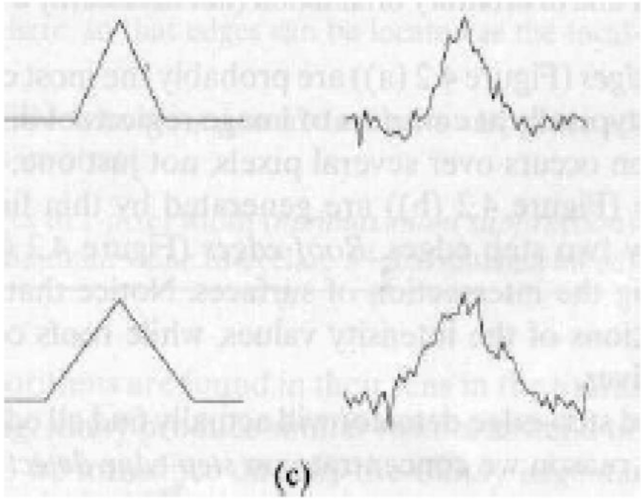
- Edges occur at boundaries between regions of different color, intensity or texture



Step Edge, Ramp Edge



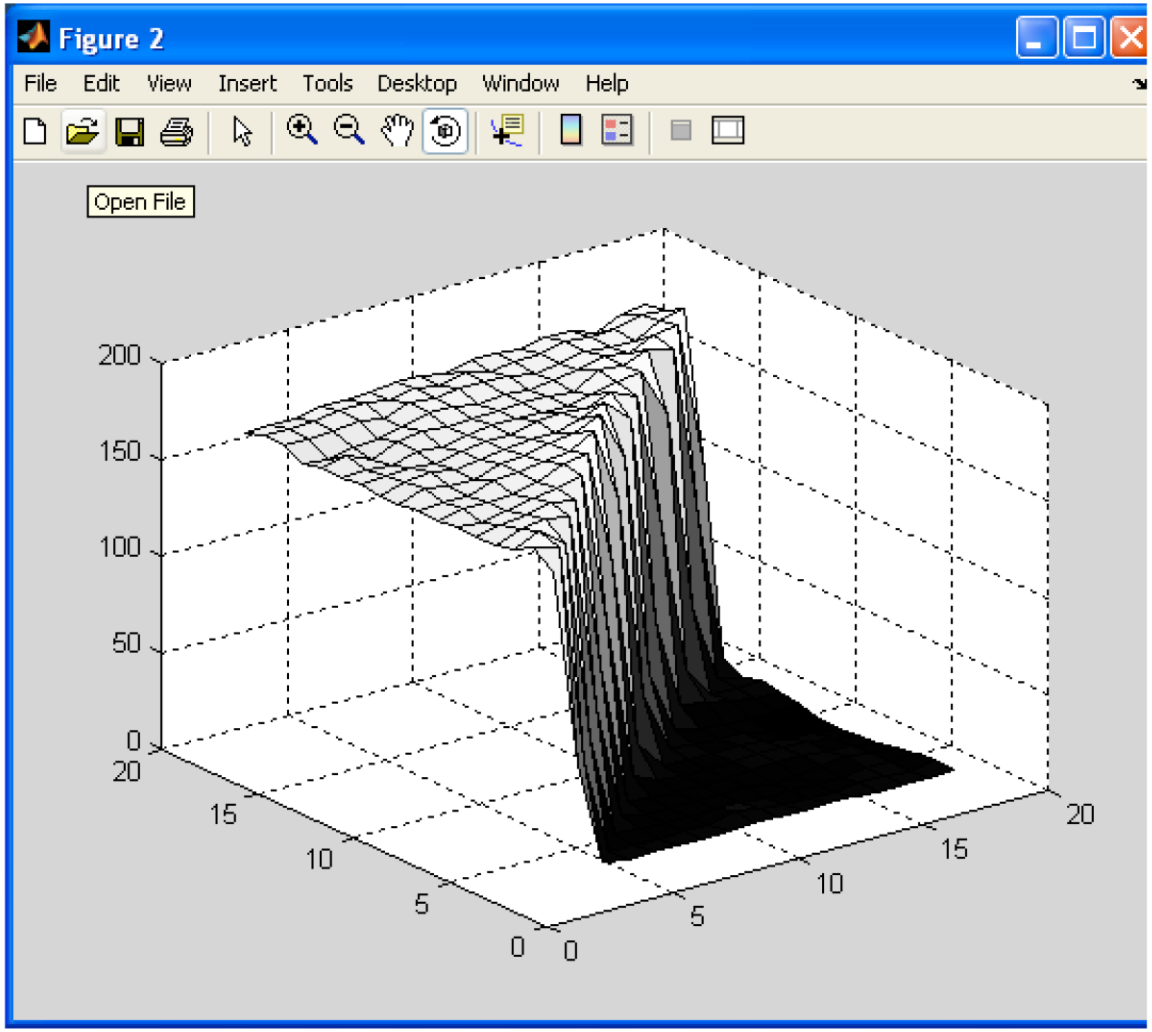
Ridge Edge



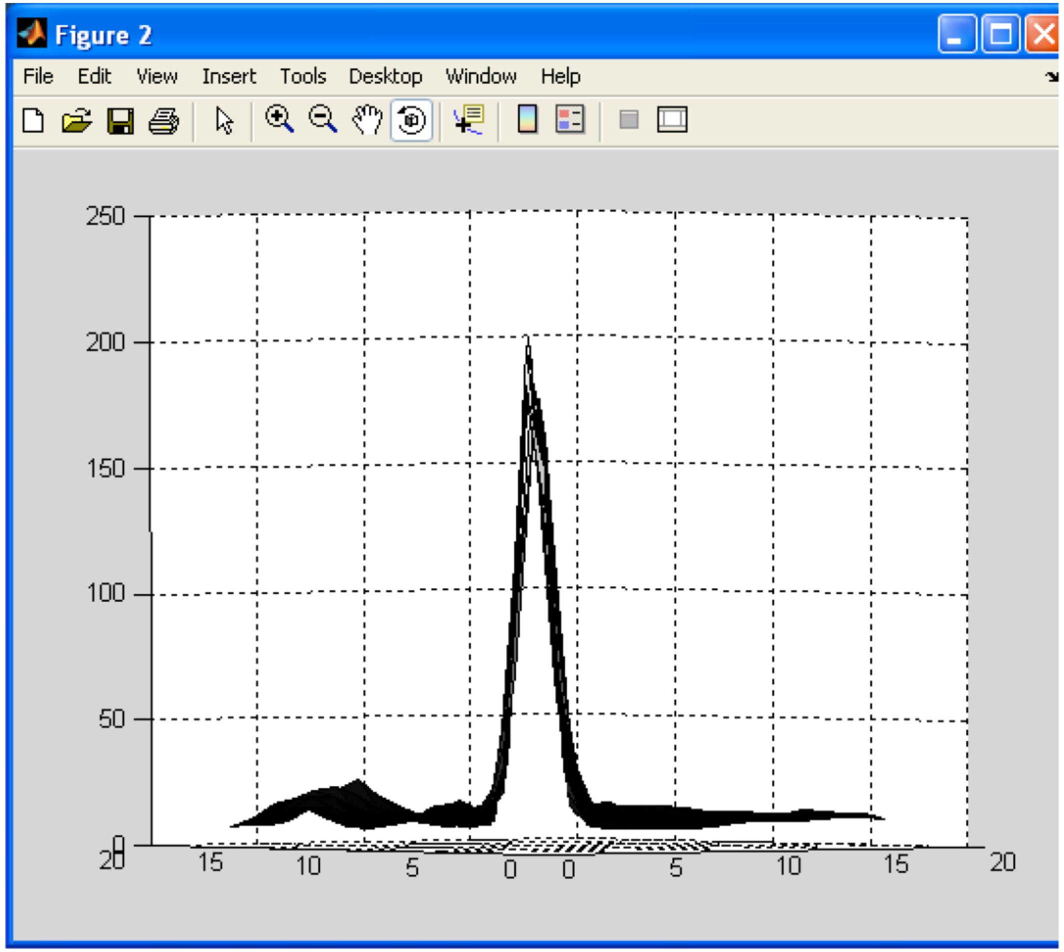
Roof Edge

Step Edge and Ramp Edge

Step Edge, Ramp Edge

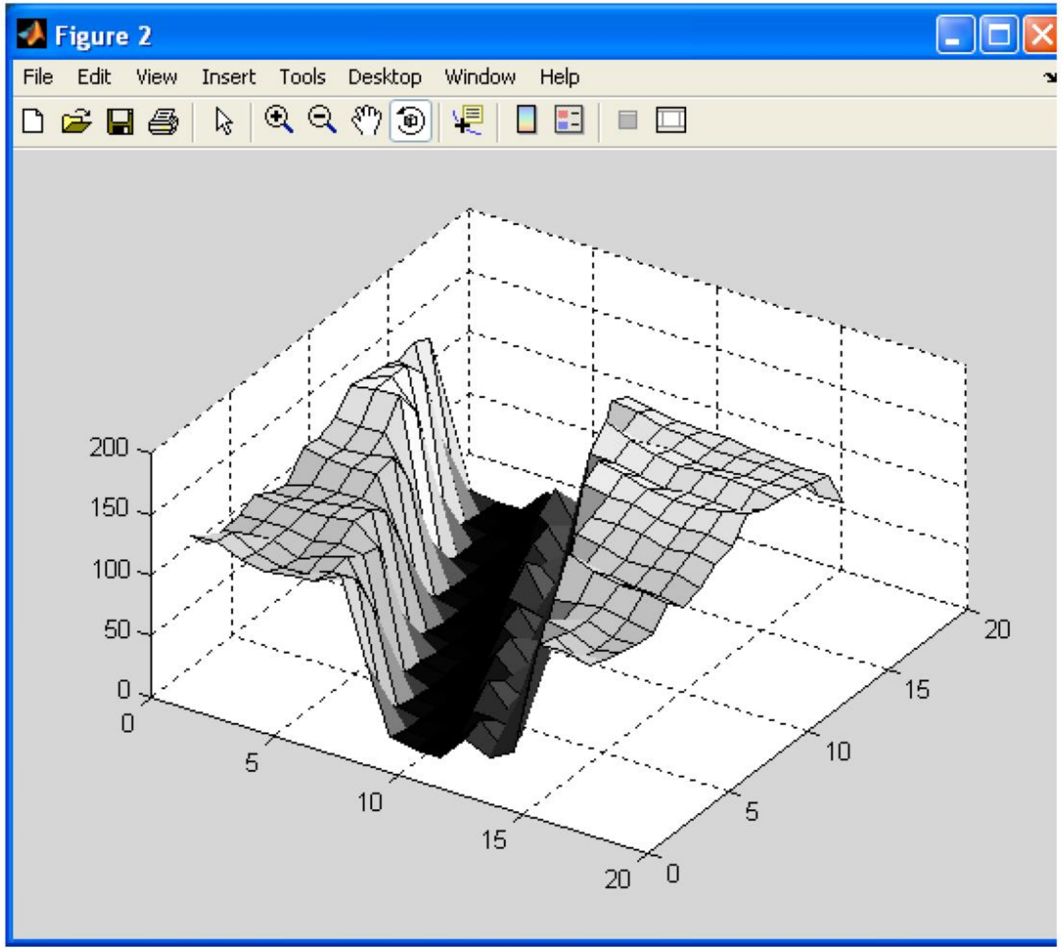


Ridge Edge



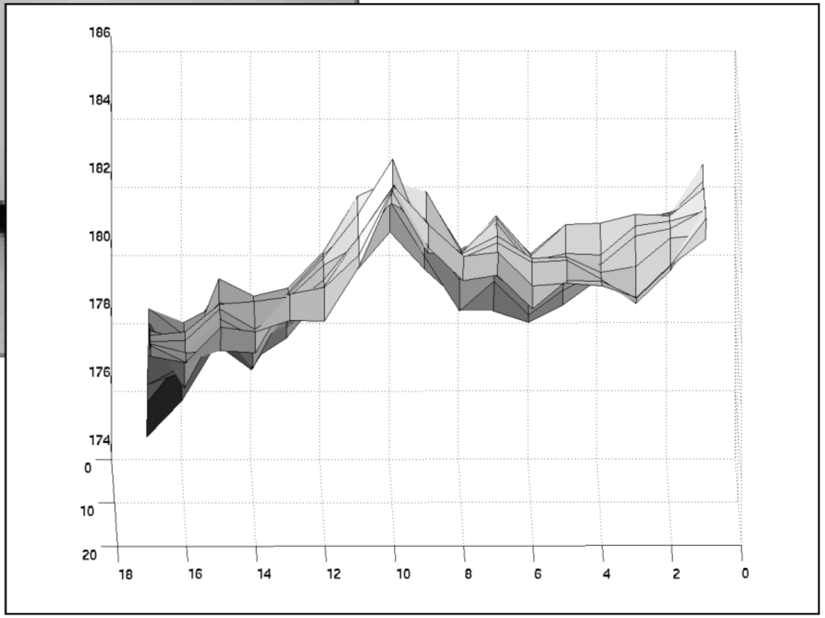
Ridge Edge

Ridge Edge



Ridge Edge

Roof Edge



Roof Edge

Image Gradients

- Use image gradients

Central difference

$$f'(x) = \lim_{h \to 0} \frac{f(x+0.5h) - f(x-0.5h)}{h}$$

-1	0	1

X derivative

Image f

Gaussian Filter h

Convolution $h \star f$

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

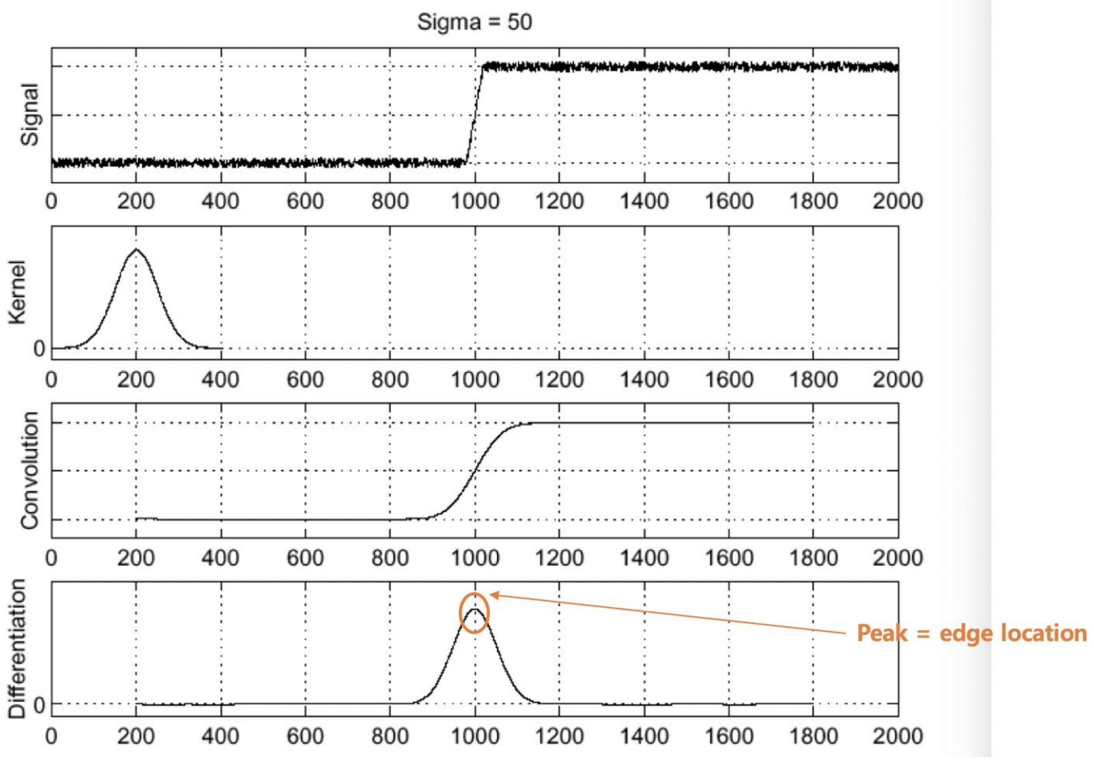
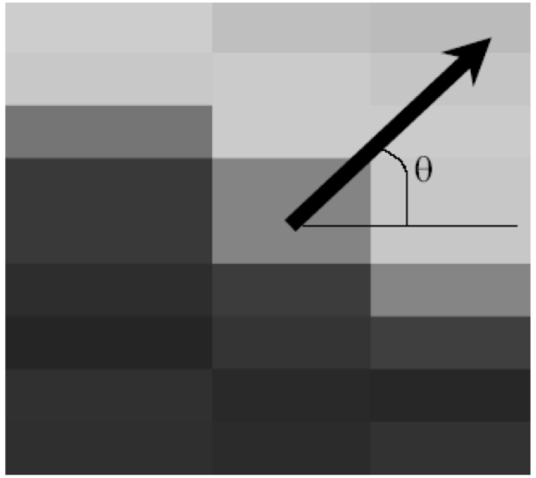
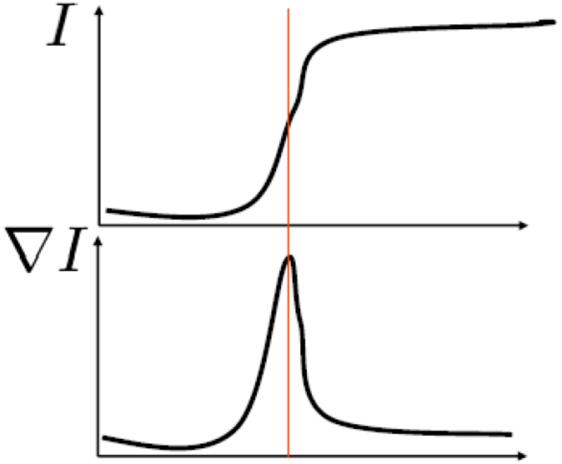


Image Gradients

Gradient Vector: $\nabla I = [\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}]^T$





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

Magnitude:

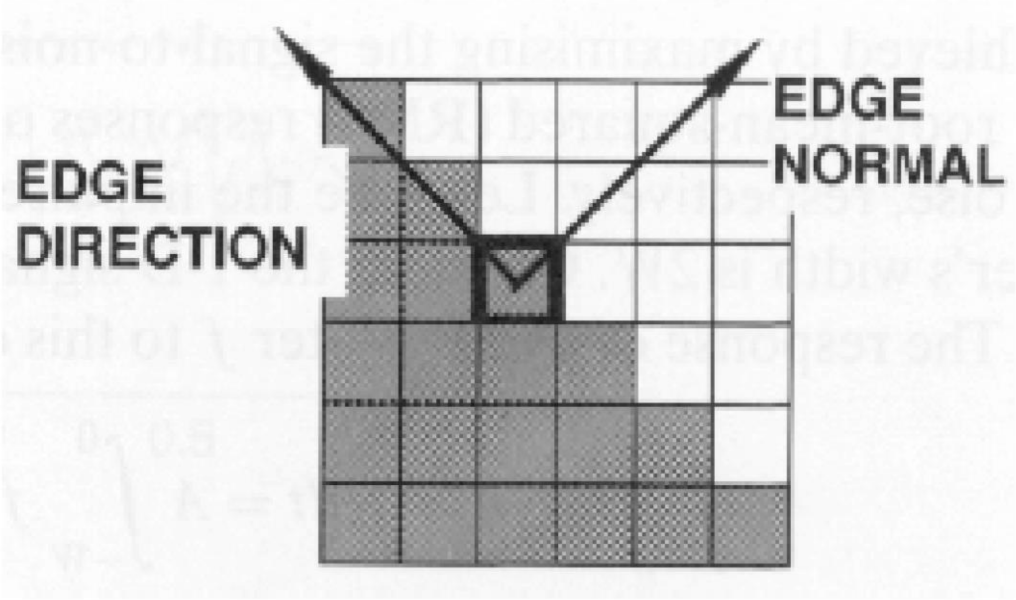
$$\theta = atan2\left(\frac{\partial I}{\partial y}, \frac{\partial I}{\partial x}\right)$$

Orientation

Edge Normal and Edge Direction

- Edge normal
  - Unit vector in the direction of maximum intensity change
  - Gradient direction

- Edge direction
  - Unit vector along edge (perpendicular to edge normal)



Edge Detection

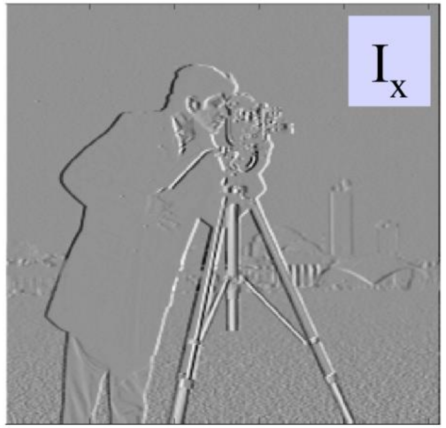
- A simple edge detector using gradient magnitude

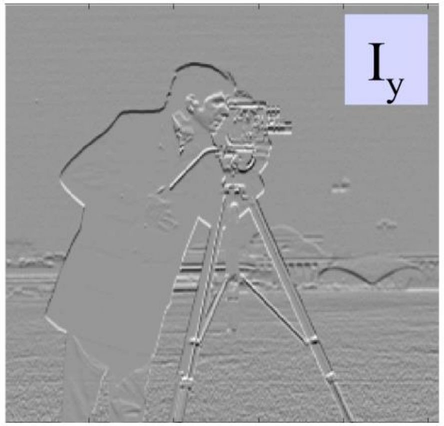
1. Compute gradient vector at each pixel by convolving image with horizontal and vertical derivative filters

2. Compute gradient magnitude at each pixel

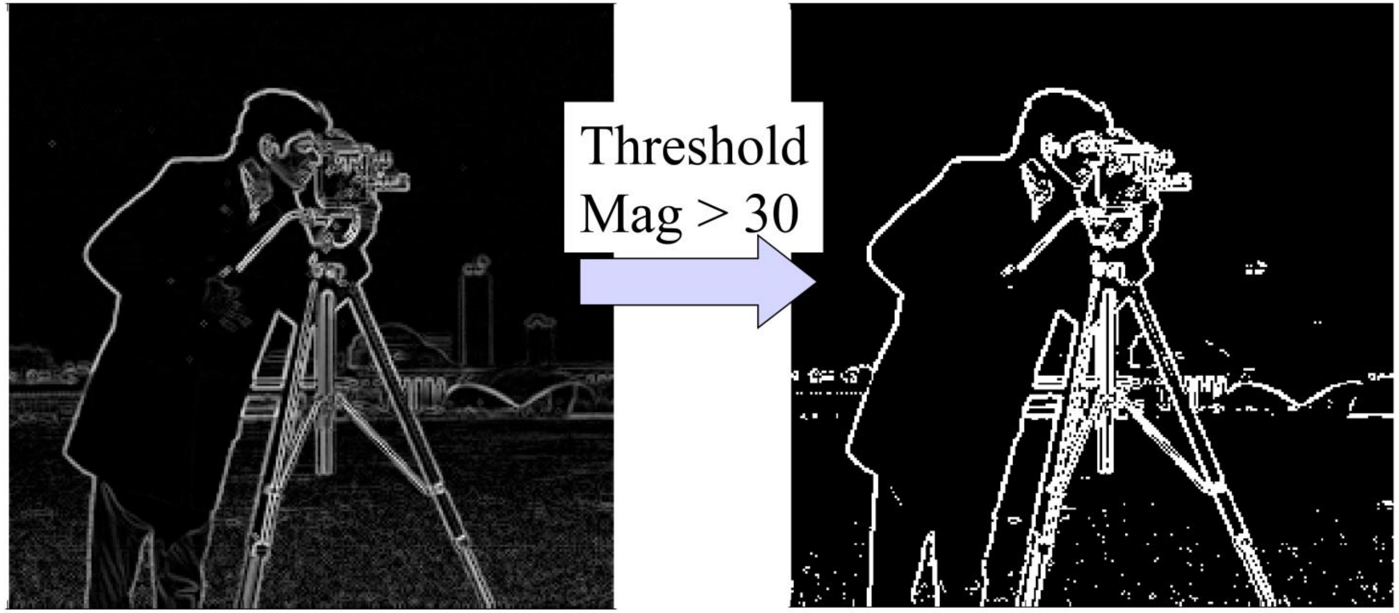
3. If magnitude at a pixel exceeds a threshold, report a possible edge point

Edge Detection





Magnitude of gradients



Edge Detection

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

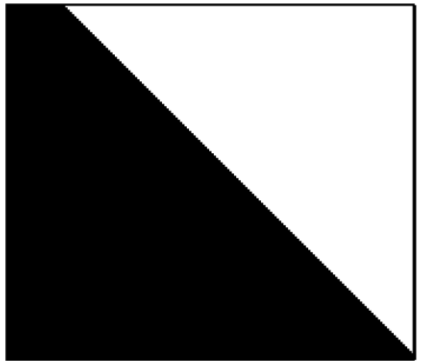
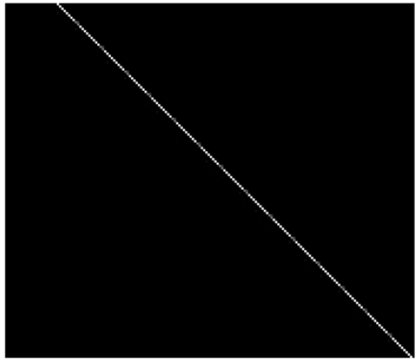


Image with Edge



Edge Location

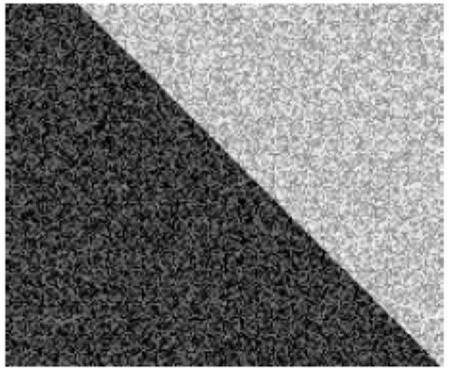
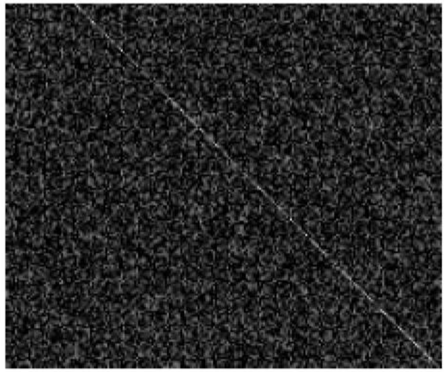
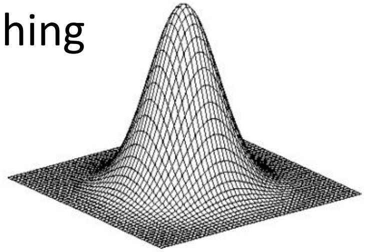


Image + Noise



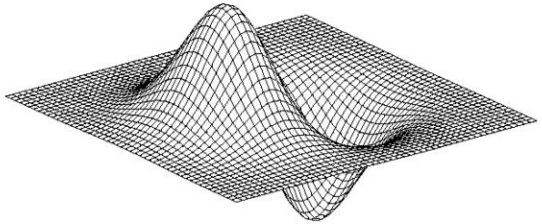
Derivatives detect
edge *and* noise

Smoothing



Gaussian

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



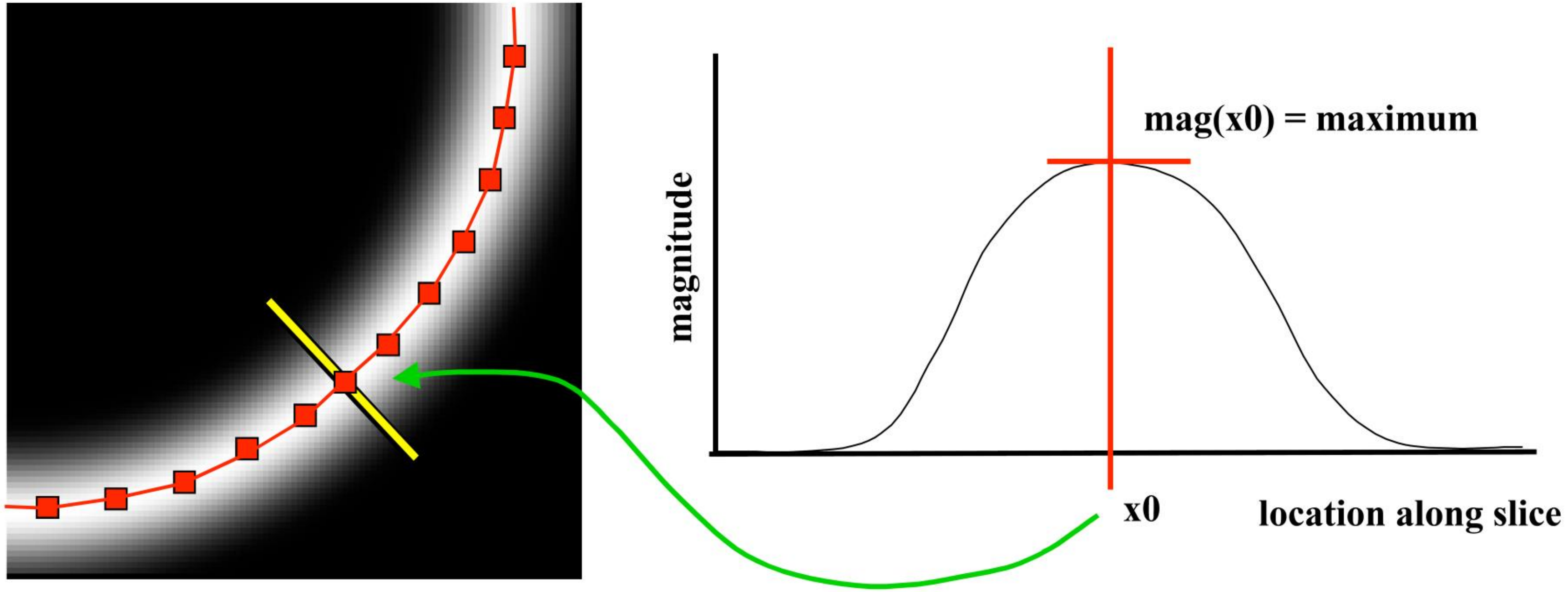
derivative of Gaussian (x)

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

Smoothed derivative removes
noise, but blurs edge

Edge Detection

- Thinning

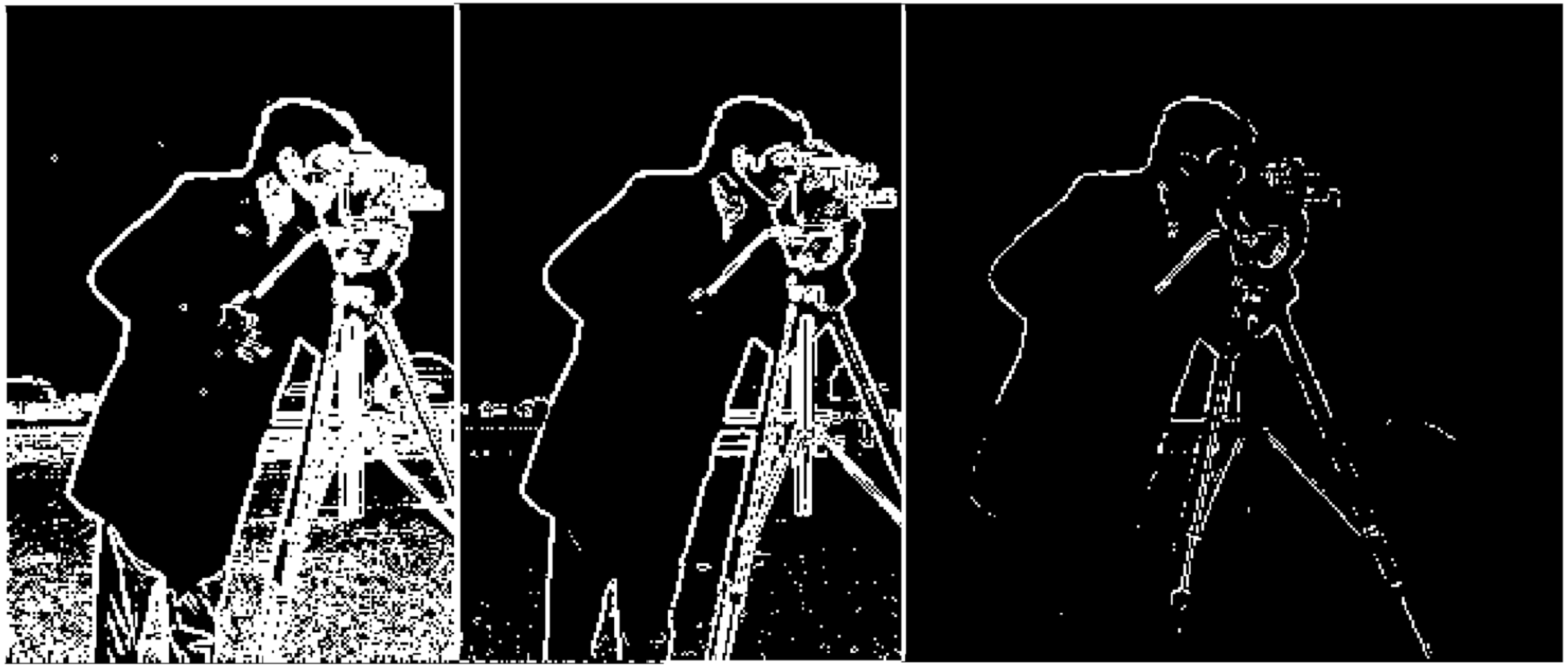


Along a 1D intensity slice normal to the curve (non-maximum suppression)

- Direction of gradient

Edge Detection

- How to chose the threshold?

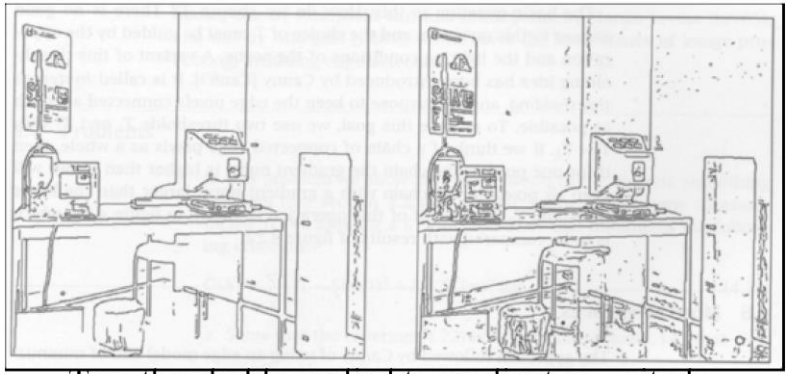


> 10 > 30 > 80

Edge Detection

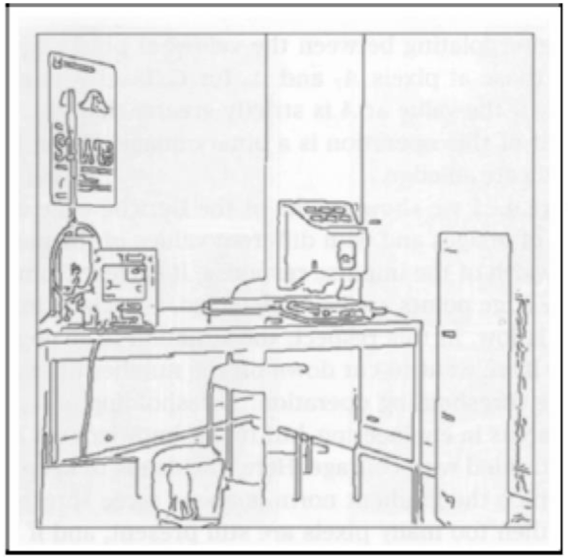
- How to chose the threshold?

- Hysteresis thresholding
  - Keep a high threshed H and a low threshold L
  - Any edge with strength < L is discarded
  - Any edge with strength > H is kept
  - An edge P with strength between L and H is kept only if there is a path of edges with strength > L connecting P to an edge of strength > H



Two thresholds applied to gradient magnitude

T = 15 T = 5



Hysteresis

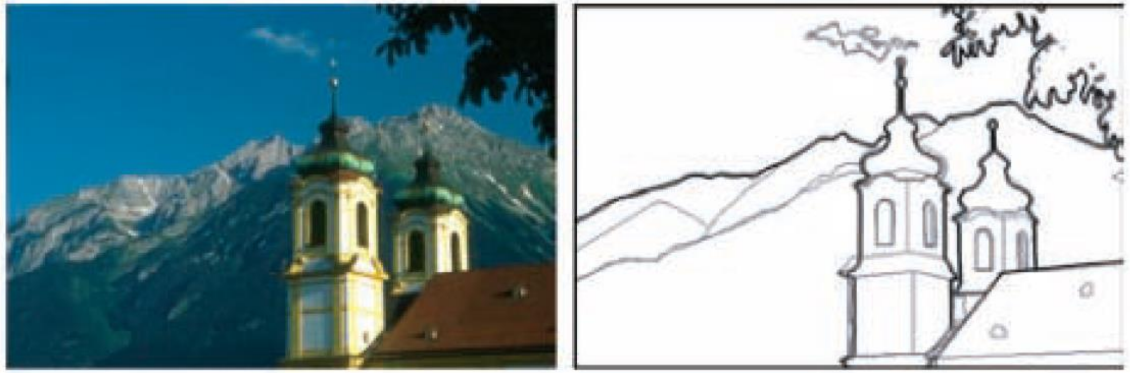
T_h=15 T_l = 5

Canny Edge Detector

J. Canny A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8, No. 6, Nov 1986

Contour Detection

- Link edge points into contours
  - Check neighboring pixels



- How to store contours?
  - A list of edgels (edge points)
  - (x, y) coordinates

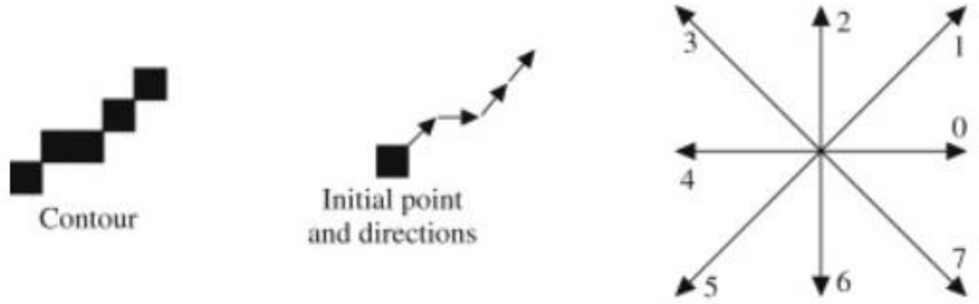
Contour Detection

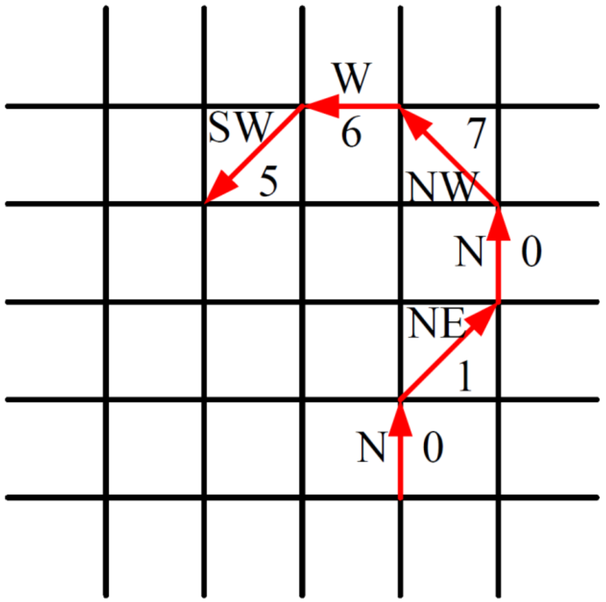
- How to store contours?
- Chain code
  - Initial coordinates
  - 8 directions (N, NE, E, SE, S, SW, W, NW)

  - 3 bits (can further be compressed)

    0 1 0 7 6 5

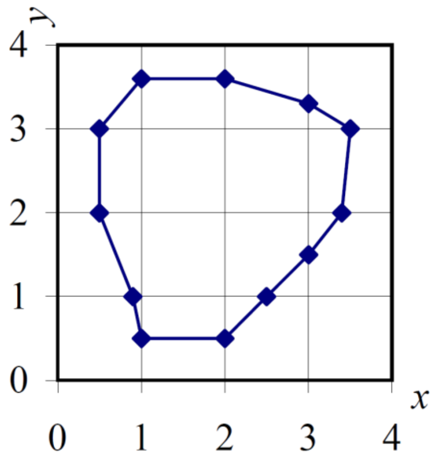
  - Not suitable for further processing

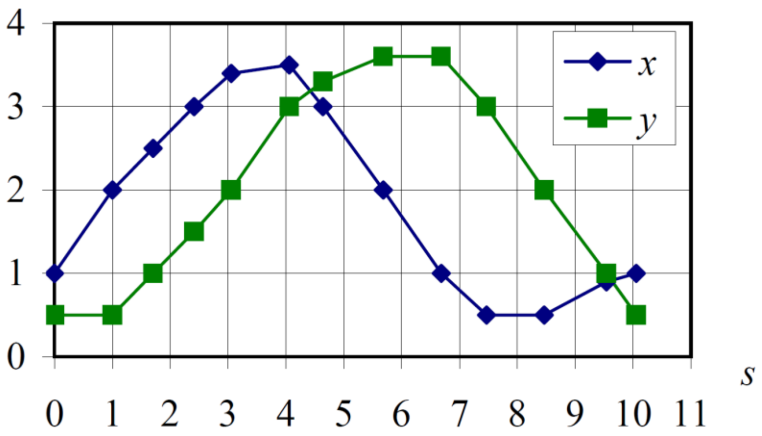




Contour Detection

- How to store contours?
- Arc-length parameterization $\mathbf{x}(s)$

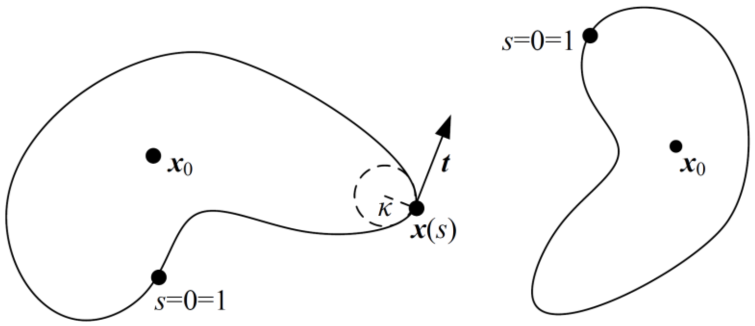




- Start point (1.0, 0.5), s = 0
- Next point (2.0, 0.5), s = 1
- Next point (2.5, 1.0), s = 1.7071

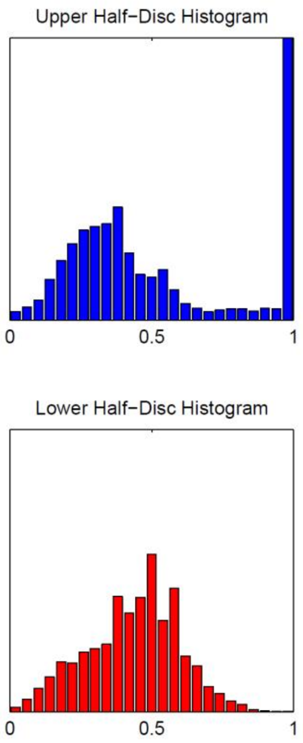
…

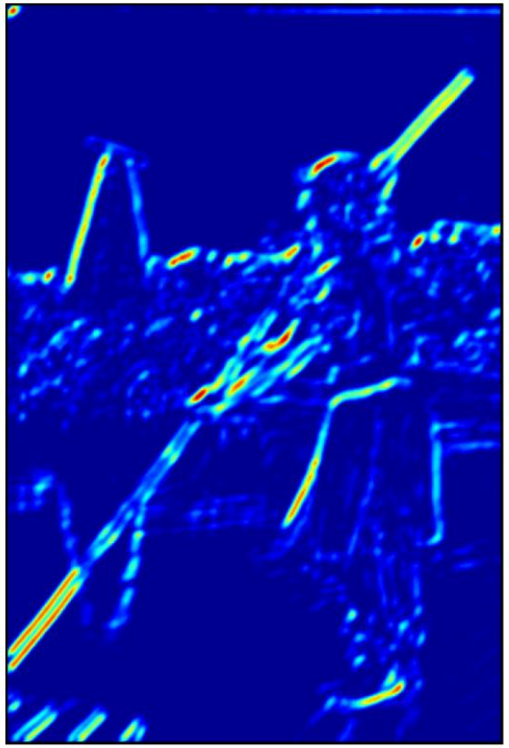
- Can be resampled
- Fourier transform by treating (x, y) as a complex number (contour matching)



mPb Contour Detector

- Oriented gradient of histograms $G(x, y, \theta)$





Gradient magnitude:

χ^2 distance between the two histograms

$$\chi^2(g, h) = \frac{1}{2} \sum_i \frac{(g(i) - h(i))^2}{g(i) + h(i)}$$

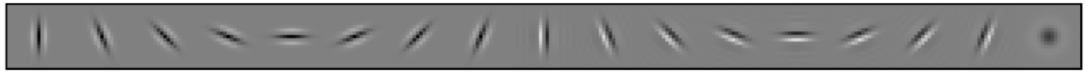
Histogram of intensity

Radius: 5 pixels $\theta = \frac{\pi}{4}$

Pablo Arbel´aez, Charless Fowlkes, Jitendra Malik. Contour Detection and Hierarchical Image Segmentation. TPAMI’10

mPb Contour Detector

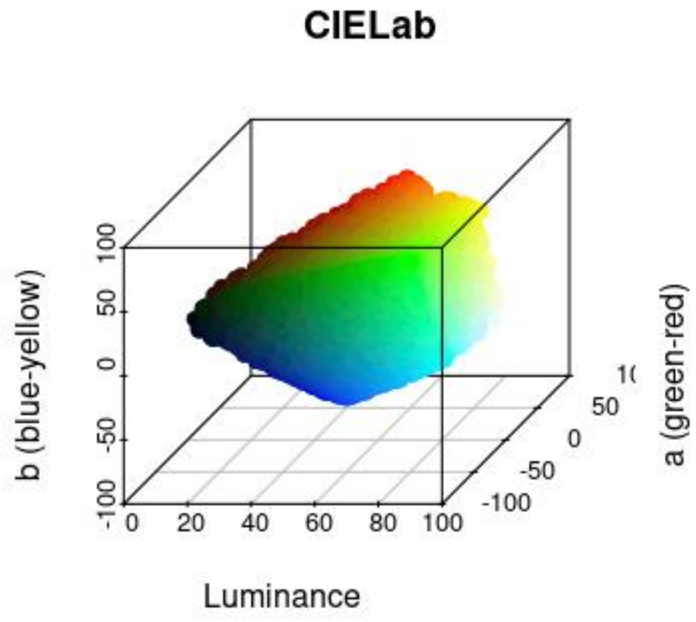
- Brightness, color, texture gradients
  - L*a*b color space: brightness, color a and color b
  - Texture: assign a pixel to a texton id



- 17 Gaussian derivative filters

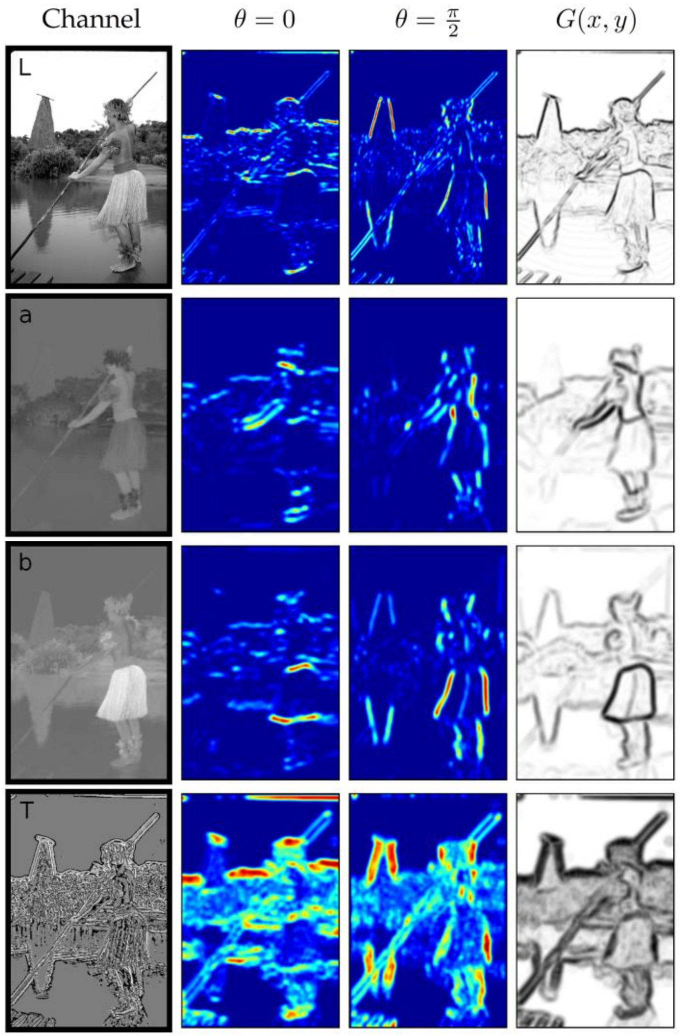
→ 17D feature vector for each pixel

- K-means clustering, K = 32, textons
- Texture image: pixels with integer [1, K]



https://cran.r-project.org/web/packages/colordistance/vignettes/color-spaces.html

mPb Contour Detector



maximum response over eight orientations in [0; π)

- Consider multiple scales $\left[\frac{\sigma}{2}, \sigma, 2\sigma\right]$

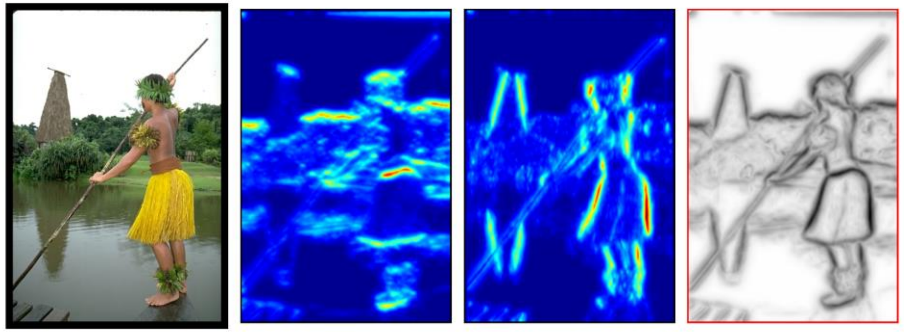
$\sigma = 5$ pixels for brightness

$\sigma = 10$ pixels for color and texture

$$mPb(x, y, \theta) = \sum_s \sum_i \alpha_{i,s} G_{i,\sigma(i,s)}(x, y, \theta)$$

Scale Channel (brightness, color a, color b, texture)

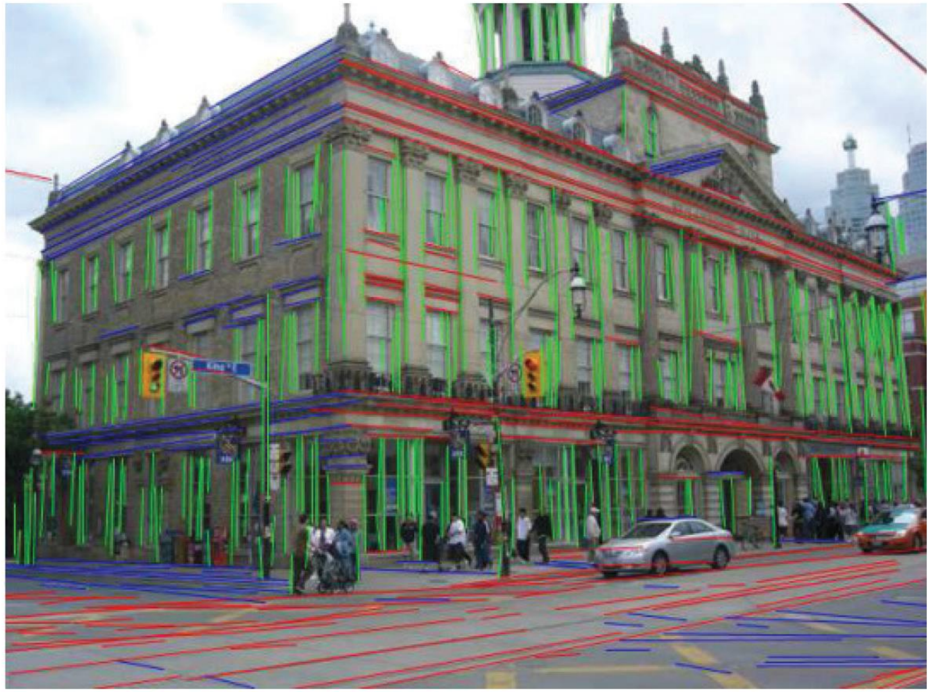
$$mPb(x, y) = \max_\theta \{mPb(x, y, \theta)\}$$



$mPb(x, y)$

Lines

- Lines are common in the human-made world

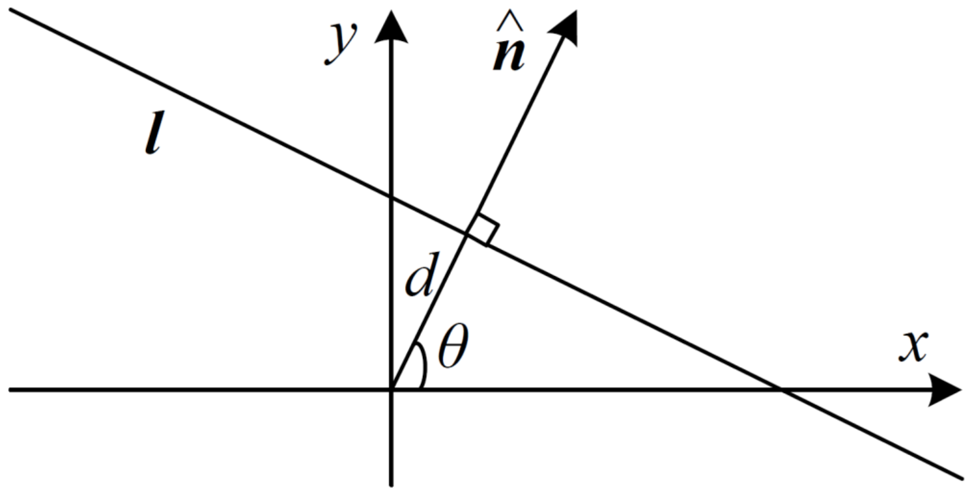


Manhattan World Assumption

Srikumar Ramalingam and Matthew Brand. Lifting 3D Manhattan Lines from a Single Image. ICCV'13.

Lines

- 2D lines



$$ax + by + c = 0$$

$$\mathbf{l} = (a, b, c)$$

Normalize by $\sqrt{a^2 + b^2}$

$$\mathbf{l} = (\hat{n}_x, \hat{n}_y, d) = (\hat{\mathbf{n}}, d)$$

$$\hat{\mathbf{n}} = (\hat{n}_x, \hat{n}_y) = (\cos\theta, \sin\theta)$$

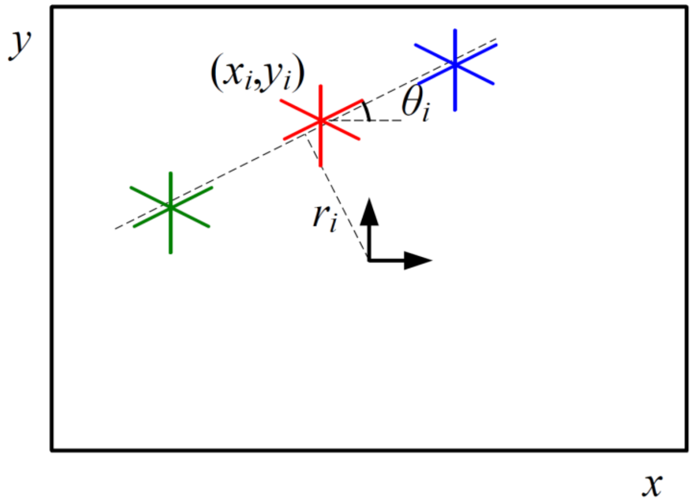
polar coordinates (θ, d)

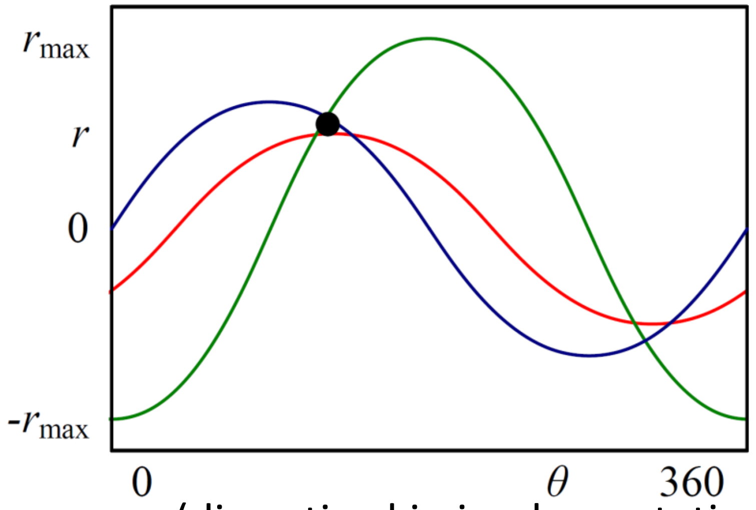
$$x \cos\theta + y \sin\theta + d = 0$$

Line Detection

- Hough transform
  - Observations vote for model parameters
  - Observations? (x_i, y_i) Edge points
  - Model parameters? (r, θ)

$$r_i(\theta) = x_i \cos\theta + y_i \sin\theta$$





Parameter space (discretized in implementation)

$$a\cos x + b\sin x = c\cos(x + \varphi)$$

where c and φ are defined as so:

$$c = \operatorname{sgn}(a)\sqrt{a^2 + b^2},$$

$$\varphi = \arctan\left(-\frac{b}{a}\right),$$

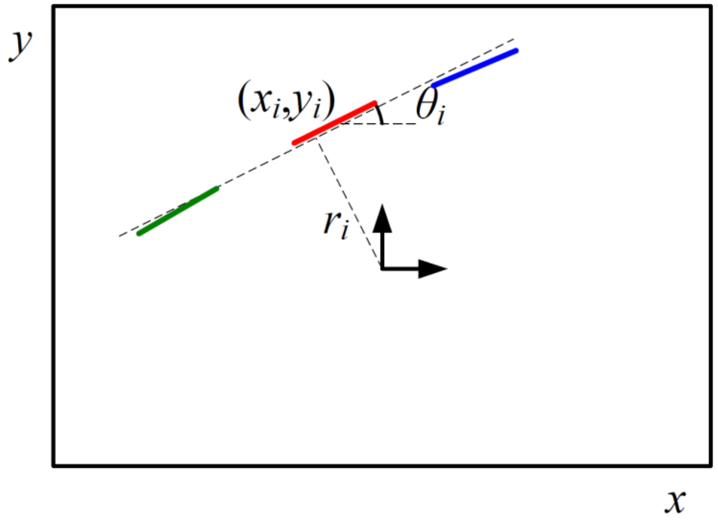
given that $a \neq 0$.

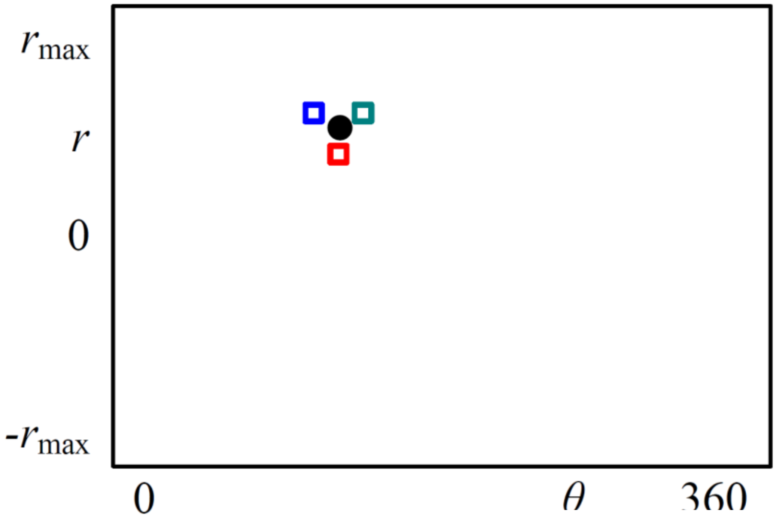
https://en.wikipedia.org/wiki/List_of_trigonometric_identities

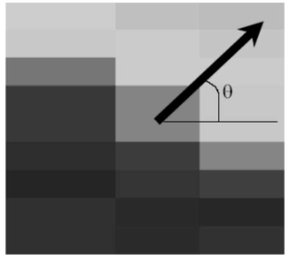
Line Detection

- Oriented Hough Transform
  - Use gradient orientation as theta

$$r_i(\theta) = x_i \cos\theta + y_i \sin\theta$$







$$\theta = atan2(\frac{\partial I}{\partial y}, \frac{\partial I}{\partial x})$$

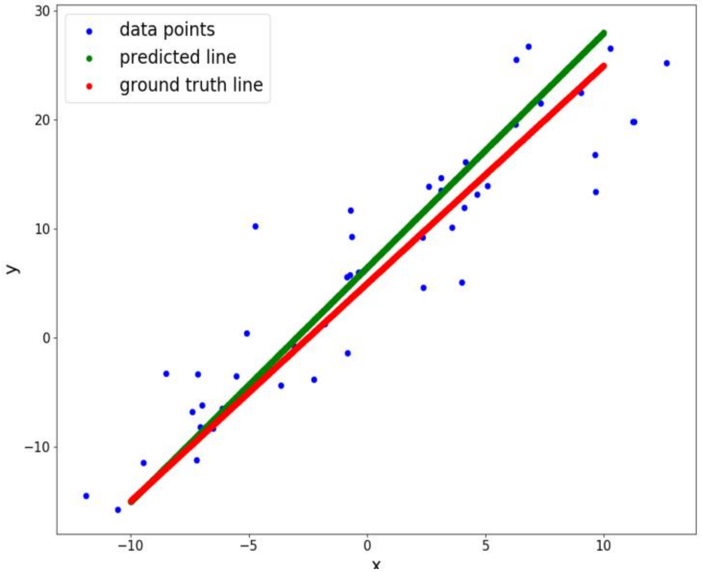
Orientation

$$\hat{\mathbf{n}}_i = (\cos\theta_i, \sin\theta_i)$$

$$r_i = \hat{\mathbf{n}}_i \cdot \mathbf{x}_i$$

Line Detection

- Random Sample Consensus (RANSAC)



- Sample two edge points
- Estimate the line parameter (θ, d)

$$x\cos\theta + y\sin\theta + d = 0$$

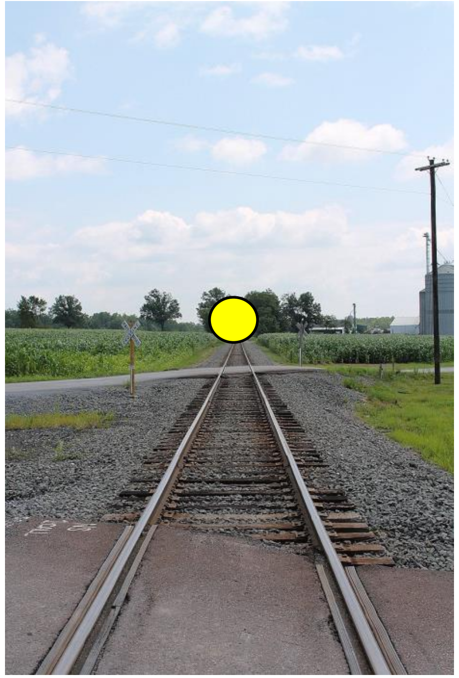
- Find how many edgels obey it

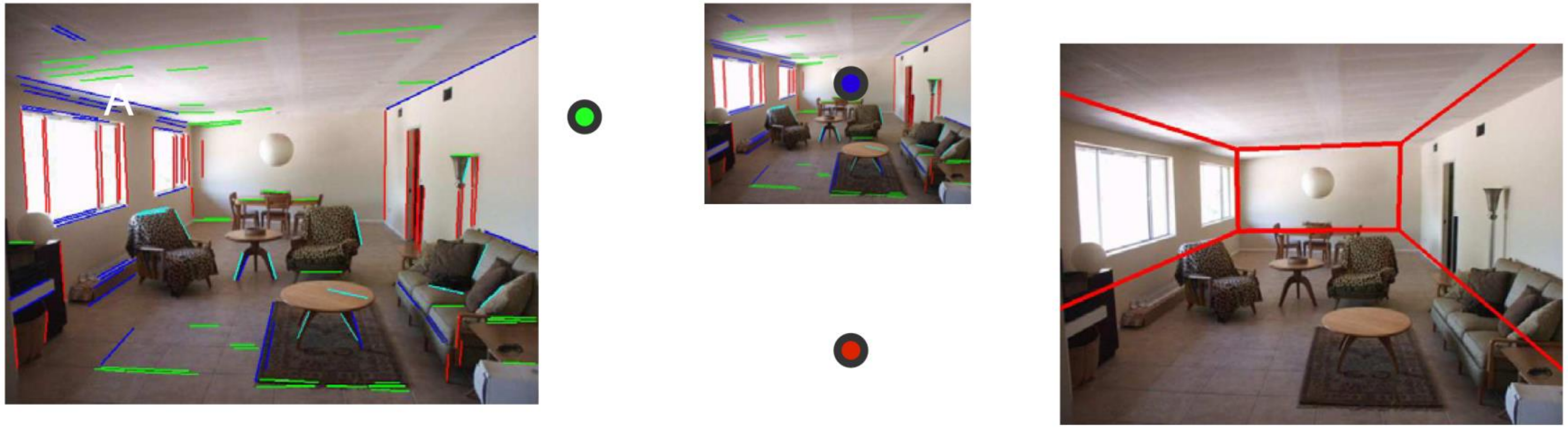
RANSAC Algorithm {
1. Selects N data items as random
2. Estimates parameter $\vec{x}$
3. Finds how many data items (of M) fit the model with parameter vector $\vec{x}$ within a user given tolerance. Call this K.
4. If K is big enough, accept fit and exit with success.
5. Repeat step 1 until 4 (as Ltimes)
6. Algorithm will be exit with fail

}

Vanishing Points

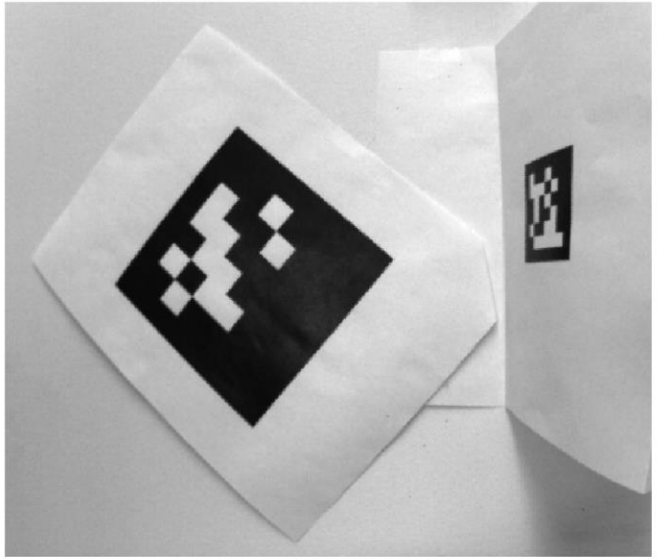
- Parallel lines in 3D converge in 2D images due to perspective projection

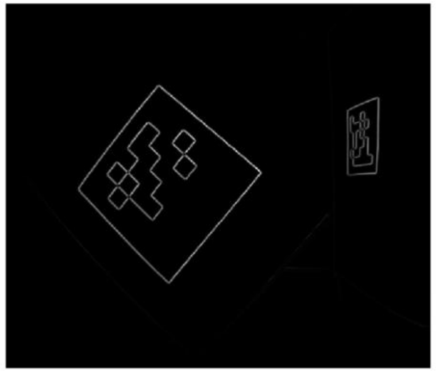




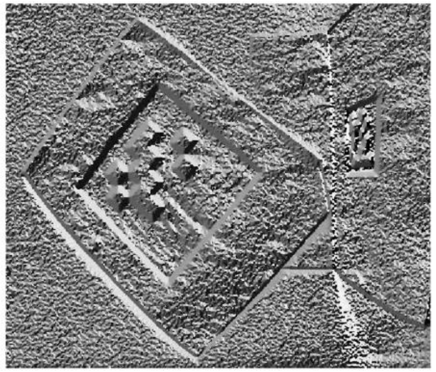
Recovering the Spatial Layout of Cluttered Rooms. Hedau et al., ICCV'09

Application: AprilTag Detection

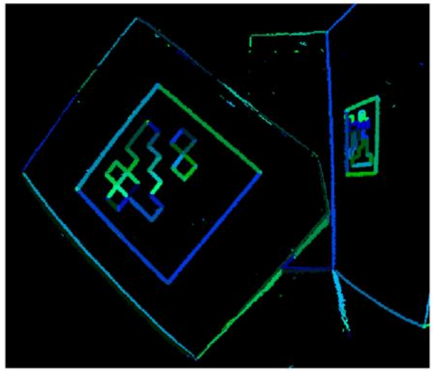




Gradient magnitudes

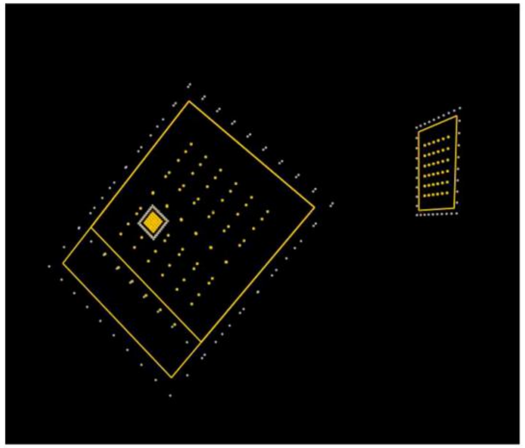


Gradient directions



Clustering

Line segments for cluster components



Quad detection

AprilTag: A robust and flexible visual fiducial system. Edwin Olson. ICRA, 2011

Further Reading

- Section 7.2, 7.4, Computer Vision, Richard Szeliski
- J. Canny. A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8, No. 6, Nov 1986.
- Pablo Arbelaez, Charless Fowlkes, Jitendra Malik. Contour Detection and Hierarchical Image Segmentation. TPAMI’10