Edge / Boundary Detection

Szeliski 4.2

Computer Vision

James Hays

1

Edge detection

- **Goal:** Identify sudden changes (discontinuities) in an image
  - Intuitively, most semantic and shape information from the image can be encoded in the edges
  - More compact than pixels

- **Ideal:** artist's line drawing (but artist is also using object-level knowledge)
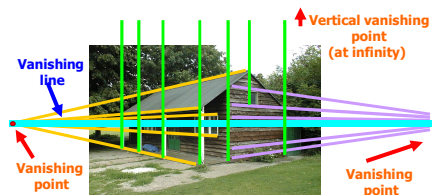
Source: D. Lowe

2

Why do we care about edges?
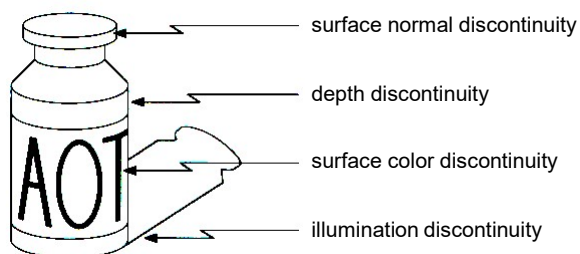
- Extract information, recognize objects

- Recover geometry and viewpoint

3

Origin of Edges

surface normal discontinuity

depth discontinuity

surface color discontinuity

illumination discontinuity

- Edges are caused by a variety of factors

Source: Steve Seitz

4

Closeup of edges

Source: D. Hoiem

5

Closeup of edges

Source: D. Hoiem

6

Closeup of edges

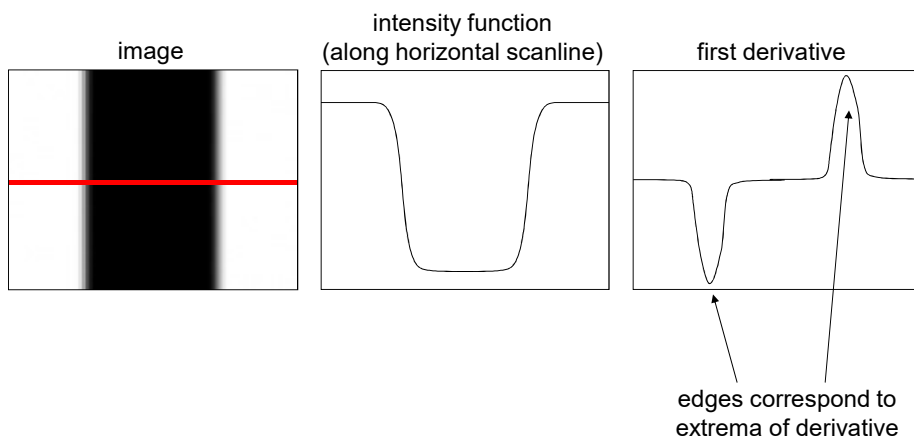Source: D. Hoiem

7

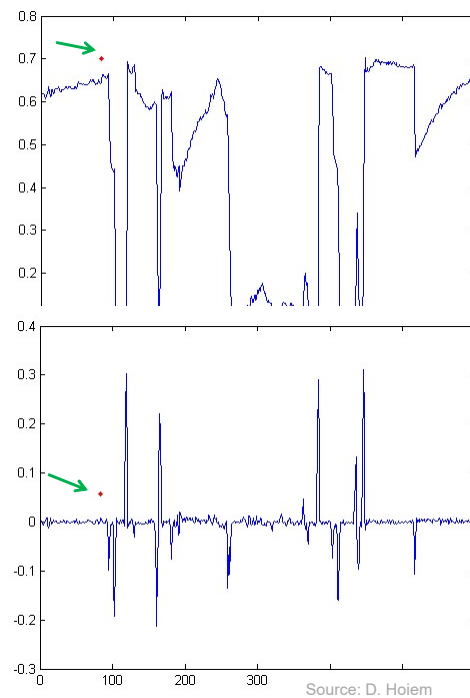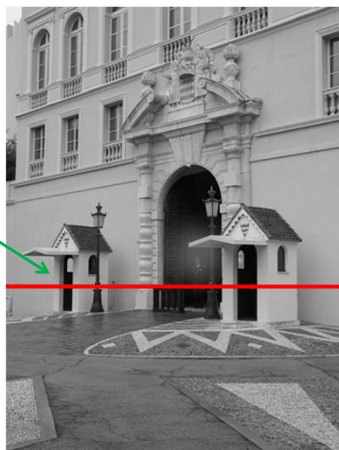## Closeup of edges

Source: D. Hoiem

8

Characterizing edges

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

image | intensity function (along horizontal scanline) | first derivative

edges correspond to extrema of derivative

9

Intensity profile

Source: D. Hoiem
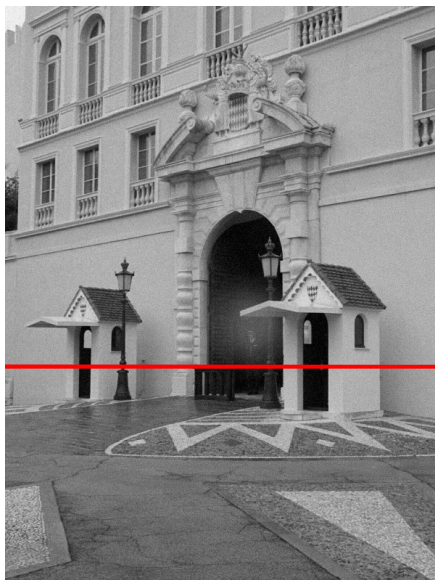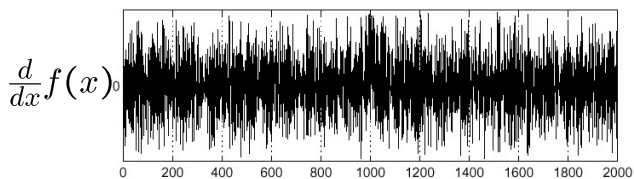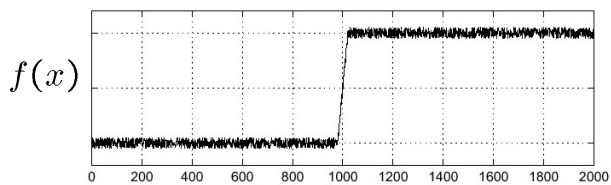
10

With a little Gaussian noise

Gradient
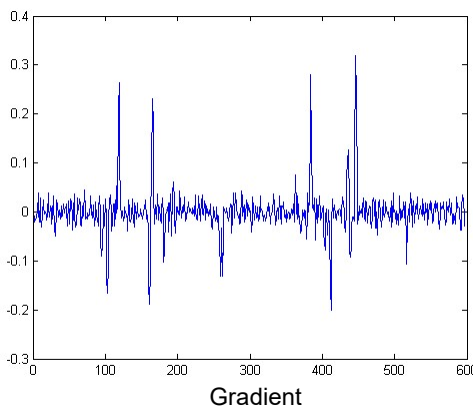
Source: D. Hoiem

11

Effects of noise

- Consider a single row or column of the image
  - Plotting intensity as a function of position gives a signal

$f(x)$

$\frac{d}{dx}f(x)$

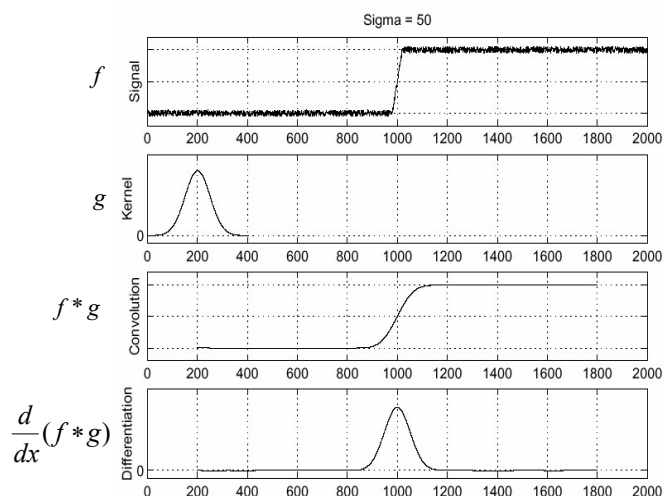Where is the edge?

Source: S. Seitz

12

Effects of noise

- Difference filters respond strongly to noise
  - Image noise results in pixels that look very different from their neighbors
  - Generally, the larger the noise the stronger the response
- What can we do about it?

Source: D. Forsyth

13

Solution: smooth first

Sigma = 50

f

g

$f * g$

$\dfrac{d}{dx}(f * g)$

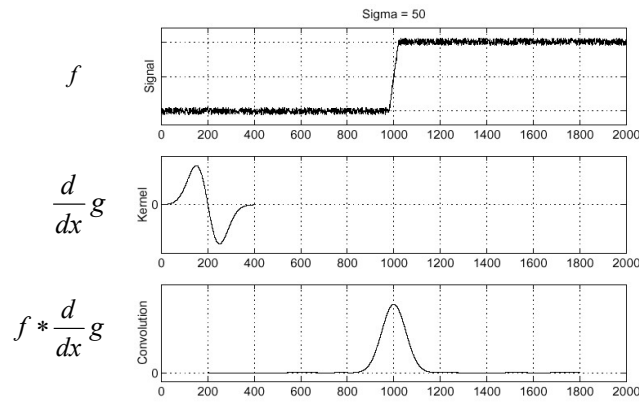- To find edges, look for peaks in $\dfrac{d}{dx}(f * g)$

Source: S. Seitz

14

Derivative theorem of convolution

- Differentiation is convolution, and convolution is associative:
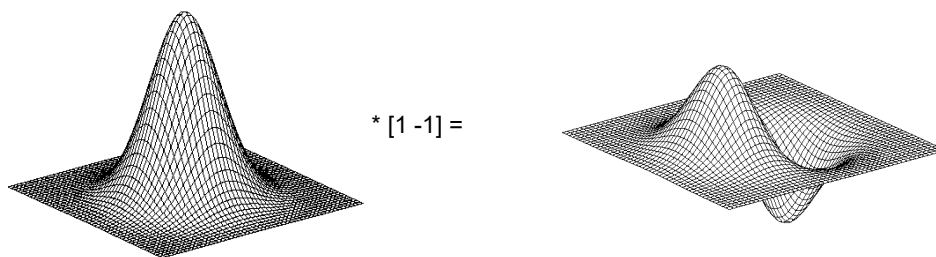
$$\frac{d}{dx}(f*g) = f*\frac{d}{dx}g$$

- This saves us one operation:

f

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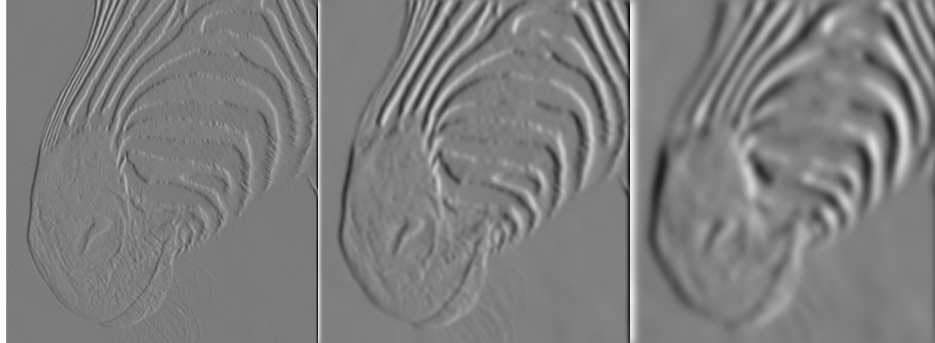
$f*\frac{d}{dx}g$

Source: S. Seitz

15

Derivative of Gaussian filter

* [1 -1] =

16

Tradeoff between smoothing and localization

| 1 pixel | 3 pixels | 7 pixels |

- Smoothed derivative removes noise, but blurs edge. Also finds edges at different "scales".

Source: D. Forsyth

17

Implementation issues

- The gradient magnitude is large along a thick "trail" or "ridge," so how do we identify the actual edge points?
- How do we link the edge points to form curves?

Source: D. Forsyth

18

Designing an edge detector

- Criteria for a good edge detector:
  - **Good detection:** the optimal detector should find all real edges, ignoring noise or other artifacts
  - **Good localization**
    - the edges detected must be as close as possible to the true edges
    - the detector must return one point only for each true edge point
- Cues of edge detection
  - Differences in color, intensity, or texture across the boundary
  - Continuity and closure
  - High-level knowledge

Source: L. Fei-Fei

19

Canny edge detector

- This is probably the most widely used edge detector in computer vision
- Theoretical model: step-edges corrupted by additive Gaussian noise
- Canny has shown that the first derivative of the Gaussian closely approximates the operator that optimizes the product of *signal-to-noise ratio* and localization

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

22,000 citations!

Source: L. Fei-Fei

20

Note about Matlab's Canny detector

- Small errors in implementation
  - Gaussian function not properly normalized
  - First filters with a Gaussian, then a difference of Gaussian (equivalent to filtering with a larger Gaussian and taking difference)
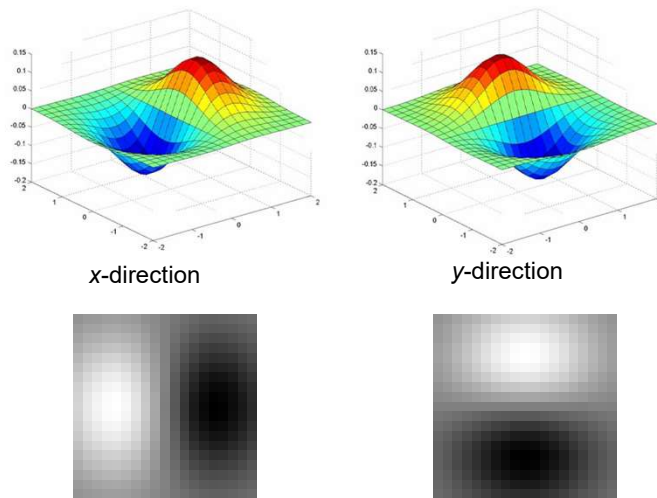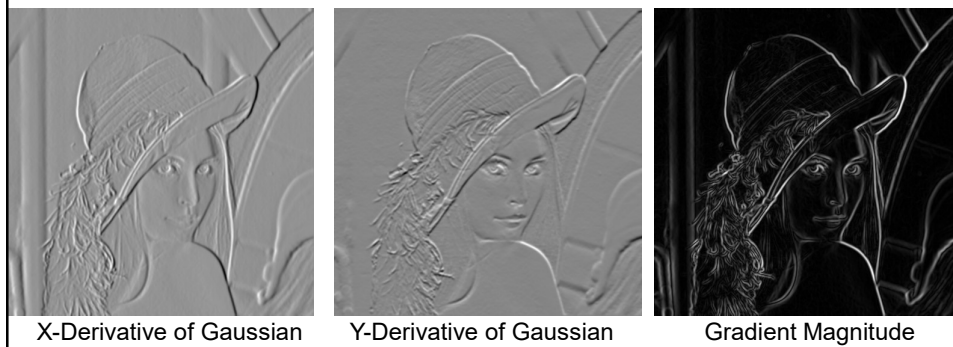
21

Example

original image (Lena)
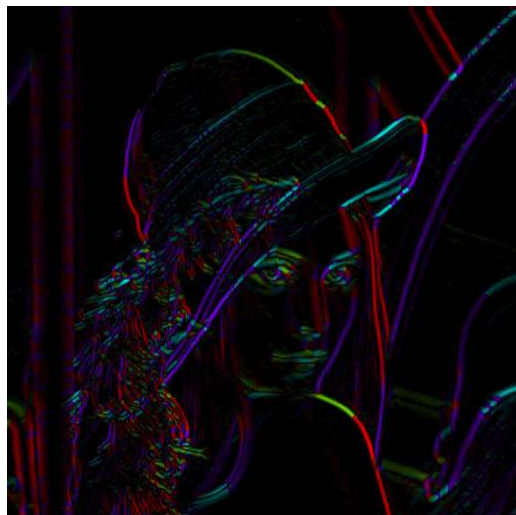
22

Derivative of Gaussian filter

x-direction   *y*-direction

23

Compute Gradients (DoG)

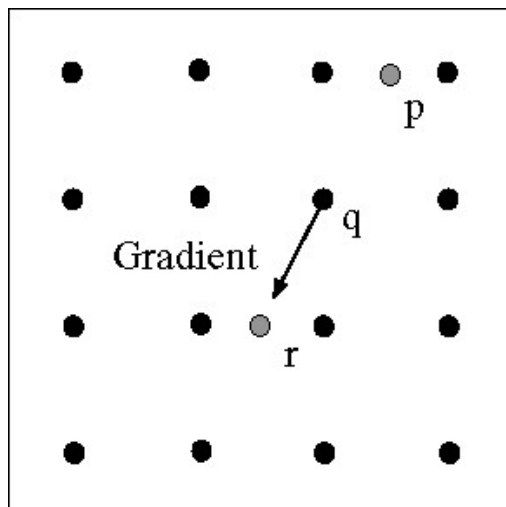X-Derivative of Gaussian   Y-Derivative of Gaussian   Gradient Magnitude

24

Get Orientation at Each Pixel

- Threshold at minimum level
- Get orientation
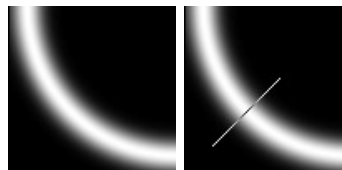
theta = atan2(gy, gx)

25

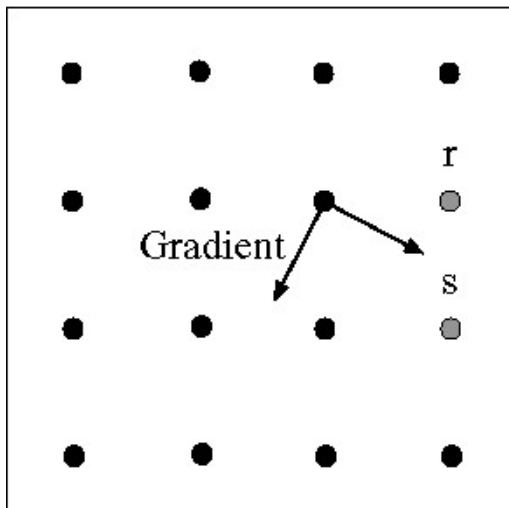# Non-maximum suppression for each orientation

At q, we have a maximum if the value is larger than those at both p and at r. Interpolate to get these values.
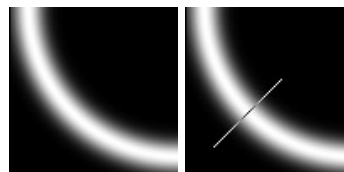
Gradient

p

q

r

Source: D. Forsyth

26

Edge linking

Assume the marked point is an edge point. Then we construct the tangent to the edge curve (which is normal to the gradient at that point) and use this to predict the next points (here either r or s).
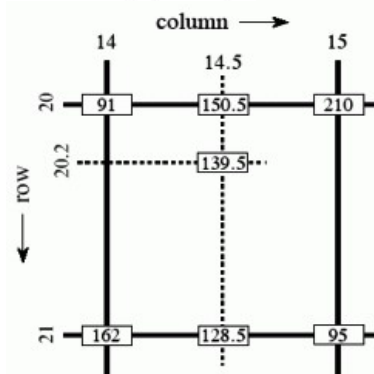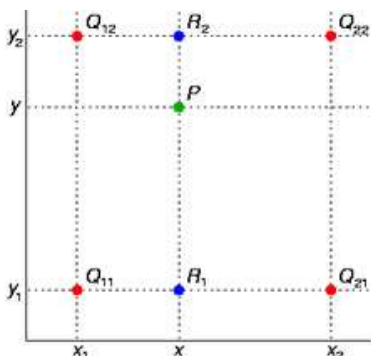
Source: D. Forsyth

27

Sidebar: Bilinear Interpolation

$$f(x,y) \approx \begin{bmatrix} 1-x & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}.$$
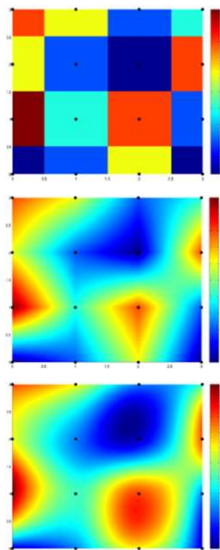

http://en.wikipedia.org/wiki/Bilinear_interpolation

28

Sidebar: Interpolation options

- imx2 = imresize(im, 2, interpolation_type)

- 'nearest'
  - Copy value from nearest known
  - Very fast but creates blocky edges

- 'bilinear'
  - Weighted average from four nearest known pixels
  - Fast and reasonable results

- 'bicubic' (default)
  - Non-linear smoothing over larger area (4x4)
  - Slower, visually appealing, may create negative pixel values

Examples from http://en.wikipedia.org/wiki/Bicubic_interpolation
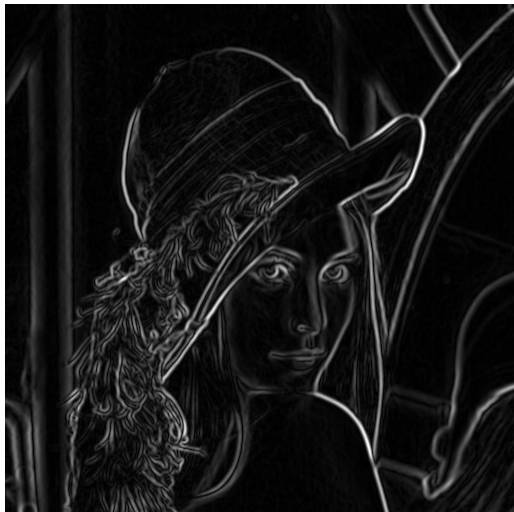
29

Before Non-max Suppression
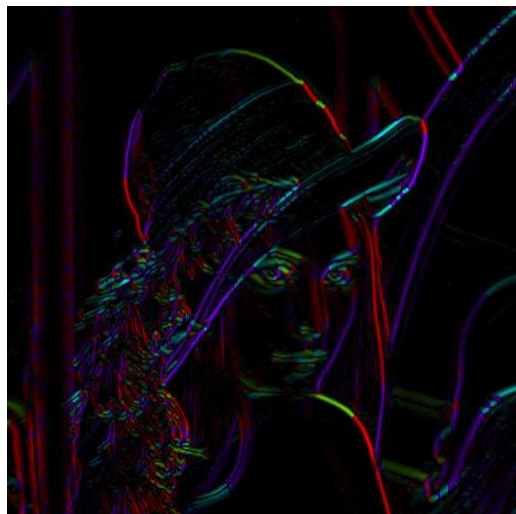
30

After non-max suppression

31

Before Non-max Suppression

32

After non-max suppression

33
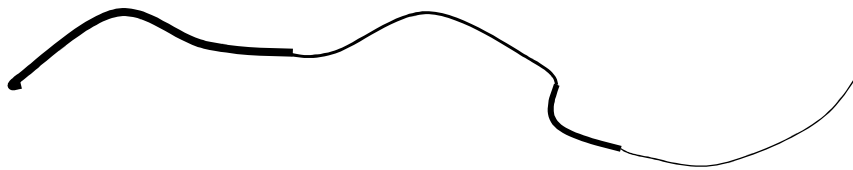
Hysteresis thresholding

- Threshold at low/high levels to get weak/strong edge pixels
- Do connected components, starting from strong edge pixels

34

Hysteresis thresholding

- Check that maximum value of gradient value is sufficiently large
  - drop-outs? use **hysteresis**
    - use a high threshold to start edge curves and a low threshold to continue them.

Source: S. Seitz

35

Final Canny Edges

36

Canny edge detector

1. Filter image with x, y derivatives of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression:
   – Thin multi-pixel wide "ridges" down to single pixel width
4. Thresholding and linking (hysteresis):
   – Define two thresholds: low and high
   – Use the high threshold to start edge curves and the low threshold to continue them

- MATLAB: edge(image, 'canny')

Source: D. Lowe, L. Fei-Fei

37

Effect of σ (Gaussian kernel spread/size)

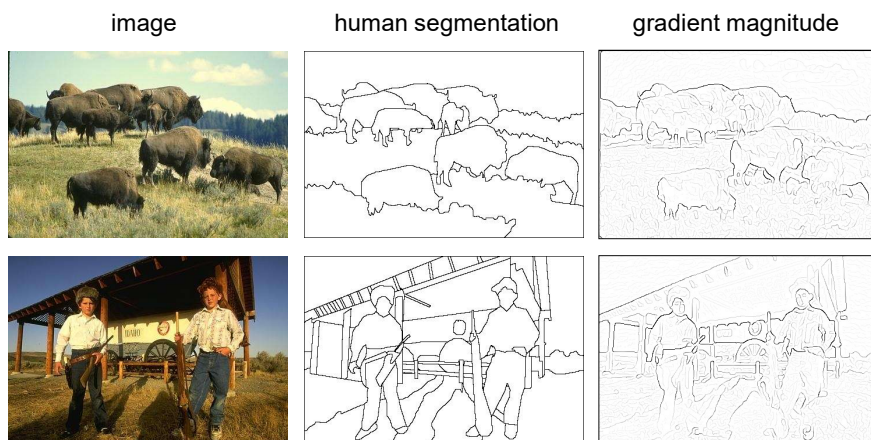| original | Canny with $\sigma = 1$ | Canny with $\sigma = 2$ |

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

Source: S. Seitz

38

19

Where do humans see boundaries?

image   human segmentation   gradient magnitude

- Berkeley segmentation database:
  http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/
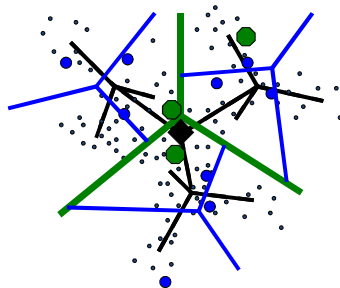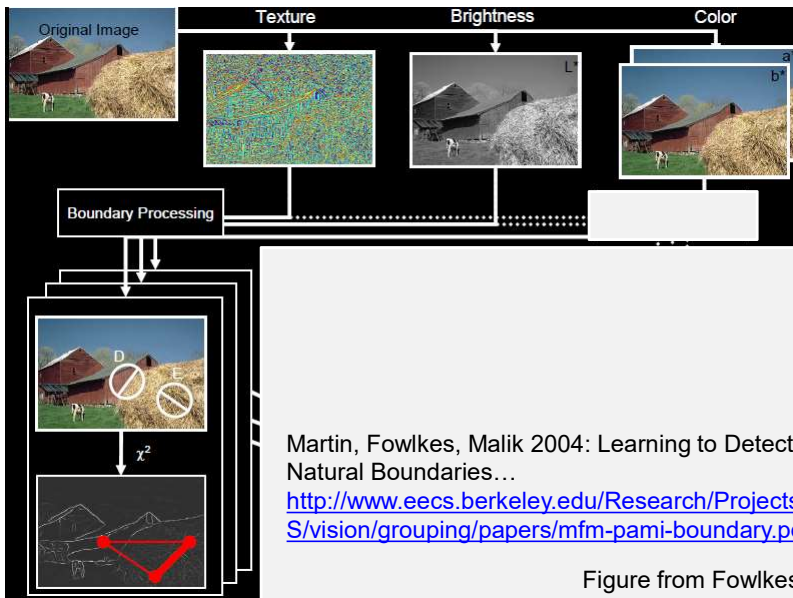
39

Building Visual Dictionaries

1. Sample patches from a database
   - E.g., 128 dimensional SIFT vectors

2. Cluster the patches
   - Cluster centers are the dictionary

3. Assign a codeword (number) to each new patch, according to the nearest cluster

40

pB boundary detector

Martin, Fowlkes, Malik 2004: Learning to Detect Natural Boundaries…
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/papers/mfm-pami-boundary.pdf

Figure from Fowlkes

41

pB Boundary Detector
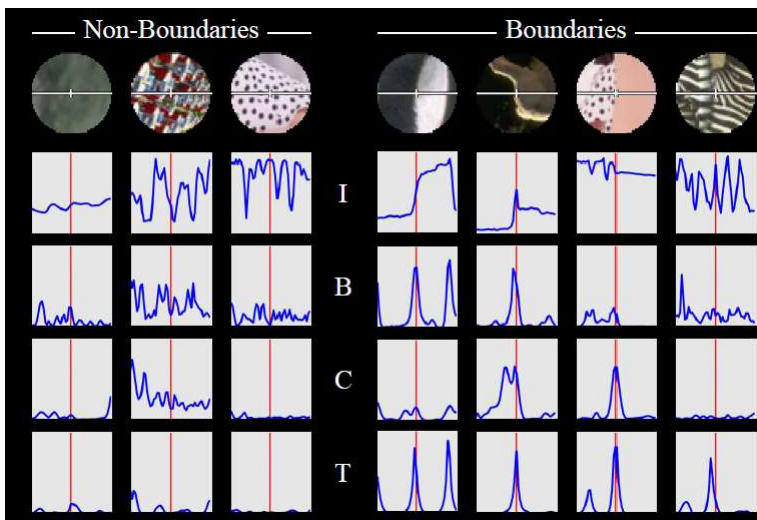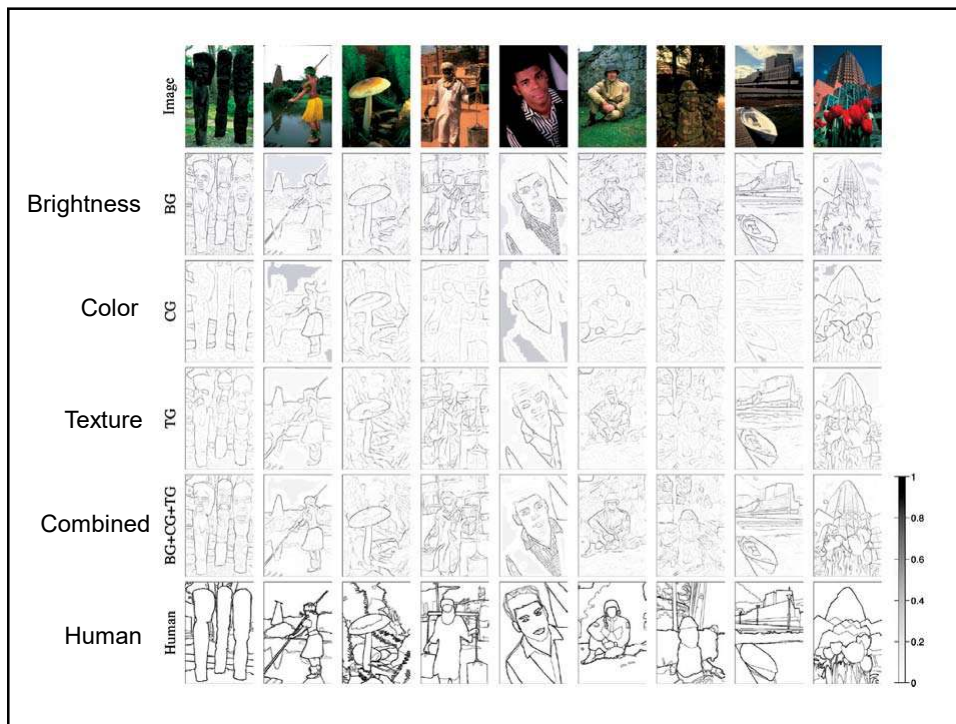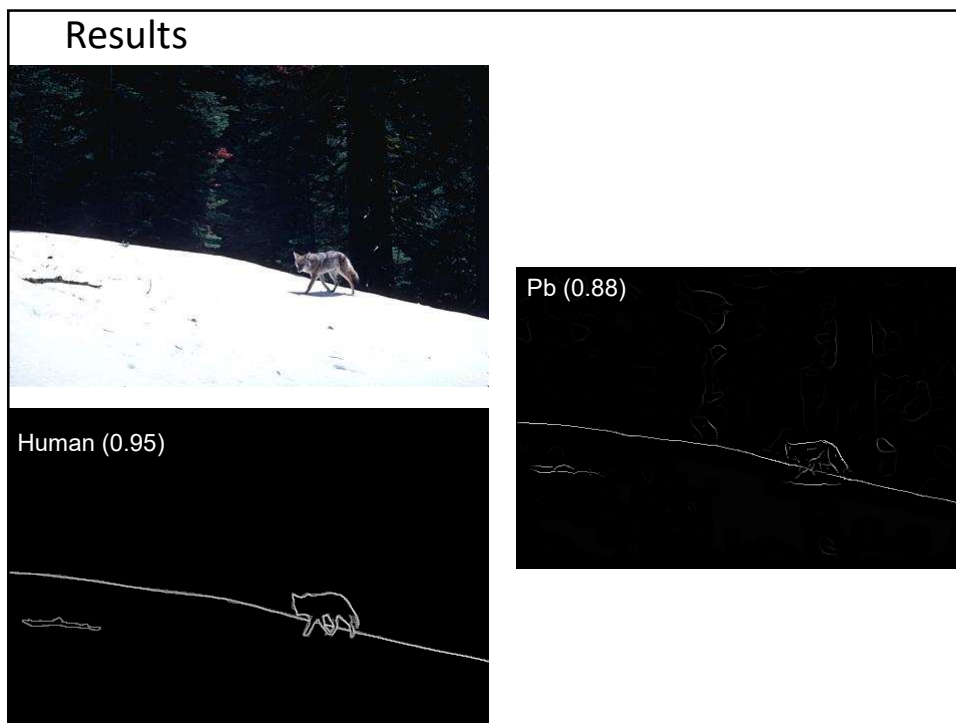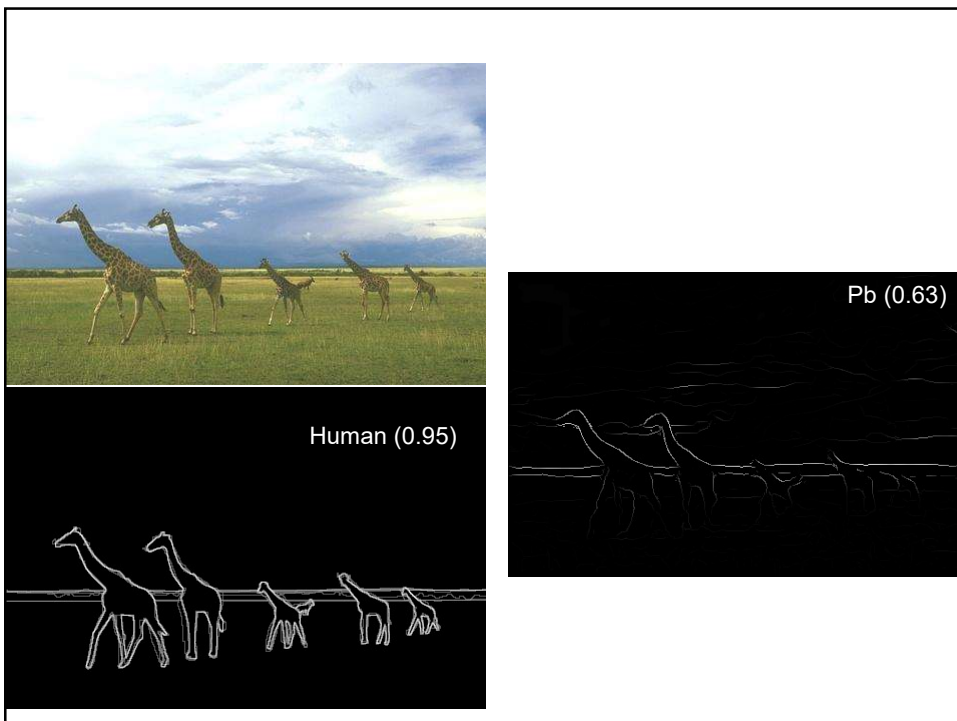

Figure from Fowlkes

42

43

Results

44

Results

45

46

Human (0.90)

Pb (0.35)

For more:
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/bench/html/108082-color.html

47

Global pB boundary detector

Extract Pb

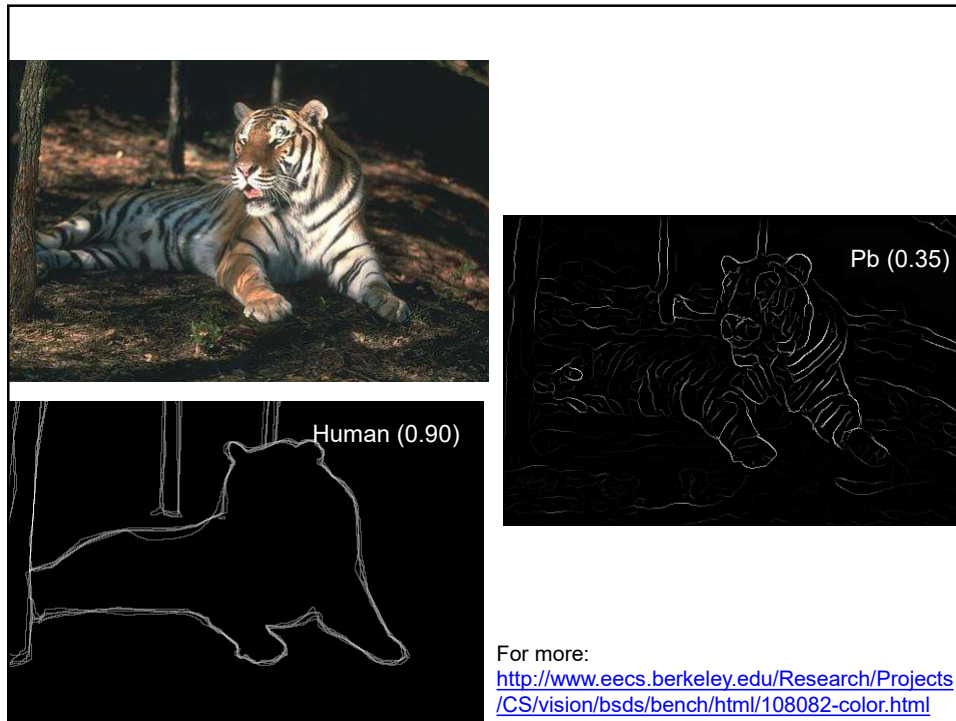Compute Eigenvectors
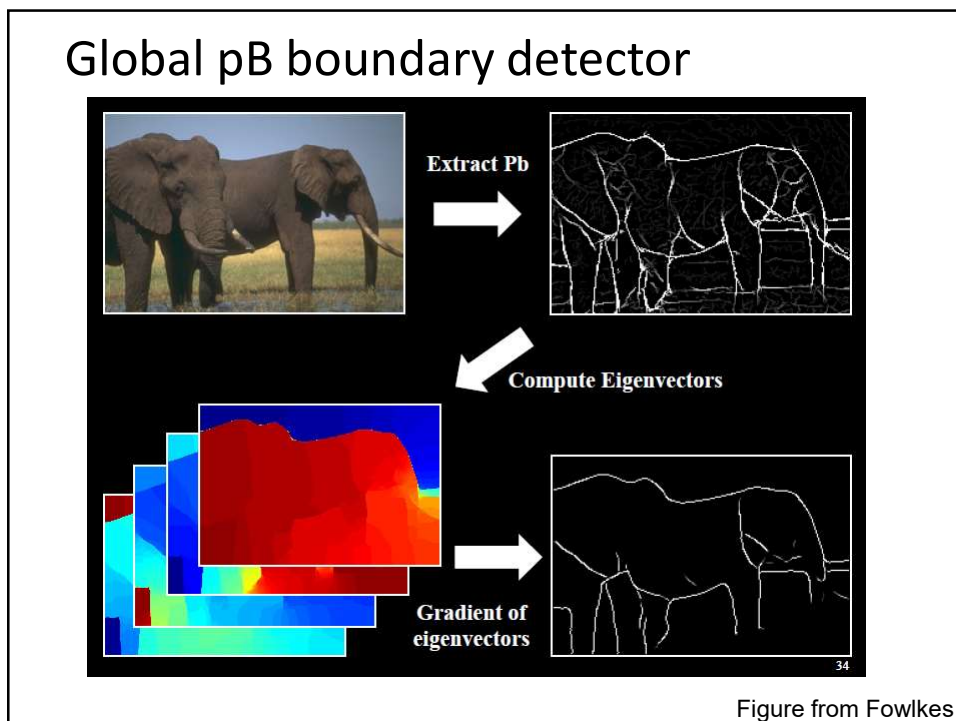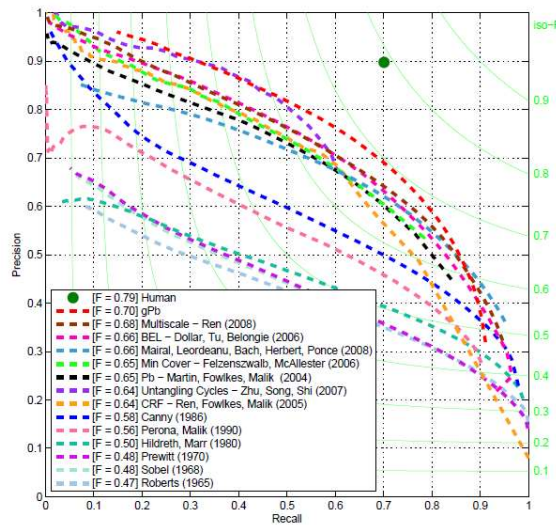
Gradient of eigenvectors

Figure from Fowlkes

48

45 years of boundary detection

Source: Arbelaez, Maire, Fowlkes, and Malik. TPAMI 2011 (pdf)

49

State of edge detection

- Local edge detection works well
  - But many false positives from illumination and texture edges
- Some methods to take into account longer contours, but could probably do better
- Modern methods that actually "learn" from data.
- Poor use of object and high-level information

50

Style and abstraction in portrait sketching, Berger et al. SIGGRAPH 2013
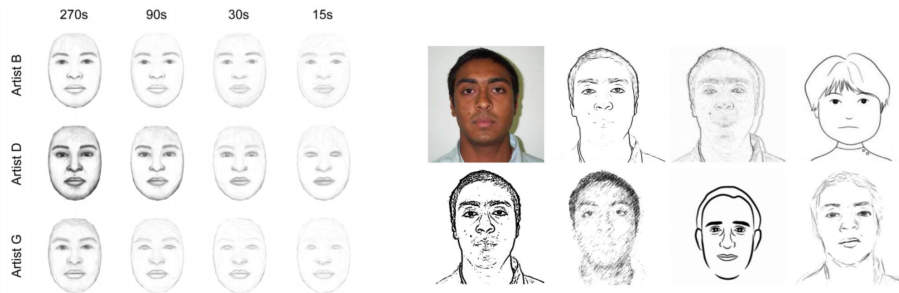
- Learn from artist's strokes so that edges are more likely in certain parts of the face.

51