Edge detection

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

Source: D. Lowe
Origin of edges

Edges are caused by a variety of factors:

- depth discontinuity
- surface color discontinuity
- illumination discontinuity
- surface normal discontinuity

Source: Steve Seitz
Edge detection

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

Source: S. 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 the above as convolution, what would be the associated filter?

Source: K. Grauman
Partial derivatives of an image

\[
\frac{\partial f(x, y)}{\partial x}
\]

-1 1

\[
\frac{\partial f(x, y)}{\partial y}
\]

\[
\begin{pmatrix}
-1 & 1 \\
1 & -1
\end{pmatrix}
\]

Which shows changes with respect to x?

Source: S. Lazebnik
Finite difference filters

Other approximations of derivative filters exist:

Prewitt:  \[ M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} ; \\ M_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \]

Sobel:  \[ M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} ; \\ M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \]

Roberts:  \[ M_x = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} ; \\ M_y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \]

Source: K. Grauman
The gradient points in the direction of most rapid increase in intensity.

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

The gradient direction 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
\[ \|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \]
Image gradient
Application: 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.

Effects of noise

Consider a single row or column of the image

Where is the edge?

Source: S. Seitz
Solution: smooth first

• To find edges, look for peaks in \( \frac{d}{dx}(f \ast g) \)

Source: S. Seitz
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:

Source: S. Seitz
Derivative of Gaussian filters

Which one finds horizontal/vertical edges?

x-direction

y-direction
Derivative of Gaussian filters

Are these filters separable?
Scale of Gaussian derivative filter

1 pixel  
3 pixels  
7 pixels

Smoothed derivative removes noise, but blurs edge. Also finds edges at different “scales”

Source: D. Forsyth
Review: Smoothing vs. derivative filters

Smoothing filters

- Gaussian: remove "high-frequency" components; "low-pass" filter
  
- What should the values sum to?
  - One: constant regions are not affected by the filter

Derivative filters

- Derivatives of Gaussian

- What should the values sum to?
  - Zero: no response in constant regions
  - High absolute value at points of high contrast

Source: S. Lazebnik
The Canny edge detector

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

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

The Canny edge detector

Slide credit: Steve Seitz
The Canny edge detector

norm of the gradient
The Canny edge detector

thresholding
The Canny edge detector

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

thresholding
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
The Canny edge detector

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

thinning
(non-maximum suppression)
Hysteresis thresholding

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

Source: Steve Seitz
Hysteresis thresholding

- Original image
- High threshold (strong edges)
- Low threshold (weak edges)
- Hysteresis threshold

Source: L. Fei-Fei
Recap: Canny edge detector

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

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

Image gradients vs. meaningful contours

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

Training data

Input images

Ground truth

Output

P. Dollar and L. Zitnick, Structured forests for fast edge detection, ICCV 2013
Credit:
Slide set developed by S. Lazebnik, U. Illinois at Urbana-Champaign