**Edge detection**

Aykut Erdem  
Dept. of Computer Engineering  
Hacettepe University

---

**Origin of edges**

- Edges are caused by a variety of factors:
  - depth discontinuity
  - surface color discontinuity
  - illumination discontinuity
  - surface normal discontinuity

---

**Characterizing edges**

- An edge is a place of rapid change in the image intensity function
  - Edges correspond to extrema of derivative
  - **Convert a 2D image into a set of curves**
    - Extracts salient features of the scene
    - More compact than pixels

---

Slide credit: N. Snavely

Slide credit: S. Seitz

Slide credit: A. Efros
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 above as convolution, what would be the associated filter?

Finite difference filters

- Other approximations of derivative filters exist:

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

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

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

Partial derivatives of an image

Which shows changes with respect to \( x \)?

\[
\frac{\partial f(x,y)}{\partial x} \approx \begin{bmatrix} -1 & 1 \end{bmatrix}
\]

\[
\frac{\partial f(x,y)}{\partial y} \approx \begin{bmatrix} -1 \\ 1 \end{bmatrix}
\]

Image gradient

- The gradient of an image: 
  \[
  \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \end{bmatrix}
  \]

  \[
  \nabla f = \begin{bmatrix} 0, 0 \end{bmatrix}
  \]

  \[
  \nabla f = \begin{bmatrix} 0, \frac{\partial f}{\partial y} \end{bmatrix}
  \]

  \[
  \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x}, 0 \end{bmatrix}
  \]

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

The edge strength is given by the gradient magnitude
\[
\| \nabla f \| = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2}
\]
Example: Sobel Operator

Original  Magnitude  Orientation

Slide credit: A. Farhadi

Effects of noise

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

\[ f(x) \]

Where is the edge?

Solution: smooth first

\[ f \]

\[ g \]

\[ f \ast g \]

\[ \frac{d}{dx}(f \ast g) \]

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

Slide credit: S. Seitz

Derivative theorem of convolution

• Differentiation is convolution, and convolution is associative:
  \( \frac{d}{dx}(f \ast g) = f \ast \frac{d}{dx}g \)

• This saves us one operation:

\[ f \]

\[ \frac{d}{dx}g \]

\[ f \ast \frac{d}{dx}g \]

Slide credit: S. Seitz
Derivative of Gaussian filter

- Are these filters separable?

Review: Smoothing vs. derivative 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
  - Can the values of a derivative filter be negative?
  - What should the values sum to?
    - Zero: no response in constant regions
    - High absolute value at points of high contrast

Scale of Gaussian derivative filter

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

original image

Slide credit: Steve Seitz

17

norm of the gradient

Slide credit: Steve Seitz

18

thresholding

Slide credit: Steve Seitz

19

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

thresholding

Slide credit: Steve Seitz

20
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.
**Recap: 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');`


Slide credit: A. Efros