Edge detection

Winter in Kraków photographed by Marcin Ryczek
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

- **Ideal:** artist’s line drawing

- **Reality:**

![Ideal Image](image1)

![Reality Image](image2)
Edge detection

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

![Diagram showing edge detection](image)

- Edges correspond to extrema of the first derivative.
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

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

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

The gradient of an image: \[ \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x}, 0 \\ 0, \frac{\partial f}{\partial y} \end{bmatrix} \]

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

The edge strength is given by the gradient magnitude

\[ \| \nabla f \| = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2} \]

Source: Steve Seitz
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

P. Perez, M. Gangnet, A. Blake, Poisson Image Editing, SIGGRAPH 2003
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?
Derivative of Gaussian filters

Are these filters separable?
Recall: Separability of the Gaussian filter

\[ G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp \left( -\frac{x^2 + y^2}{2\sigma^2} \right) \]

\[ = \left( \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{x^2}{2\sigma^2} \right) \right) \left( \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{y^2}{2\sigma^2} \right) \right) \]

The 2D Gaussian can be expressed as the product of two functions, one a function of \( x \) and the other a function of \( y \)

In this case, the two functions are the (identical) 1D Gaussian

Source: D. Lowe
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
- 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
Building an edge detector

original image

final output
Building an edge detector

norm of the gradient
Building an edge detector

Thresholded norm of the gradient

How to turn these thick regions of the gradient into curves?
Non-maximum suppression

For each location $q$ above threshold, check that the gradient magnitude is higher than at neighbors $p$ and $r$ along the direction of the gradient.

- May need to interpolate to get the magnitudes at $p$ and $r$. 
Non-maximum suppression

Another problem: pixels along this edge didn’t survive the thresholding
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
Data-driven edge detection

P. Dollar and L. Zitnick, Structured forests for fast edge detection, ICCV 2013
Data-driven edge detection

S. Xie and Z. Tu, Holistically-nested edge detection, ICCV 2015