Detecting Edges in Image

- Sobel Edge Detector

Image \( I \)

\[
\begin{bmatrix}
1 & 0 & -1 \\
2 & 0 & -2 \\
1 & 0 & -1
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & -2 & -1
\end{bmatrix}
\]

\[
\frac{d}{dx} I \\
\frac{d}{dy} I
\]

\[
\left(\frac{d}{dx} I \right)^2 + \left(\frac{d}{dy} I \right)^2
\]

Threshold

Edges
Sobel Edge Detector

\[
\Delta = \sqrt{\left(\frac{d}{dx} I\right)^2 + \left(\frac{d}{dy} I\right)^2}
\]

\[
\Delta \geq \text{Threshold} = 100
\]
Marr and Hildreth Edge Operator

- Smooth by Gaussian
  \[ S = G_\sigma \ast I \]
  \[ G_\sigma = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2 + y^2}{2\sigma^2}} \]

- Use Laplacian to find derivatives
  \[ \Delta^2 S = \frac{\partial^2}{\partial x^2} S + \frac{\partial^2}{\partial y^2} S \]

\[ \Delta^2 S = \Delta^2 (G_\sigma \ast I) = \Delta^2 G_\sigma \ast I \]

\[ \Delta^2 G_\sigma = -\frac{1}{\sqrt{2\pi\sigma^3}} \left( 2 - \frac{x^2 + y^2}{\sigma^2} \right) e^{-\frac{x^2 + y^2}{2\sigma^2}} \]
Marr and Hildreth Edge Operator

\[ \Delta^2 G_\sigma = - \frac{1}{\sqrt{2\pi}\sigma^3} \left( 2 - \frac{x^2 + y^2}{\sigma^2} \right) e^{\frac{x^2 + y^2}{2\sigma^2}} \]

\[
\begin{array}{cccccc}
0.0008 & 0.0066 & 0.0215 & 0.031 & 0.0215 & 0.0066 \\
0.0066 & 0.0438 & 0.0982 & 0.108 & 0.0982 & 0.0438 \\
0.0215 & 0.0982 & 0 & -0.242 & 0 & 0.0982 \\
0.031 & 0.108 & -0.242 & -0.7979 & -0.242 & 0.108 \\
0.0215 & 0.0982 & 0 & -0.242 & 0 & 0.0982 \\
0.0066 & 0.0438 & 0.0982 & 0.108 & 0.0982 & 0.0438 \\
0.0008 & 0.0066 & 0.0215 & 0.031 & 0.0215 & 0.0066
\end{array}
\]

Marr and Hildreth Edge Operator

Image \( I \) \( \rightarrow \) \( \Delta^2 G_\sigma \) \( \rightarrow \) \( \Delta^2 G_\sigma \ast I \) \( \rightarrow \) \( \text{Zero Crossings} \) \( \rightarrow \) \( \text{Detection} \) \( \rightarrow \) \( \text{Edge Image} \)

\[ \Delta^2 G_\sigma \ast I \]

\[ \text{Zero Crossings} \]
Quality of an Edge Detector

- Robustness to Noise
- Localization
- Too Many/Too less Responses

True Edge

Poor robustness to noise

Poor localization

Too many responses
Canny Edge Detector

- Criterion 1: Good Detection: The optimal detector must minimize the probability of false positives as well as false negatives.

- Criterion 2: Good Localization: The edges detected must be as close as possible to the true edges.

- Single Response Constraint: The detector must return one point only for each edge point.

Canny Edge Detector

- Difficult to find closed-form solutions.

Figure 4.15 A comparison between the Canny operator and the first derivative of a Gaussian.
Canny Edge Detector

- Convolution with derivative of Gaussian
- Non-maximum Suppression
- Hysteresis Thresholding

Canny Edge Detector

- Smooth by Gaussian
  \[ S = G_\sigma * I \]
  \[ G_\sigma = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2+y^2}{2\sigma^2}} \]

- Compute \( x \) and \( y \) derivatives
  \[ \Delta S = \begin{bmatrix} \frac{\partial}{\partial x} S & \frac{\partial}{\partial y} S \end{bmatrix}^T = [S_x, S_y]^T \]

- Compute gradient magnitude and orientation
  \[ |\Delta S| = \sqrt{S_x^2 + S_y^2} \]
  \[ \theta = \tan^{-1} \frac{S_y}{S_x} \]
Canny Edge Operator

\[ \Delta S = \Delta(G_{\sigma} * I) = \Delta G_{\sigma} * I \]

\[ \Delta G_{\sigma} = \begin{bmatrix} \frac{\partial G_{\sigma}}{\partial x} & \frac{\partial G_{\sigma}}{\partial y} \end{bmatrix}^T \]

\[ \Delta S = \begin{bmatrix} \frac{\partial G_{\sigma} * I}{\partial x} & \frac{\partial G_{\sigma} * I}{\partial y} \end{bmatrix}^T \]
Canny Edge Detector

\[ |\Delta S| = \sqrt{S_x^2 + S_y^2} \]

We can do this by looking for a maximum along a slice normal to the curve (non-maximum suppression). These points should form a curve. There are then two algorithmic issues: at which point is the maximum, and where is the next one?

Non-Maximum Suppression

We wish to mark points along the curve where the magnitude is biggest. We can do this by looking for a maximum along a slice normal to the curve (non-maximum suppression). These points should form a curve. There are then two algorithmic issues: at which point is the maximum, and where is the next one?
Non-Maximum Suppression

- Suppress the pixels in ‘Gradient Magnitude Image’ which are not local maximum

\[
M(x, y) = \begin{cases} 
|\Delta S|(x, y) & \text{if } |\Delta S|(x, y) > |\Delta S|(x', y') \\
& \text{and } |\Delta S|(x, y) > |\Delta S|(x'', y'') \\
0 & \text{otherwise}
\end{cases}
\]

(x', y') and (x'', y'') are the neighbors of (x, y) in |\Delta S|
along the direction normal to an edge

Non-Maximum Suppression

\[
\tan \theta = \frac{S_y}{S_x}
\]

0: -0.4142 < \tan \theta ≤ 0.4142
1: 0.4142 < \tan \theta < 2.4142
2: |\tan \theta| ≥ 2.4142
3: -2.4142 < \tan \theta ≤ -0.4142
Non-Maximum Suppression

\[ |\Delta s| = \sqrt{S_x^2 + S_y^2} \]

\( M \geq \text{Threshold} = 25 \)

Hysteresis Thresholding
Hysteresis Thresholding

- If the gradient at a pixel is above ‘High’, declare it an ‘edge pixel’
- If the gradient at a pixel is below ‘Low’, declare it a ‘non-edge-pixel’
- If the gradient at a pixel is between ‘Low’ and ‘High’ then declare it an ‘edge pixel’ if and only if it is connected to an ‘edge pixel’ directly or via pixels between ‘Low’ and ‘High’
Finding Connected Components

- Scan the binary image left to right top to bottom
- If there is an unlabeled pixel $p$ with a value of ‘1’
  - assign a new label to it
  - Recursively check the neighbors of pixel $p$ and assign the same label if they are unlabeled with a value of ‘1’.
- Stop when all the pixels with value ‘1’ have been labeled.

Connectedness
Suggested Reading

- Chapter 8, David A. Forsyth and Jean Ponce, "Computer Vision: A Modern Approach"
- Chapter 4, Emanuele Trucco, Alessandro Verri, "Introductory Techniques for 3-D Computer Vision"
- Chapter 2, Mubarak Shah, “Fundamentals of Computer Vision”