Outline

• Extracting useful information from Images
  – Edge Detection
    • Canny
Recap: Edge Detection

- The purpose of Edge Detection is to find jumps in the brightness function (of an image) and mark them.
Recap: Edge Detection / Input
Recap: Edge Detection / Output
Laplacian of Gaussian (LoG) and Canny Edge Detector

Marr and Hildreth Filtering, 1980. (LoG)
• Smooth Image with Gaussian Filter
• Applying the Laplacian for a Gaussian-filtered image can be done in one step of convolution.
• Find zero-crossings
• Find slope of zero-crossings
• Apply threshold to slope and mark edges

J. Canny. 1986. (Canny)
• Smooth Image with Gaussian filter
• Compute Derivative of filtered image
• Find Magnitude and Orientation of gradient
• Apply Non-max suppression
• Apply Thresholding (Hysteresis)
Laplacian of Gaussian (LoG) and Canny Edge Detector

Marr and Hildreth Filtering, 1980. (LoG)
• Smooth Image with Gaussian Filter
• Applying the Laplacian for a Gaussian-filtered image can be done in one step of convolution.
• Find zero-crossings
• Find slope of zero-crossings
• Apply threshold to slope and mark edges

J. Canny. 1986. (Canny)
• STEP 1: Smooth Image with Gaussian filter
• STEP 2: Compute Derivative of filtered image
• STEP 3: Find Magnitude and Orientation of gradient
• STEP 4: Apply Non-max suppression
• STEP 5: Apply Thresholding (Hysteresis)
Why Canny?

• Three main criteria
  – Good Detection: The ability to locate and mark all real edges
  – Good Localisation: Minimal distance between the detected edge and real edge
  – Clear Response: Only one response per edge
Why Canny?

- Canny’s initial point of departure was to use the first derivative of the Gaussian as an operator and determine its greatest value, rather than proceeding to the second derivative.
- This is slightly more accurate than LoG; however, it was actually arrived at by mathematical analysis as a close approximation to the optimal operator for the criteria
  - Low error rate
  - Good localization
Theory behind Canny Edge Detector

• Canny pointed out that it is difficult to design a general edge detector which is optimal;
Theory behind Canny Edge Detector

• Canny pointed out that it is difficult to design a general edge detector which is optimal;
• He assumed step edge model for his detector as a main starting point

• Here is the paper:
Step 1: Smoothing with Gaussian

Ex: if sigma = 1 pixel wide, then total window you want to use for smoothing is 7!
    if sigma = 2 pixel wide, then total window you want to use for smoothing is 13!
    ...

μ - 3σ  μ - 2σ  μ - σ  μ  μ + σ  μ + 2σ  μ + 3σ
Step 1: Smoothing with Gaussian

Ex: if sigma = 1 pixel wide, then total window you want to use for smoothing is 7!
   if sigma = 2 pixel wide, then total window you want to use for smoothing is 13!

... total window = 6*sigma + 1
(3sigma rule: half of the window)
Practical: Use 1D Gaussians for efficiency!

2D Gaussian = 1D Gaussian \times 1D Gaussian
Step 2: Take Derivative (Compute Gradient)

Derivative --> $\nabla (G \ast I)$

G: Gaussian
I: image
Step 3: Find Magnitude and Orientation at each pixel

Derivative $\Rightarrow \nabla(G \ast I)$

Calculate the magnitude $\Rightarrow |\nabla(G \ast I)|$

Calculate the orientation $\Rightarrow \bar{n} = \frac{\nabla(G \ast I)}{|\nabla(G \ast I)|}$

G: Gaussian  
I: image

at each pixel!!!!!
Step 3: Find Magnitude and Orientation at each pixel

Derivative $\Rightarrow \nabla (G \ast I)$

Calculate the magnitude $\Rightarrow |\nabla (G \ast I)|$

Calculate the orientation $\Rightarrow \bar{n} = \frac{\nabla (G \ast I)}{|\nabla (G \ast I)|}$

at each pixel!!!!!!

Edge is located at the local maximum of the image $I$ convolved with operator $G_n$ at the direction $n$
Example: Canny-Gradients

X-Derivative of Gaussian  Y-Derivative of Gaussian  Gradient Magnitude
Example: Gradient Orientation

\[
\theta = \text{atan2}(l_y, l_x)
\]
Step 4: Non-maximum suppression

- Non-maximum suppression obtains points where the gradient magnitude is at a maximum along the direction of the gradient.
Step 4: Non-maximum suppression

If gradient responses at r and p are smaller than q, q is an edge (p and r are interpolated).

Search directions (gradient and perpendicular to gradient)
Example: Non-maximum suppression

Before Non-Max Suppression  After Non-Max Suppression
Step 5: Hysteresis Thresholding [L, H]

- You can now use a single threshold to remove some of the non-edge points.
Step 5: Hysteresis Thresholding $[L, H]$

- You can now use a single threshold to remove some of the non-edge points.
- But better to use two thresholds (low and high)! ~ called double thresholding.
Step 5: Hysteresis Thresholding \([L, H]\)

- If the gradient at a pixel is
  - above "High", declare it as an ‘edge pixel’
  - below "Low", declare it as a “non-edge-pixel”
  - between “low” and “high”
    - Consider its neighbors iteratively then declare it an “edge pixel” if it is \textbf{connected} to an ‘edge pixel’ \textbf{directly} or via pixels between “low” and “high”.

\[
\begin{array}{c}
\text{Gradient magnitude} \\
\end{array}
\begin{array}{c}
\text{High} \\
\text{low}
\end{array}
\]
1. Threshold at low/high levels to get weak/strong edge pixels
2. Do connected components, starting from strong edge pixels
How to choose Thresholds

• The detection of edges is based on comparing the edge gradient with a threshold. This threshold value can be chosen low enough only when there is no noise in the image, so that all true edges can be detected without miss.

• In noisy images, however, the threshold selection becomes a problem of maximum likelihood ratio optimization based on Bayes decision theory
  – Practically $\text{Thigh} \sim 1.5 \text{Tlow}$
Example: Hysteresis Thresholding
Example: Final Canny Edges
Recap: Hysteresis thresholding

- Strong edges will be included in the edge map.
- Weak(er) edges will be included in the edge map if and only if
  - They are connected to strong edges.
Example Canny Edge Detector
Effect of Gaussian Kernel (smoothing)

The choice of $\sigma$ depends on desired behavior

- large $\sigma$ detects large scale edges
- small $\sigma$ detects fine features

original  Canny with $\sigma = 1$  Canny with $\sigma = 2$
Effect of Gaussian Kernel (smoothing)

- Noisy image
- Canny filter, $\sigma = 1$
- Canny filter, $\sigma = 3$
Fig 5.13 (a) A tiny example picture consisting of 8×8 pixels, shown after smoothing with a Gaussian convolution filter; (b) Gradient magnitude in the example picture as determined by the Sobel operator. The magnitude values were linearly scaled so that the largest magnitude corresponds to white pixels.
Fig 5.15  (a) Gradient sectors at each image pixel as defined in Fig. 5.14a. Note that, due to using a $3 \times 3$ Sobel filter, no gradients can be computed in the top or bottom row or in the first or last column of the image. (b) The same image after non-maxima suppression. A pixel was removed if its gradient magnitude (shown as intensity) was exceeded by that of either of its neighbors in the direction of the arrow.
Fig 5.16 Hysteresis thresholding on the example image. (a) Initial thresholding removes some of the non-zero pixels and creates five edges and five edge candidates. (b) The four pixels that are connected to edges either directly or via other edges pixels are turned into edges, and the disconnected one is removed, creating the output of the Canny edge detector.
One last example

Sobel

Canny
Summary: Why Canny?

- It is less sensitive to noise
- It removes streaking by using double thresholding
- Offer good localization
- Provides one-pixel wide edges (thin)
Questions?