**Outline**
- What is edge detection
  - Origin of edges
  - Noise problem
- Different edge detectors:
  - Canny edge detector
  - Zero-crossing edge detector

**Objectives:**
- Convert a 2D image into a set of profile curves
- Extracts salient features of the scene

**Profiles of Intensity Edges**
- 1st-order derivatives:
  - Produce thicker edges
  - Single response at step edges
  - Response is the highest at the edge location
  - Find the maximum response location
- 2nd-order derivatives:
  - More sensitive to noises
  - Double response at step edges
  - Response is zero at edge location
  - Find the zero-crossing location

**Origin of Edges**
- Edges are caused by a variety of factors
  - Surface normal discontinuity
  - Depth discontinuity
  - Surface color discontinuity
  - Illumination discontinuity
Noise Problem

- Consider a 1D problem:
  - Detect jump from a noisy background
  - Small noises can cause large value after derivative calculation
- Conclusion:
  - Derivative calculating is very sensitive to noises
  - Smoothing should be applied to filter the noises

Apply Gaussian Smoothing

Derivative of Gaussian (DoG)

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\frac{\partial}{\partial x} G(x, y) = -\frac{x}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\frac{\partial}{\partial y} G(x, y) = -\frac{y}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$D_{\text{DoG}} = \begin{bmatrix}
-1 & -2 & 0 & 2 & 1 \\
-4 & -10 & 0 & 10 & 4 \\
-7 & -17 & 0 & 17 & 7 \\
-4 & -10 & 0 & 10 & 4 \\
-1 & -2 & 0 & 2 & 1 
\end{bmatrix}$$

Response of DoG Filter

Laplacian of Gaussian (LoG)

$$LoG(x, y) = \frac{1}{80} \begin{bmatrix}
1 & 1 & 1 \\
1 & 5 & 6 & 5 & 1 \\
1 & 5 & 0 & -11 & 0 & 5 & 1 \\
1 & 6 & -11 & -36 & -11 & 6 & 1 \\
1 & 5 & 0 & -11 & 0 & 5 & 1 \\
1 & 5 & 6 & 5 & 1 \\
1 & 1 & 1 
\end{bmatrix}$$

$$LoG(x, y) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2} = \frac{x^2 + y^2 - 2\sigma^2}{2\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Response of LoG Filter
Canny Edge Detector

- Calculate 1st order derivative:
  - Apply DoG in both horizontal & vertical directions
  - Calculate both gradient magnitude and direction
- Perform non-maximal suppression:
  - Any gradient magnitude that is not a local peak along the gradient direction is set to zero
- Eliminate noisy edges using two threshold values:
  - Any pixel that has a gradient value greater than the higher threshold is classed as an edge point
  - Any pixel that connects to existing edge points and has a gradient value above the lower threshold is also classed as an edge point

Zero-Crossing Edge Detector

- Calculate 2nd order derivative:
  - Apply LoG filter
- Threshold the derivative value at zero:
  - Produce a binary image where the boundaries represent the locations of zero crossing points
- Detect boundaries:
  - Mark each foreground point that has at least one background neighbor
  - Or using morphological operators