Edge Detection - Overview

Key Concepts to Master

1. Edge Detection Fundamentals

What is an edge?

- Significant local change in intensity
- Boundary between regions
- Indicates depth discontinuity, surface orientation change, or material property change

Detection methods:

- **First-order derivatives:** Gradient-based (Sobel, Prewitt, Roberts)
- **Second-order derivatives:** Laplacian-based (LoG, Marr-Hildreth)

2. Sobel Operator

MEMORIZE THESE MASKS!

```
Horizontal (Gx):        Vertical (Gy):
[-1  0  1]              [-1 -2 -1]
[-2  0  2]              [ 0  0  0]
[-1  0  1]              [ 1  2  1]
```

Gradient magnitude:

```
|G| = √(Gx² + Gy²)
```

Gradient direction:

```
θ = atan2(Gy, Gx)  or  θ = arctan(Gy / Gx)
```

Hand calculation example:

```
Image patch:
[50  50  50]
[50  50  100]
[50  50  100]

Gx = (-1×50)+(0×50)+(1×50)+(-2×50)+(0×50)+(2×100)+(-1×50)+(0×50)+(1×100)
   = -50 + 0 + 50 - 100 + 0 + 200 - 50 + 0 + 100 = 150

Gy = (-1×50)+(-2×50)+(-1×50)+(0×50)+(0×50)+(0×100)+(1×50)+(2×50)+(1×100)
   = -50 - 100 - 50 + 0 + 0 + 0 + 50 + 100 + 100 = 50

|G| = √(150² + 50²) = √(22500 + 2500) = √25000 ≈ 158
θ = atan2(50, 150) ≈ 18.4°
```

3. Prewitt Operator

Similar to Sobel, different weights:

```
Horizontal (Gx):        Vertical (Gy):
[-1  0  1]              [-1 -1 -1]
[-1  0  1]              [ 0  0  0]
[-1  0  1]              [ 1  1  1]
```

Comparison with Sobel:

- Prewitt: Uniform weighting
- Sobel: Center row/column weighted more (2×)
- Sobel typically better in practice

4. Laplacian Operator

Second-order derivative detector:

Common 3×3 Laplacian masks:

```
4-connected:          8-connected:
[ 0  1  0]            [ 1  1  1]
[ 1 -4  1]            [ 1 -8  1]
[ 0  1  0]            [ 1  1  1]
```

Properties:

- Isotropic (rotation invariant)
- Sensitive to noise
- Detects zero-crossings
- No direction information

5. Laplacian of Gaussian (LoG)

Combines smoothing + edge detection:

Process:

1. Smooth with Gaussian: $G(x,y) = \exp(-(x^2 + y^2)/(2\sigma^2))$
2. Apply Laplacian: ∇^2

Combined operator:

```
LoG = ∇²G = (x² + y² - 2σ²) / σ⁴ × exp(-(x² + y²)/(2σ²))
```

Properties:

- Reduces noise sensitivity
- σ controls scale of edges detected
- Zero-crossings indicate edges

6. Canny Edge Detection

Multi-stage optimal edge detector:

Algorithm steps:

1. **Gaussian smoothing:** Reduce noise
2. **Gradient calculation:** Find magnitude and direction (Sobel)
3. **Non-maxima suppression:** Thin edges to 1 pixel width

4. **Double thresholding:** High & low thresholds
5. **Edge tracking by hysteresis:** Connect weak edges to strong

Parameters:

- **σ (sigma):** Gaussian smoothing strength
  - Larger σ → smoother, fewer edges
  - Smaller σ → more edges, more noise
- **High threshold:** Strong edges (definite edges)
- **Low threshold:** Weak edges (candidate edges)
- **Threshold ratio:** Typically high/low = 2:1 or 3:1

How many parameters? 3 parameters: σ, T_high, T_low

7. Non-Maxima Suppression (NMS)

CRITICAL: Understand this process! Purpose: Thin thick edges to 1-pixel width

Algorithm:

1. For each pixel with gradient magnitude M and direction θ:
2. Interpolate gradient values of two neighbors along θ direction
3. If M is greater than both neighbors → keep
4. Otherwise → suppress (set to 0)

Example:

```
Gradient magnitude:   Gradient direction:
[10  20  30]          [→  →  →]
[15  40  25]          [→  →  →]
[10  20  30]          [→  →  →]

Center pixel (40):
- Check left neighbor (15) and right neighbor (25)
- 40 > 15 and 40 > 25 → KEEP
- Result: Center stays 40, others may be suppressed
```

4 quantized directions:

- 0° (→): Check left and right
- 45° (↗): Check diagonal neighbors
- 90° (↑): Check up and down
- 135° (↖): Check other diagonal

MATLAB Quick Reference

```
% Sobel edge detection
[Gx, Gy] = imgradientxy(img, 'sobel');
[Gmag, Gdir] = imgradient(Gx, Gy);

% Or use edge function
edges = edge(img, 'sobel', threshold);
```

```
% Prewitt
[Gx, Gy] = imgradientxy(img, 'prewitt');

% Laplacian
h = fspecial('laplacian');
edges = imfilter(img, h);

% LoG
h = fspecial('log', hsize, sigma);
edges = imfilter(img, h);

% Canny (most common)
edges = edge(img, 'canny', [low_thresh high_thresh], sigma);
% Example: edge(img, 'canny', [0.1 0.3], 1.5);
```

Study Checklist

- [ ] Sobel masks memorized
- [ ] Can apply Sobel/Prewitt by hand
- [ ] Can calculate gradient magnitude and direction
- [ ] Understand Laplacian operation
- [ ] Know LoG advantages over Laplacian
- [ ] Know all 5 Canny steps
- [ ] Understand effect of each Canny parameter
- [ ] Can perform non-maxima suppression
- [ ] Know difference between 1st and 2nd order methods

Common Exam Problems

1. **Apply Sobel to image patch** → Show all calculations
2. **Calculate gradient magnitude and direction**
3. **How many Canny parameters and what do they do?** → 3: σ, T_high, T_low
4. **Explain non-maxima suppression** → Thin edges by comparing along gradient direction
5. **Compare first vs second-order methods**

Comparison Table

Method	Order	Noise Sensitivity	Edge Localization	Direction Info
Sobel	1st	Medium	Good	Yes
Prewitt	1st	Medium	Good	Yes
Laplacian	2nd	High	Good	No
LoG	2nd	Low	Good	No
Canny	1st	Low	Excellent	Yes

Answers to Common Exam Problems

1. Apply Sobel to image patch Given a 3x3 patch, apply Gx and Gy masks separately. Multiply element-wise and sum. Example with patch [50 50 50; 50 50 100; 50 50 100]:

- `Gx = (-1)(50)+(0)(50)+(1)(50)+(-2)(50)+(0)(50)+(2)(100)+(-1)(50)+(0)(50)+(1)(100) = 150`
- `Gy = (-1)(50)+(-2)(50)+(-1)(50)+(0)(50)+(0)(50)+(0)(100)+(1)(50)+(2)(50)+(1)(100) = 50`

2. Calculate gradient magnitude and direction From Gx and Gy above:

- `|G| = sqrt(150^2 + 50^2) = sqrt(25000) = 158.1`
- `theta = atan2(50, 150) = 18.4 degrees`

3. How many Canny parameters and what do they do? 3 parameters:

- `sigma` -- controls Gaussian smoothing. Larger = fewer edges, less noise.
- `T_high` -- strong edge threshold. Pixels above this are definite edges.
- `T_low` -- weak edge threshold. Pixels between T_low and T_high are edges only if connected to a strong edge (hysteresis).

4. Explain non-maxima suppression Thins thick edges to 1-pixel width. For each pixel, look at the gradient direction theta, check the two neighbors along that direction. If the pixel's gradient magnitude is greater than both neighbors, keep it; otherwise suppress to 0. Directions are quantized to 4 bins: 0, 45, 90, 135 degrees.

5. Compare first vs second-order methods

- First-order (Sobel, Prewitt, Canny): use gradient (first derivative). Find edges where gradient magnitude is large. Provide edge direction. Less noise-sensitive.
- Second-order (Laplacian, LoG): use second derivative. Find edges at zero-crossings. No direction info. More noise-sensitive (Laplacian alone), but LoG mitigates this with Gaussian smoothing. Laplacian is isotropic (rotation invariant).

Related Topics

- Lecture 4: Edge Detection
- Project 2: Edge Detection
- Links to filtering (preprocessing)