Corner Detection - Overview

Key Concepts to Master

1. Why Feature Points (Keypoints)?

Purpose:

- **Distinctive:** Easy to localize precisely
- **Repeatable:** Can find same point under different conditions
- **Invariant:** Robust to rotation, scale, illumination changes
- **Sparse:** Efficient computation and matching
- **Informative:** Encode local structure

Applications:

- Image matching and alignment
- Object recognition
- 3D reconstruction
- Motion tracking

2. What is a Corner?

Definition:

- Point where intensity changes significantly in multiple directions
- Intersection of two or more edges
- High gradient in both x and y directions

Why corners are good features:

- Unique and easy to localize
- More stable than edges alone
- Provide 2D constraints (edges only 1D)

3. Image Patch Comparison

How to compare two patches?

Sum of Squared Differences (SSD):

```
SSD = Σ(I₁(x,y) - I₂(x,y))²
```

- Lower SSD = more similar
- Simple but sensitive to intensity changes

Normalized Cross-Correlation (NCC):

```
NCC = Σ(I₁·I₂) / √(Σ I₁² · Σ I₂²)
```

- Range: [-1, 1], higher = more similar
- Invariant to linear intensity changes

4. Error Function E(u,v)

Measures intensity change when shifting window by (u,v):

```
E(u,v) = Σ w(x,y) [I(x+u, y+v) - I(x,y)]²
        x,y
```

Where:

- w(x,y) = window function (often Gaussian)
- I(x,y) = image intensity
- (u,v) = shift/displacement

Taylor approximation:

```
I(x+u, y+v) ≈ I(x,y) + Iₓ·u + Iᵧ·v
```

Simplified E(u,v):

```
E(u,v) ≈ [u v] M [u]
                 [v]
```

5. Second Moment Matrix M

Definition:

```
M = Σ w(x,y) [Iₓ²    IₓIᵧ]
    x,y      [IₓIᵧ   Iᵧ² ]
```

Where:

- $I_x = \partial I/\partial x$ (gradient in x direction)
- $I_y = \partial I/\partial y$ (gradient in y direction)

Alternative notation:

```
M = [A  C]
    [C  B]

Where:
A = Σ Iₓ²
B = Σ Iᵧ²
C = Σ IₓIᵧ
```

6. Eigenvalue Interpretation of M

Eigenvalues λ_1, λ_2 describe local structure:

λ_1	λ_2	Interpretation
Small	Small	Flat region (no edges)
Large	Small	Edge (gradient in one direction)
Large	Large	**Corner** (gradients in multiple directions)

Geometric interpretation:

- M describes ellipse of intensity changes

- Eigenvalues = lengths of ellipse axes
- Eigenvectors = directions of ellipse axes

Corner condition:

- Both λ_1 and λ_2 must be large
- Indicates significant intensity variation in all directions

7. Harris Corner Response Function R

Formula:

```
R = det(M) - k·(trace(M))²
  = λ₁λ₂ - k(λ₁ + λ₂)²
```

Where:

- $\det(M) = \lambda_1\lambda_2 = AB - C^2$
- $\mathrm{trace}(M) = \lambda_1 + \lambda_2 = A + B$
- k = empirical constant (typically 0.04 - 0.06)

Decision criteria:

- R > threshold → **Corner**
- R < 0 → Edge
- |R| small → Flat region

Why this works:

- Avoids expensive eigenvalue computation
- det(M) large when both eigenvalues large
- trace(M) penalizes if only one eigenvalue large

8. Harris Corner Detection Algorithm

Step 1: Compute gradients

```
Ix = imfilter(img, [-1 0 1]);
Iy = imfilter(img, [-1; 0; 1]);
```

Step 2: Compute products of gradients

```
Ix2 = Ix .^ 2;
Iy2 = Iy .^ 2;
Ixy = Ix .* Iy;
```

Step 3: Apply Gaussian weighting

```
g = fspecial('gaussian', [9 9], sigma);
A = imfilter(Ix2, g);
B = imfilter(Iy2, g);
C = imfilter(Ixy, g);
```

Step 4: Compute corner response

```
k = 0.04;
R = (A .* B - C.^2) - k * (A + B).^2;
```

Step 5: Threshold and non-maxima suppression

```
corners = (R > threshold);
% Apply NMS to get local maxima
```

Hand Calculation Example

Given image patch (3×3):

```
[10  10  10]
[10  50  50]
[10  50  50]
```

Step 1: Compute gradients (using simple differences)

```
Ix (horizontal):       Iy (vertical):
[ 0   0   0]           [ 0   0   0]
[ 0  40   0]           [ 0  40  40]
[ 0  40   0]           [ 0   0   0]
```

Step 2: At center pixel (50):

```
Iₓ² = 40² = 1600
Iᵧ² = 40² = 1600
IₓIᵧ = 40×40 = 1600
```

Step 3: Second moment matrix (simplified, no Gaussian)

```
M = [1600  1600]
    [1600  1600]
```

Step 4: Harris response

```
det(M) = 1600×1600 - 1600×1600 = 0
trace(M) = 1600 + 1600 = 3200
R = 0 - 0.04×(3200)² = -409600
```

Conclusion: R < 0 → This is an **edge**, not a corner!

MATLAB Quick Reference

```
% Harris corner detection (manual)
Ix = imfilter(double(img), [-1 0 1]);
Iy = imfilter(double(img), [-1; 0; 1]);
```

```
Ix2 = Ix .^ 2;
Iy2 = Iy .^ 2;
Ixy = Ix .* Iy;

g = fspecial('gaussian', [9 9], 1.5);
A = imfilter(Ix2, g);
B = imfilter(Iy2, g);
C = imfilter(Ixy, g);

k = 0.04;
R = (A .* B - C.^2) - k * (A + B).^2;

threshold = 0.01 * max(R(:));
corners = (R > threshold);

% Built-in function
corners = detectHarrisFeatures(img);
imshow(img); hold on;
plot(corners);
```

Study Checklist

- [ ] Understand why corners are useful features
- [ ] Know how to compare image patches (SSD, NCC)
- [ ] Can explain error function E(u,v)
- [ ] Can compute second moment matrix M
- [ ] Understand eigenvalue interpretation
- [ ] Can compute Harris response R by hand
- [ ] Know Harris algorithm steps
- [ ] Understand role of k parameter

Common Exam Questions

Q1: Why are feature points useful?

- Distinctive, repeatable, invariant, sparse, informative

Q2: What does the second moment matrix M represent?

- Local intensity gradient structure
- Eigenvalues indicate edge/corner/flat

Q3: How to compute Harris R?

- $R = \det(M) - k \cdot (\text{trace}(M))^2$
- $R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$

Q4: What do eigenvalues of M tell us?

- Both large → corner
- One large → edge
- Both small → flat

Q5: Given gradients I_x, I_y at a point, compute M

- M = [I_x^2 I_xI_y] [I_xI_y I_y^2]

Answers to Common Exam Questions

Q1: Why are feature points useful? They are distinctive (easy to localize), repeatable (found again under different conditions), invariant (robust to rotation, scale, illumination), sparse (efficient), and informative (encode local structure). Used for matching, recognition, 3D reconstruction, and tracking.

Q2: What does the second moment matrix M represent? M encodes the local gradient structure around a pixel. Its eigenvalues describe the intensity variation: both large = corner (intensity changes in all directions), one large = edge (change in one direction only), both small = flat region. Geometrically, M defines an ellipse whose axes correspond to the eigenvalues/eigenvectors.

Q3: How to compute Harris R? `R = det(M) - k*(trace(M))^2`, where `det(M) = AB - C^2` and `trace(M) = A + B`, with `A = sum(Ix^2)`, `B = sum(Iy^2)`, `C = sum(Ix*Iy)`, and k typically 0.04-0.06. Equivalently `R = lambda1*lambda2 - k*(lambda1 + lambda2)^2`. R > threshold = corner, R < 0 = edge, |R| small = flat.

Q4: What do eigenvalues of M tell us?

- Both lambda1 and lambda2 large: **corner** -- intensity changes significantly in multiple directions
- One large, one small: **edge** -- intensity changes in one direction only
- Both small: **flat region** -- no significant intensity change

Q5: Given gradients Ix, Iy at a point, compute M Example: if `Ix = 40, Iy = 30` at a pixel, then `M = [Ix^2, Ix*Iy; Ix*Iy, Iy^2] = [1600, 1200; 1200, 900]`. For a window of pixels, sum these contributions (with Gaussian weighting) across all pixels in the window.

Related Topics

- Lecture 6: Corner Detection
- SIFT uses multi-scale corner detection
- Feature matching for image alignment