SIFT (Scale-Invariant Feature Transform) - Overview

Key Concepts to Master

1. What Does SIFT Do?

Purpose:

- Detect distinctive local features in images
- Match features between different images
- Robust to scale, rotation, illumination changes

SIFT Pipeline:

1. **Scale-space extrema detection**
2. **Keypoint localization**
3. **Orientation assignment**
4. **Descriptor generation**

2. Properties of SIFT Features

Invariance:

- **Scale invariant:** Detects features at their characteristic scale
- **Rotation invariant:** Orientation normalized
- **Illumination invariant:** Descriptor normalized
- **Partial viewpoint invariance**

Robustness:

- Handles occlusion (local features)
- Robust to noise
- Works with partial matches

Distinctiveness:

- 128-dimensional descriptor
- Highly discriminative
- Low false positive rate

3. Scale Space

Concept:

- Continuous function of scale σ
- Represents image at multiple scales simultaneously
- Detects features at their "natural" scale

Scale-space representation:

```
L(x, y, σ) = G(x, y, σ) * I(x, y)
```

Where:

- G(x, y, σ) = Gaussian kernel with scale σ

- I(x, y) = input image
- σ = scale parameter

Difference of Gaussians (DoG):

```
D(x, y, σ) = L(x, y, kσ) - L(x, y, σ)
```

- Approximates scale-normalized Laplacian
- Used to find scale-space extrema
- k = constant scale factor (typically √2)

4. Scale-Space Pyramid Structure

Octaves:

- Each octave = doubling of scale (σ)
- Image progressively downsampled
- Typically 4-5 octaves

Scales per octave:

- Multiple Gaussian blurs at different σ
- Typically 3-5 scales per octave
- DoG computed between adjacent scales

Structure:

```
Octave 1 (original size):
  G(σ), G(kσ), G(k²σ), ...
  DoG₁, DoG₂, DoG₃, ...

Octave 2 (half size):
  G(2σ), G(2kσ), G(2k²σ), ...
  DoG₁, DoG₂, DoG₃, ...
```

5. Keypoint Detection

Finding extrema in DoG:

- Compare pixel to 26 neighbors
  - 8 in same scale
  - 9 in scale above
  - 9 in scale below
- If pixel is local maximum or minimum → candidate keypoint

Refinement:

- Sub-pixel localization (Taylor expansion)
- Reject low-contrast points
- Reject edge responses (similar to Harris)

6. Orientation Assignment

Purpose:

- Achieve rotation invariance

- Assign consistent orientation to keypoint

Method:

1. Compute gradient magnitude and direction in neighborhood
2. Build orientation histogram (36 bins, 10° each)
3. Peak in histogram = dominant orientation
4. Multiple peaks → multiple keypoints with different orientations

7. SIFT Descriptor

128-dimensional vector:

- 4×4 grid of cells around keypoint
- 8-bin orientation histogram per cell
- 4×4×8 = 128 dimensions

Properties:

- Gradient-based (captures shape)
- Normalized (illumination invariant)
- Thresholded (reduce non-linear effects)

Matching:

- Compare descriptors using Euclidean distance
- Typically use ratio test: dist(best)/dist(second-best) < threshold

---

Conceptual Understanding

Why Scale Space?

Problem: Features appear at different scales

- Small σ → detects fine details
- Large σ → detects coarse structures

Solution: Search for features across all scales

- True scale-invariant detection

Why DoG?

Efficient approximation:

- DoG ≈ scale-normalized Laplacian
- Much faster to compute
- Good for finding blobs

Why 128 dimensions?

Balance:

- Distinctive enough (high discrimination)
- Not too large (efficient matching)
- 4×4 spatial bins captures local structure
- 8 orientation bins captures gradient distribution

MATLAB Quick Reference

```
% SIFT detection (using VLFeat or Computer Vision Toolbox)

% Detect SIFT features
points = detectSIFTFeatures(img);

% Extract SIFT descriptors
[features, valid_points] = extractFeatures(img, points);

% Match features between two images
indexPairs = matchFeatures(features1, features2);

% Visualize
figure;
imshow(img); hold on;
plot(points.selectStrongest(50));

% Manual scale-space (conceptual)
sigmas = [1.6, 2.0, 2.5, 3.2, 4.0];
for i = 1:length(sigmas)
    g = fspecial('gaussian', [21 21], sigmas(i));
    L{i} = imfilter(img, g);
end

% DoG
for i = 1:length(sigmas)-1
    DoG{i} = L{i+1} - L{i};
end
```

Study Checklist

- [ ] Know what SIFT does (feature detection + description)
- [ ] Understand SIFT invariance properties
- [ ] Know concept of scale space
- [ ] Understand DoG (Difference of Gaussians)
- [ ] Know scale-space pyramid structure (octaves)
- [ ] Understand keypoint detection in DoG
- [ ] Know purpose of orientation assignment
- [ ] Understand SIFT descriptor (128D)

Common Exam Questions

Q1: What does SIFT do?

- Detects and describes local features
- Scale and rotation invariant

Q2: What are properties of SIFT features?

- Scale invariant, rotation invariant, illumination invariant

Q3: What is scale space?

- Multi-scale representation: L(x, y, σ) = G(σ) * I
- Allows detection at characteristic scale

Q4: What is DoG?

- Difference of Gaussians: D = G(kσ) - G(σ)
- Approximates Laplacian
- Used for keypoint detection

Q5: How is scale-space pyramid organized?

- Multiple octaves (doubling scale)
- Multiple scales per octave
- DoG between adjacent scales

Q6: Why 128 dimensions in SIFT descriptor?

- 4×4 spatial grid × 8 orientation bins
- Balance between distinctiveness and efficiency

Key Advantages Over Harris

- Scale invariant (Harris is not)
- Includes descriptor (Harris just detects)
- More robust to transformations

Answers to Common Exam Questions

Q1: What does SIFT do? SIFT detects and describes local image features that are invariant to scale and rotation. It finds keypoints at their characteristic scale using Difference of Gaussians, assigns each a dominant orientation, and generates a 128-dimensional descriptor. These descriptors can be matched across images for recognition, stitching, and 3D reconstruction.

Q2: What are properties of SIFT features?

- **Scale invariant:** keypoints detected at their natural scale via DoG pyramid
- **Rotation invariant:** descriptor computed relative to dominant gradient orientation
- **Illumination invariant:** descriptor is normalized, reducing sensitivity to brightness changes
- **Partially viewpoint invariant:** local features tolerate moderate viewpoint changes
- **Robust to noise and occlusion** since features are local and sparse

Q3: What is scale space? A continuous multi-scale representation: `L(x, y, sigma) = G(x, y, sigma) * I(x, y)` . The image is convolved with Gaussians of increasing sigma, representing it at progressively coarser scales. This allows detecting features at the scale where they are most prominent.

Q4: What is DoG? Difference of Gaussians: `D(x, y, sigma) = L(x, y, k*sigma) - L(x, y, sigma)` . It approximates the scale-normalized Laplacian of Gaussian. Extrema (local maxima/minima) in DoG across both space and scale are candidate keypoints. Computed by simply subtracting adjacent Gaussian-blurred images -- very efficient.

Q5: How is the scale-space pyramid organized?

- Divided into **octaves** (each octave doubles the effective scale; image is halved in size)

- Within each octave, multiple Gaussian blurs at incrementally increasing sigma (typically 3-5 per octave)
- DoG images computed between adjacent Gaussian images within each octave
- Keypoints found by comparing each DoG pixel to its 26 neighbors (8 same scale + 9 above + 9 below)

Q6: Why 128 dimensions in the SIFT descriptor? The region around each keypoint is divided into a 4x4 grid of cells. In each cell, an 8-bin gradient orientation histogram is computed. 4 x 4 x 8 = **128 dimensions**. This balances distinctiveness (enough bins to discriminate) with efficiency (not so many that matching is slow or overfitting occurs).

Related Topics

- Lecture 8: SIFT
- Image Pyramids (similar multi-scale concept)
- Corner Detection (Harris - single scale version)
- Feature Matching