Scale-Invariant Feature Transform (SIFT)

Concept, Pipeline, and Mathematical Understanding

Dr. Madhu Oruganti

AI & DS

January 5, 2026

Why Do We Need SIFT?

In real-world images:

- Objects appear at different sizes
- Objects may rotate
- Illumination conditions vary

Problem: Pixel-based comparison fails.

Solution: Extract stable, local features.

What is SIFT?

SIFT is a feature extraction algorithm that:

- Detects distinctive local keypoints
- Describes them using robust descriptors
- Is invariant to scale and rotation

Output:

- Keypoints (location, scale, orientation)
- 128-dimensional descriptors

SIFT Pipeline

1. Scale-space construction
2. Difference of Gaussian (DoG)
3. Keypoint localization
4. Orientation assignment
5. Descriptor construction

Scale-Space: Concept

Objects appear at different sizes due to distance.

Idea: Detect features at multiple scales so size changes do not affect detection.

Scale-Space: Mathematical Formulation

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

Where:

- $I(x, y)$ – input image
- $G(x, y, \sigma)$ – Gaussian filter
- σ – scale parameter

Gaussian Kernel

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

Small $\sigma \rightarrow$ fine details Large $\sigma \rightarrow$ coarse structures

Difference of Gaussian (DoG)

Direct Laplacian of Gaussian is expensive.

Solution: Use Difference of Gaussian (DoG) as approximation.

DoG Formula

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

- k – constant scale multiplier
- Highlights blob-like structures

Scale-Space Extrema Detection

- Each pixel compared with 26 neighbors:
  - 8 in same scale
  - 9 in scale above
  - 9 in scale below
- Local maxima/minima $\rightarrow$ keypoint candidates

Why Keypoint Localization?

Some detected points:

- Are low contrast
- Lie along edges

These points are unstable and must be removed.

Localization Using Taylor Expansion

$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}$$

Result: Accurate keypoint position and scale.

Why Orientation Assignment?

If the image rotates:

- Pixel locations change
- Gradient patterns remain similar

Goal: Achieve rotation invariance.

Gradient Computation

$$m(x, y) = \sqrt{L_x^2 + L_y^2}$$

$$\theta(x, y) = \tan^{-1}\left(\frac{L_y}{L_x}\right)$$

- m – gradient magnitude
- θ – gradient orientation

Orientation Histogram

- Gradients around keypoint collected
- Orientation histogram constructed
- Dominant peak $\rightarrow$ keypoint orientation

Descriptor Construction

- Local patch around keypoint selected
- Patch divided into 4×4 regions
- Each region has 8 orientation bins

$$4 \times 4 \times 8 = 128$$

Descriptor size = 128

Why the Descriptor is Robust

- Gradient-based $\rightarrow$ illumination robustness
- Normalized $\rightarrow$ contrast invariance
- Local $\rightarrow$ tolerant to occlusion

Feature Matching

- Compare descriptors using Euclidean distance
- Nearest neighbor gives best match
- Ratio test removes false matches

SIFT Summary

- Scale invariance $\rightarrow$ scale-space
- Rotation invariance $\rightarrow$ orientation assignment
- Robust descriptor $\rightarrow$ 128-D vector
- Widely used in classical vision tasks

Final Takeaway

SIFT detects stable local features and describes them in a way that is invariant to scale and rotation.