SIFT keypoint detection

Keypoint detection with scale selection

- We want to extract keypoints with characteristic scale that is \textit{covariant} with the image transformation.
Basic idea

- Convolve the image with a “blob filter” at multiple scales and look for extrema of filter response in the resulting *scale space*

Blob detection

Find maxima and minima of blob filter response in space and scale

Source: N. Snavely
Blob filter

Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D

\[ \nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \]
Recall: Edge detection

\[ f \]

Edge = maximum of derivative

\[ \frac{d}{dx} g \]

Derivative of Gaussian

\[ f * \frac{d}{dx} g \]

Convolution

Source: S. Seitz
Edge detection, Take 2

Edge = zero crossing of second derivative

Source: S. Seitz
From edges to blobs

- Edge = ripple
- Blob = superposition of two ripples

Spatial selection: the magnitude of the Laplacian response will achieve a maximum at the center of the blob, provided the scale of the Laplacian is “matched” to the scale of the blob
Scale selection

- We want to find the characteristic scale of the blob by convolving it with Laplacians at several scales and looking for the maximum response.
- However, Laplacian response decays as scale increases:

![Graphs showing the decay of Laplacian response with increasing scale.](Image)
Scale normalization

- The response of a derivative of Gaussian filter to a perfect step edge decreases as $\sigma$ increases.

\[
\frac{1}{\sigma \sqrt{2\pi}}
\]
Scale normalization

• The response of a derivative of Gaussian filter to a perfect step edge decreases as $\sigma$ increases

• To keep response the same (scale-invariant), must multiply Gaussian derivative by $\sigma$

• Laplacian is the second Gaussian derivative, so it must be multiplied by $\sigma^2$
Effect of scale normalization

Original signal

Unnormalized Laplacian response

Scale-normalized Laplacian response

maximum
Blob detection in 2D

- Scale-normalized Laplacian of Gaussian:

\[
\nabla^2_{\text{norm}} g = \sigma^2 \left( \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)
\]
Blob detection in 2D

• At what scale does the Laplacian achieve a maximum response to a binary circle of radius $r$?
Blob detection in 2D

- At what scale does the Laplacian achieve a maximum response to a binary circle of radius $r$?
- To get maximum response, the zeros of the Laplacian have to be aligned with the circle
- The Laplacian is given by (up to scale):
  \[ \left( x^2 + y^2 - 2\sigma^2 \right) e^{-\frac{(x^2+y^2)}{2\sigma^2}} \]
- Therefore, the maximum response occurs at $\sigma = \frac{r}{\sqrt{2}}$. 
Scale-space blob detector

1. Convolve image with scale-normalized Laplacian at several scales
Scale-space blob detector: Example
Scale-space blob detector: Example

sigma = 11.9912
Scale-space blob detector

1. Convolve image with scale-normalized Laplacian at several scales
2. Find maxima of squared Laplacian response in scale-space
Scale-space blob detector: Example
Eliminating edge responses

- Laplacian has strong response along edge
Eliminating edge responses

• Laplacian has strong response along edge

• Solution: filter based on Harris response function over neighborhoods containing the “blobs”
Efficient implementation

- Approximating the Laplacian with a difference of Gaussians:

\[
L = \sigma^2 \left( G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)
\]

(Laplacian)

\[
DoG = G(x, y, k\sigma) - G(x, y, \sigma)
\]

(Difference of Gaussians)
Efficient implementation

David G. Lowe.

From feature detection to feature description

• Scaled and rotated versions of the same neighborhood will give rise to blobs that are related by the same transformation

• What to do if we want to compare the appearance of these image regions?
  • *Normalization*: transform these regions into same-size circles
  • Problem: rotational ambiguity
Eliminating rotation ambiguity

• To assign a unique orientation to circular image windows:
  • Create histogram of local gradient directions in the patch
  • Assign canonical orientation at peak of smoothed histogram
SIFT features

- Detected features with characteristic scales and orientations:

David G. Lowe.
From feature detection to feature description

Detection is *covariant*:

\[ \text{features}(\text{transform}(\text{image})) = \text{transform}(\text{features}(\text{image})) \]

Description is *invariant*:

\[ \text{features}(\text{transform}(\text{image})) = \text{features}(\text{image}) \]
SIFT descriptors

- Inspiration: complex neurons in the primary visual cortex

Properties of SIFT

Extraordinarily robust detection and description technique

• Can handle changes in viewpoint
  – Up to about 60 degree out-of-plane rotation
• Can handle significant changes in illumination
  – Sometimes even day vs. night
• Fast and efficient—can run in real time
• Lots of code available

Source: N. Snavely
A hard keypoint matching problem

NASA Mars Rover images
Answer below (look for tiny colored squares...)
What about 3D rotations?
What about 3D rotations?

- Affine transformation approximates viewpoint changes for roughly planar objects and roughly orthographic cameras
Affine adaptation

Consider the second moment matrix of the window containing the blob:

\[
M = \sum_{x,y} w(x, y) \begin{bmatrix}
I_x^2 & I_x I_y \\
I_x I_y & I_y^2
\end{bmatrix} = R^{-1} \begin{bmatrix}
\lambda_1 & 0 \\
0 & \lambda_2
\end{bmatrix} R
\]

Recall:

\[
\begin{bmatrix} u \\ v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}
\]

This ellipse visualizes the “characteristic shape” of the window.
Affine adaptation