CMPSCI 670: Computer Vision
Blob detection

University of Massachusetts, Amherst
October 1, 2014

Instructor: Subhransu Maji
Blob detection

Source: L. Lazebnik
We want to extract features with characteristic scale that is covariant with the image transformation such as scaling and translation.

Matching regions across scales
**Invariance**

- The property should not change when the input is transformed
- For e.g., an *intensity invariant* corner detector finds the same corners even if the intensity of the image is changed

**Covariance**

- The property should be transformed according to the image transformation
- For e.g., a *translation covariant* corner detector finds the same corners translated by the amount the image is translated
Corner location is not covariant to scaling!

All points will be classified as edges!
Blob detection: Basic idea

- To detect blobs, convolve the image with a “blob filter” at multiple scales and look for extrema of filter response in the resulting scale space.
- This will give us a scale and space covariant detector.

Source: L. Lazebnik
Blob detection: Basic idea

Find maxima \textit{and minima} of blob filter response in space \textit{and scale}

Source: N. Snavely
Blob filter

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

\[ \nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \]

Source: L. Lazebnik
Recall: Edge detection

\[ f \]

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

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

Source: S. Seitz
Edge detection, take 2

Edge detection involves detecting significant changes in intensity across an image. This can be achieved through the use of second derivative of Gaussian (Laplacian). The second derivative of the Gaussian function, \( \frac{d^2}{dx^2} g \), is used as a kernel to convolve with the signal \( f \), to detect edges. The zero crossing of this second derivative is used as an edge detection criterion.

Source: S. Seitz
• 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

Source: L. Lazebnik
• 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.

Source: L. Lazebnik
• 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 $\sigma^2$

Source: L. Lazebnik
Effect of scale normalization

Original signal

Unnormalized Laplacian response

Scale-normalized Laplacian response

Source: L. Lazebnik
Blob detection in 2D

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

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

Scale-normalized:

Source: L. Lazebnik
• At what scale does the Laplacian achieve a maximum response to a binary circle of radius $r$?
• 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):

\[
(x^2 + y^2 - 2\sigma^2) e^{-(x^2+y^2)/2\sigma^2}
\]

• Therefore, the maximum response occurs at

\[
\sigma = r / \sqrt{2}.
\]
We define the characteristic scale of a blob as the scale that produces peak of Laplacian response in the blob center.


Source: L. Lazebnik
1. Convolve image with scale-normalized Laplacian at several scales

Source: L. Lazebnik
Scale-space blob detector: Example
Scale-space blob detector: Example

sigma = 11.9912
1. Convolve image with scale-normalized Laplacian at several scales

2. Find maxima of squared Laplacian response in scale-space

Source: L. Lazebnik
Scale-space blob detector: Example

Source: L. Lazebnik
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)

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

(Difference of Gaussians)

\[ (x^2 + y^2 - 2\sigma^2) e^{-(x^2+y^2)/2\sigma^2} \]

Is the Laplacian separable?

Source: L. Lazebnik
Efficient implementation

• 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

Source: L. Lazebnik
SIFT features

- Detected features with characteristic scales and orientations:


Source: L. Lazebnik
Detection is **covariant**:  
\[
\text{features(\text{transform(image)\})} = \text{transform(features(image))}
\]

Description is **invariant**:  
\[
\text{features(\text{transform(image)\})} = \text{features(image)}
\]

Source: L. Lazebnik
SIFT descriptors

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
Affine adaptation

• Affine transformation approximates viewpoint changes for roughly planar objects and roughly orthographic cameras
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.

Source: L. Lazebnik
Affine adaptation example

Scale-invariant regions (blobs)

Source: L. Lazebnik
Affine adaptation example

Affine-adapted blobs

Source: L. Lazebnik
Further readings and thoughts …

• More about scale-space
  • T. Lindeberg, *Scale-space theory: A basic tool for analyzing structures at different scales*, Journal of Applied Statistics, 1994

• SIFT descriptor in detail
  • David G. Lowe, *Distinctive Image Features from Scale-Invariant Keypoints*, IJCV 2004

• How good are local point detectors and descriptors?

• Chapter 4, R. Szeliski’s book