Blob detection
Achieving scale covariance

- Goal: independently detect corresponding regions in scaled versions of the same image
- Need *scale selection* mechanism for finding characteristic region size that is *covariant* with the image transformation
Recall: Edge detection

\[ f \]

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

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

Edge = maximum of derivative

Source: S. Seitz
Edge detection, Take 2

$\frac{d^2}{dx^2} g$

Second derivative of Gaussian (Laplacian)

$\frac{d^2}{dx^2} g$ 

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:

Why does this happen?
Scale normalization

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

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

\[ \nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \]
Blob detection in 2D

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

Scale-normalized:

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

• At what scale does the Laplacian achieve a maximum response to a binary circle of radius $r$?
Scale selection

- 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 $\sigma = r / \sqrt{2}$. 

![Image of a circle and its Laplacian graph]
Characteristic scale

- We define the characteristic scale of a blob as the scale that produces peak of Laplacian response in the blob center.

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
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

Invariance and covariance properties

• Laplacian (blob) response is *invariant* w.r.t. rotation and scaling
• Blob location and scale is *covariant* w.r.t. rotation and scaling
• What about intensity change?
Achieving affine covariance

- Affine transformation approximates viewpoint changes for roughly planar objects and roughly orthographic cameras
Achieving affine covariance

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:

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

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

Scale-invariant regions (blobs)
Affine adaptation example

Affine-adapted blobs
From covariant detection to invariant description

- Geometrically transformed versions of the same neighborhood will give rise to regions 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.
Affine normalization

- Problem: There is no unique transformation from an ellipse to a unit circle
  - We can rotate or flip a unit circle, and it still stays a unit circle
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
From covariant regions to invariant features

Extract affine regions → Normalize regions → Eliminate rotational ambiguity → Compute appearance descriptors

SIFT (Lowe '04)
Invariance vs. covariance

Invariance:
• $\text{features}(\text{transform(image)}) = \text{features(image)}$

Covariance:
• $\text{features}(\text{transform(image)}) = \text{transform(features(image))}$

Covariant detection => invariant description