Local feature: Blob detection

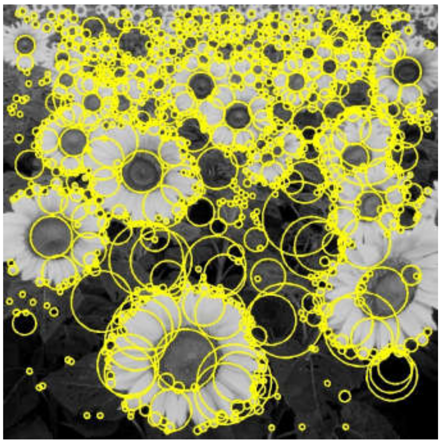
Lu Peng
School of Computer Science,
Beijing University of Posts and Telecommunications

0

Machine Vision Technology							
Semantic information				Metric 3D information			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

1

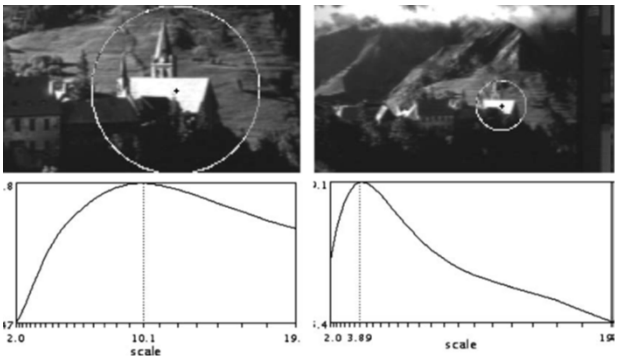
Blob detection



Source： S. Lazebnik

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



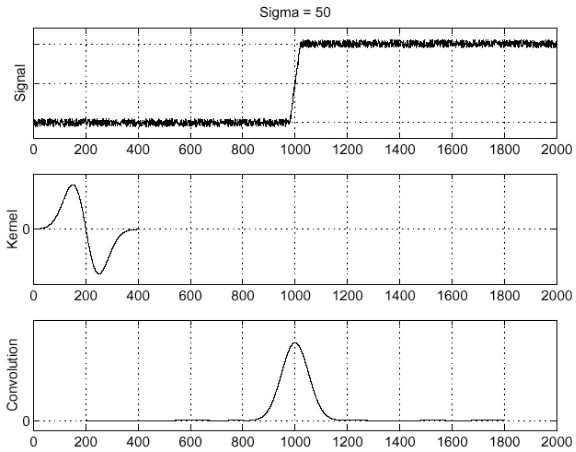
Source： S. Lazebnik

Recall: Edge detection

f

$\frac{d}{dx} g$

$f * \frac{d}{dx} g$



Edge

Derivative of Gaussian

Edge = maximum of derivative

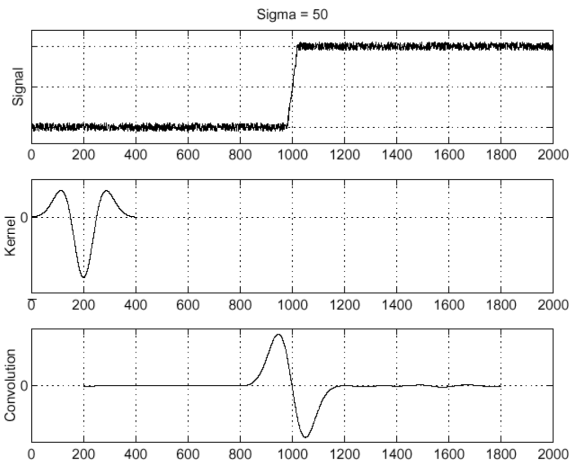
Source: S. Seitz

Edge detection, Take 2

f

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

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



Edge

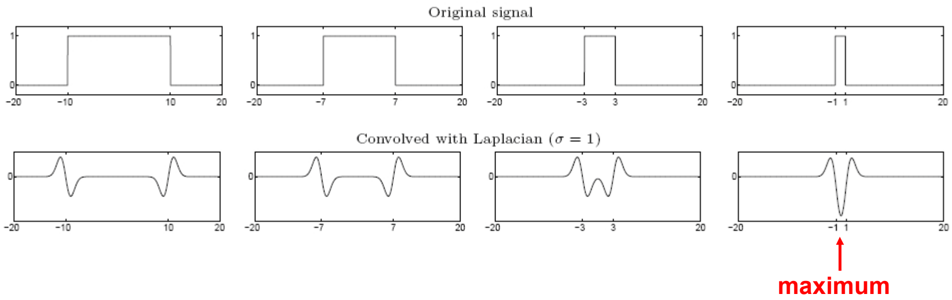
Second derivative of Gaussian (Laplacian)

Edge = zero crossing of second derivative

Source: S. Seitz

From edges to blobs

- Edge = ripple
- Blob = superposition of two ripples



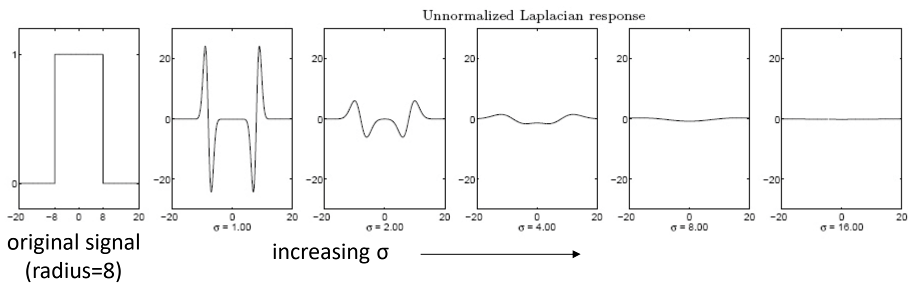
Spatial selection: the magnitude of the Laplacian response will achieve a maximum at the center of the blob, provided the scale of the Laplacian is “matched” to the scale of the blob

Source： S. Lazebnik

6

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:



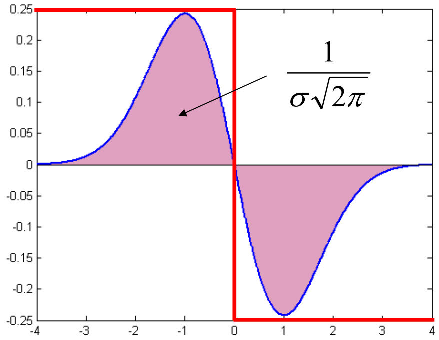
Why does this happen?

Source： S. Lazebnik

7

Scale normalization

- The response of a derivative of Gaussian filter to a perfect step edge decreases as σ increases



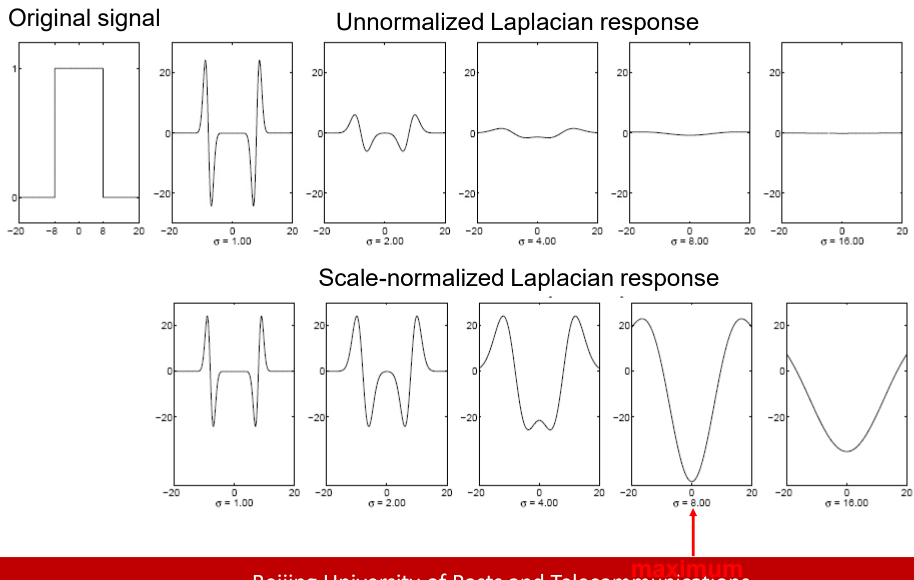
Source: S. Lazebnik

Scale normalization

- The response of a derivative of Gaussian filter to a perfect step edge decreases as σ increases
- To keep response the same (scale-invariant), must multiply Gaussian derivative by σ
- Laplacian is the second Gaussian derivative, so it must be multiplied by σ^2

Source: S. Lazebnik

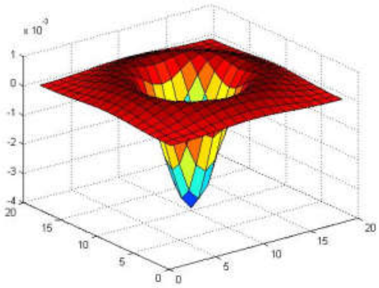
Effect of scale normalization

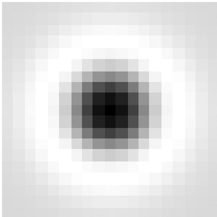


Source： S. Lazebnik

Blob detection in 2D

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



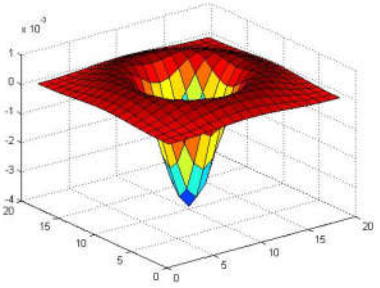


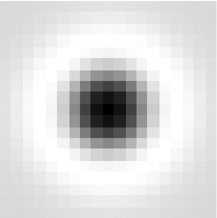
$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

Source： S. Lazebnik

Blob detection in 2D

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



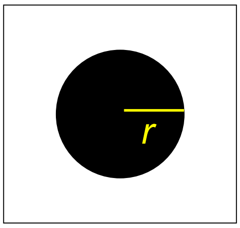


Scale-normalized: $$\nabla^2_{\text{norm}} g = \sigma^2 \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$$

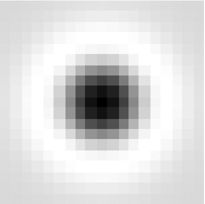
Source： S. Lazebnik

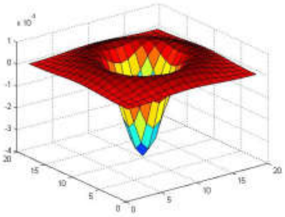
Scale selection

- At what scale does the Laplacian achieve a maximum response to a binary circle of radius r?



image



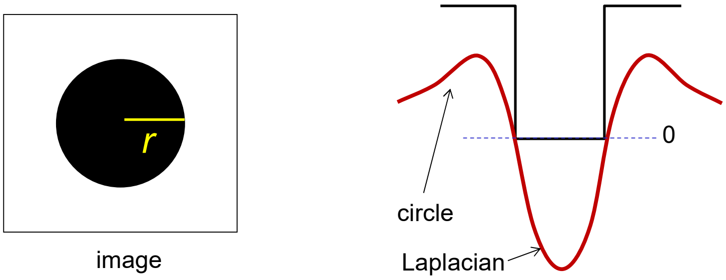


Laplacian

Source： S. Lazebnik

Scale selection

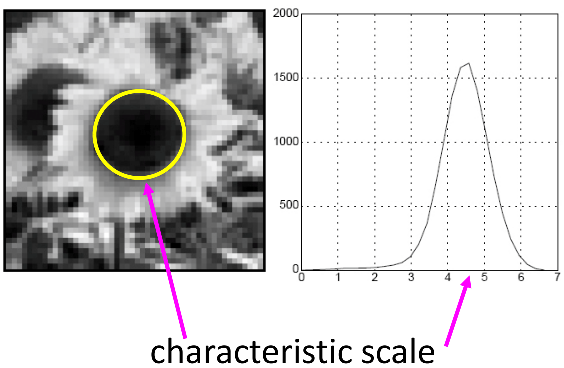
- At what scale does the Laplacian achieve a maximum response to a binary circle of radius r?
- To get maximum response, the zeros of the Laplacian have to be aligned with the circle
- The Laplacian is given by (up to scale): $(x^2 + y^2 - 2\sigma^2)\, e^{-(x^2+y^2)/2\sigma^2}$
- Therefore, the maximum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



Source： S. Lazebnik

Characteristic scale

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



T. Lindeberg (1998). "Feature detection with automatic scale selection." *International Journal of Computer Vision* **30** (2): pp 77--116.

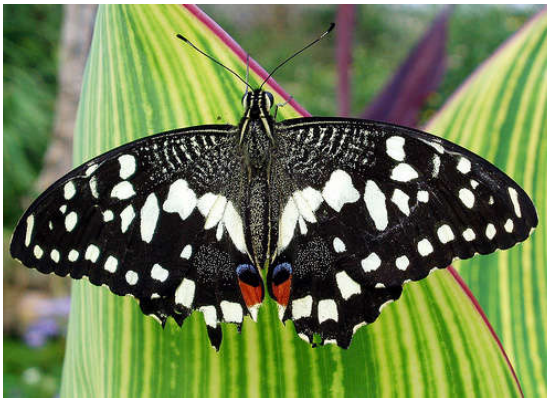
Source： S. Lazebnik

Scale-space blob detector

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

Source: S. Lazebnik

Scale-space blob detector: Example



Source: S. Lazebnik

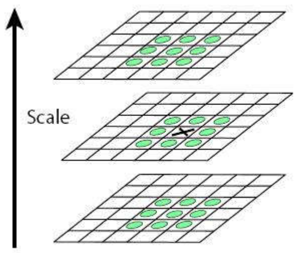
Scale-space blob detector: Example

sigma = 11.9912

Source： S. Lazebnik

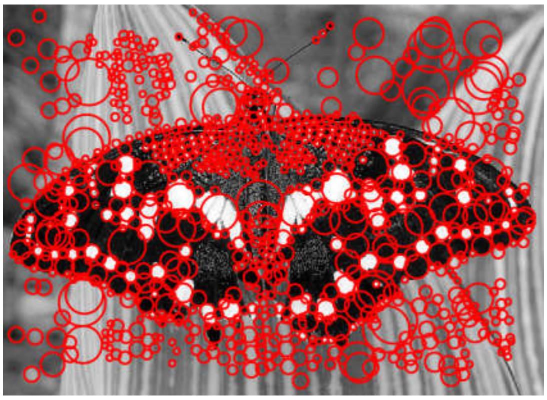
Scale-space blob detector

1. Convolve image with scale-normalized Laplacian at several scales
2. Find maxima of squared Laplacian response in scale-space



Source： S. Lazebnik

Scale-space blob detector: Example



Source: S. Lazebnik

Efficient implementation

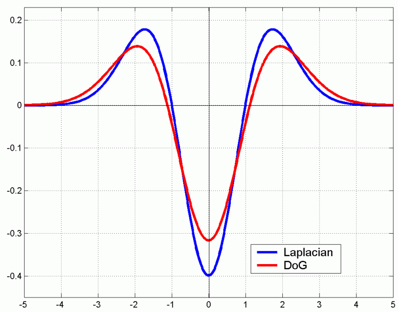
Approximating the Laplacian with a difference of Gaussians:

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

(Laplacian)

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

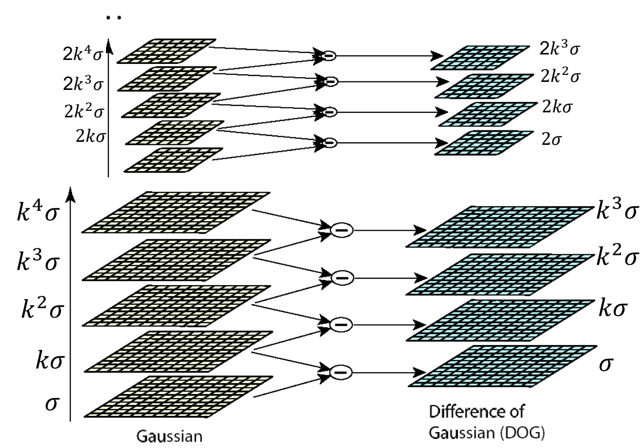
(Difference of Gaussians)



Source: S. Lazebnik

Efficient implementation

$k = 2^{1/s}$



David G. Lowe. ["Distinctive image features from scale-invariant keypoints."](#) *IJCV* 60 (2), pp. 91-110, 2004.

Source： S. Lazebnik

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?

Source： S. Lazebnik

Achieving affine covariance

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

Source: S. Lazebnik

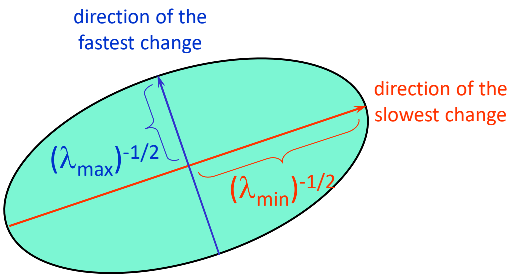
Achieving affine covariance

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:

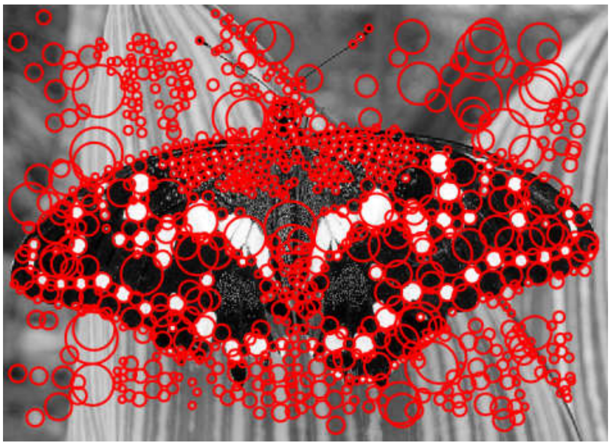
$$[u \ \ v] \ M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$$



This ellipse visualizes the “characteristic shape” of the window

Source: S. Lazebnik

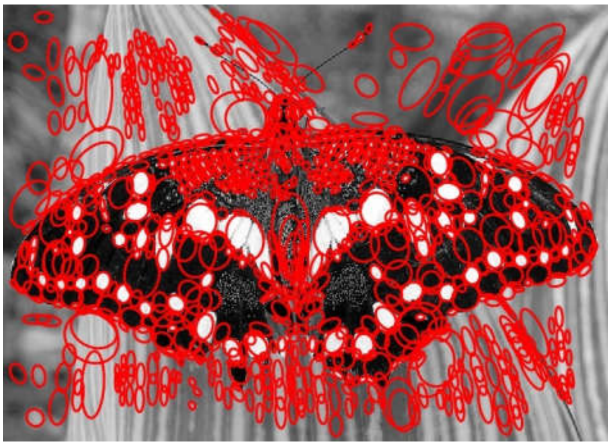
Affine adaptation example



Scale-invariant regions (blobs)

Source： S. Lazebnik

Affine adaptation example



Affine-adapted blobs

Source： S. Lazebnik

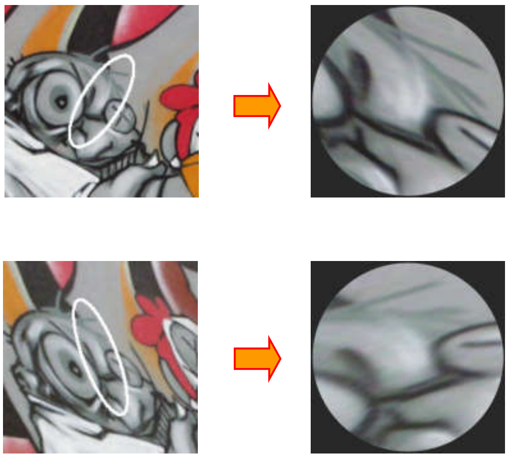
From covariant detection to invariant description

- 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

Source： S. Lazebnik

Affine normalization

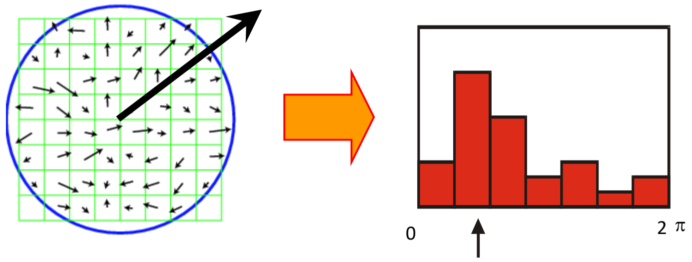
- Problem: There is no unique transformation from an ellipse to a unit circle
  - We can rotate or flip a unit circle, and it still stays a unit circle



Source： S. Lazebnik

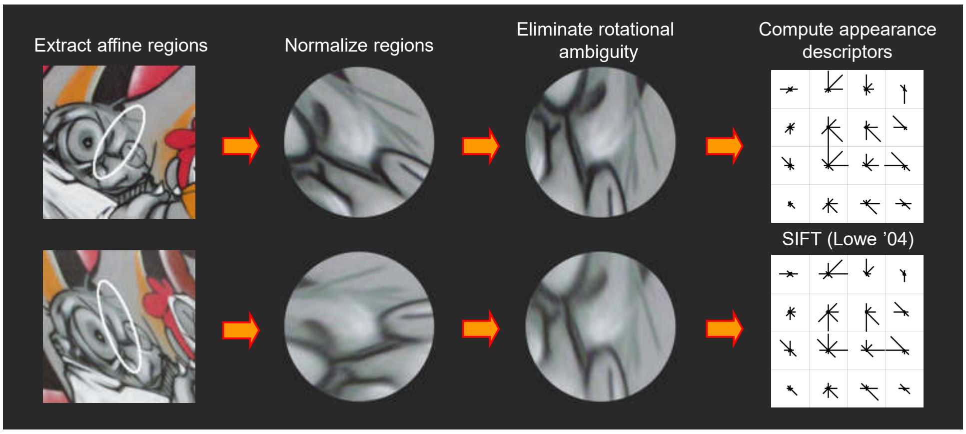
Eliminating rotation ambiguity

- 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： S. Lazebnik

From covariant regions to invariant features



Source： S. Lazebnik

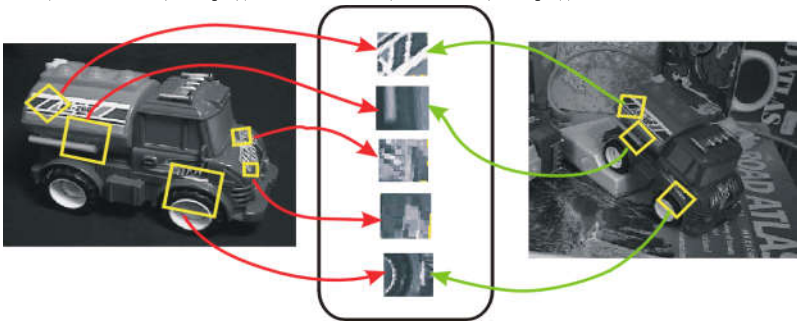
Invariance vs. covariance

Invariance:

- features(transform(image)) = features(image)

Covariance:

- features(transform(image)) = transform(features(image))



Covariant detection => invariant description

Source： S. Lazebnik