COLLEGE OF ENGINEERING
ELECTRICAL ENGINEERING & COMPUTER SCIENCE
UNIVERSITY OF MICHIGAN

Local Image Features

EECS 598-08 Fall 2014
Foundations of Computer Vision

Instructor: Jason Corso (jjcorso)
web.eecs.umich.edu/~jjcorso/t/598F14

Readings: FP 5; SZ 4.2, 4.3
Date: 9/29/14

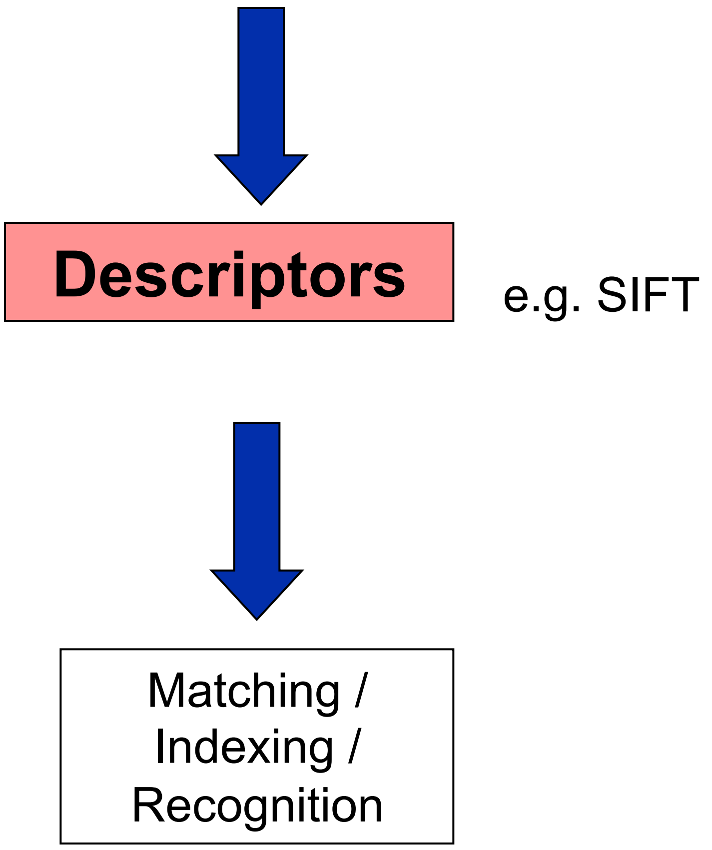
Plan

- What are local image features and why are they useful.
- Local Image Feature Detection
- Invariance
- Local Image Feature Description

Change to be more comprehensive and integrated with this lecture…

Goal:

Identify interesting regions from the images (edges, corners, blobs…)

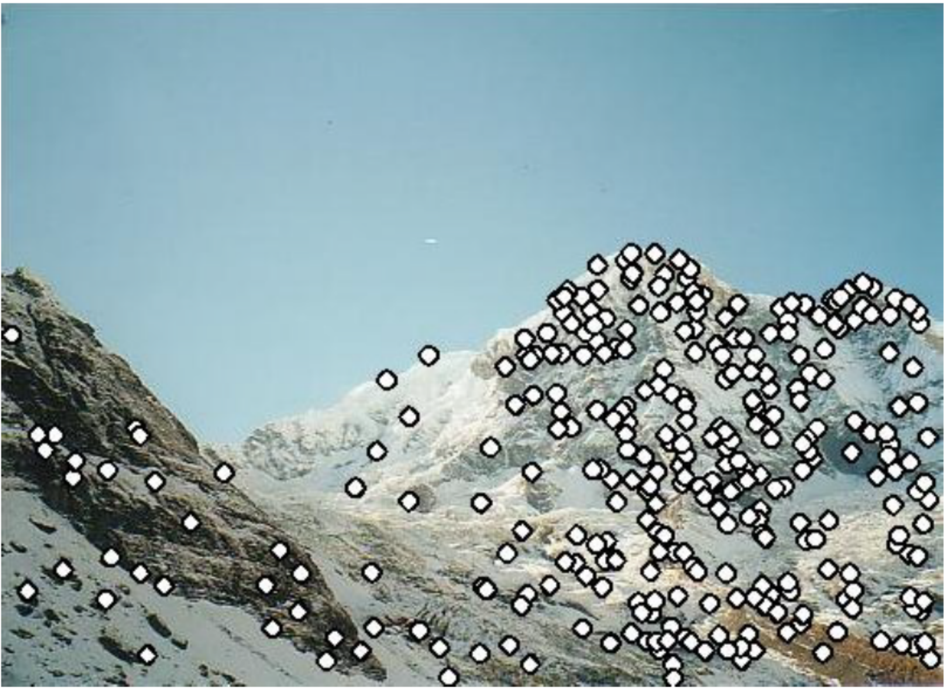


Source: Savarese Slides

Application: Image Stitching

Sources for this example: I. Kokkinos, D. Frolova, D. Simakov.

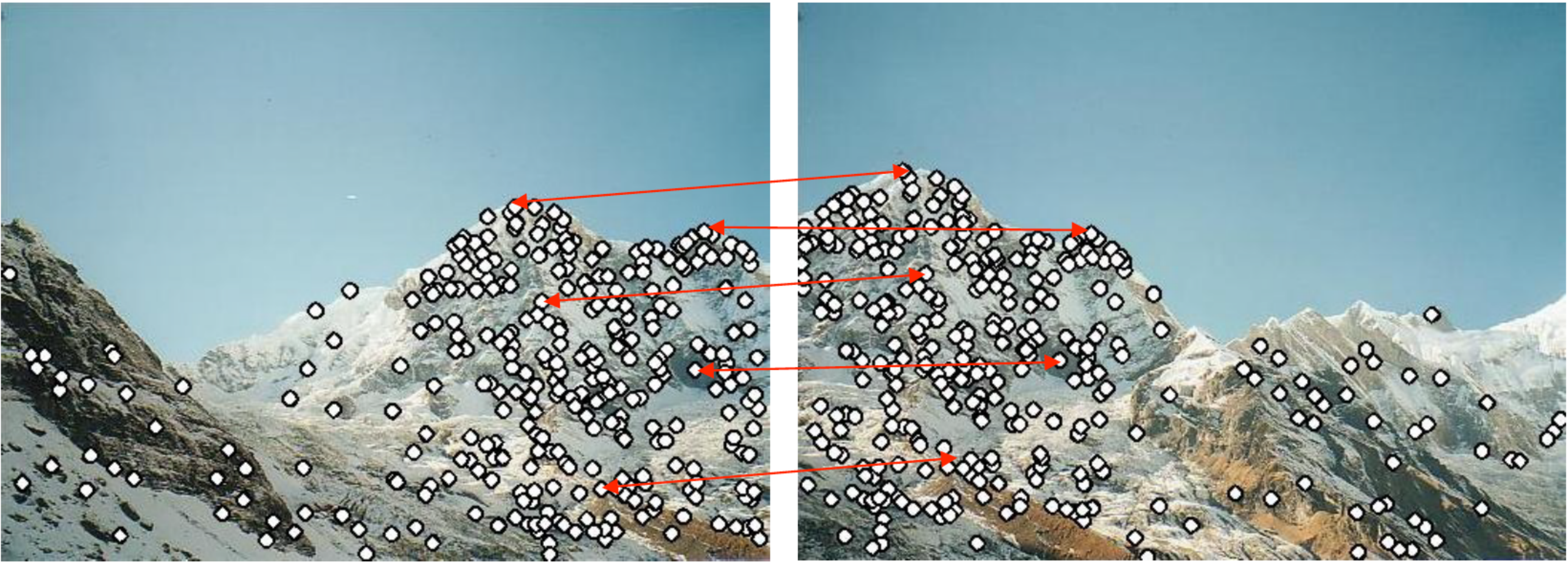
Application: Image Stitching



1. Detect feature points in both images.

Sources for this example: I. Kokkinos, D. Frolova, D. Simakov.

Application: Image Stitching



1. Detect feature points in both images.
2. Find corresponding pairs of feature points.

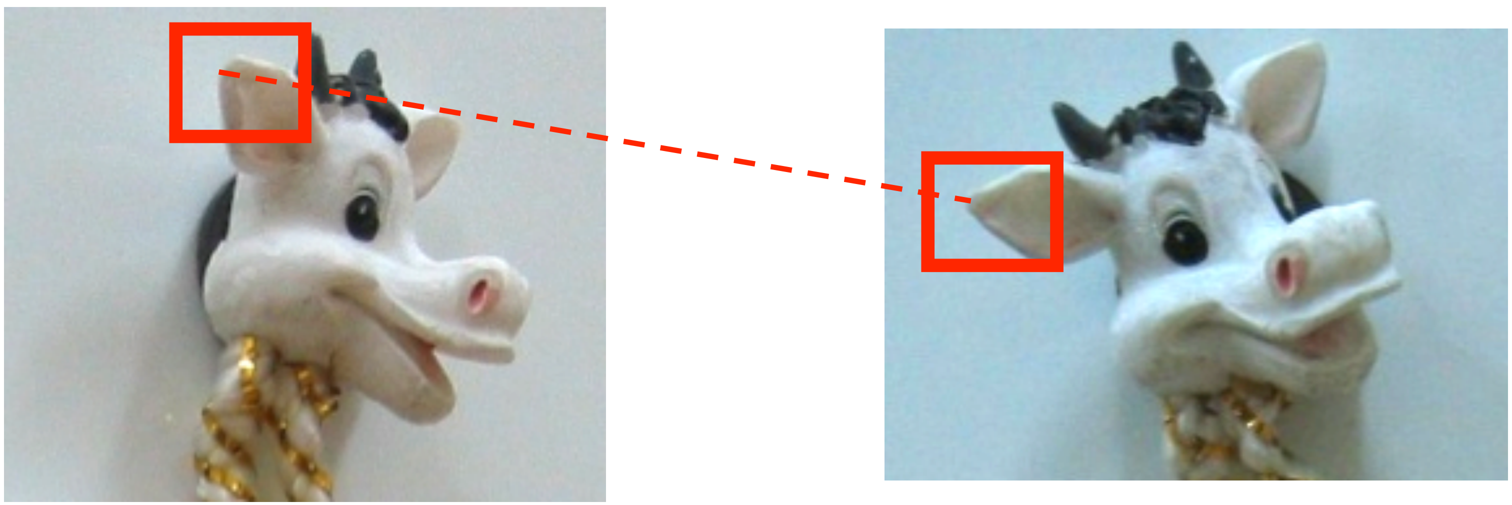
Sources for this example: I. Kokkinos, D. Frolova, D. Simakov.

Application: Image Stitching

1. Detect feature points in both images.
2. Find corresponding pairs of feature points.
3. Use the pairs the align the images.

Sources for this example: I. Kokkinos, D. Frolova, D. Simakov.

Application: Estimating Fundamental Matrix



1. Detect feature points in both images.
2. Find corresponding pairs of feature points.
3. Use the pairs to estimate epipolar geometry across images.

Sources for this example: S. Savarese.

Application: Detect Object Instances

1. Detect feature points in both images.
2. Find corresponding pairs of feature points.
3. Use the pairs to match object instances.

Sources for this example: S. Savarese.

Local Image Point Applications

- Image alignment (stitching, mosaics)
- 3D reconstruction
- Motion tracking
- Object recognition
- Indexing and database retrieval
- Robot navigation

Advantages of local features

Locality
- features are local, so robust to occlusion and clutter

Distinctiveness:
- can differentiate a large database of objects

Quantity
- hundreds or thousands in a single image

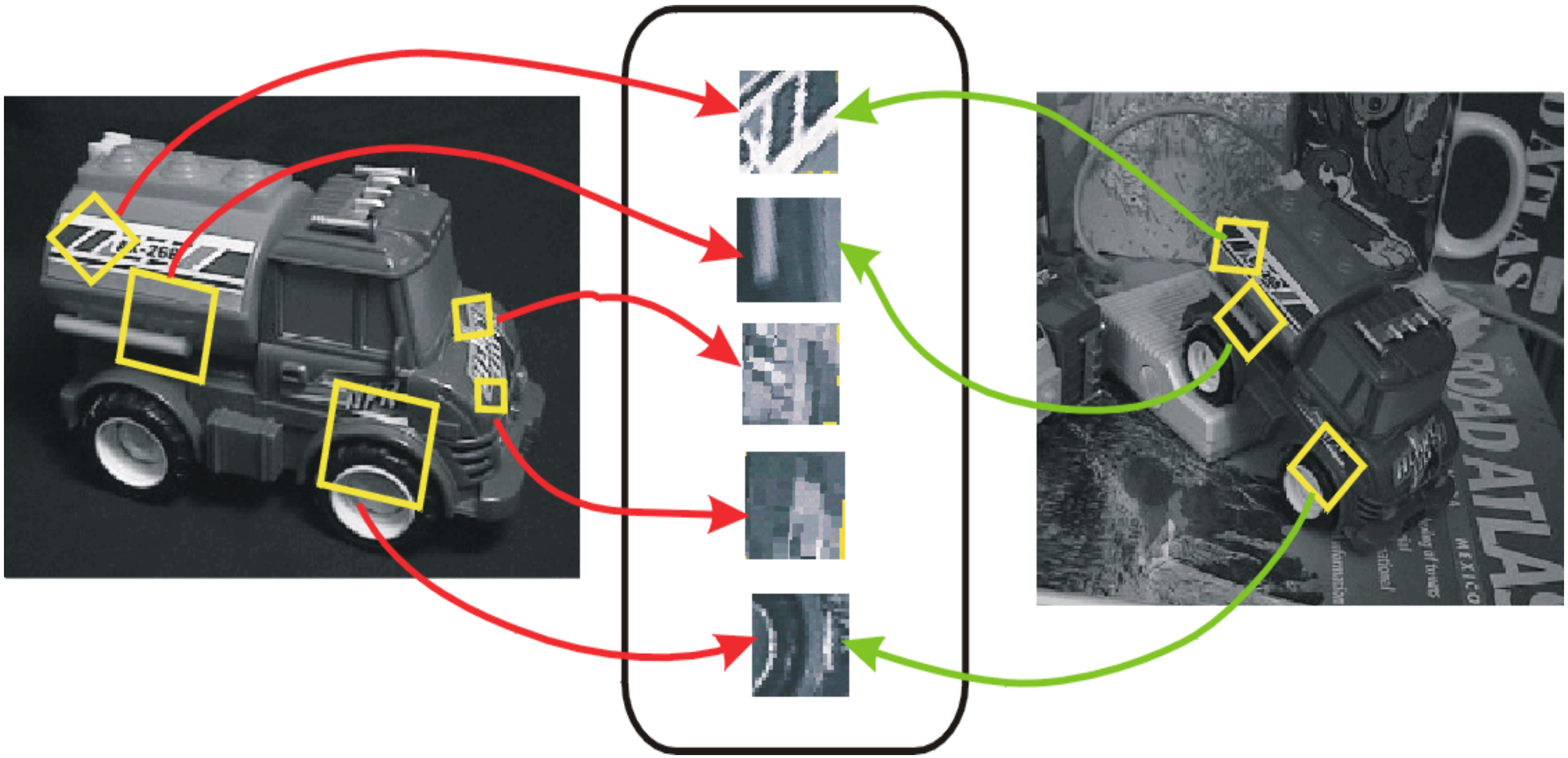
Efficiency
- real-time performance achievable

Generality
- exploit different types of features in different situations

Source: S. Seitz slides.

Challenges

- Repeatability
- Uniqueness
- Invariance



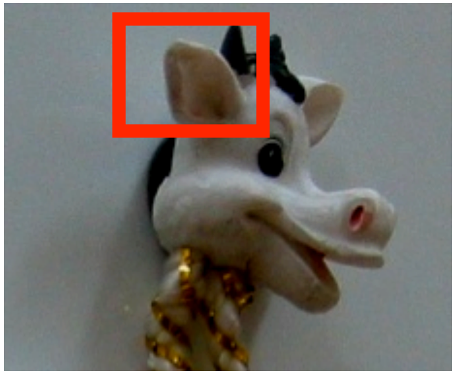
Descriptors

Sources for this example: I. Kokkinos, D. Frolova, D. Simakov, S. Seitz.

What makes a good feature?

Source for this example: S. Seitz.

Repeatability



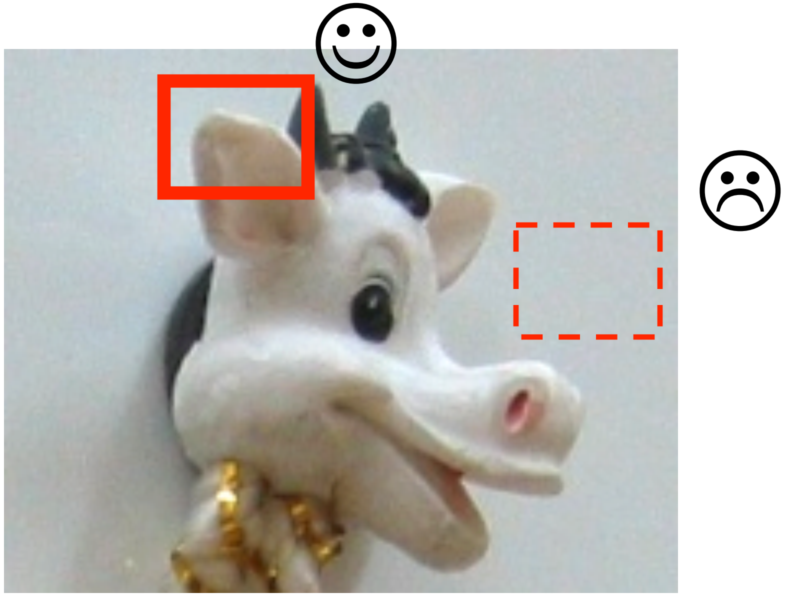
Illumination invariance

Scale invariance

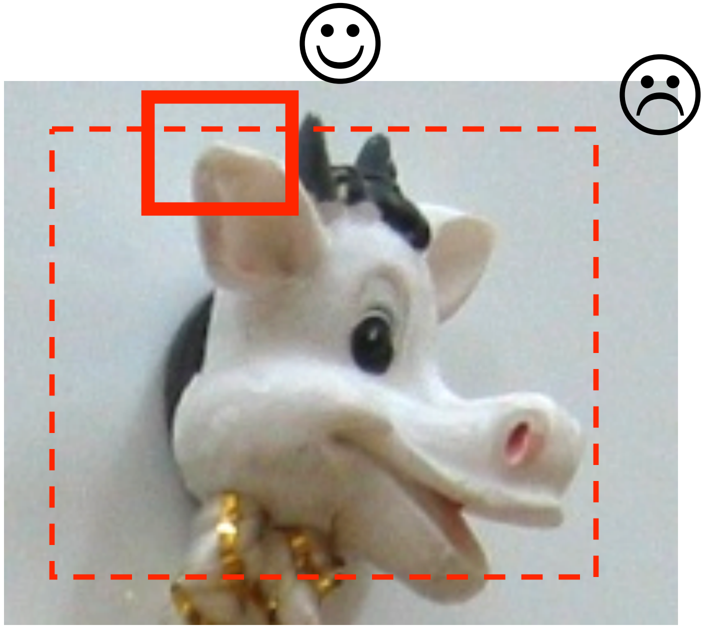
Pose invariance
- Rotation
- Affine

Source for this example: Savarese.

- Saliency



- Locality



Source for this example: Savarese.

One criterion is uniqueness

Look for image regions that are unusual

– Lead to unambiguous matches in other images

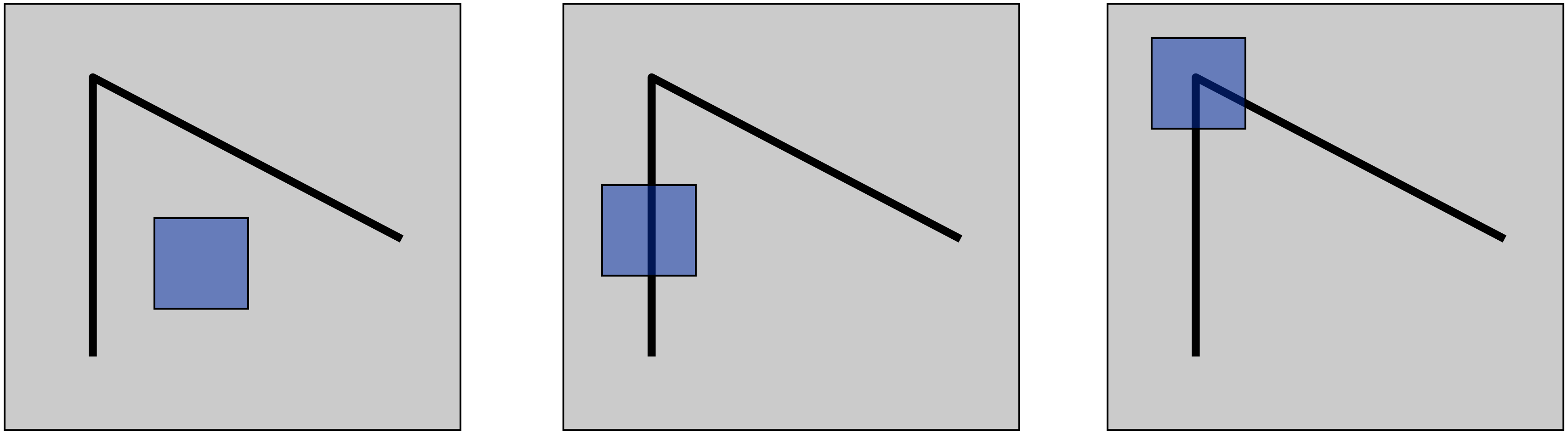
How to define “unusual”?

Source: S. Seitz.

Local measures of uniqueness

Suppose we only consider a small window of pixels

- What defines whether a feature is a good or bad candidate?

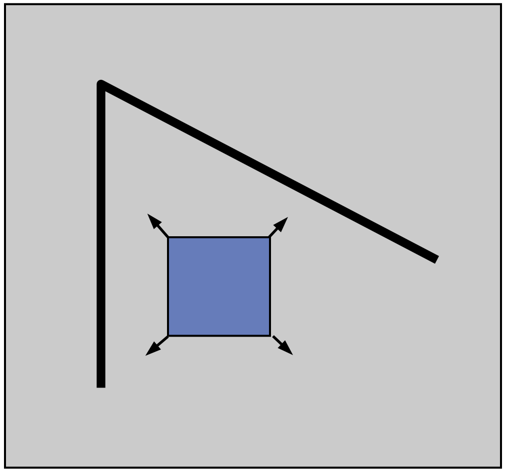


Sources: I. Kokkinos, D. Frolova, D. Simakov, S. Seitz.

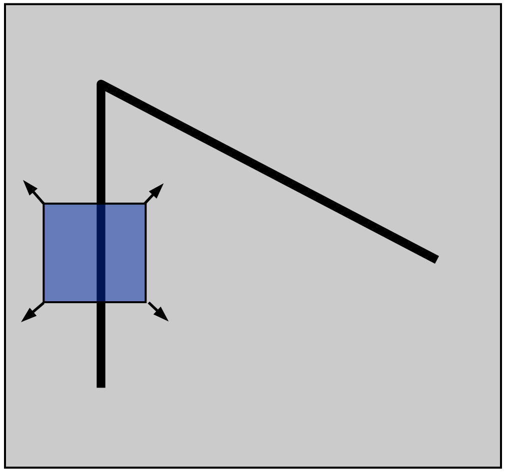
Feature detection

Local measure of feature uniqueness

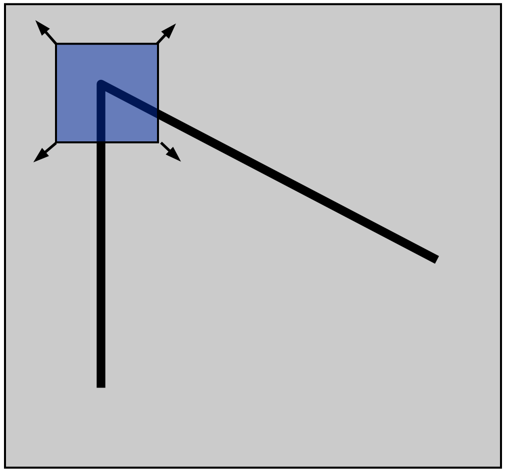
- How does the window change when you shift it?
- Shifting the window in *any direction* causes a *big change*



"flat" region:
no change in all directions



"edge":
no change along the edge direction



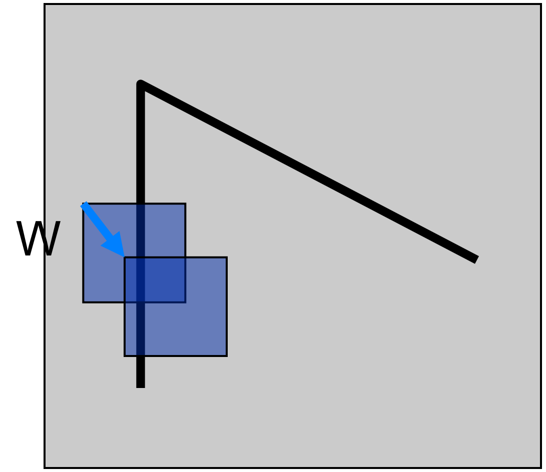
"corner":
significant change in all directions

Sources: I. Kokkinos, D. Frolova, D. Simakov, S. Seitz.

Feature detection: the math

Consider shifting the window W by (u,v)

- how do the pixels in W change?
- compare each pixel before and after by summing up the squared differences (SSD)
- this defines an SSD "error" of *E(u,v)*:



$$E(u,v) = \sum_{(x,y)\in W} \left[I(x+u,y+v) - I(x,y)\right]^2$$

Source: S. Seitz.

Small motion assumption

- Taylor Series expansion of I

$$I(x+u, y+v) = I(x,y) + \frac{\partial I}{\partial x} u \frac{\partial I}{\partial y} v + \text{higher order terms}$$

- If the motion is small, then the first order approx. is good:

$$I(x+u, y+v) \approx I(x,y) + \frac{\partial I}{\partial x} u \frac{\partial I}{\partial y} v$$

$$\approx I(x,y) + \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

shorthand

$$I_x = \frac{\partial I}{\partial x}$$

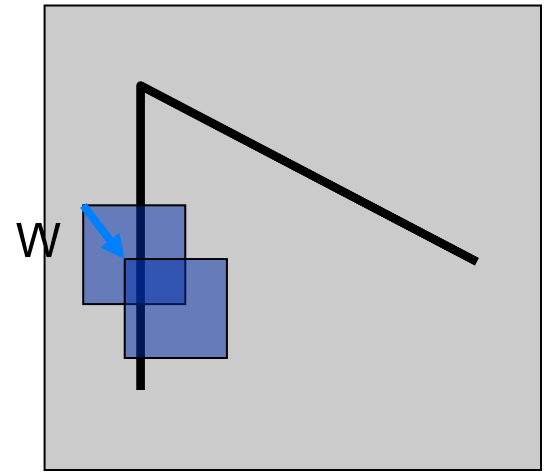
- Plug this back into the objective function.

Source: S. Seitz.

Feature detection: the math

Consider shifting the window W by (u,v)

- how do the pixels in W change?
- compare each pixel before and after by summing up the squared differences
- this defines an "error" of E(u,v):



$$E(u,v) = \sum_{(x,y)\in W} \left[I(x+u,y+v) - I(x,y)\right]^2$$

$$\approx \sum_{(x,y)\in W} \left[I(x,y) + [I_x \; I_y]\begin{bmatrix} u \\ v \end{bmatrix} - I(x,y)\right]^2$$

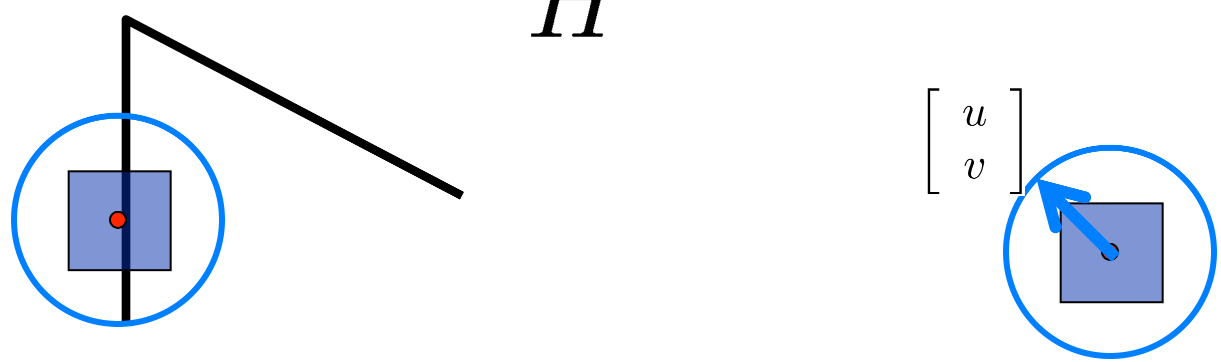
$$\approx \sum_{(x,y)\in W} \left[[I_x \; I_y]\begin{bmatrix} u \\ v \end{bmatrix}\right]^2$$

Source: S. Seitz.

Feature detection: the math

This can be rewritten:

$$E(u,v) = \sum_{(x,y)\in W} [u\ v] \underbrace{\begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}}_{H} \begin{bmatrix} u \\ v \end{bmatrix}$$



For the example above

- You can move the center of the window to anywhere on the blue unit circle
- Which directions will result in the largest and smallest E values?
- We can find these directions by looking at the eigenvectors of *H*

Source: S. Seitz.

Quick eigenvalue/eigenvector review

The **eigenvectors** of a matrix **A** are the vectors **x** that satisfy:

$$Ax = \lambda x$$

The scalar λ is the **eigenvalue** corresponding to **x**

- The eigenvalues are found by solving:

$$det(A - \lambda I) = 0$$

- In our case, ***A*** = ***H*** is a 2x2 matrix, so we have

$$det \begin{bmatrix} h_{11} - \lambda & h_{12} \\ h_{21} & h_{22} - \lambda \end{bmatrix} = 0$$

- The solution:

$$\lambda_{\pm} = \frac{1}{2} \left[(h_{11} + h_{22}) \pm \sqrt{4h_{12}h_{21} + (h_{11} - h_{22})^2} \right]$$

Once you know λ, you find **x** by solving

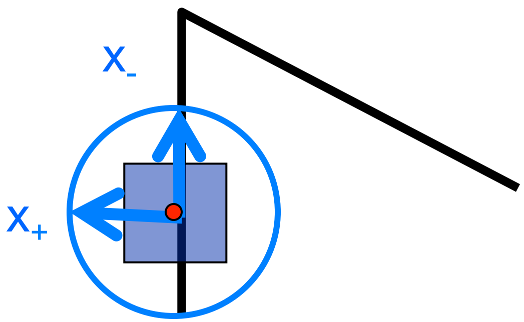
$$\begin{bmatrix} h_{11} - \lambda & h_{12} \\ h_{21} & h_{22} - \lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0$$

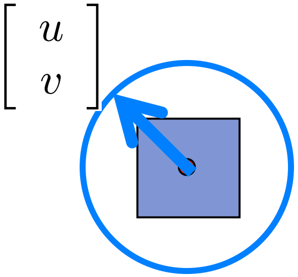
Source: S. Seitz.

Feature detection: the math

This can be rewritten:

$$E(u,v) = \sum_{(x,y)\in W} [u\ v] \underbrace{\begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}}_{H} \begin{bmatrix} u \\ v \end{bmatrix}$$





Eigenvalues and eigenvectors of H

- Define shifts with the smallest and largest change (E value)
- x_+ = direction of largest increase in E.
- λ_+ = amount of increase in direction x_+
- x_- = direction of smallest increase in E.
- λ- = amount of increase in direction x_+

$$\begin{aligned} Hx_+ &= \lambda_+ x_+ \\ Hx_- &= \lambda_- x_- \end{aligned}$$

Source: S. Seitz.

Feature detection: the math

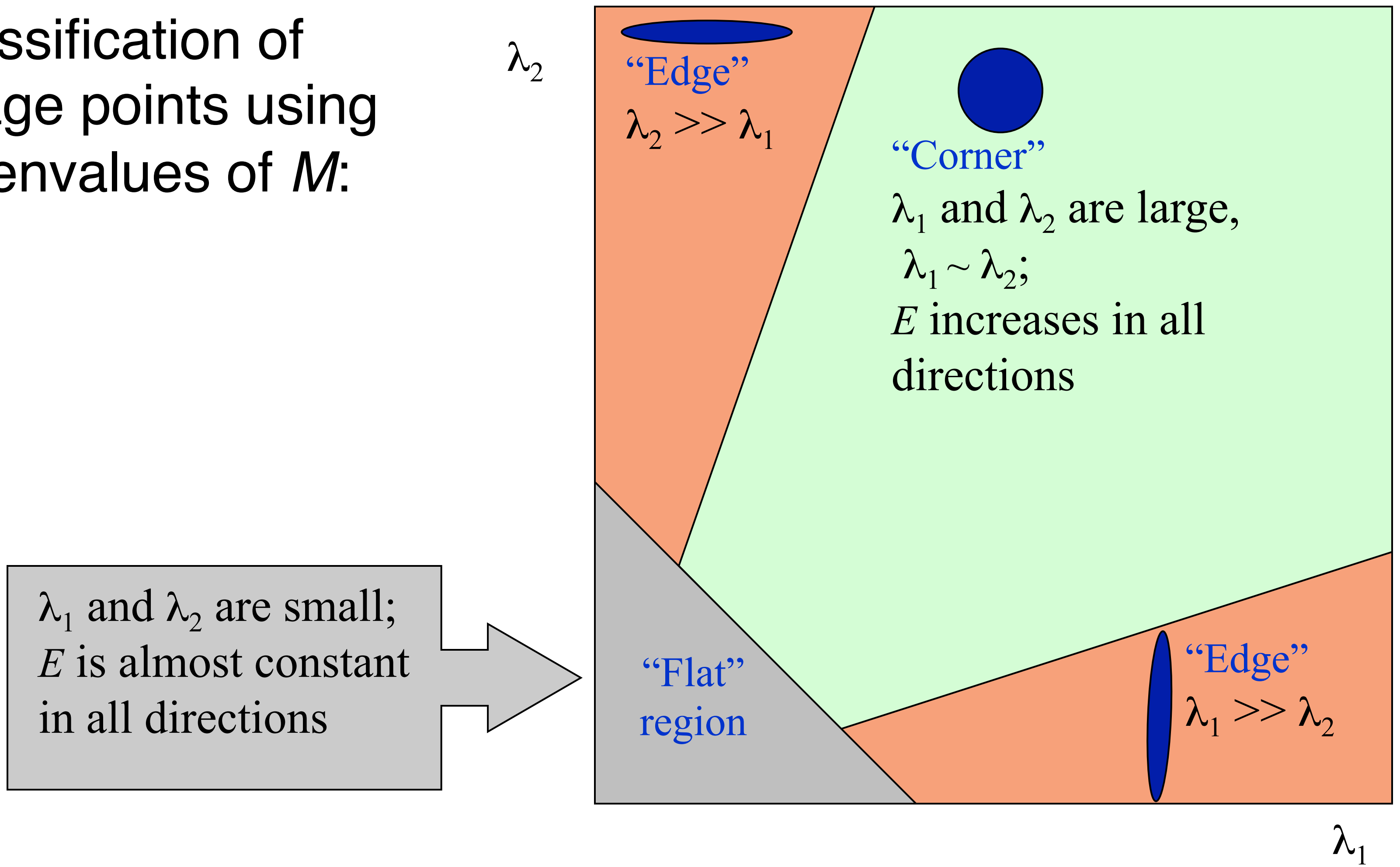
How are λ_+, x_+, λ_-, and x_+ relevant for feature detection?

- What's our feature scoring function?

Source: S. Seitz.

Feature detection: the math

Classification of image points using eigenvalues of *M*:



Source: Kokkinos, Saverese.

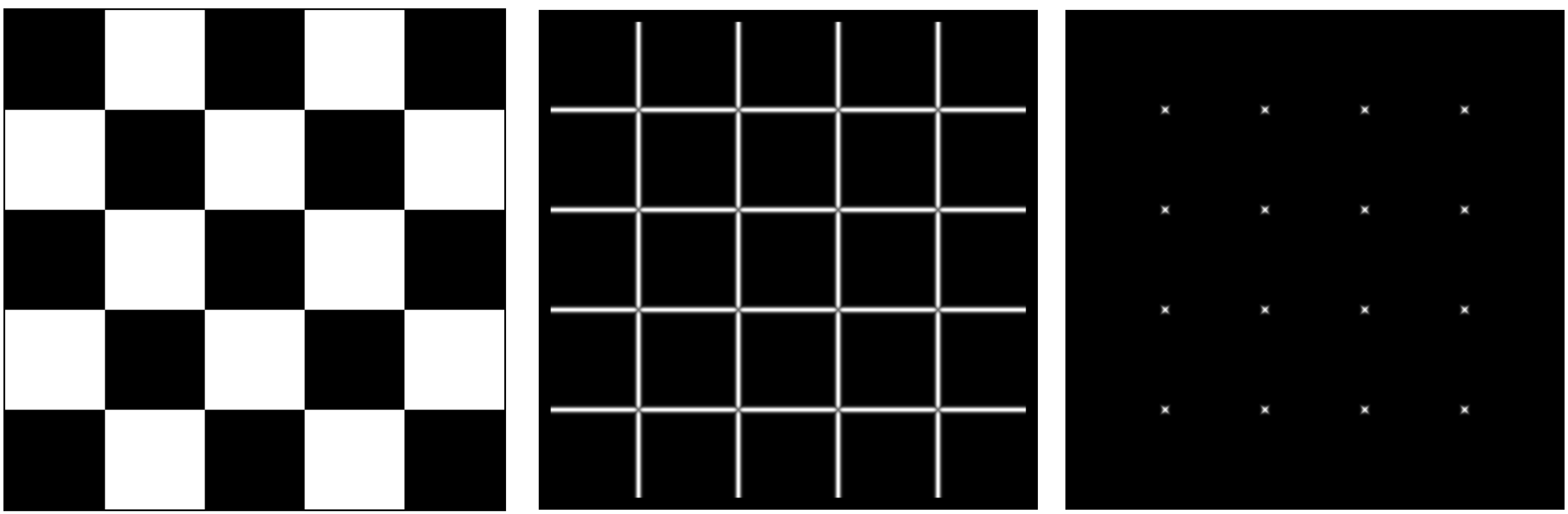
Feature detection: the math

How are λ_+, x_+, λ_-, and x_+ relevant for feature detection?

- What's our feature scoring function?

Want *E(u,v)* to be *large* for small shifts in *all* directions

- the *minimum* of *E(u,v)* should be large, over all unit vectors [u v]
- this minimum is given by the smaller eigenvalue (λ_-) of *H*



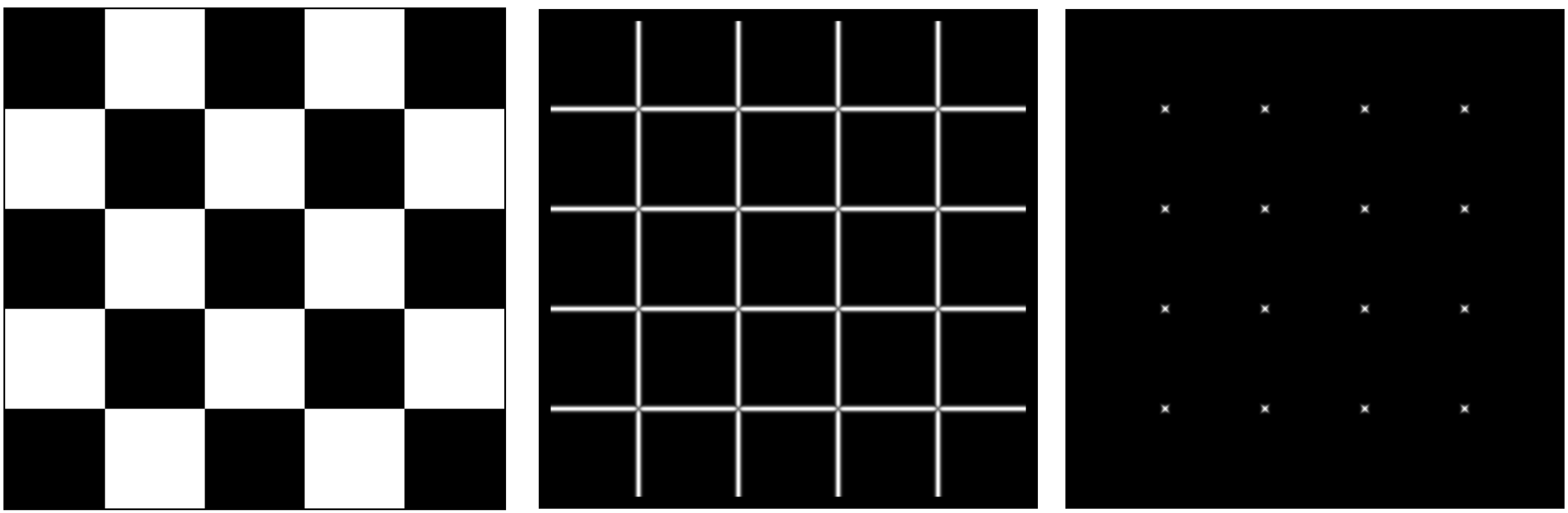
I λ_+ λ_-

Source: S. Seitz.

Feature detection summary

Here's what you do

- Compute the gradient at each point in the image
- Create the *H* matrix from the entries in the gradient
- Compute the eigenvalues.
- Find points with large response (λ_- > threshold)
- Choose those points where λ_- is a local maximum as features



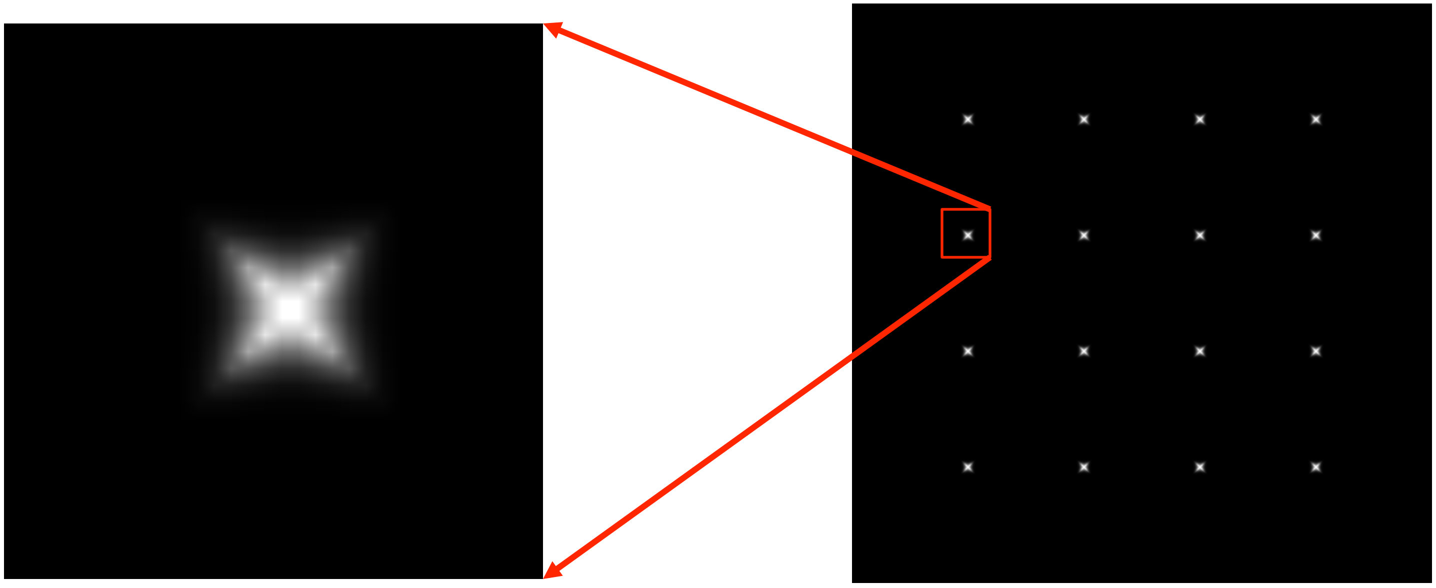
I λ_+ λ_-

Source: S. Seitz.

Feature detection summary

Here's what you do

- Compute the gradient at each point in the image
- Create the *H* matrix from the entries in the gradient
- Compute the eigenvalues.
- Find points with large response (λ_- > threshold)
- Choose those points where λ_- is a local maximum as features



λ_-

Source: S. Seitz.

The Harris operator

λ_- is a variant of the "Harris operator" for feature detection

$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

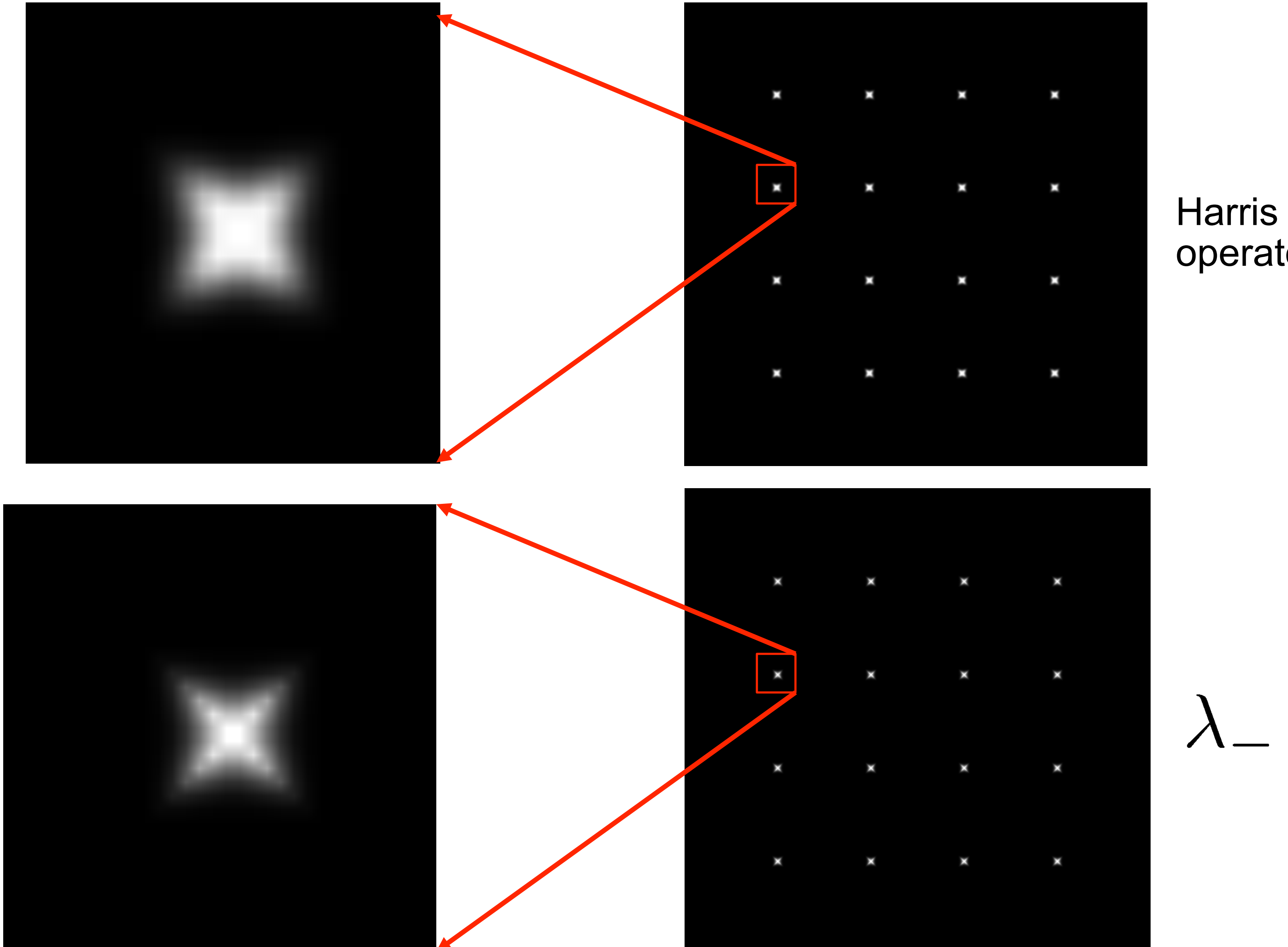
$$= \frac{determinant(H)}{trace(H)}$$

- The *trace* is the sum of the diagonals, i.e., $trace(H) = h_{11} + h_{22}$
- Very similar to λ_- but less expensive (no square root)
- Called the "Harris Corner Detector" or "Harris Operator"
- Lots of other detectors, this is one of the most popular

C.Harris and M.Stephens. "A Combined Corner and Edge Detector." *Proceedings of the 4th Alvey Vision Conference*: pages 147--151. 1988.

Source: S. Seitz.

The Harris operator

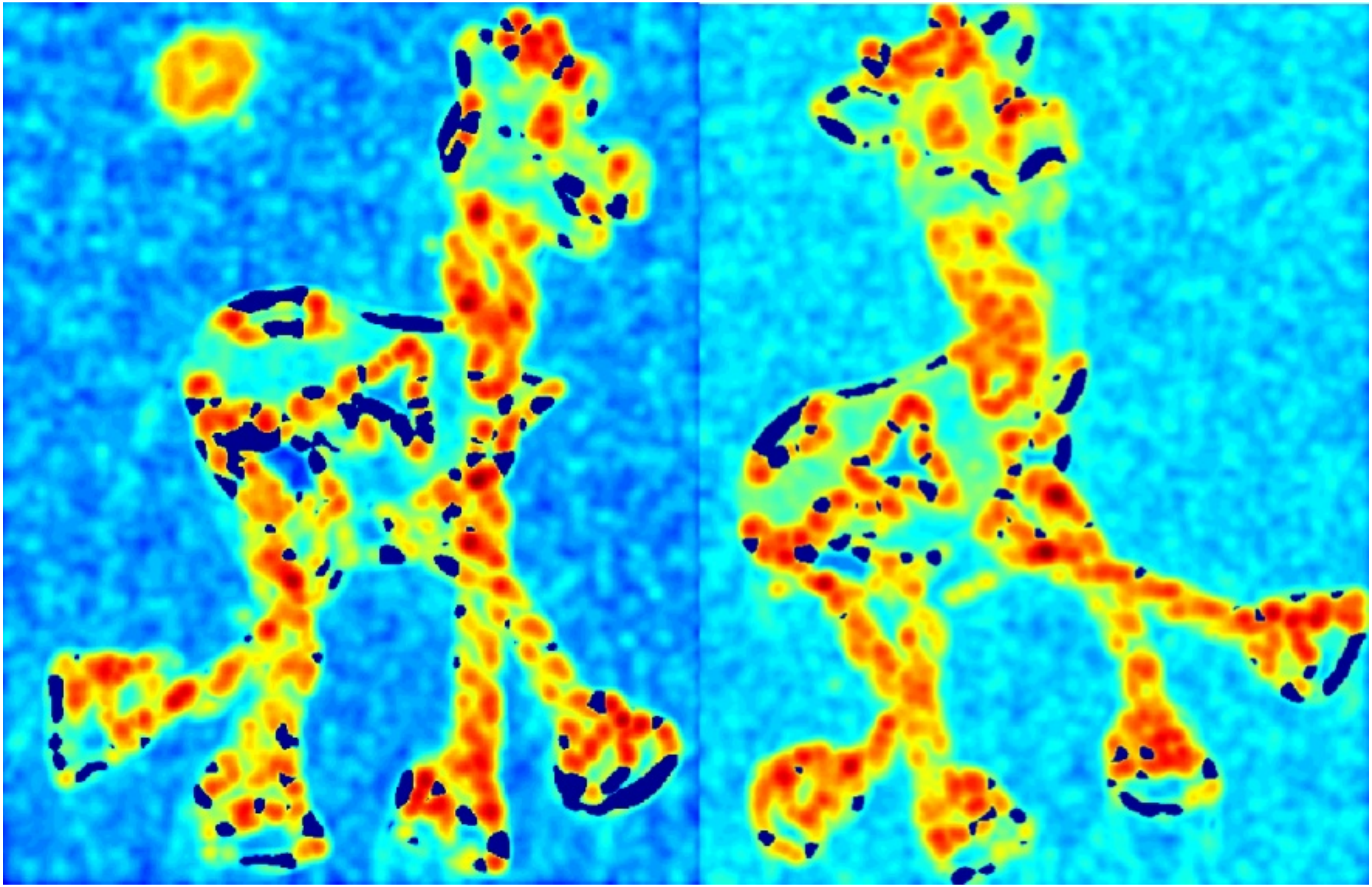


Source: S. Seitz.

Harris detector example

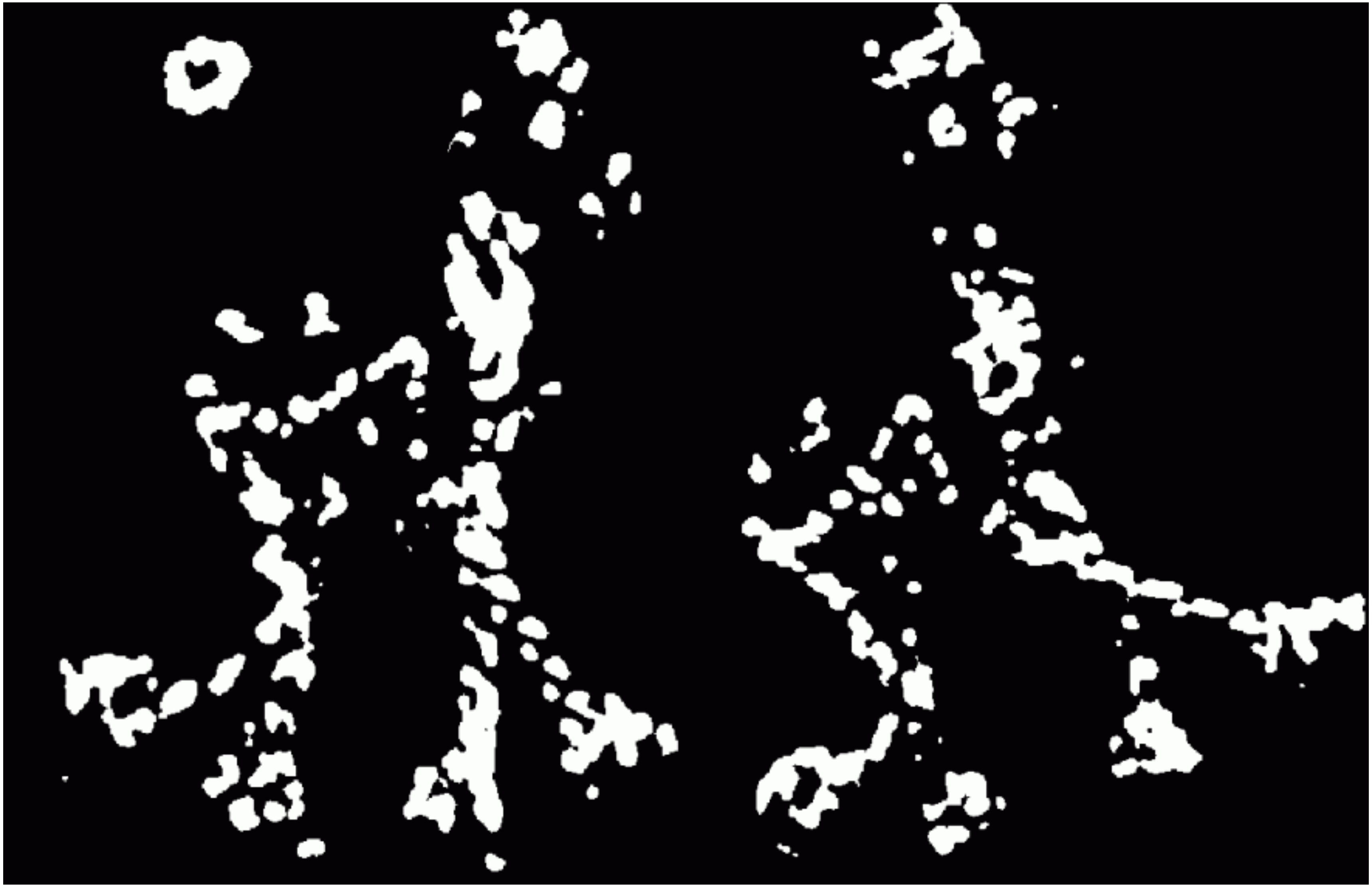
Source: S. Seitz.

f value (red high, blue low)



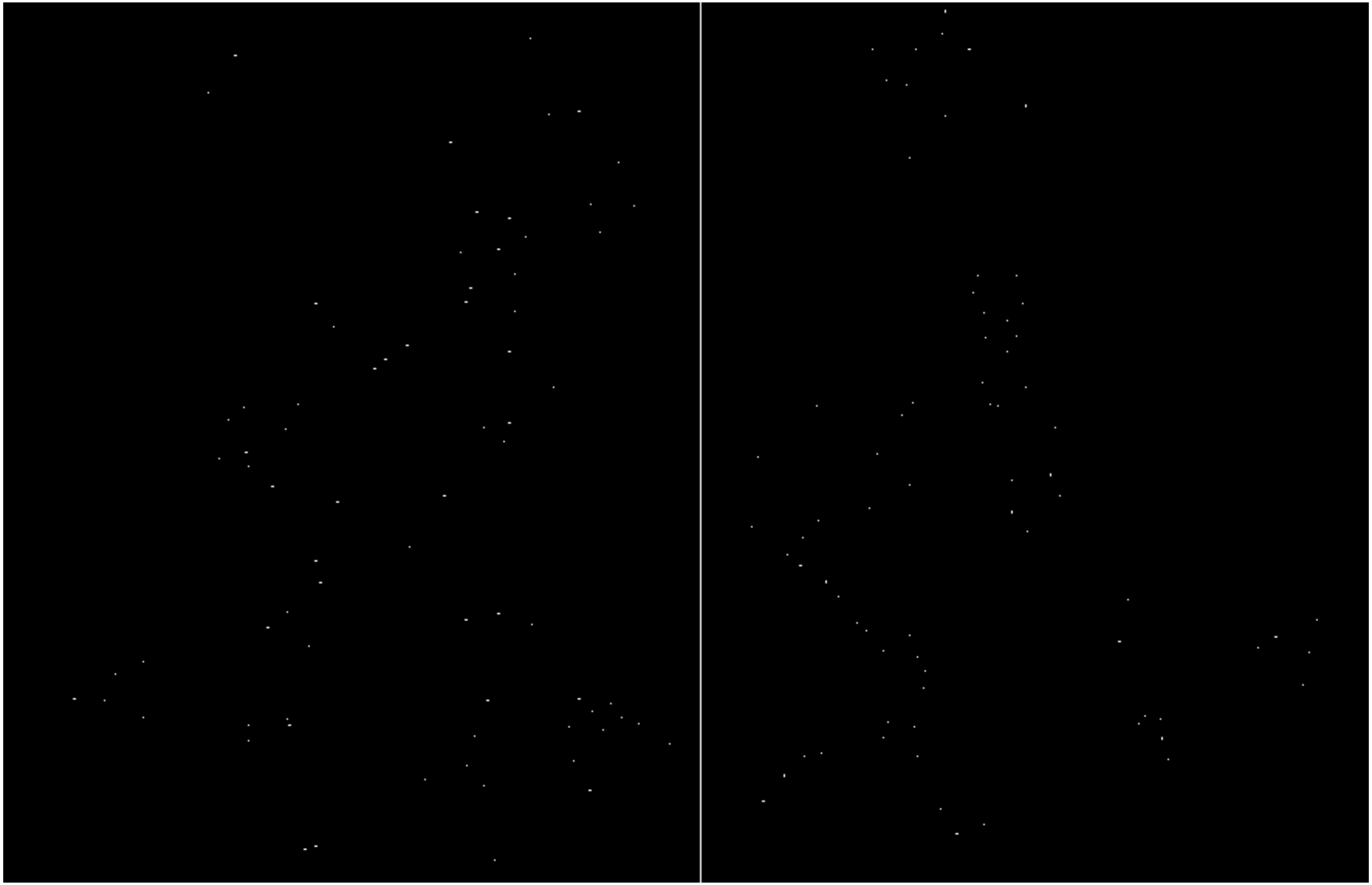
Source: S. Seitz.

Threshold (f > value)



Source: S. Seitz.

Find local maxima of f



Source: S. Seitz.

Harris features (in red)

Towards Invariance

Suppose you **rotate** the image by some angle

- Will you still pick up the same features?

What if you change the brightness?

Scale?

Invariance defined:

Suppose we are comparing two images I and J.

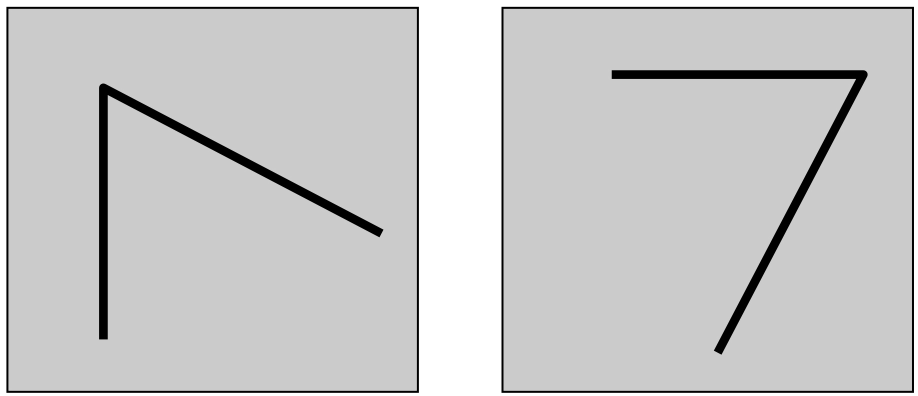
J may be a transformed version of I

We want to detect the same features from I and J regardless of the transformation: this is **transformational invariance**.

Source: S. Seitz.

Harris Detector: Some Properties

- Is the Harris detector rotationally invariant?



***Corner response R* is invariant to image rotation**

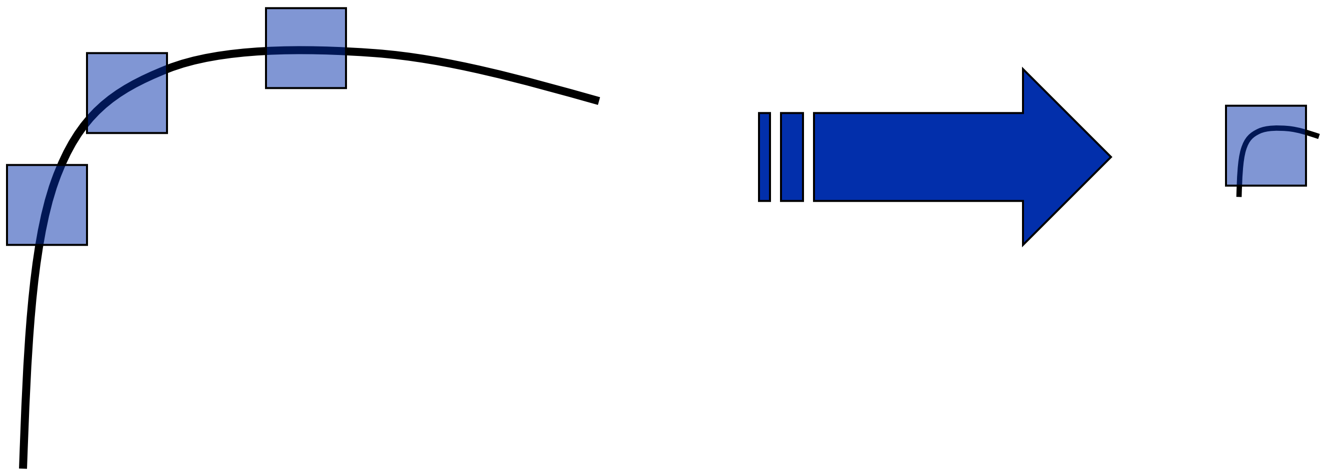
$$H = U^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U \rightarrow f(\lambda_1, \lambda_2)$$ doesn't change!

Source: S. Savarese.

Harris Detector: Some Properties

- Is it scale invariant?

Corner response R **is not scale invariant!**



All points will be classified as edges

Corner !

Source: S. Savarese.

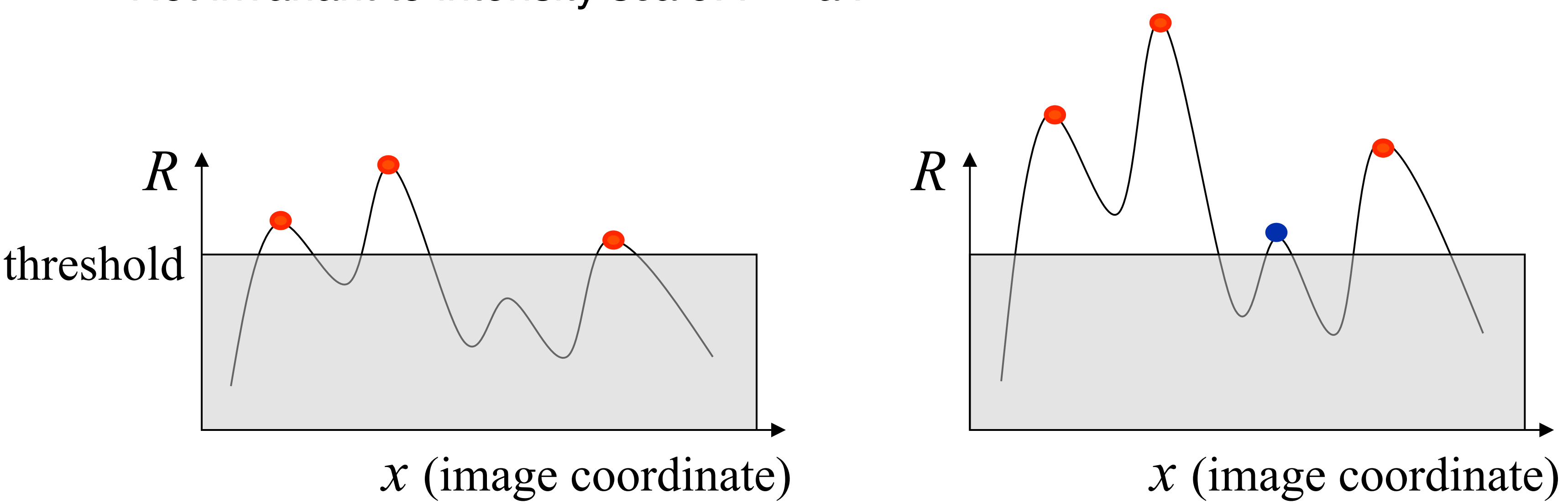
Harris Detector: Some Properties

- Partial invariance to *affine intensity* changes

$I \rightarrow s\,I + b$

- invariance to intensity shift $I \rightarrow I + b$ *(why?)*
  (only derivatives are used)

- Not invariant to intensity scale: $I \rightarrow a\,I$



Source: S. Savarese.

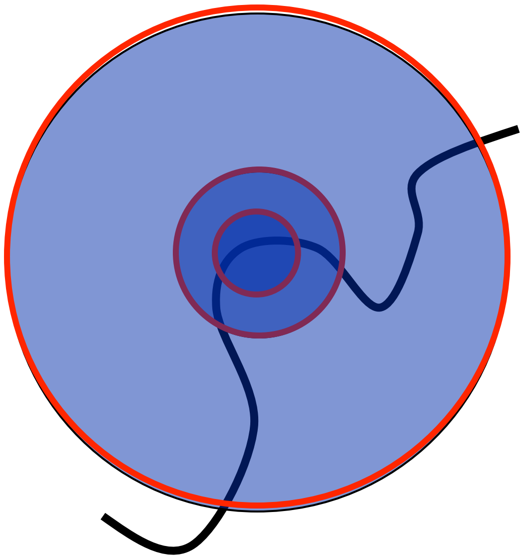
Invariance

Detector	Illumination	Rotation	Scale	View point
Harris corner	partial	Yes	No	No

Source: S. Savarese.

Scale invariant detection

Suppose you're looking for corners



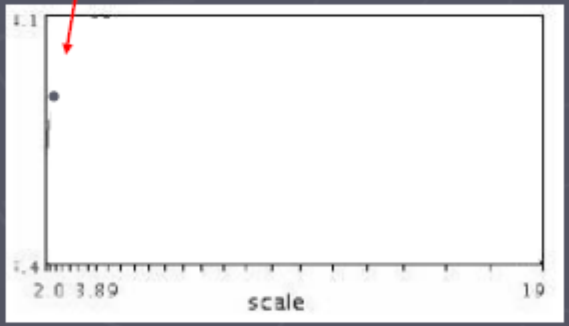
Key idea: find scale that gives local maximum of f

- f is a local maximum in both position and scale
- Common definition of f: Laplacian
  (or difference between two Gaussian filtered images with different sigmas)

Source: S. Seitz.

Automatic scale selection

Lindeberg et al., 1996



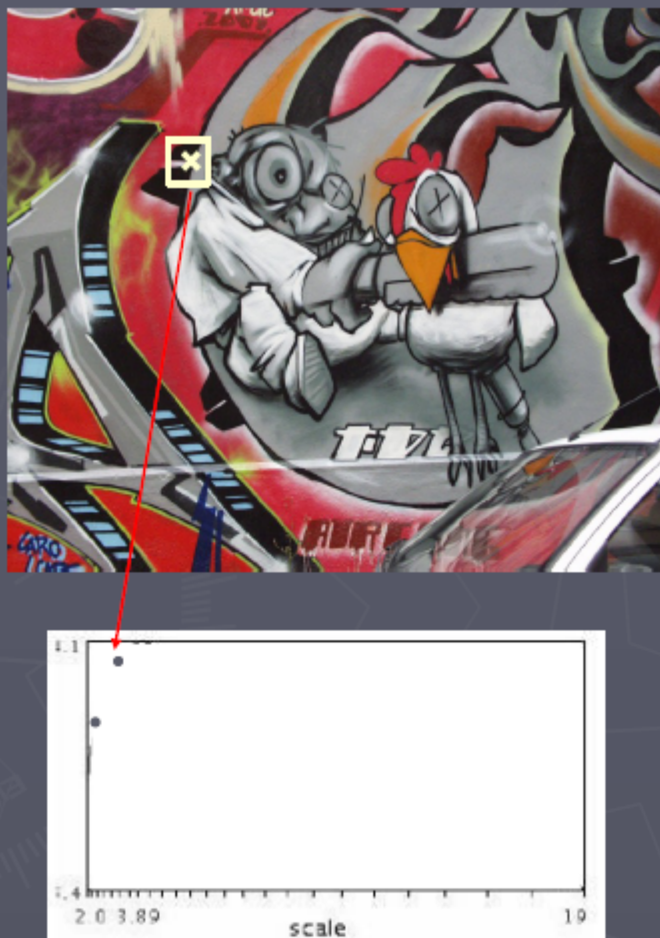
$f(I_{i_1 \ldots i_m}(x,\sigma))$

Slide from Tinne Tuytelaars

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)



$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

Automatic scale selection

Function responses for increasing scale

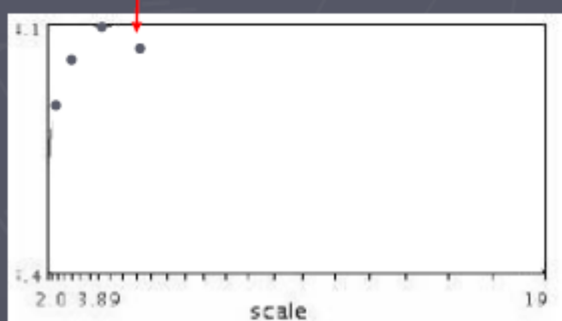
Scale trace (signature)

$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

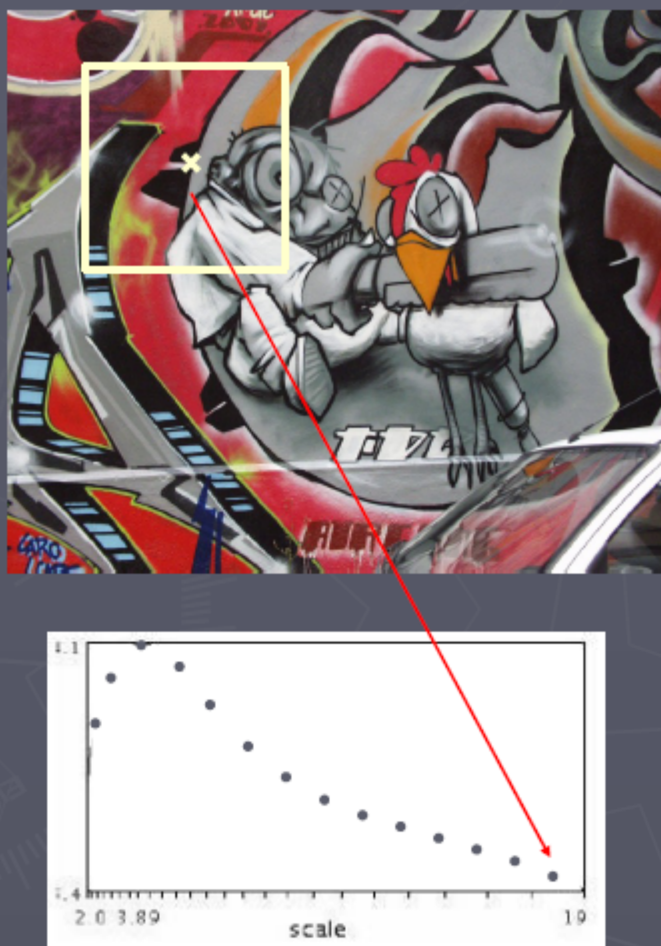


$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

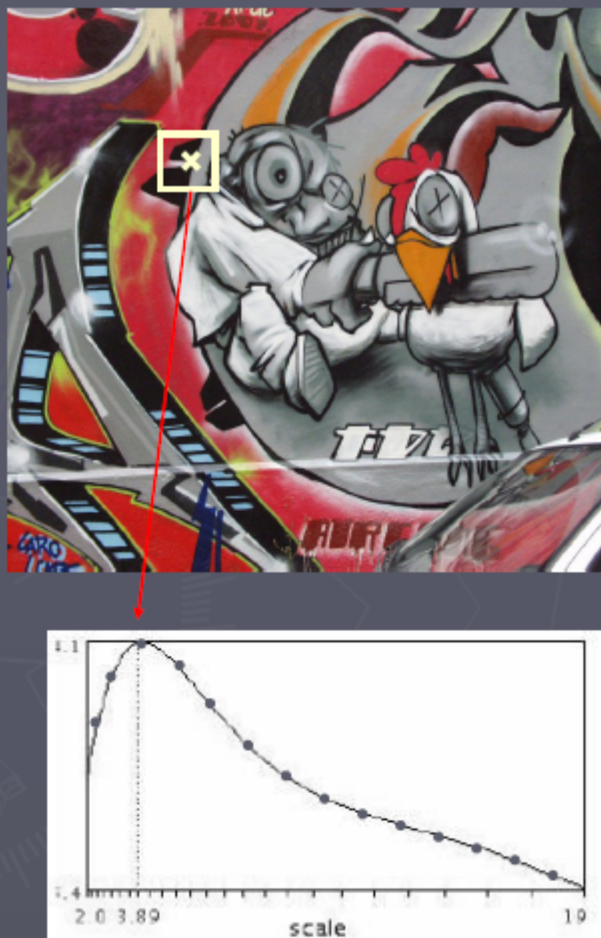


$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

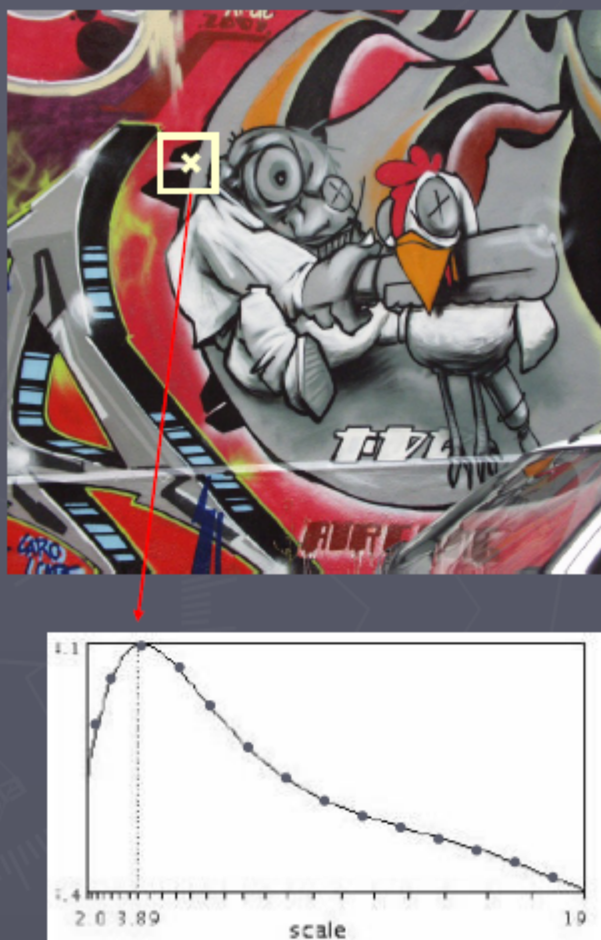


$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

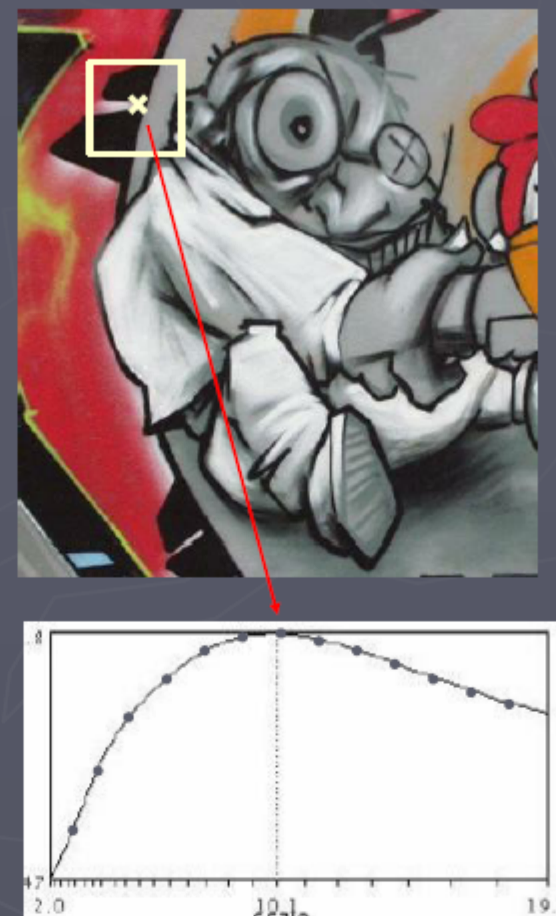
Automatic scale selection

Function responses for increasing scale

Scale trace (signature)



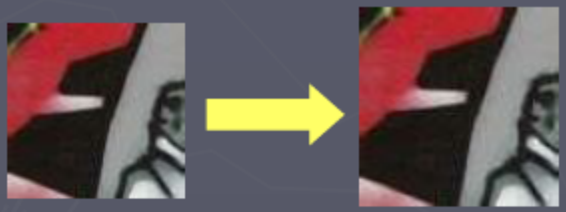
$f(I_{i_1 \ldots i_m}(x,\sigma))$

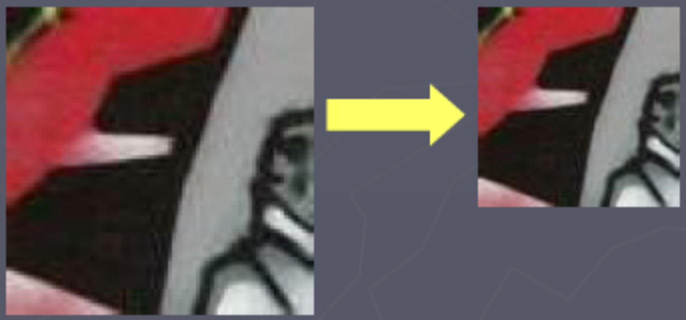


$f(I_{i_1 \ldots i_m}(x',\sigma'))$

Automatic scale selection

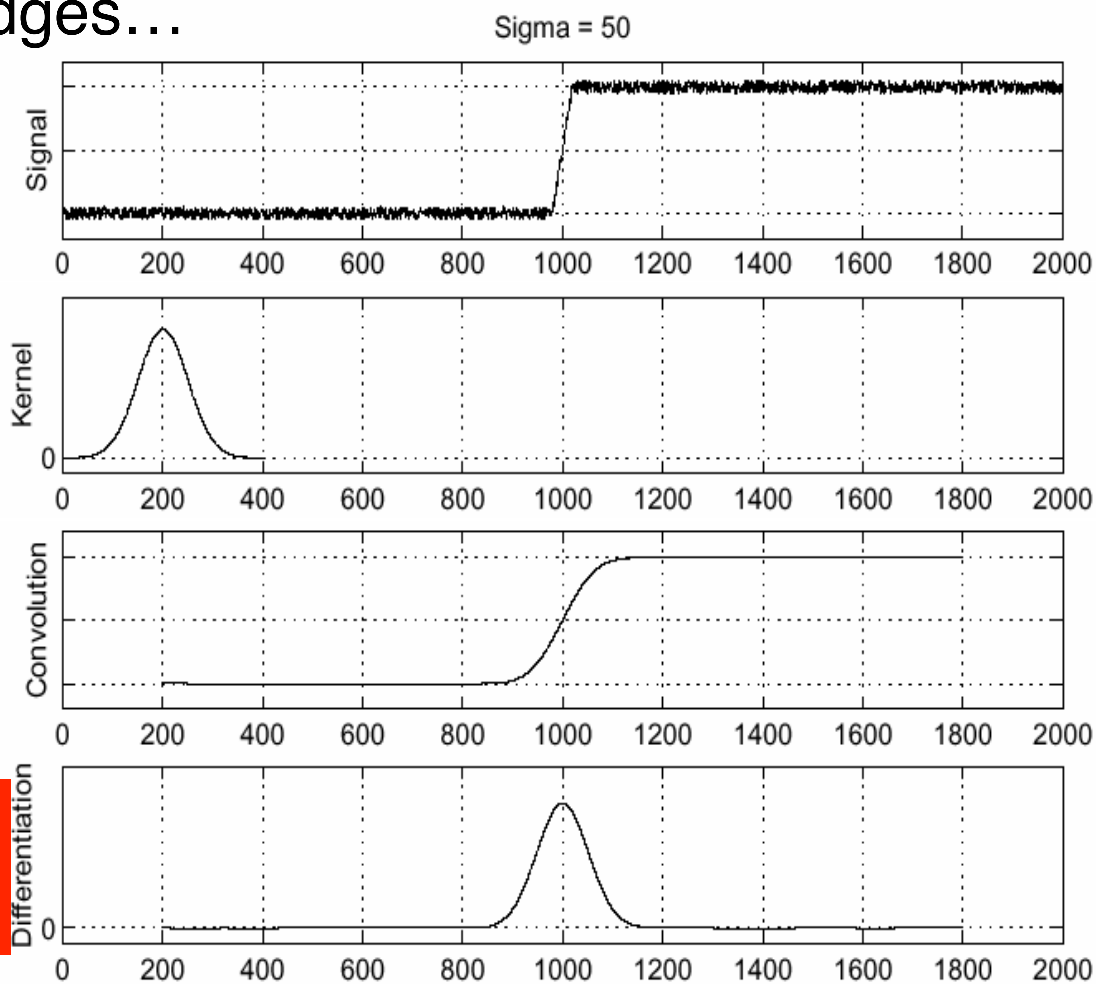
Normalize: rescale to fixed size





Scale-Invariant Feature Detection Example

- Recall: Edges…



f

g

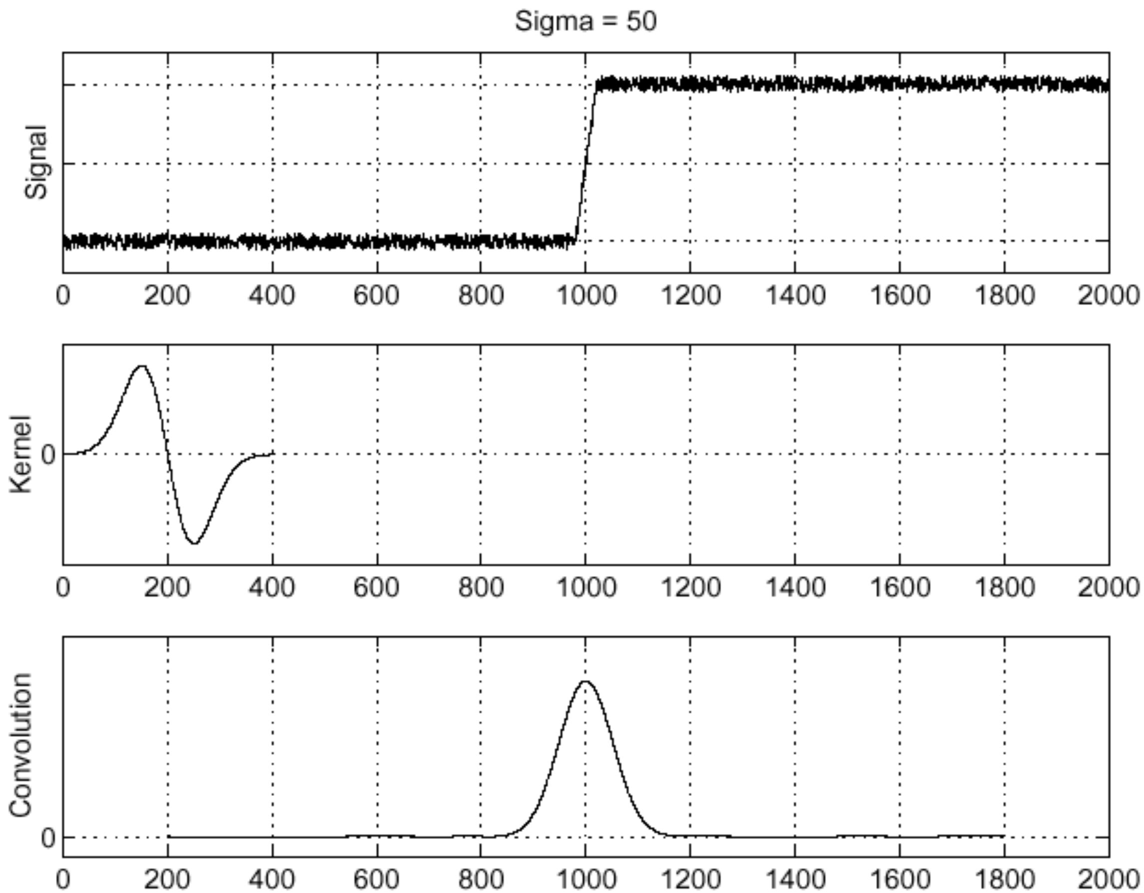
$f * g$

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

$= \frac{dg}{dx} * f$ DoG filter

Source: S. Seitz.

Edge detection



$$f$$

Edge

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

Derivative of Gaussian

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

Edge = maximum of derivative

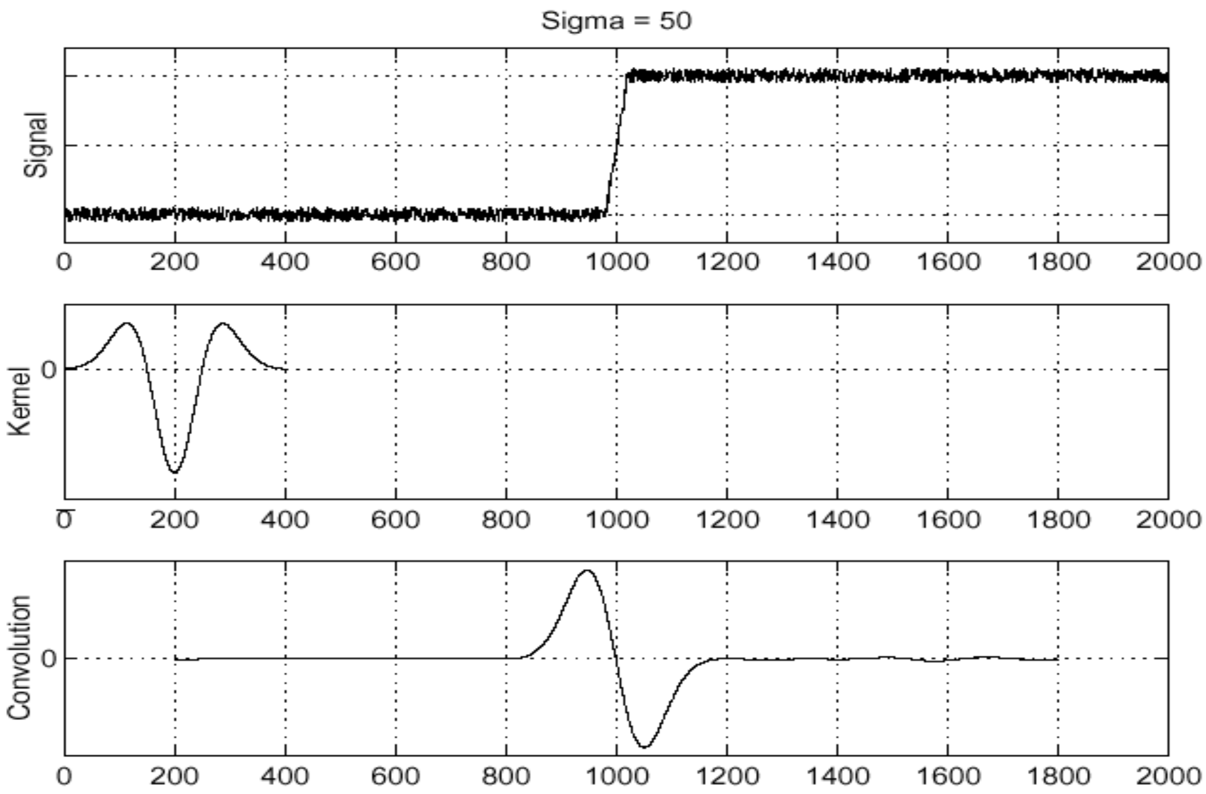
Source: S. Seitz.

Edge detection as zero crossing

f

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

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



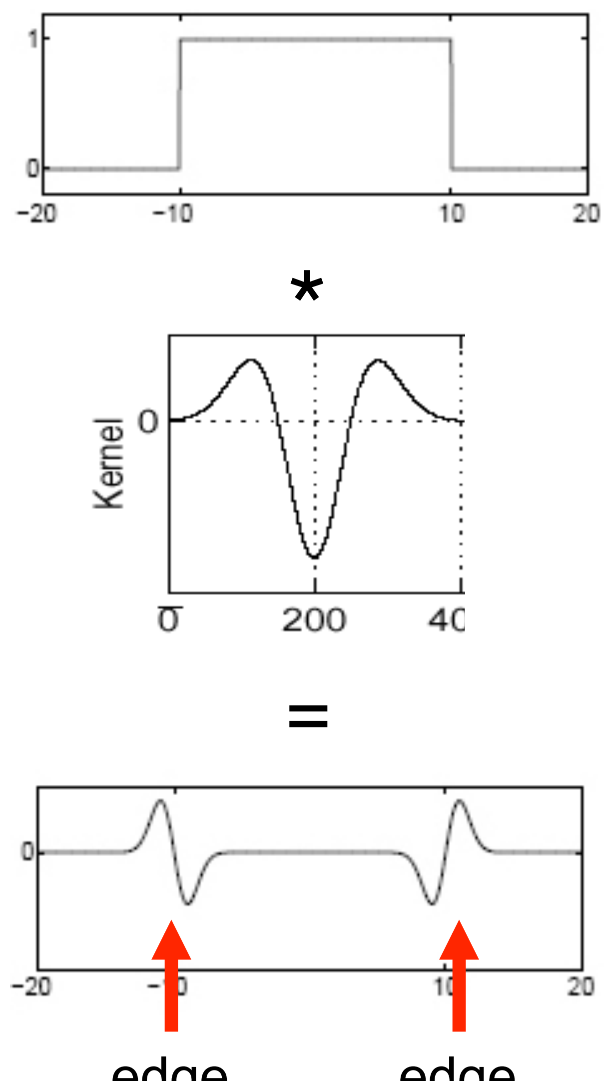
Edge

Second derivative of Gaussian (Laplacian)

Edge = zero crossing of second derivative

Source: S. Seitz.

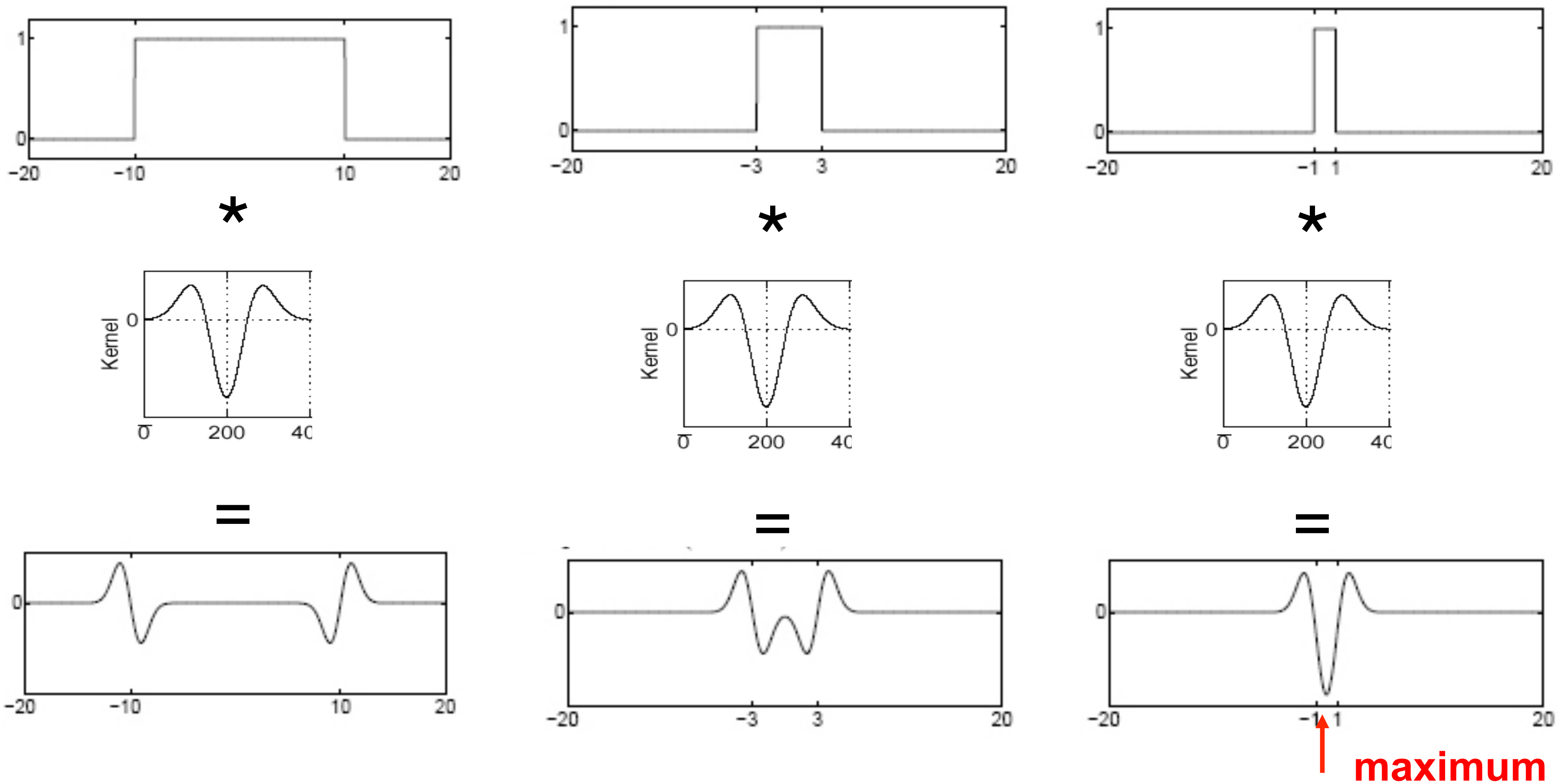
Edge detection as zero crossing



Source: S. Seitz.

From edges to blobs

- Blob = superposition of nearby edges

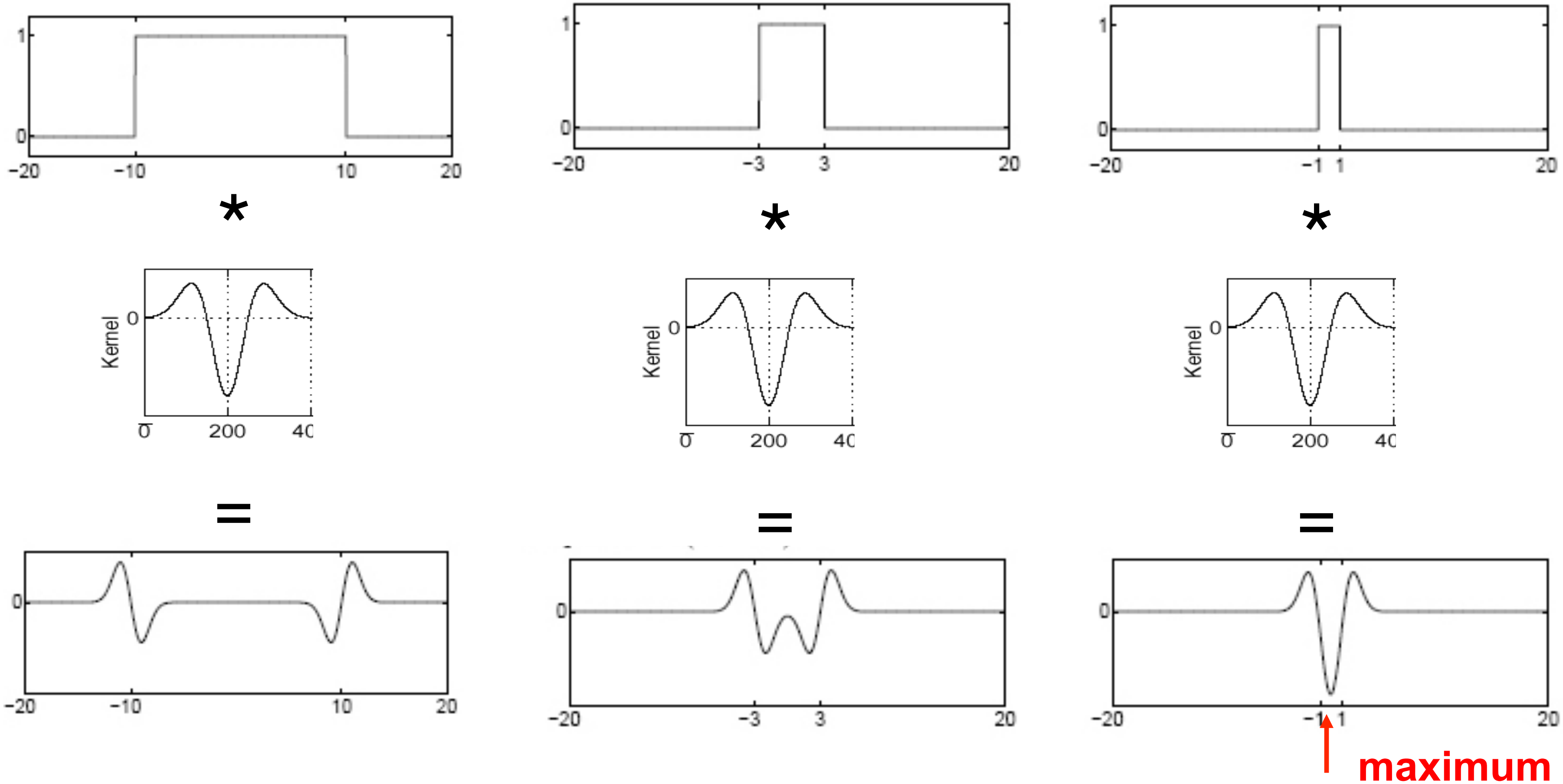


Magnitude of the Laplacian response achieves a maximum at the center of the blob, provided the scale of the Laplacian is “matched” to the scale of the blob

Source: S. Seitz.

From edges to blobs

- Blob = superposition of nearby edges

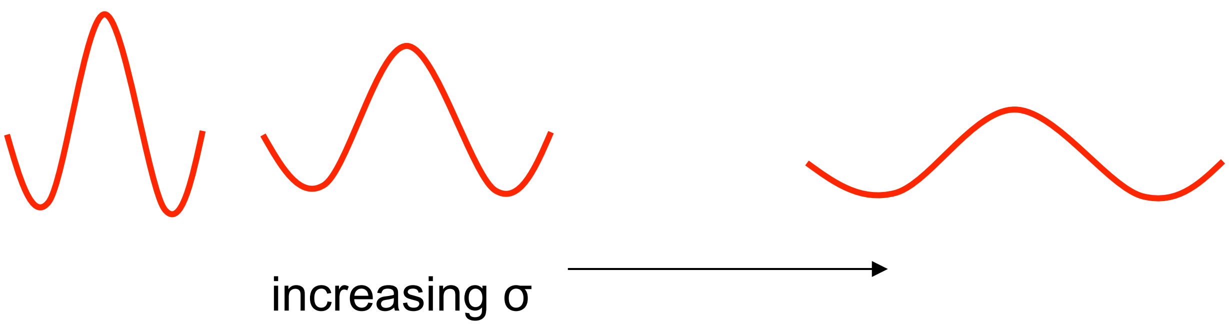


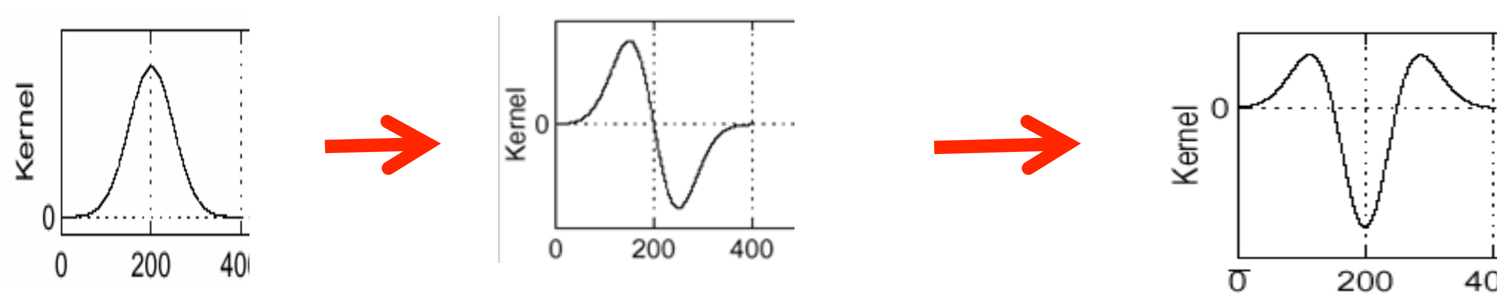
What if the blob is slightly thicker or slimmer?

Source: S. Seitz.

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





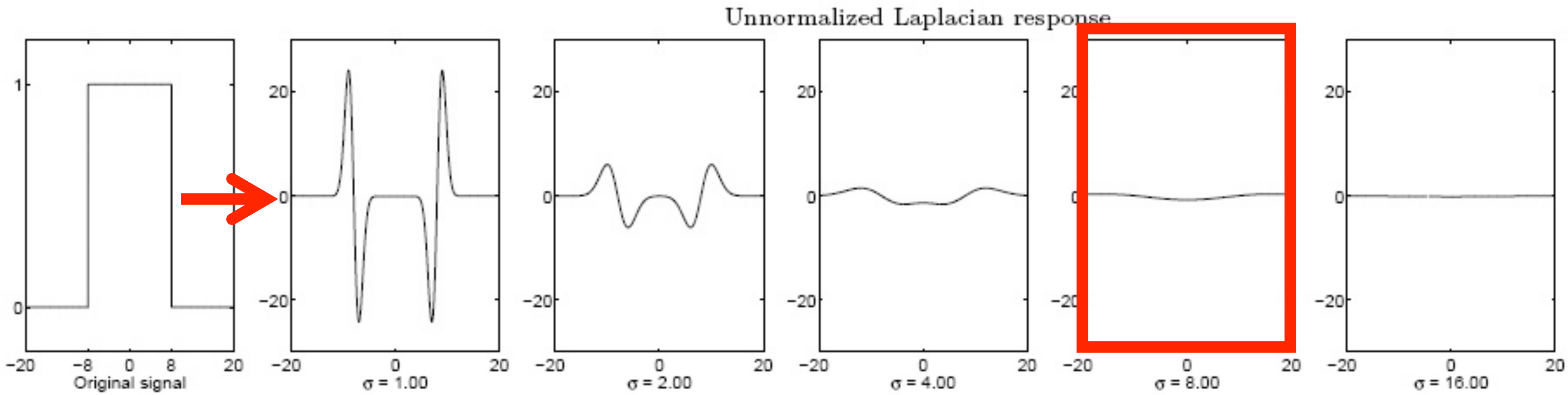
$$g \qquad \frac{d}{dx}g \qquad \frac{d^2}{dx^2}g$$

Source: S. Seitz.

Scale selection

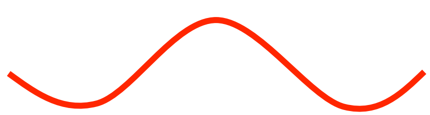
- We want to find the characteristic scale of the blob by convolving it with Laplacians at several scales and looking for the maximum response

•However, Laplacian response decays as scale increases:



original signal
(radius=8)

This should give the max response ☹

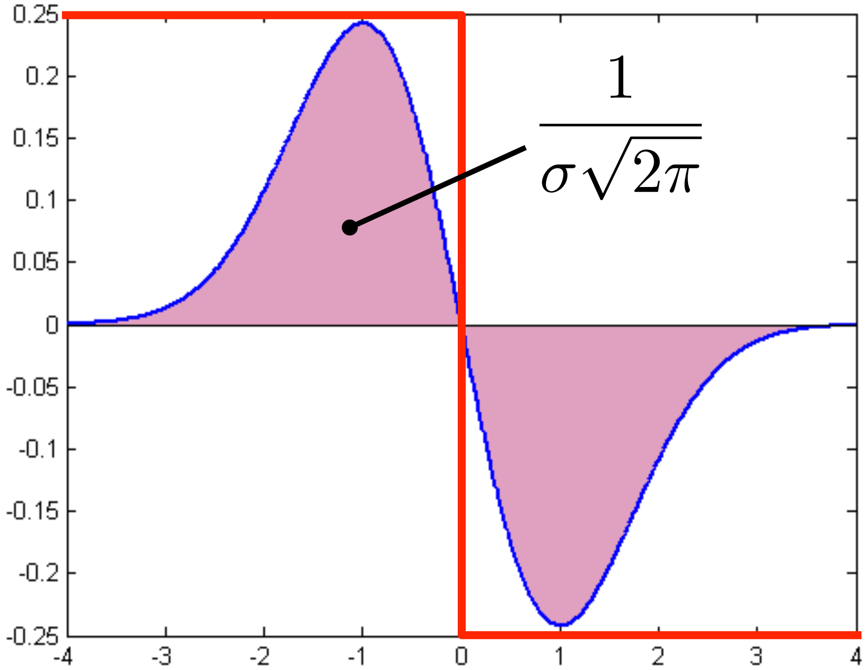


increasing σ →

Source: S. Seitz.

Scale normalization

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



Source: S. Seitz.

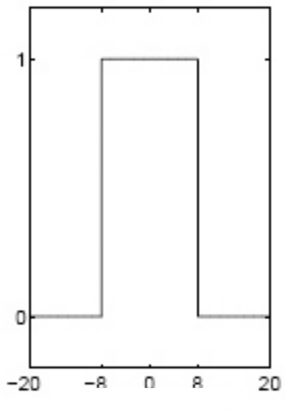
Scale normalization

- 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

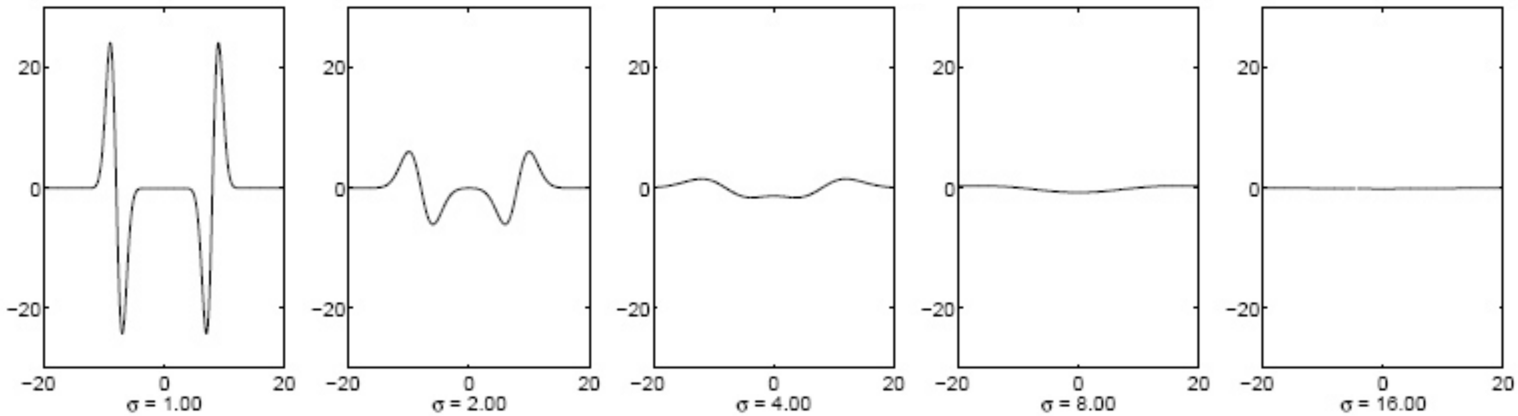
Source: S. Seitz, S. Savarese.

Effect of scale normalization

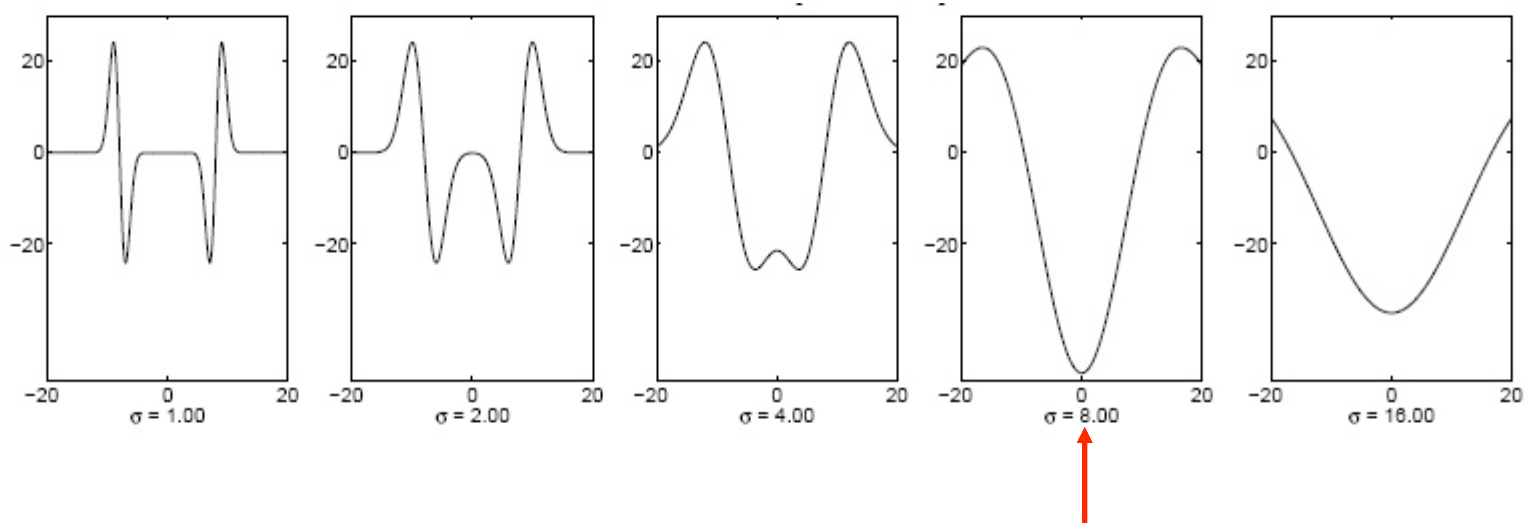
Original signal



Unnormalized Laplacian response



Scale-normalized Laplacian response

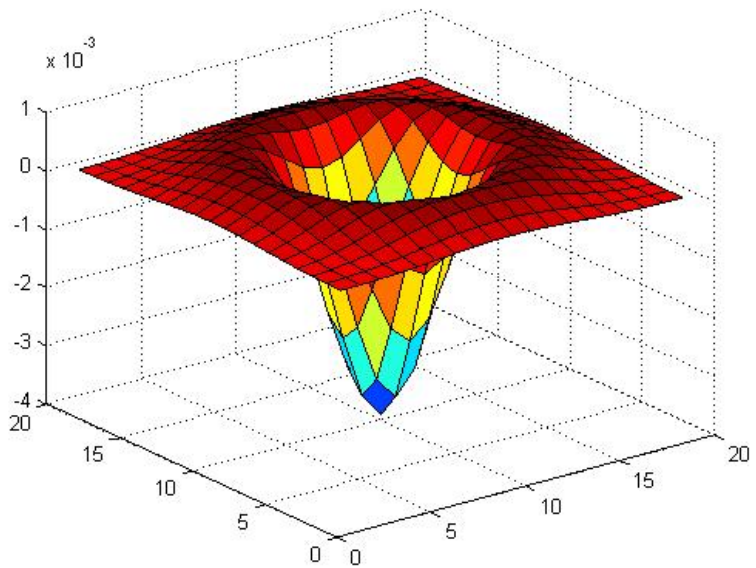


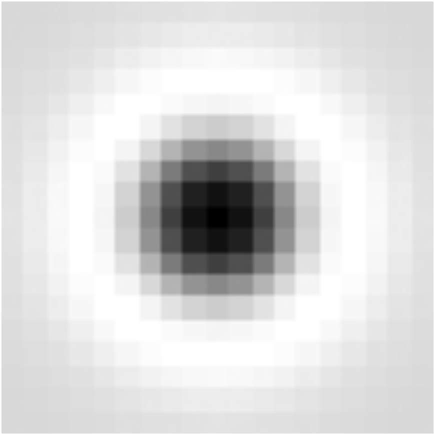
Maximum ☺

Source: S. Seitz, S. Savarese.

Blob detection in 2D

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



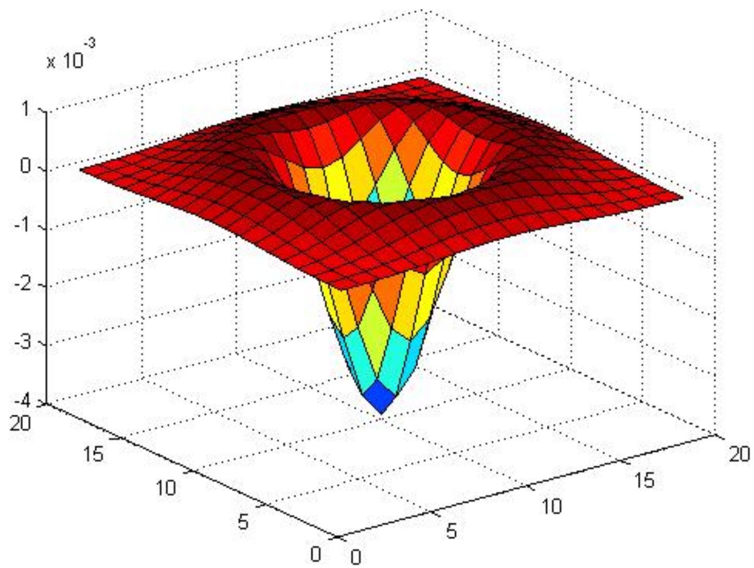


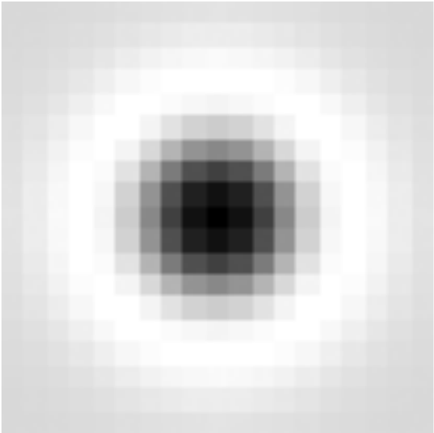
$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

Source: S. Seitz, S. Savarese.

Blob detection in 2D

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



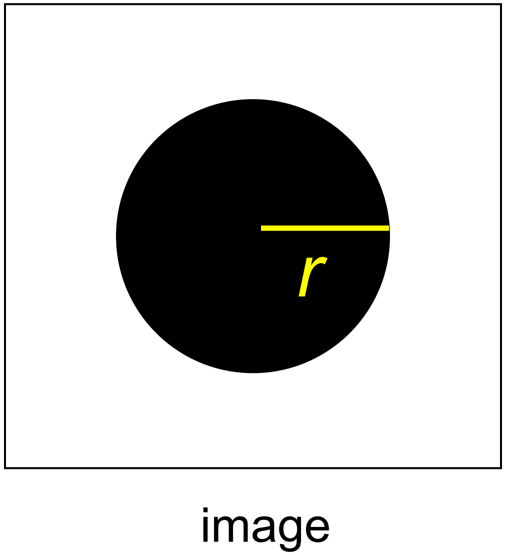


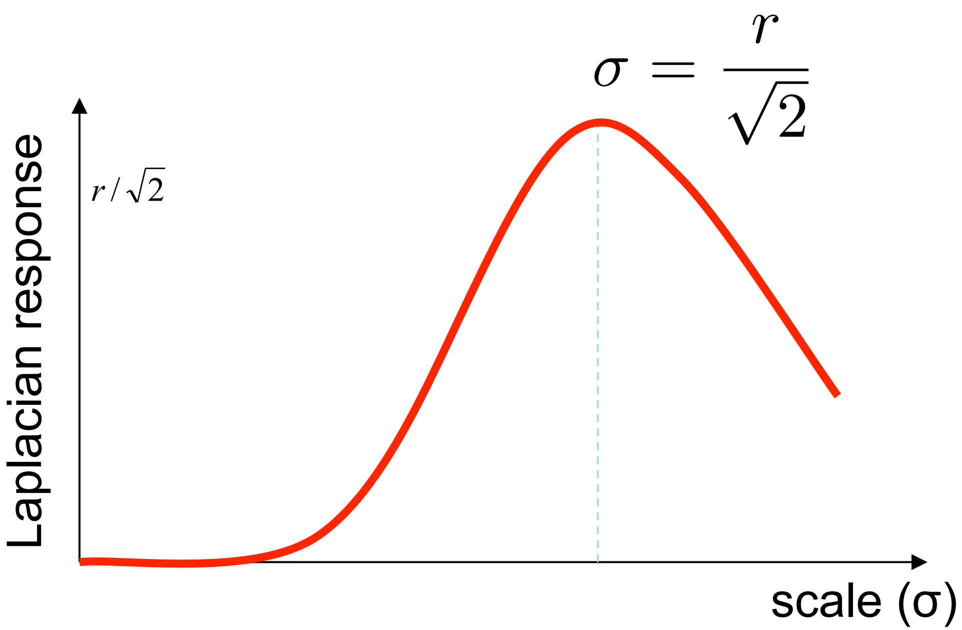
Scale-normalized: $\nabla^2_{\text{norm}} g = \sigma^2 \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$

Source: S. Seitz, S. Savarese.

Scale selection

- For a binary circle of radius r , the Laplacian achieves a maximum at

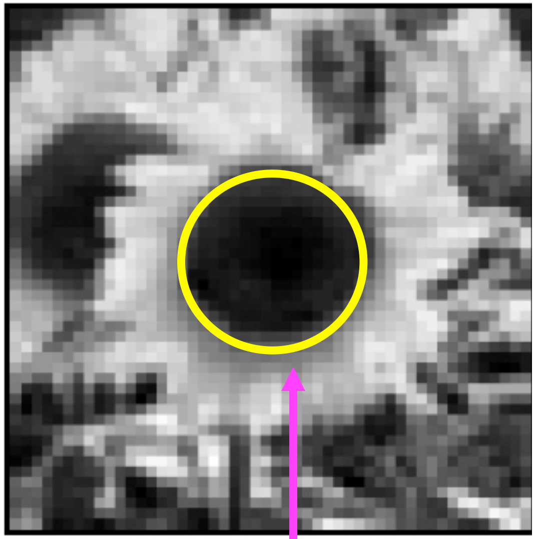


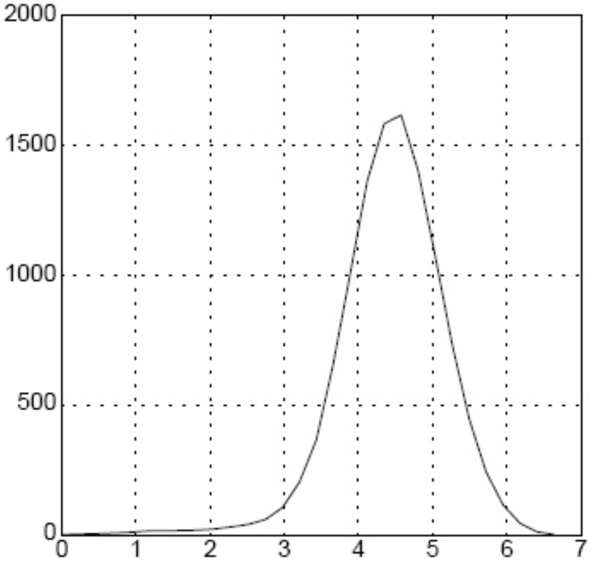


Source: S. Seitz, S. Savarese.

Characteristic scale

- We define the characteristic scale as the scale that produces peak of Laplacian response





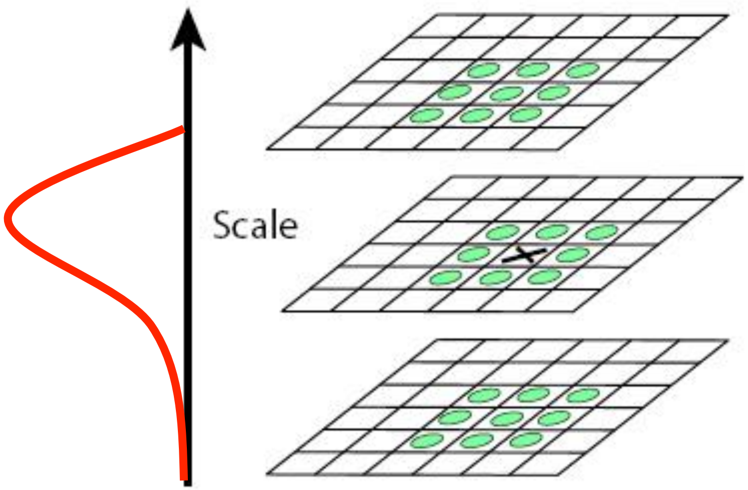
characteristic scale

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

Source: S. Seitz, S. Savarese.

Scale-space blob detector

1. Convolve image with scale-normalized Laplacian at several scales
2. Find maxima of squared Laplacian response in scale-space
3. This indicates if a blob has been detected
4. And what is its intrinsic scale



Source: S. Seitz, S. Savarese.

Scale-space blob detector: example

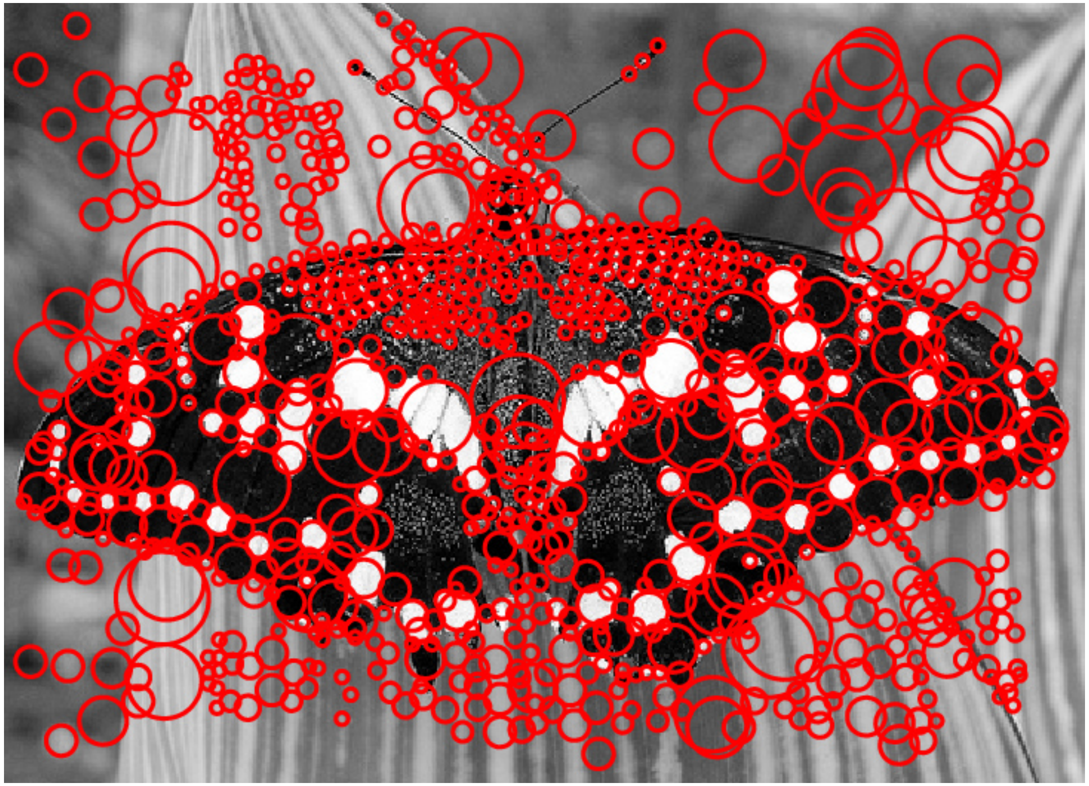
Source: S. Seitz, S. Savarese.

Scale-space blob detector: example

sigma = 11.9912

Source: S. Seitz, S. Savarese.

Scale-space blob detector: example



Source: S. Seitz, S. Savarese.

Difference of Gaussians Approximations to Laplacian

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04

- Approximating the Laplacian with a difference of Gaussians:

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

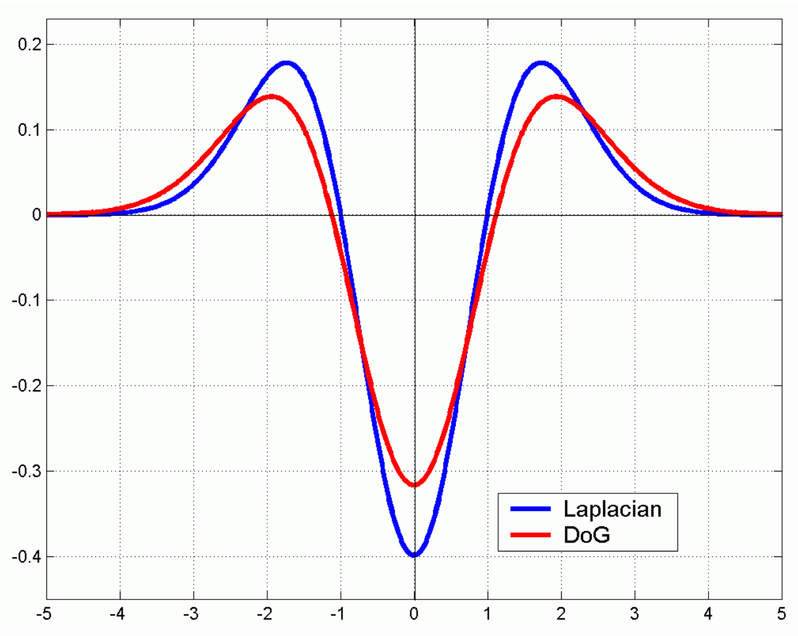
Laplacian

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

Difference of Gaussians

or

Difference of gaussian blurred images at scales k σ and σ

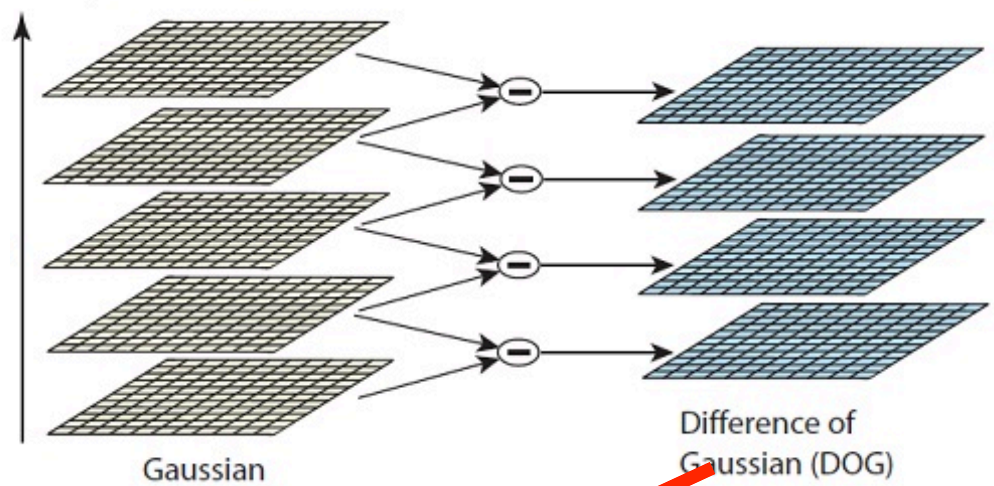


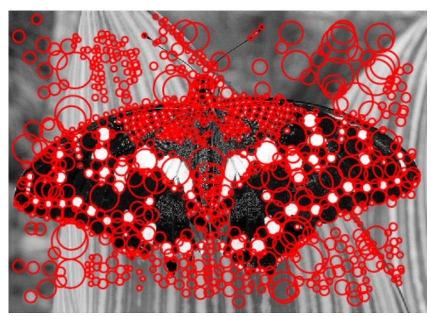
$$G(x,y,k\sigma) - G(x,y,\sigma) \approx (k-1)\sigma^2 L$$

Source: S. Seitz, S. Savarese.

Difference of Gaussians (DoG)

k





Output: location, scale, orientation (more later)

Source: S. Seitz, S. Savarese.

Invariance

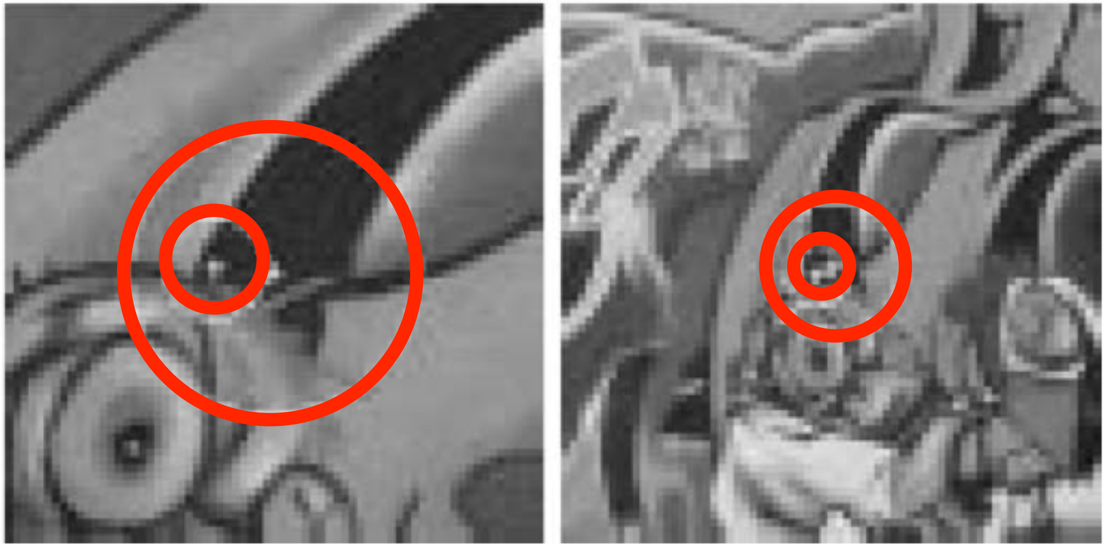
Detector	Illumination	Rotation	Scale	View point
Harris corner	Yes	Yes	No	No
Lowe '99 (DoG)	Yes	Yes	Yes	No

Source: S. Savarese.

Harris-Laplace

[Mikolajczyk & Schmid ’01]

- Collect locations (x,y) of detected Harris features for $\sigma = \sigma_1 \ldots \sigma_2$ (the sigma is here comes from g_x, g_y)
- For each detected location (x,y) and for each σ, reject detection if Laplacian(x,y, σ) is not a local maximum



Output: location, scale

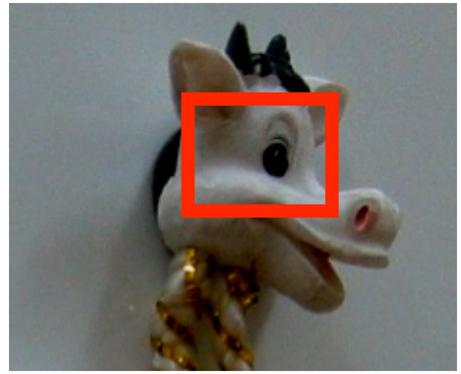
Source: S. Savarese.

Invariance

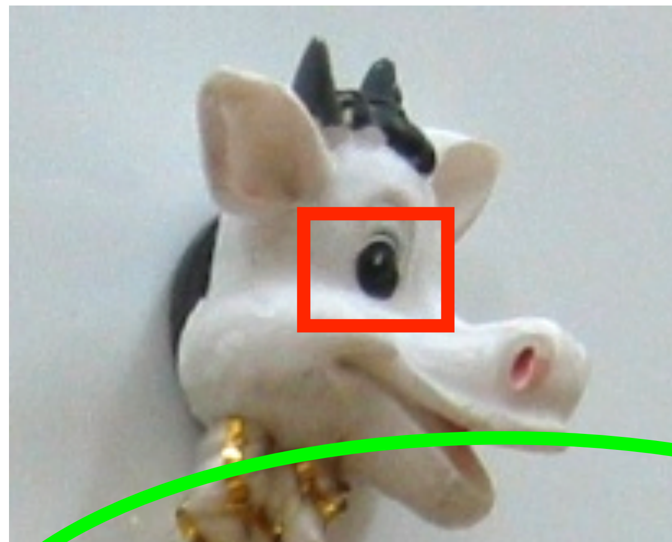
Detector	Illumination	Rotation	Scale	View point
Harris corner	Yes	Yes	No	No
Lowe '99 (DoG)	Yes	Yes	Yes	No
Mikolajczyk & Schmid '01	Yes	Yes	Yes	No

Source: S. Savarese.

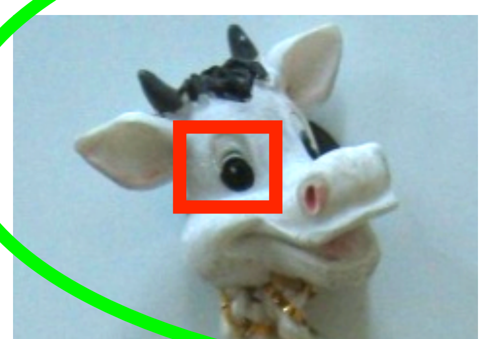
Repeatability



Illumination invariance

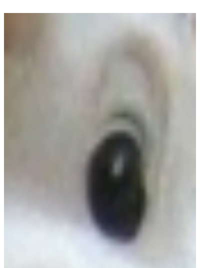


Scale invariance



Pose invariance

- Rotation
- Affine

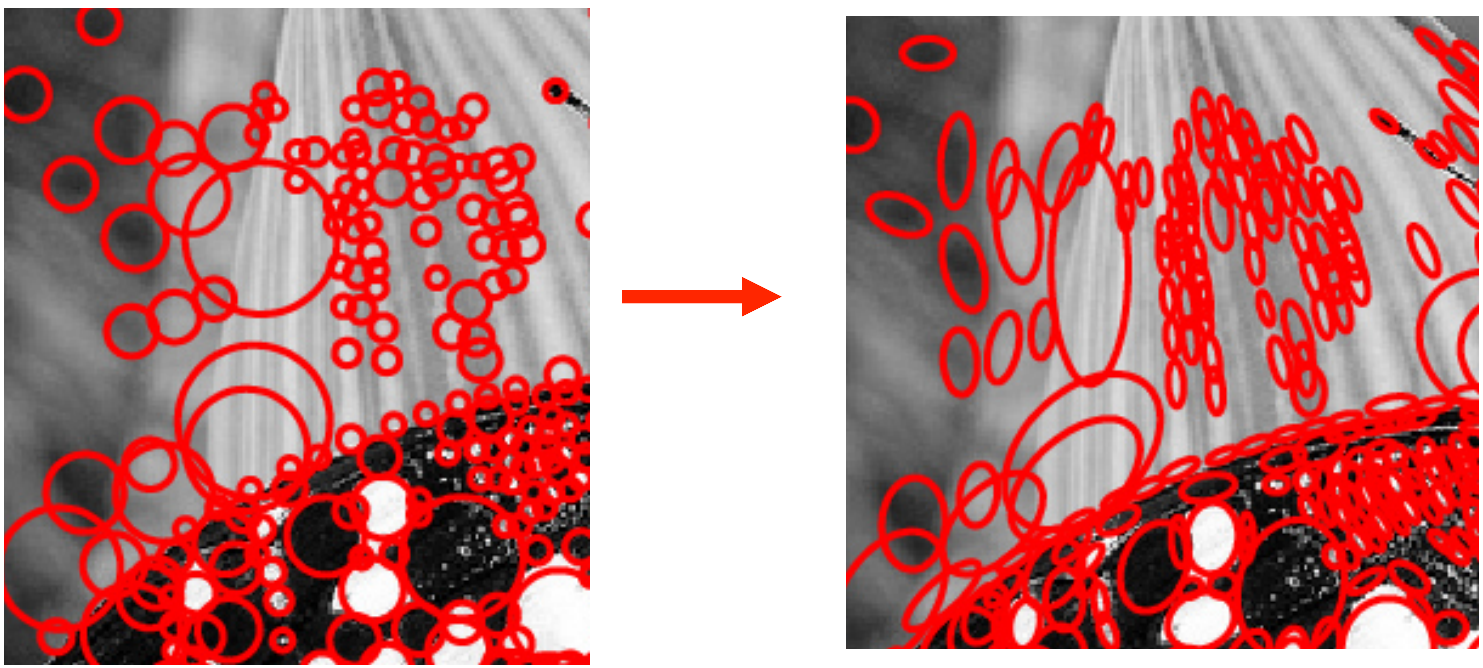


Source: S. Savarese.

Affine invariance

K. Mikolajczyk and C. Schmid,
Scale and Affine invariant interest point detectors, IJCV 60(1):63-86, 2004.

Similarly to characteristic scale selection, detect the characteristic shape of the local feature

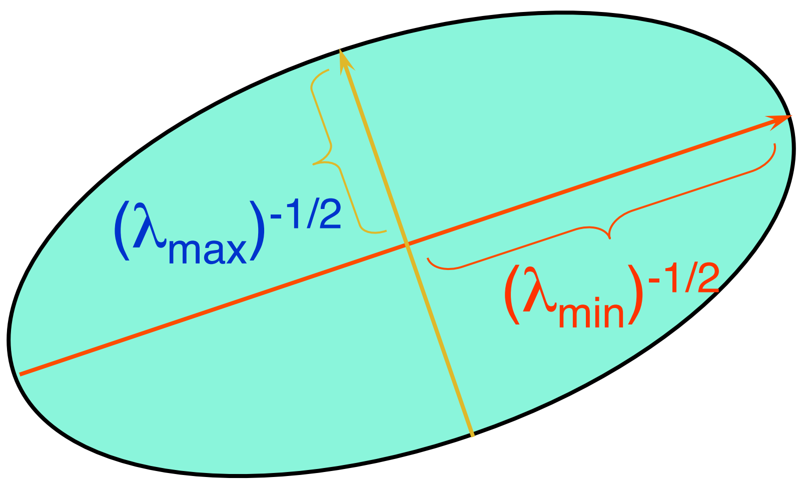


Source: S. Savarese.

Affine invariance

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

We can visualize *M* as an ellipse with axis lengths determined by the eigenvalues and orientation determined by *R*



The second moment ellipse can be viewed as the "characteristic shape" of a region

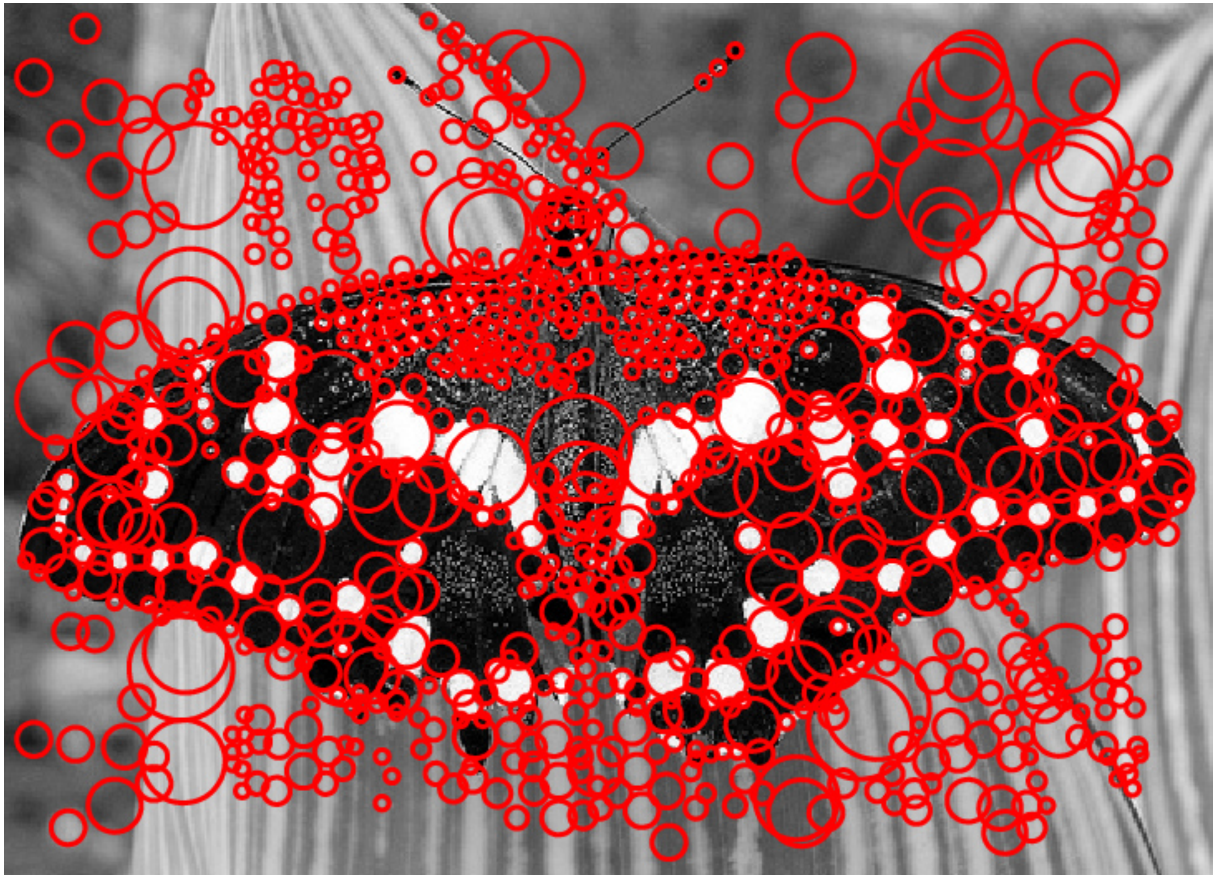
Source: S. Savarese.

Affine adaptation

1. Detect initial region with Harris Laplace
2. Estimate affine shape with M
3. Normalize the affine region to a circular one
4. Re-detect the new location and scale in the normalized image
5. Go to step 2 if the eigenvalues of the M for the new point are not equal [detector not yet adapted to the characteristic shape]

Source: S. Savarese.

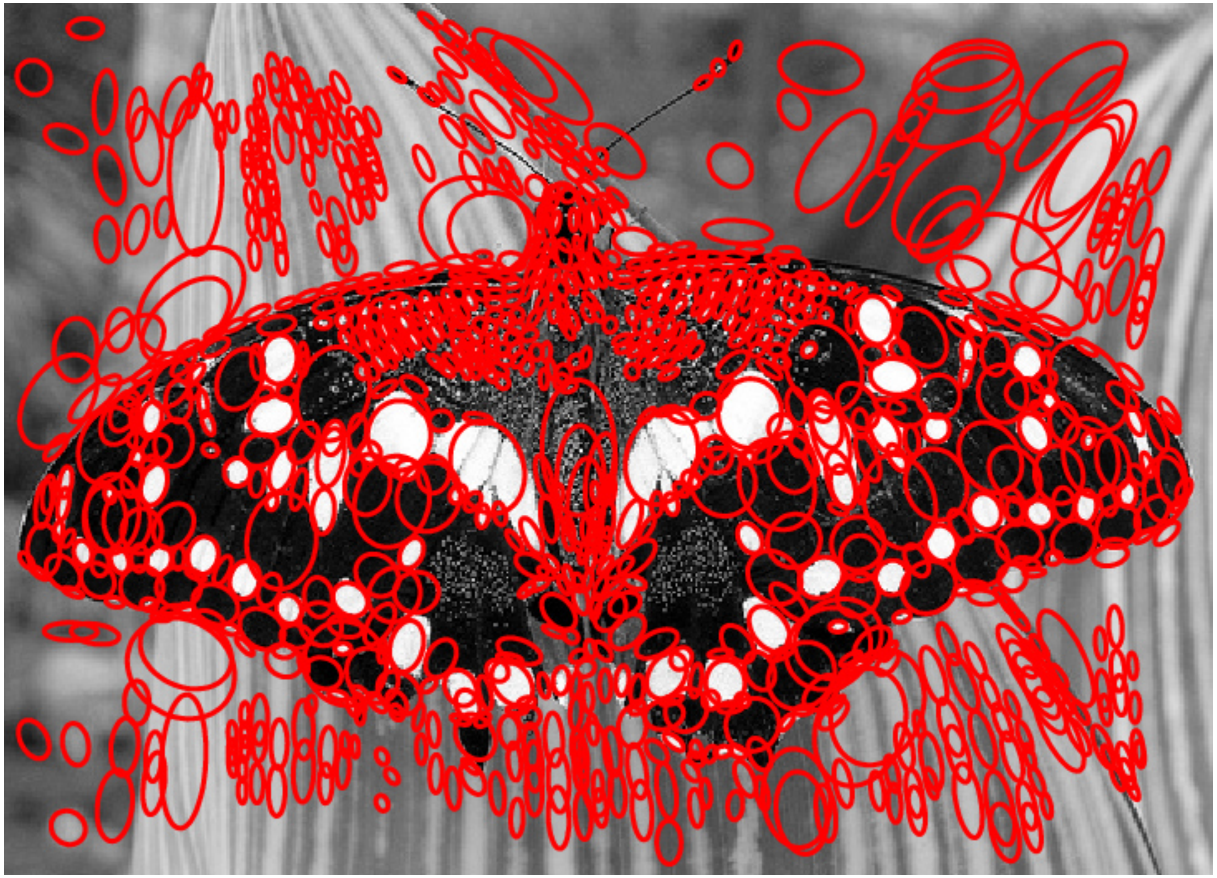
Without affine invariance



Scale-invariant regions (blobs)

Source: S. Savarese.

With affine invariance



Affine-adapted blobs

Invariance

Detector	Illumination	Rotation	Scale	View point
Harris corner	Yes	Yes	No	No
Lowe '99 (DoG)	Yes	Yes	Yes	No
Mikolajczyk & Schmid '01	Yes	Yes	Yes	No
Mikolajczyk & Schmid '02	Yes	Yes	Yes	Yes

Source: S. Savarese.

Detector	Illumination	Rotation	Scale	View point
Harris corner	Yes	Yes	No	No
Lowe '99 (DoG)	Yes	Yes	Yes	Yes
Mikolajczyk & Schmid '01, '02	Yes	Yes	Yes	Yes
Tuytelaars, '00	Yes	Yes	No (Yes '04)	Yes
Kadir & Brady, 01	Yes	Yes	Yes	no
Matas, '02	Yes	Yes	Yes	no

Source: S. Savarese.

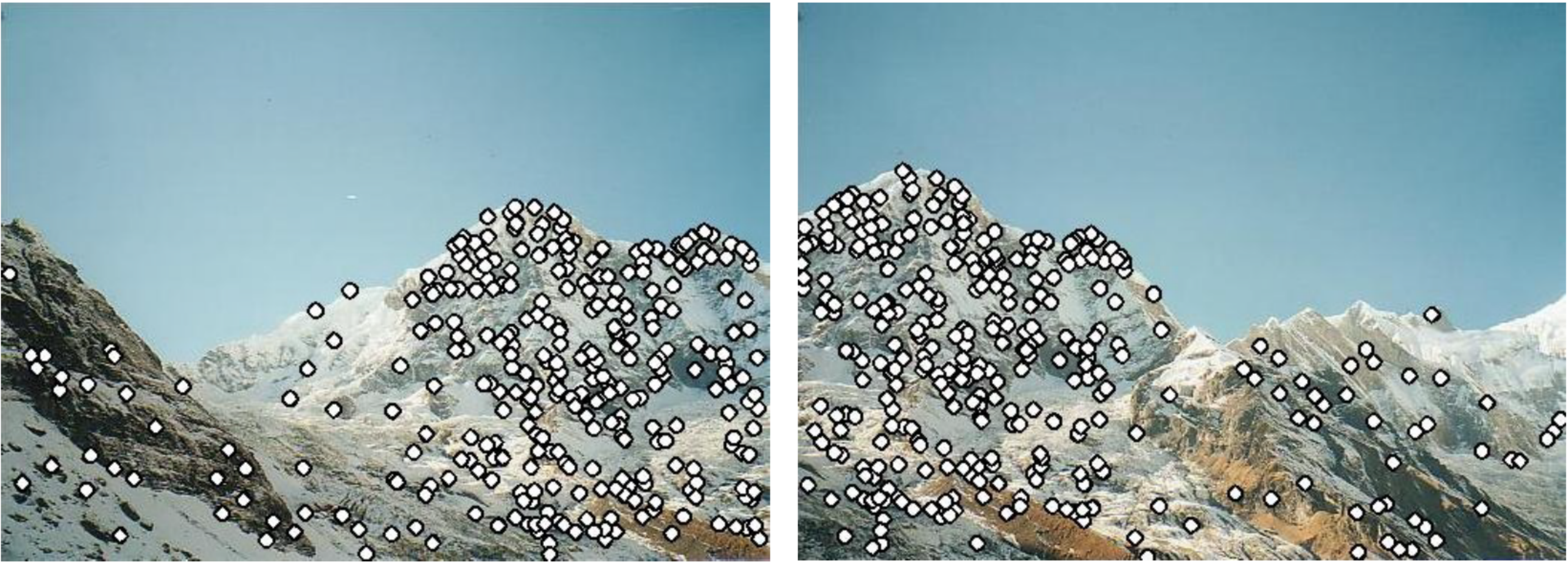
Feature Descriptors

Overview

Application: Image Stitching

Sources for this example: I. Kokkinos, D. Frolova, D. Simakov.

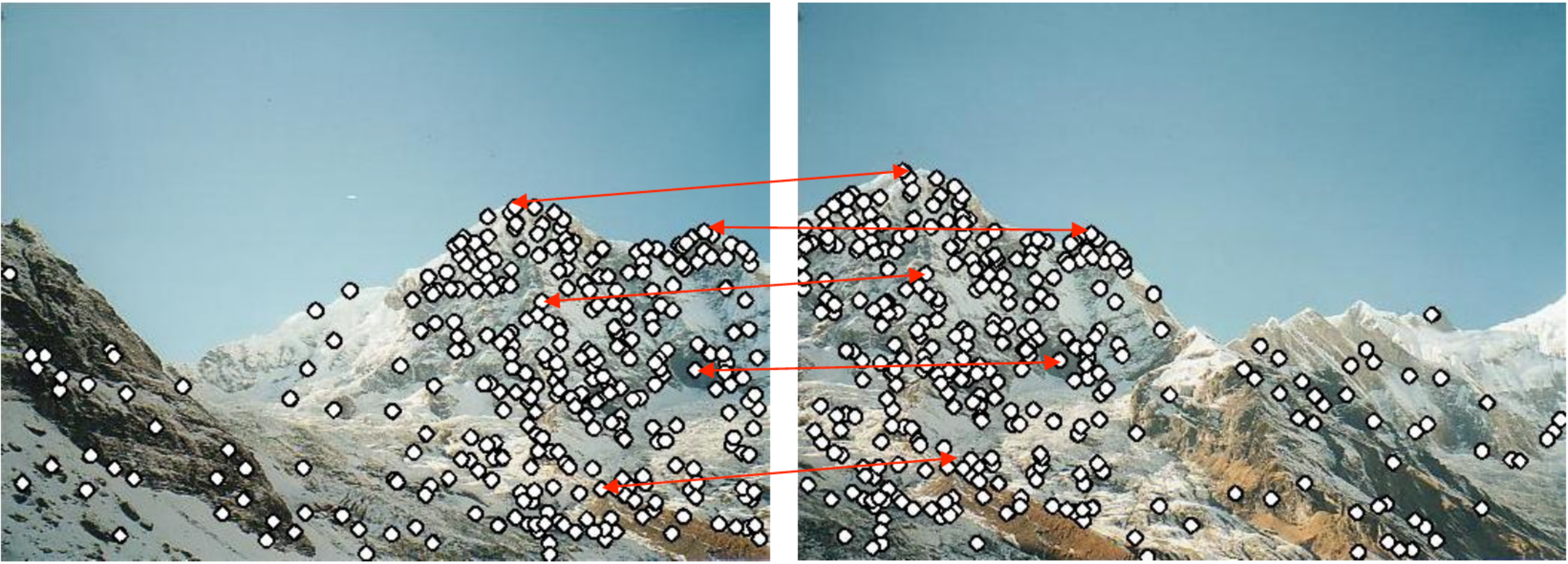
Application: Image Stitching



1. Detect feature points in both images.

Sources for this example: I. Kokkinos, D. Frolova, D. Simakov.

Application: Image Stitching



1. Detect feature points in both images.
2. Find corresponding pairs of feature points.

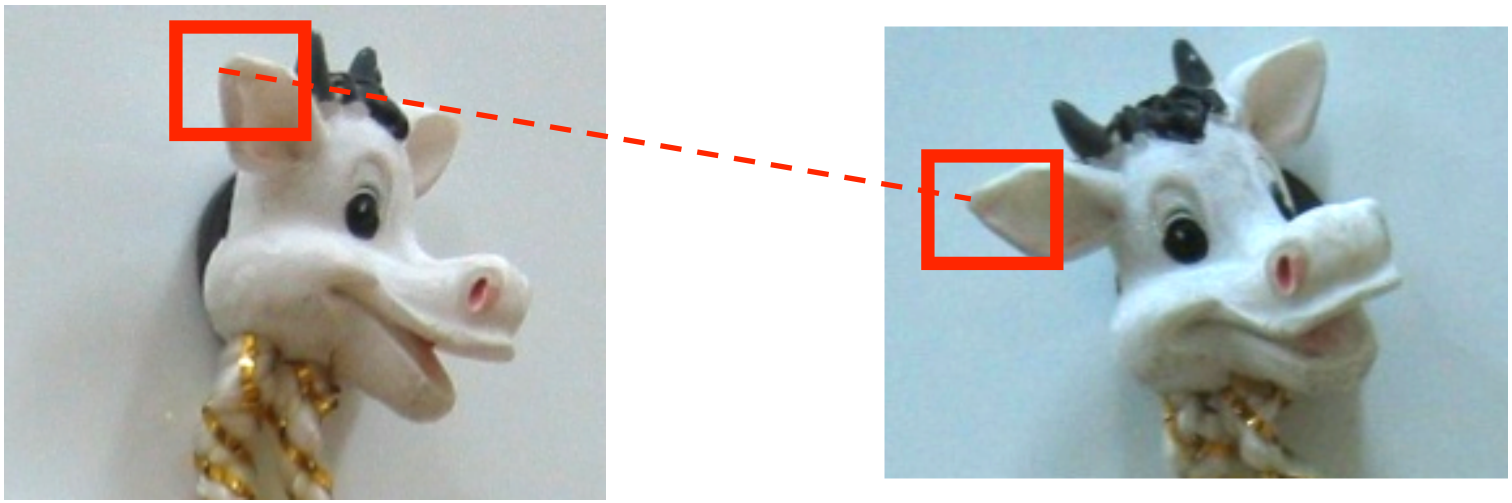
Sources for this example: I. Kokkinos, D. Frolova, D. Simakov.

Application: Image Stitching

1. Detect feature points in both images.
2. Find corresponding pairs of feature points.
3. Use the pairs the align the images.

Sources for this example: I. Kokkinos, D. Frolova, D. Simakov.

Application: Estimating Fundamental Matrix



1. Detect feature points in both images.
2. Find corresponding pairs of feature points.
3. Use the pairs to estimate epipolar geometry across images.

Source: S. Savarese.

Application: Detect Object Instances

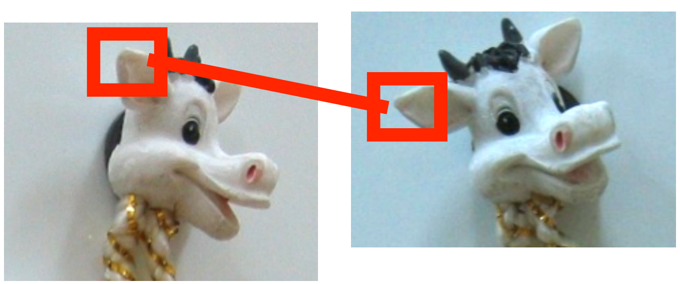
1. Detect feature points in both images.
2. Find corresponding pairs of feature points.
3. Use the pairs to match object instances.

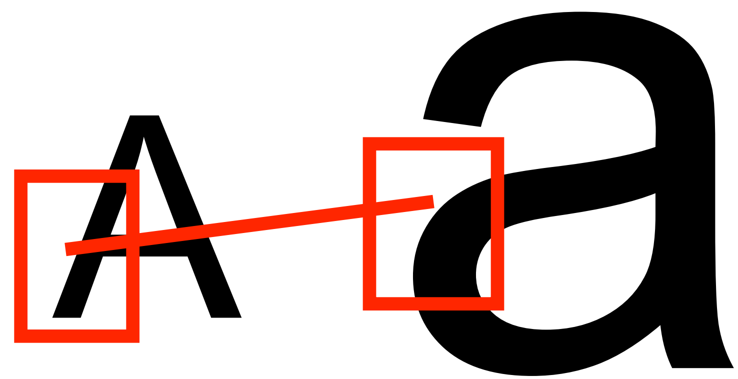
Source: S. Savarese.

Challenges

Depending on the application a descriptor must incorporate information that is:

- Invariant w.r.t:
  - Illumination
  - Pose
  - Scale
  - Intraclass variability





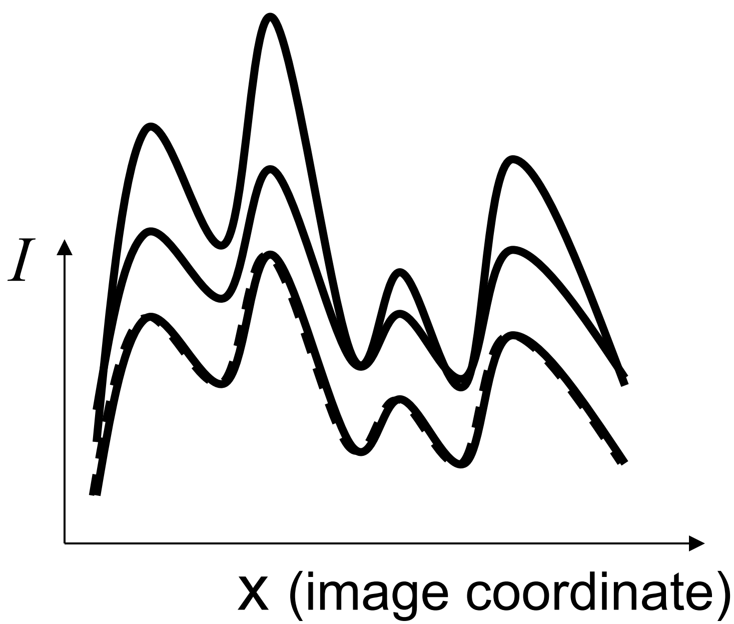
- **Highly distinctive** (allows a single feature to find its correct match with good probability in a large database of features)

Source: S. Savarese.

Illumination normalization

- A*ffine intensity* change:

$$I \rightarrow I + b$$
$$\rightarrow a\,I + b$$



•Make each patch zero mean:

$$\mu = \frac{1}{N}\sum_{x,y} I(x,y)$$

$$Z(x,y) = I(x,y) - \mu$$

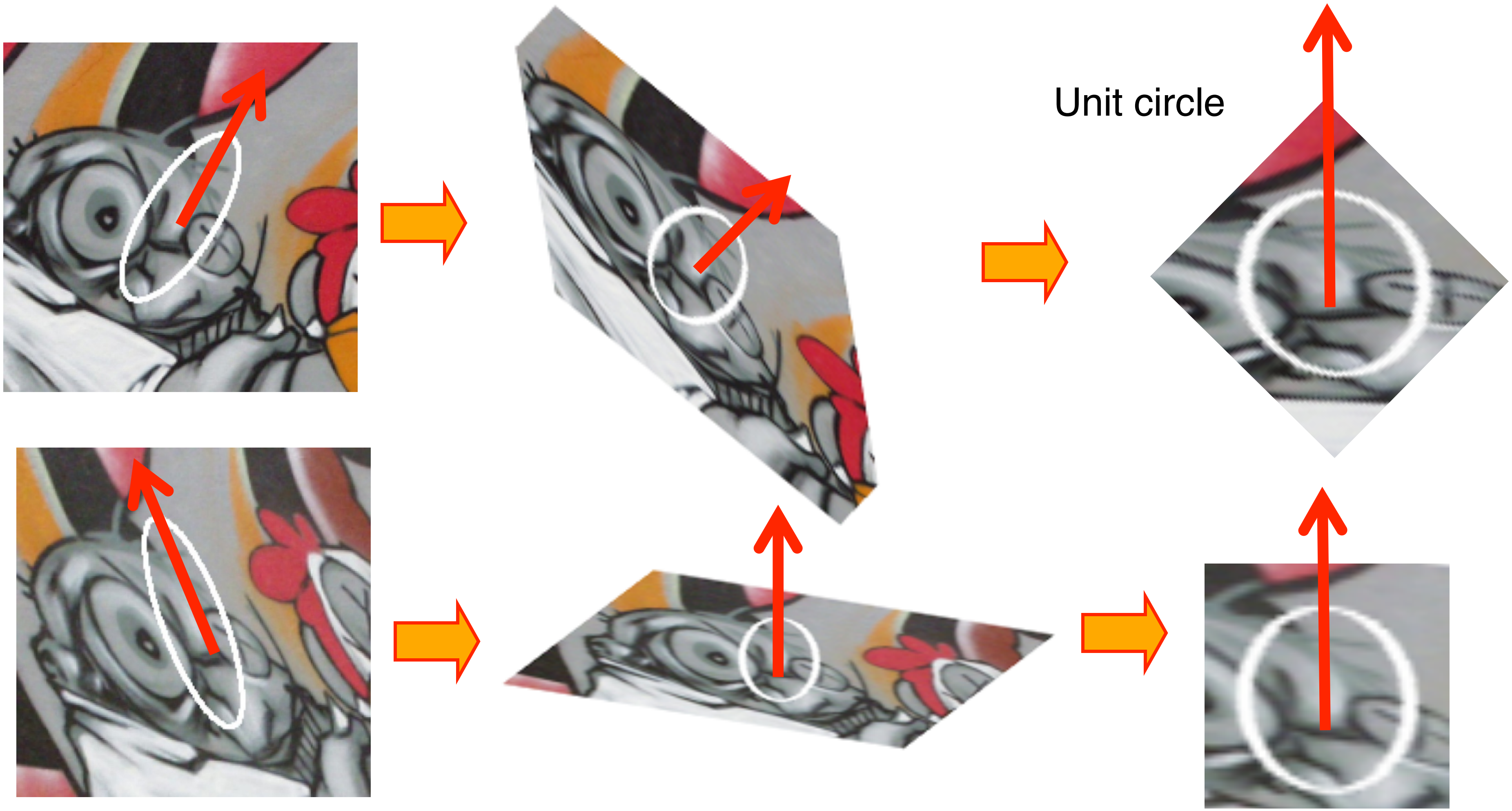
•Then make unit variance:

$$\sigma^2 = \frac{1}{N}\sum_{x,y} Z(x,y)^2$$

$$ZN(x,y) = \frac{Z(x,y)}{\sigma}$$

Source: S. Savarese.

Pose normalization

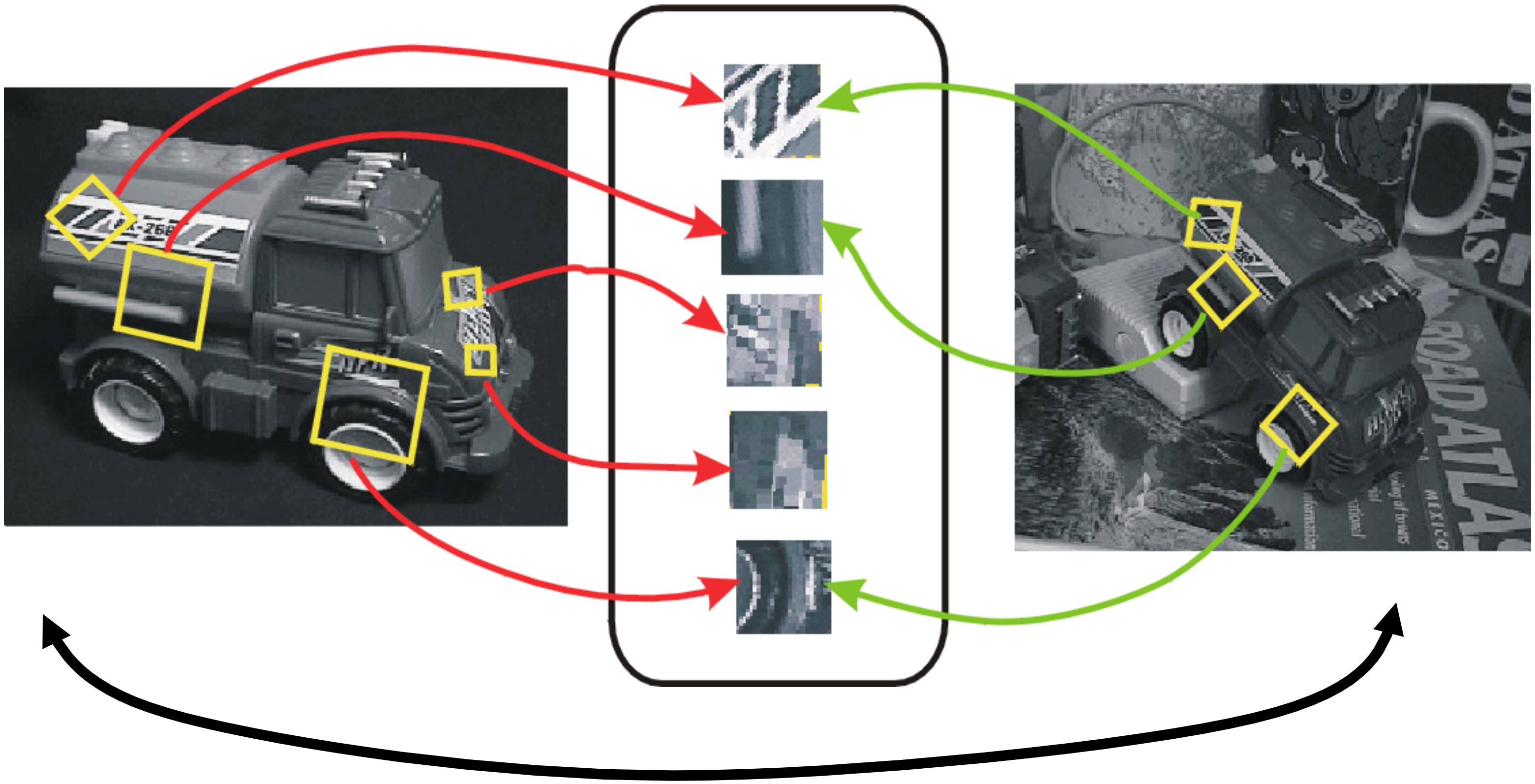


NOTE: location, scale, rotation & affine pose are given by the detector or calculated within the detected regions

Source: S. Savarese.

Pose normalization

- Keypoints are transformed in order to be invariant to translation, rotation, scale, and other geometrical parameters [Lowe 2000]



Courtesy of D. Lowe

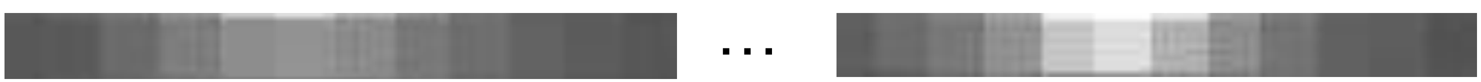
Change of scale, pose, illumination…

Source: S. Savarese.

The simplest descriptor

1 x NM vector of pixel intensities

w= [



…]

$$w_n = \frac{(w - \overline{w})}{\left\| (w - \overline{w}) \right\|}$$

Makes the descriptor invariant with respect to affine transformation of the illumination condition

Source: S. Savarese.

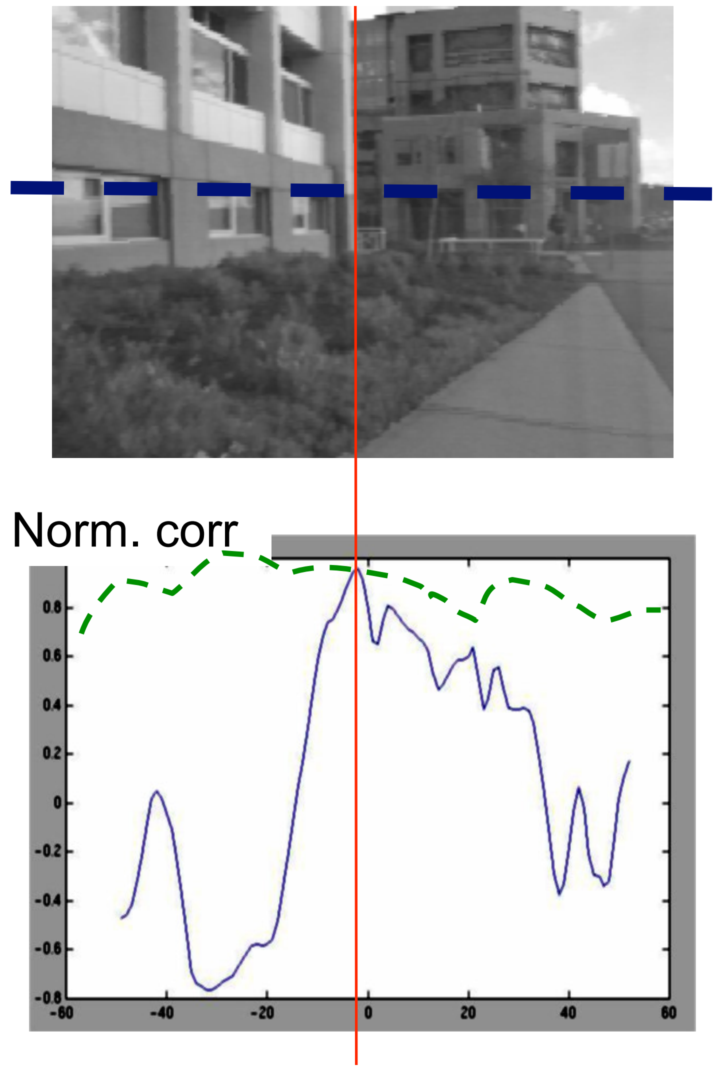
Why not?

- Sensitive to small variation of:
    - location
    - Pose
    - Scale
    - intra-class variability

- Poorly distinctive

Source: S. Savarese.

Sensitive to pose variations



Normalized Correlation:

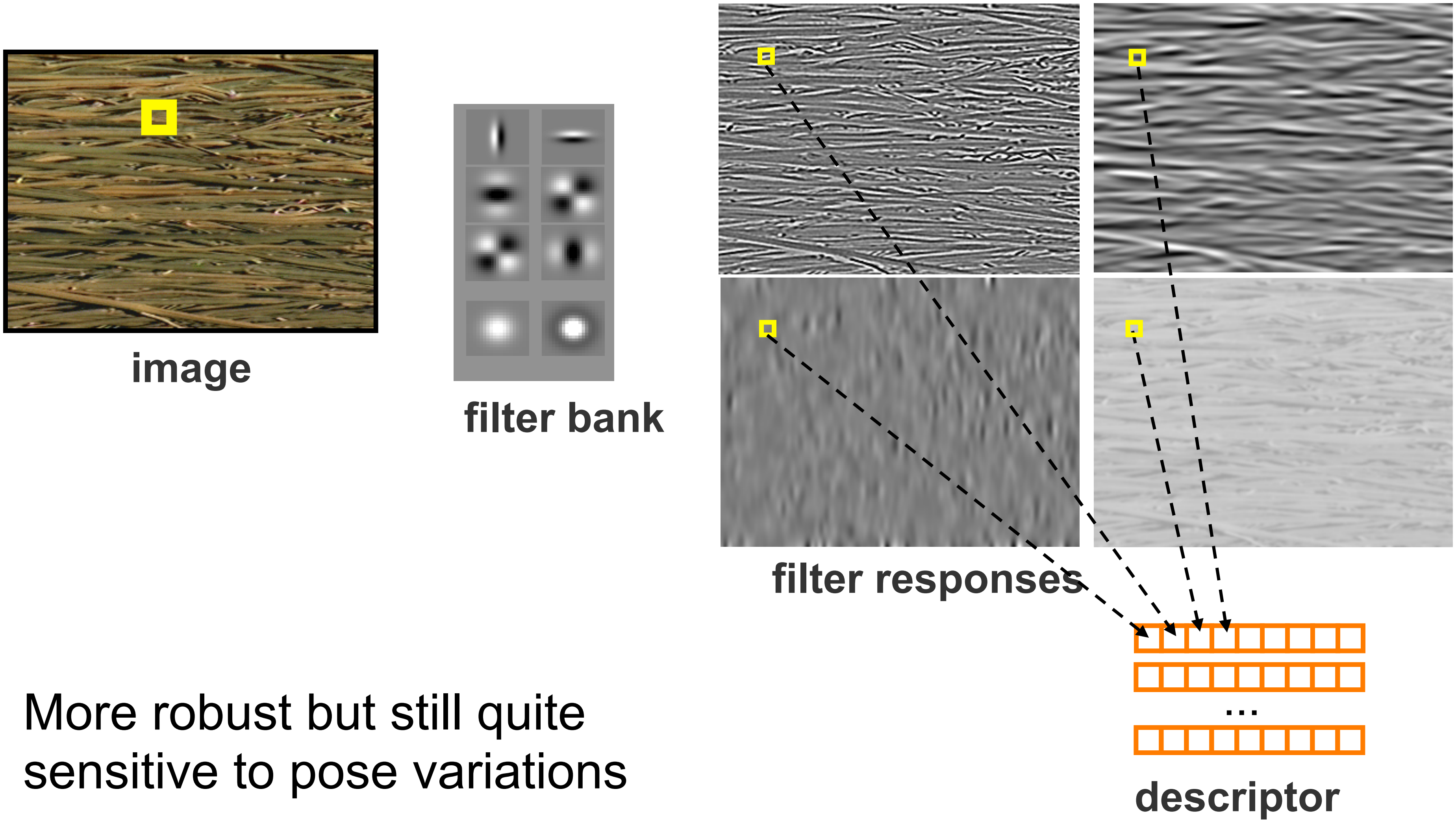
$$w_n \cdot w'_n = \frac{(w-\overline{w})(w'-\overline{w}')}{\left\|(w-\overline{w})(w'-\overline{w}')\right\|}$$

Source: S. Savarese.

Detector	Illumination	Pose	Intra-class variab.
PATCH	Good	Poor	Poor

Source: S. Savarese.

Bank of filters



More robust but still quite sensitive to pose variations

Source: S. Savarese.

Detector	Illumination	Pose	Intra-class variab.
PATCH	Good	Poor	Poor
FILTERS	Good	Medium	Medium

Source: S. Savarese.

SIFT descriptor

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04

- Alternative representation for image patches
- Location and characteristic scale s given by DoG detector

Source: S. Savarese.

SIFT descriptor

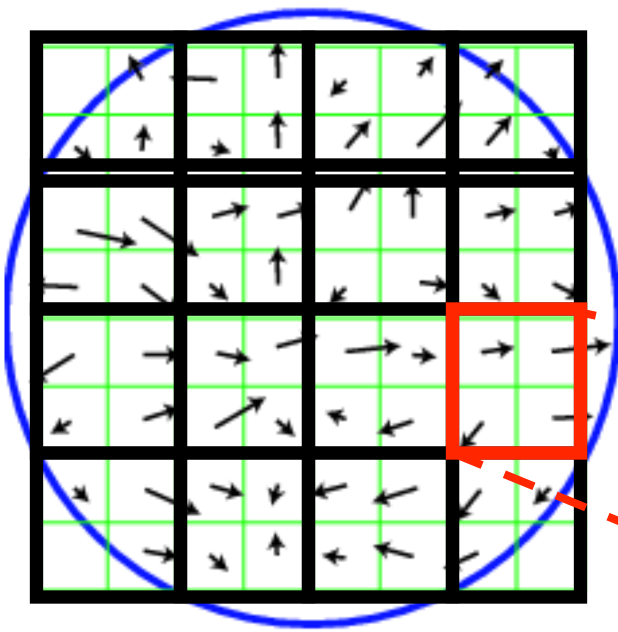
David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04

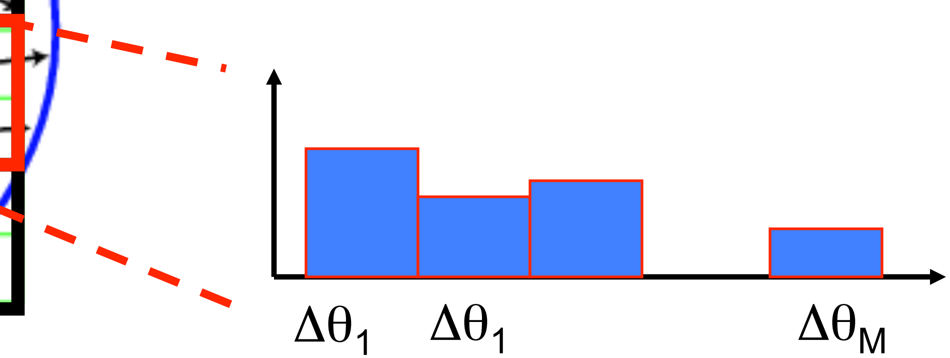
- Alternative representation for image patches
- Location and characteristic scale s given by DoG detector

•Compute gradient at each pixel

• N x N spatial bins

• Compute an histogram of M orientations for each bin



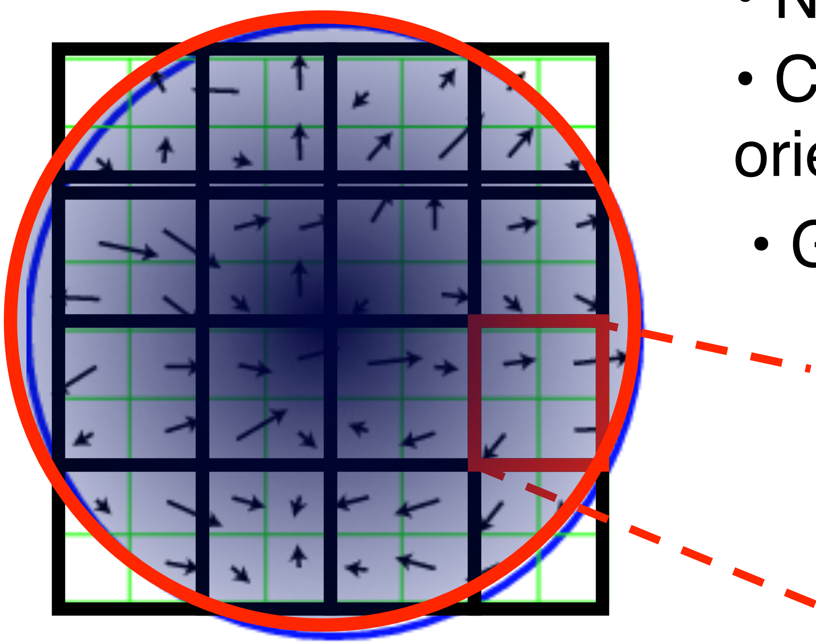


Source: S. Savarese.

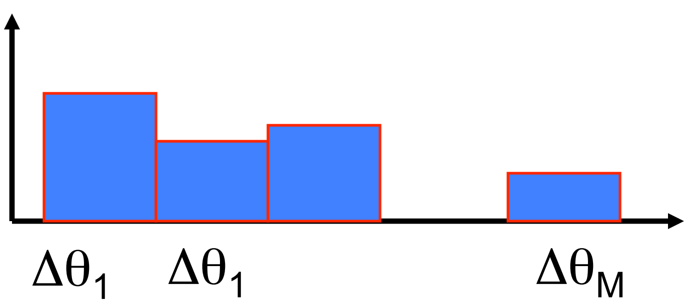
SIFT descriptor

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04

- Alternative representation for image patches
- Location and characteristic scale s given by DoG detector



- Compute gradient at each pixel
- N x N spatial bins
- Compute an histogram of M orientations for each bin
- Gaussian center-weighting

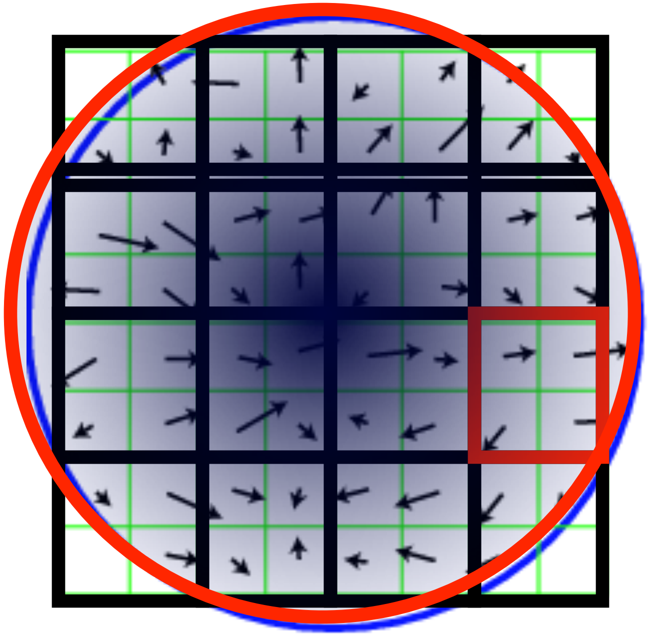


Source: S. Savarese.

SIFT descriptor

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04

- Alternative representation for image patches
- Location and characteristic scale s given by DoG detector



- Compute gradient at each pixel
- N x N spatial bins
- Compute an histogram of M orientations for each bin
  - Gaussian center-weighting
  - Normalized unit norm

Typically M = 8; N= 4
1 x 128 descriptor

Source: S. Savarese.

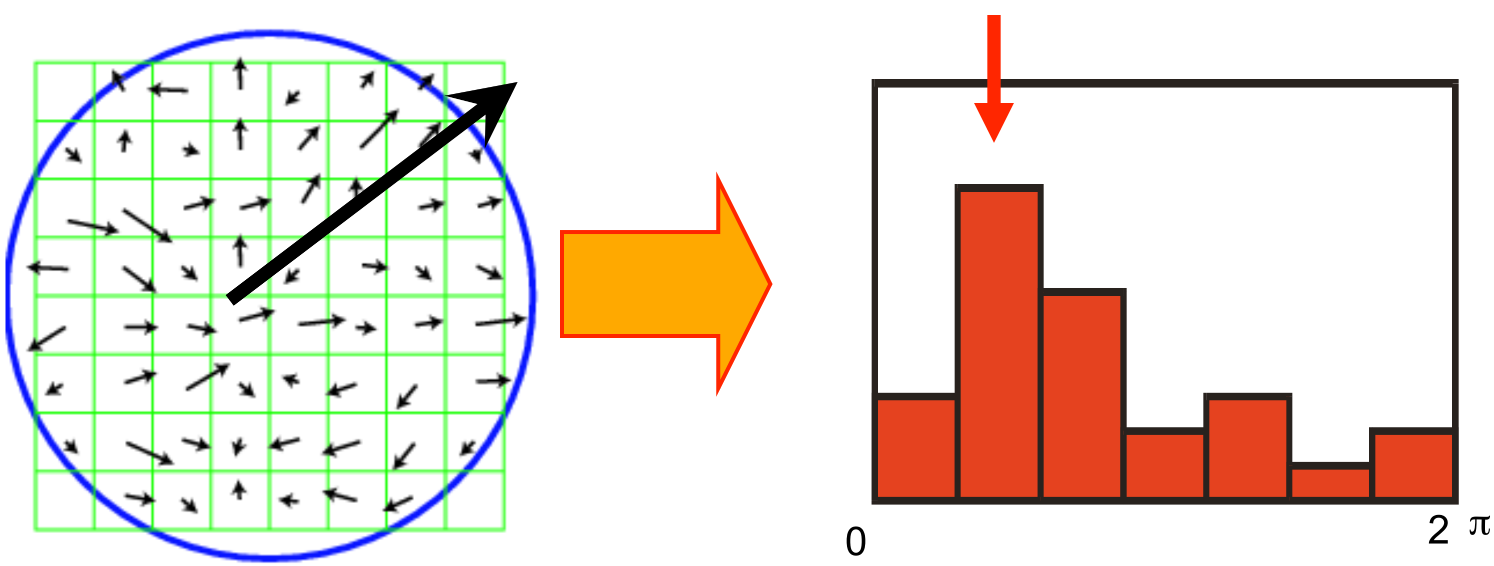
SIFT Descriptor

- Robust w.r.t. small variation in:
  - Illumination (thanks to gradient & normalization)
  - Pose (small affine variation thanks to orientation histogram)
  - Scale (scale is fixed by DOG)
  - Intra-class variability (small variations thanks to histograms)

Source: S. Savarese.

Rotational Invariance

- Find dominant orientation by building smoothed orientation histogram
- Rotate all orientations by the dominant orientation



This makes the SIFT descriptor rotational invariant

Source: S. Savarese.

SIFT Rotational Invariance Example

(b)

Source: S. Savarese.

Rotation invariance (Alternate)

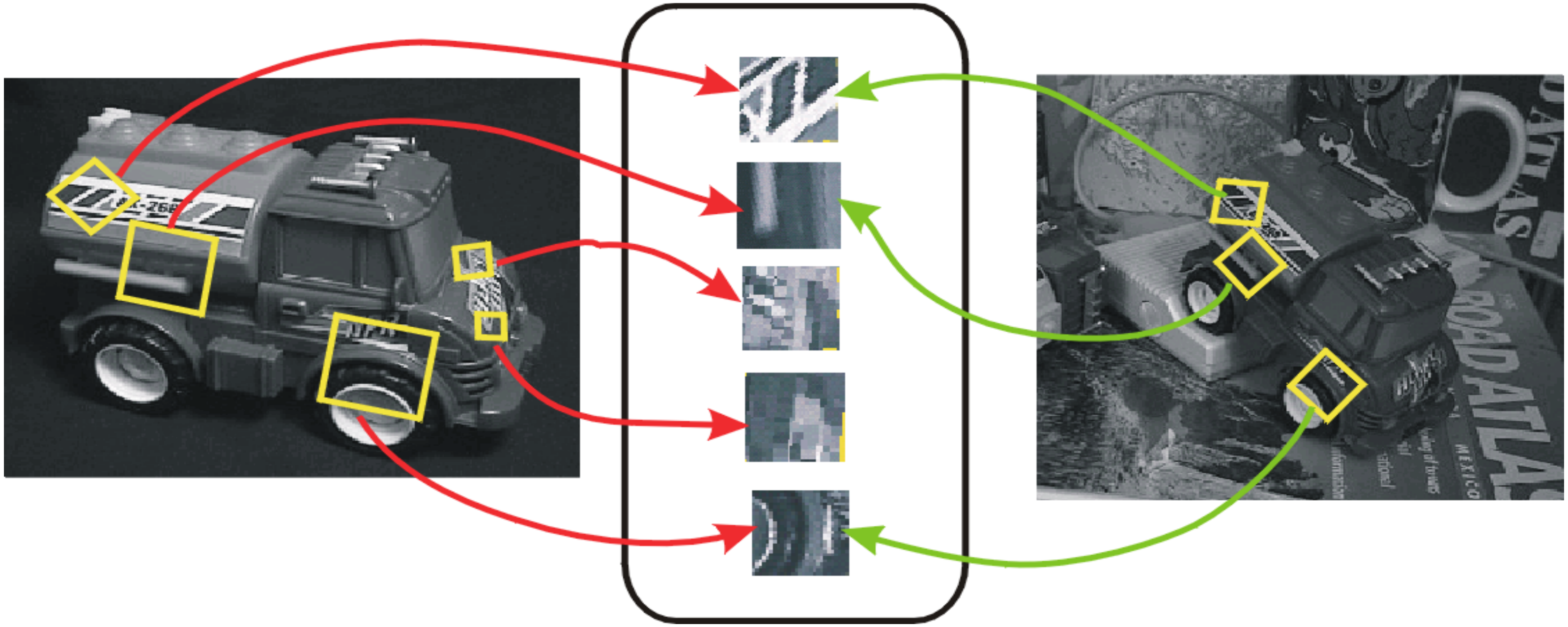
Find dominant orientation of the image patch

- This is given by $\mathbf{x}_+$, the eigenvector of **H** corresponding to λ_+
  - λ_+ is the *larger* eigenvalue
- Rotate the patch according to this angle

Figure by Matthew Brown

Source: S. Seitz.

SIFT Rotational Invariance Example



Source: S. Savarese.

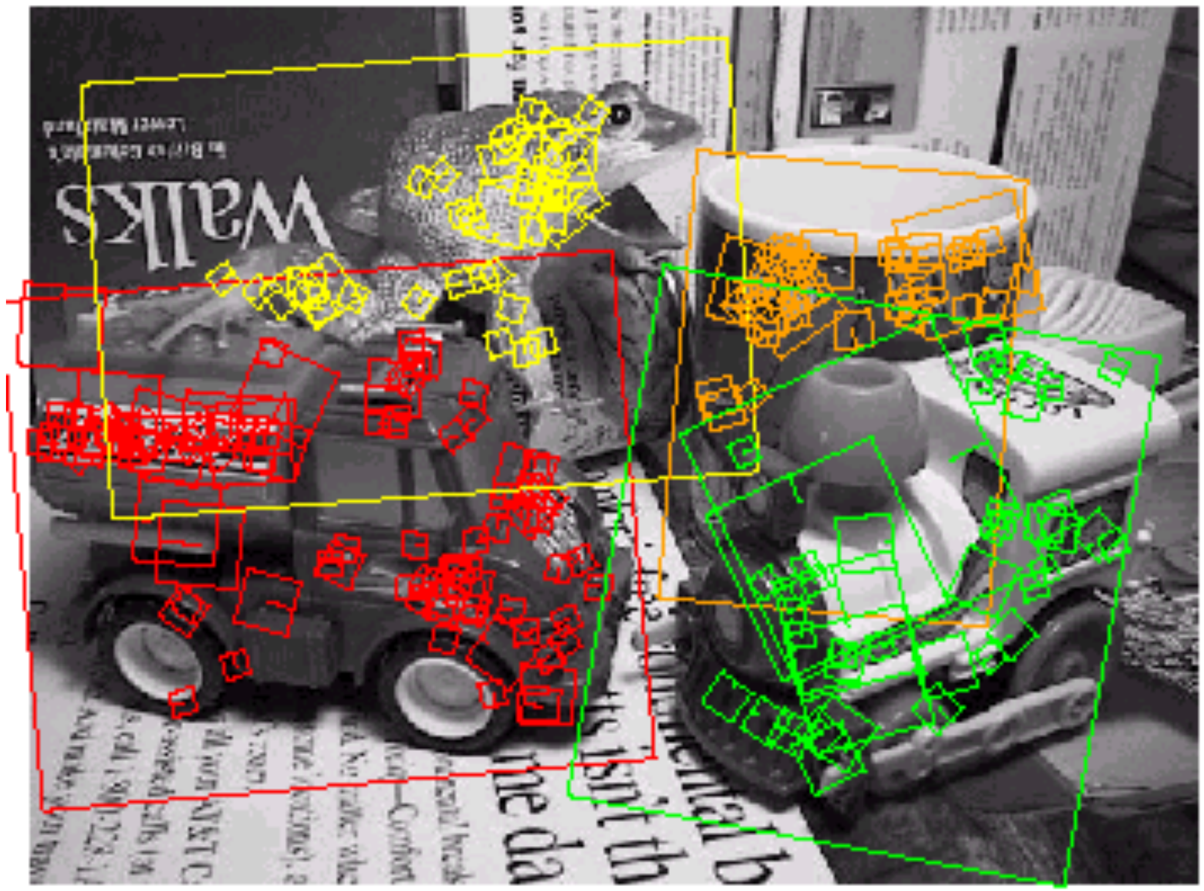
Matching Using SIFT

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04

Source: S. Savarese.

Matching Using SIFT

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04



Source: S. Savarese.

Detector	Illumination	Pose	Intra-class variab.
PATCH	Good	Poor	Poor
FILTERS	Good	Medium	Medium
SIFT	Good	Good	Medium

Source: S. Savarese.

Next Lecture: Segmentation and Clustering

- Readings: FP 6.2, 9; SZ 5.2-5.4