Scale-invariant Feature Detection

Feature description and matching

Announcements

- HW 1 and PA 2 out tonight or tomorrow
- Schedule will be updated shortly
- Artifact voting out today
  - Please vote

Feature extraction: Corners

Slides from Rick Szeliski, Svetlana Lazebnik, and Kristin Grauman

The second moment matrix

The surface *E*(*u*,*v*) is locally approximated by a quadratic form.

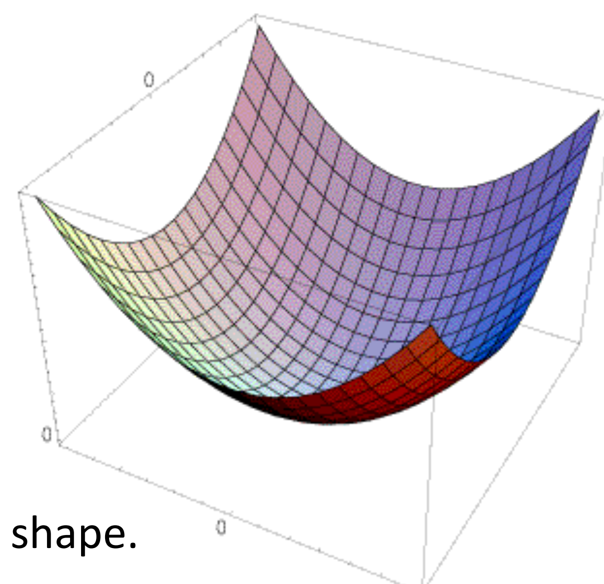
$$E(u,v) \approx Au^2 + 2Buv + Cv^2$$

$$\approx \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

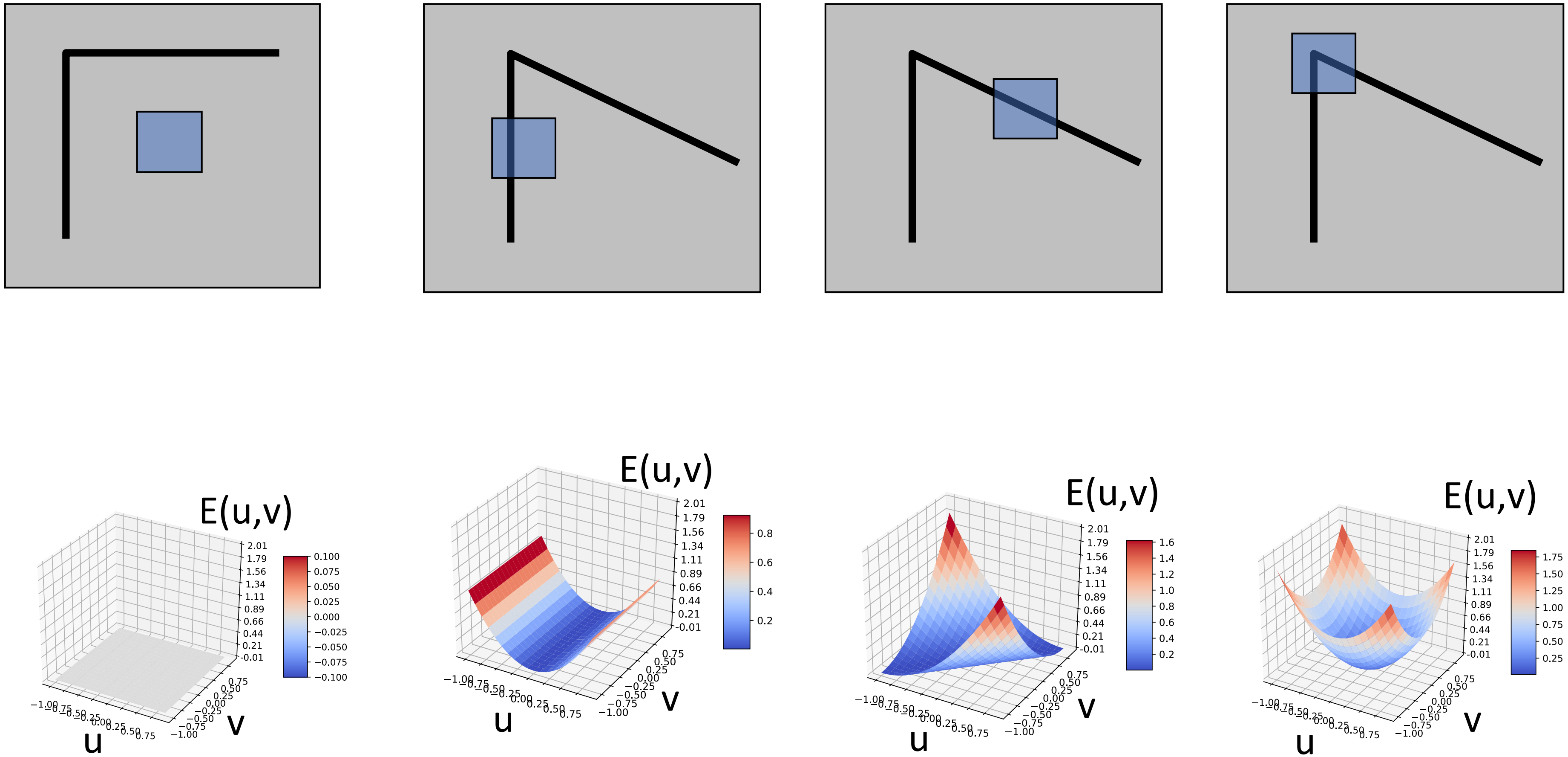
$$A = \sum_{(x,y)\in W} I_x^2$$

$$B = \sum_{(x,y)\in W} I_x I_y$$

$$C = \sum_{(x,y)\in W} I_y^2$$



Let’s try to understand its shape.



E(u,v)
u
v
E(u,v)
u
v
E(u,v)
u
v
E(u,v)
u
v

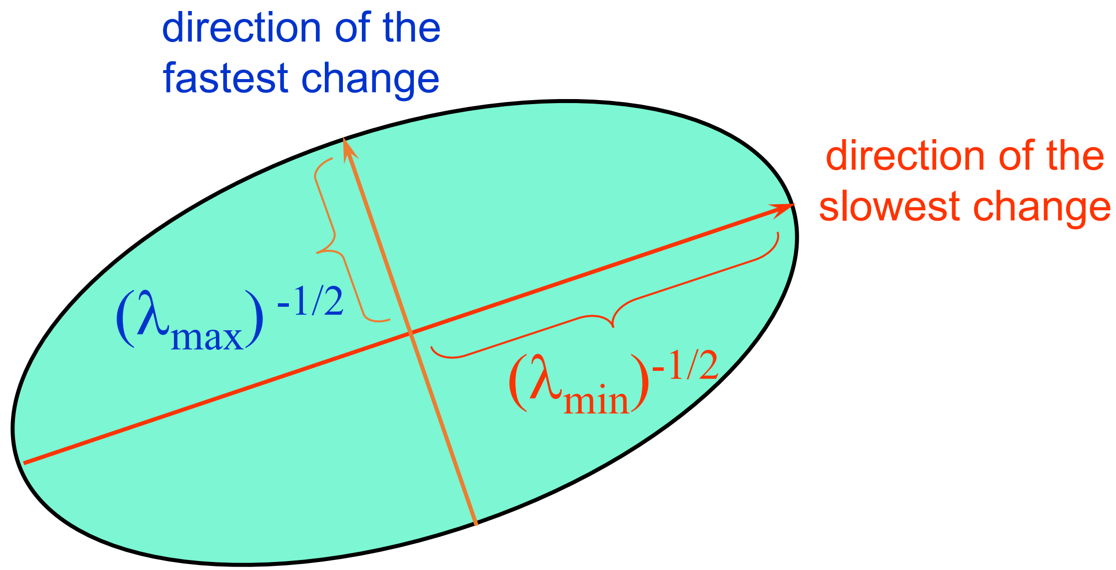
Interpreting the second moment matrix

Consider a horizontal "slice" of *E*(*u*, *v*): $[u \; v] \; M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

This is the equation of an ellipse.

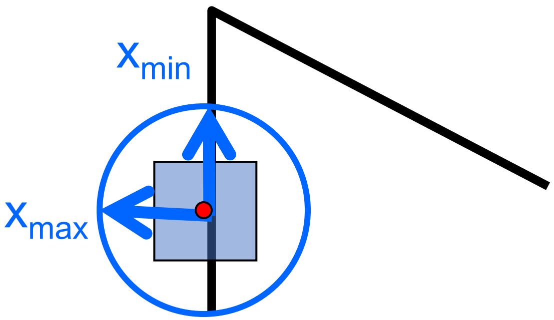
Diagonalization of M: $M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$

The axis lengths of the ellipse are determined by the eigenvalues and the orientation is determined by *R*



Corner detection: the math

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$



M $x_{\max} = \lambda_{\max} x_{\max}$

M $x_{\min} = \lambda_{\min} x_{\min}$

Eigenvalues and eigenvectors of M

- Define shift directions with the smallest and largest change in error
- x_{max} = direction of largest increase in *E*
- λ_{max} = amount of increase in direction x_{max}
- x_{min} = direction of smallest increase in *E*
- λ_{min} = amount of increase in direction x_{min}

The Harris operator

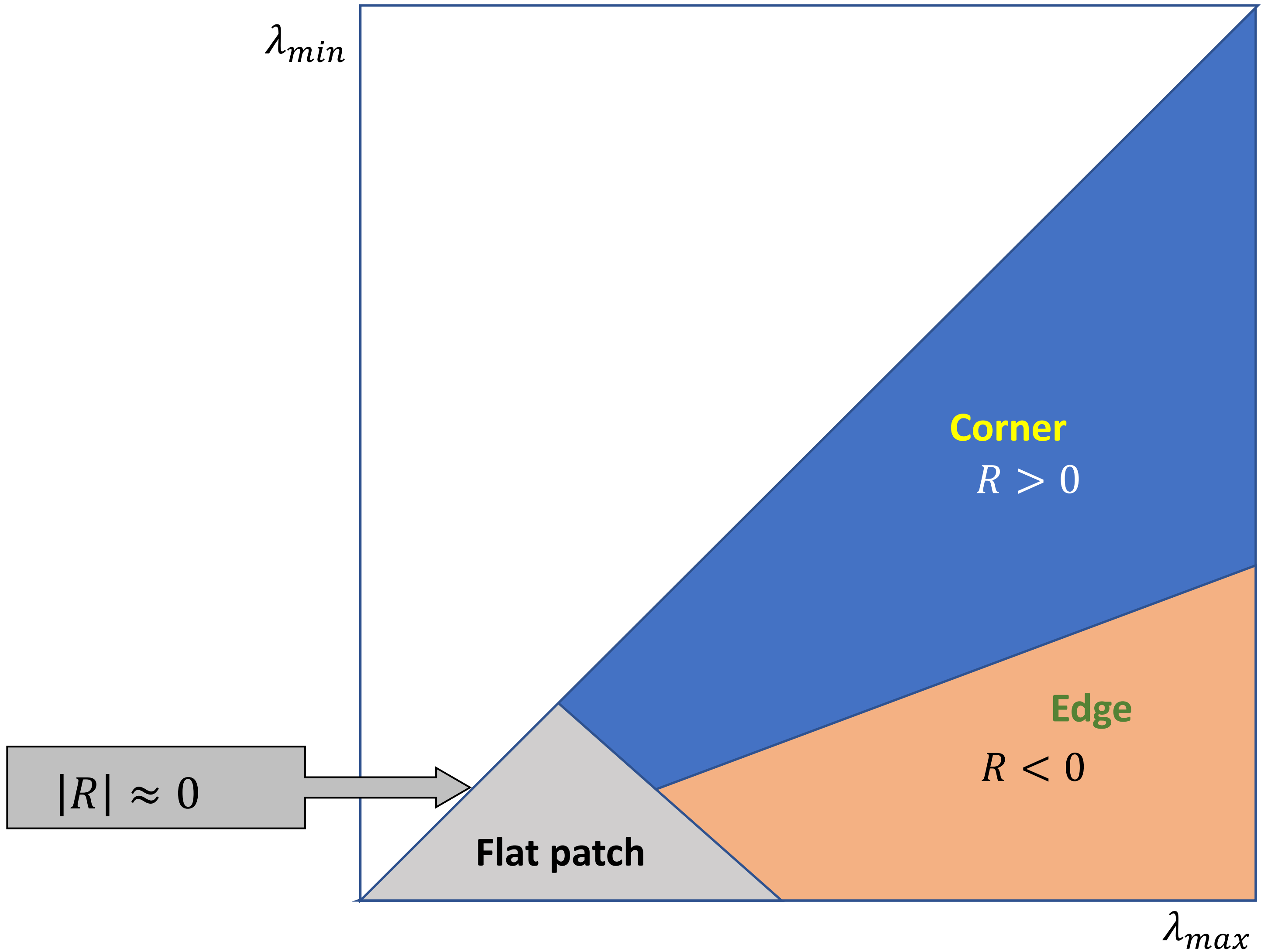
λ_{min} is a variant of the "Harris operator" for feature detection

$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$
$$= \frac{determinant(H)}{trace(H)}$$

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

Corner response function

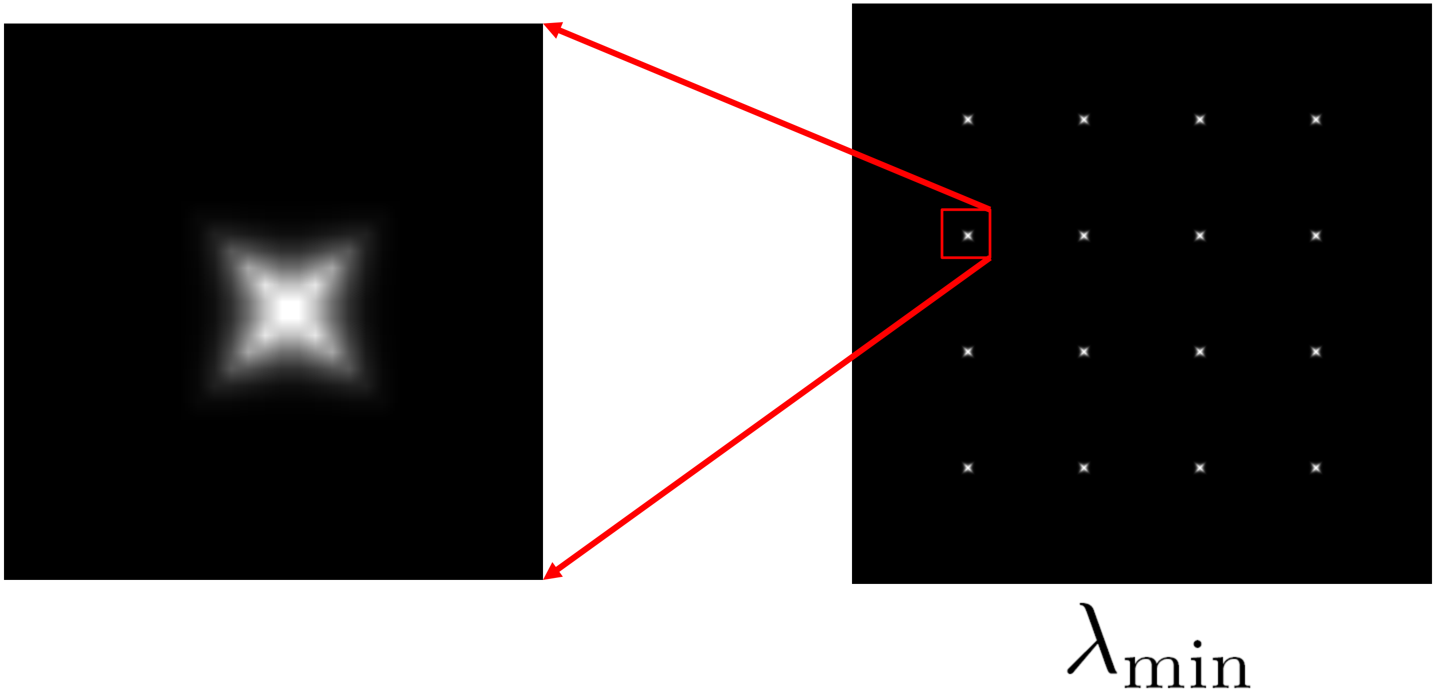
$$R = \det(M) - \alpha \, \text{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$



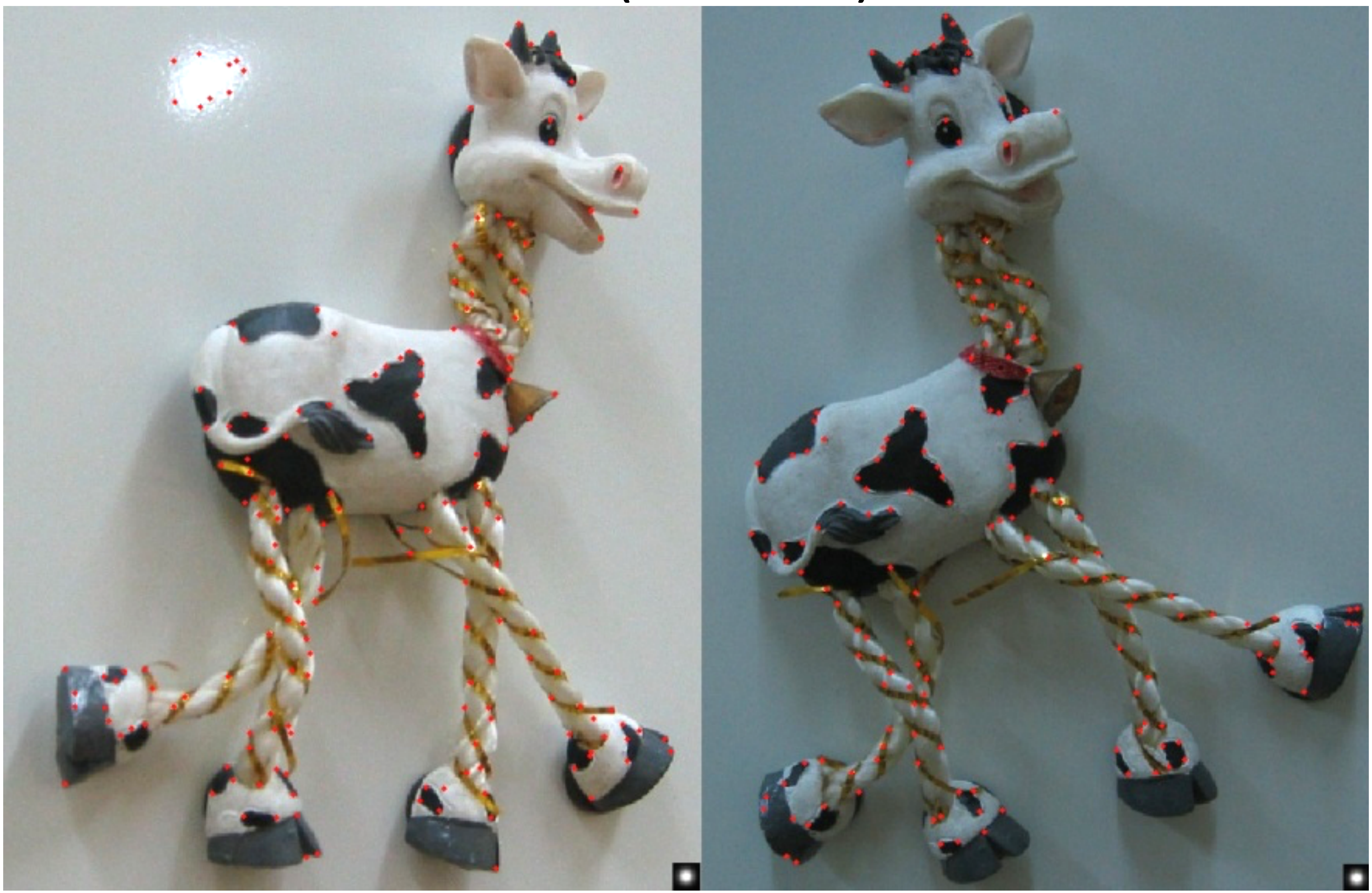
Corner detection summary

Here's what you do

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



Harris features (in red)



Harris Corners – Why so complicated?

- Can’t we just check for regions with lots of gradients in the x and y directions?
  - No! A diagonal line would satisfy that criteria

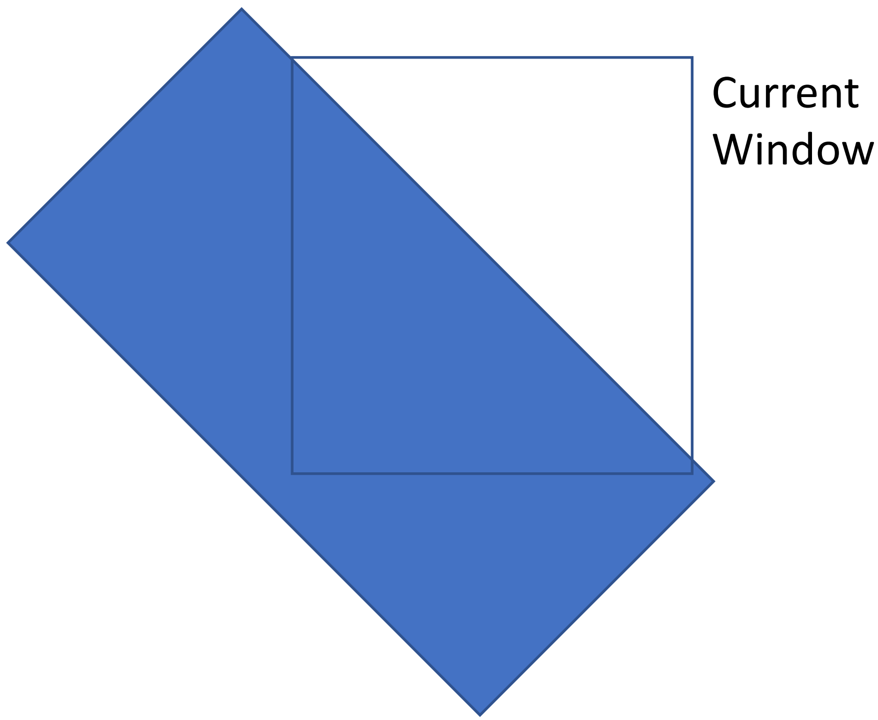
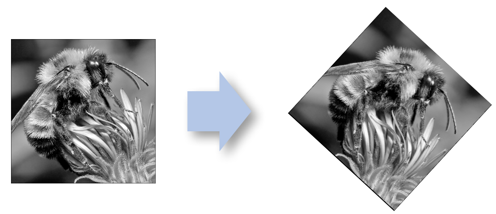


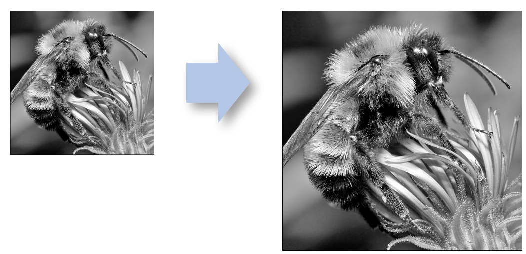
Image transformations

- Geometric

Rotation

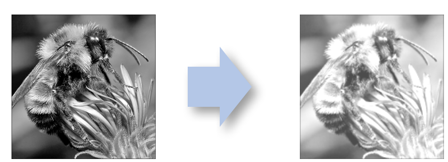


Scale



- Photometric

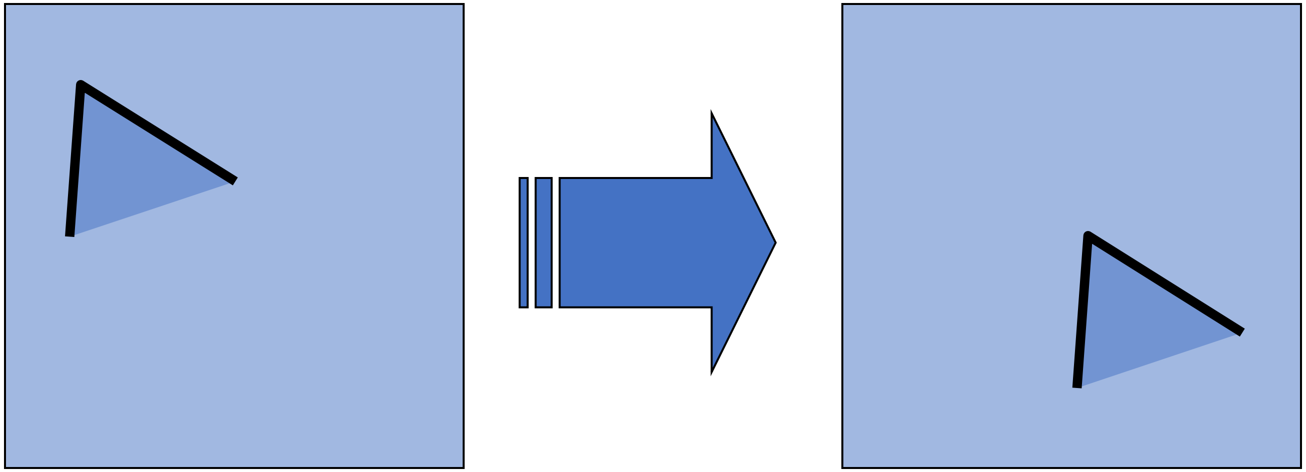
Intensity change



Invariance and covariance

- We want corner locations to be *invariant* to photometric transformations and *covariant* to geometric transformations
  - **Invariance:** image is transformed and corner locations do not change
  - **Equivariance:** if we have two transformed versions of the same image, features should be detected in corresponding locations

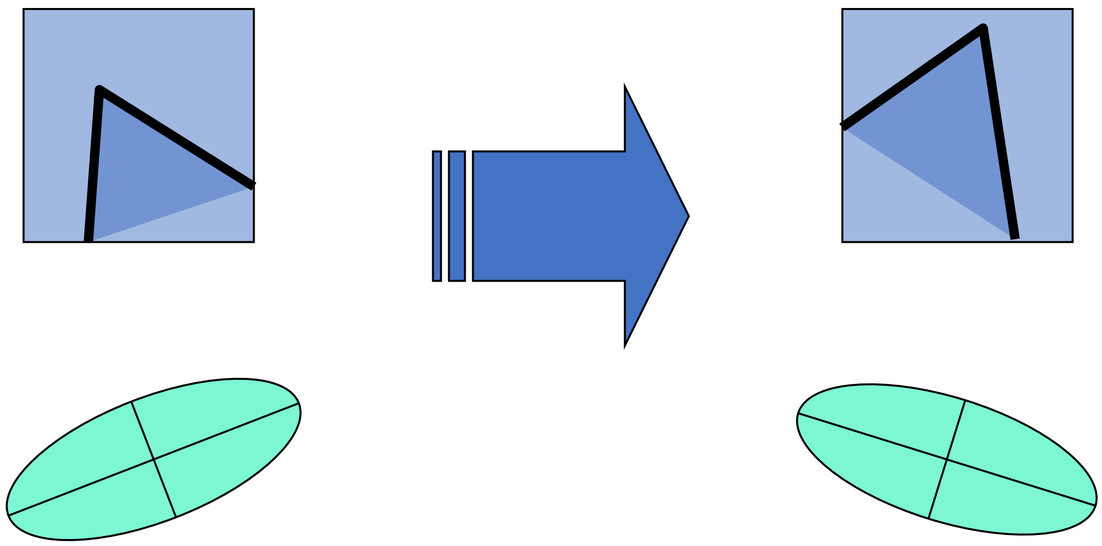
Image translation



- Derivatives and window function are shift-invariant

Corner location is equivariant w.r.t. translation

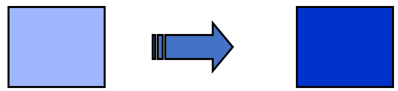
Image rotation



Second moment ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner location is equivariant w.r.t. rotation

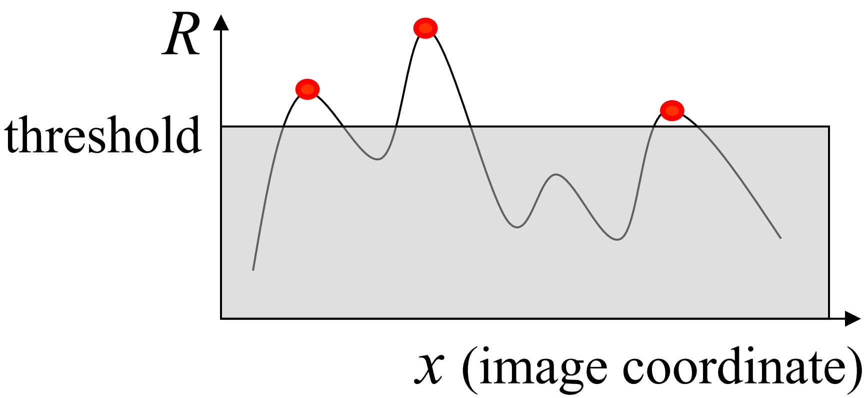
Affine intensity change

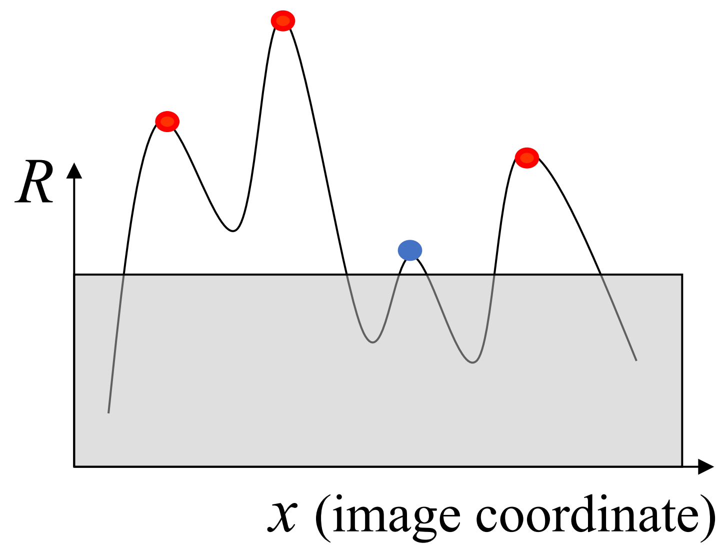


$I \rightarrow a\,I + b$

- Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$

- Intensity scaling: $I \rightarrow a\,I$

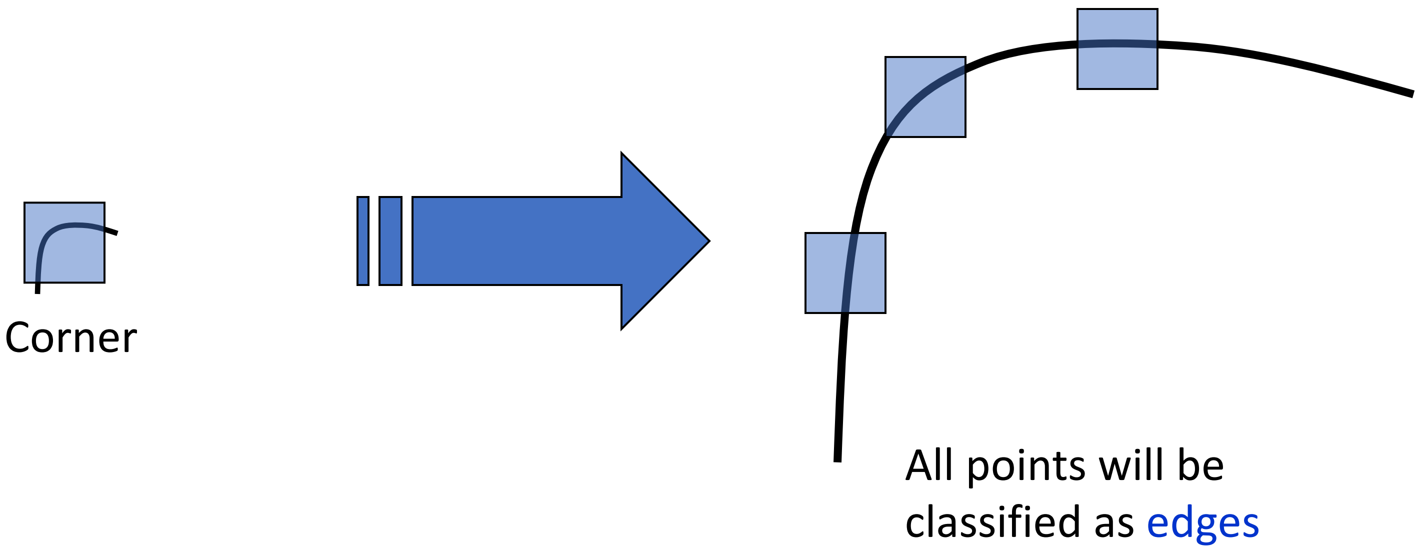




Partially invariant to affine intensity change

Harris Detector: Invariance Properties

- Scaling

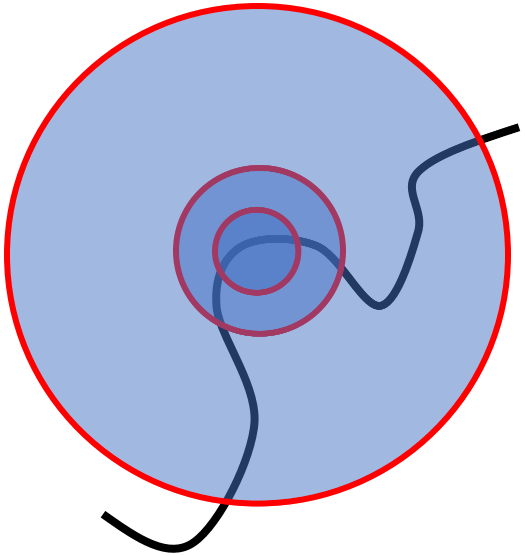


Not invariant to scaling

So far: can localize in x-y, but not scale

Scale invariant detection

Suppose you're looking for corners

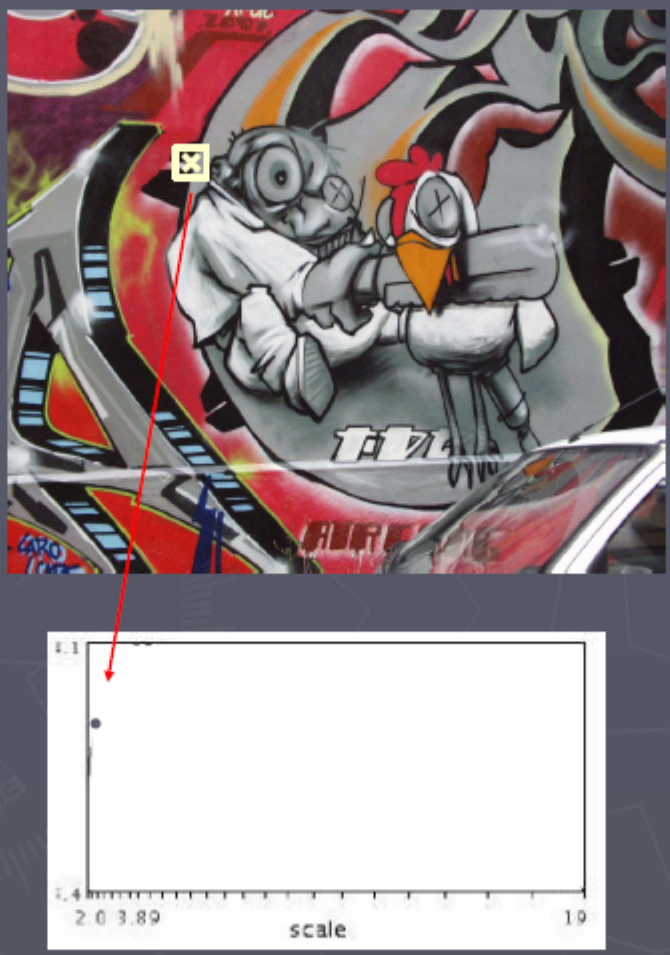


Key idea: find scale that gives local maximum of *f*

- in both position and scale
- One definition of *f*: the Harris operator

Automatic scale selection

Lindeberg et al., 1996



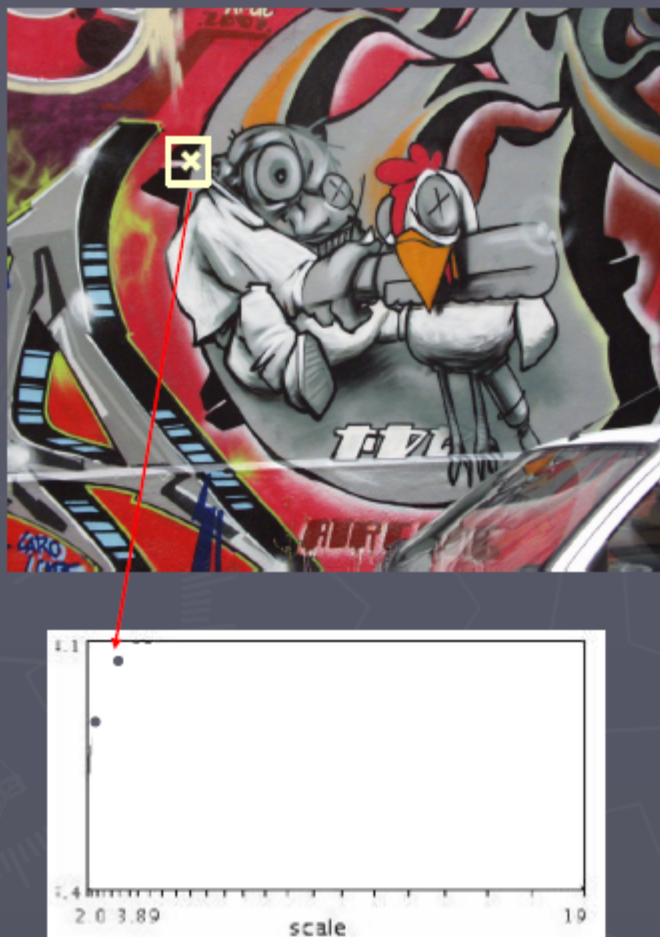
$f(I_{i_1 \dots i_m}(x,\sigma))$

Slide from Tinne Tuytelaars

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

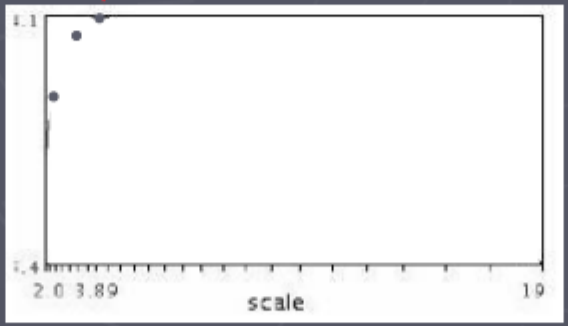


$$f(I_{i_1 \dots i_m}(x,\sigma))$$

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

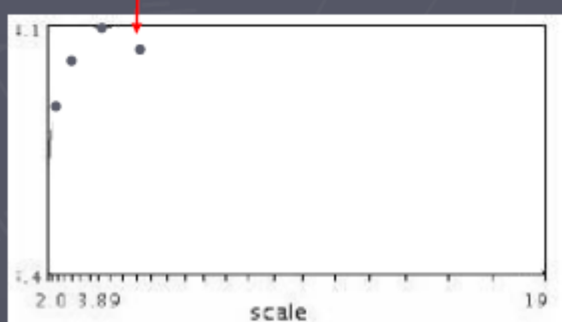


$f(I_{i_1 \ldots i_m}(x, \sigma))$

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

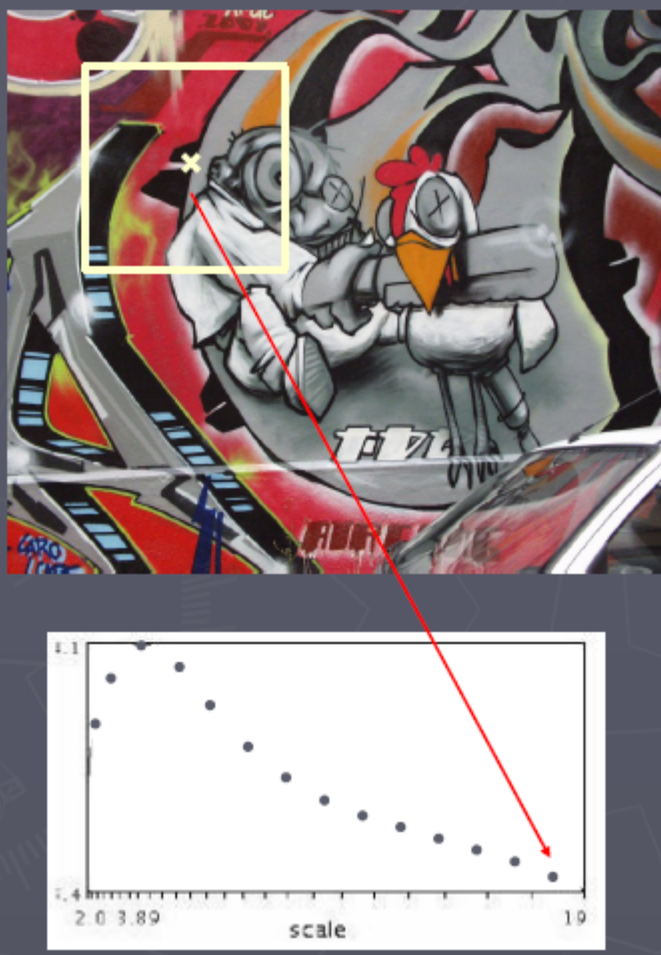


$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

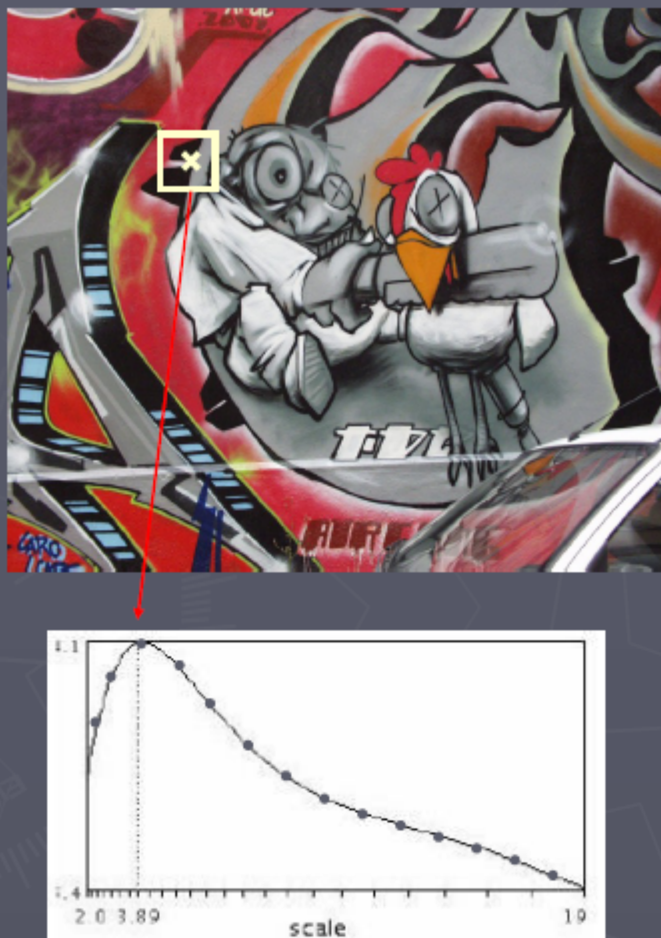


$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

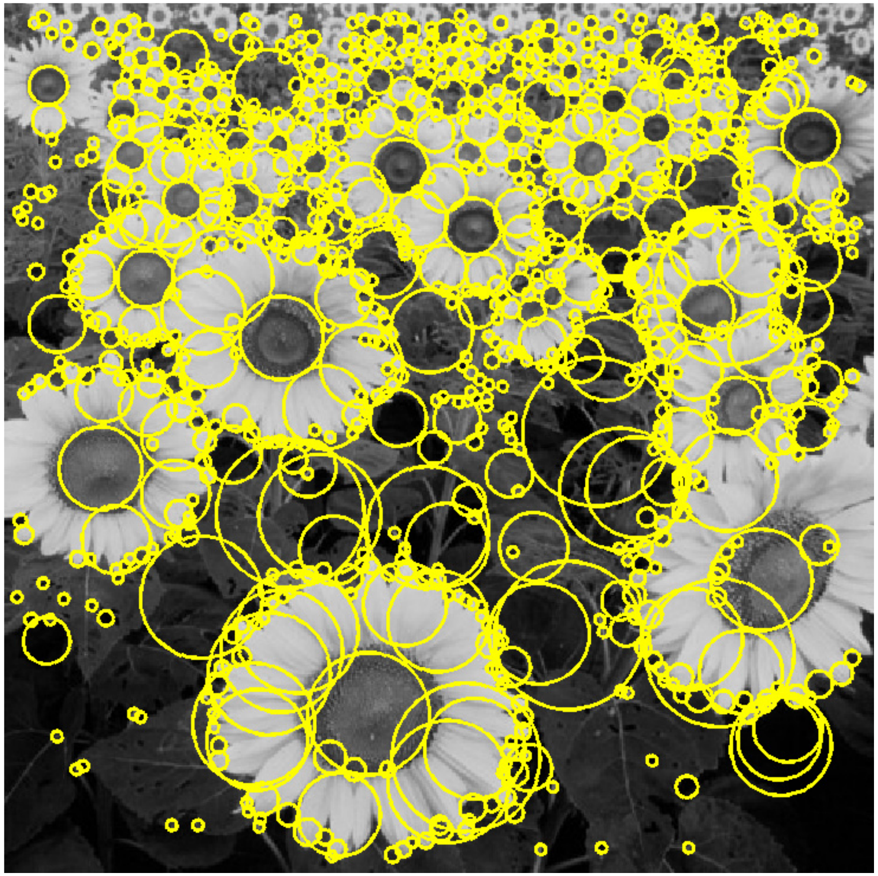


$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

Implementation

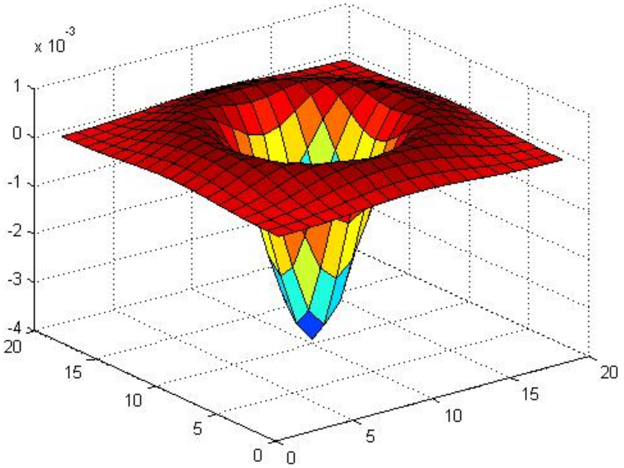
- Instead of computing *f* for larger and larger windows, we can implement using a fixed window size with a Gaussian pyramid

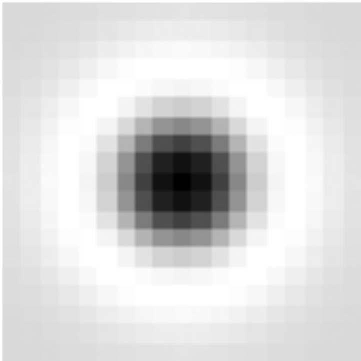
Feature extraction: Corners and blobs



Another common definition of *f*

- The *Laplacian of Gaussian* (*LoG*)



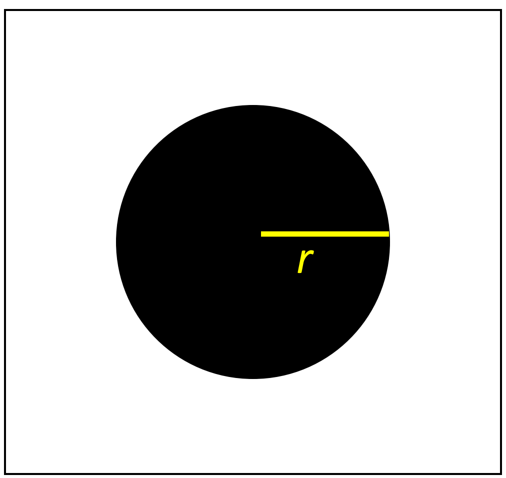


$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

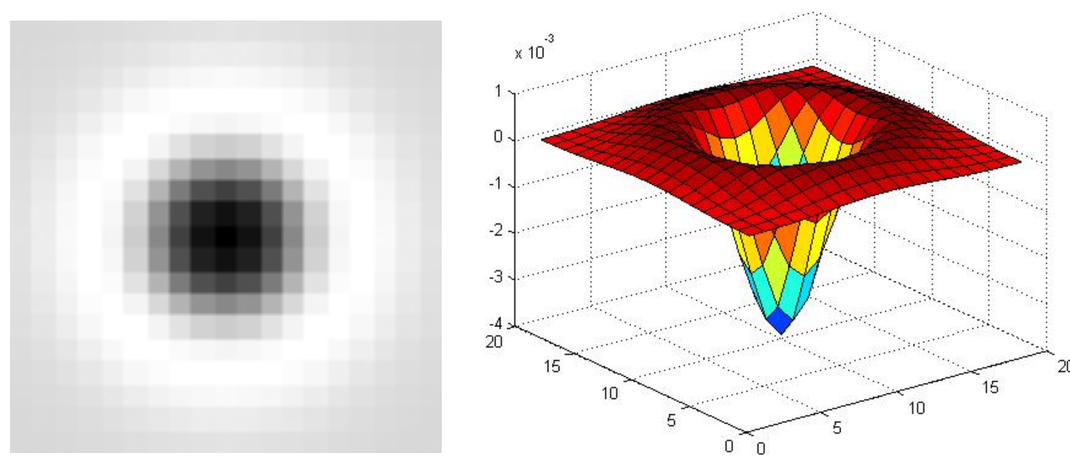
(very similar to a Difference of Gaussians (DoG) – i.e. a Gaussian minus a slightly smaller Gaussian)

Scale selection

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



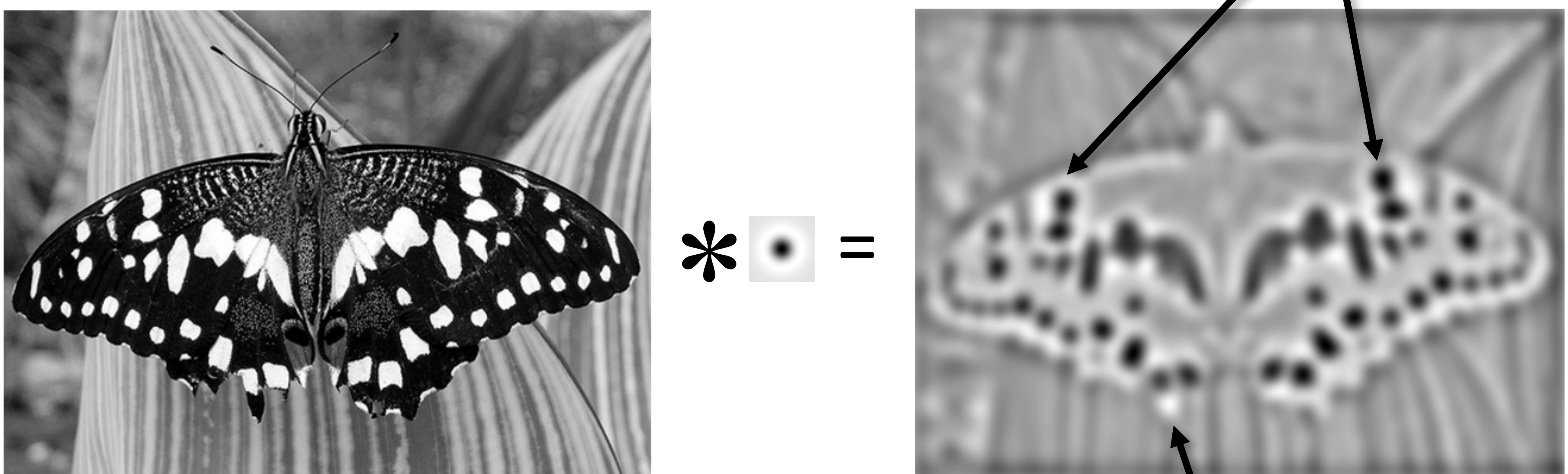
image



Laplacian

Laplacian of Gaussian

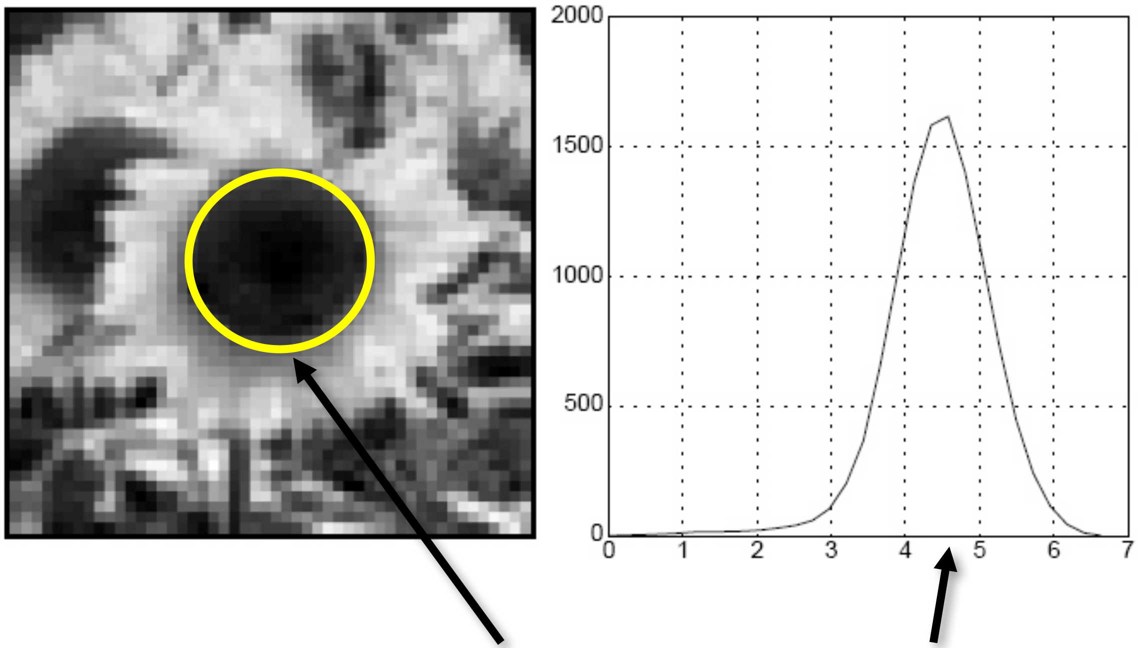
- “Blob” detector



- Find maxima *and minima* of LoG operator in space and scale

Characteristic scale

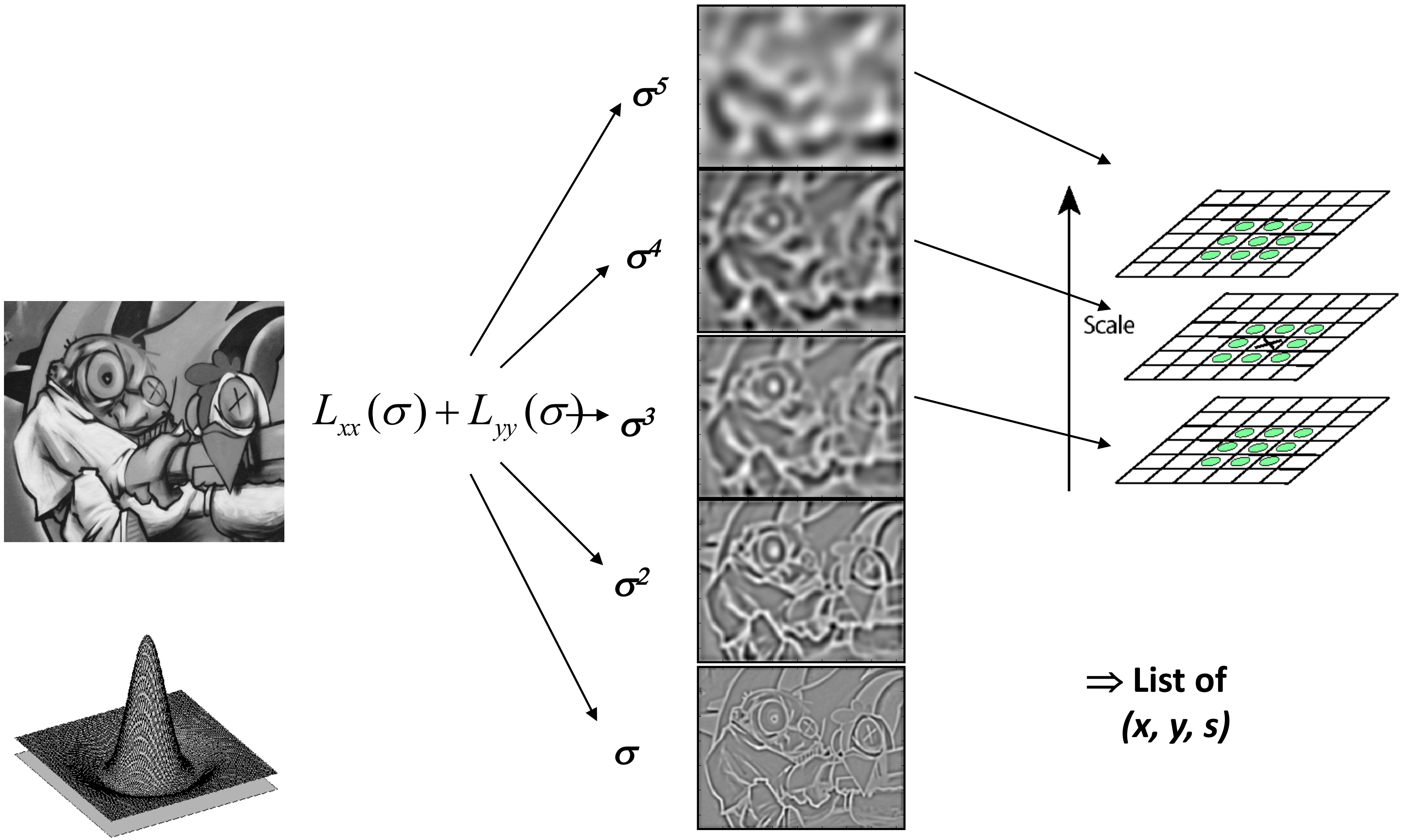
- The scale that produces peak of Laplacian response



characteristic scale

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

Find local maxima in position-scale space



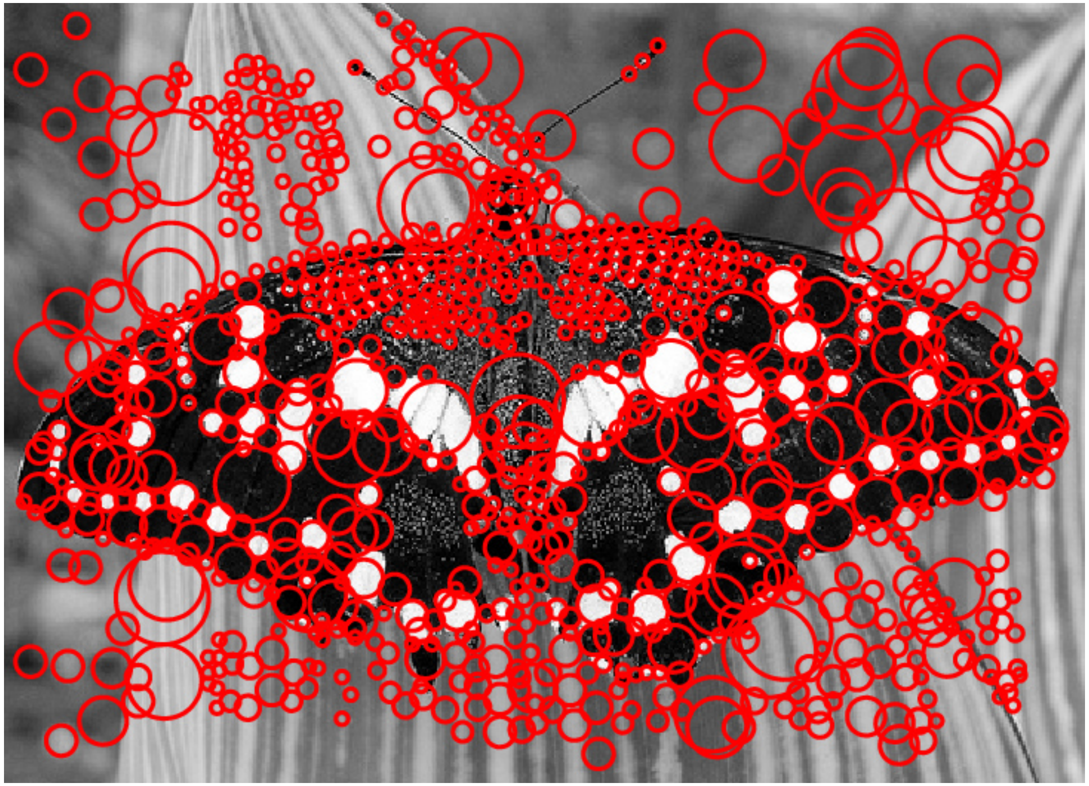
K. Grauman, B. Leibe

Scale-space blob detector: Example

Scale-space blob detector: Example

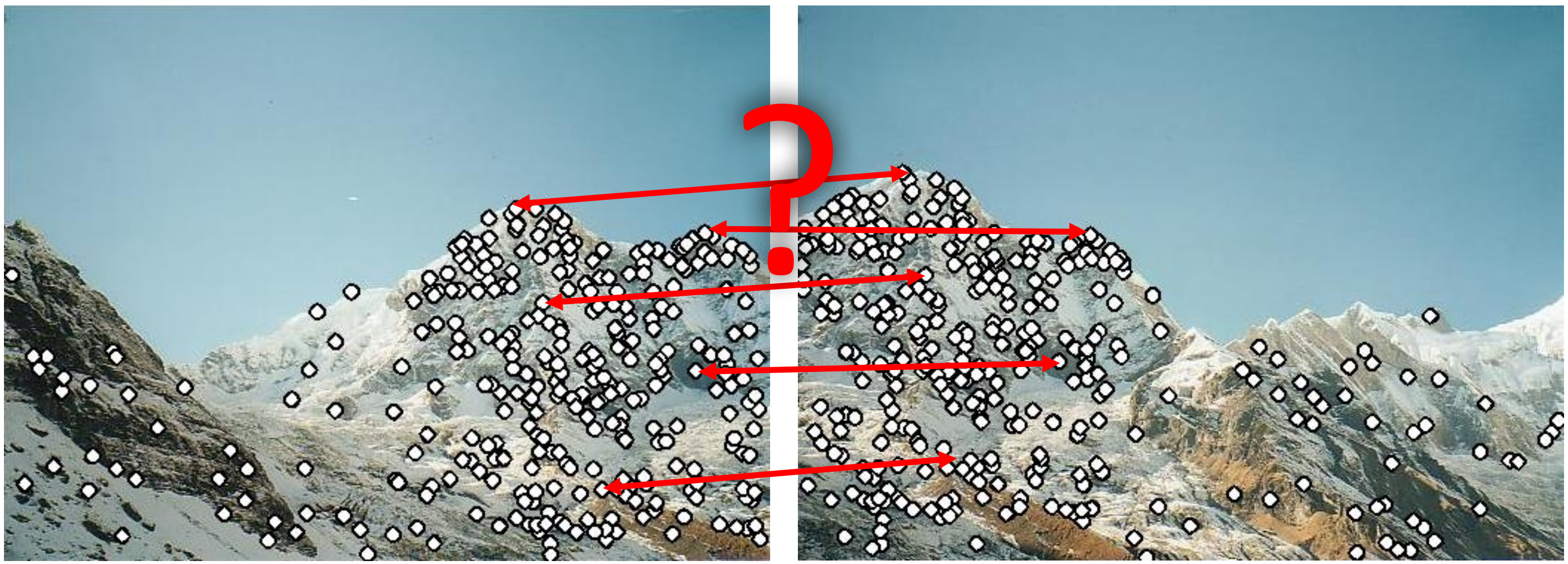
sigma = 11.9912

Scale-space blob detector: Example



Matching feature points

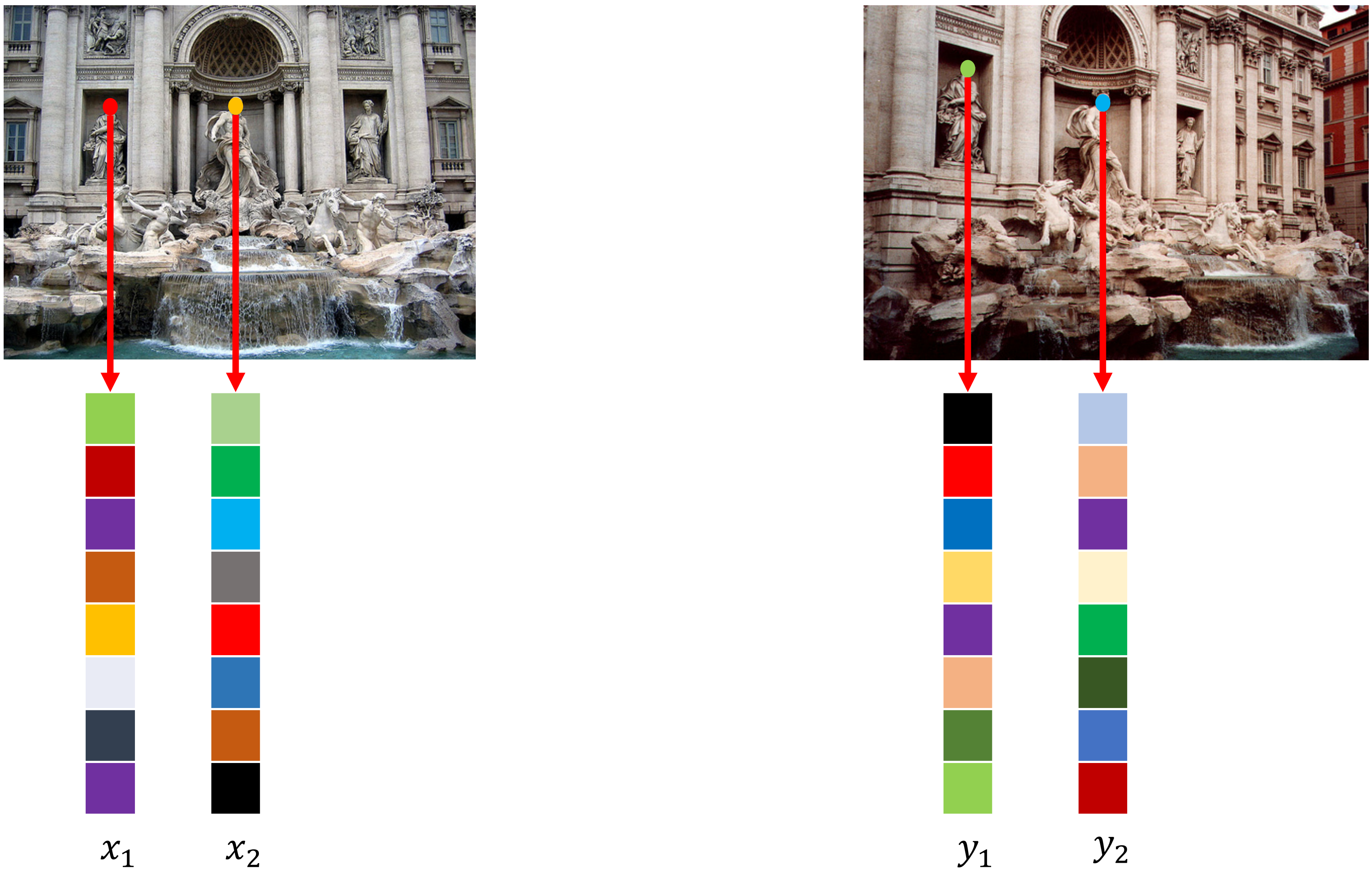
We know how to detect good points
Next question: **How to match them?**



Two interrelated questions:

1. How do we *describe* each feature point?
2. How do we *match* descriptions?

Feature descriptor



Feature matching

- Measure the distance between (or similarity between) every pair of descriptors

	$\boldsymbol{y_1}$	$\boldsymbol{y_2}$
$\boldsymbol{x_1}$	$d(x_1, y_1)$	$d(x_1, y_2)$
$\boldsymbol{x_2}$	$d(x_2, y_1)$	$d(x_2, y_2)$

Invariance vs. discriminability

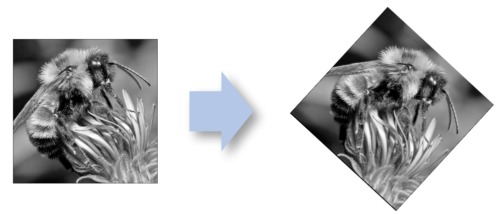
- Invariance:
  - Distance between descriptors should be small even if image is transformed

- Discriminability:
  - Descriptor should be highly unique for each point (far away from other points in the image)

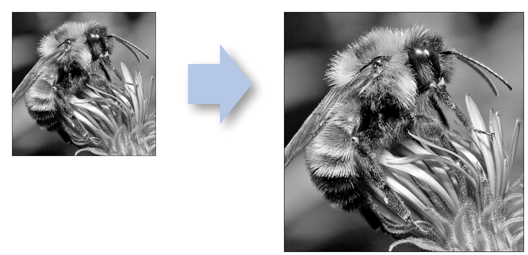
Image transformations

- Geometric

Rotation

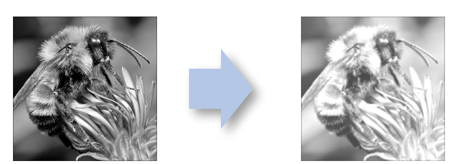


Scale



- Photometric

Intensity change



Invariance

- Most feature descriptors are designed to be invariant to
  - Translation, 2D rotation, scale

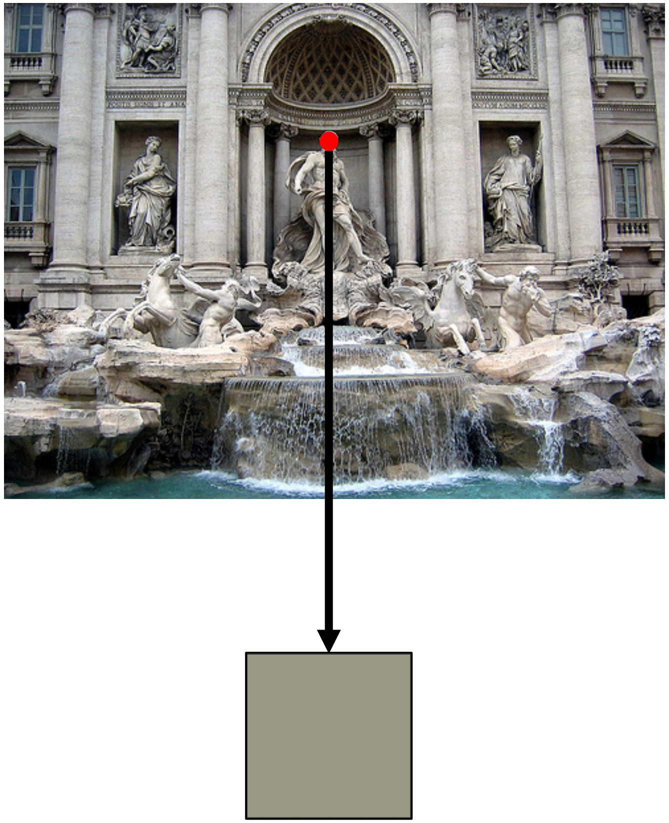
- They can usually also handle
  - Limited 3D rotations (SIFT works up to about 60 degrees)
  - Limited affine transformations (some are fully affine invariant)
  - Limited illumination/contrast changes

How to achieve invariance

Design an invariant feature descriptor

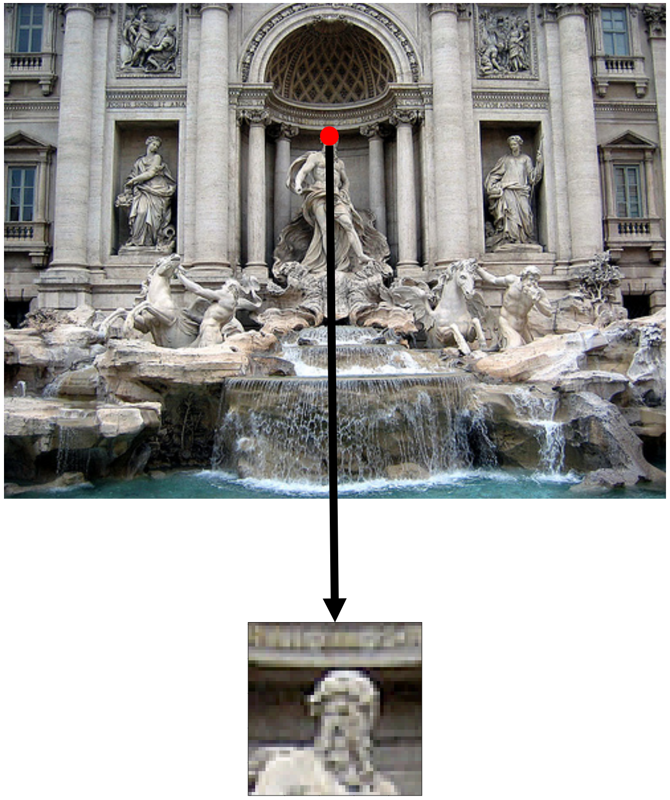
- Simplest descriptor: a single 0
  - What’s this invariant to?
  - Is this discriminative?
- Next simplest descriptor: a single pixel
  - What’s this invariant to?
  - Is this discriminative?

The aperture problem



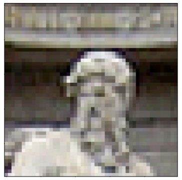
The aperture problem

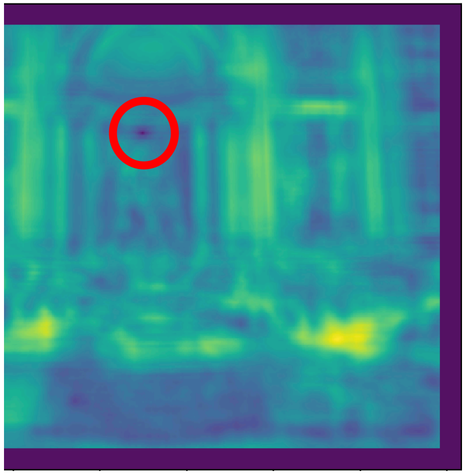
- Use a whole patch instead of a pixel?



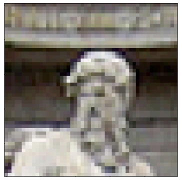
SSD

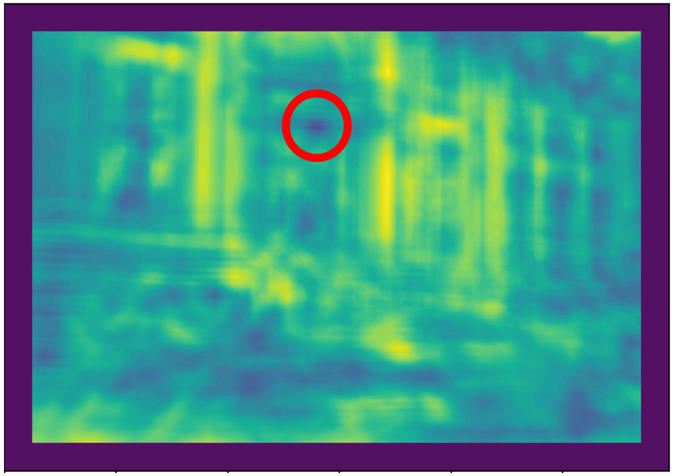
- Use as descriptor the whole patch
- Match descriptors using euclidean distance
- $d(x, y) = ||x - y||^2$



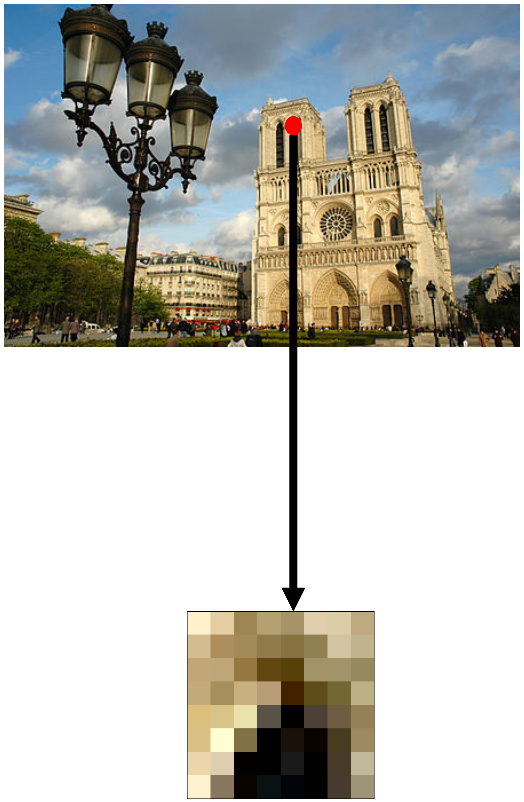


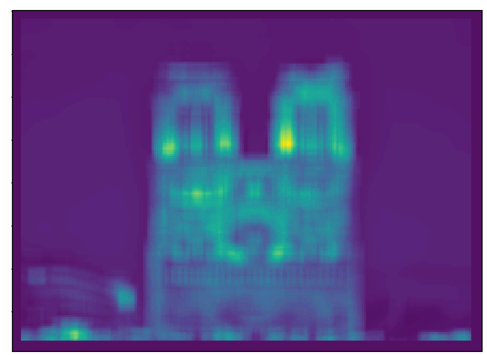
SSD





SSD

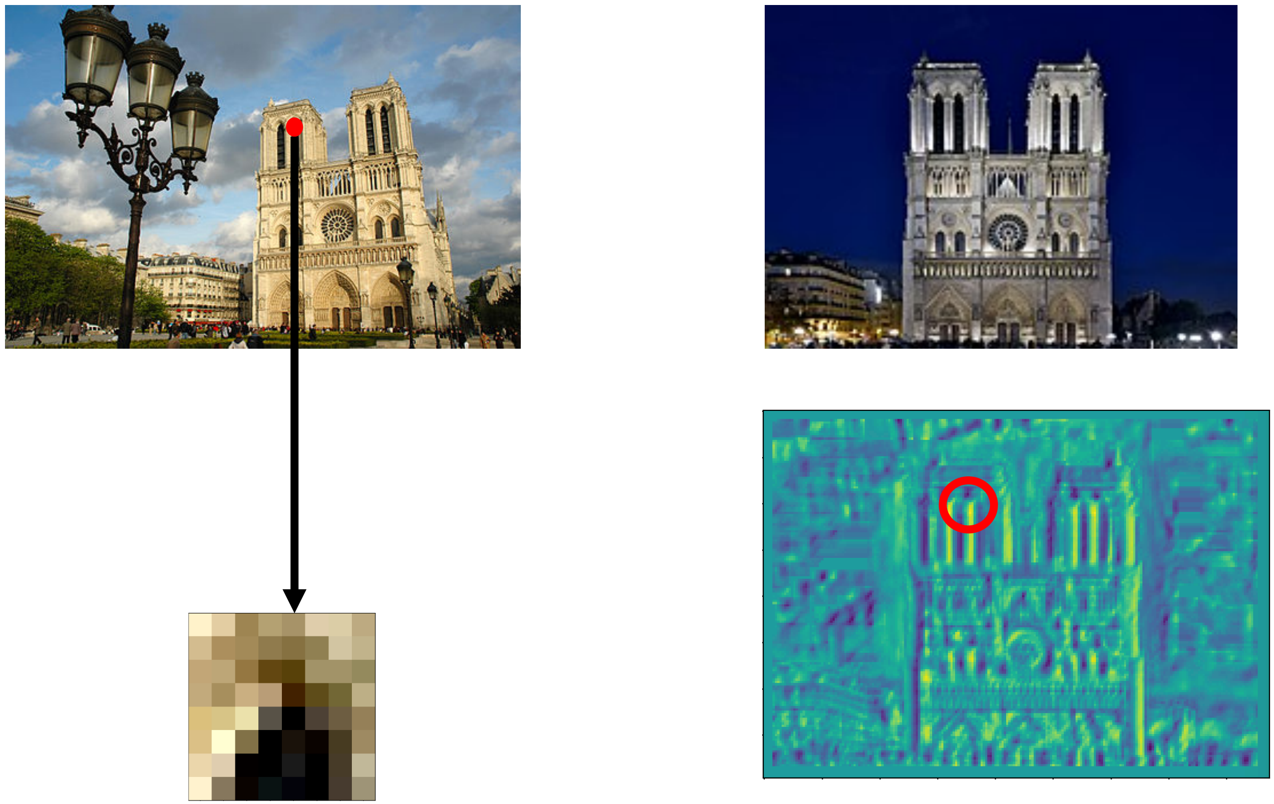




NCC - Normalized Cross Correlation

- Lighting and color change pixel intensities
- Example: increase brightness / contrast
- $I' = \alpha I + \beta$
- Subtract patch mean: invariance to β
- Divide by norm of vector: invariance to α
- $x' = x - \langle x \rangle$
- $x'' = \frac{x'}{||x'||}$
- *similarity* = $x'' \cdot y''$

NCC - Normalized cross correlation



Basic correspondence

- Image patch as descriptor, NCC as similarity
- Invariant to?
  - Photometric transformations?
  - Translation?
  - Rotation?

Rotation invariance for feature descriptors

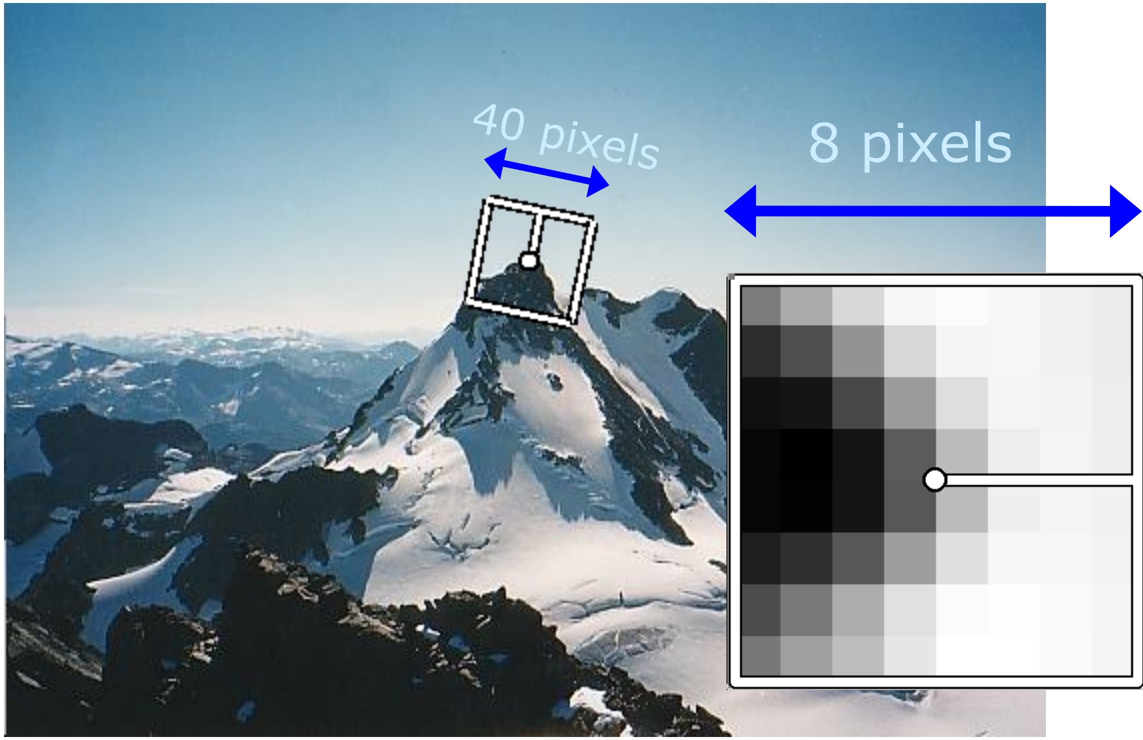
- Find dominant orientation of the image patch
  - This is given by $\mathbf{x}_{\mathbf{max}}$, the eigenvector of **M** corresponding to λ_{max} (the *larger* eigenvalue)
  - Rotate the patch according to this angle

Figure by Matthew Brown

Multiscale Oriented PatcheS descriptor

Take 40x40 square window around detected feature

- Scale to 1/5 size (using prefiltering)
- Rotate to horizontal
- Sample 8x8 square window centered at feature
- Intensity normalize the window by subtracting the mean, dividing by the standard deviation in the window



Adapted from slide by Matthew Brown

Detections at multiple scales

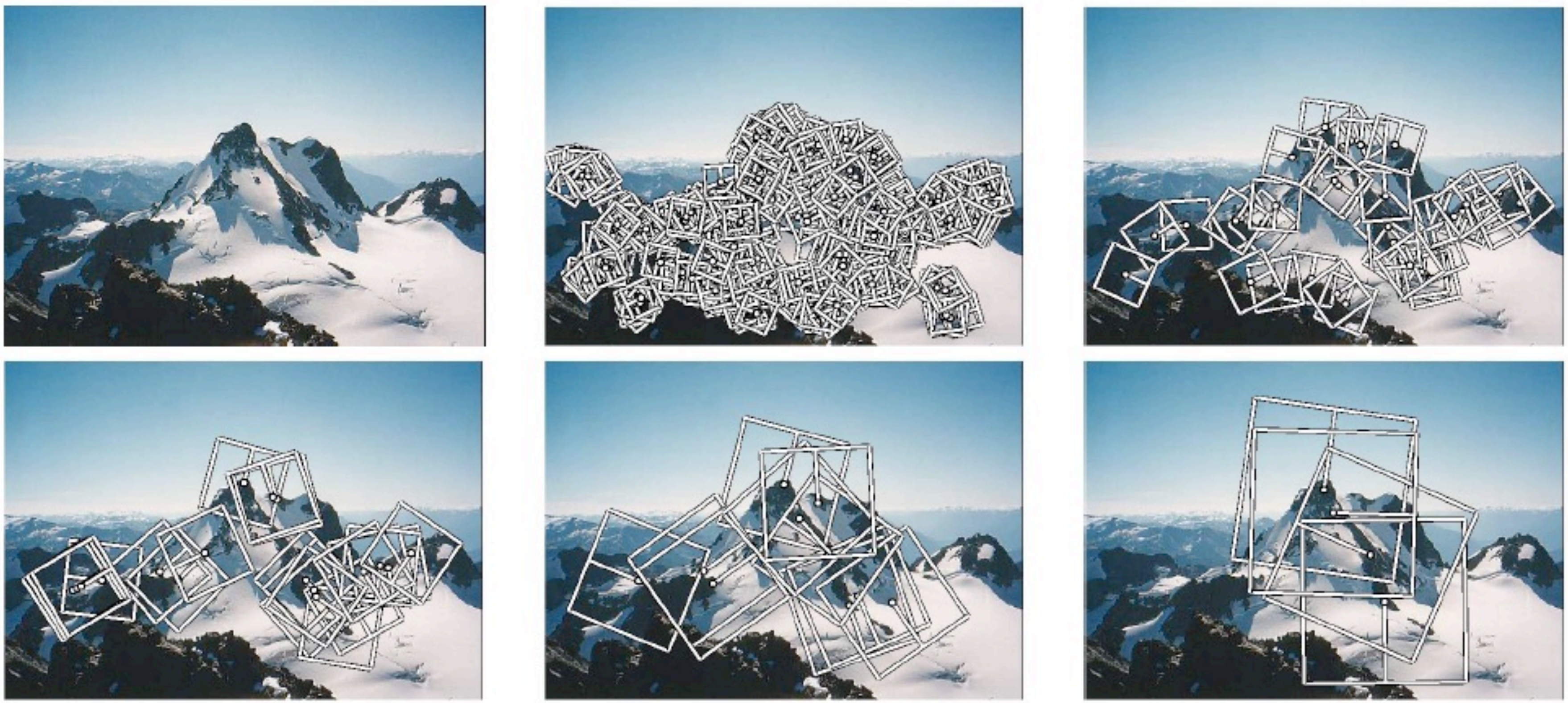


Figure 1. Multi-scale Oriented Patches (MOPS) extracted at five pyramid levels from one of the Matier images. The boxes show the feature orientation and the region from which the descriptor vector is sampled.

Feature matching

Given a feature in I_1, how to find the best match in I_2?

1. Define distance function that compares two descriptors
2. Test all the features in I_2, find the one with min distance