Interest Points for Image Representation

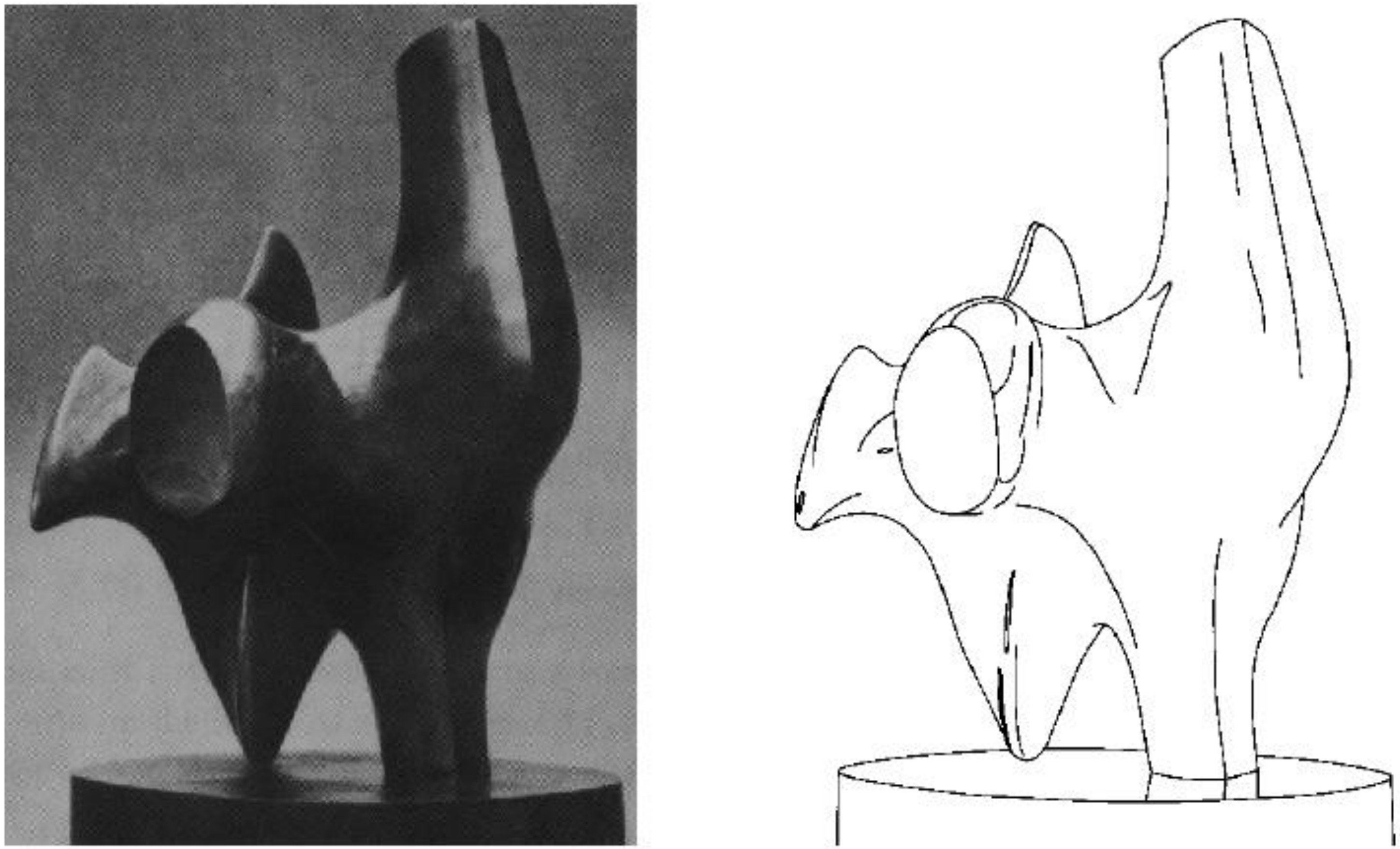
Jianping Fan
Dept of Computer Science
UNC-Charlotte

Course Website:
http://webpages.uncc.edu/jfan/itcs5152.html

Corner Point Detection

----Harris Corner Detector

What edge image can tell us?

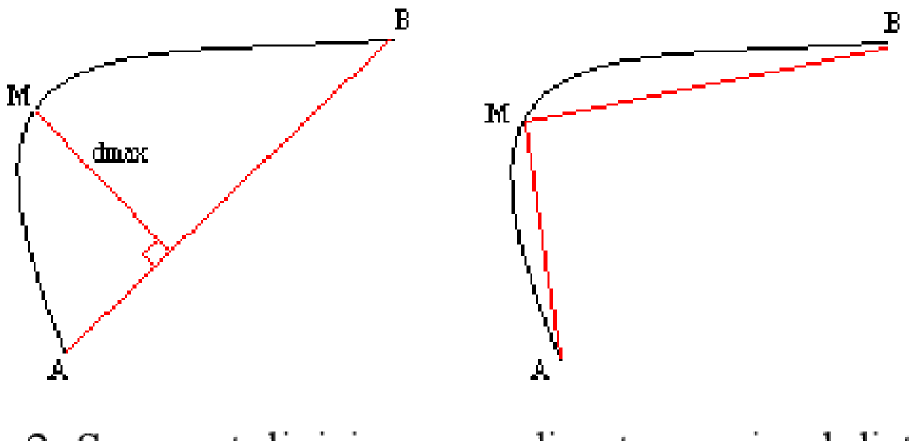


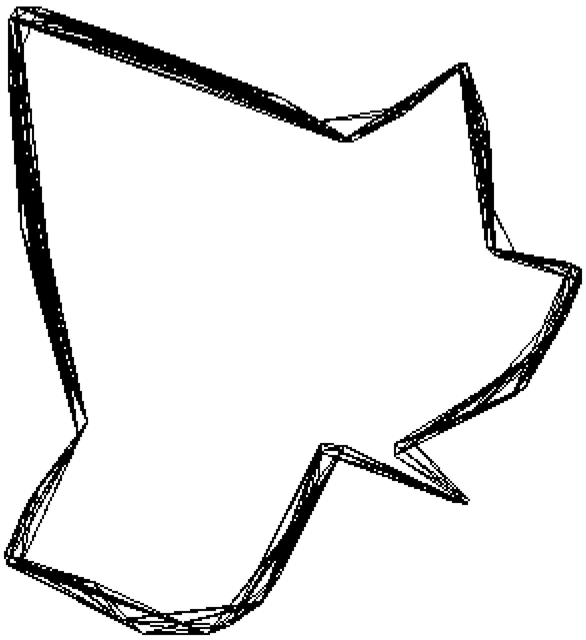
Object Boundaries & Structures

What edge image can tell us?

Object Boundaries & Structures

Corner Points on Edge





Polygon Approximation for Edge Representation

What edge image can tell us?

Image

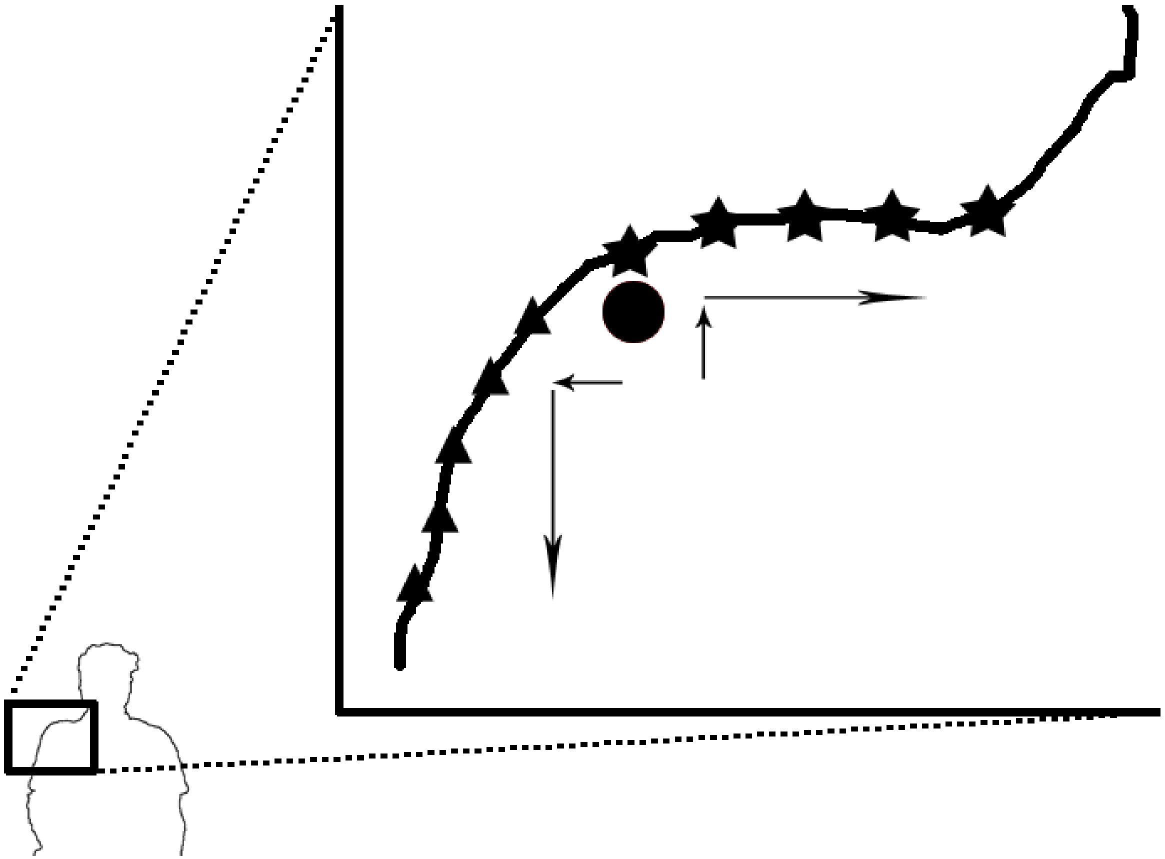
Edge Image

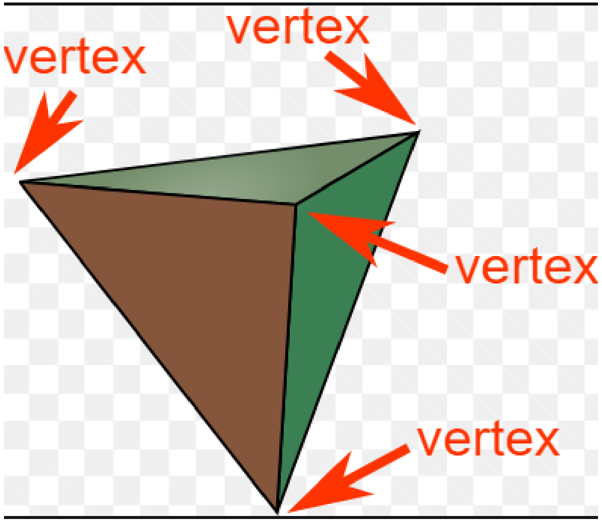
Object Boundaries
& Surfaces

Interest Points
(Corner Points)

Image Representation

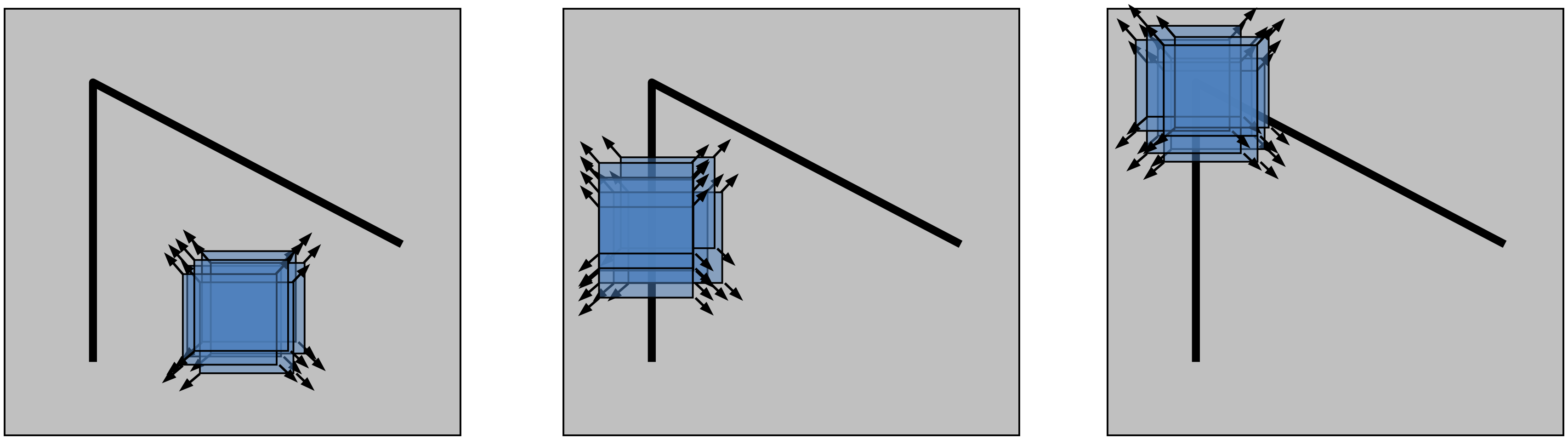
Corner Points on Edge





More Compact Image Representation via Corners

Harris Detector: Intuition



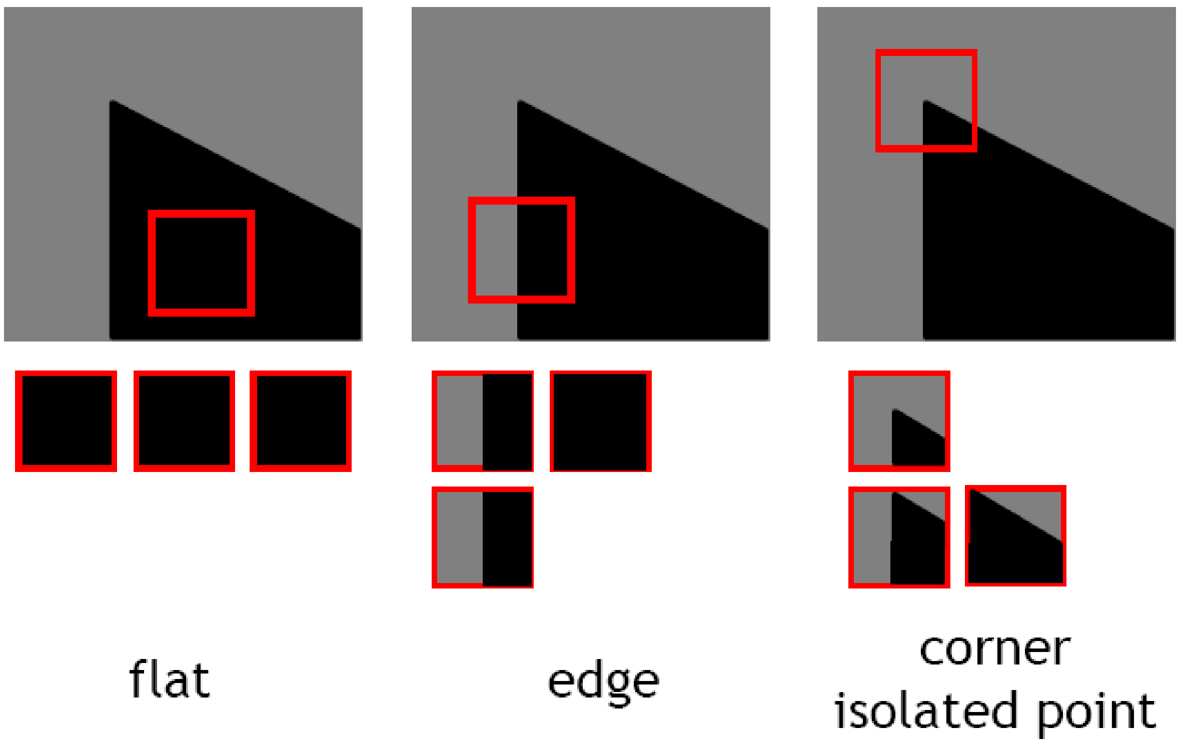
"flat" region:
no change
in all
directions

"edge":
no change
along the
edge direction

"corner":
**significant
change in all
directions**

Corner Detector

- **Shift in any direction would result in a significant change at a corner.**



Algorithm:

- Shift in horizontal, vertical, and diagonal **directions** by one pixel.
- Calculate the **absolute value of the MSE** for each shift.
- Take the minimum as the **cornerness response**.

How to define & calculate cornerness response?

Harris Detector: Mathematics

Change of intensity for the shift [*u*,*v*]:

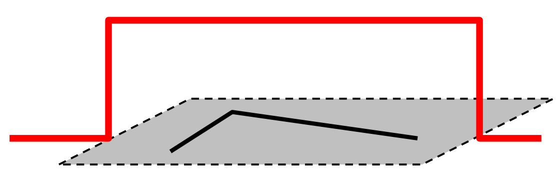
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

Window function

Shifted intensity

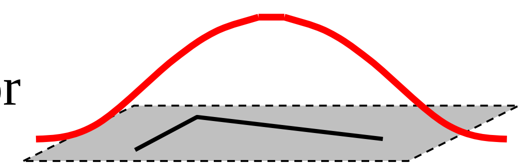
Intensity

Window function $w(x,y)$ =



1 in window, 0 outside

or



Gaussian

Neighborhood pattern, weights, …….

Harris Detector: Mathematics

Apply Taylor series expansion of Intensity Change:

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

$$= \sum_{x,y} w(x,y)\left[I_x u + I_y v + O(u^2,v^2)\right]^2$$

$$E(u,v) = Au^2 + 2Cuv + Bv^2$$

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

$$B = \sum_{x,y} w(x,y) I_y^2(x,y)$$

$$C = \sum_{x,y} w(x,y) I_x(x,y) I_y(x,y)$$

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

Harris Detector: Mathematics

- **Hessian Matrix**

$$\mathbf{H} = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1\,\partial x_n} \\[2.2ex] \dfrac{\partial^2 f}{\partial x_2\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2\,\partial x_n} \\[2.2ex] \vdots & \vdots & \ddots & \vdots \\[2.2ex] \dfrac{\partial^2 f}{\partial x_n\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Harris Detector: Mathematics

For small shifts [*u*,*v*] we have the following approximation:

$$E(u,v) \cong [u,v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

where *M* is a 2×2 matrix computed from **image derivatives**:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Corner Detection: Mathematics

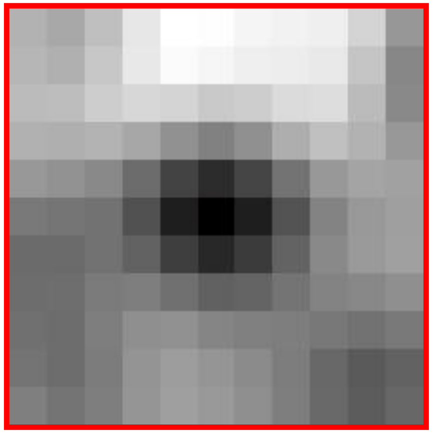
Local quadratic approximation of *E*(*u*,*v*) in the neighborhood of (0,0) is given by the *second-order Taylor expansion*:

$$E(u,v) \approx E(0,0) + [u \;\; v]\begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2}[u \;\; v]\begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{uv}(0,0) & E_{vv}(0,0) \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}$$

Always 0

First derivative is 0

$E(u, v)$



Corner Detection: Mathematics

The quadratic approximation simplifies to

$$E(u,v) \approx [u \ \ v] \ M \begin{bmatrix} u \\ v \end{bmatrix}$$

where M is a *second moment matrix* computed from image derivatives:

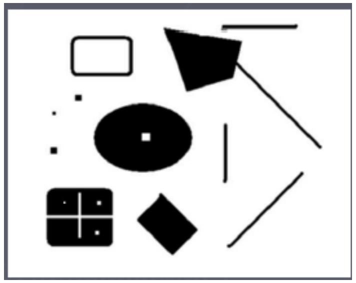
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

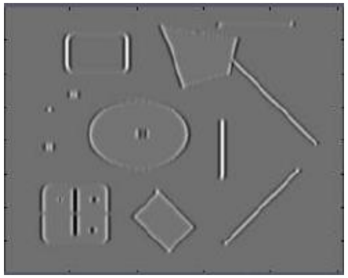
$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y] = \sum \nabla I (\nabla I)^T$$

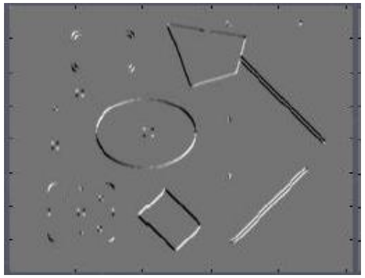
Corners as distinctive interest points

$$M = \sum w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

2 x 2 matrix of image derivatives (averaged in neighborhood of a point).







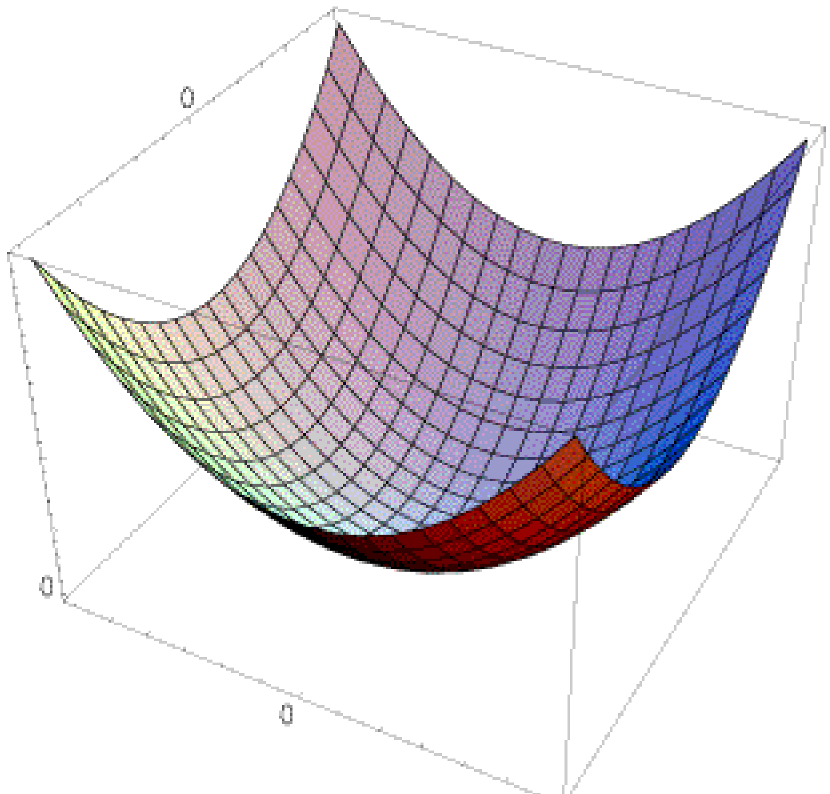
Notation: $I_x \Leftrightarrow \frac{\partial I}{\partial x}$ $I_y \Leftrightarrow \frac{\partial I}{\partial y}$ $I_x I_y \Leftrightarrow \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}$

Interpreting the second moment matrix

The surface $E(u,v)$ is locally approximated by a quadratic form. Let's try to understand its shape.

$$E(u,v) \approx [u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

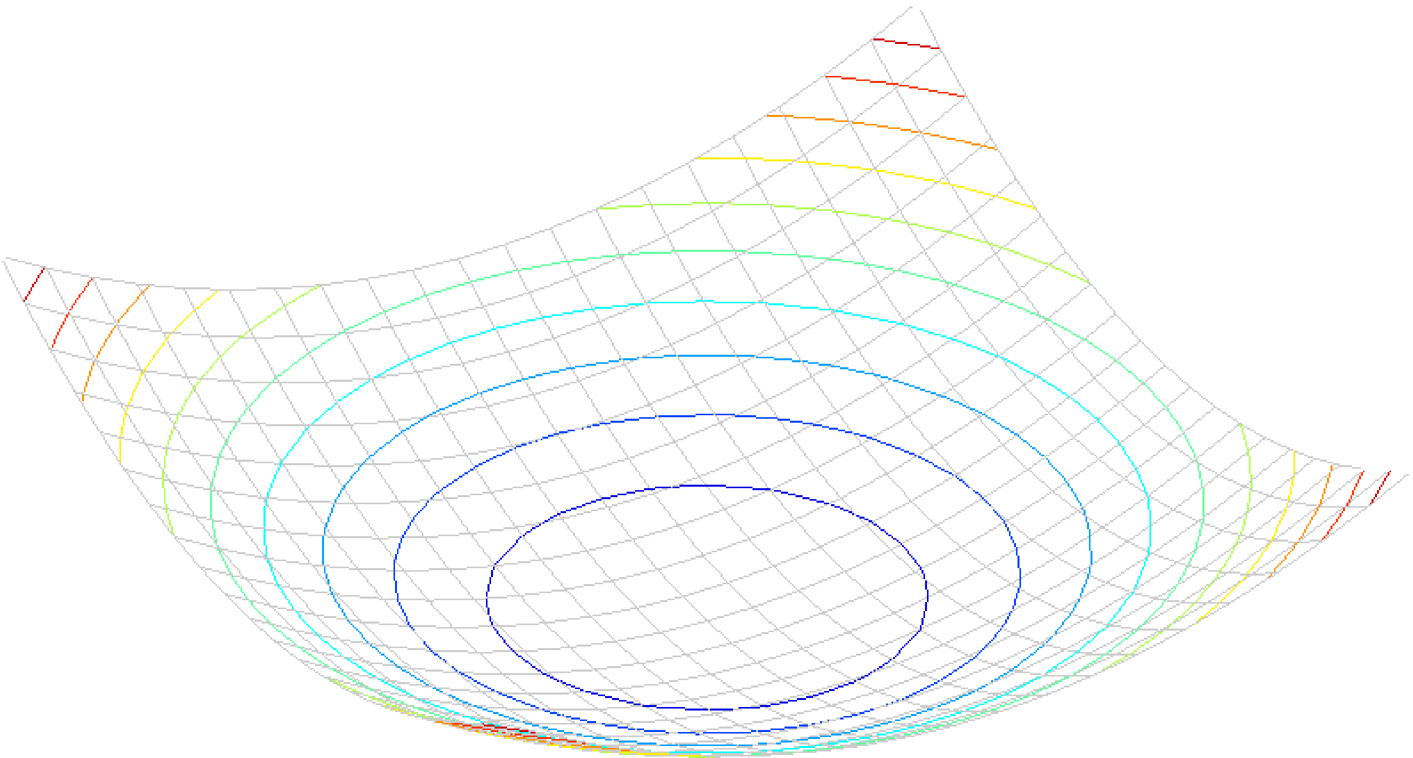
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$



Interpreting the second moment matrix

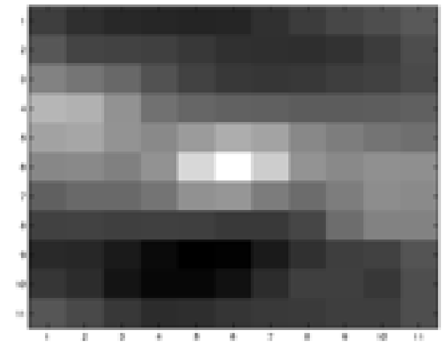
Consider a horizontal "slice" of $E(u, v)$: $[u \ \ v] \ M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

This is the equation of an ellipse.



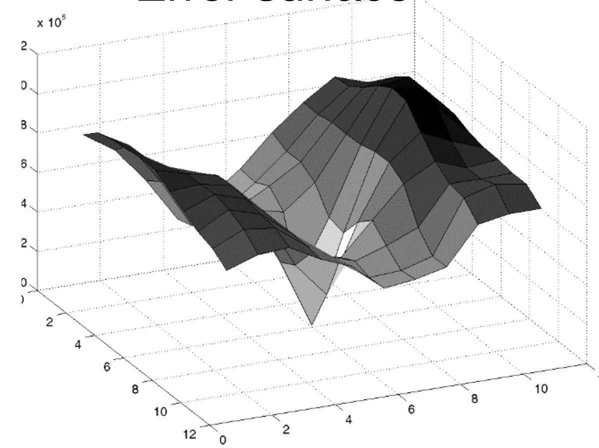
Selecting Good Features

Image patch



12x10^5

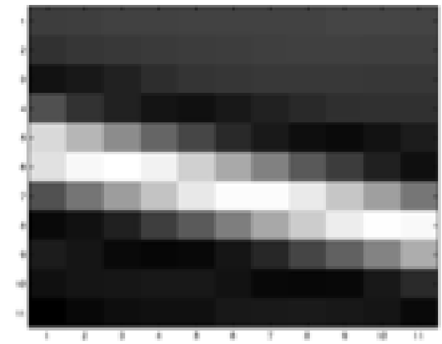
Error surface



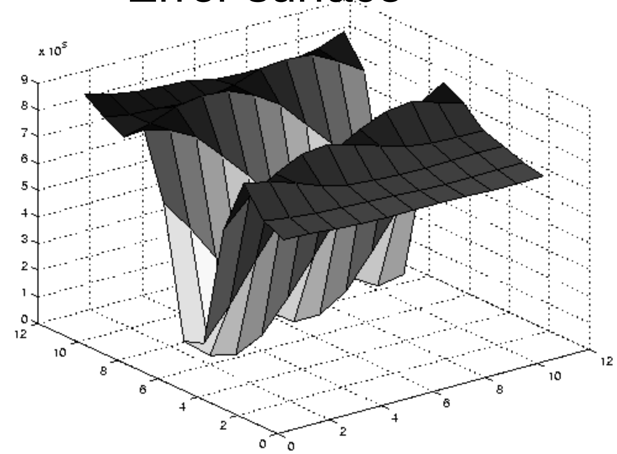
λ_1 and λ_2 are large

Selecting Good Features

Image patch



9x10^5 Error surface

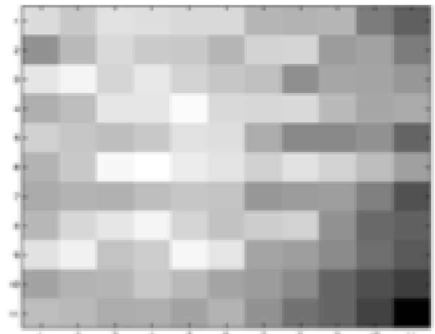


large λ_1, small λ_2

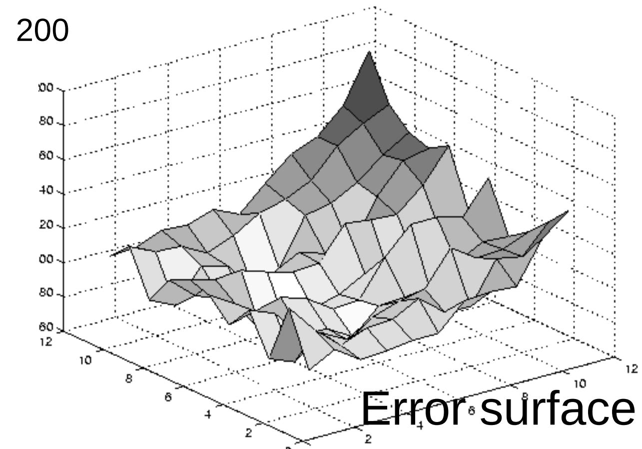
Darya Frolova, Denis Simakov The Weizmann Institute of Science
http://www.wisdom.weizmann.ac.il/~deniss/vision_spring04/files/InvariantFeatures.ppt

Selecting Good Features

Image patch



(contrast auto-scaled)



Error surface

(vertical scale exaggerated relative to previous plots)

small λ_1, small λ_2

Darya Frolova, Denis Simakov The Weizmann Institute of Science
http://www.wisdom.weizmann.ac.il/~deniss/vision_spring04/files/InvariantFeatures.ppt

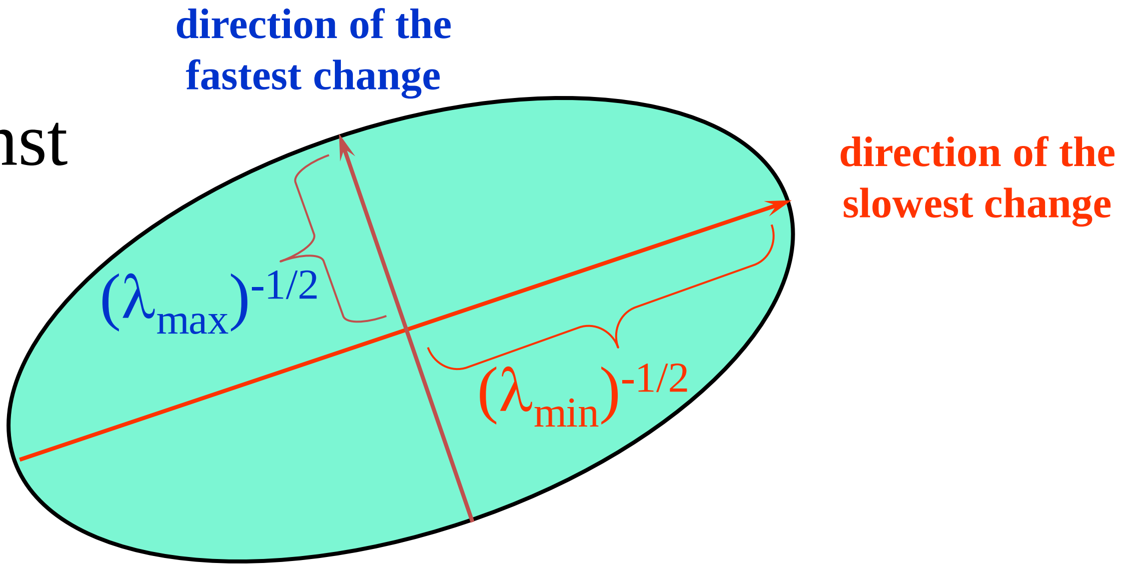
Harris Detector: Mathematics

Intensity change in shifting window: **eigenvalue analysis**

$$E(u,v) \cong [u,v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

λ_1, λ_2 – eigenvalues of M

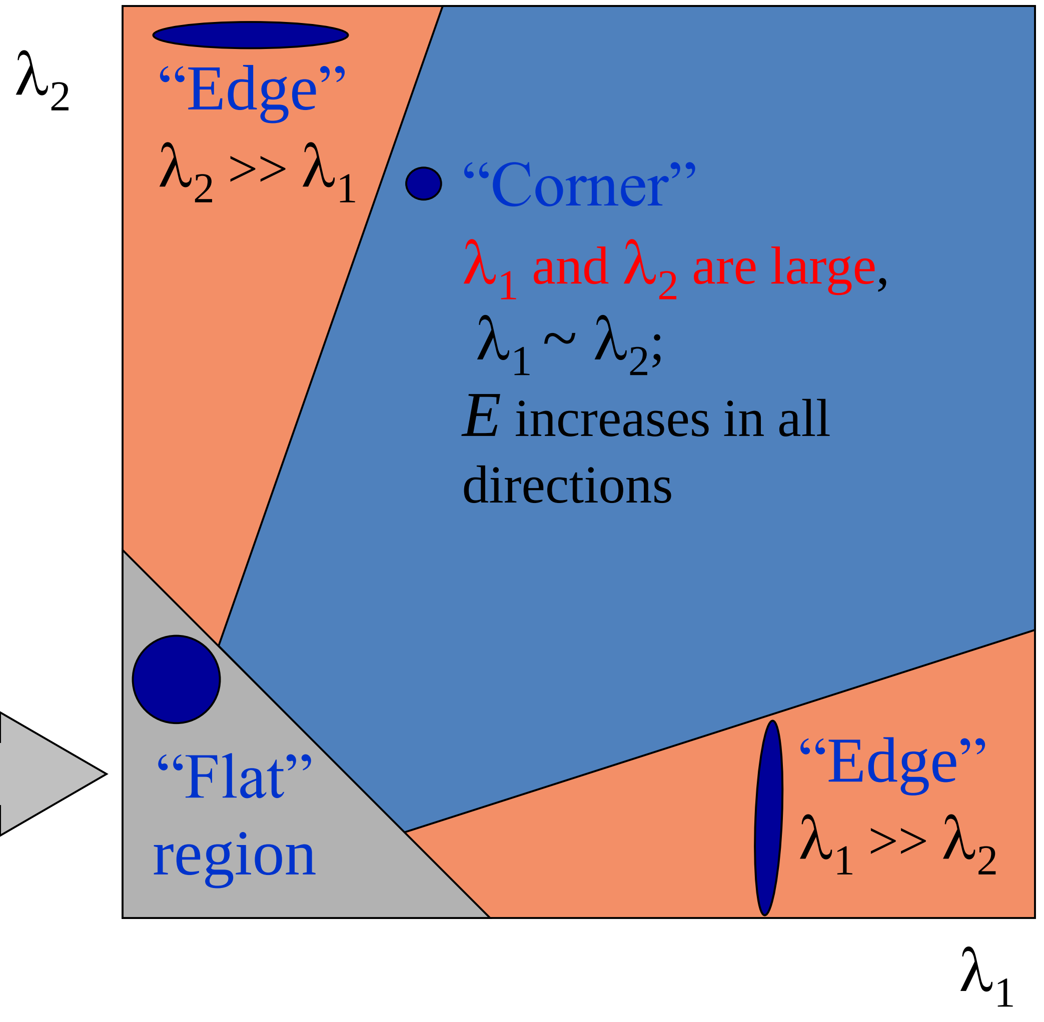
Ellipse $E(u,v)$ = const



Harris Detector: Mathematics

Classification of image points using eigenvalues of *M*:

λ_1 and λ_2 are small;
E is almost constant in all directions

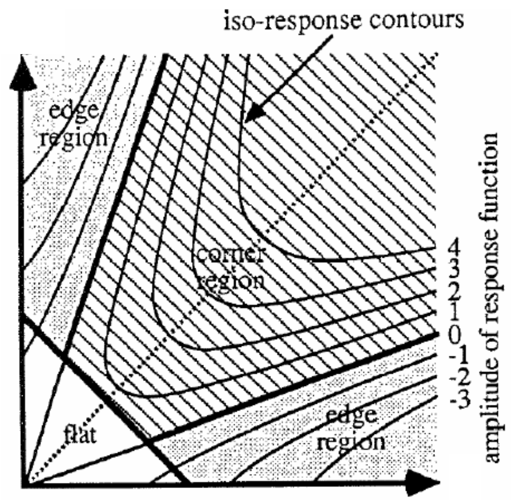


Harris corner detector

Measure of cornerness response R:

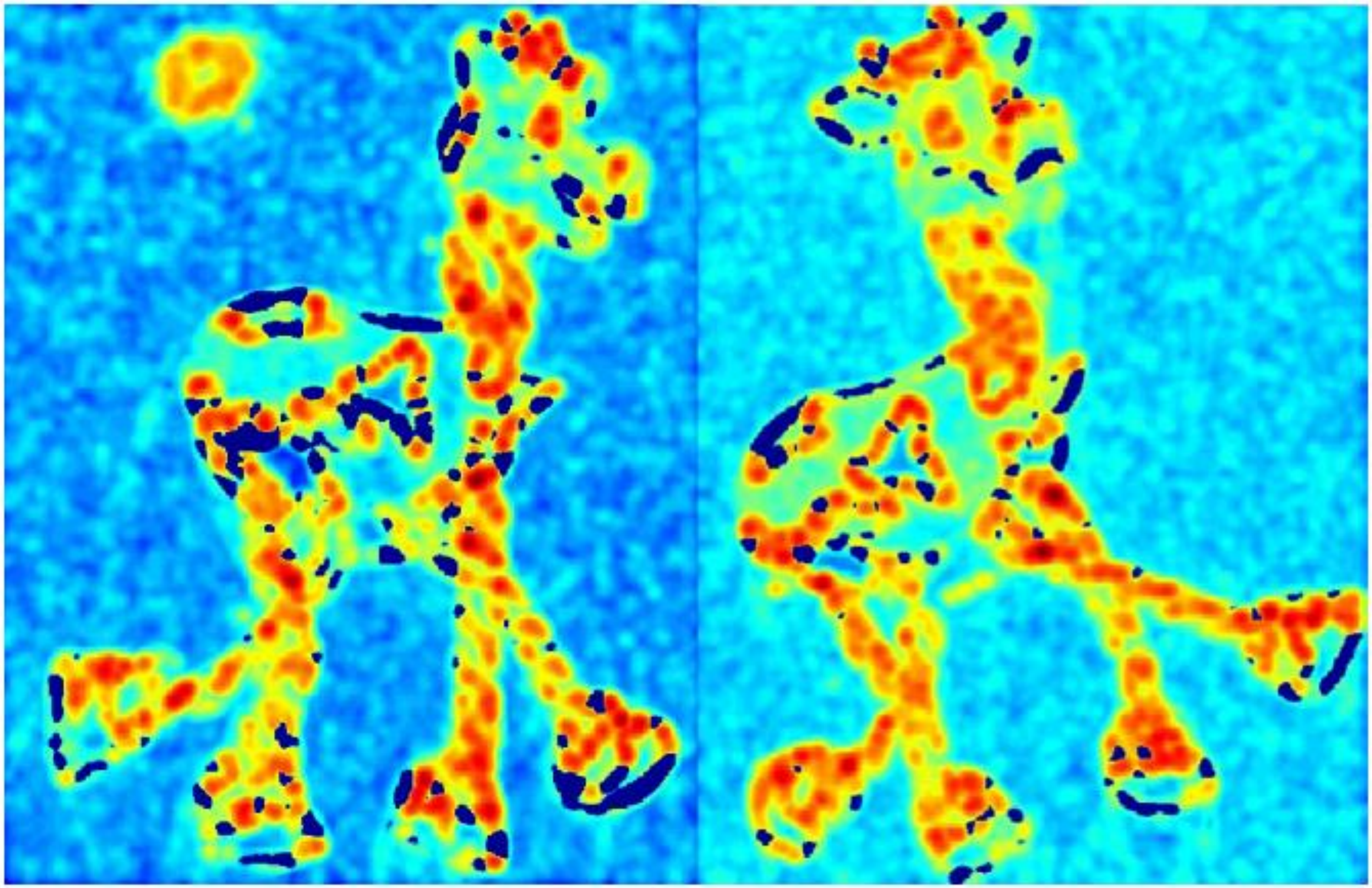
$$R = \det M - k\left(\operatorname{trace} M\right)^2$$

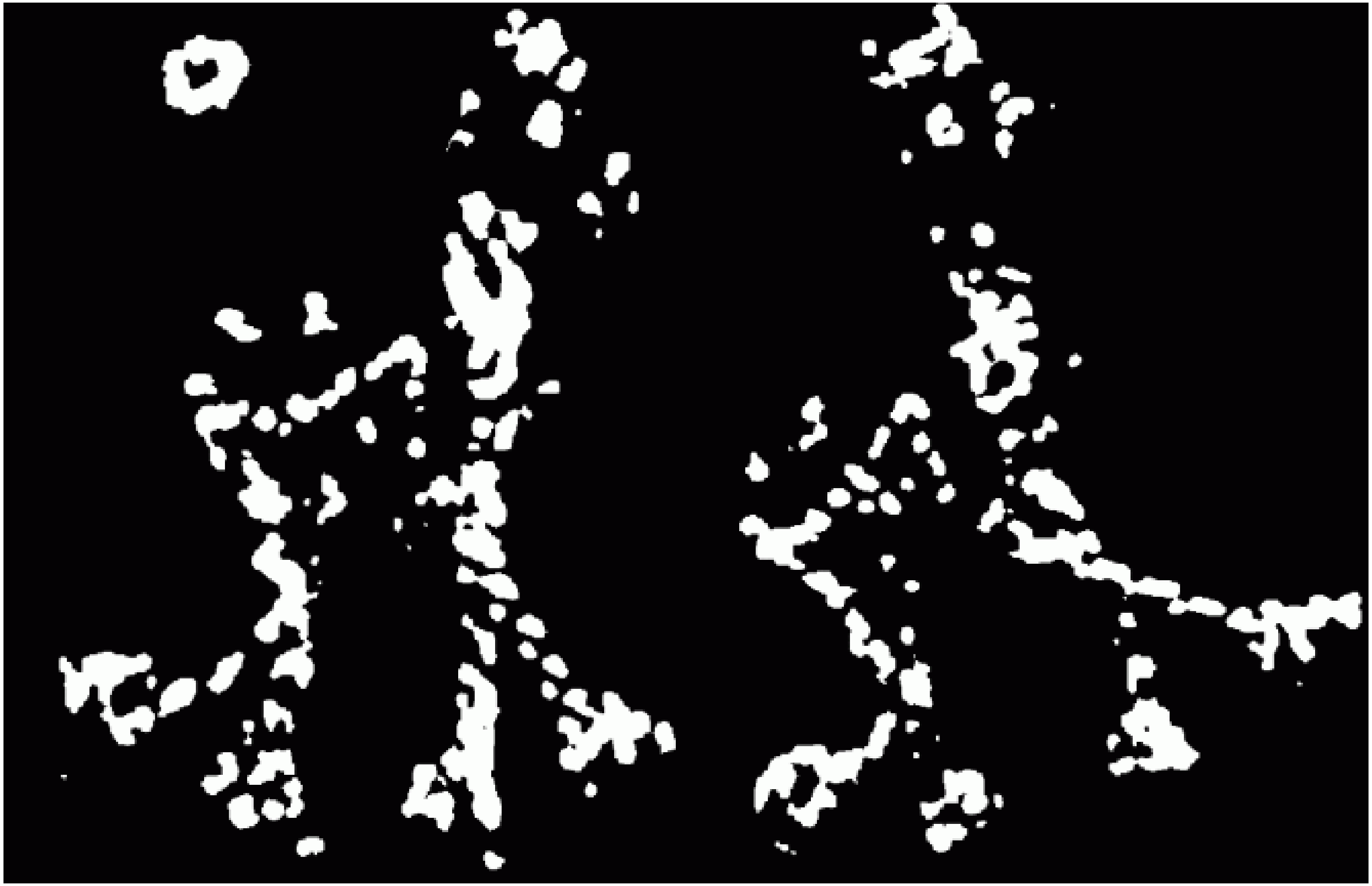
$$\det M = \lambda_1 \lambda_2$$
$$\operatorname{trace} M = \lambda_1 + \lambda_2$$



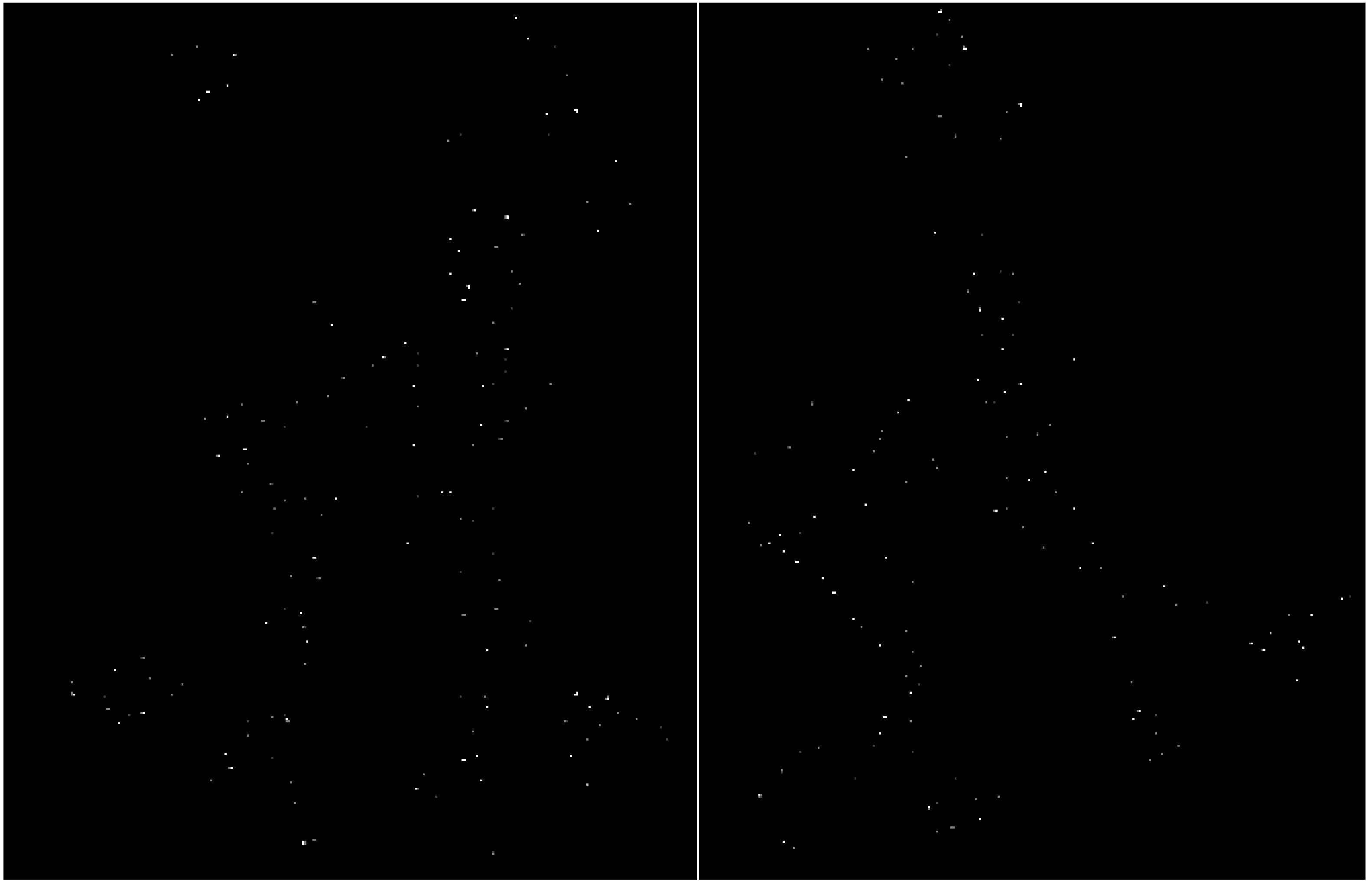
(k – empirical constant, k = 0.04-0.06)

No need to compute eigenvalues explicitly!





Eliminate small responses.



Find local maxima of the remaining.

Harris Detector: Scale

R_{min}= 0

R_{min}= 1500

Harris corner detector algorithm

- Compute image gradients I_x I_y for all pixels
- For each pixel
  - Compute

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

    by looping over neighbors x,y
  - compute

$$R = \det M - k\left(\operatorname{trace} M\right)^2$$

- Find points with large corner response function R (R > threshold)
- Take the points of locally maximum R as the detected feature points (ie, pixels where R is bigger than for all the 4 or 8 neighbors).

Darya Frolova, Denis Simakov The Weizmann Institute of Science

http://www.wisdom.weizmann.ac.il/~deniss/vision_spring04/files/InvariantFeatures.ppt

Key Steps for Harris Detector

1. Compute x and y derivatives of image

$$I_x = G_\sigma^x * I \quad I_y = G_\sigma^y * I$$

2. Compute products of derivatives at every pixel

$$I_{x2} = I_x . I_x \quad I_{y2} = I_y . I_y \quad I_{xy} = I_x . I_y$$

3. Compute the sums of the products of derivatives at each pixel

$$S_{x2} = G_{\sigma'} * I_{x2} \quad S_{y2} = G_{\sigma'} * I_{y2} \quad S_{xy} = G_{\sigma'} * I_{xy}$$

4. Define at each pixel (x, y) the matrix

$$H(x,y) = \begin{bmatrix} S_{x2}(x,y) & S_{xy}(x,y) \\ S_{xy}(x,y) & S_{y2}(x,y) \end{bmatrix}$$

5. Compute the response of the detector at each pixel

$$R = Det(H) - k(Trace(H))^2$$

6. Threshold on value of R. Compute nonmax suppression.

R is the cornerness response, corner points with local maximum of R

Summary of Harris Detector

- **Determinant of Matrix**

In the case of a 2 × 2 matrix the determinant may be defined as

$$|A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc.$$

$$|A| = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} =$$

$$= aei + bfg + cdh - ceg - bdi - afh.$$

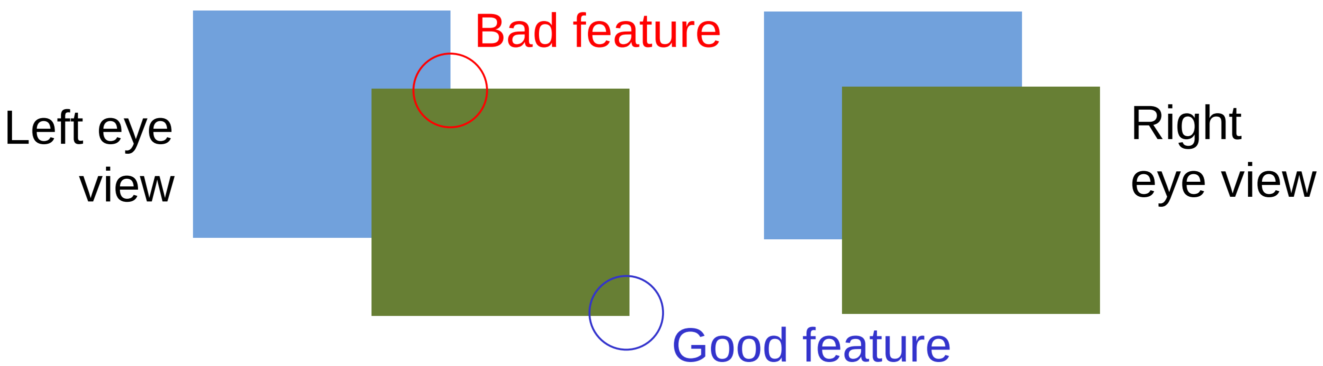
- **Trace of Matrix**

Let **A** be a matrix and its trace tr(**A**)

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \Rightarrow \operatorname{tr}(\mathbf{A}) = \sum_{i=1}^{3} a_{ii} = a_{11} + a_{22} + a_{33}$$

Selecting Good Features

- What's a "good feature"?
  - Satisfies brightness constancy—looks the same in both images
  - Has sufficient texture variation
  - Does not have too much texture variation
  - Corresponds to a "real" surface patch—see below:



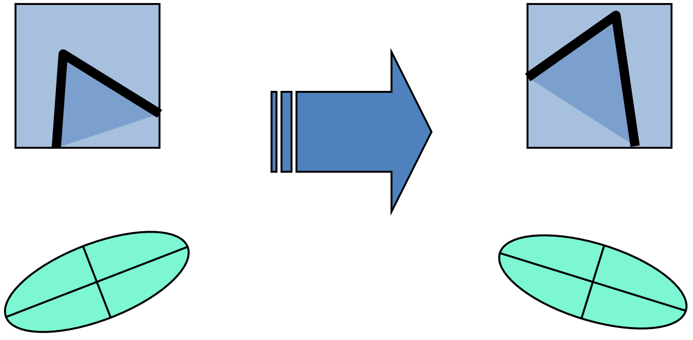
  - Does not deform too much over time

Properties of Corner Points

- **Rotation invariance**
- **Partial invariance to *affine intensity* change**
- But: **non-invariant to *image scale*!**

Harris Detector: Some Properties

- **Rotation invariance**



Ellipse rotates but its shape (i.e. eigenvalues) remains the same

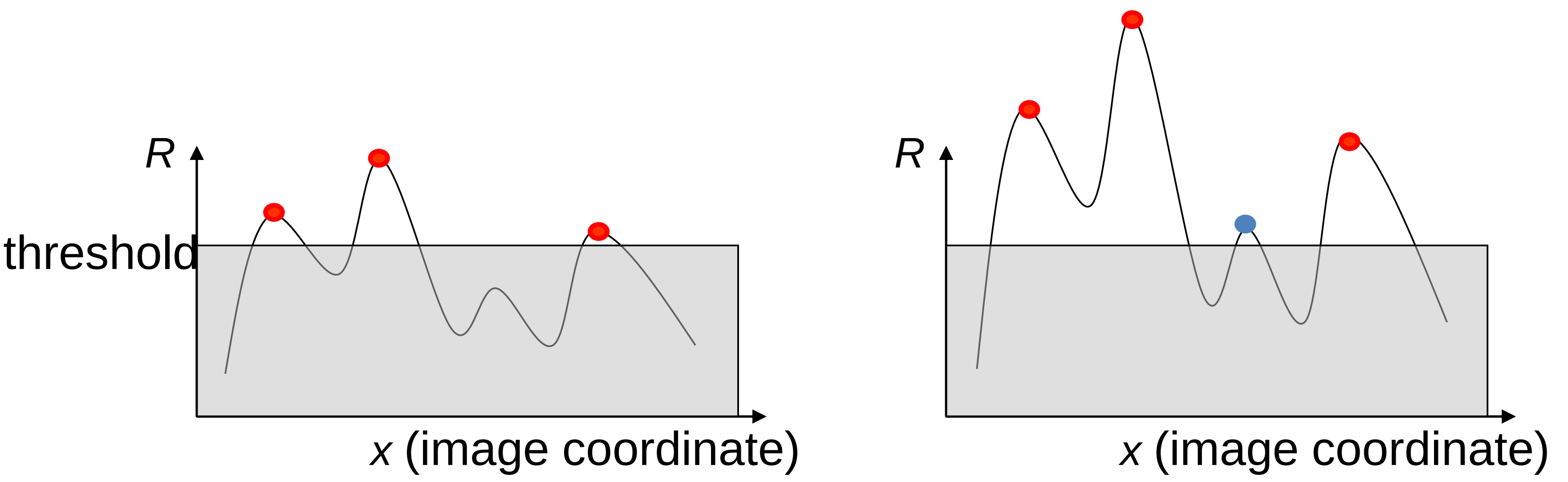
Corner response R **is invariant to image rotation**

Harris Detector: Some Properties

- **Partial invariance to *affine intensity* change**

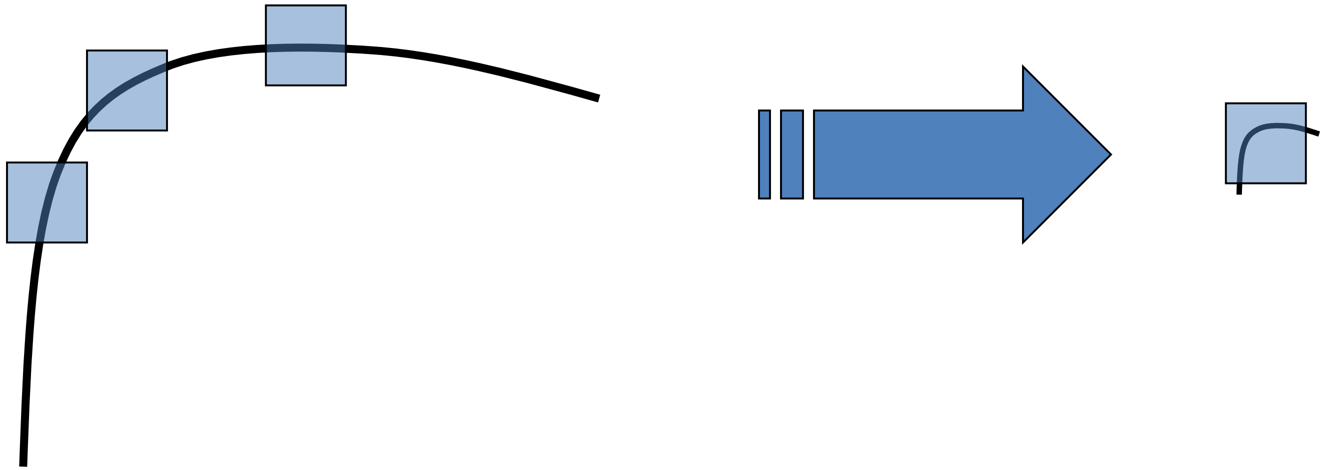
  ✓ Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$

  ✓ Intensity scale: $I \rightarrow a\, I$



Harris Detector: Some Properties

- But: **non-invariant to *image scale*!**



All points will be classified as edges

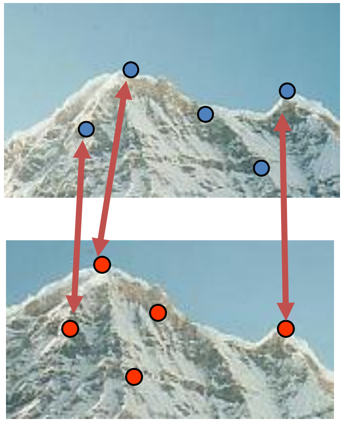
Corner !

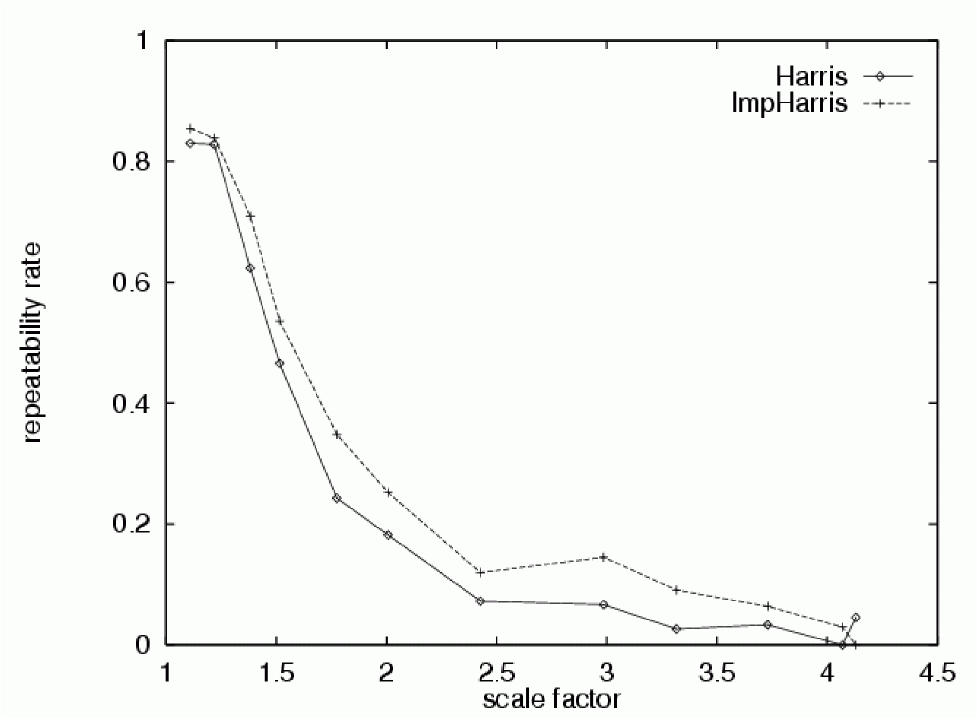
Harris Detector: Some Properties

- Quality of Harris detector for different scale changes

Repeatability rate:

$$\frac{\text{\# correspondences}}{\text{\# possible correspondences}}$$





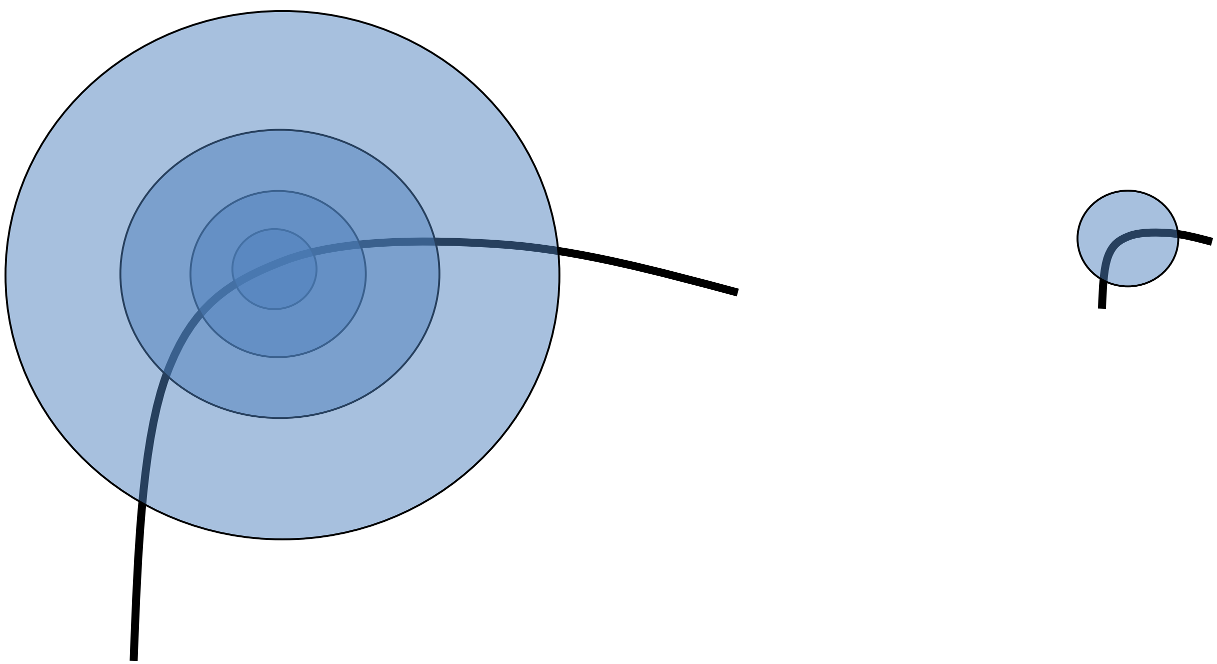
C.Schmid et.al. "Evaluation of Interest Point Detectors". IJCV 2000

Additional Invariances for Representation

- Scale Invariance
- Affine Invariance

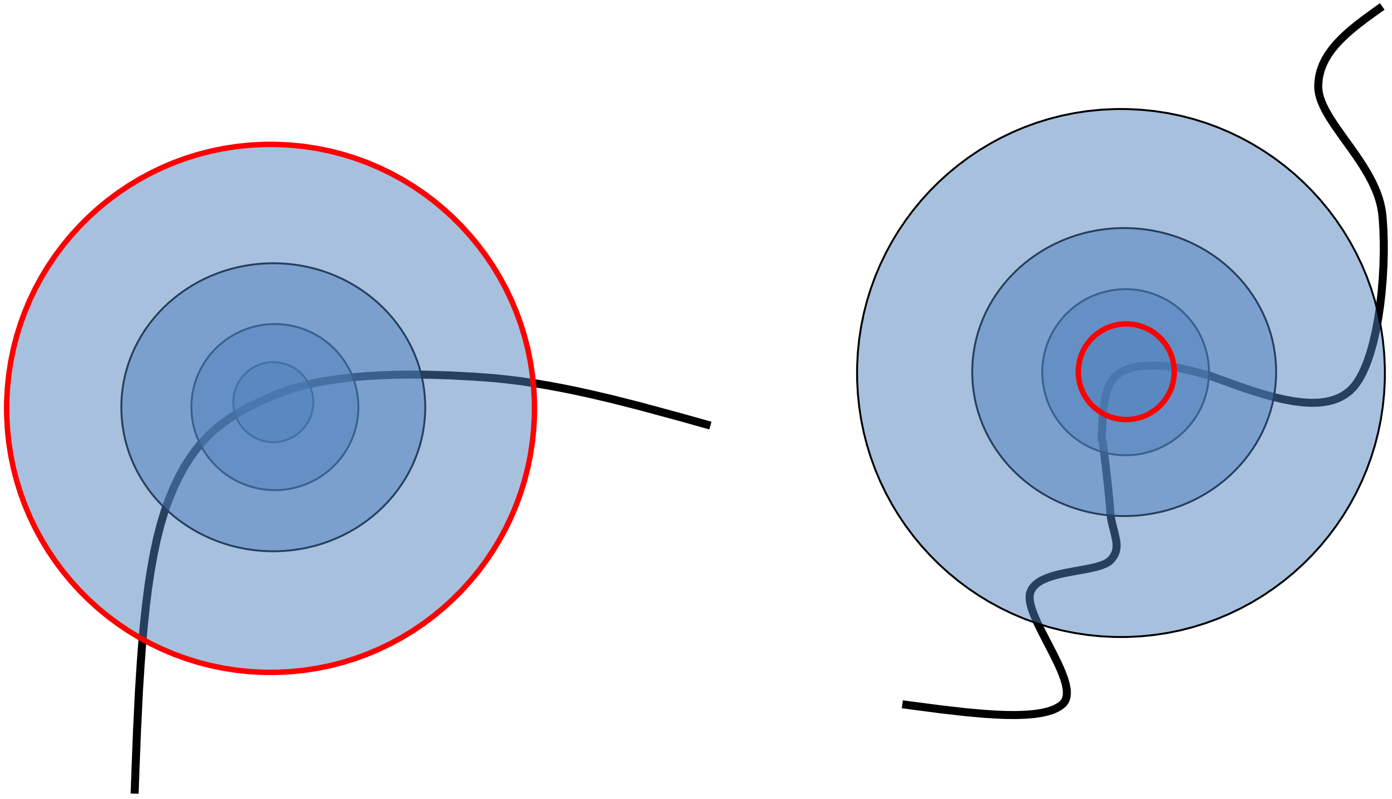
Scale Invariant Detection

- Consider regions (e.g. circles) of **different sizes** around a point
- Regions of corresponding sizes will look the same in both images



Scale Invariant Detection

- The problem: how do we choose corresponding circles ***independently*** in each image?

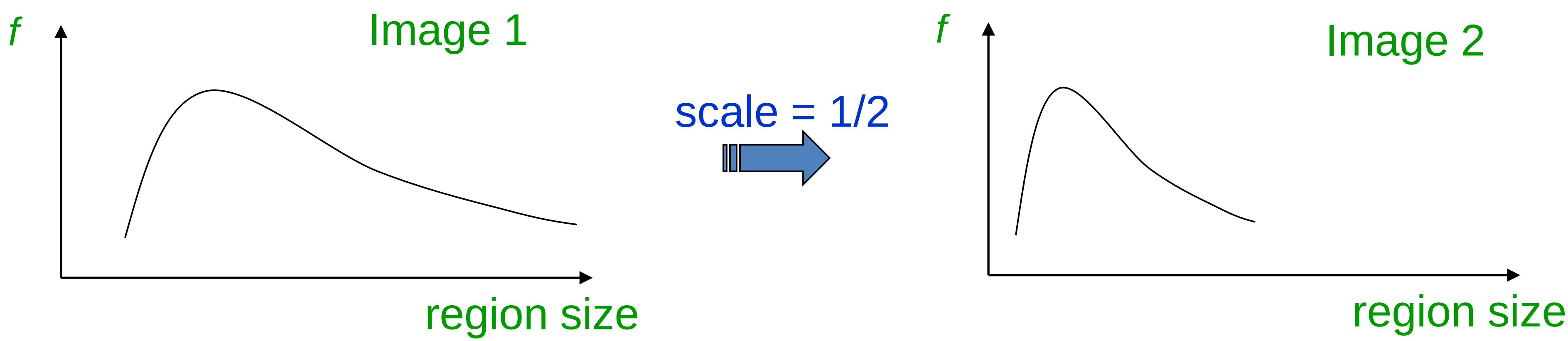


Scale Invariant Detection

- Solution:
  - Design a function on the region (circle), which is "scale invariant" (the same for corresponding regions, even if they are at different scales)

    Example: average intensity. For corresponding regions (even of different sizes) it will be the same.

  - For a point in one image, we can consider it as a function of region size (circle radius)

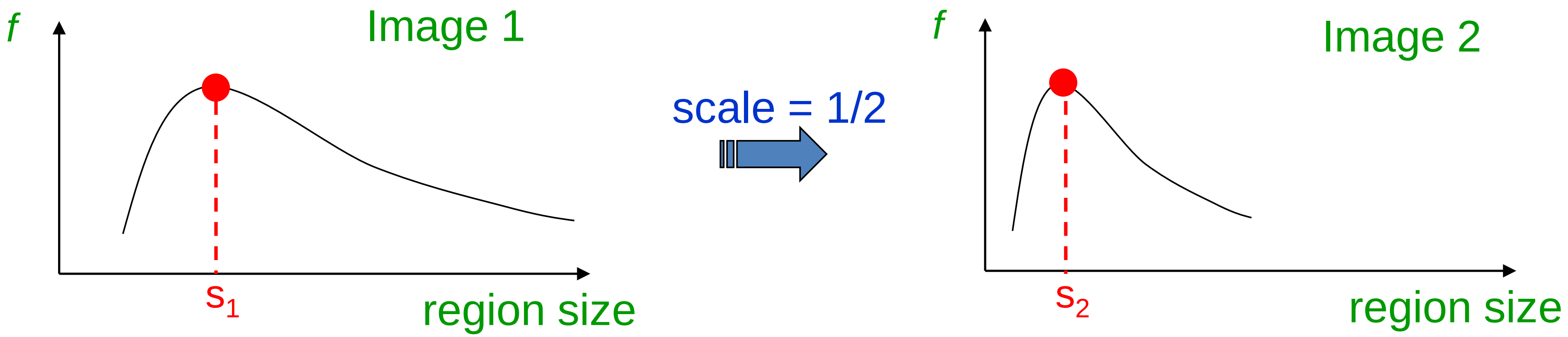


Scale Invariant Detection

- Common approach:

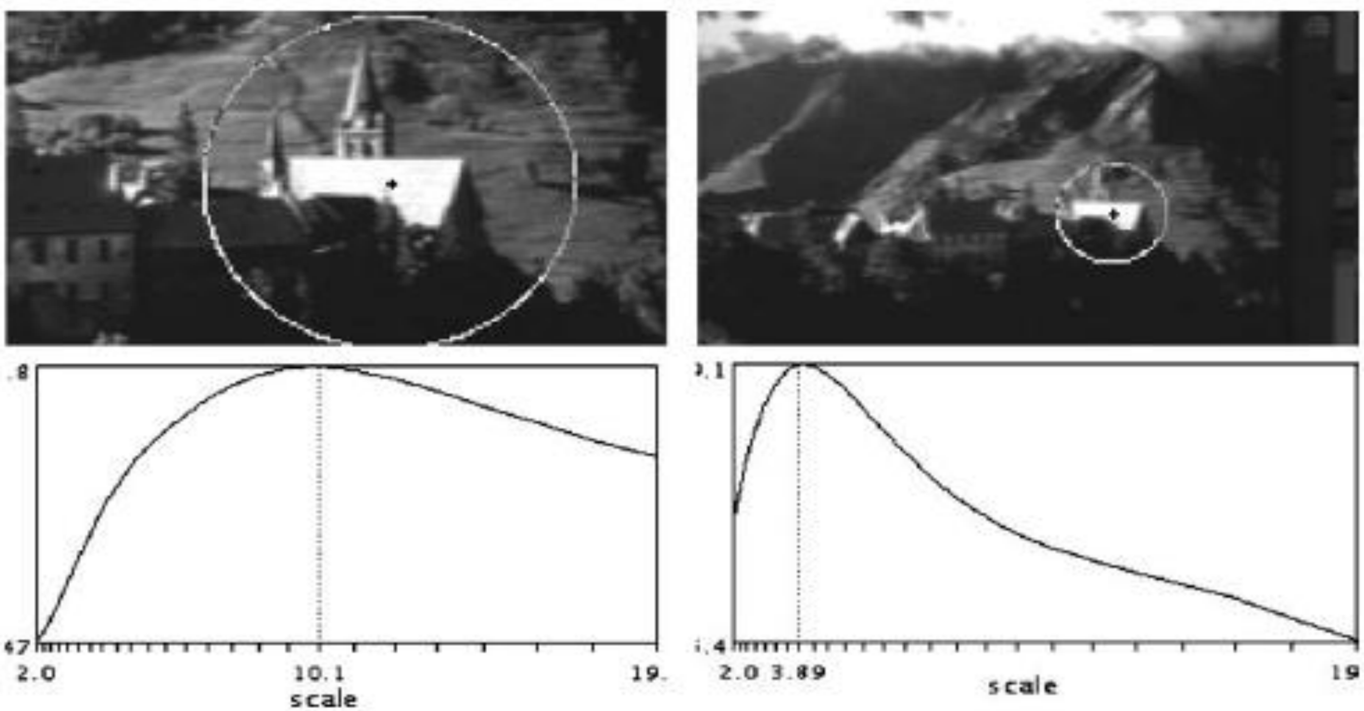
Take a local maximum of this function

Observation: region size, for which the maximum is achieved, should be *invariant* to image scale.



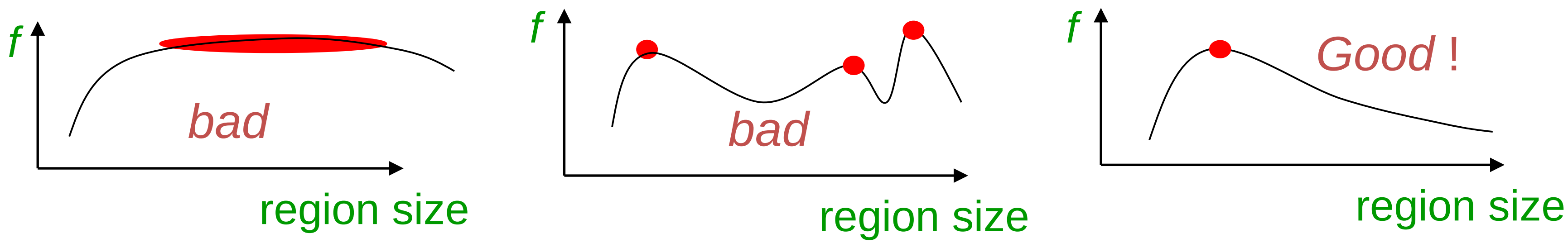
Characteristic Scale

Ratio of scales corresponds to a scale factor between two images



Scale Invariant Detection

- A "good" function for scale detection:
  has one stable sharp peak



- For usual images: a good function would be a one which responds to contrast (sharp local intensity change)

Scale Invariant Detection

- **Functions for determining scale**

$$f = \text{Kernel} * \text{Image}$$

Kernels:

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

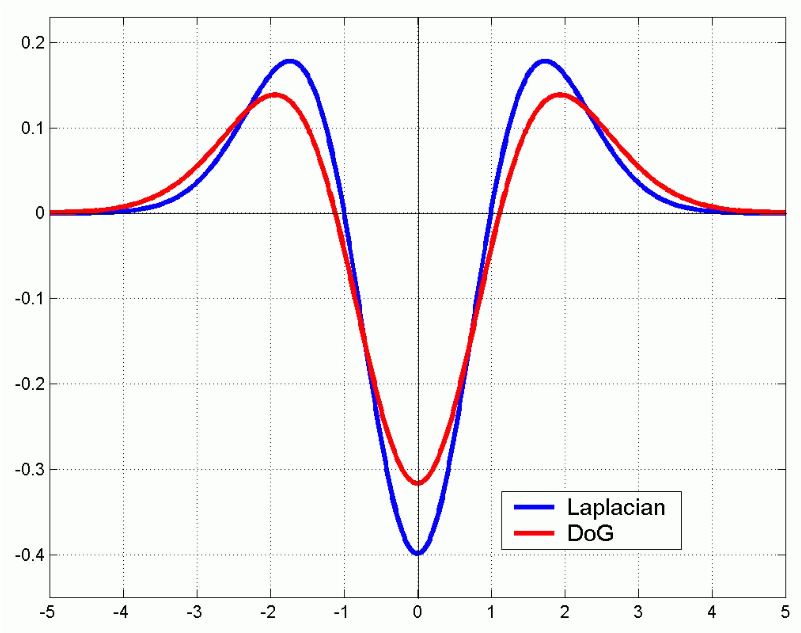
(Laplacian)

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

(Difference of Gaussians)

where Gaussian

$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



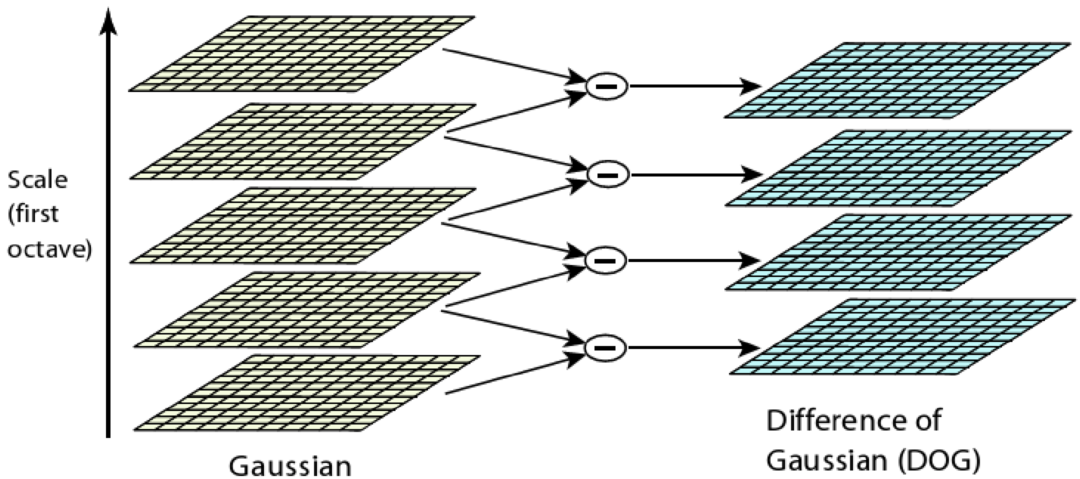
L or DoG kernel is a matching filter. It finds blob-like structure. It turns out to be also successful in getting characteristic scale of other structures, such as corner regions.

Difference-of-Gaussians

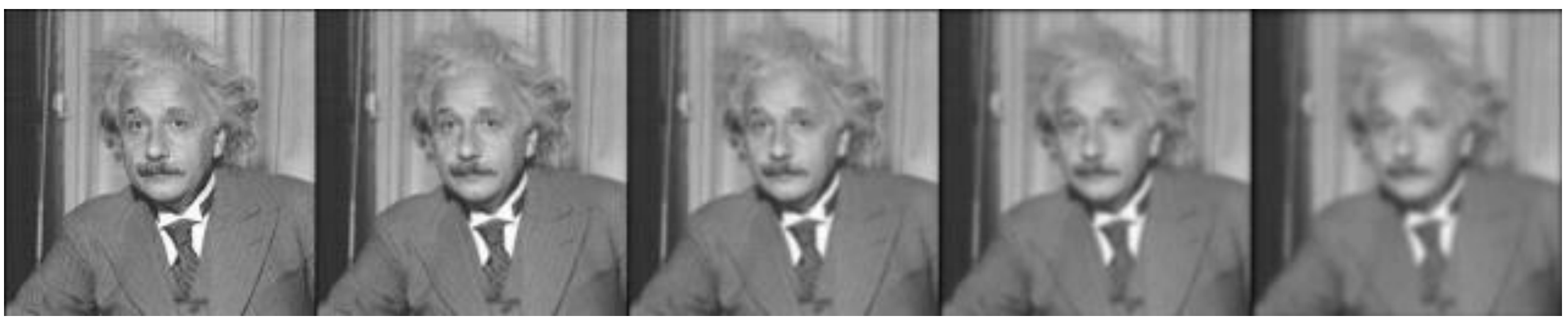
$$G\left(k^{2}\sigma\right)*I$$

$$G(k\sigma)*I$$

$$G(\sigma)*I$$

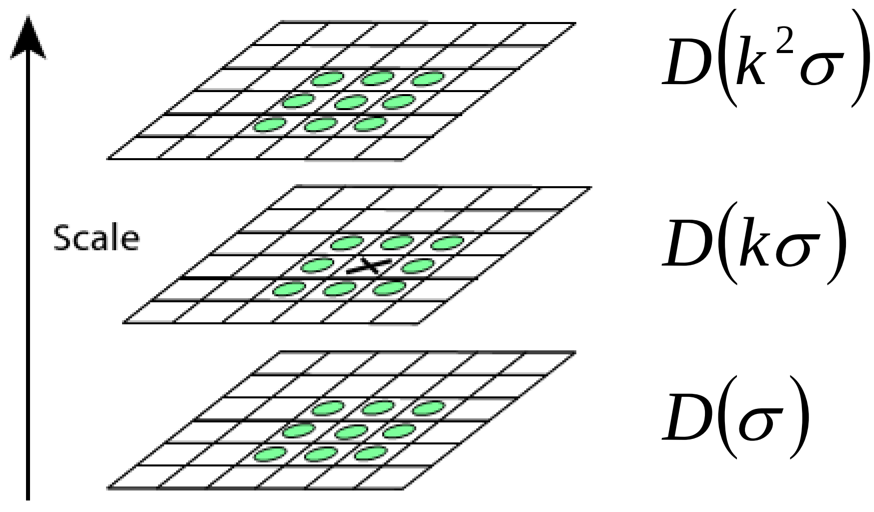


$$D(\sigma) \equiv \left(G(k\sigma) - G(\sigma)\right)*I$$



Scale-Space Extrema

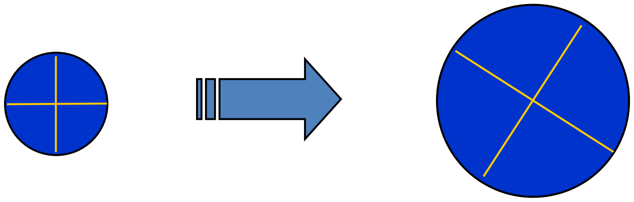
- Choose all extrema within 3x3x3 neighborhood.



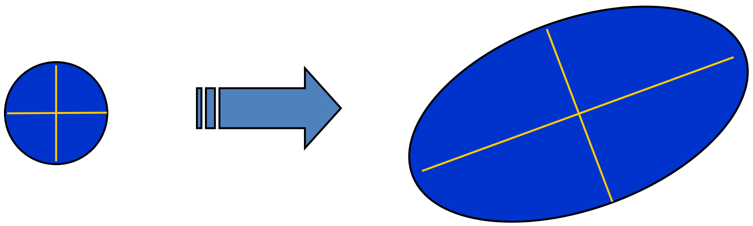
X is selected if it is larger or smaller than all 26 neighbors

Affine Invariant Detection

- Above we considered:
  Similarity transform (rotation + uniform scale)

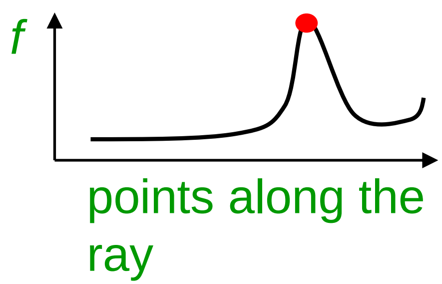


- Now we go on to:
  Affine transform (rotation + non-uniform scale)

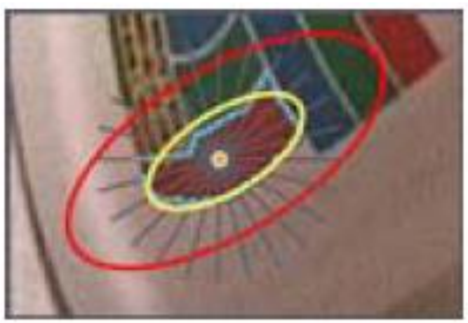


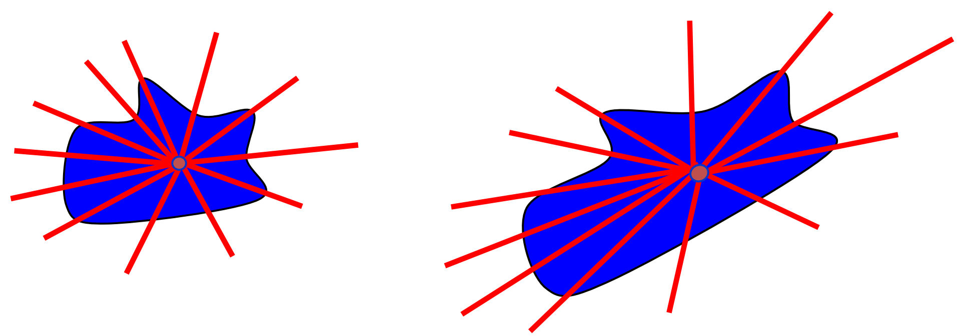
Affine Invariant Detection

- Take a local intensity extremum as initial point
- Go along every ray starting from this point and stop when **extremum of function** f is reached



$$f(t) = \frac{\left| I(t) - I_0 \right|}{\frac{1}{t} \int_{o}^{t} \left| I(t) - I_0 \right| dt}$$

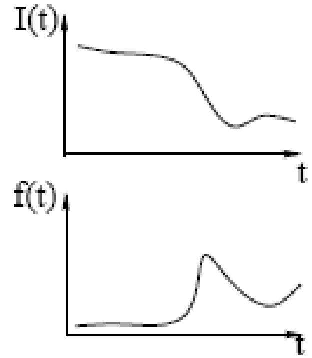




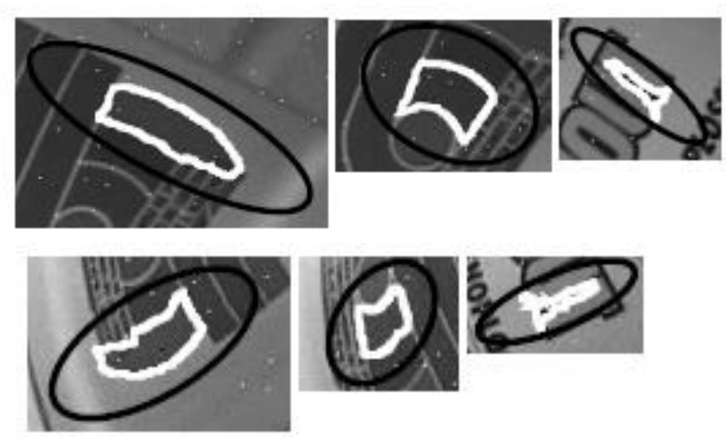
T.Tuytelaars, L.V.Gool. “Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions”. BMVC 2000.

Affine Invariant Detection

- Such extrema occur at positions where intensity suddenly changes compared to the intensity changes up to that point.
- In theory, leaving out the denominator would still give invariant positions. In practice, the local extrema would be shallow, and might result in inaccurate positions.



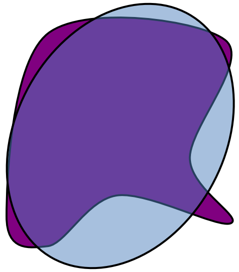
$$f(t) = \frac{\left| I(t) - I_0 \right|}{\frac{1}{t}\int_o^t \left| I(t) - I_0 \right| dt}$$



T.Tuytelaars, L.V.Gool. “Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions”. BMVC 2000.

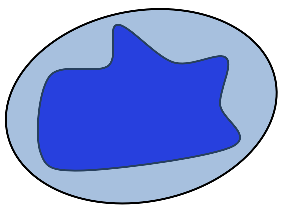
Affine Invariant Detection

- The regions found may not exactly correspond, so we approximate them with ellipses
- Find the ellipse that best fits the region

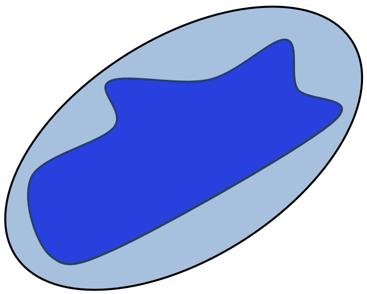


Affine Invariant Detection

- Covariance matrix of region points defines an ellipse:



$$q = Ap$$



$$p^T \Sigma_1^{-1} p = 1$$

$$\Sigma_1 = \left\langle pp^T \right\rangle_{\text{region 1}}$$

$$q^T \Sigma_2^{-1} q = 1$$

$$\Sigma_2 = \left\langle qq^T \right\rangle_{\text{region 2}}$$

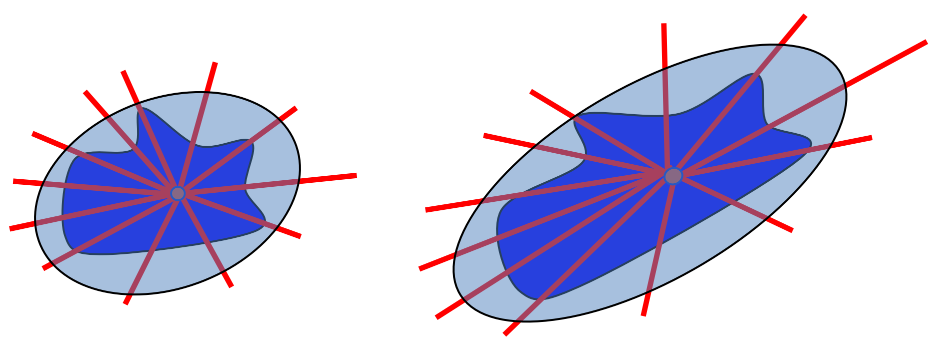
($p = [x, y]^T$ is relative to the center of mass)

$$\Sigma_2 = A \Sigma_1 A^T$$

Ellipses, computed for corresponding regions, also correspond!

Affine Invariant Detection

- **Algorithm summary** (**detection of affine invariant region**):
  - Start from a *local intensity extremum* point
  - Go in ***every direction*** until the point of extremum of some function f
  - Curve connecting the points is the region boundary
  - Compute the covariance matrix
  - Replace the region with *ellipse*



T.Tuytelaars, L.V.Gool. “Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions”. BMVC 2000.

Affine Invariant Detection

- Maximally Stable Extremal Regions
  - *Threshold* image intensities: $I > I_0$
  - Extract *connected components* ("Extremal Regions")
  - Find "Maximally Stable" regions
  - Approximate a region with an *ellipse*

J.Matas et.al. "Distinguished Regions for Wide-baseline Stereo". Research Report of CMP, 2001.

Affine Invariant Detection : Summary

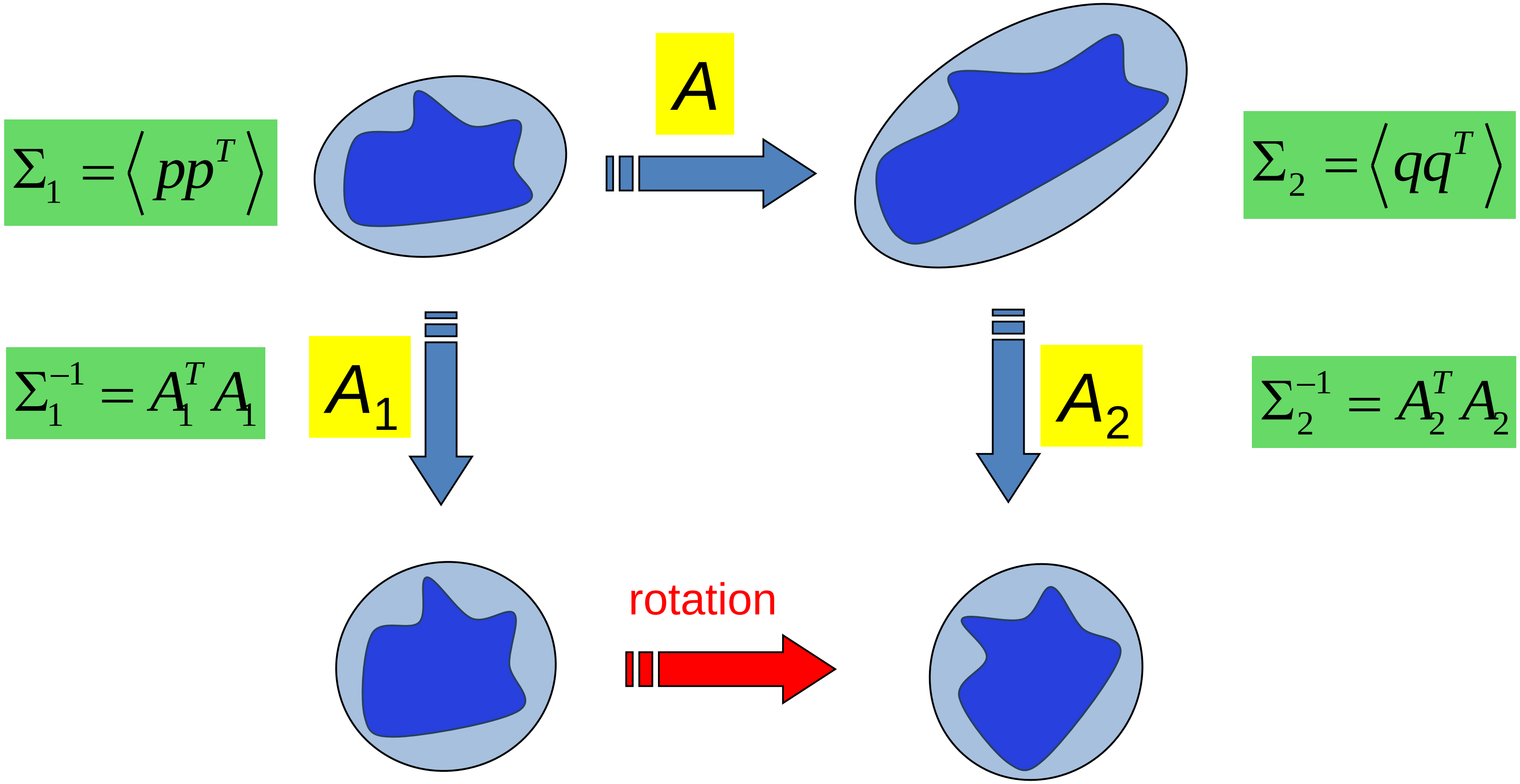
- **Under affine transformation**, we do not know in advance shapes of the corresponding regions
- Ellipse given by geometric covariance matrix of a region robustly approximates this region
- For corresponding regions ellipses also correspond

Methods:

1. Search for extremum along rays [Tuytelaars, Van Gool]:
2. Maximally Stable Extremal Regions [Matas et.al.]

Affine Invariant Descriptors

- Find affine normalized frame



- Compute rotational invariant descriptor in this normalized frame

J.Matas et.al. “Rotational Invariants for Wide-baseline Stereo”. Research Report of CMP, 2003

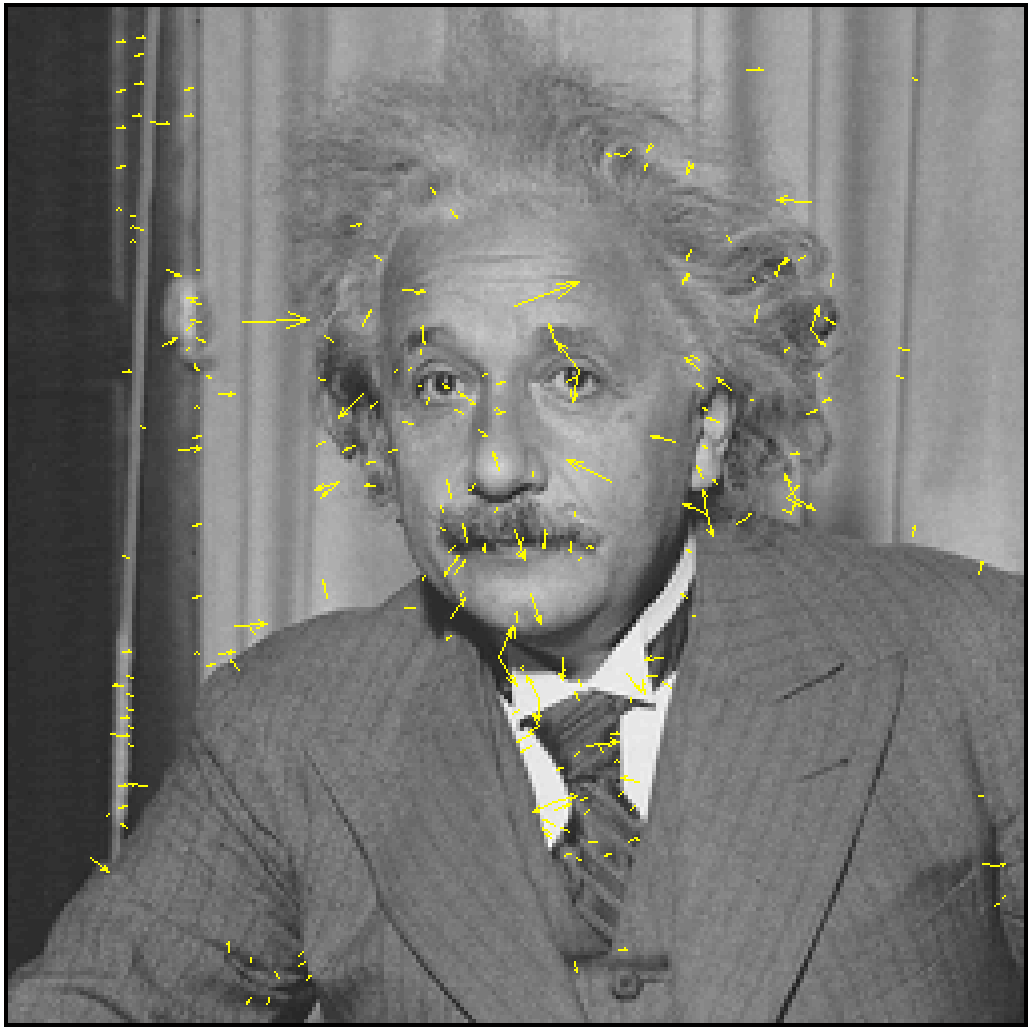
Affine covariant regions

Question

Why invariant properties are important/attractive for image representation?

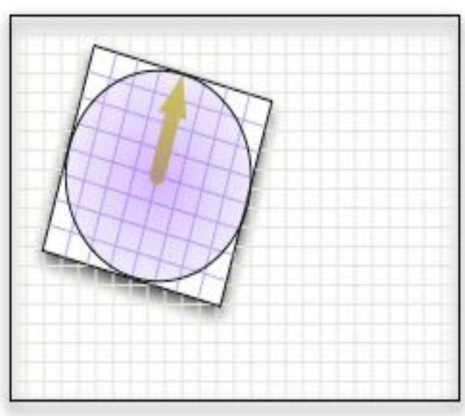
Keypoints and SIFT
(Scale Invariant Feature Transform)

Keypoints & SIFT Descriptors



Keypoint & SIFT Descriptor

- 16x16 Gradient window is taken. Partitioned into 4x4 subwindows.
- Histogram of 4x4 samples in 8 directions
- Gaussian weighting around center(σ is 0.5 times that of the scale of a keypoint)
- 4x4x8 = 128



Keypoint Detection
& Orientation Determination
& Neighborhood Pattern

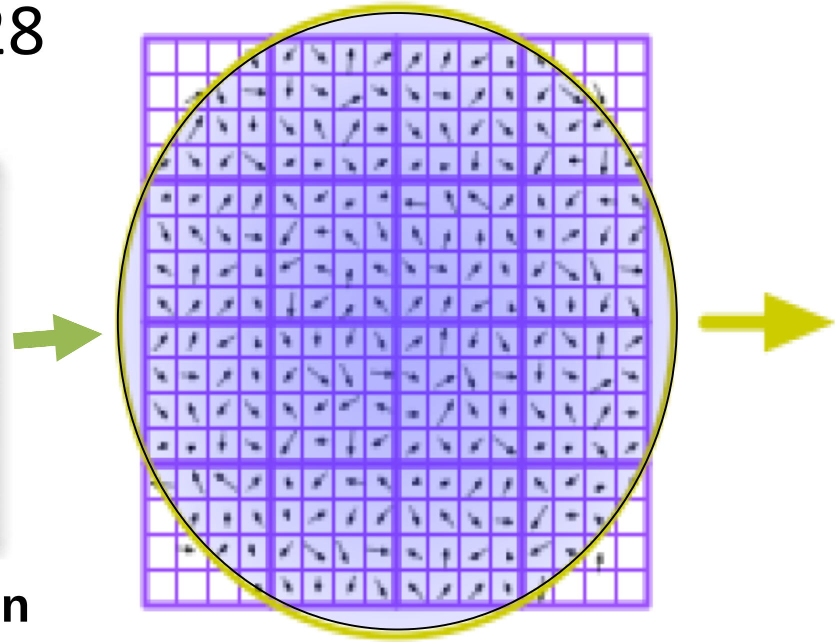
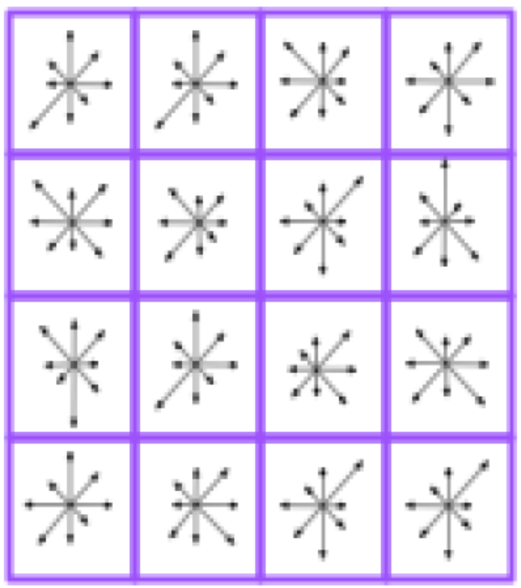
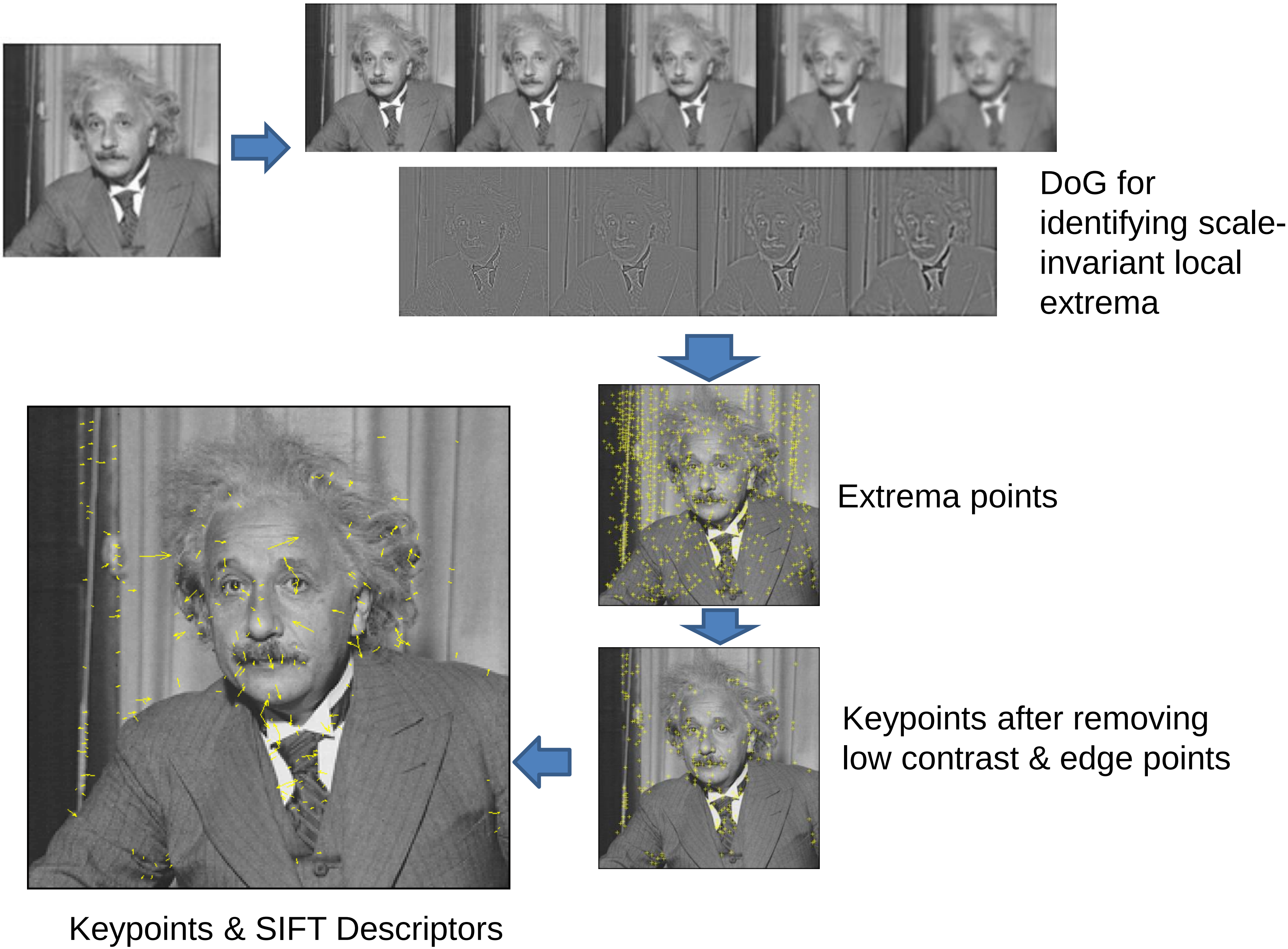


Image gradients

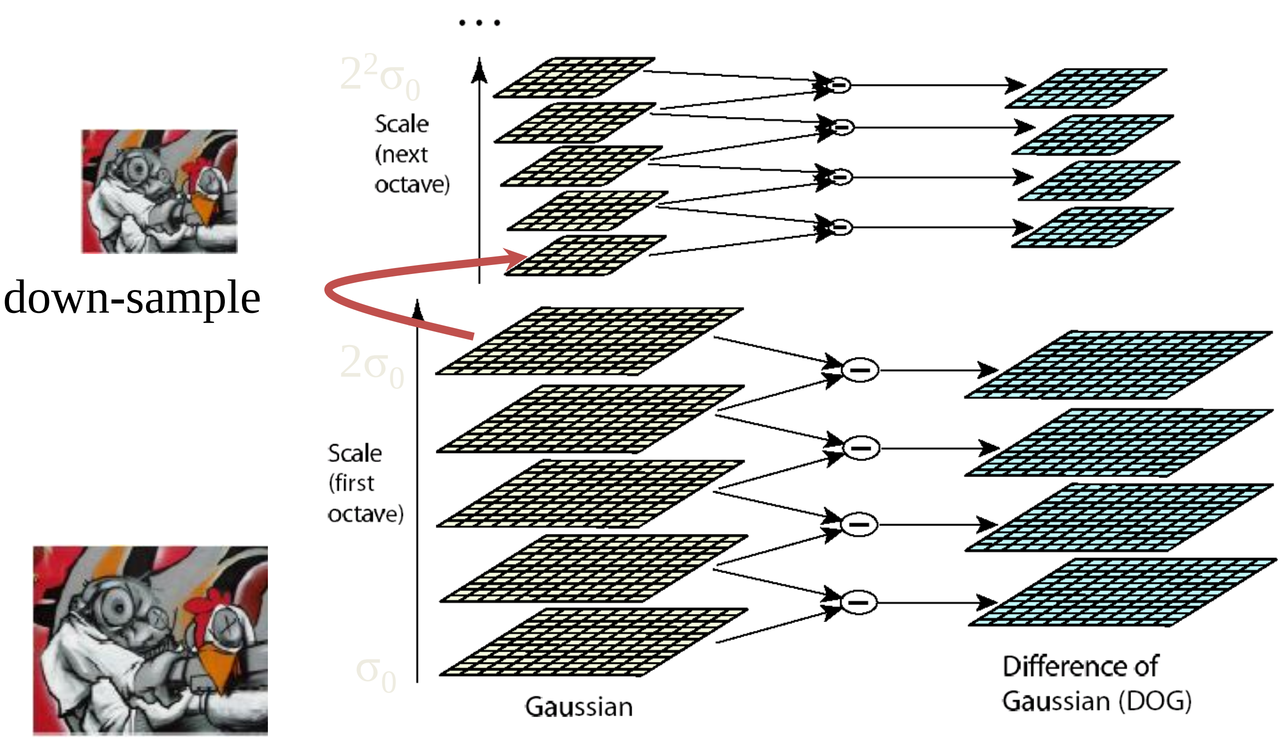


Keypoint descriptor


DoG for identifying scale-invariant local extrema
Extrema points
Keypoints after removing low contrast & edge points
Keypoints & SIFT Descriptors

Scale-space Extrema Detection

- DoG images are grouped by octaves (i.e., <u>doubling</u> of σ_0)
- Fixed number of levels per octave

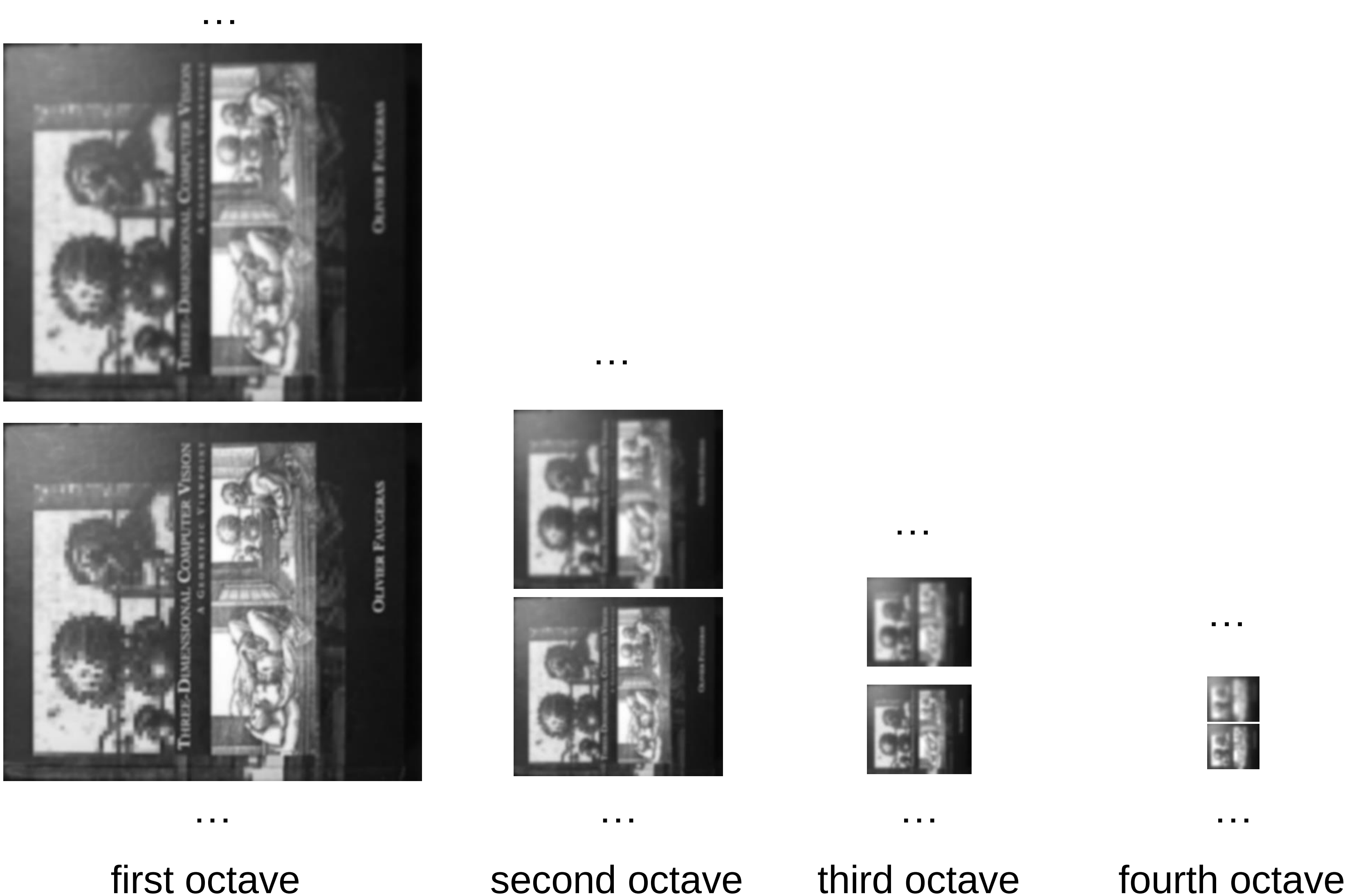


$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

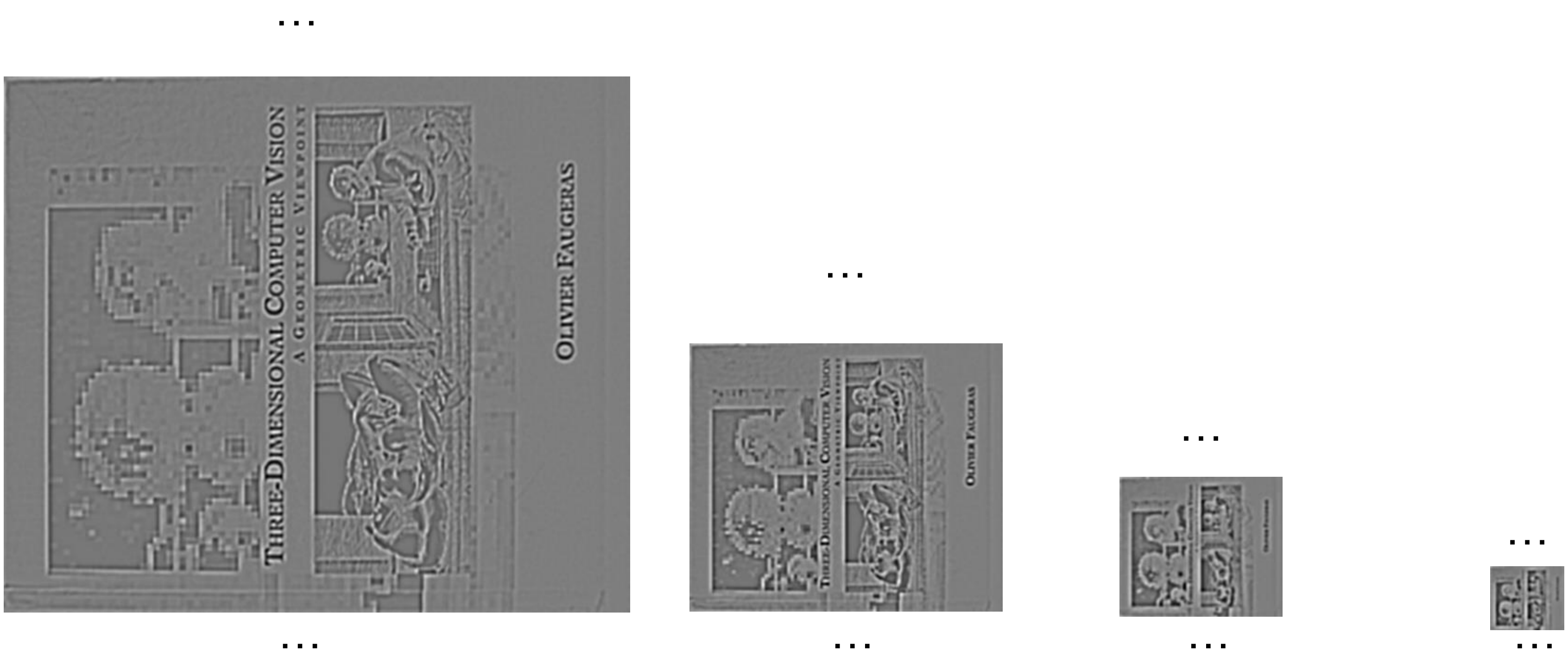
where

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

Scale space images



Difference-of-Gaussian images



first octave second octave third octave fourth octave

Scale-space extrema detection

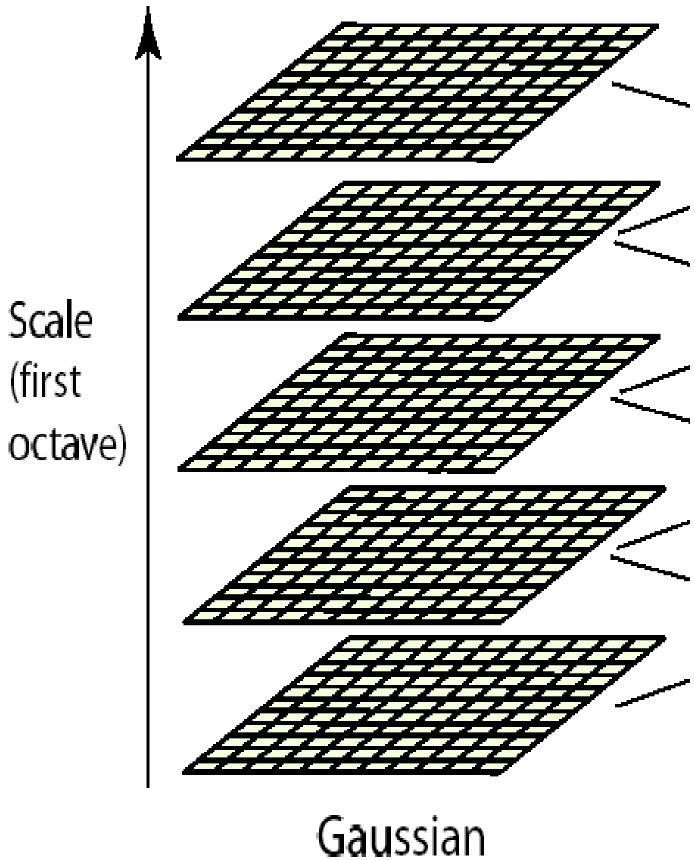
- Find the points, **whose surrounding patches (with some scale) are distinctive**
- An approximation to the scale-normalized Laplacian of Gaussian

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}$$

Scale-space Extrema Detection



- Images within each octave are separated by a constant factor **k**

- If each octave is divided in **s** intervals:

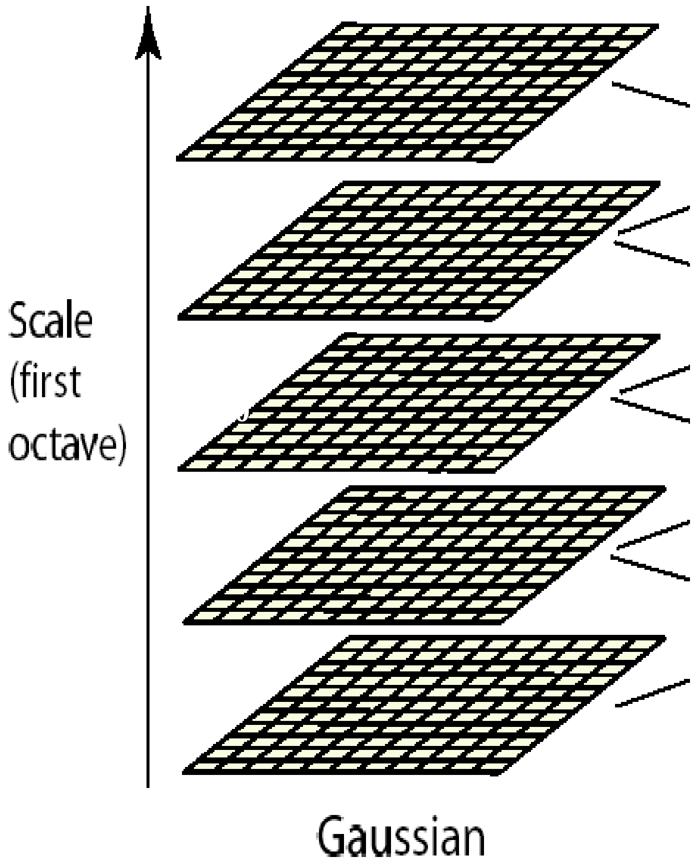
$$\mathbf{k^s=2 \quad or \quad k=2^{1/s}}$$

Choosing SIFT parameters

- Parameters (i.e., scales per octave, σ_0 etc.) can be chosen experimentally based on keypoint (i) repeatability, (ii) localization, and (iii) matching accuracy.

- In Lowe's paper:
  - Keypoints extracted from 32 real images (outdoor, faces, aerial etc.)
  - Images were subjected to a wide range of transformations (i.e., rotation, scaling, shear, change in brightness, noise).

Scale-space Extrema Detection



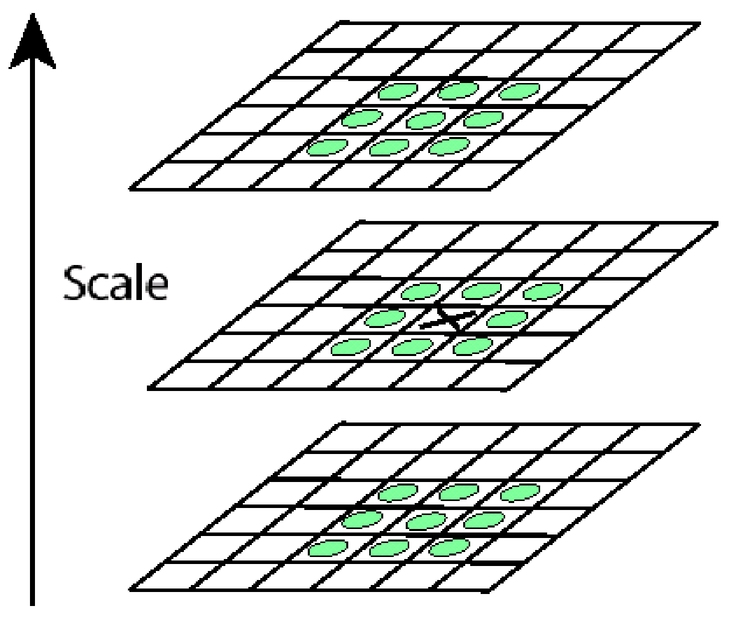
• Pre-smoothing discards high frequencies.

• **Double** the size of the input image (i.e., using linear interpolation) prior to building the first level of the DoG pyramid.

• Increases the number of stable keypoints by a factor of 4.

Scale-space Extrema Detection (cont'd)

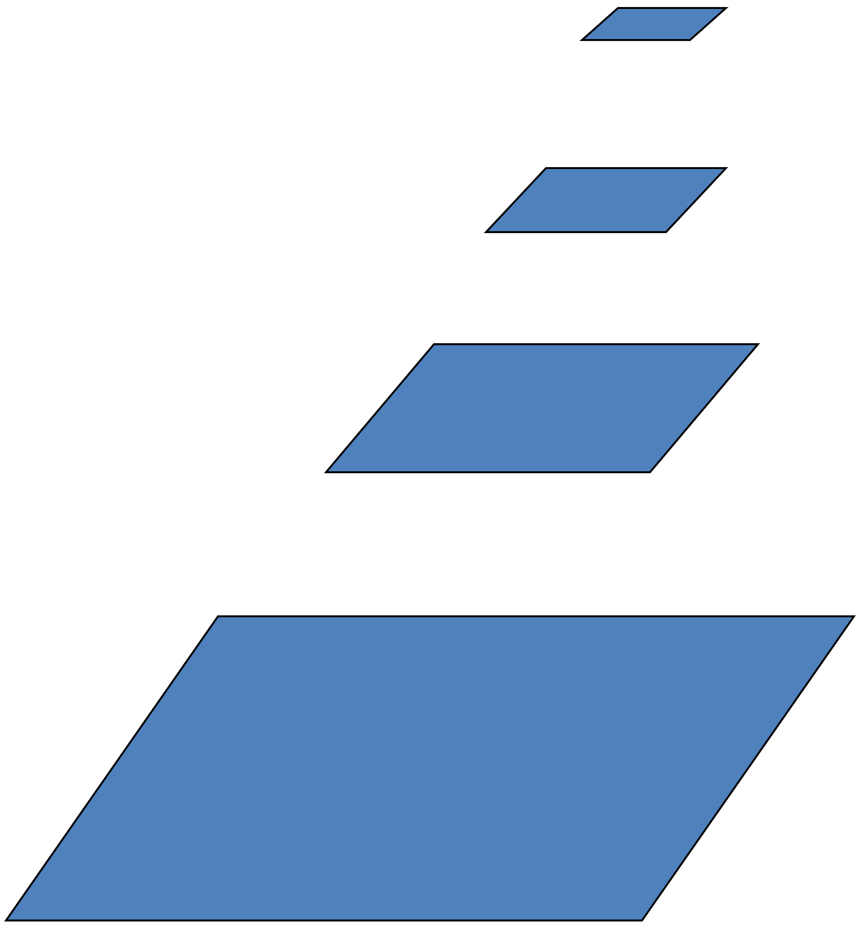
- Extract **local extrema** (i.e., minima or maxima) in DoG pyramid.
  - -Compare each point to its **8 neighbors** at the same level, **9 neighbors** in the level above, and **9 neighbors** in the level below (i.e., 26 total).



Scale-space extrema detection

- Goal: Identify locations and scales that can be repeatably assigned under different views of the same scene or object.
- Method: search for stable features across multiple scales using a continuous function of scale.
- Prior work has shown that under a variety of assumptions, the best function is a Gaussian function.
- The scale space of an image is a function $L(x,y,\sigma)$ that is produced from the convolution of a Gaussian kernel (at different scales) with the input image.

Aside: Image Pyramids



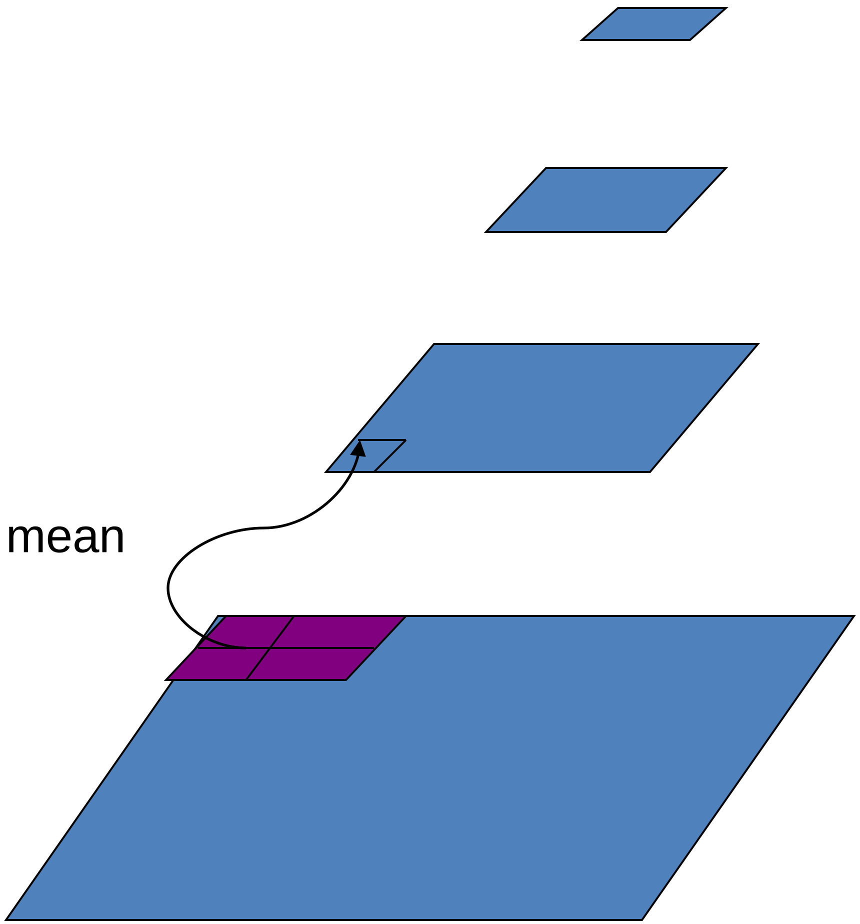
And so on.

3rd level is derived from the 2nd level according to the same funtion

2nd level is derived from the original image according to some function

Bottom level is the original image.

Aside: Mean Pyramid



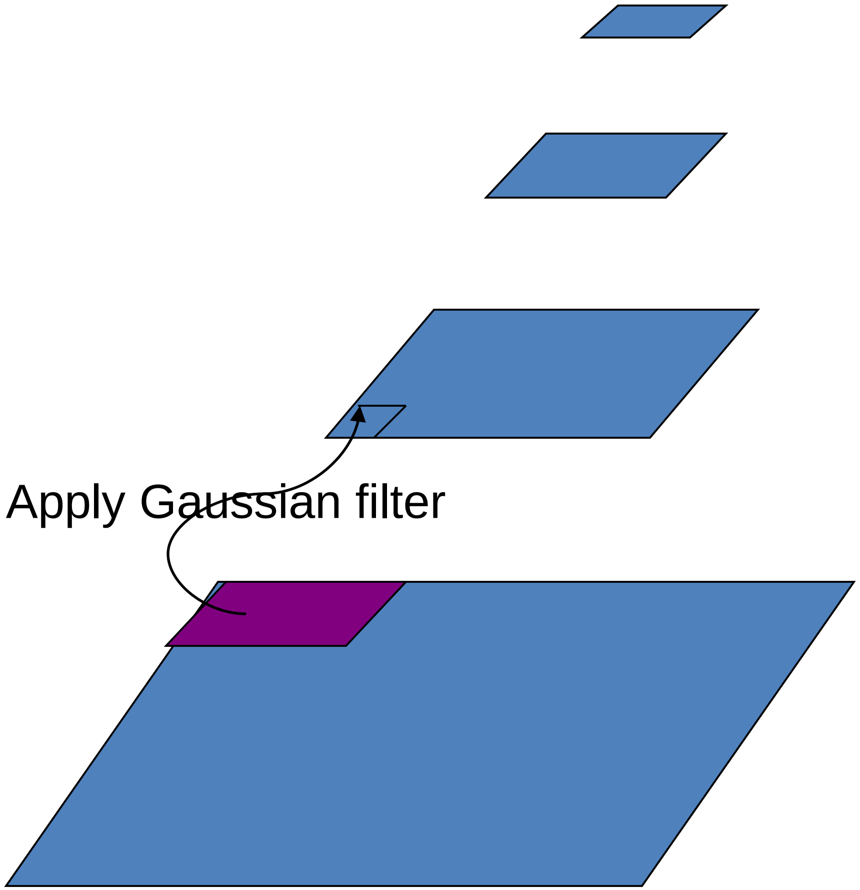
And so on.

At 3rd level, each pixel is the mean of 4 pixels in the 2nd level.

At 2nd level, each pixel is the mean of 4 pixels in the original image.

Bottom level is the original image.

Aside: Gaussian Pyramid
At each level, image is smoothed and reduced in size.



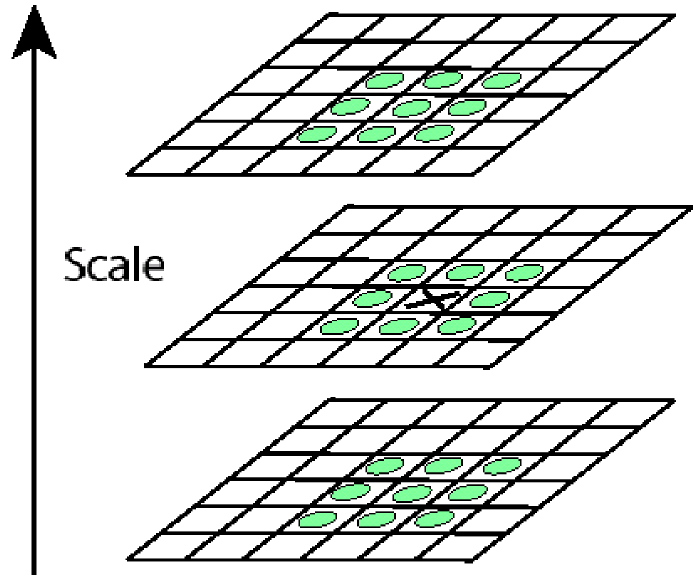
And so on.

At 2^{nd} level, each pixel is the result of applying a Gaussian mask to the first level and then subsampling to reduce the size.

Bottom level is the original image.

1. Keypoint Localization (Detection)

- Determine the location and scale of keypoints to **sub-pixel** and **sub-scale** accuracy by fitting a 3D quadratic polynomial:



$$X_i = (x_i, y_i, \sigma_i)$$ keypoint location

$$\Delta X = (x - x_i, y - y_i, \sigma - \sigma_i)$$ offset

$$X_i \leftarrow X_i + \Delta X$$ sub-pixel, sub-scale Estimated location

Substantial improvement to matching and stability!

1. Keypoint Localization (Detection)

- Use Taylor expansion to locally approximate D(x,y,σ) (i.e., DoG function) and **estimate Δx**:

$$D(\Delta X) = D(X_i) + \frac{\partial D^T(X_i)}{\partial \mathrm{X}} \Delta \mathrm{X} + \frac{1}{2} \Delta \mathrm{X}^T \frac{\partial^2 D(X_i)}{\partial \mathrm{X}^2} \Delta \mathrm{X}$$

- **Find the extrema of D(ΔX)**:

$$\frac{\partial D(X_i)}{\partial X} + \frac{\partial^2 D(X_i)}{\partial X^2} \Delta X = 0$$

1. Keypoint Localization (Detection)

$$\frac{\partial^2 D(X_i)}{\partial X^2}\Delta X = -\frac{\partial D(X_i)}{\partial X} \rightarrow \Delta X = -\frac{\partial^2 D^{-1}(X_i)}{\partial X^2}\frac{\partial D(X_i)}{\partial X}$$

- ΔX can be computed by solving a 3x3 linear system:

$$\begin{bmatrix} \frac{\partial^2 D}{\partial \sigma^2} & \frac{\partial^2 D}{\partial \sigma y} & \frac{\partial^2 D}{\partial \sigma x} \\ \frac{\partial^2 D}{\partial \sigma y} & \frac{\partial^2 D}{\partial y^2} & \frac{\partial^2 D}{\partial yx} \\ \frac{\partial^2 D}{\partial \sigma x} & \frac{\partial^2 D}{\partial yx} & \frac{\partial^2 D}{\partial x^2} \end{bmatrix} \begin{bmatrix} \Delta\sigma \\ \Delta y \\ \Delta x \end{bmatrix} = -\begin{bmatrix} \frac{\partial D}{\partial \sigma} \\ \frac{\partial D}{\partial y} \\ \frac{\partial D}{\partial x} \end{bmatrix}$$

use finite differences:

$$\frac{\partial D}{\partial \sigma} = \frac{D_{k+1}^{i,j} - D_{k-1}^{i,j}}{2}$$

$$\frac{\partial^2 D}{\partial \sigma^2} = \frac{D_{k-1}^{i,j} - 2D_k^{i,j} + D_{k+1}^{i,j}}{1}$$

$$\frac{\partial^2 D}{\partial \sigma y} = \frac{(D_{k+1}^{i+1,j} - D_{k-1}^{i+1,j}) - (D_{k+1}^{i-1,j} - D_{k-1}^{i-1,j})}{4}$$

If $\Delta X > 0.5$ in any dimension, repeat.

1. Keypoint Localization (Detection)

- Reject keypoints having **low contrast**.
  - – i.e., sensitive to noise

If $|D(X_i + \Delta X)| < 0.03$ reject keypoint

– i.e., assumes that image values have been normalized in [0,1]

1. Keypoint Localization (Detection)

- Reject points **lying on edges** (or being close to edges)
- Harris uses the auto-correlation matrix:

$$A_W(x,y) = \sum_{x \in W, y \in W} \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix}$$

$$\mathbf{R(A_W) = det(A_W) - \alpha\ trace^2(A_W)}$$

or $$\mathbf{R(A_W) = \lambda_1 \lambda_2 - \alpha (\lambda_1 + \lambda_2)^2}$$

1. Keypoint Localization (Detection)

- SIFT uses the Hessian matrix (for efficiency).
  - i.e., Hessian encodes principal curvatures

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

α: largest eigenvalue (λ_{max})
β: smallest eigenvalue (λ_{min})
(proportional to principal curvatures)

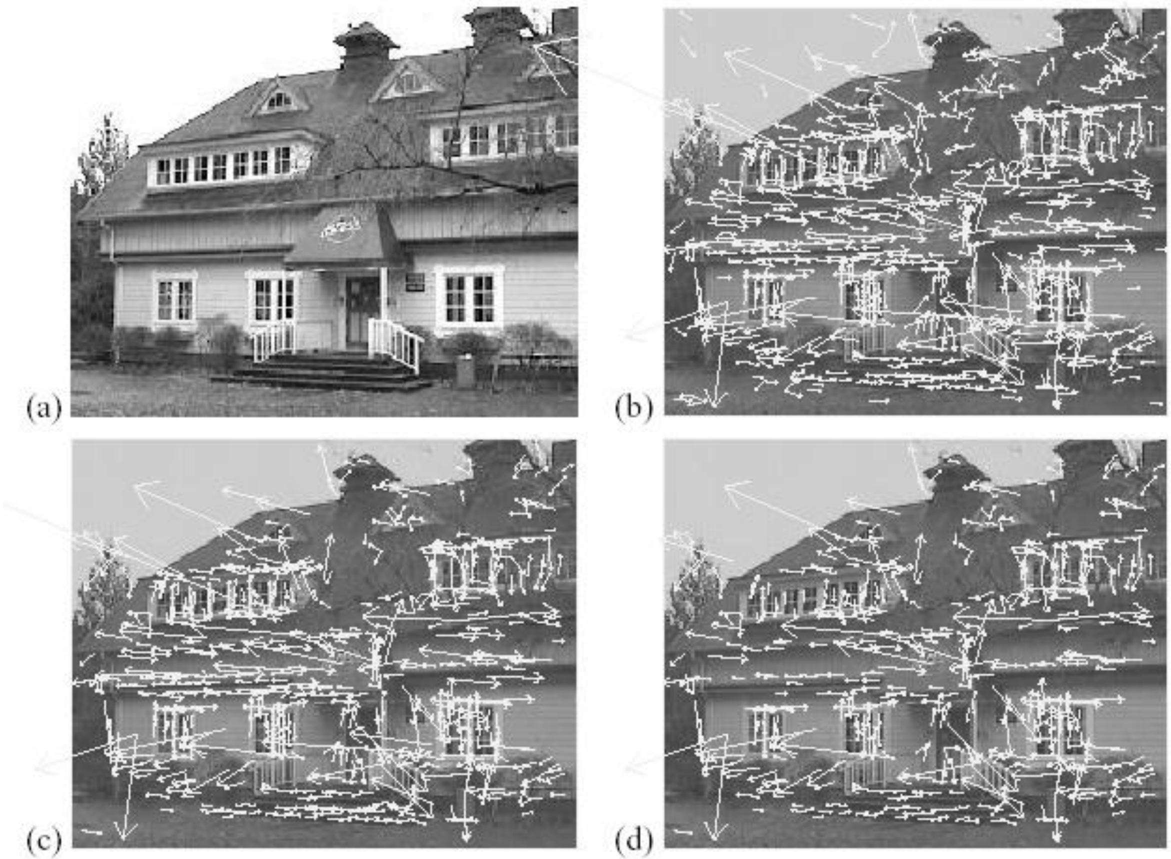
$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$
$$\mathrm{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

$$\rightarrow \frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r},$$

($r = \alpha/\beta$)

Reject keypoint if: $\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}$ (SIFT uses $r = 10$)

1. Keypoint Localization (Detection)


(a) (b) (c) (d)

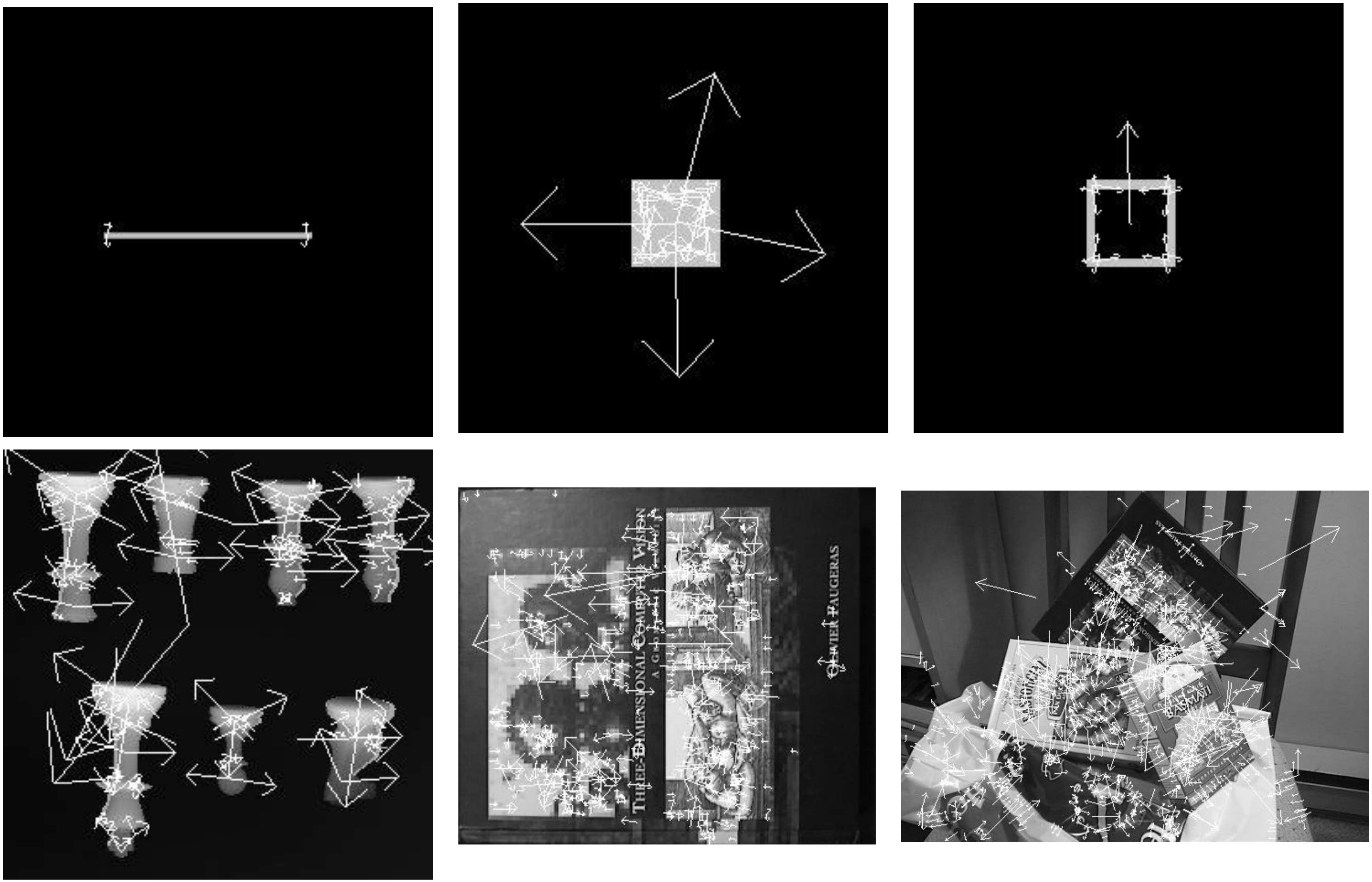
(a) 233x189 image

(b) 832 DoG extrema

(c) 729 left after low contrast threshold

(d) 536 left after testing ratio based on Hessian

Keypoint images



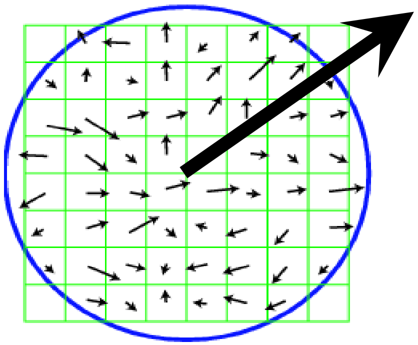
2. Orientation Assignment

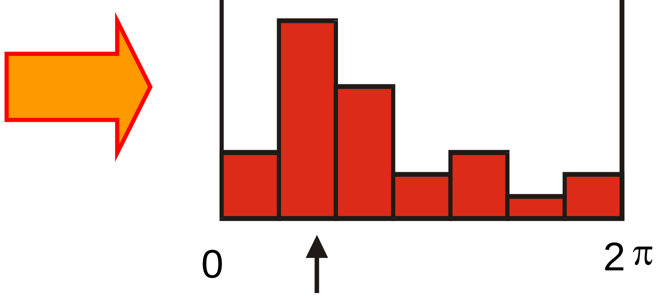
- Create histogram of gradient directions, within a region around the keypoint, at selected scale:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = a\tan 2((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y)))$$



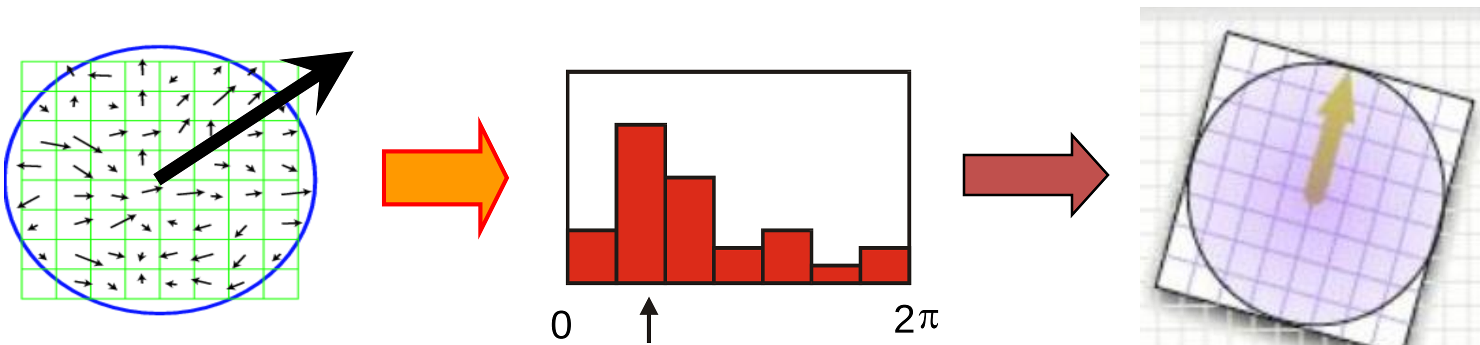


36 bins (i.e., 10^o per bin)

- Histogram entries are **weighted** by (i) gradient magnitude and (ii) a Gaussian function with σ equal to 1.5 times the scale of the keypoint.

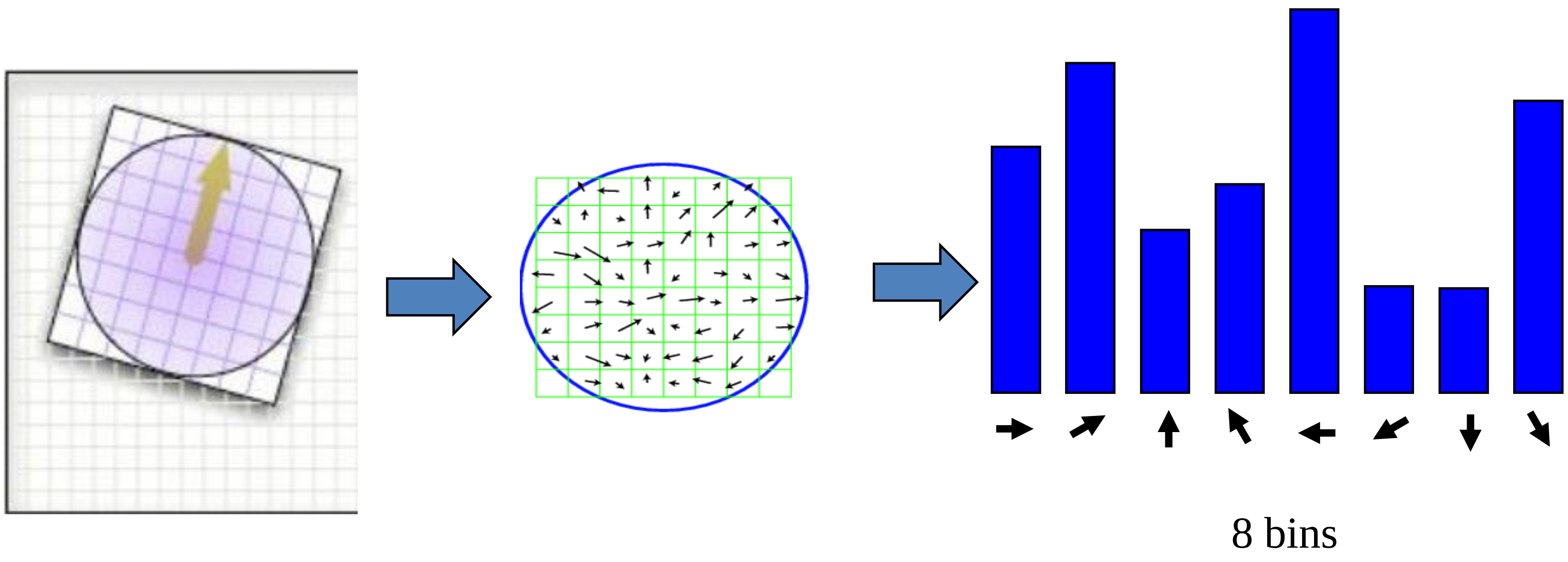
2. Orientation Assignment (cont'd)

- Assign canonical orientation at peak of smoothed histogram (fit parabola to better localize peak).



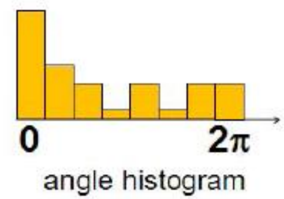
- In case of peaks within 80% of highest peak, multiple orientations assigned to keypoints.
  - About 15% of keypoints has multiple orientations assigned.
  - Significantly improves stability of matching.

3. Keypoint Descriptor

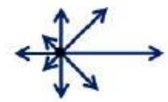


3. Keypoint Descriptor (cont'd)

1. Take a 16 x16 window around detected interest point.
2. Divide into a 4x4 grid of cells.
3. Compute histogram in each cell.



(8 bins)



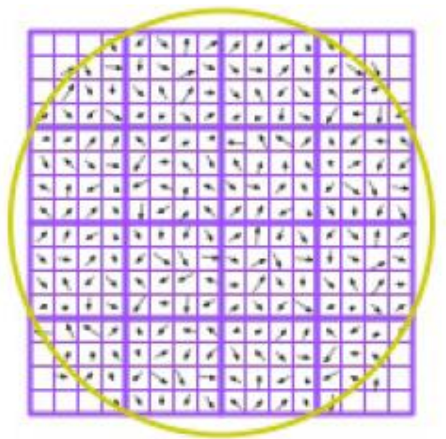
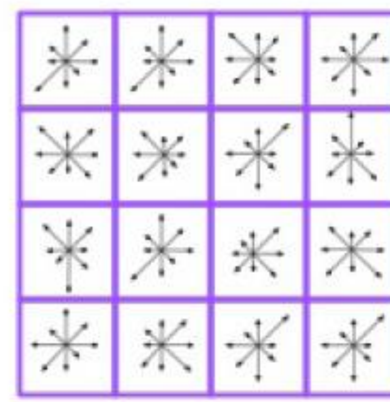


Image gradients

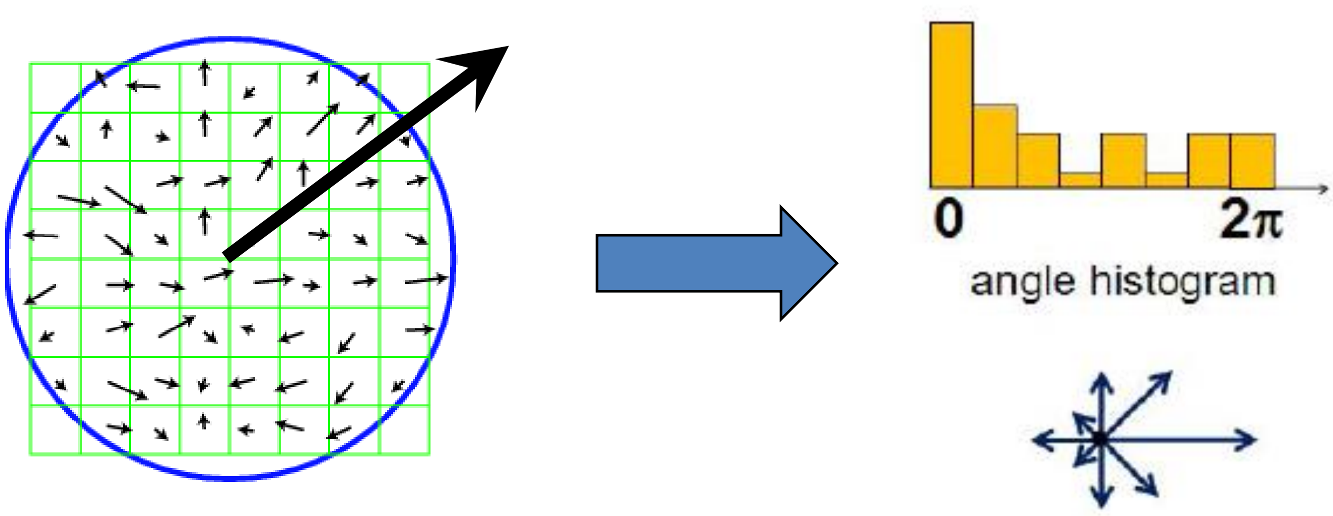


Keypoint descriptor

16 histograms x 8 orientations
= 128 features

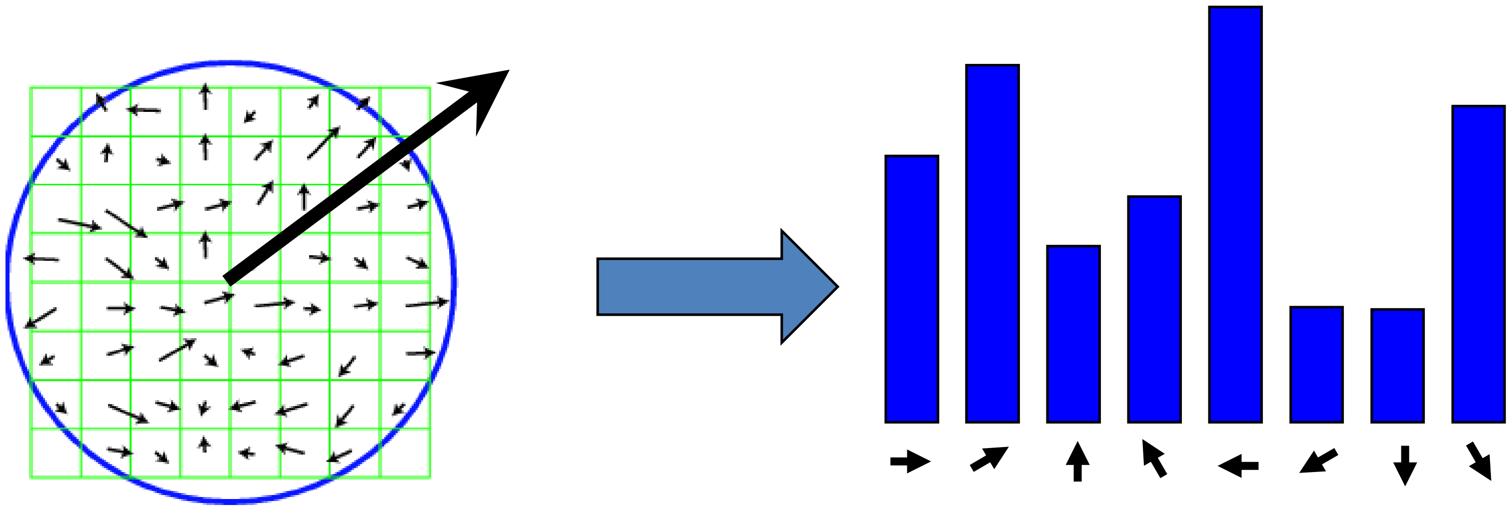
3. Keypoint Descriptor (cont'd)

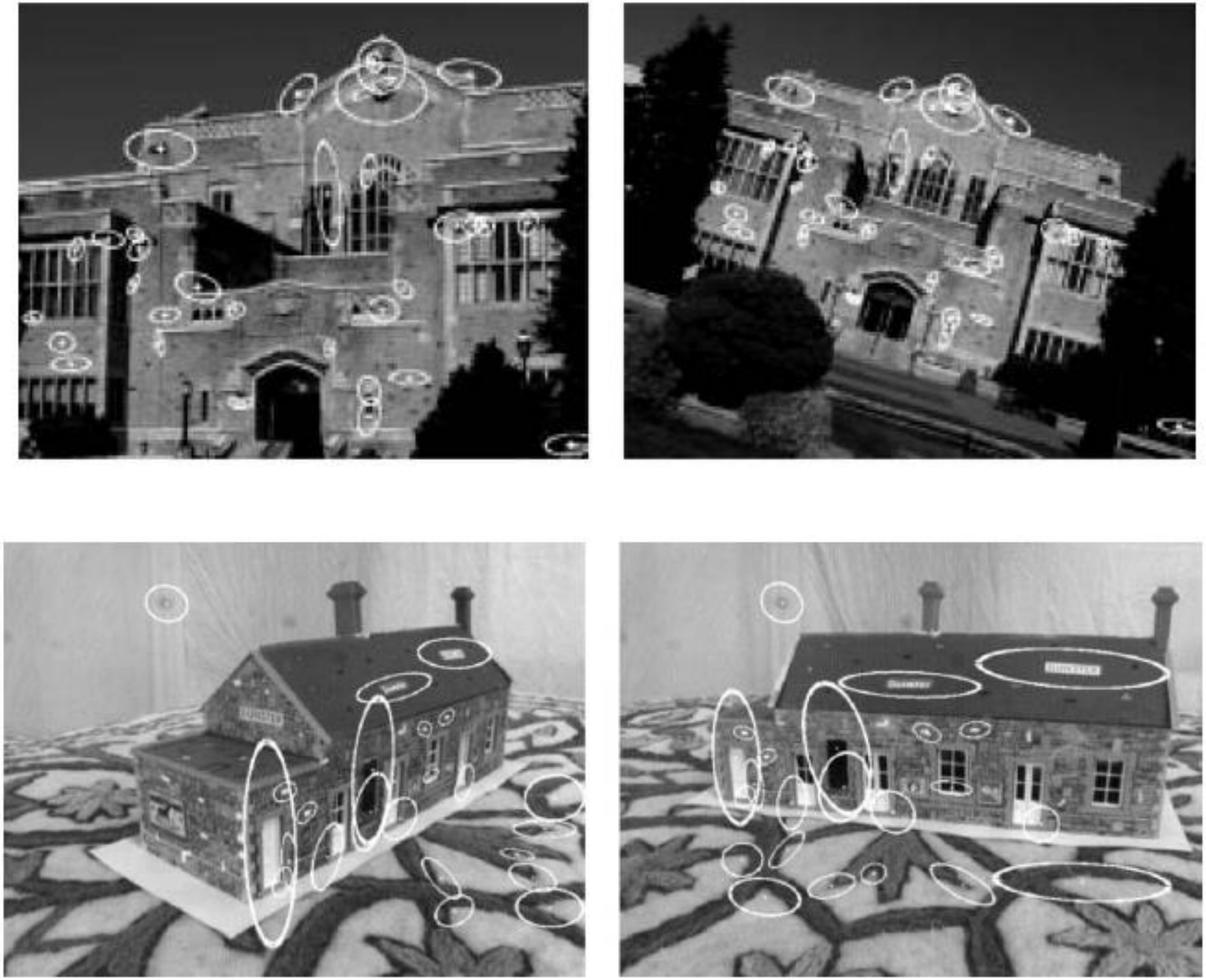
- Each histogram entry **is** **weighted** by (i) gradient magnitude and (ii) a Gaussian function with σ equal to 0.5 times the width of the descriptor window.



3. Keypoint Descriptor (cont'd)

- Partial Voting: distribute histogram entries into adjacent bins (i.e., additional robustness to shifts)
  - Each entry is added to all bins, multiplied by a weight of 1-d, where d is the distance from the bin it belongs.





SIFT Steps - Review

(1) Scale-space extrema detection
- Extract scale and rotation invariant interest points (i.e., keypoints).

(2) Keypoint localization
- Determine location and scale for each interest point.
- Eliminate “weak” keypoints

(3) Orientation assignment
- Assign one or more orientations to each keypoint.

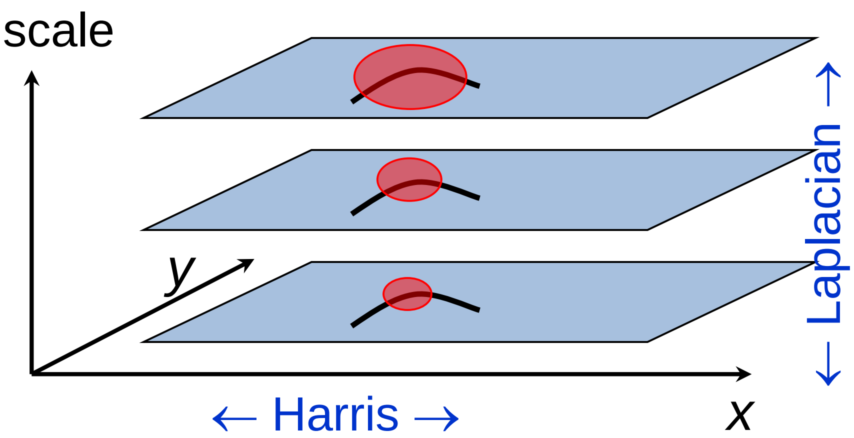
(4) Keypoint descriptor
- Use local image gradients at the selected scale.

D. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”, **International Journal of Computer Vision**, 60(2):91-110, 2004.

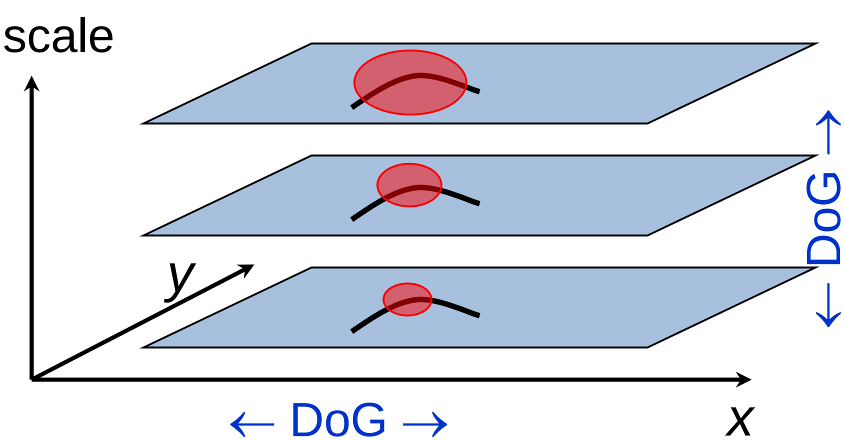
Cited 9589 times (as of 3/7/2011)

Scale Invariant Detectors

- Harris-Laplacian[1]
  Find local maximum of:
  – Harris corner detector in space (image coordinates)
  – Laplacian in scale



- SIFT (Lowe)[2]
  Find local maximum of:
  – **Difference of Gaussians** in space and scale



[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 200
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". Accepted to IJCV

Harris-Laplacian vs. SIFT (Lowe)

Harris-Laplacian Corner Detector:

- **Rotation invariance**
- **Partial invariance to *affine intensity* change**
- But: **non-invariant to *image scale*!**

SIFT (Lowe) Detector:

- **Affine Invariance (including rotation)**
- **Scale Invariance**
- **Intensity Change Invariance**

Applications of Keypoints & SIFT

Wide baseline stereo

[Image from T. Tuytelaars ECCV 2006 tutorial]

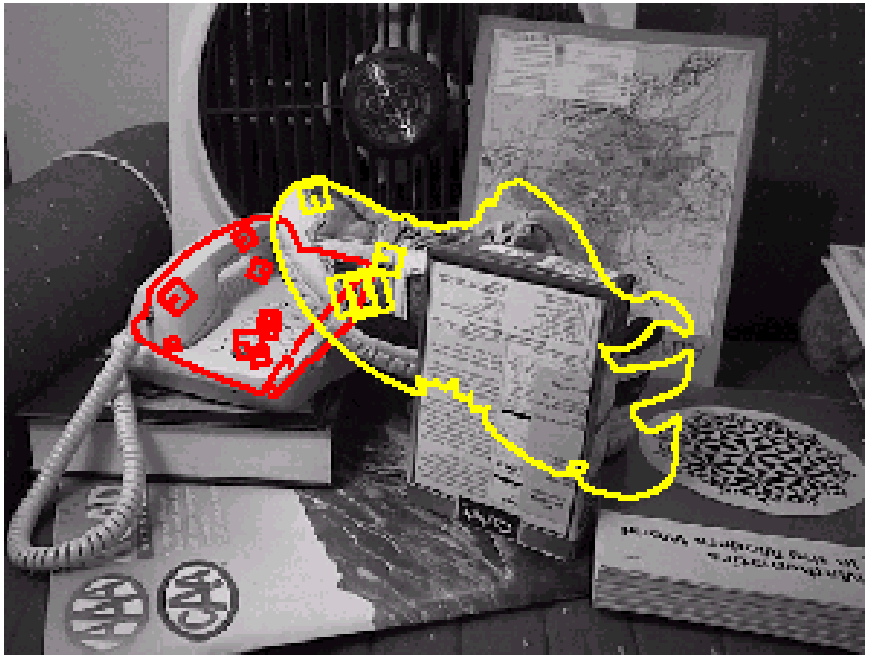
Panorama stitching

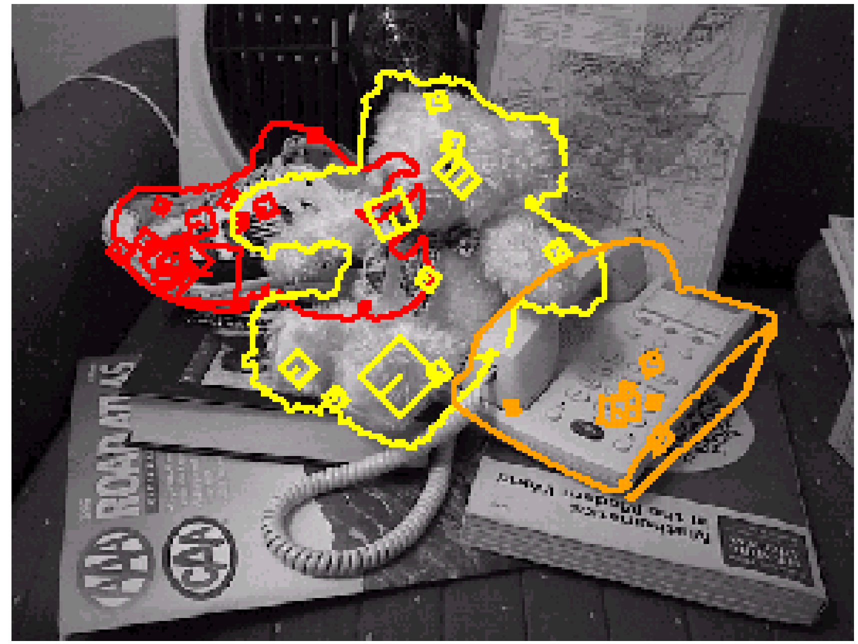
(a) Matier data set (7 images)

(b) Matier final stitch

Brown, Szeliski, and Winder, 2005

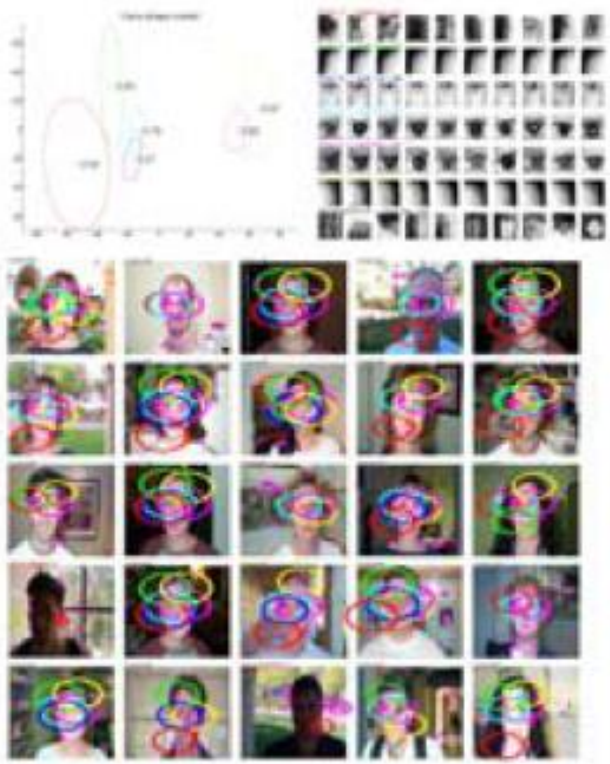
Recognition under occlusion





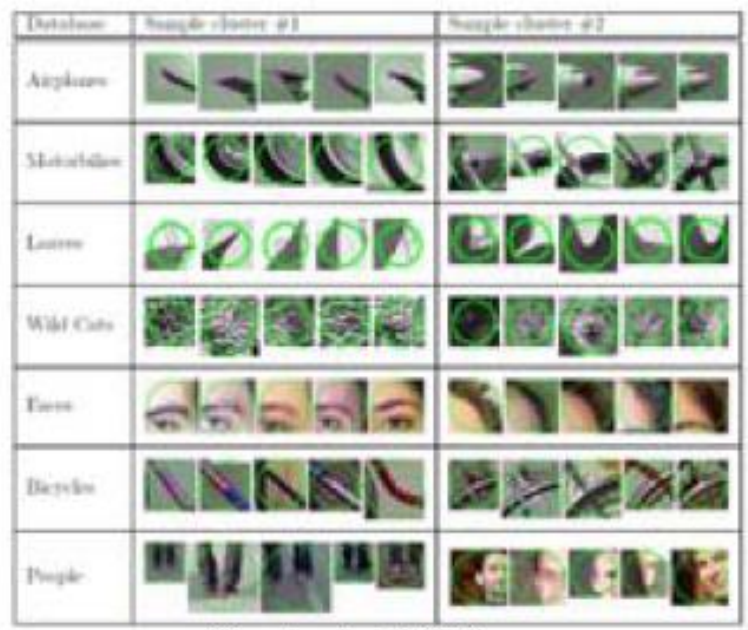
Recognition of categories

Constellation model



Weber et al. (2000)
Fergus et al. (2003)

Bags of words

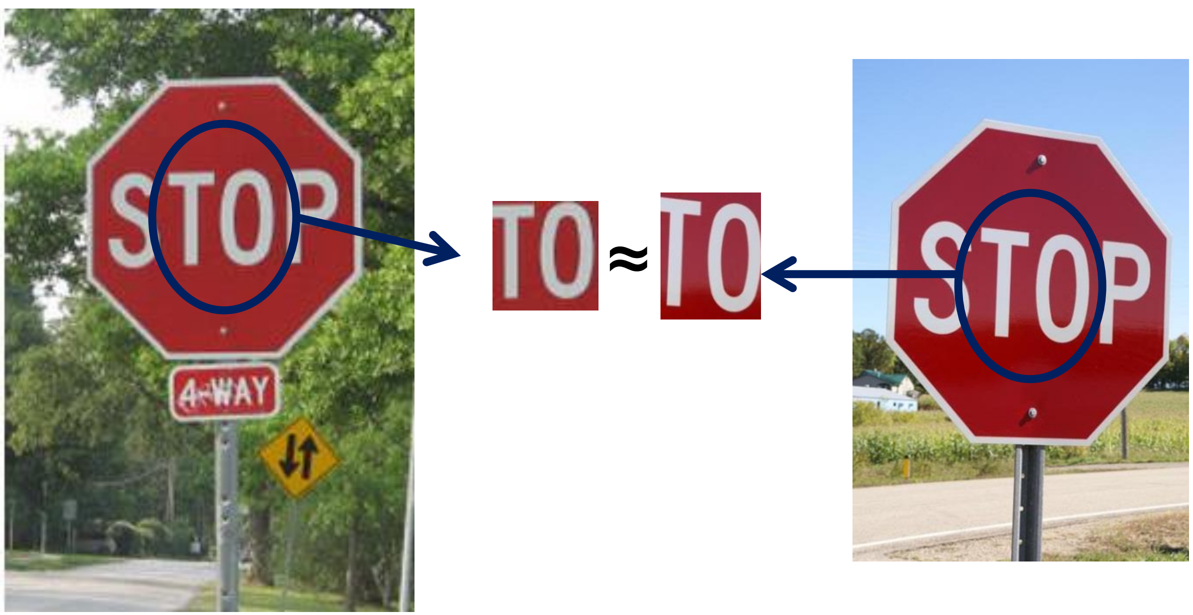


Csurka et al. (2004)
Dorko & Schmid (2005)
Sivic et al. (2005)
Lazebnik et al. (2006), …

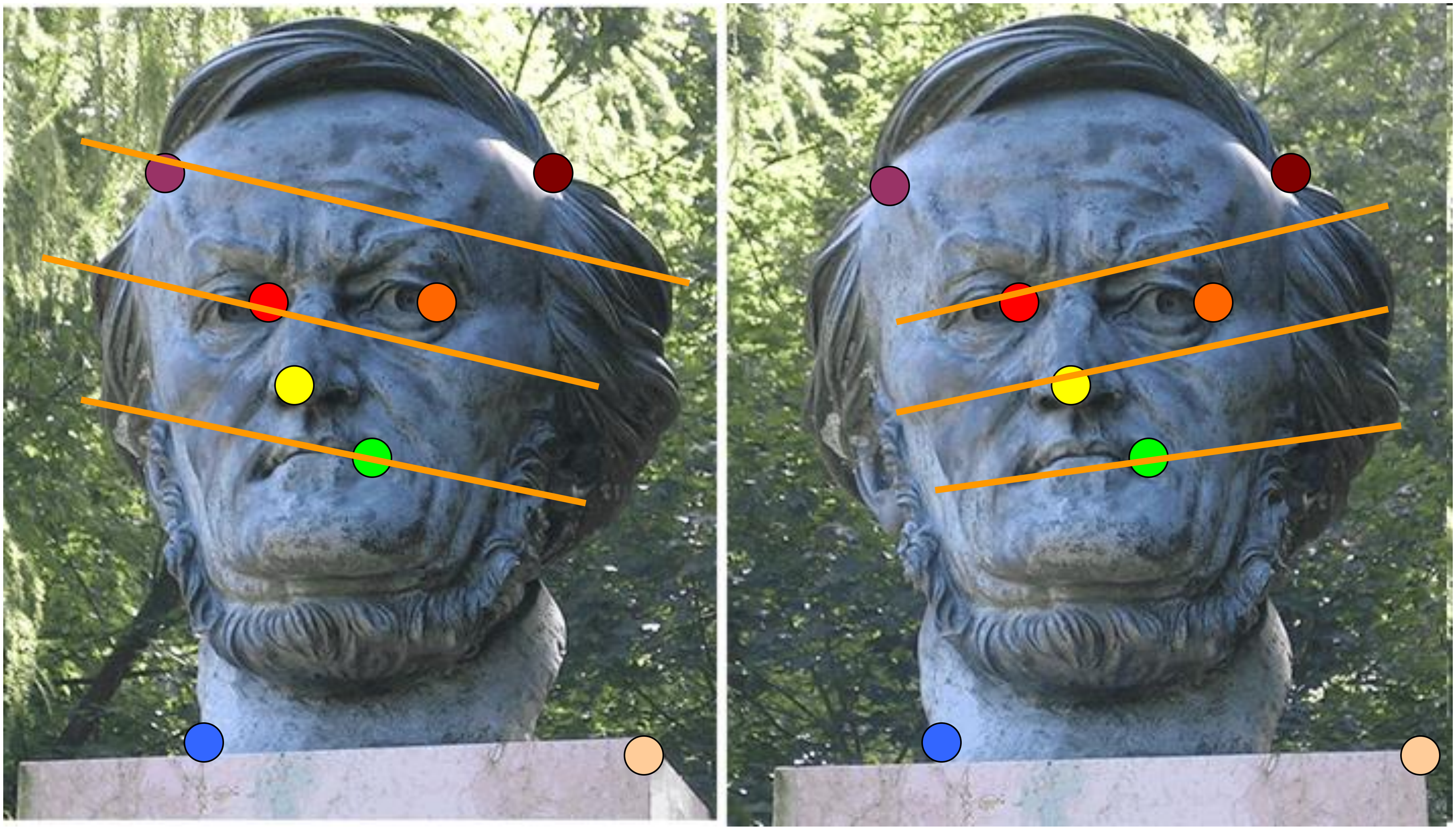
[Slide from Lazebnik, Sicily 2006]

Correspondence across views

- Correspondence: matching points, patches, edges, or regions across images



Example: estimating "fundamental matrix" that corresponds two views

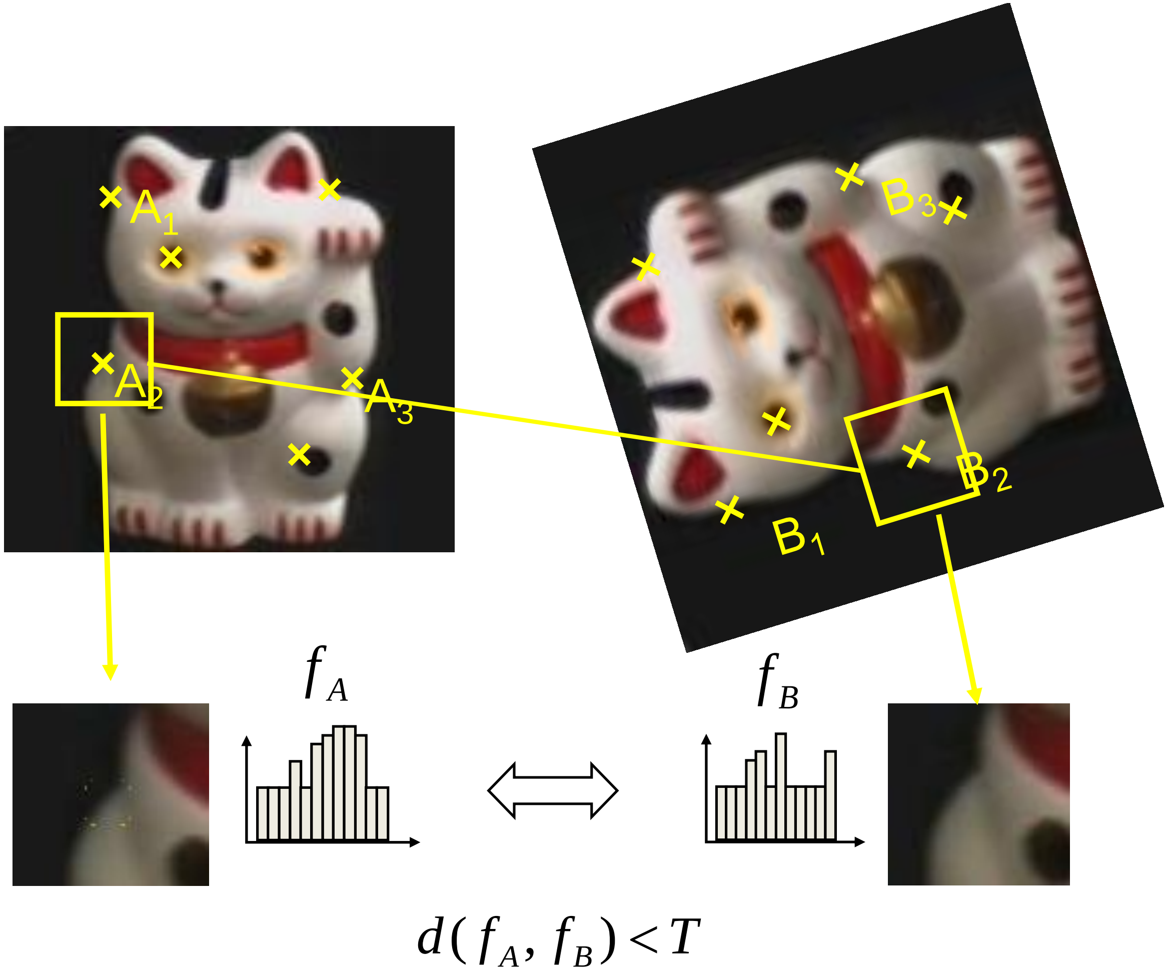


Slide from Silvio Savarese

Applications

- Feature points are used for:
  - **Image alignment**
  - 3D reconstruction
  - Motion tracking
  - Robot navigation
  - Indexing and database retrieval
  - Object recognition

Overview of Keypoint Matching



1. **Find a set of distinctive key-points**
2. **Define a region around each keypoint**
3. **Compute a local descriptor from the normalized region**
4. **Match local descriptors**

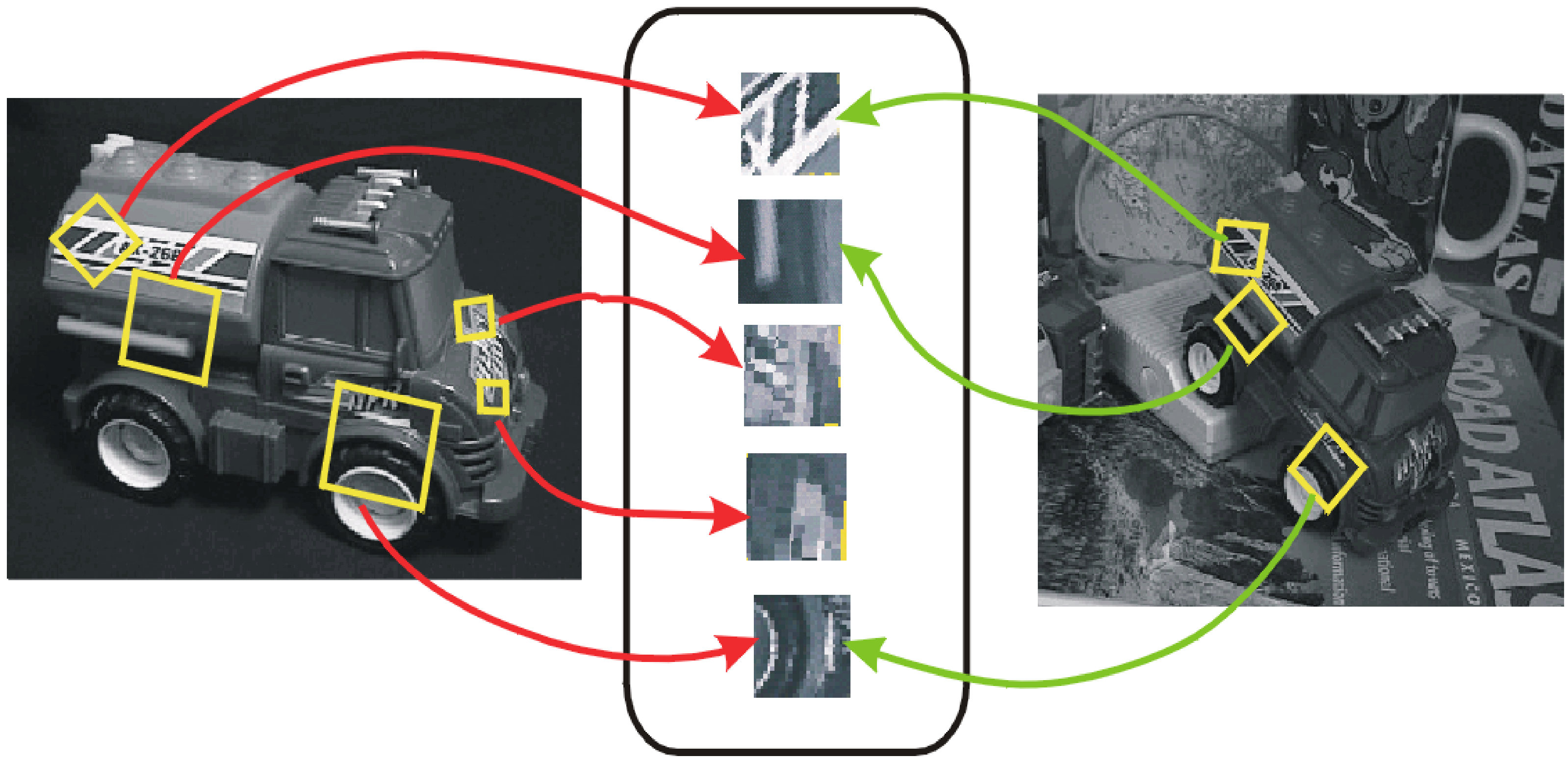
K. Grauman, B. Leibe

Goals for Keypoints

Detect points that are *repeatable* and *distinctive*

Invariant Local Features

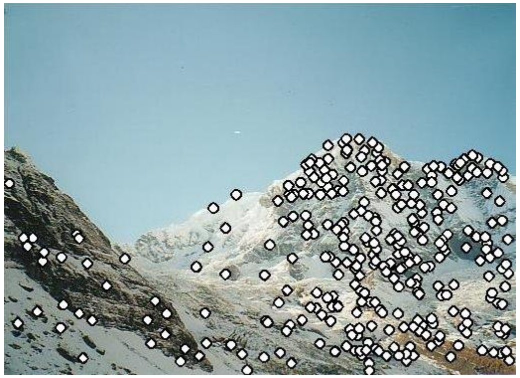
Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



Features Descriptors

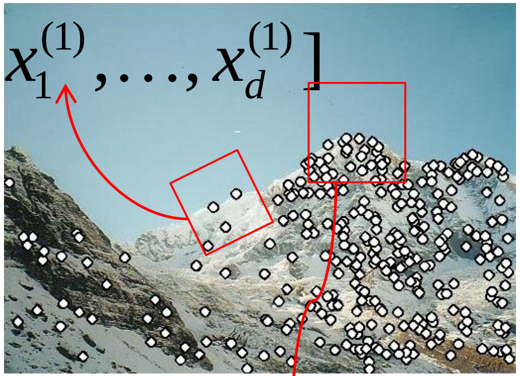
Local features: main components

1) **Detection**: Identify the interest points



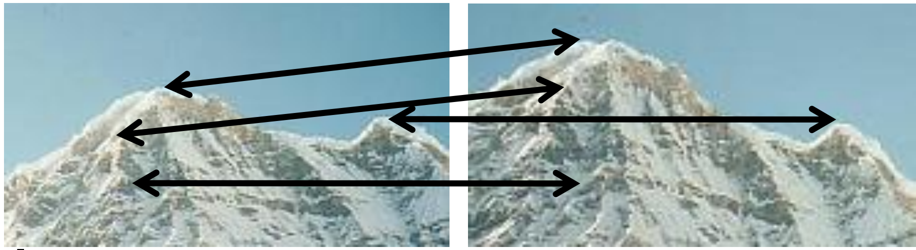
2) **Description**: Extract vector feature descriptor surrounding each interest point.

$$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$$



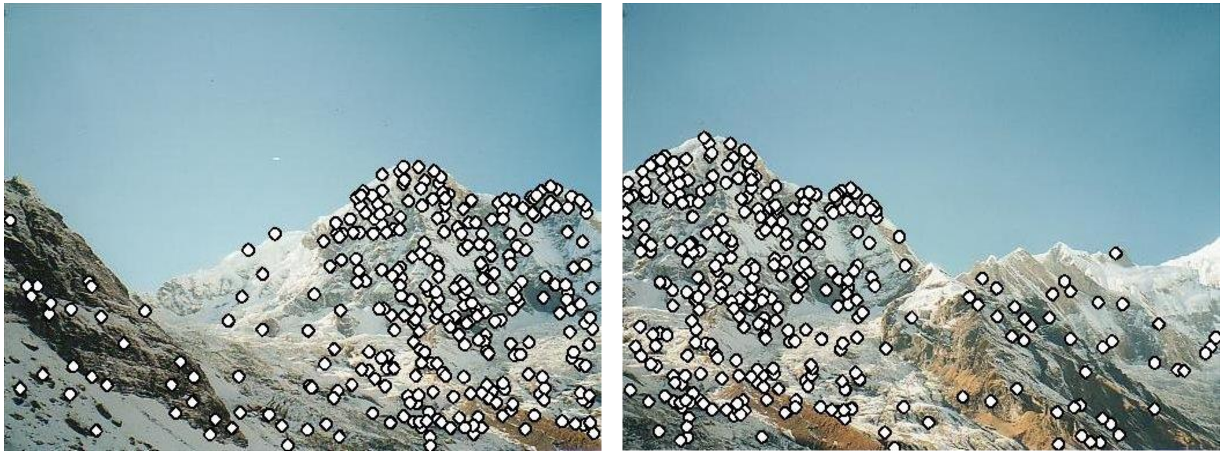
$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$

3) **Matching**: Determine correspondence between descriptors in two views



Matching could be very time-consuming!

Characteristics of good features



- **Repeatability**
  - The same feature can be found in several images despite geometric and photometric transformations
- **Saliency**
  - Each feature is distinctive
- **Compactness and efficiency**
  - Many fewer features than image pixels
- **Locality**
  - A feature occupies a relatively small area of the image; robust to clutter and occlusion

Goal: interest operator repeatability

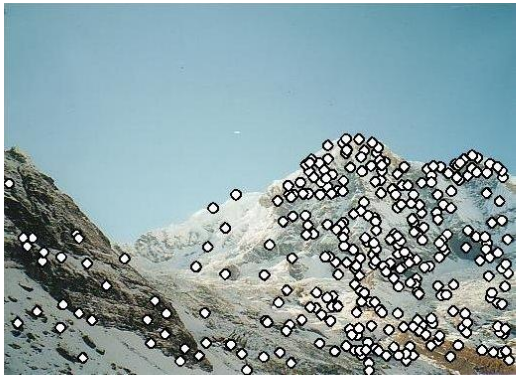
- We want to detect (at least some of) the same points in both images.


No chance to find true matches!

- Yet we have to be able to run the detection procedure *independently* per image.

Kristen Grauman

Local features: main components



1) Detection: Identify the interest points

2) Description: Extract vector feature descriptor surrounding each interest point.

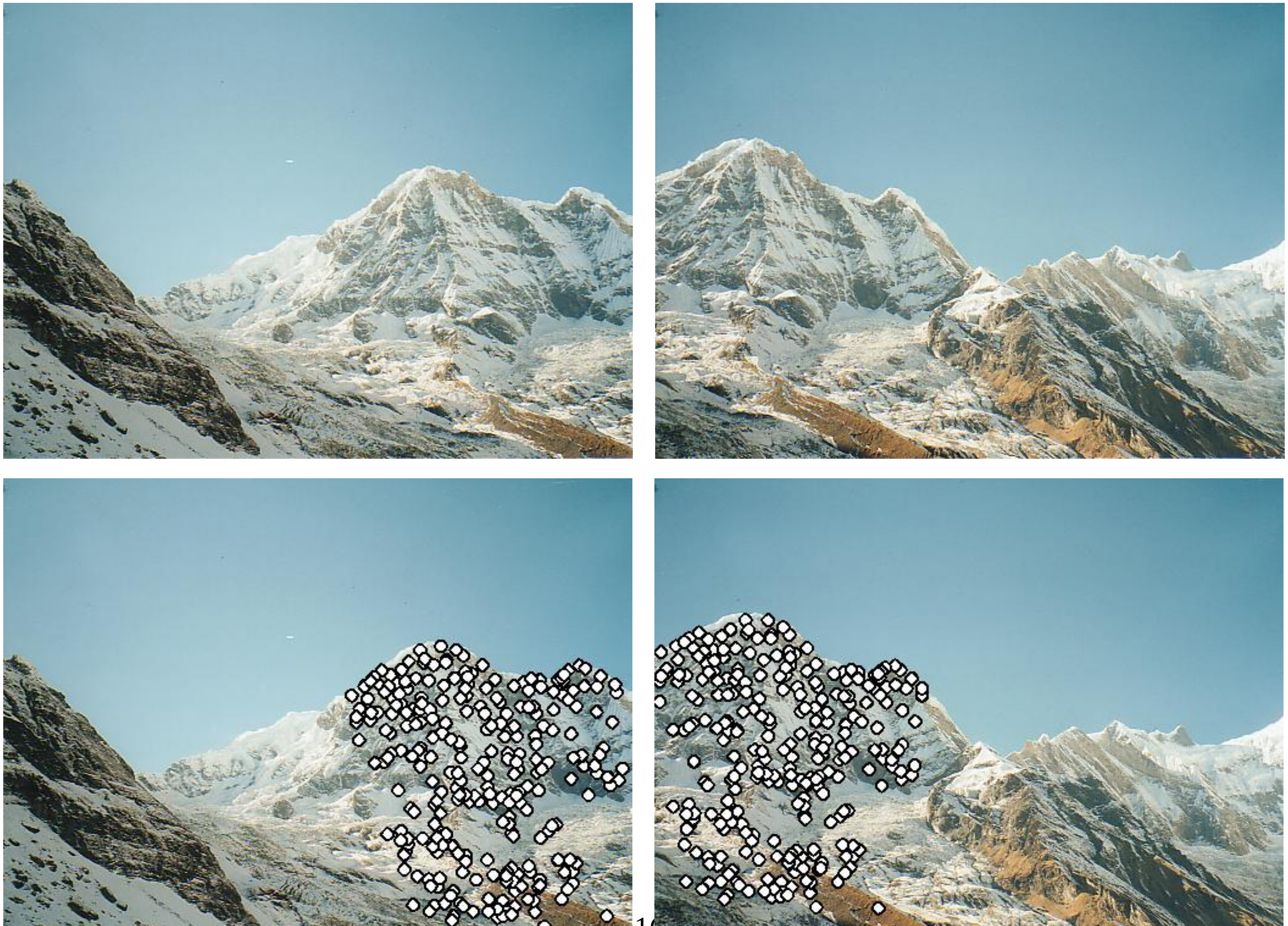
3) Matching: Determine correspondence between descriptors in two views

Matching could be very time-consuming!

Many Existing Detectors Available

Hessian & Harris	[Beaudet ‘78], [Harris ‘88]
Laplacian, DoG	[Lindeberg ‘98], [Lowe 1999]
Harris-/Hessian-Laplace	[Mikolajczyk & Schmid ‘01]
Harris-/Hessian-Affine	[Mikolajczyk & Schmid ‘04]
EBR and IBR	[Tuytelaars & Van Gool ‘04]
MSER	[Matas ‘02]
Salient Regions	[Kadir & Brady ‘01]
Others…	

Some Matching Results from Matt Brown



Some Matching Results