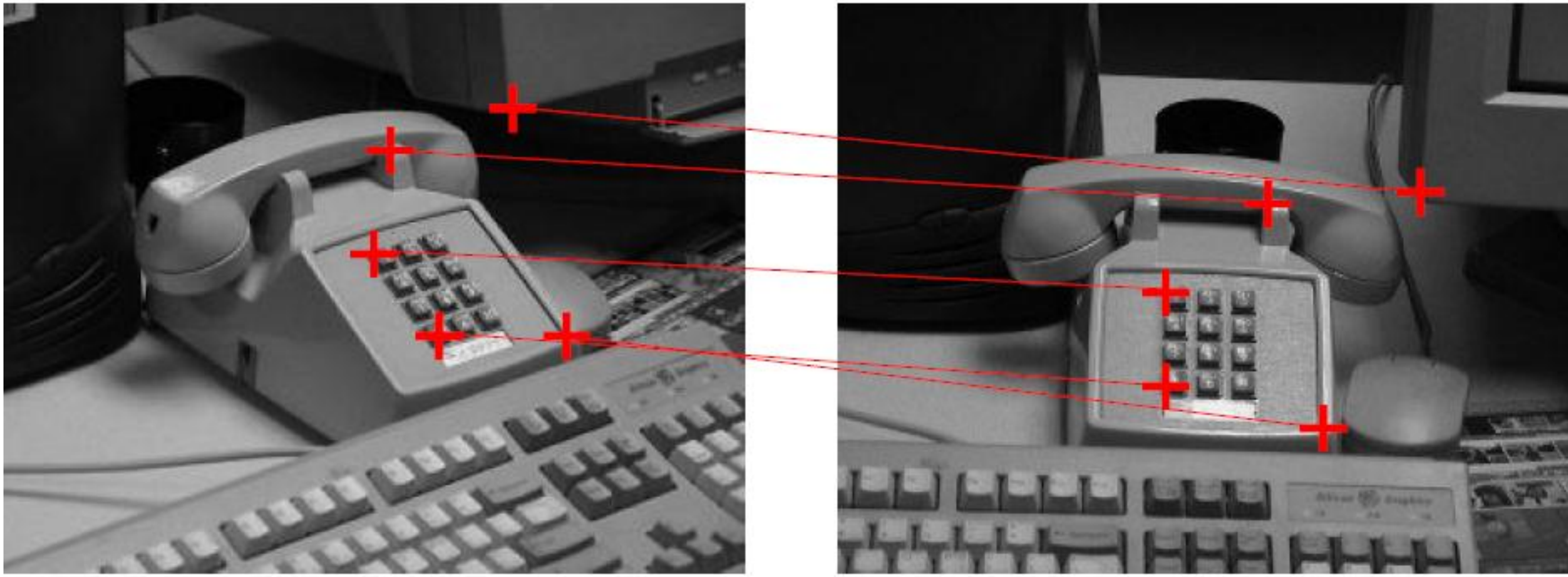
Detectors: Harris, Scale invariance, Harris-Laplace, affine Harris-Laplace

Example: Finding Correspondences Between Images



- First step toward 3-D reconstruction: Find correspondences between *feature points* in two images of a scene
- Object recognition: Find correspondences between *feature points* in “training” and “test” image

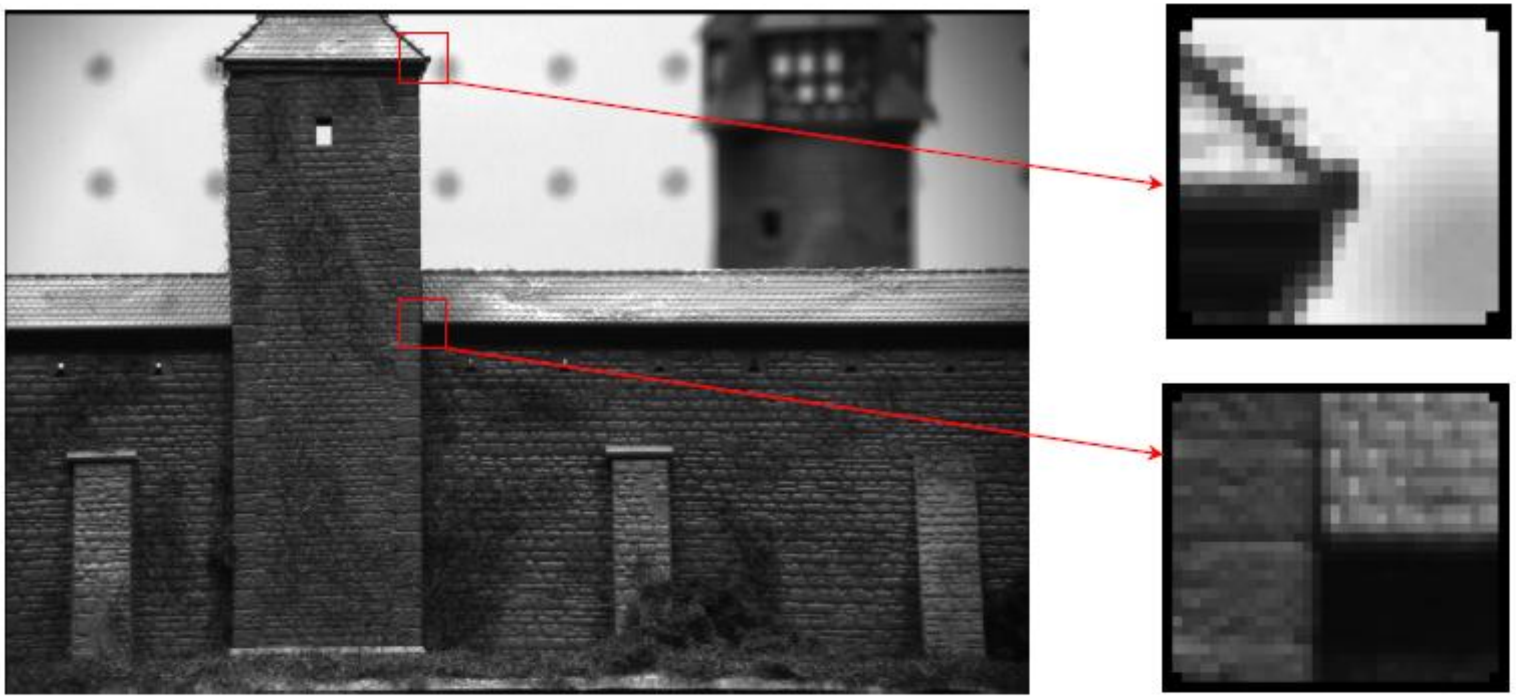
2,106 images, 819,242 points

http://grail.cs.washington.edu/projects/rome/

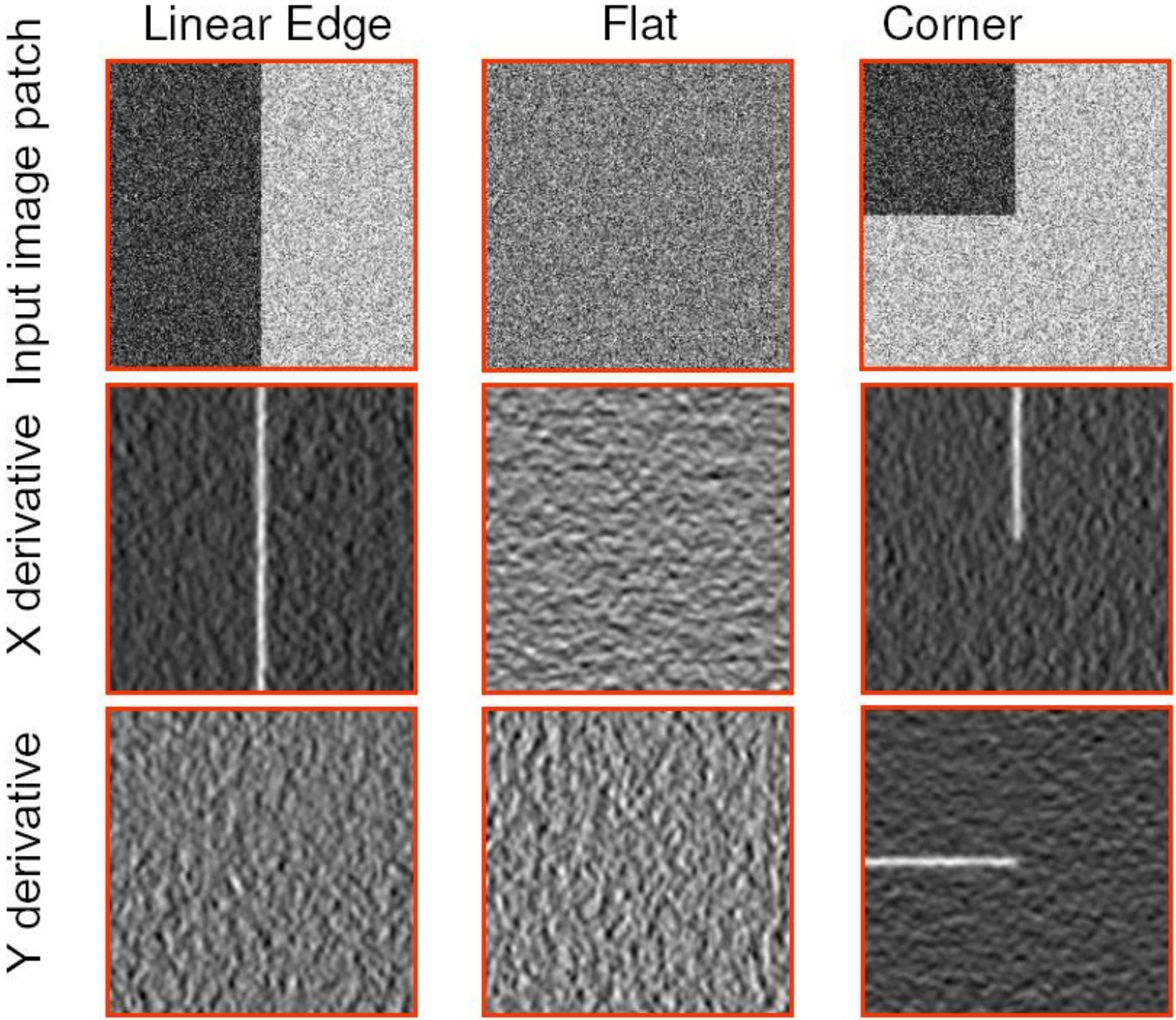
Building Rome in a Day
Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz and Richard Szeliski
International Conference on Computer Vision, 2009, Kyoto, Japan.

Interest points as “corner-like”: Basic Harris detector

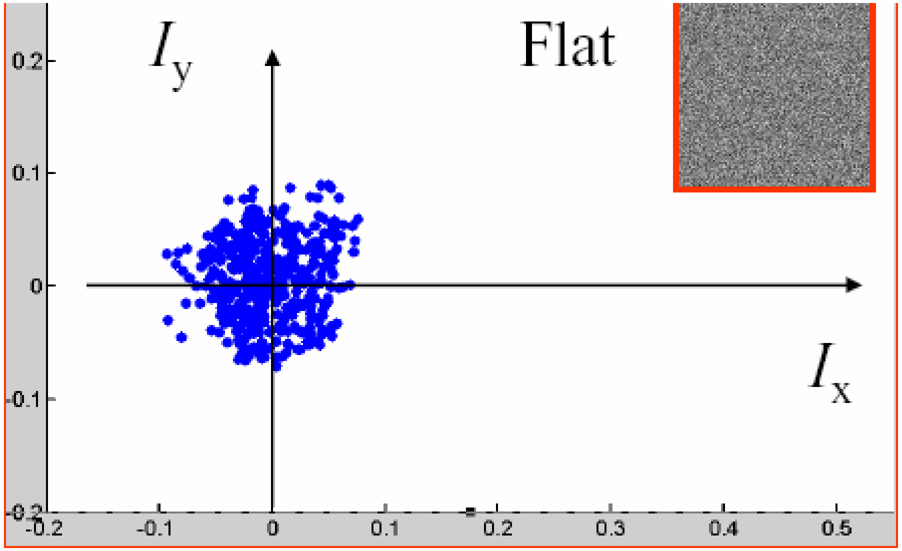
Interest Points

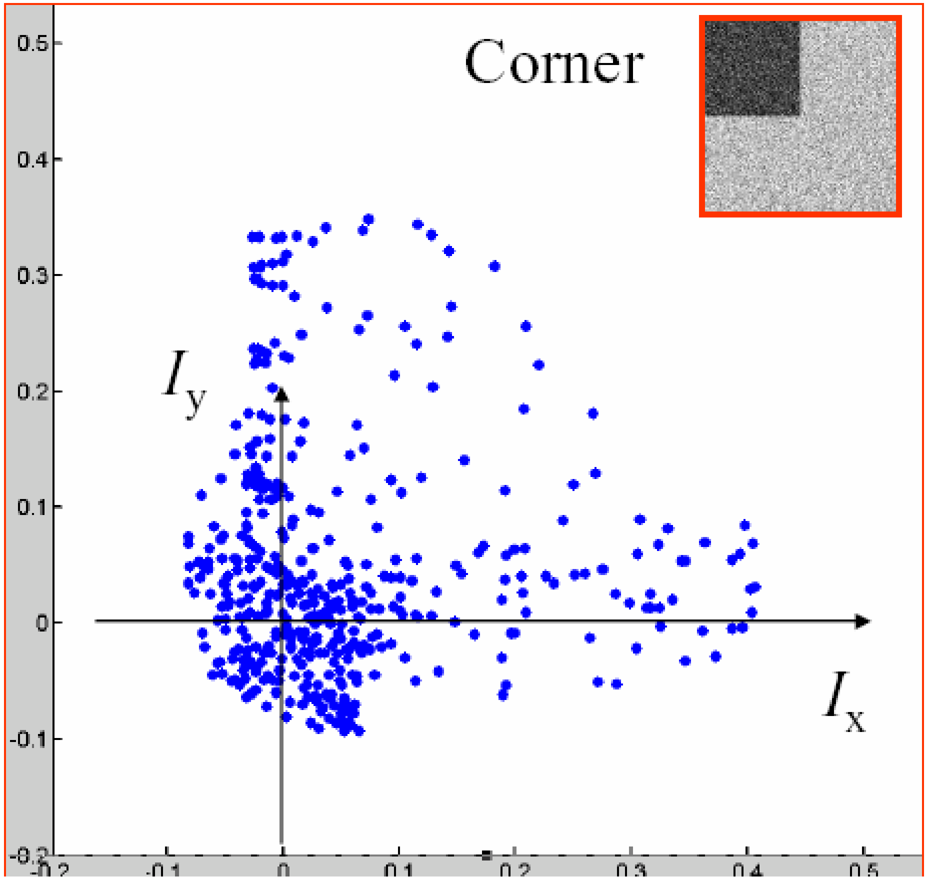


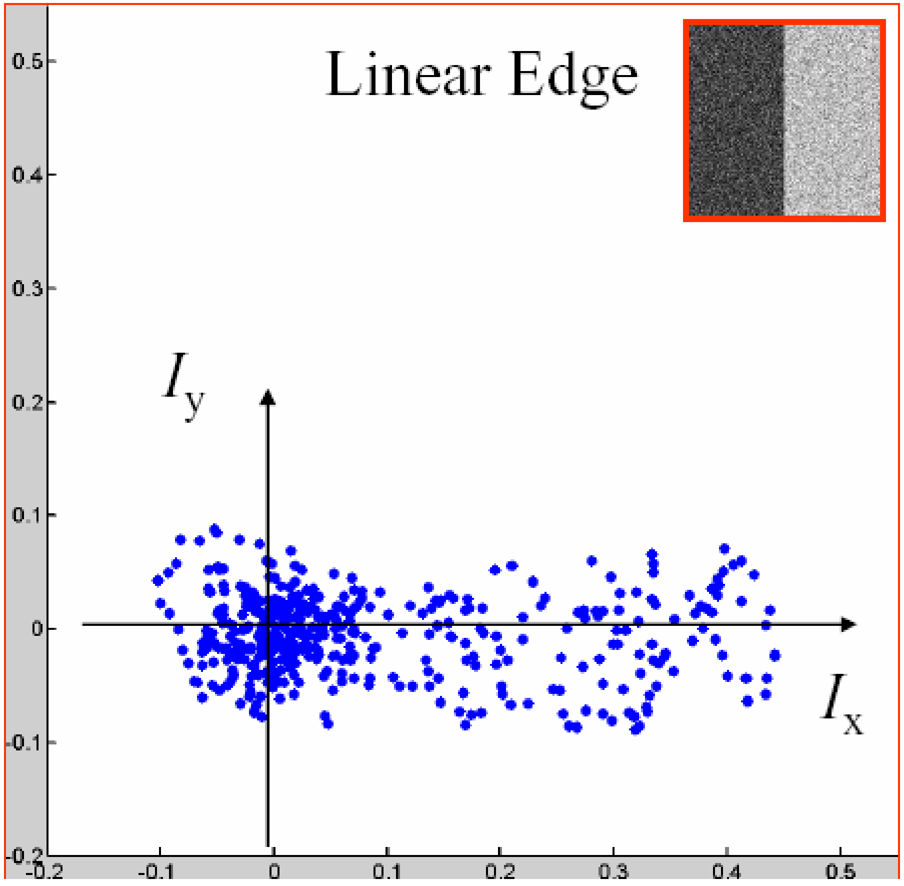
• Intuitively, junctions of contours.
• Generally more stable features over changes of viewpoint
• Intuitively, large variations in the neighborhood of the point in all directions


Linear Edge
Flat
Corner
Input image patch
X derivative
Y derivative

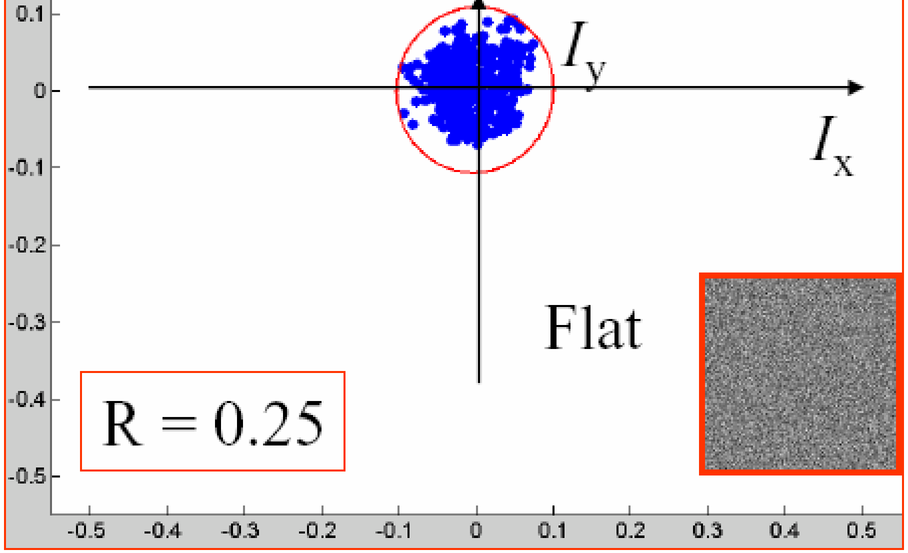
The distribution of the *x* and *y* derivatives is very different for all three types of patches

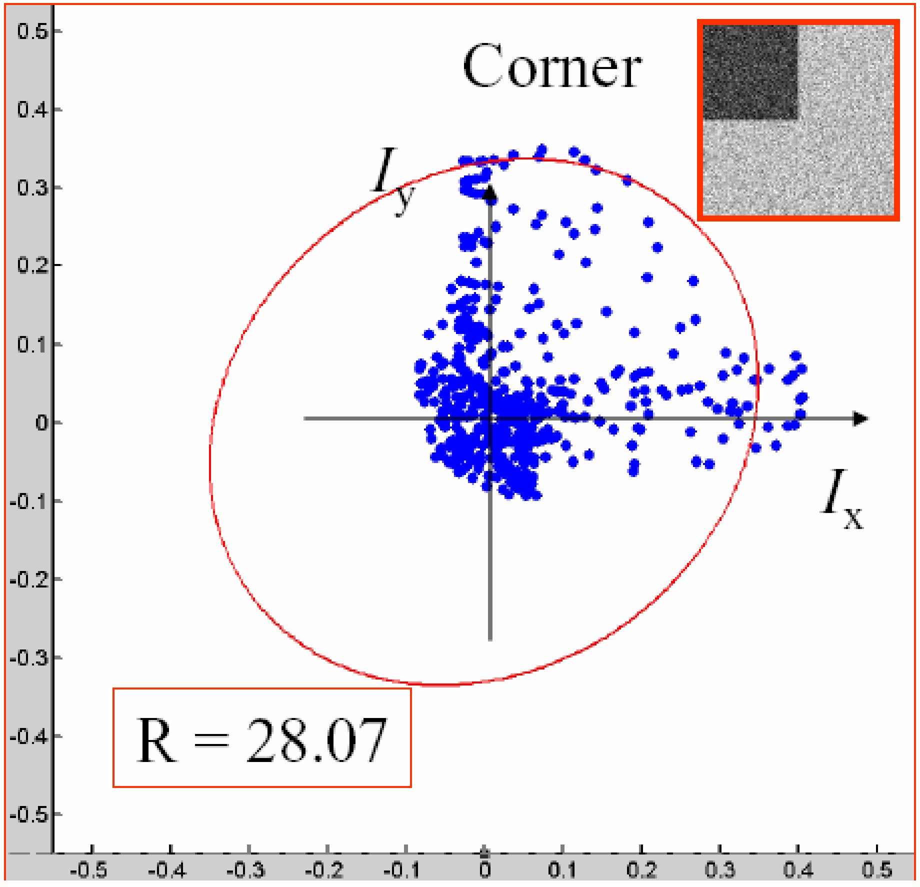


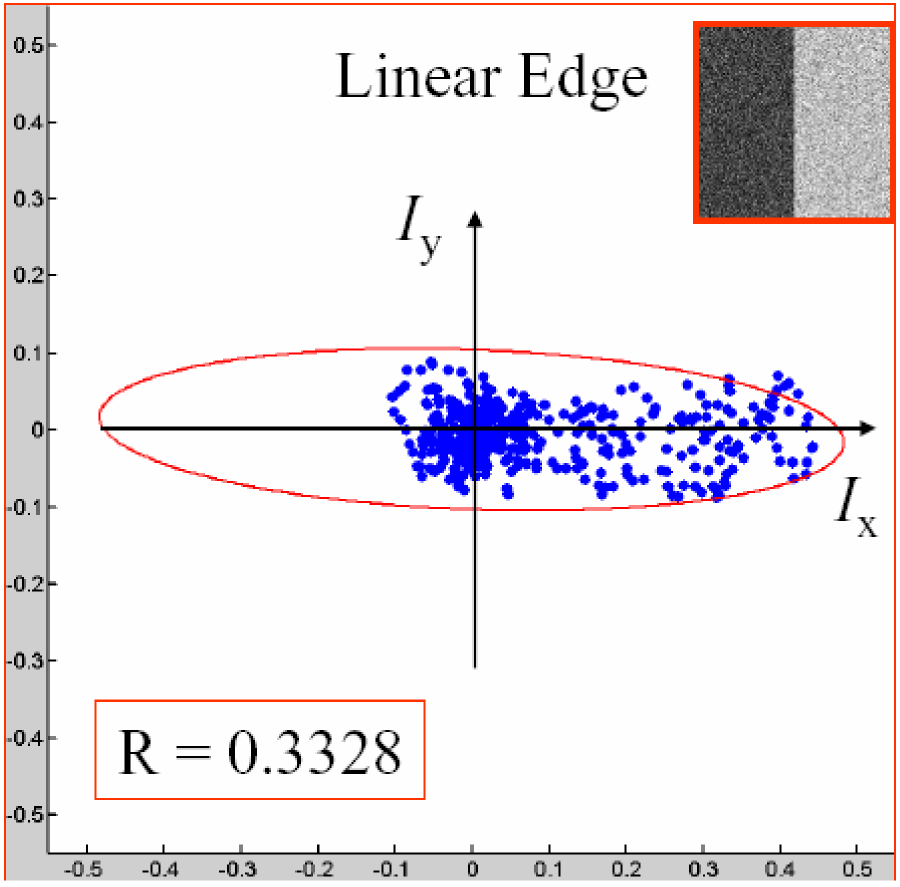




The distribution of x and y derivatives can be characterized by the shape and size of the principal component ellipse







How to evaluate "interestness"?

- The distribution of gradients in a neighborhood W is represented by the inertia (shape, Harris) matrix:

$$\boldsymbol{H} = \begin{bmatrix} \sum_{W} \boldsymbol{I}_x^2 & \sum_{W} \boldsymbol{I}_x \boldsymbol{I}_y \\ \sum_{W} \boldsymbol{I}_x \boldsymbol{I}_y & \sum_{W} \boldsymbol{I}_y^2 \end{bmatrix}$$

- Elongations of the distribution = Eigenvalues of H: λ_{min}, λ_{max}
  – We want λ_{min}, λ_{max} to be approx. equal
  – We want λ_{min}, λ_{max} to be large

Harris detector and its variants

– We want $\lambda_{\min}$, $\lambda_{\max}$ to be approx. equal

– We want $\lambda_{\min}$, $\lambda_{\max}$ to be large

$$\boldsymbol{R} = 4\frac{\lambda_{\min}\lambda_{\max}}{\left(\lambda_{\min} + \lambda_{\max}\right)^2}$$

- ($R = 1$ if $\lambda_{\min}=\lambda_{\max}$ but keep only the ones with large $\lambda_{\max}$)

$$\boldsymbol{R} = \frac{\lambda_{\min}\lambda_{\max}}{\left(\lambda_{\min} + \lambda_{\max}\right)}$$

$$\boldsymbol{R} = \lambda_{\min}\lambda_{\max} - \boldsymbol{k}\left(\lambda_{\min} + \lambda_{\max}\right)^2$$

Comp. efficient definition

- For any symmetric matrix H:
  - $\text{Det}(H) = \lambda_{\min}\lambda_{\max}$
  - $\text{Trace}(H) = \lambda_{\min}+\lambda_{\max}$ (The trace is the sum of the diagonal of H)

$$R = 4\frac{Det(H)}{Trace(H)^2}$$

$$R = \frac{Det(H)}{Trace(H)}$$

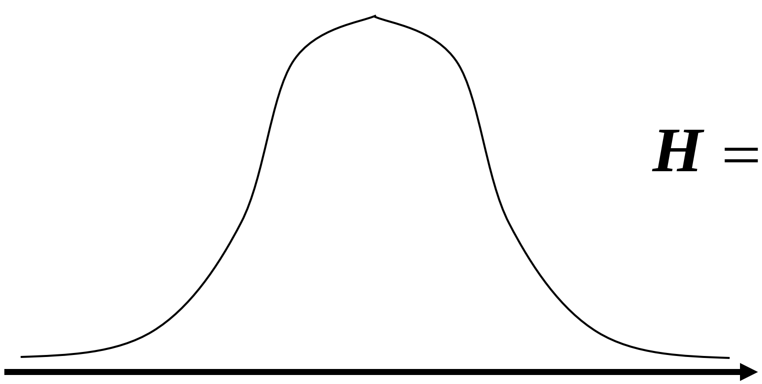
$$R = Det(H) - kTrace(H)^2$$

Computation of H and gradients

- I_x means convolution with Gaussian of σ
- H should computed with different weights → Convolution with Gaussian

Constant weights

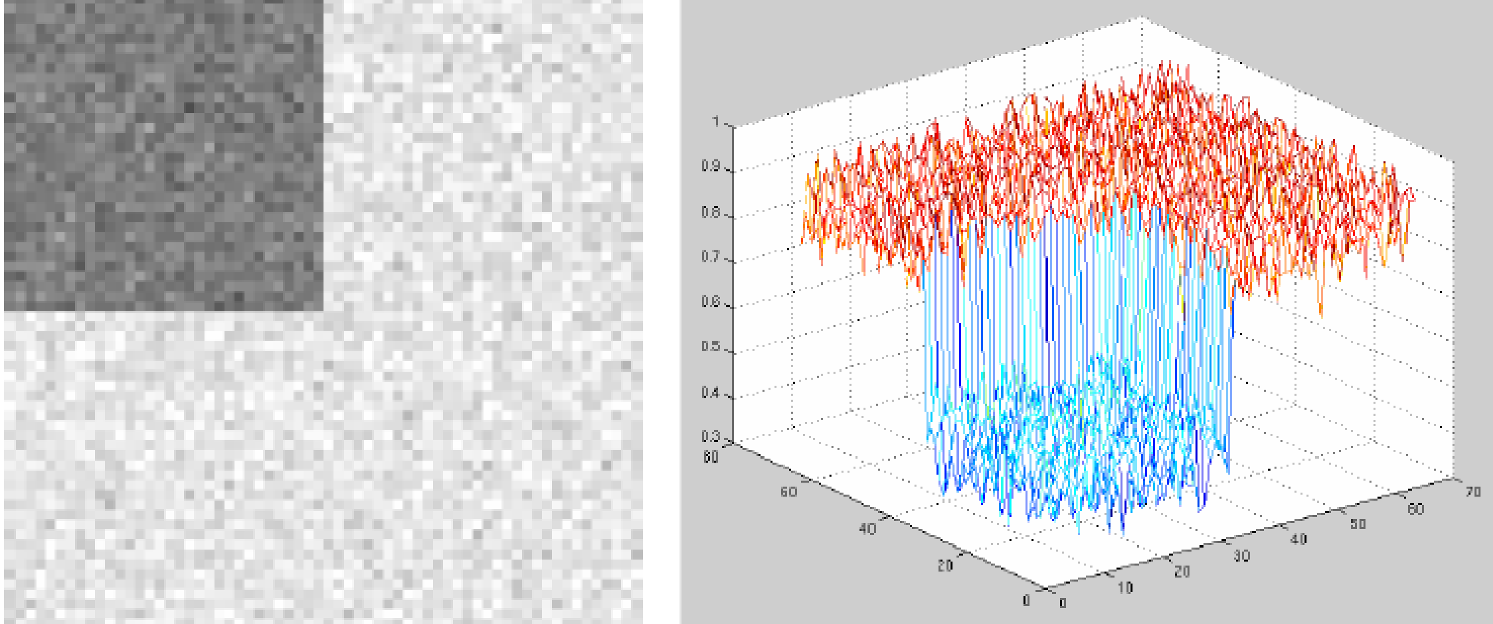
$$\boldsymbol{H} = \begin{bmatrix} \sum_{W} \boldsymbol{I}_x^2 & \sum_{W} \boldsymbol{I}_x \boldsymbol{I}_y \\ \sum_{W} \boldsymbol{I}_x \boldsymbol{I}_y & \sum_{W} \boldsymbol{I}_y^2 \end{bmatrix}$$



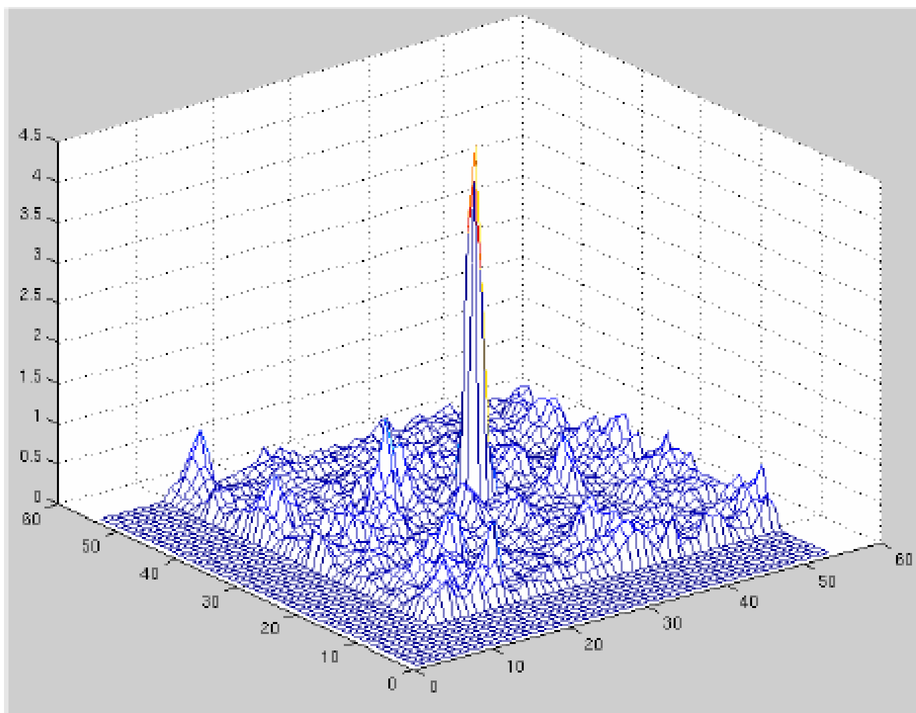
Gaussian weights

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{G}_\sigma * \boldsymbol{I}_x^2 & \boldsymbol{G}_\sigma * \boldsymbol{I}_x \boldsymbol{I}_y \\ \boldsymbol{G}_\sigma * \boldsymbol{I}_x \boldsymbol{I}_y & \boldsymbol{G}_\sigma * \boldsymbol{I}_y^2 \end{bmatrix} = \boldsymbol{G}_\sigma * \begin{bmatrix} \boldsymbol{I}_x^2 & \boldsymbol{I}_x \boldsymbol{I}_y \\ \boldsymbol{I}_x \boldsymbol{I}_y & \boldsymbol{I}_y^2 \end{bmatrix}$$

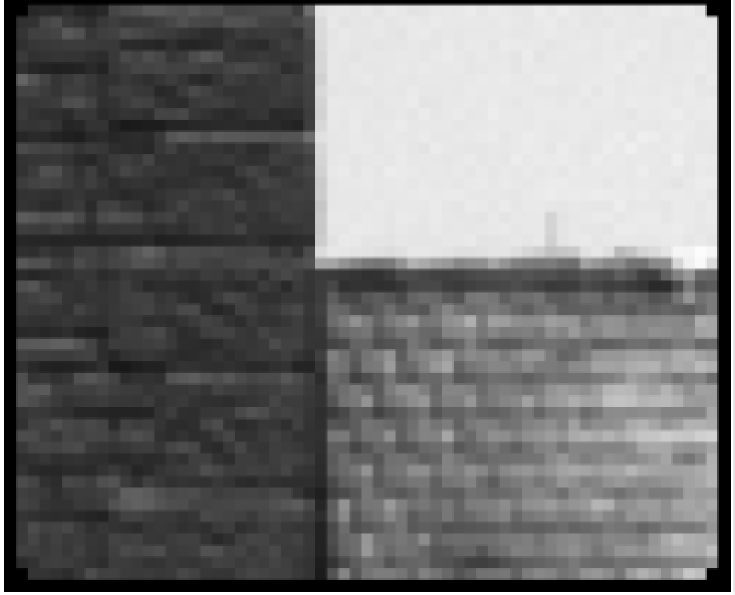
Input

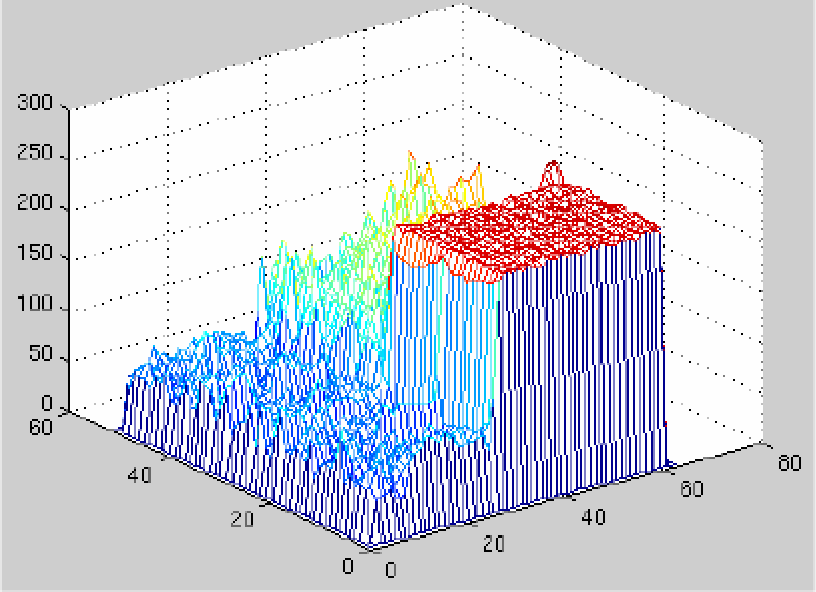


Response of interest point operator

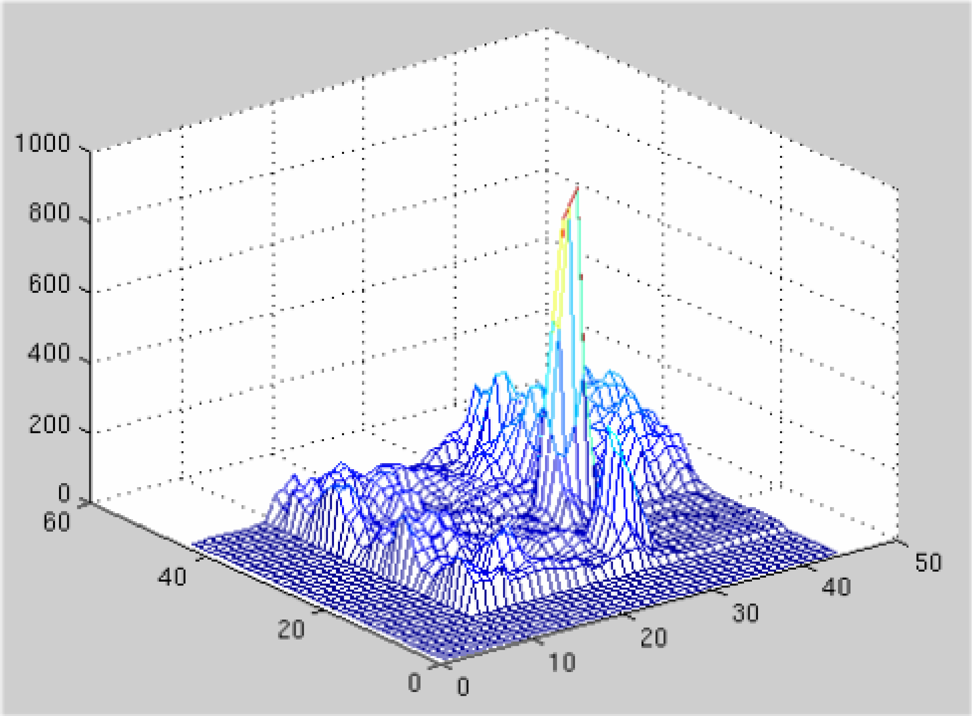


Input





Response of interest point operator



Interest point detection example

Interest points example

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

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

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

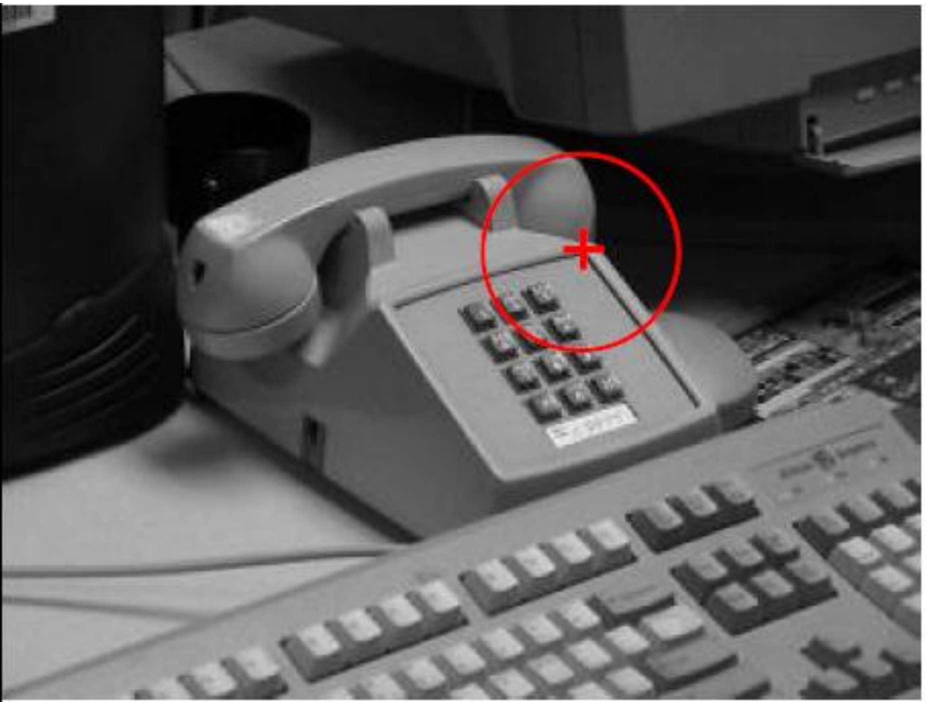
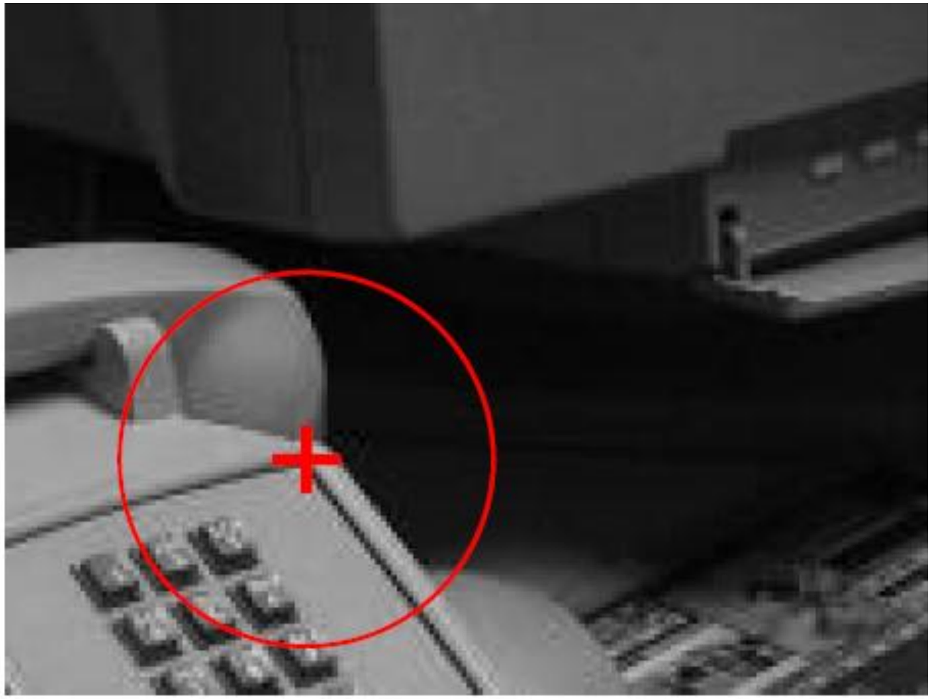
$$H(x,y) = \begin{bmatrix} S_{x2}(x,y) & S_{xy}(x,y) \\ S_{xy}(x,y) & S_{y2}(x,y) \end{bmatrix}$$

5. Compute the response of the detector at each pixel

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

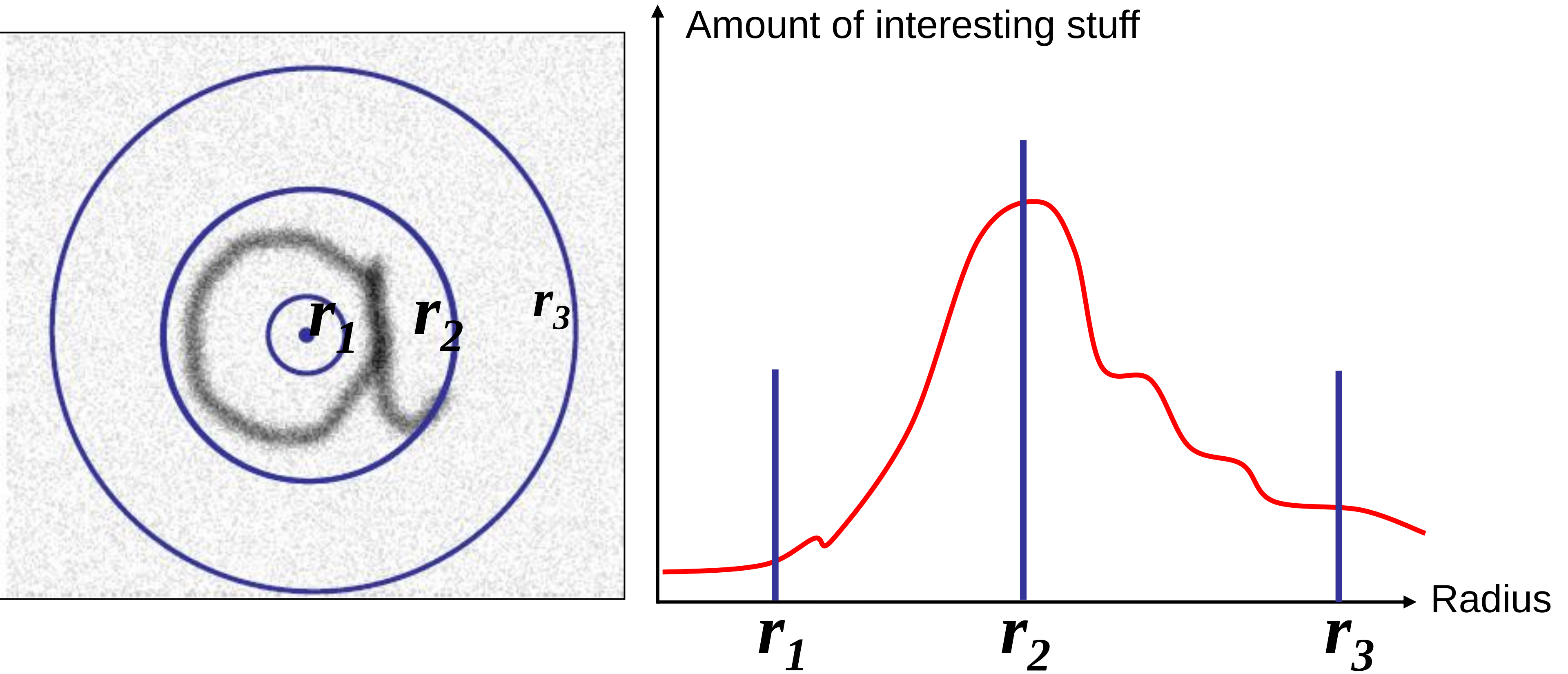
Characteristic scale

Scale Selection

- We need to decide what window size to use for computing the Harris matrix
- Equivalently, we need to choose the value of σ'
- The window size (or σ') must be consistent between different magnifications of the image

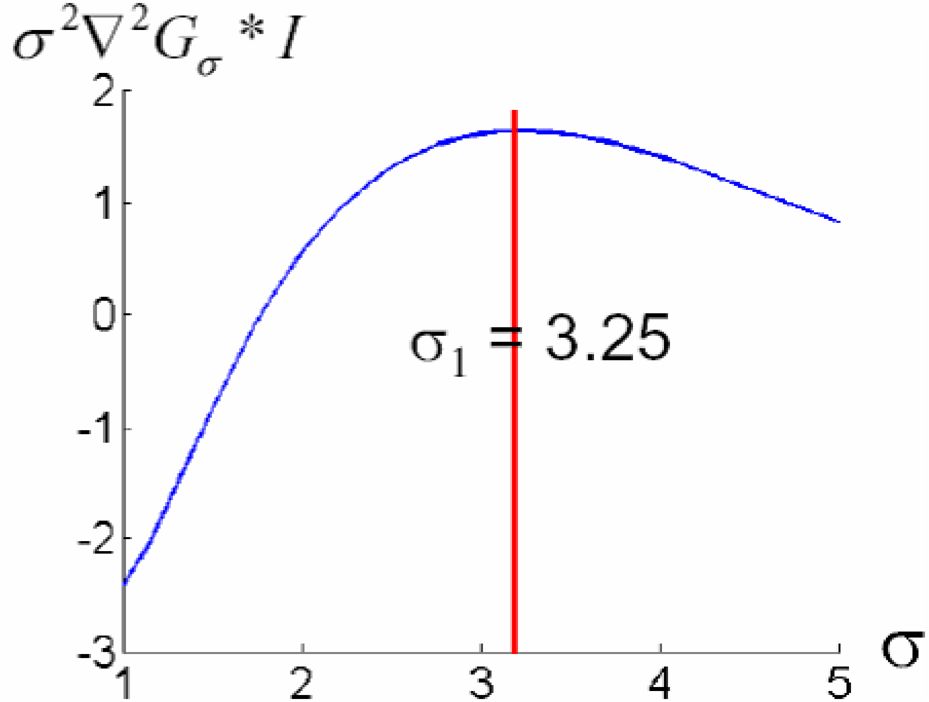
How to choose a neighborhood size? → Intuition

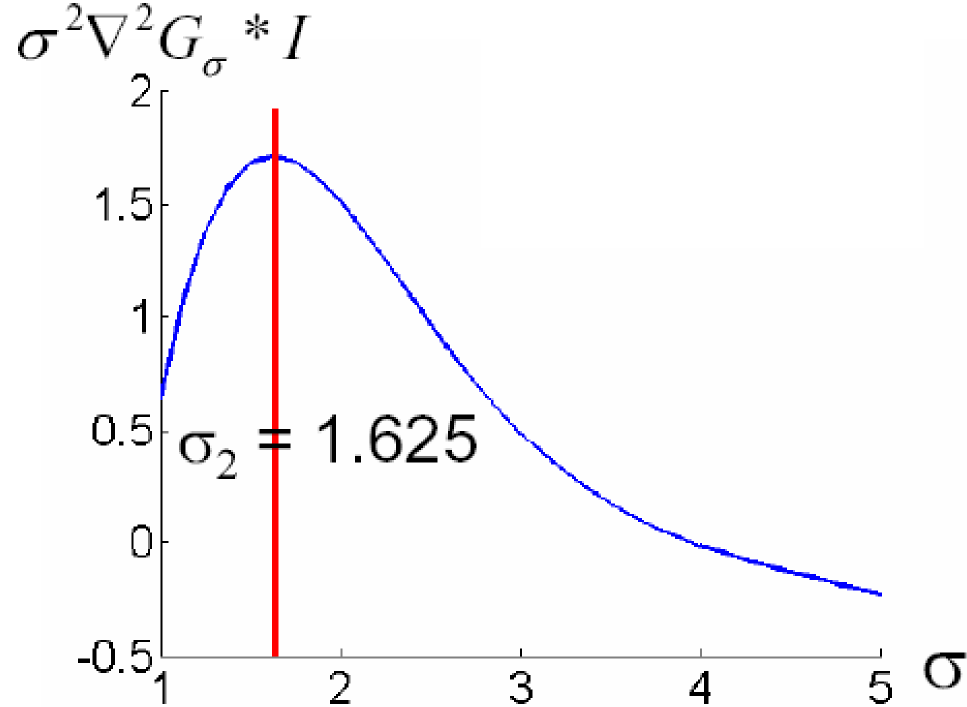


Best radius: Local extrema of function that measures amount of interesting stuff

X 2

$\sigma_1 = 2 \times \sigma_2$!!!





- Why did we use the normalized Laplacian in the previous example?
- Justification (and basis for most scale selection operations in computer vision):
- *Scale Selection Principle (T. Lindeberg):*

In the absence of other evidence, assume that a scale level, at which some (possibly non-linear) combination of normalized derivatives assumes a local maximum over scales, can be treated as reflecting a characteristic length of a corresponding structure in the data.

What are normalized derivatives?

$$\sigma^{n+m} \frac{\partial^{n+m} f}{\partial x^n \partial y^m}$$

Example using 2nd order derivatives

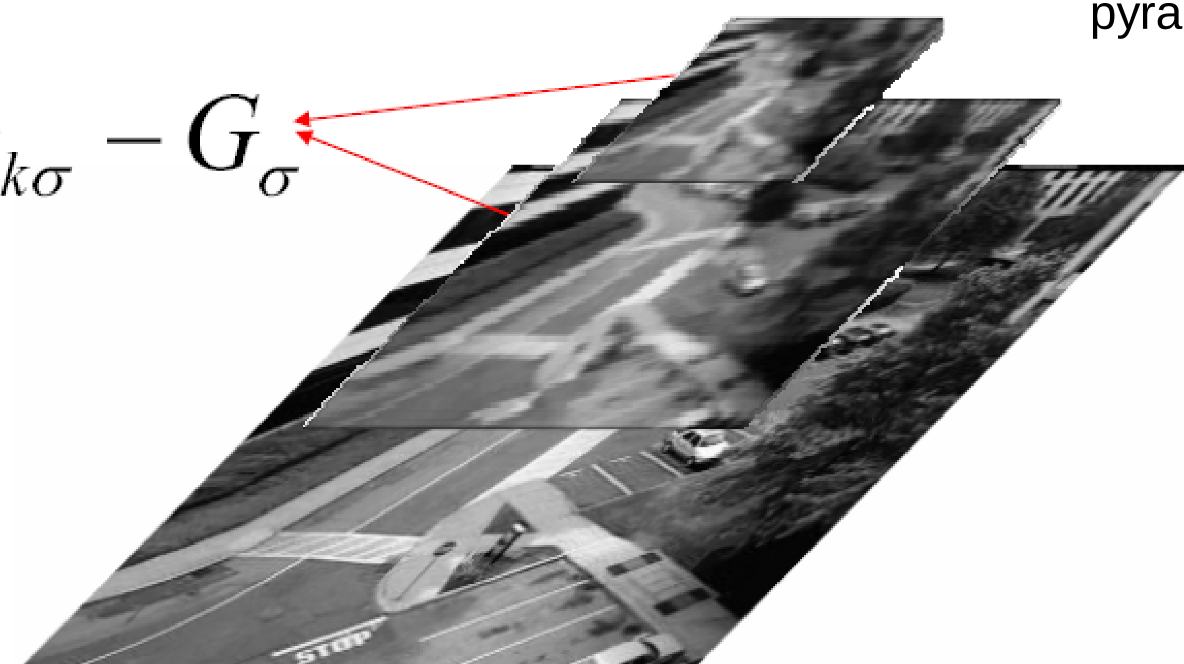
$$\sigma^2 \nabla^2 f = \sigma^2 \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right)$$

$$\boxed{\nabla_\sigma^2 I = \nabla^2 G_\sigma * I}$$

Constant independent of σ

$$\boxed{G_{k\sigma} - G_\sigma \approx (k-1)\sigma^2 \nabla^2 G_\sigma}$$

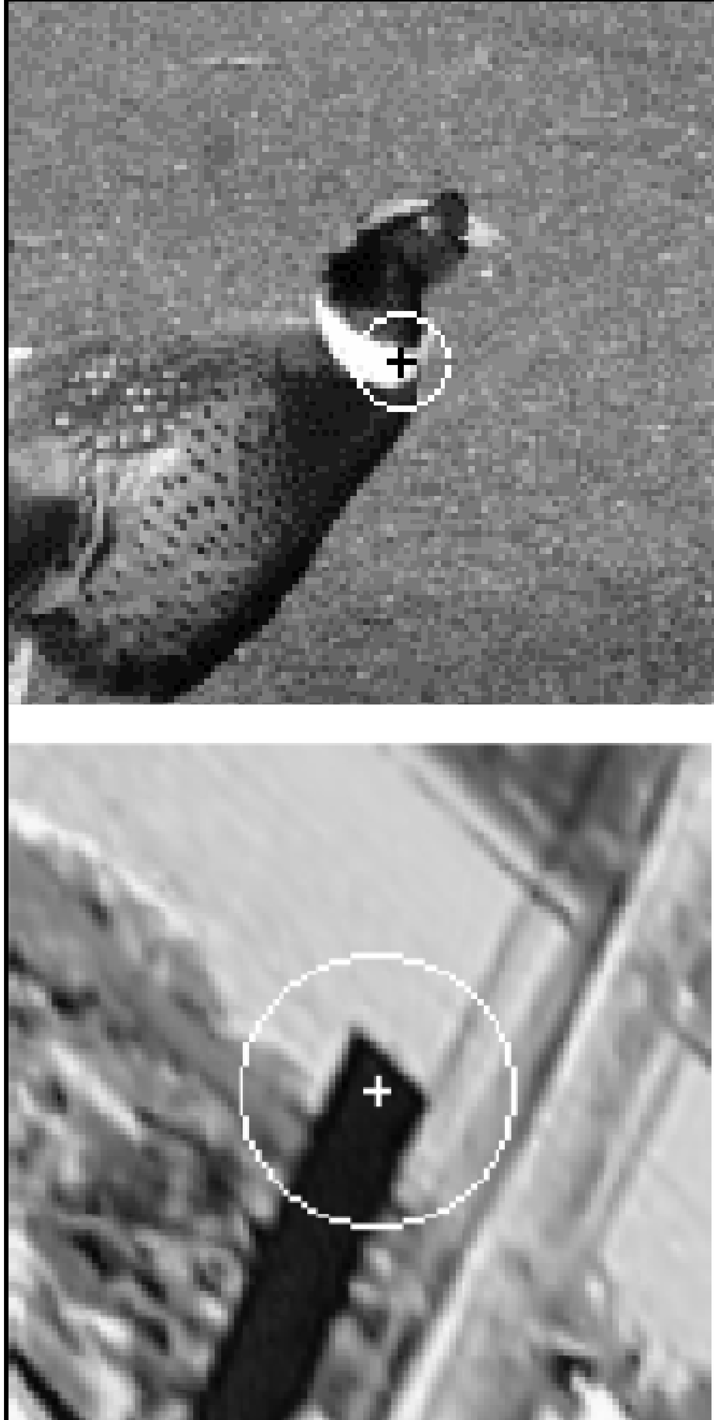
$$G_{k\sigma} - G_\sigma$$

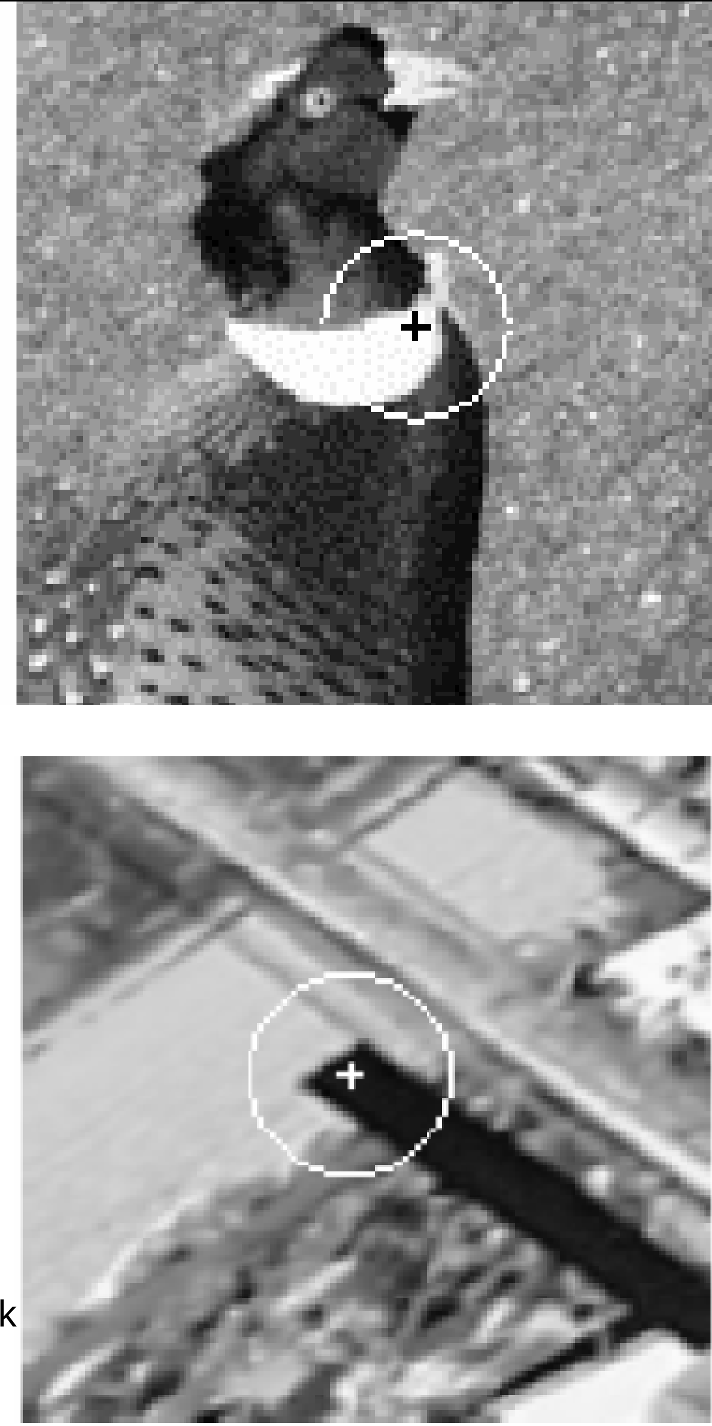


The Laplacian of a Gaussian can be approximated by the difference of two Gaussian.

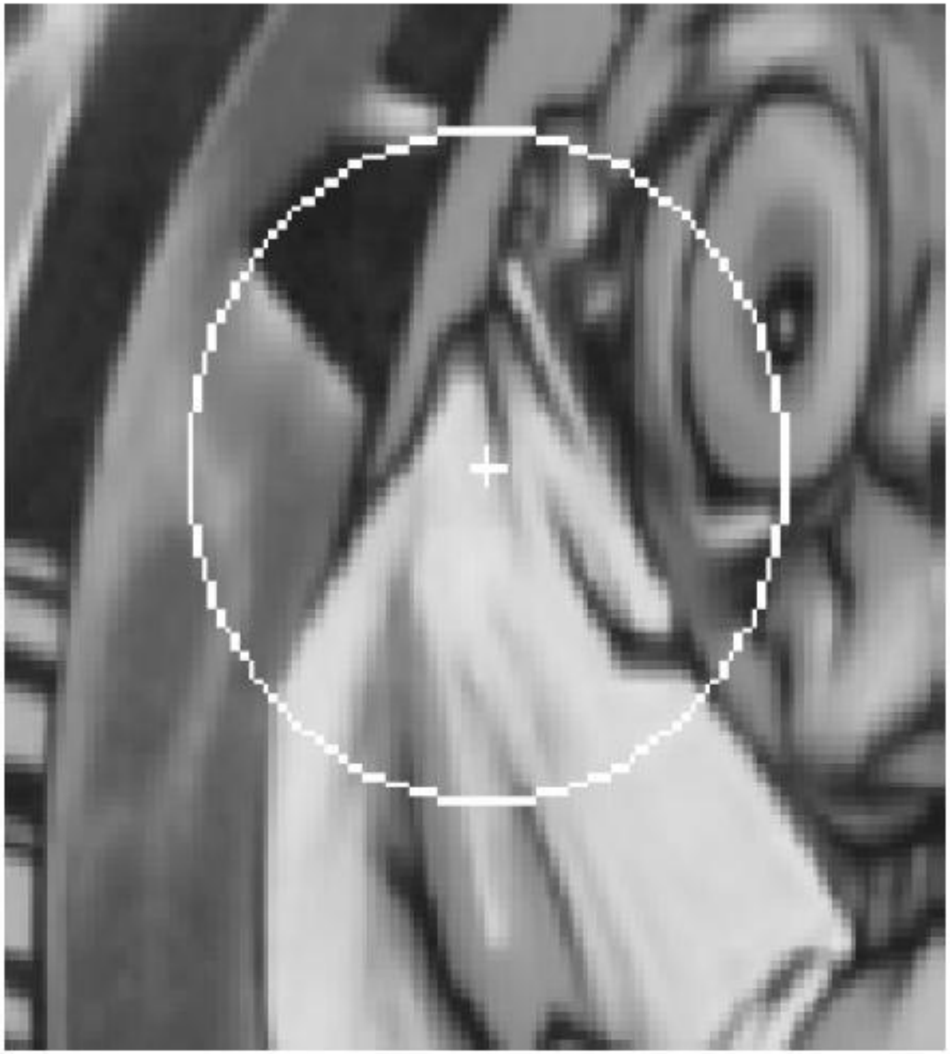
To compare the Laplacian at different scales we need to explain more carefully what the approximation is.

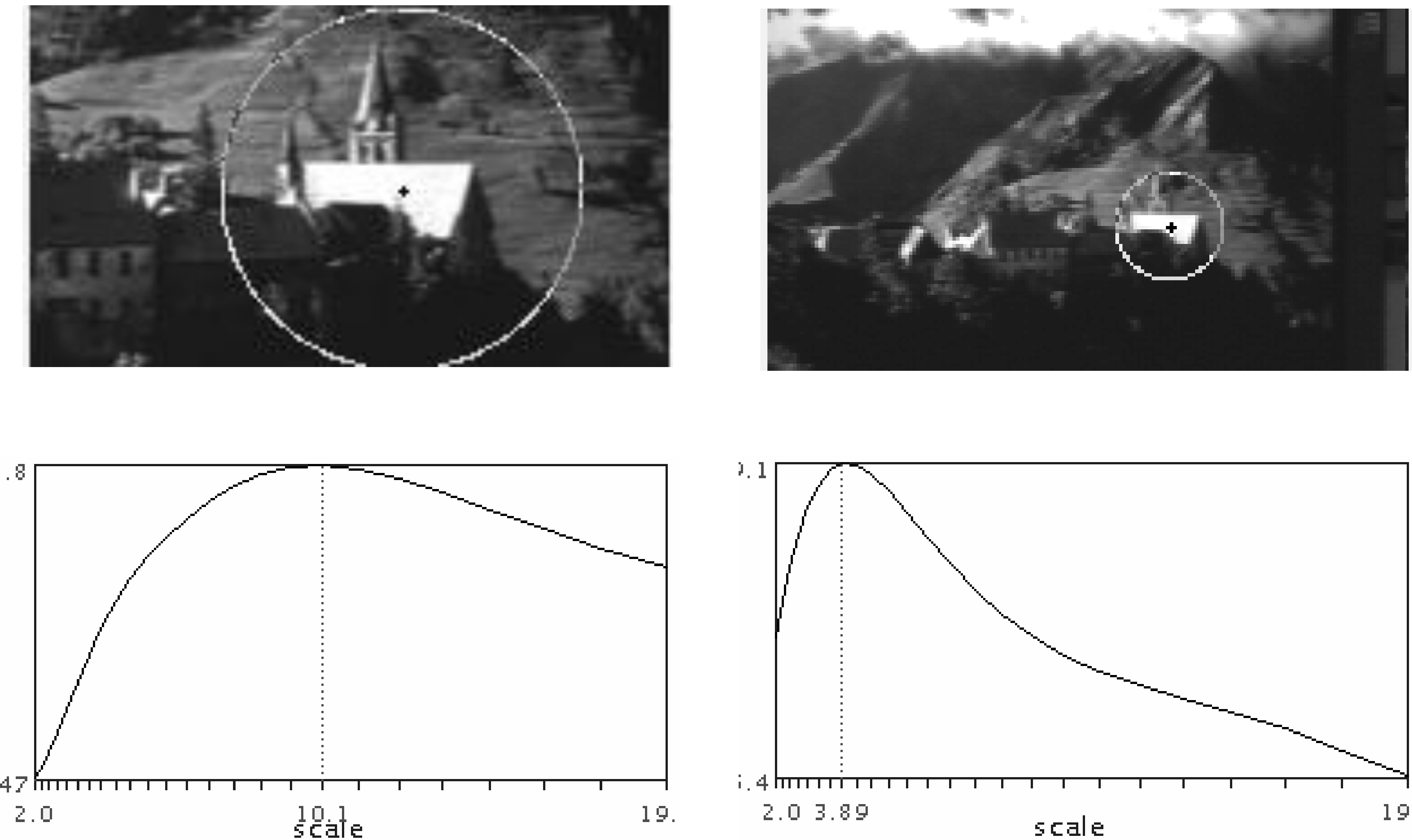
In practice: the scaled Laplacian can be computed by taking the difference between level in a Gaussian pyramid





(From Mikolajczyk and Schmid'02)

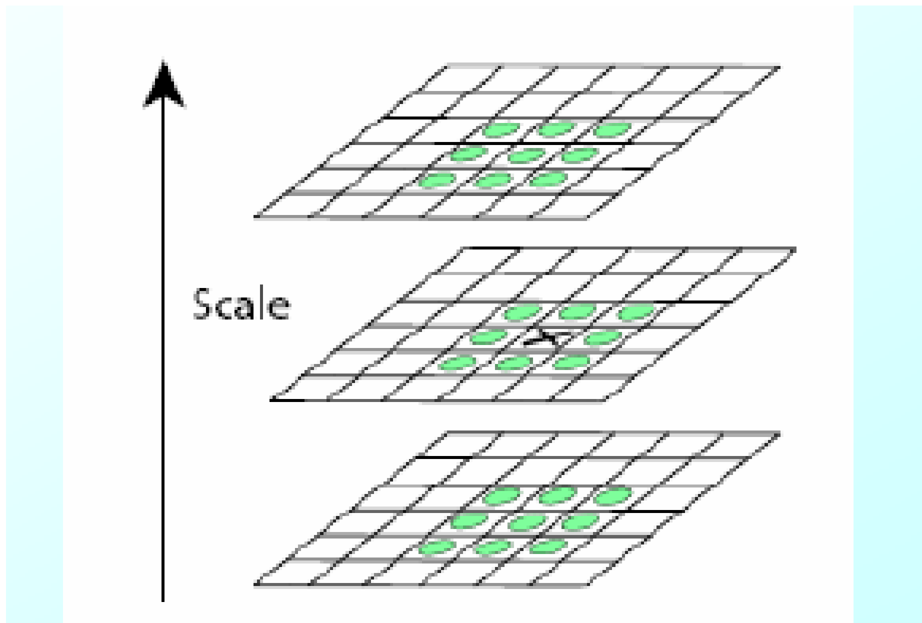


Laplacian

Interest points as “blob-like”: Basic Laplacian detector = Local maxima of characteristic scale

From Laplacian pyramid to detected points

- Find local extrema of $\nabla^2 G_\sigma$ over x,y, and σ (scale)



Combining the two: Harris-Laplace detector

Combining Harris and Laplace: Harris-Laplace detector

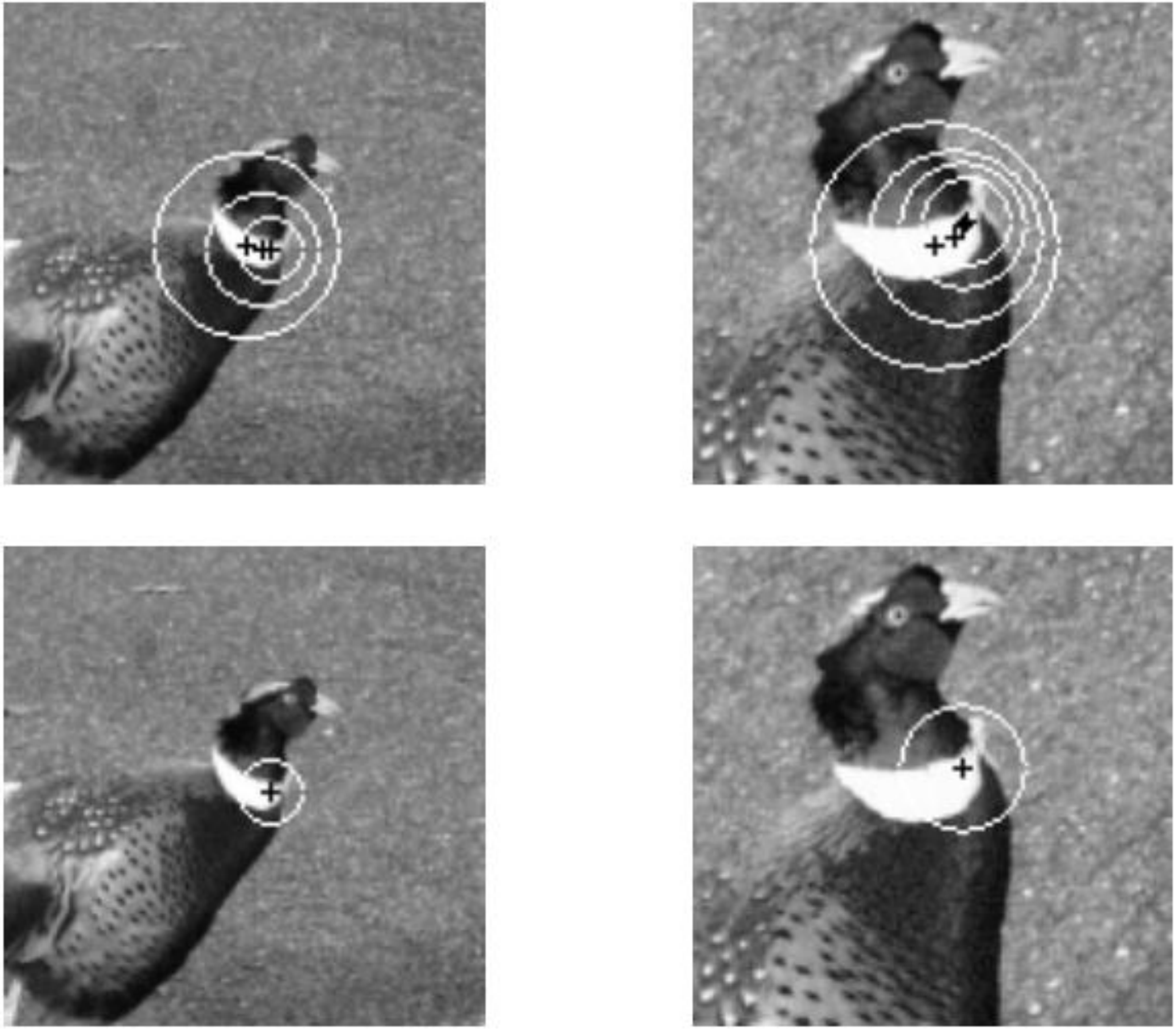
$$H = G_{\sigma_I} * \begin{bmatrix} I^2_{x,\sigma_D} & I_{x,\sigma_D} I_{y,\sigma_D} \\ I_{x,\sigma_D} I_{y,\sigma_D} & I^2_{y,\sigma_D} \end{bmatrix} \quad R = Det(H) - k(Trace(H))^2$$

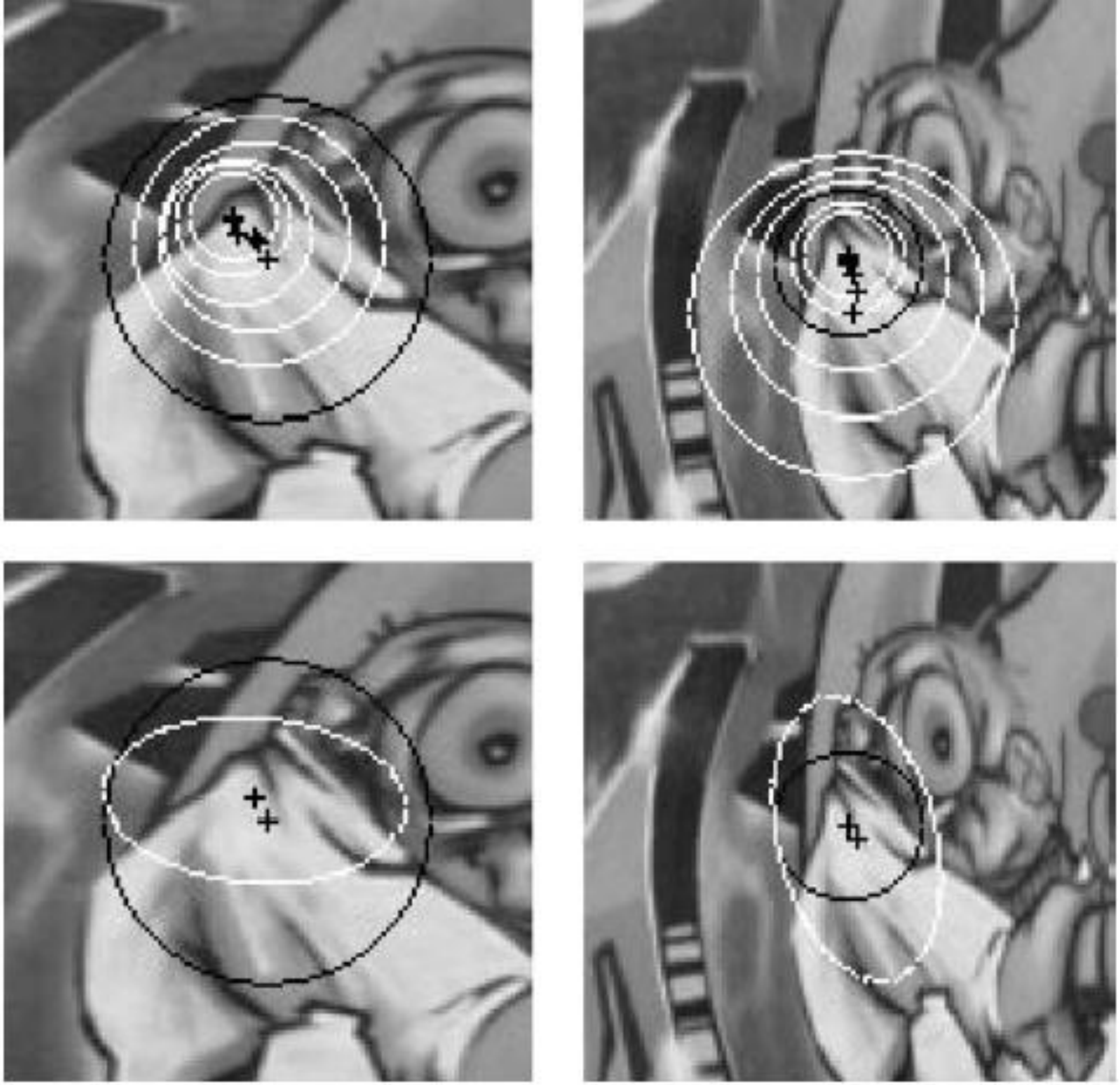
- The location *x* of the points detected by Harris is not scale invariant → Depends on the choice of σ_I and σ_D.
- Reduce to one parameter: $\sigma_D = s\sigma_I$ (s = 0.7)
- The Laplacian trick gives us a good σ but not where the interest point is.
- Chicken and egg problem:
  - If we knew *x* we could estimate σ
  - If we knew σ we could find *x*

Harris-Laplace: Algorithm *summary*

1. Try Harris at different scales and report initial points and associated scales
   - Try different σ_I of the form $k^n \sigma_o$ $(k = \sqrt{2})$
   - Report points with large *R*
2. For each detected point *x*
   - Estimate characteristic scale σ_c as maximum of
     $$\sigma^2 \nabla^2 \boldsymbol{G}_\sigma$$
   - Find the point *x'* with the maximum of *R* in a 8x8 neighborhood of x by using the new scale $\sigma_I = \sigma_c$
   - Replace *x* by *x'*

Iterate until convergence





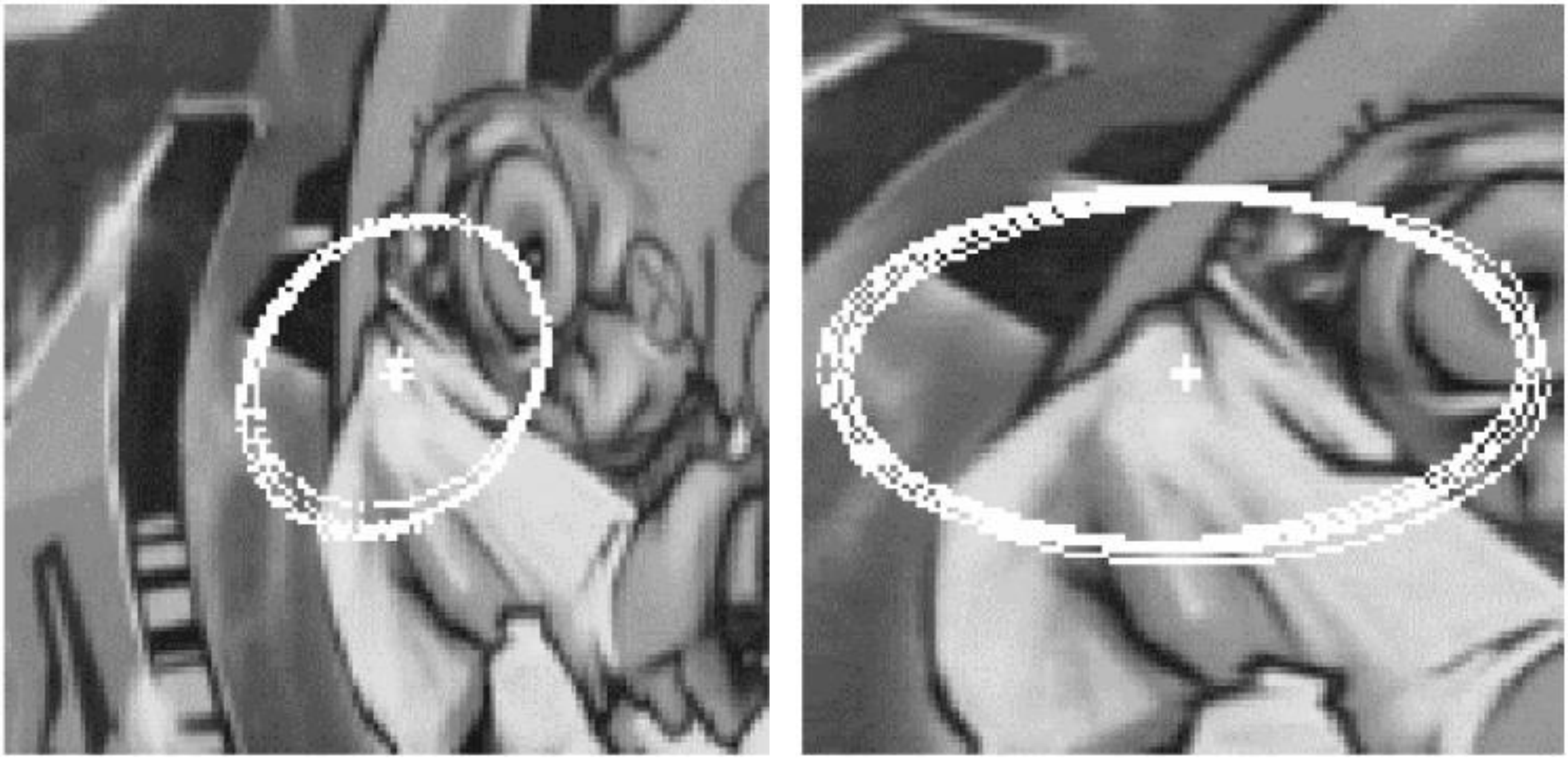
Example from *Mikolajczyk and Schmid 2004*

Affine-invariant detection (*overview only*)

- Need to define a richer description of "neighborhood" or scale
- Use directional derivatives instead: σ_I replaced by Σ_I, σ_D replaced by Σ_D
- Σ_I represent an elliptical "neighborhood" instead of a circular one
- More degrees of freedom to search through but conceptually similar algorithm:

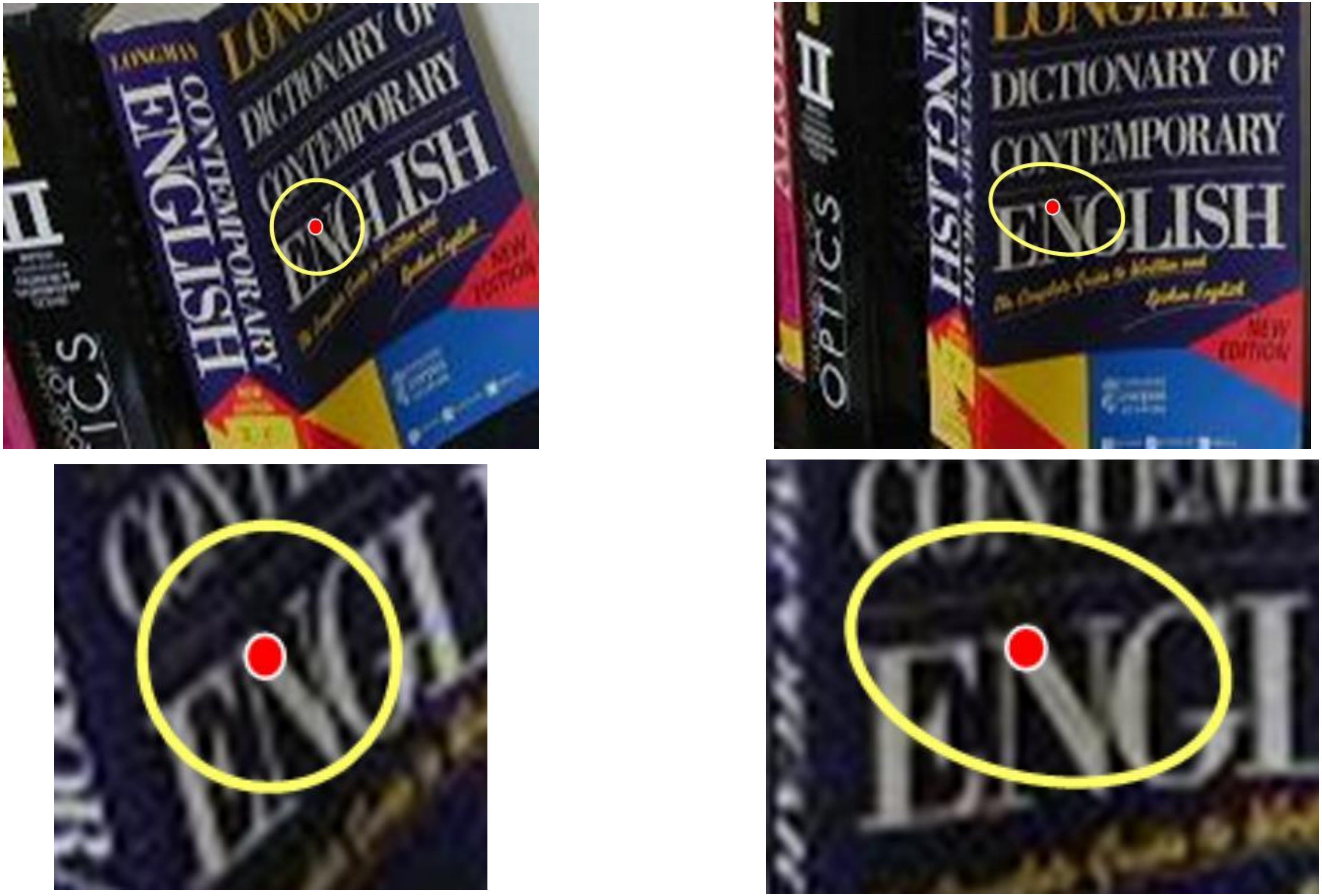
- Assume $\Sigma_D = s\Sigma_I$
- Find x's with initial selections of Σ'_I
- Iterate:
  - Re-estimate "scale" Σ_I
  - Adjust the location of x based on new Σ_I

State of the Art: Affine Invariance

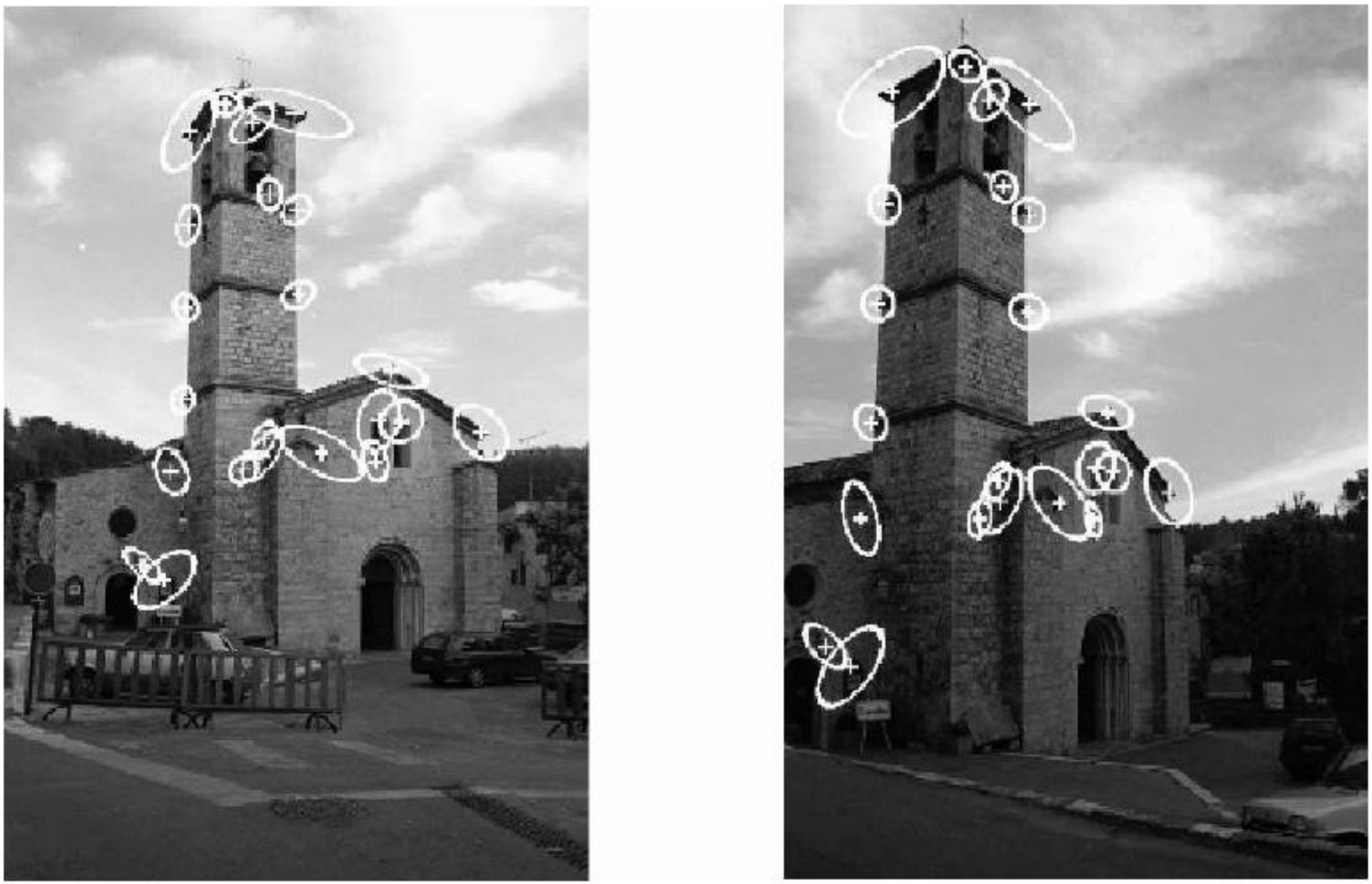


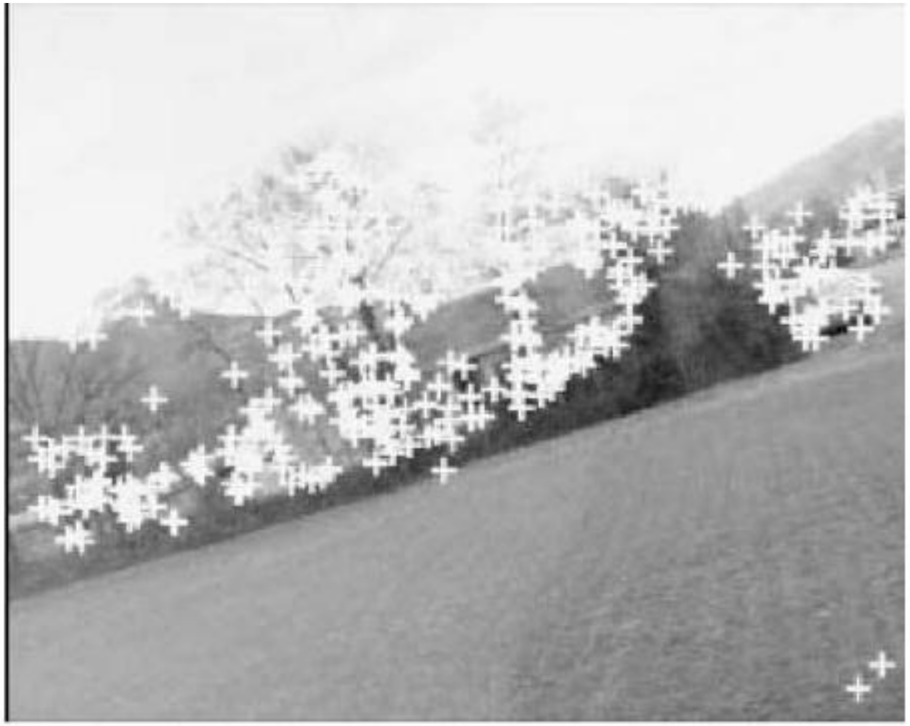
Example from *Mikolajczyk and Schmid 2004*

Affine Invariance



Application: Finding Correspondences





Initial detections
Scale: 4.9
Rotation: 19^{o}

Example from *Mikolajczyk and Schmid 2004*

Final matches: 32 correct correspondences
Scale: 4.9
Rotation: 19º

Example from *Mikolajczyk and Schmid 2004*

Application: Finding Correspondences

Scale change: 1.7

Viewpoint change: 50^o

Example from *Mikolajczyk and Schmid 2004*