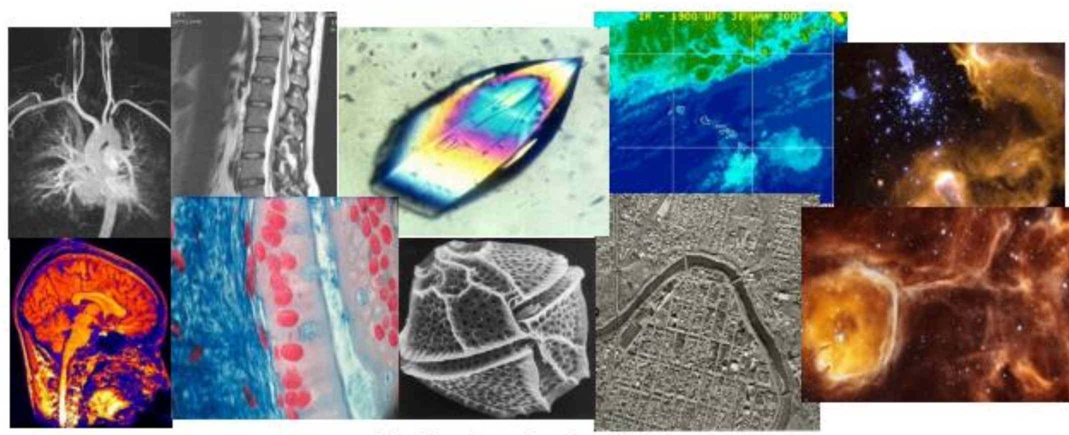
Image Features

XUEJIN CHEN

陈雪锦

- Vision is useful: Images and video are everywhere!

Surveillance and security

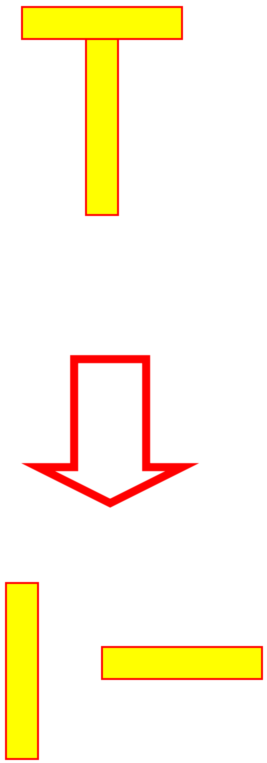


Medical and scientific images

How to UNDERSTAND?

COMPARE with
WHAT we learn before?

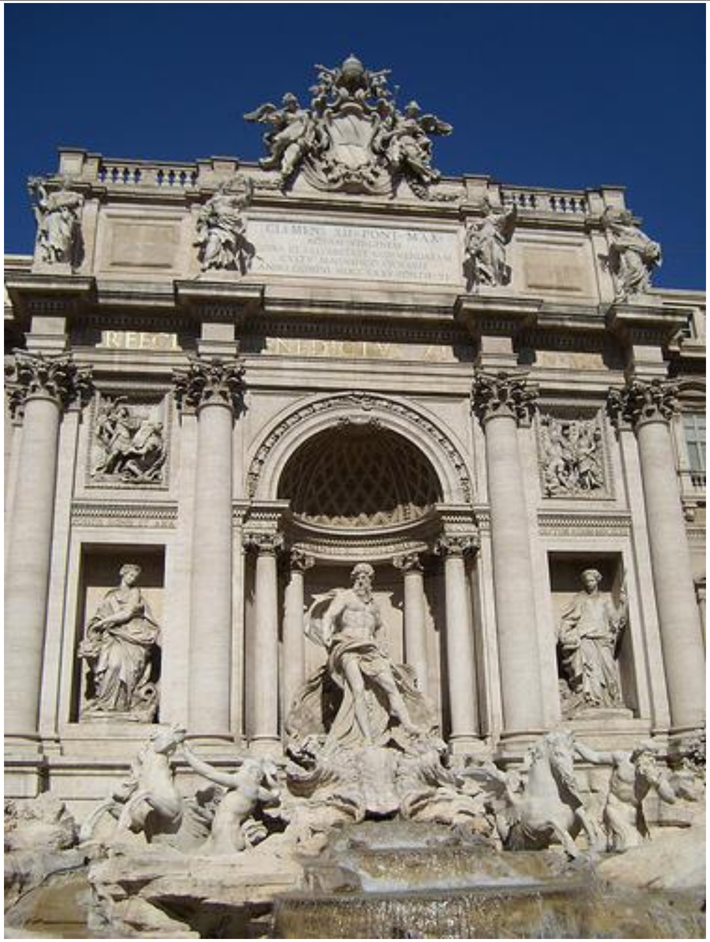
Visual Features



Shape, color, texture, etc

Image Matching

by Diva Sian



by swashford

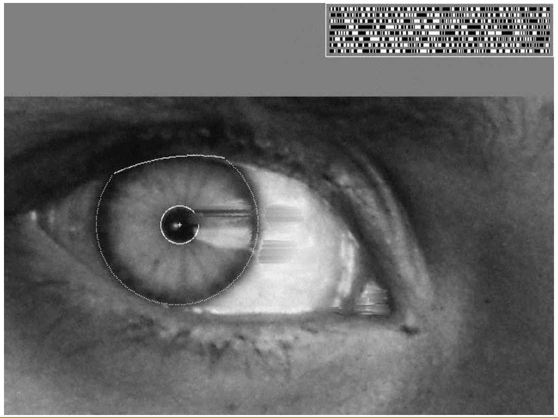
Harder Case

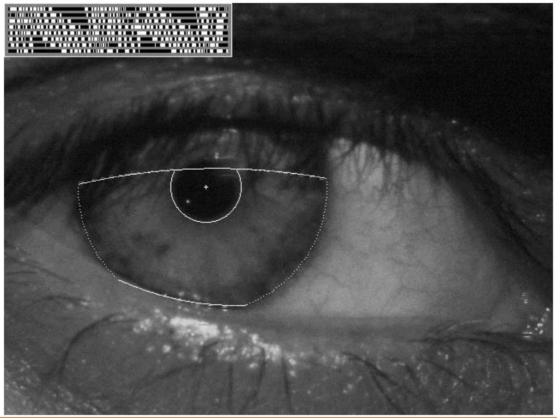
by Diva Sian

by scgbt

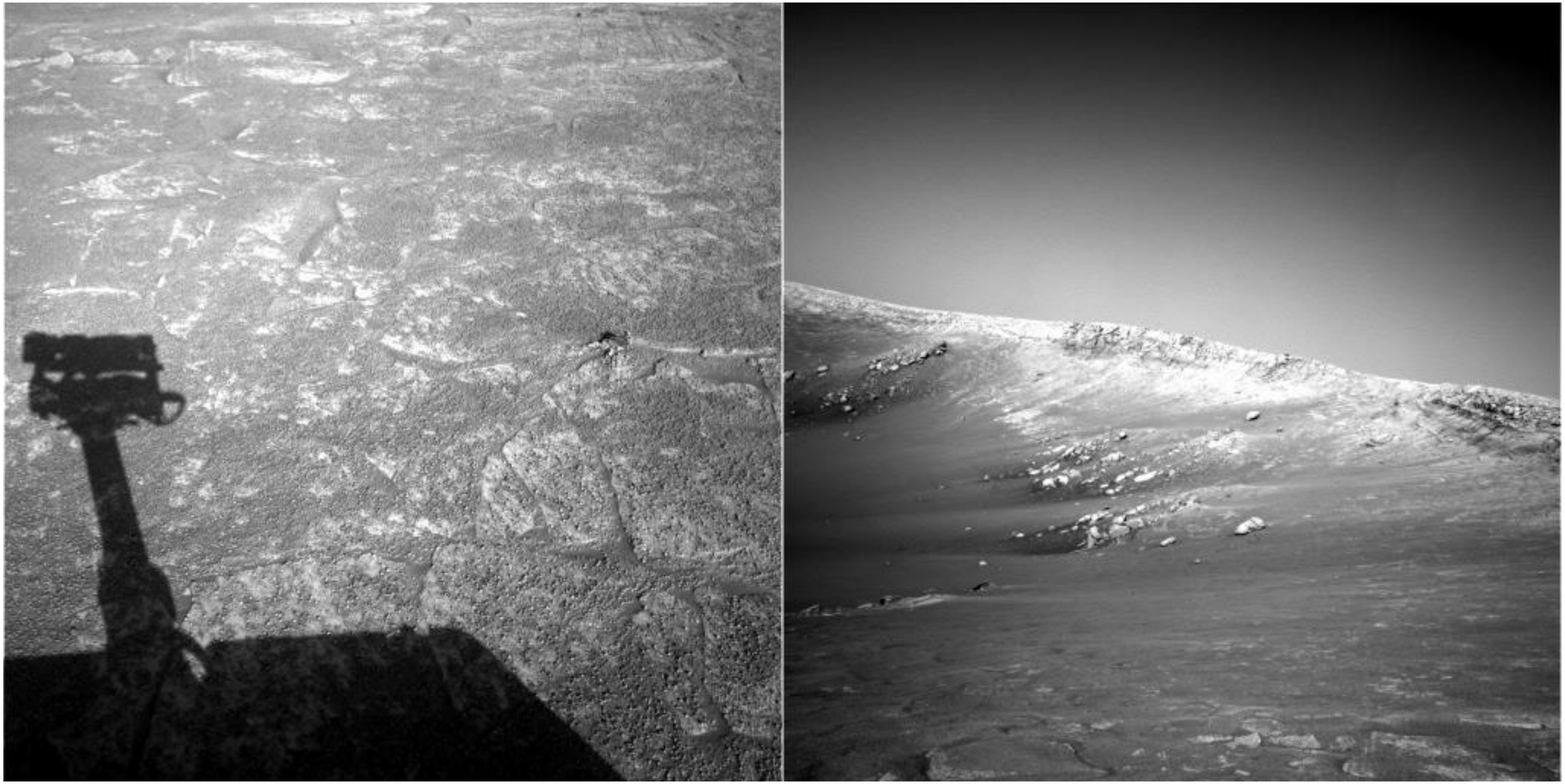
Even Harder Case

"*How the Afghan Girl was Identified by Her Iris Patterns*" Read the [story]



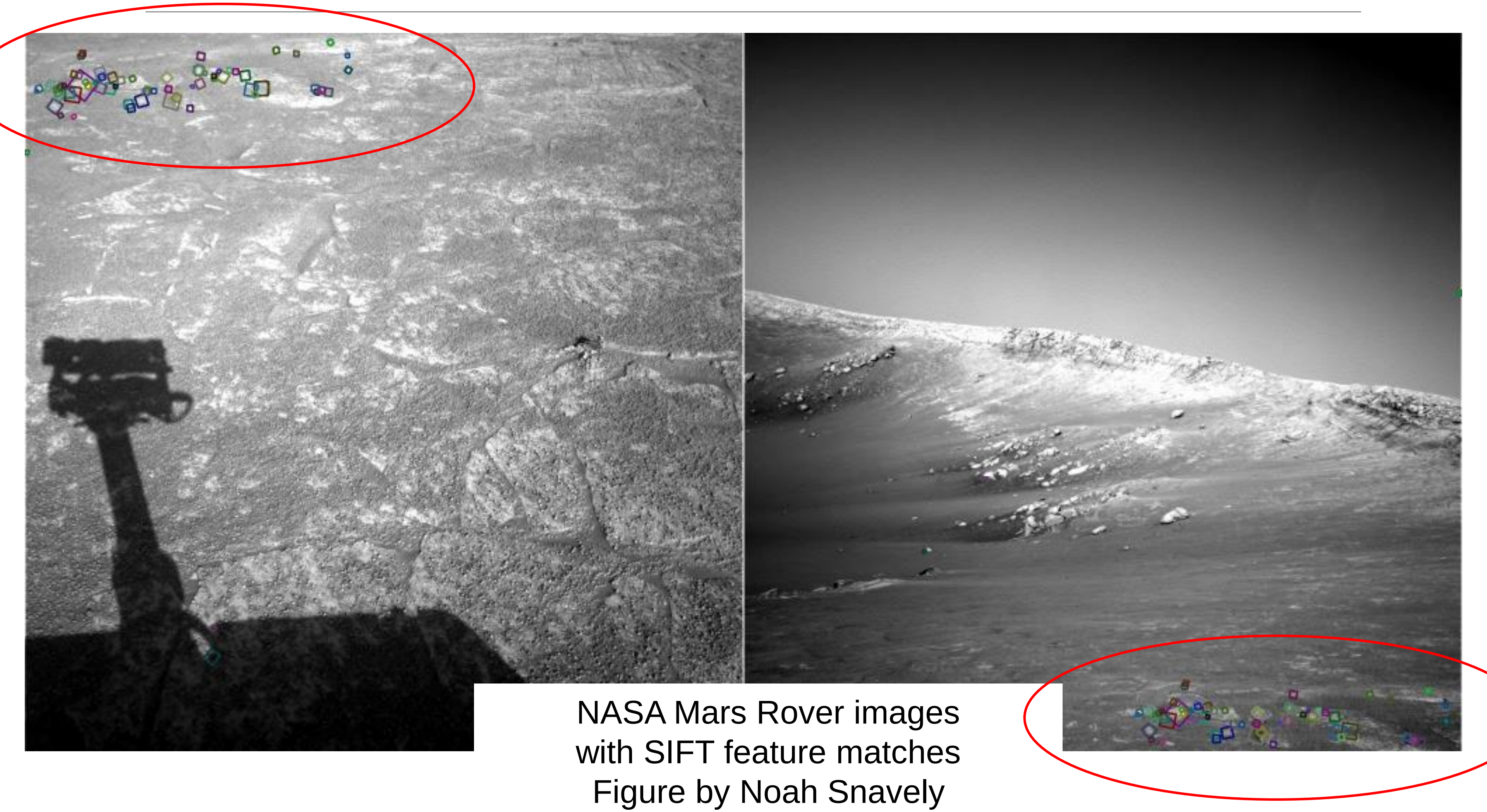


Harder still?



NASA Mars Rover images

Answer below (look for tiny colored squares…)



NASA Mars Rover images
with SIFT feature matches
Figure by Noah Snavely

Features

All is Vanity, by C. Allan Gilbert, 1873-1929

Image Matching

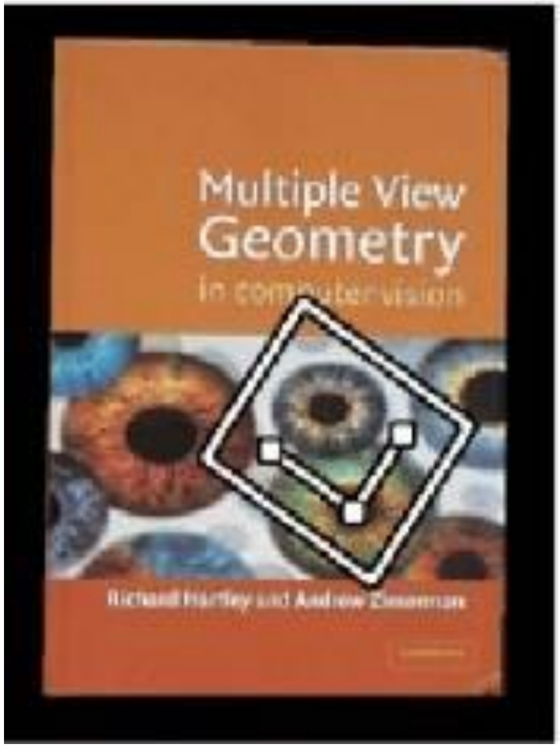
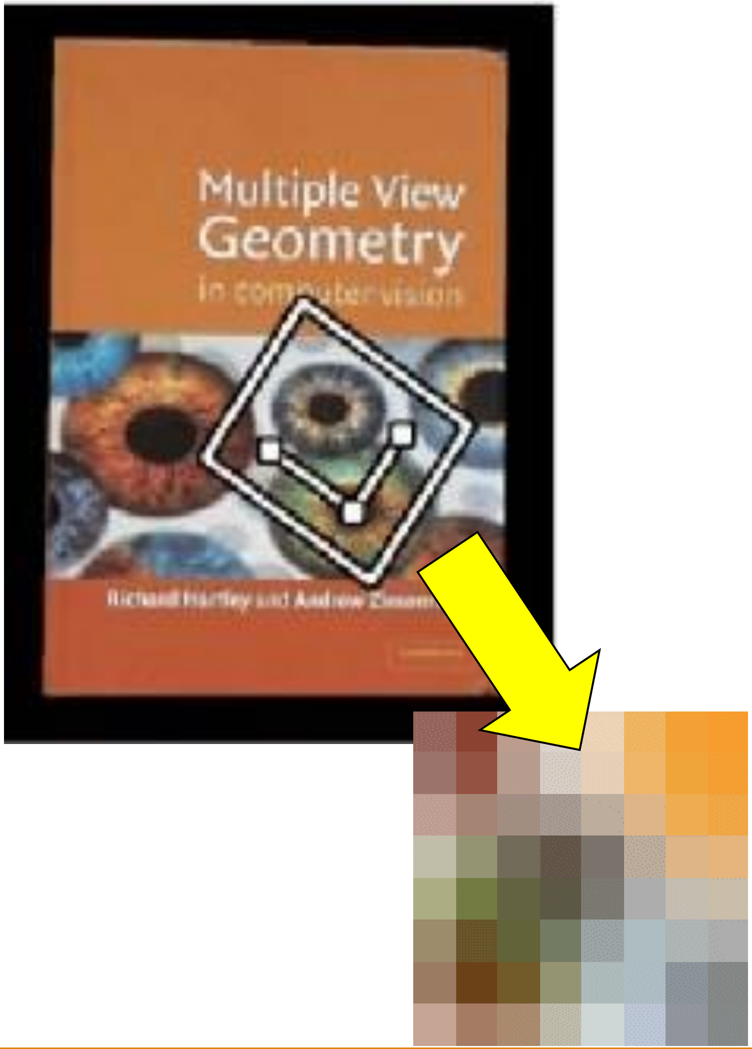


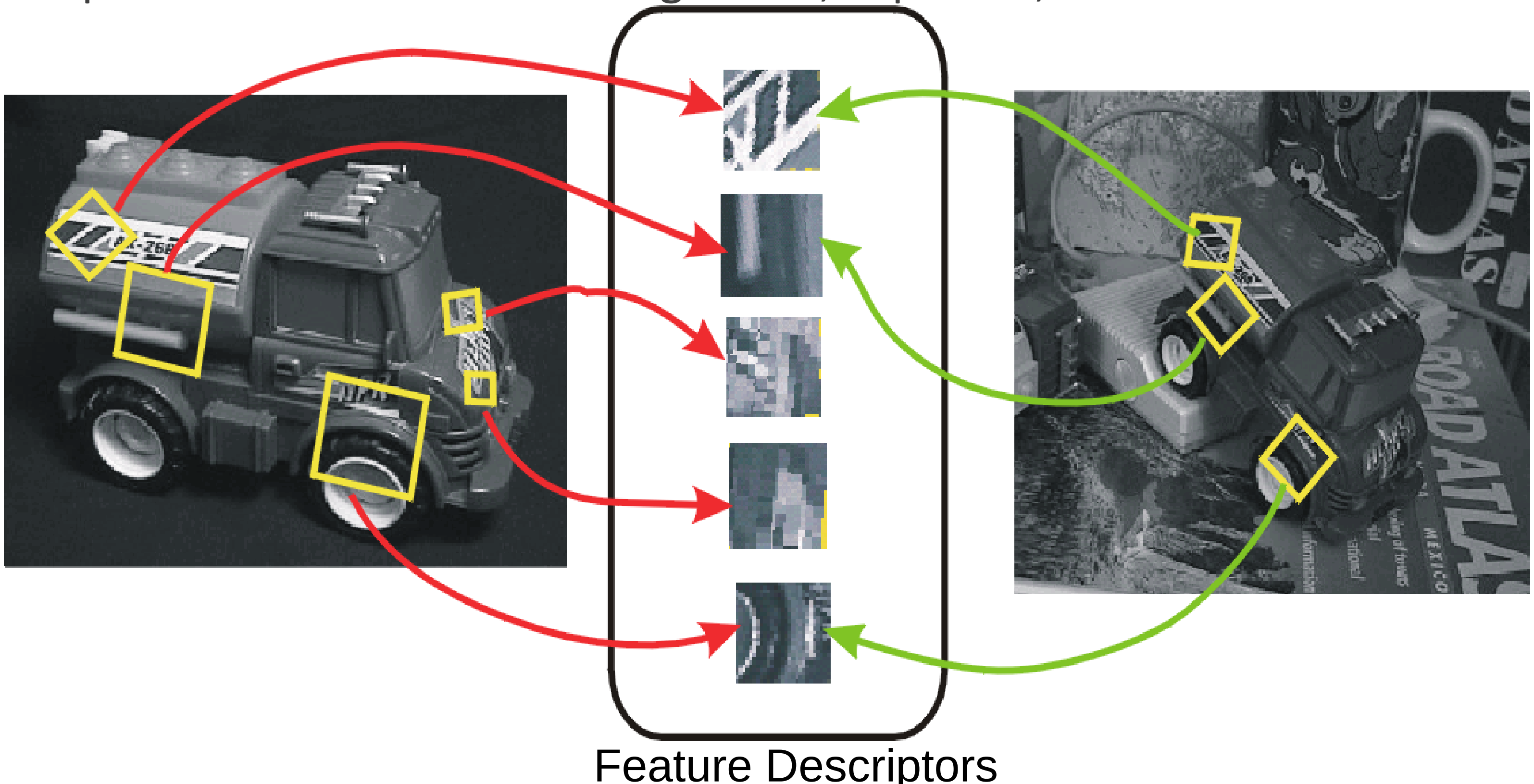
Image Matching



Invariant Local Features

Find features that are invariant to transformations

- geometric invariance: translation, rotation, scale
- photometric invariance: brightness, exposure, …



Advantages of Local Features

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

Distinctiveness:
- can differentiate a large database of objects

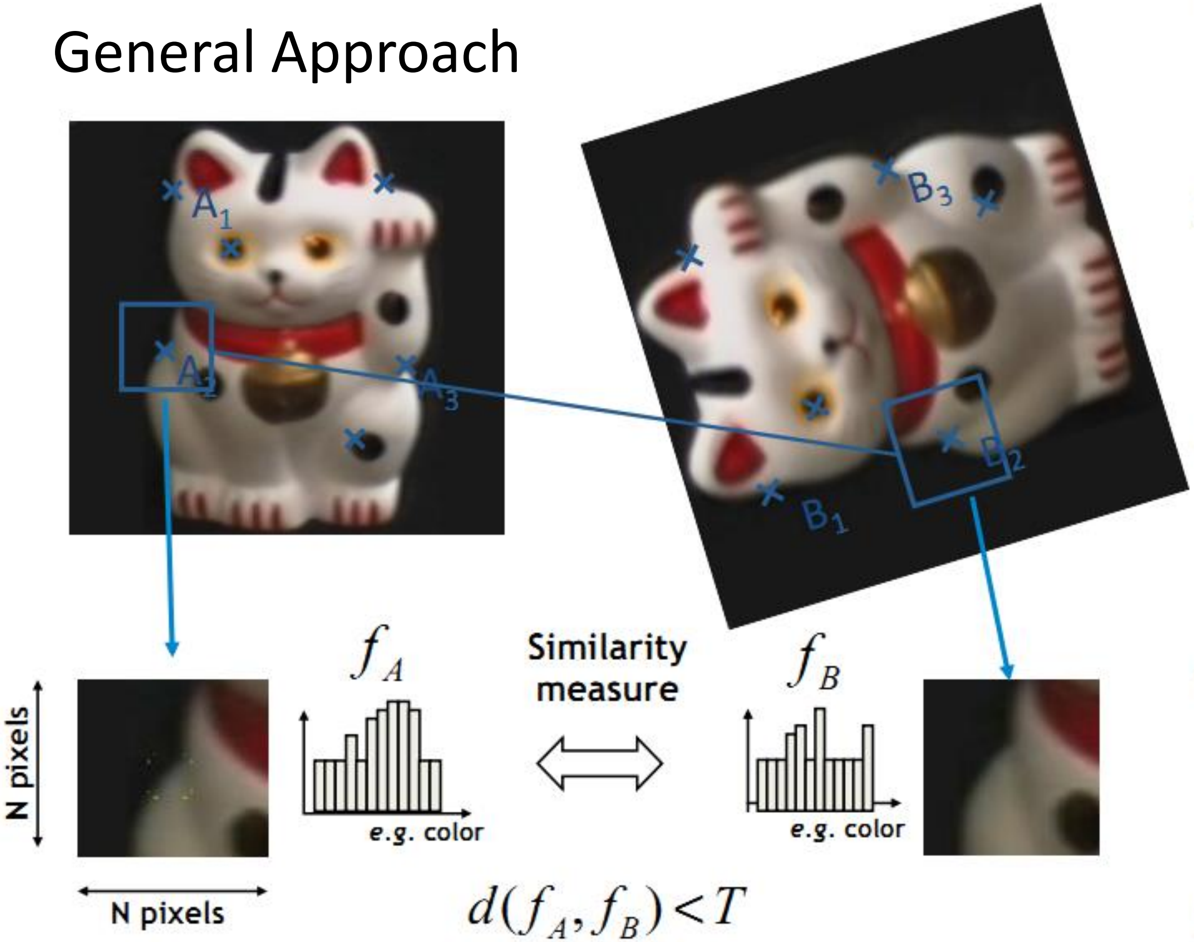
Quantity
- hundreds or thousands in a single image

Efficiency
- real-time performance achievable

Generality
- exploit different types of features in different situations

Image Matching

General Approach



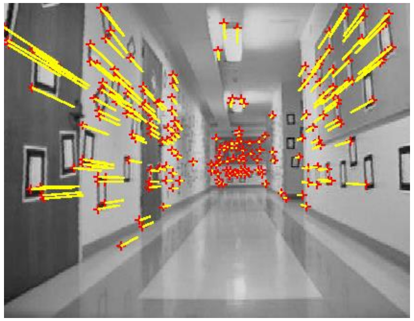
1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

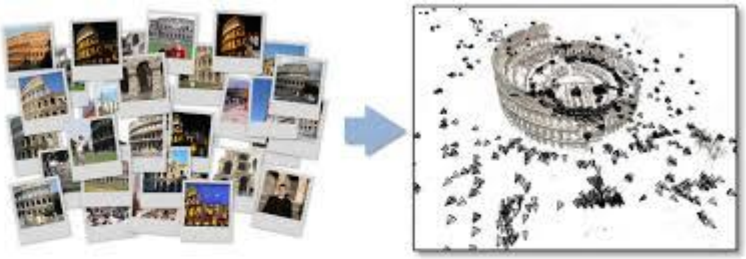
Slide credit: Bastian Leibe

More Motivation…

Feature points are used for:

- Image alignment (e.g., mosaics)
- 3D reconstruction
- Motion tracking
- Object recognition
- Indexing and database retrieval
- Robot navigation
- … other





Features

Point/patch,

Edge/curve

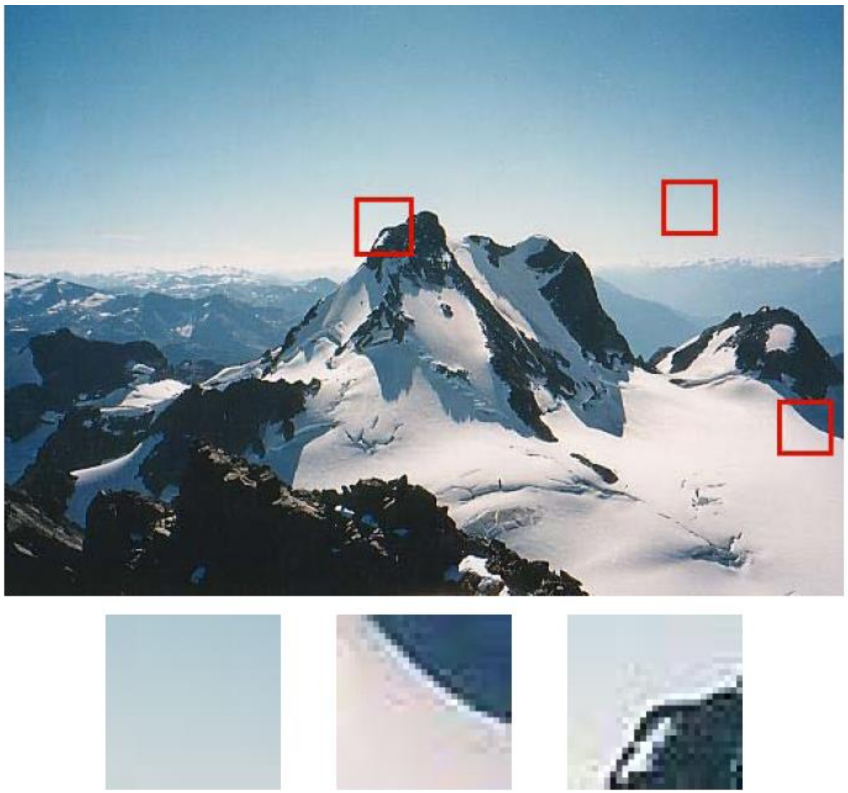
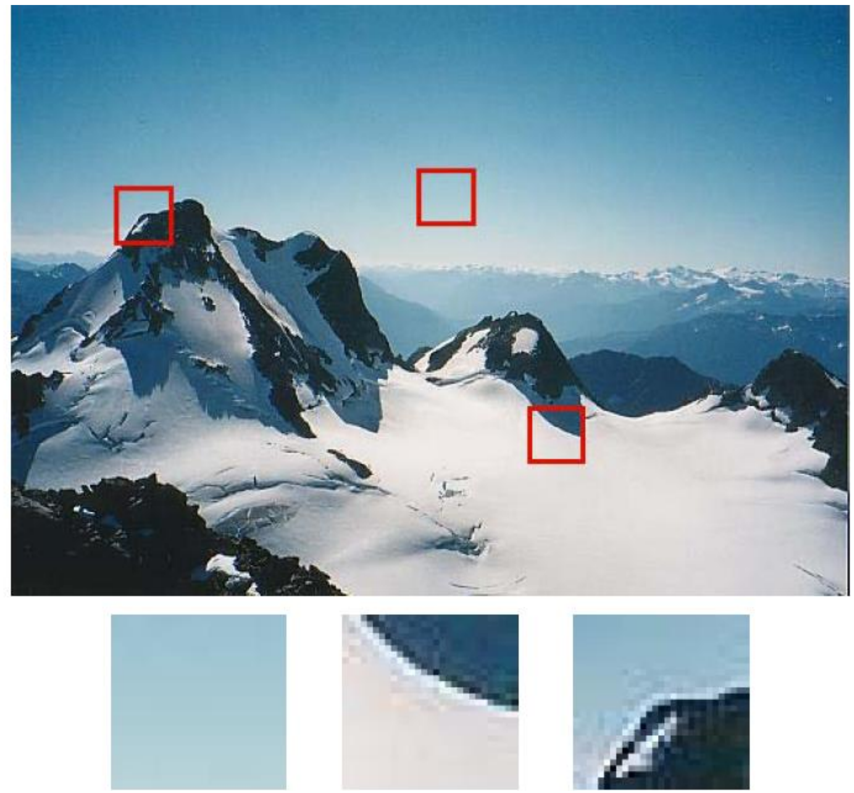
Region

Want Uniqueness

Image regions that are unusual

- Lead to unambiguous matches in other images

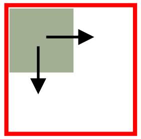
How to define “unusual”?

Corner Detection

Basic idea: Find points where two edges meet—i.e., high gradient in two directions

“Cornerness” is undefined at a single pixel, because there’s only one gradient per point
- Look at the gradient behavior over a small window

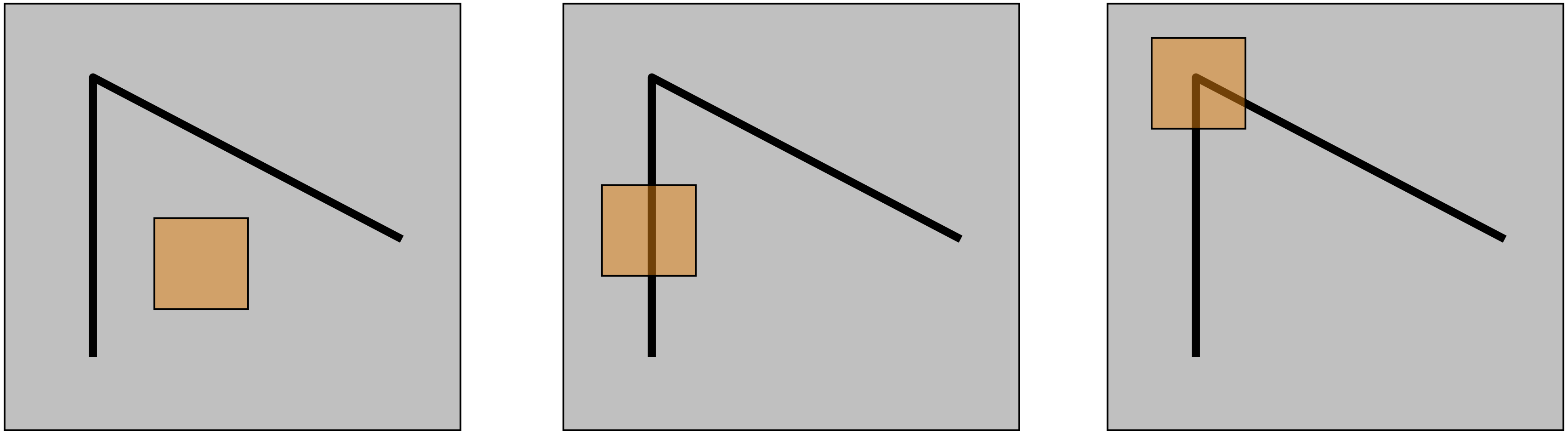


Categories image windows based on gradient statistics
- **Constant**: Little or no brightness change
- **Edge**: Strong brightness change in single direction
- **Flow**: Parallel stripes
- **Corner/spot**: Strong brightness changes in orthogonal directions

Local Measures of Uniqueness

Suppose we only consider a small window of pixels

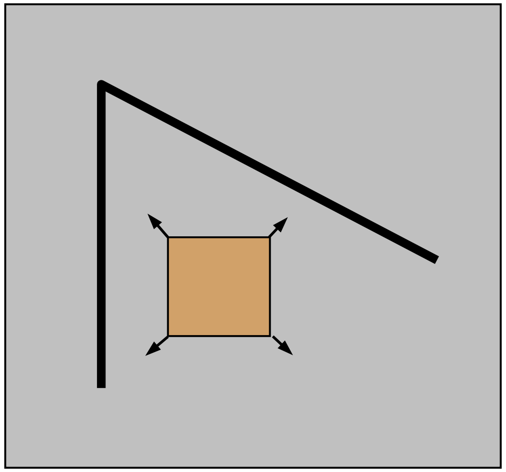
- What defines whether a feature is a good or bad candidate?

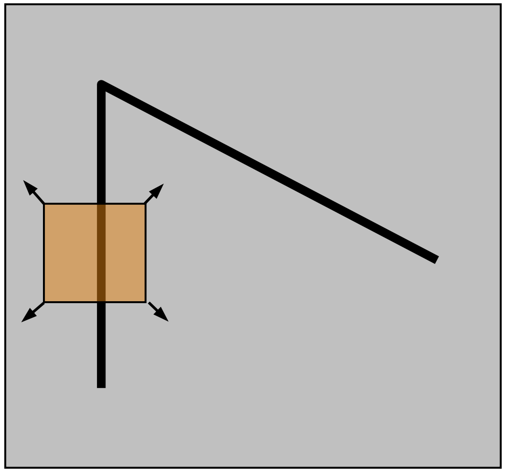


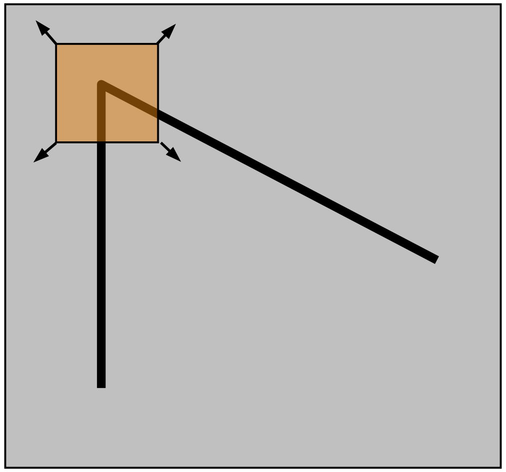
Feature Detection

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

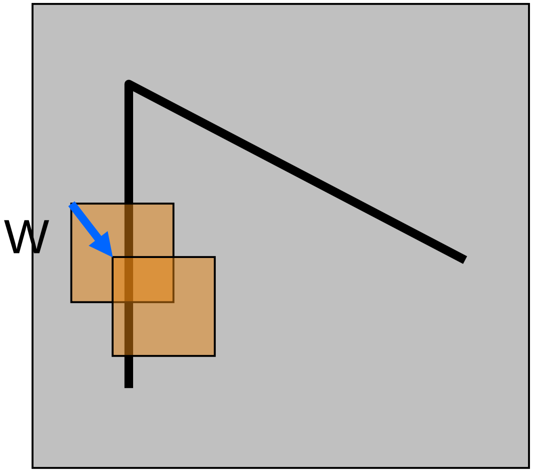
"corner":
significant change in all directions

Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.

Feature Detection: Mathematics

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)*:



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

Harris Detector: Mathematics

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

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

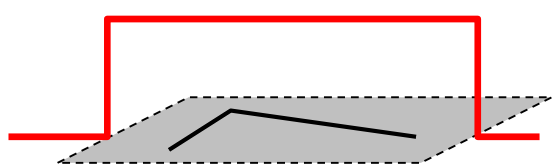
Auto-correlation function

Window function

Shifted intensity

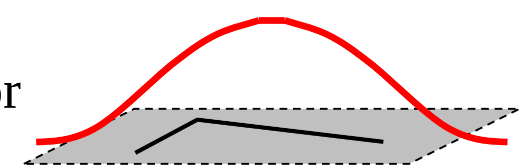
Intensity

Window function $w(x,y)$ =



1 in window, 0 outside

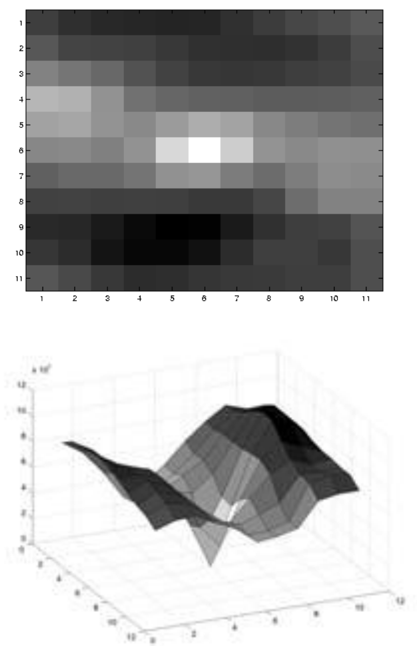
or



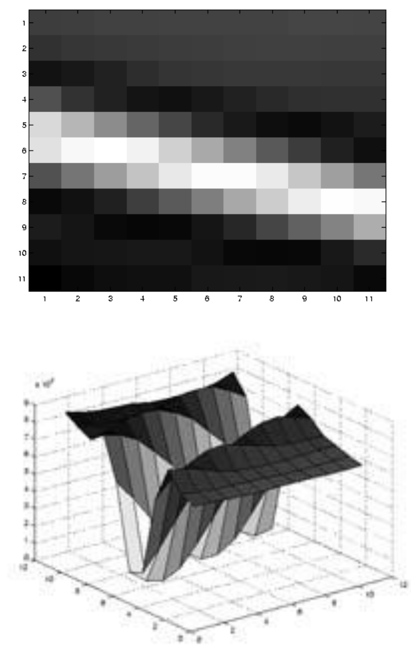
Gaussian

Auto-Correlation Function

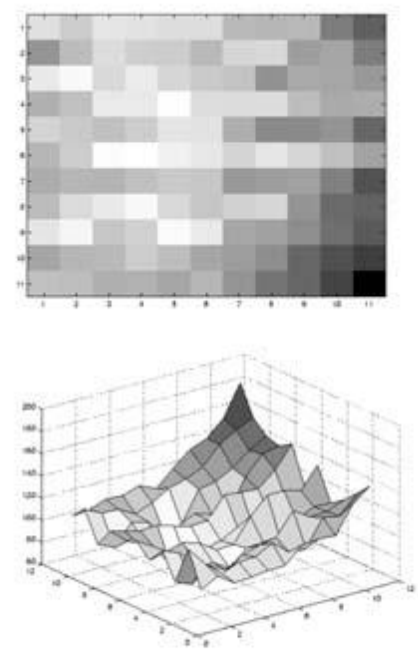
Auto-Correlation Function



Good unique minimum



1D aperture problem



No good peak

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 (u,v) is small, then first order approx is good

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

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

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

Feature Detection: Mathematics

Image gradient $\nabla I_0(\boldsymbol{x}_i)$

- Harris detector with a [-2,-1,0,1,2] filter for Ix
- Gaussian filter

Small Motion Assumption

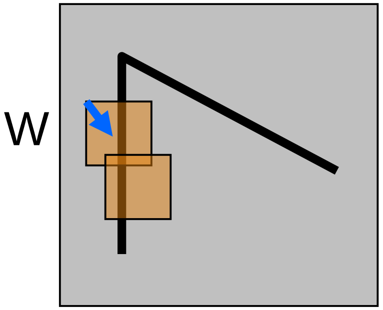
$$I(x+u, y+v) \approx I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

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

Plugging this into the formula on

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

Feature Detection: Mathematics



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

Feature Detection: Mathematics

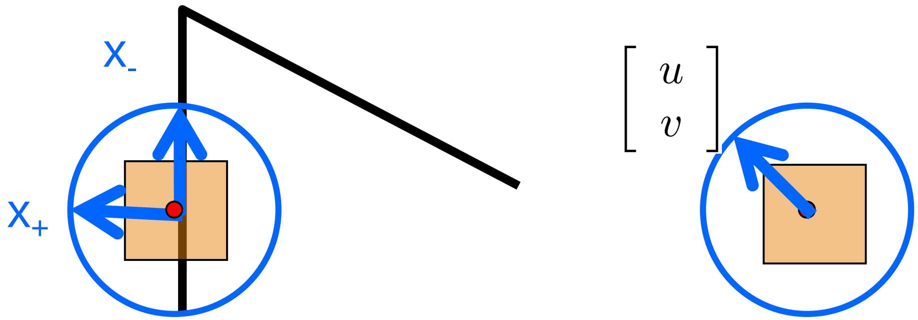
This can be rewritten:

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

$$= (u,v)A\begin{pmatrix} u \\ v \end{pmatrix}$$

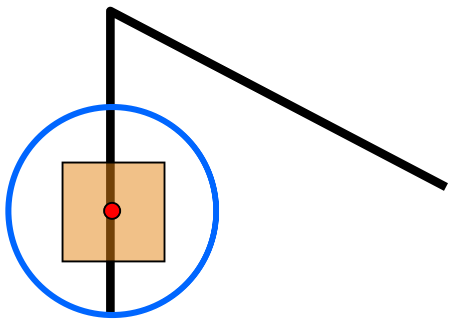
$$A = \sum_i \omega(\mathbf{x}_i) \begin{bmatrix} I_x^2(\mathbf{x}_i) & I_x(\mathbf{x}_i)I_y(\mathbf{x}_i) \\ I_x(\mathbf{x}_i)I_y(\mathbf{x}_i) & I_y^2(\mathbf{x}_i) \end{bmatrix}$$

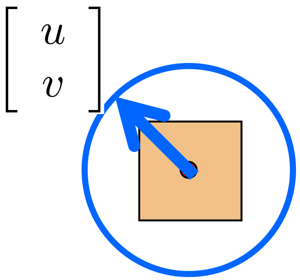
Auto-correlation matrix



Feature Detection: Mathematics

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





For the example above

- You can move the center of the blue 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 *A*

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*** 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:

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

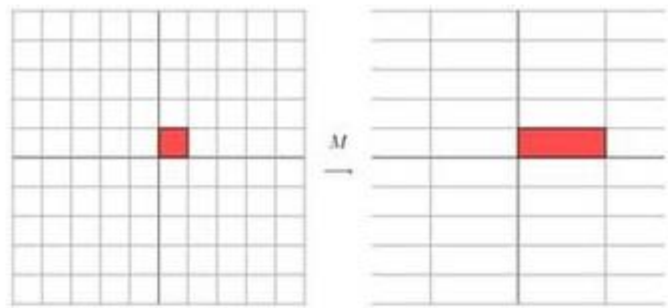
Once you know λ, you find **x** by solving

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

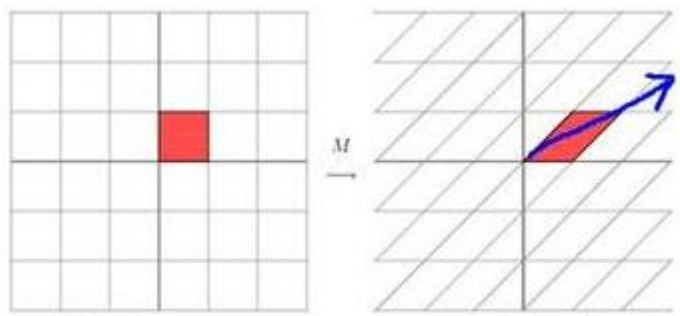
Quick eigenvalue/eigenvector review

A: A linear transformation

$$A = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}$$



$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

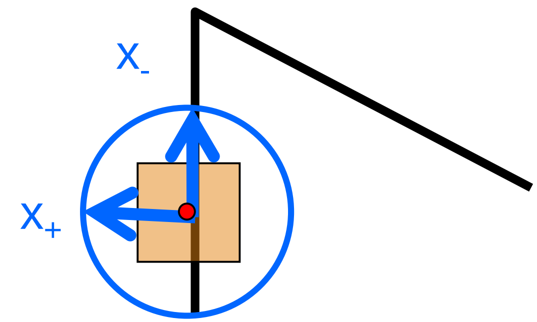


Feature Detection: Mathematics

This can be rewritten:

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

$$A = \sum_i \omega(\mathbf{x}_i) \begin{bmatrix} I_x^2(\mathbf{x}_i) & I_x(\mathbf{x}_i) I_y(\mathbf{x}_i) \\ I_x(\mathbf{x}_i) I_y(\mathbf{x}_i) & I_y^2(\mathbf{x}_i) \end{bmatrix}$$



Eigenvalues and eigenvectors of H = A

- 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_+

$$Ax_+ = \lambda_+ x_+$$
$$Ax_- = \lambda_- x_-$$

$$x_+ A x_+ = \lambda_+$$
$$x_- A x_- = \lambda_-$$

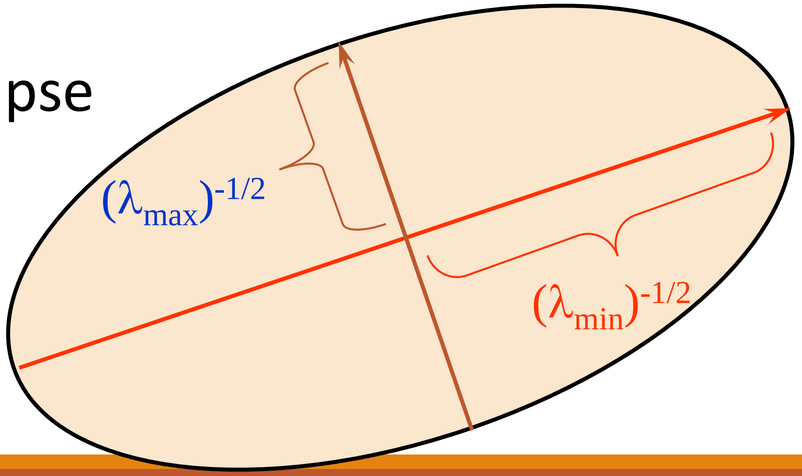
Feature Detection: Mathematics

Intensity change in shifting window: eigenvalue analysis

$$E(u,v) \cong [u,v]A\begin{bmatrix} u \\ v \end{bmatrix} \qquad \lambda_1, \lambda_2 – \text{eigenvalues of } A$$

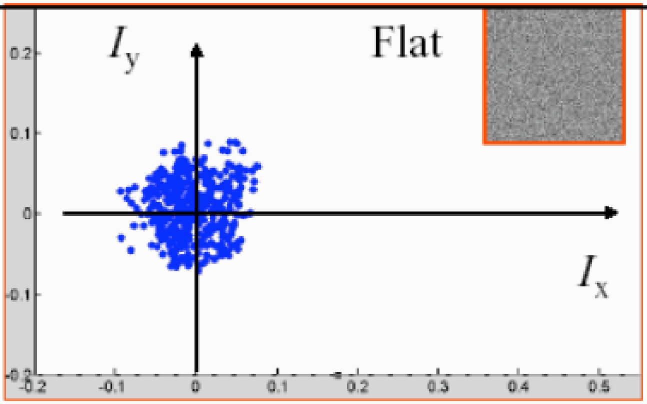
If we try every possible orientation (u,v), the max. change in intensity is λ_{max}

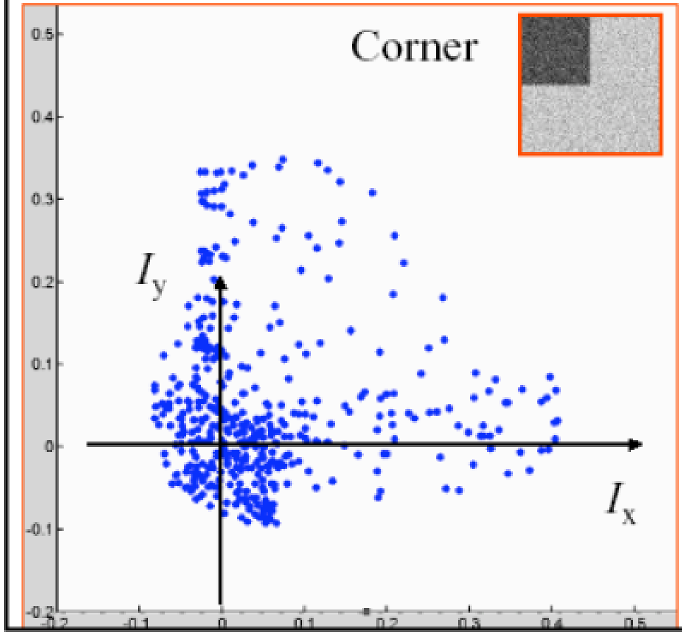
$E(u,v)$= constant on the ellipse

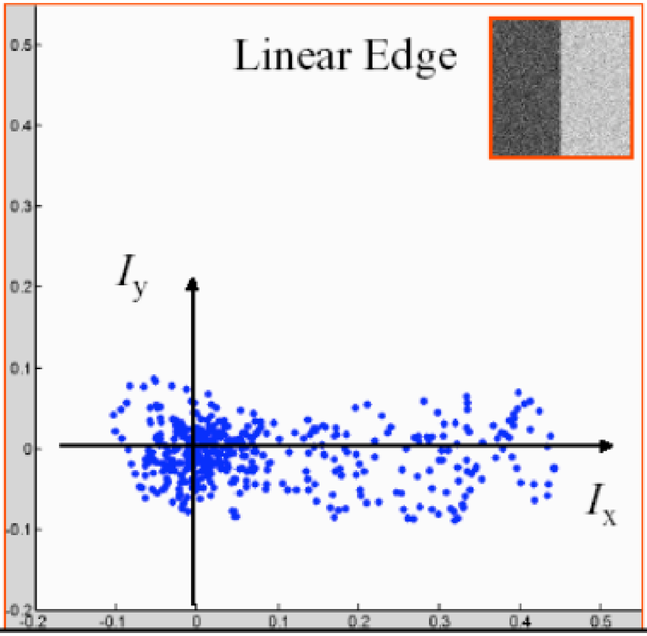


Plotting Derivatives as 2D Points

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



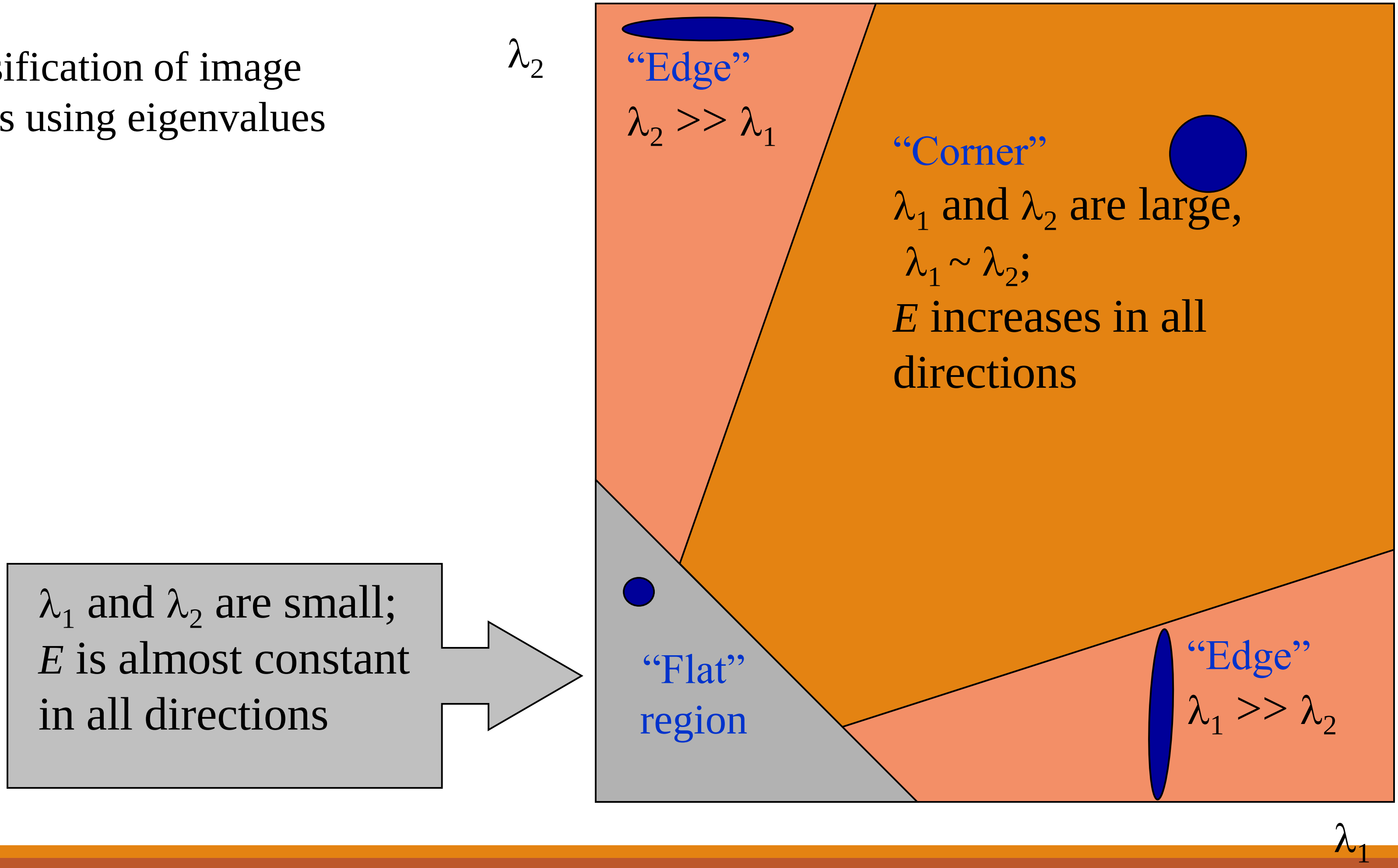




Slide from Robert Collins

Feature Detection: Mathematics

Classification of image points using eigenvalues of *H*:



Feature Detection: Mathematics

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

- What's our feature scoring function?

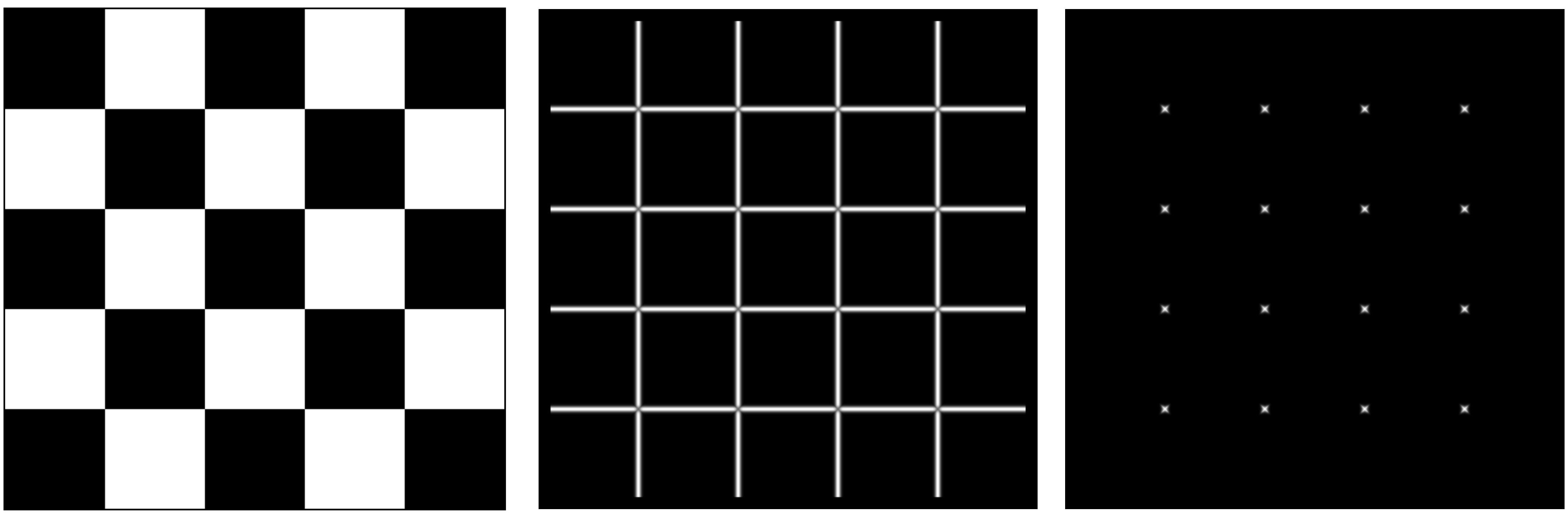
Feature Detection: Mathematics

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 A

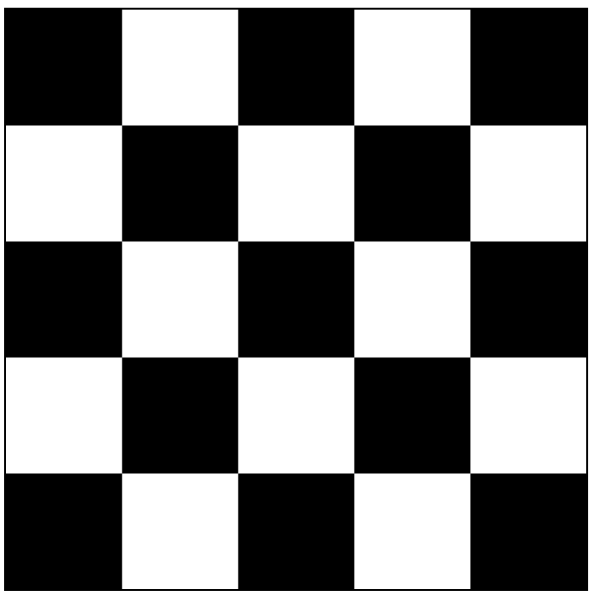


I λ_+ λ_-

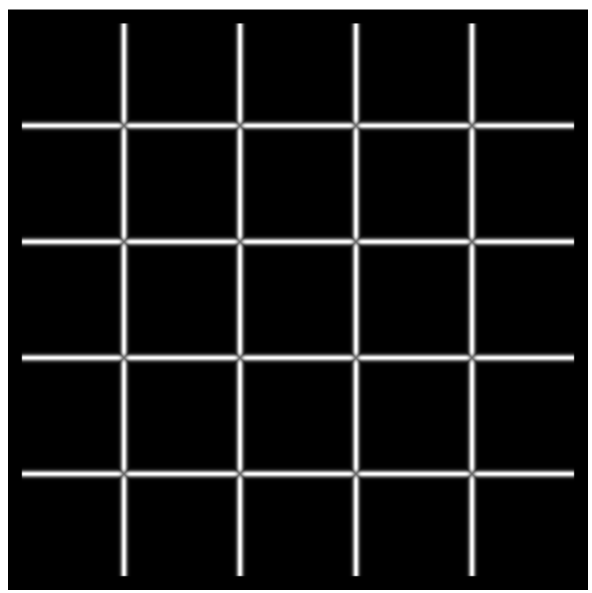
Feature Detection

Here's what you do

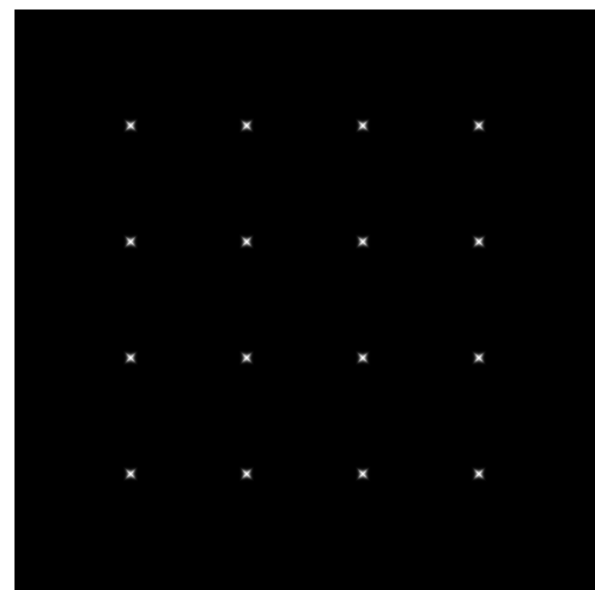
- Compute the gradient at each point in the image
- Create the *A* matrix from the entries in the gradient
- Compute the eigenvalues.
- Find points with large response (λ_- > threshold)
- Choose those points where λ_- is a local maximum as features



I



λ_+

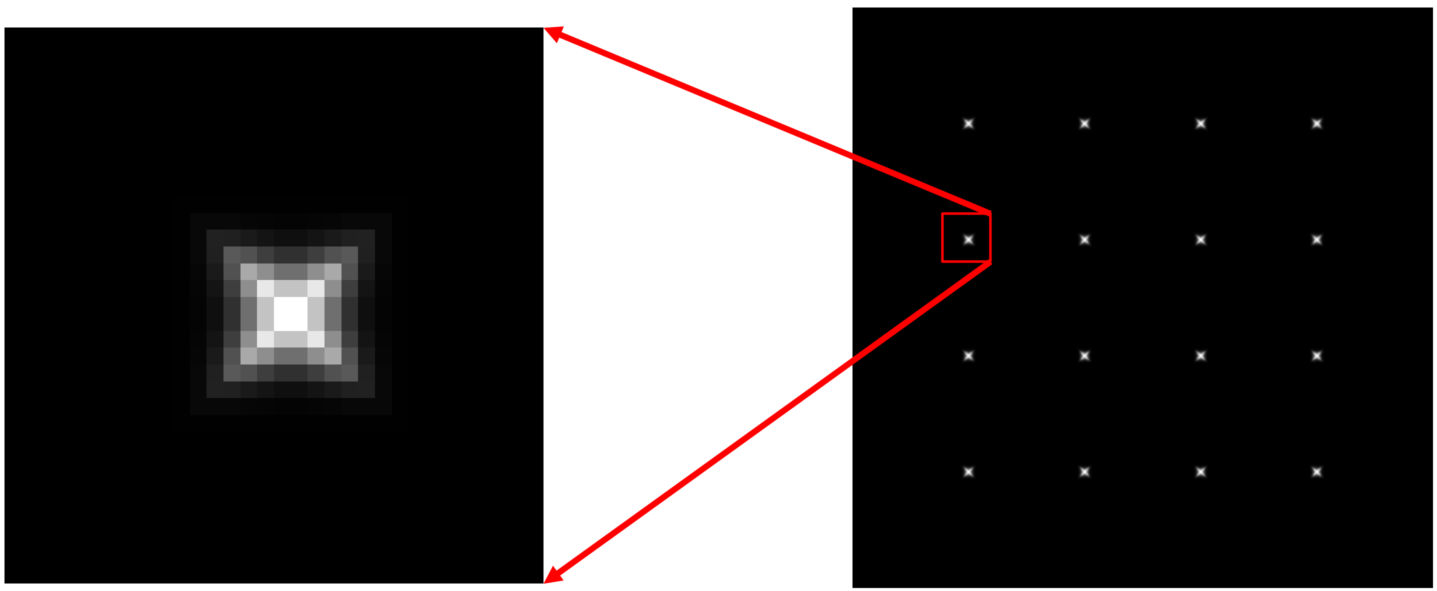


λ_-

Feature Detection

Here's what you do

- Compute the gradient at each point in the image
- Create the *A* matrix from the entries in the gradient
- Compute the eigenvalues.
- Find points with large response (λ_- > threshold) [Shi and Tomasi 1994]
- Choose those points where λ_- is a local maximum as features



$\lambda\_$

Harris Detector

Harris and Stephens 1988

Measure of corner response:

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

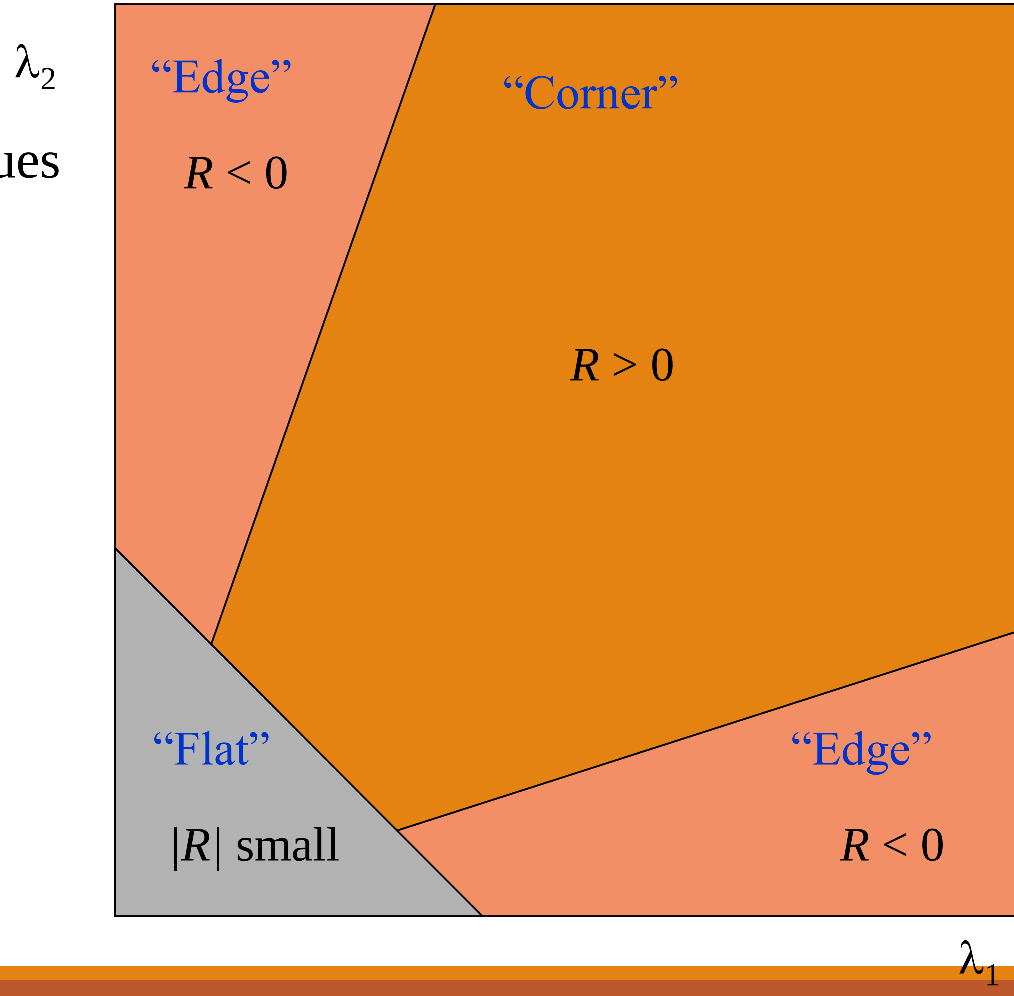
$$\det A = \lambda_1 \lambda_2$$
$$\operatorname{trace} A = \lambda_1 + \lambda_2$$

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

- The *trace* is the sum of the diagonals, i.e., *trace(A)* = $a_{11} + a_{22}$
- Very similar to λ_- but less expensive (no square root)

Harris Detector: Mathematics

- R depends only on eigenvalues of A
- R is large for a corner
- R is negative with large magnitude for an edge
- $|R|$ is small for a flat region



Harris Detector

The Algorithm:

- Find points with large corner response function R (R > threshold)
- Take the points of local maxima of R

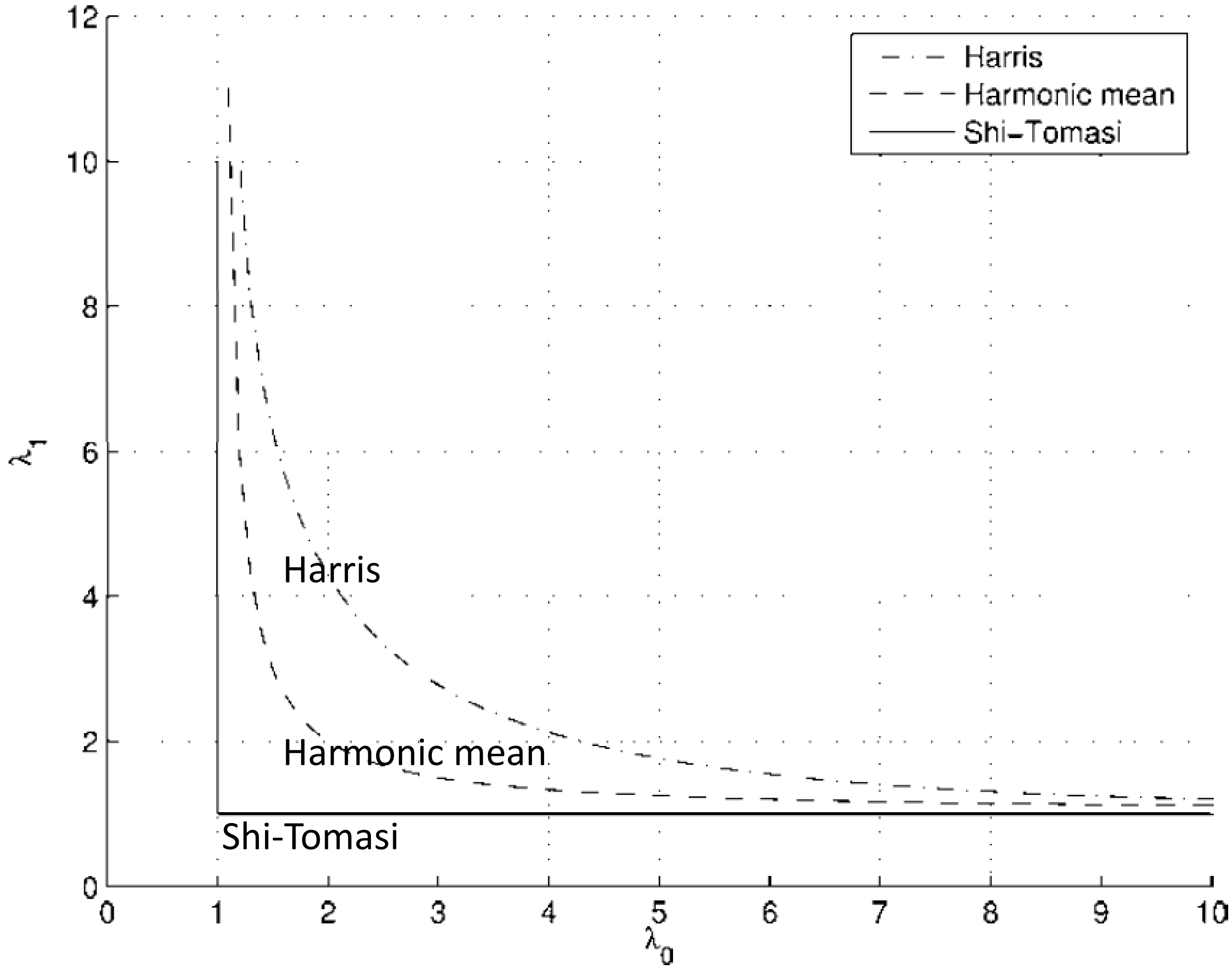
Harmonic Mean

Brown, M., Szeliski, R., and Winder, S. (2005)

$$f = \frac{\det A}{\operatorname{tr} A} = \frac{\lambda_0 \lambda_1}{\lambda_0 + \lambda_1}$$

- Smoother function in the region where $\lambda_0 \approx \lambda_1$

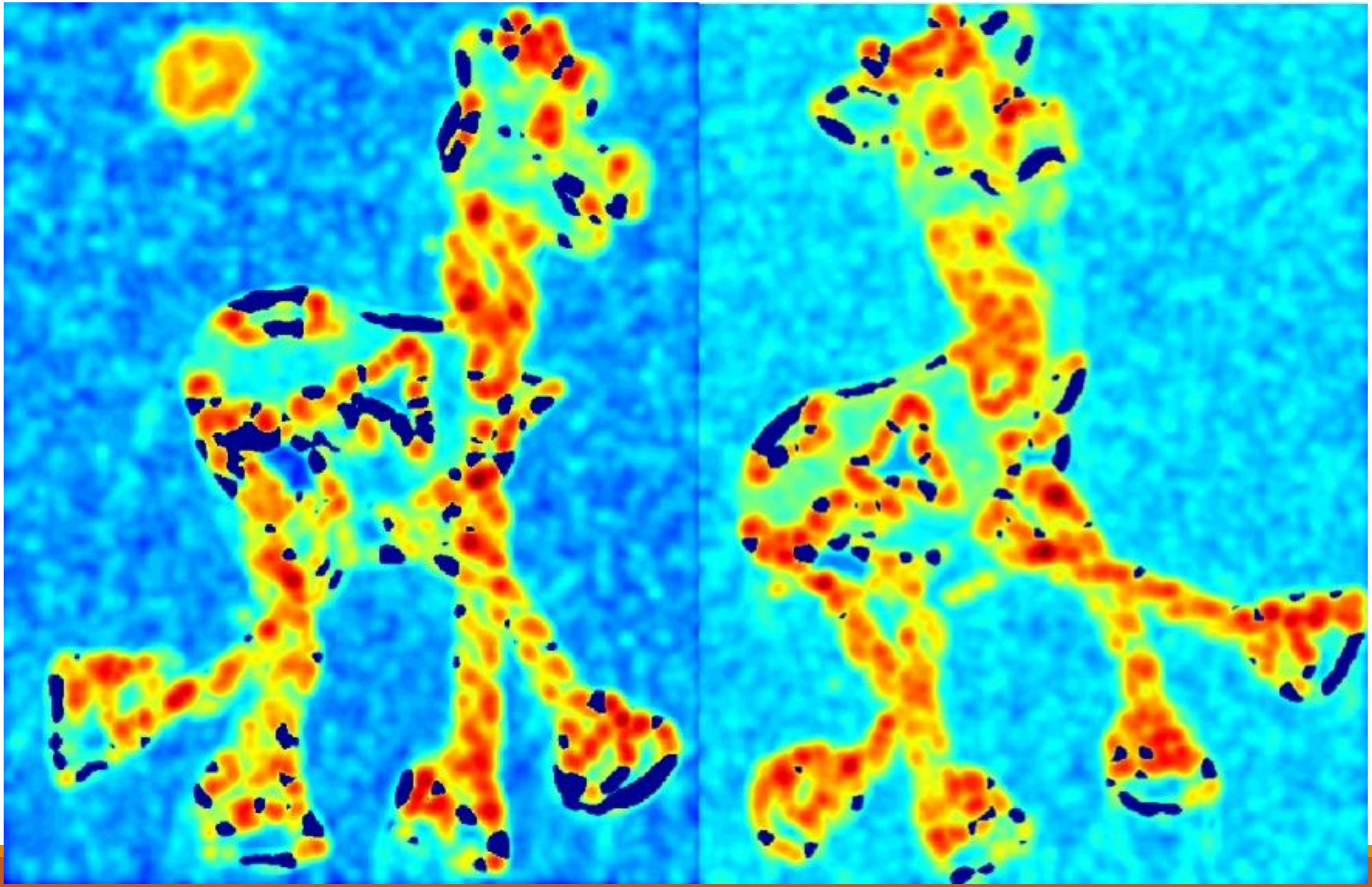
Isocontours of Response



Harris Detector: Workflow

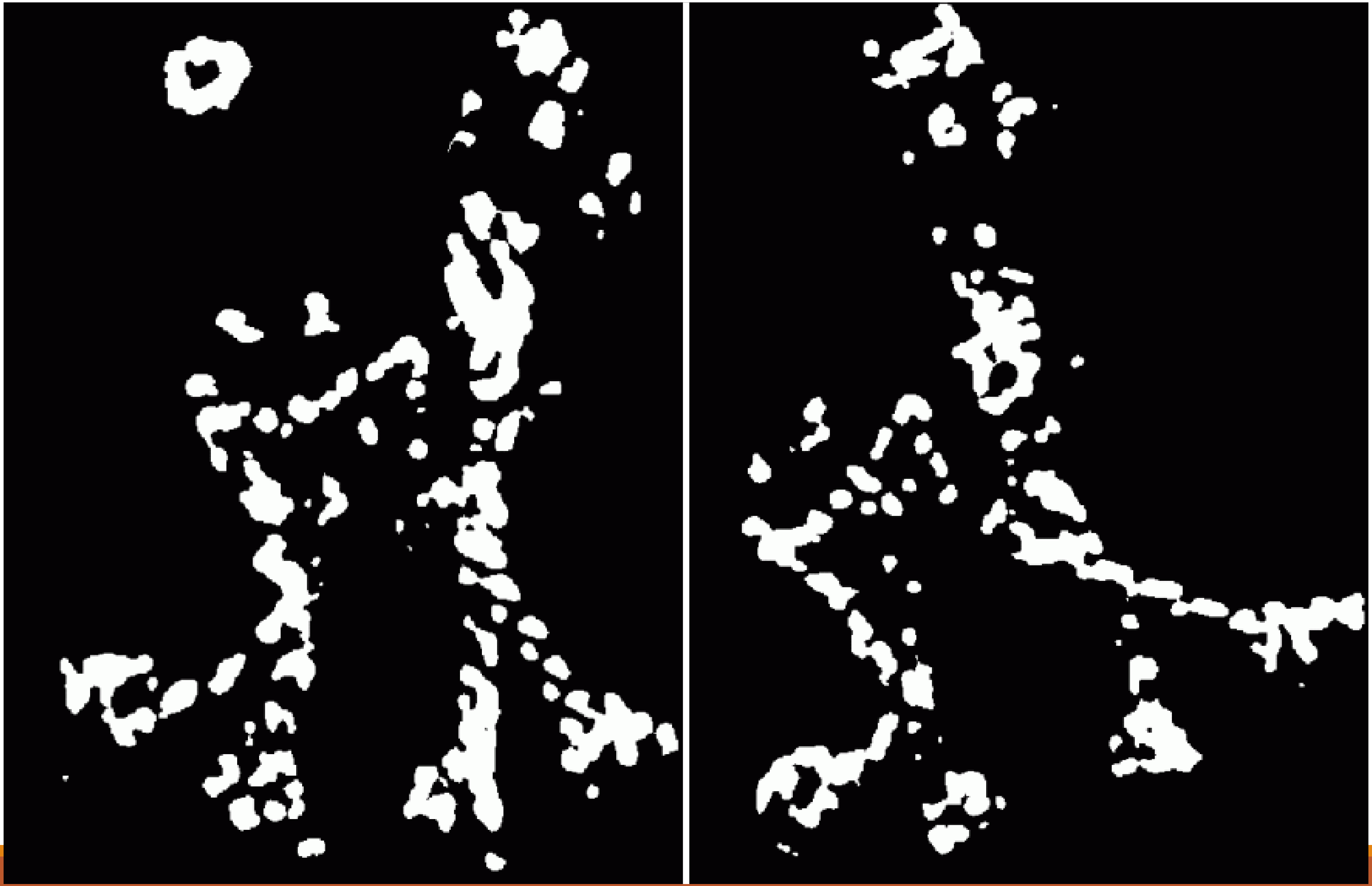
Harris Detector: Workflow

Compute corner response R



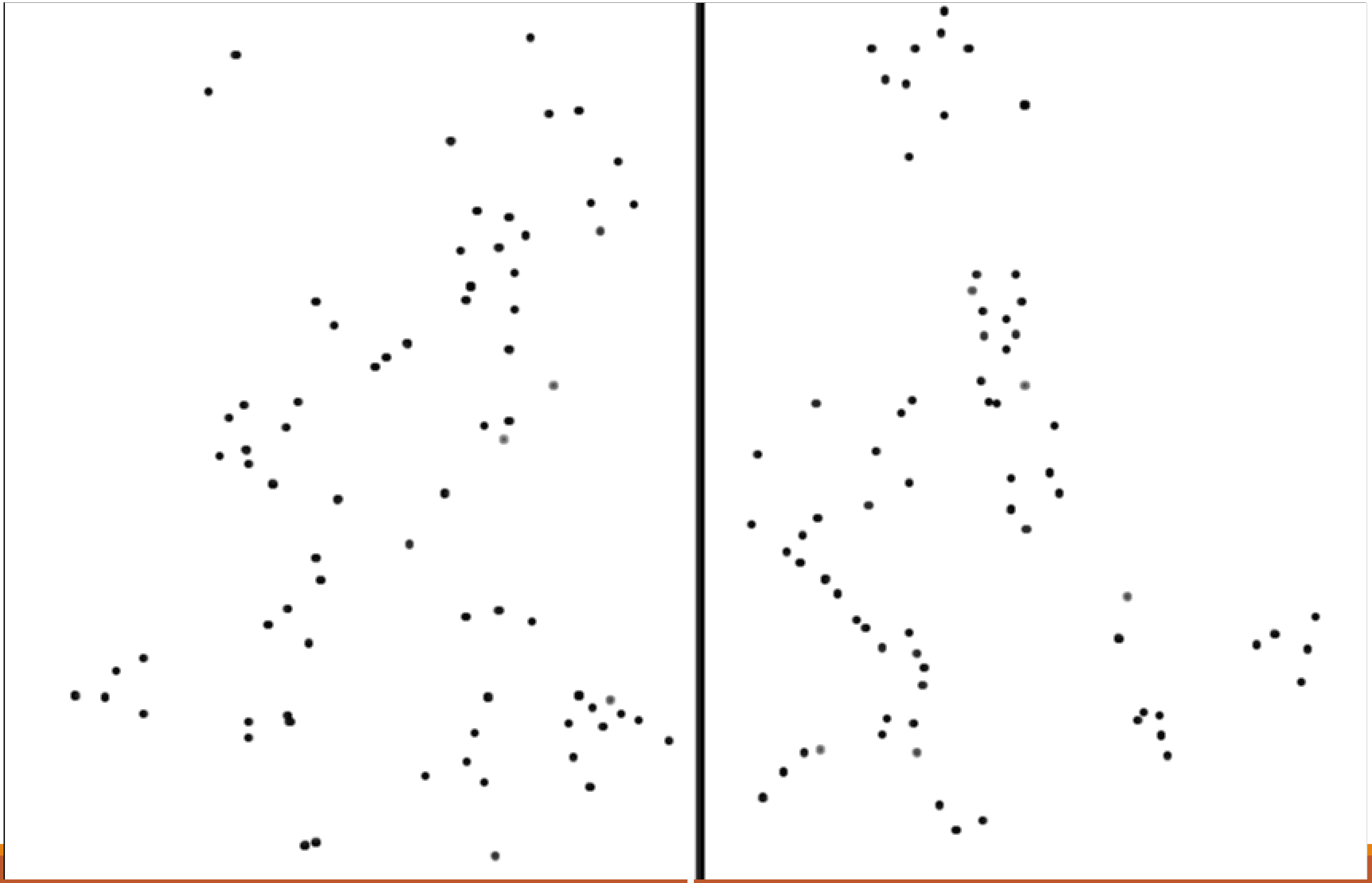
Harris Detector: Workflow

Find points with large corner response: $R > \text{threshold}$



Harris Detector: Workflow

Take only the points of local maxima of R

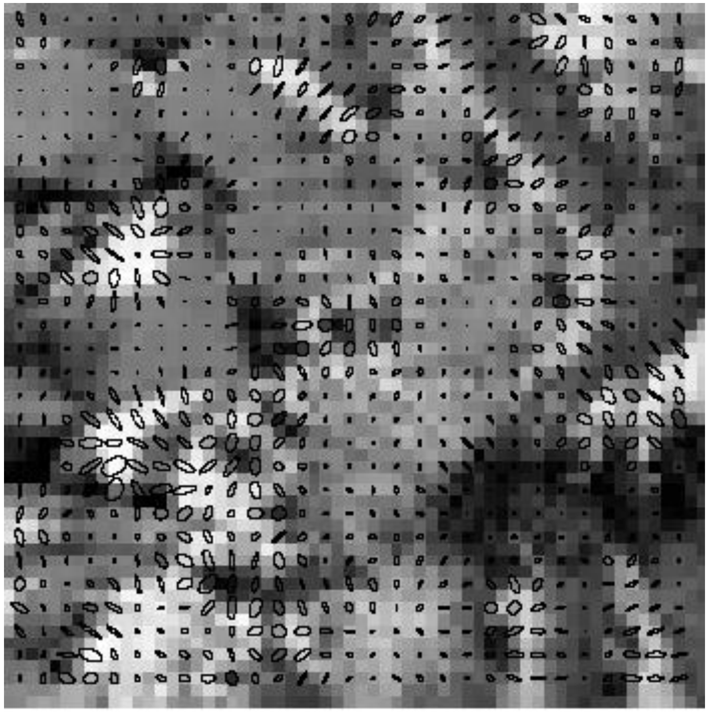


Harris Detector: Workflow

Example: Gradient Covariances

Corners are where both eigenvalues are big

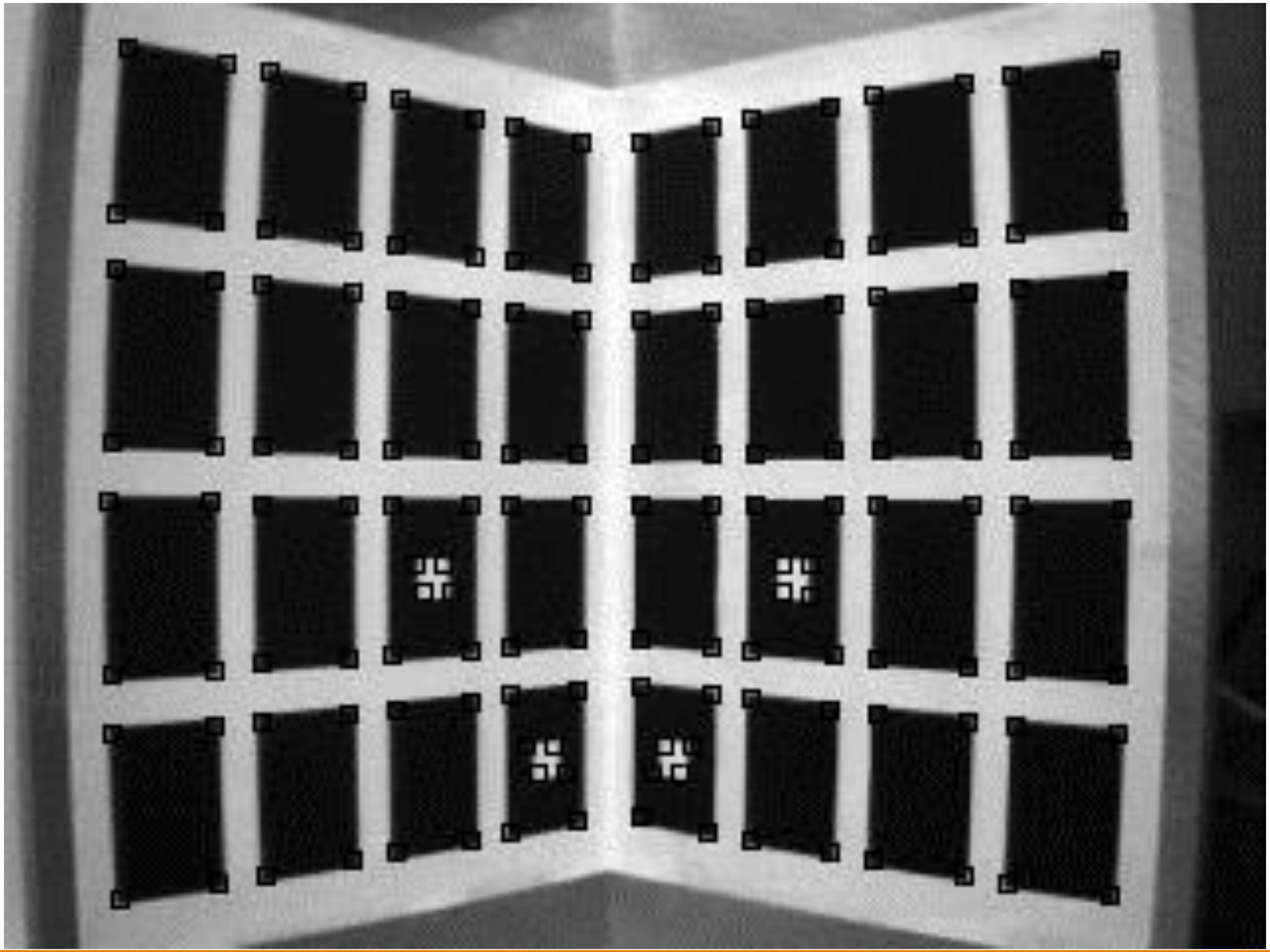
Full image



from Forsyth & Ponce

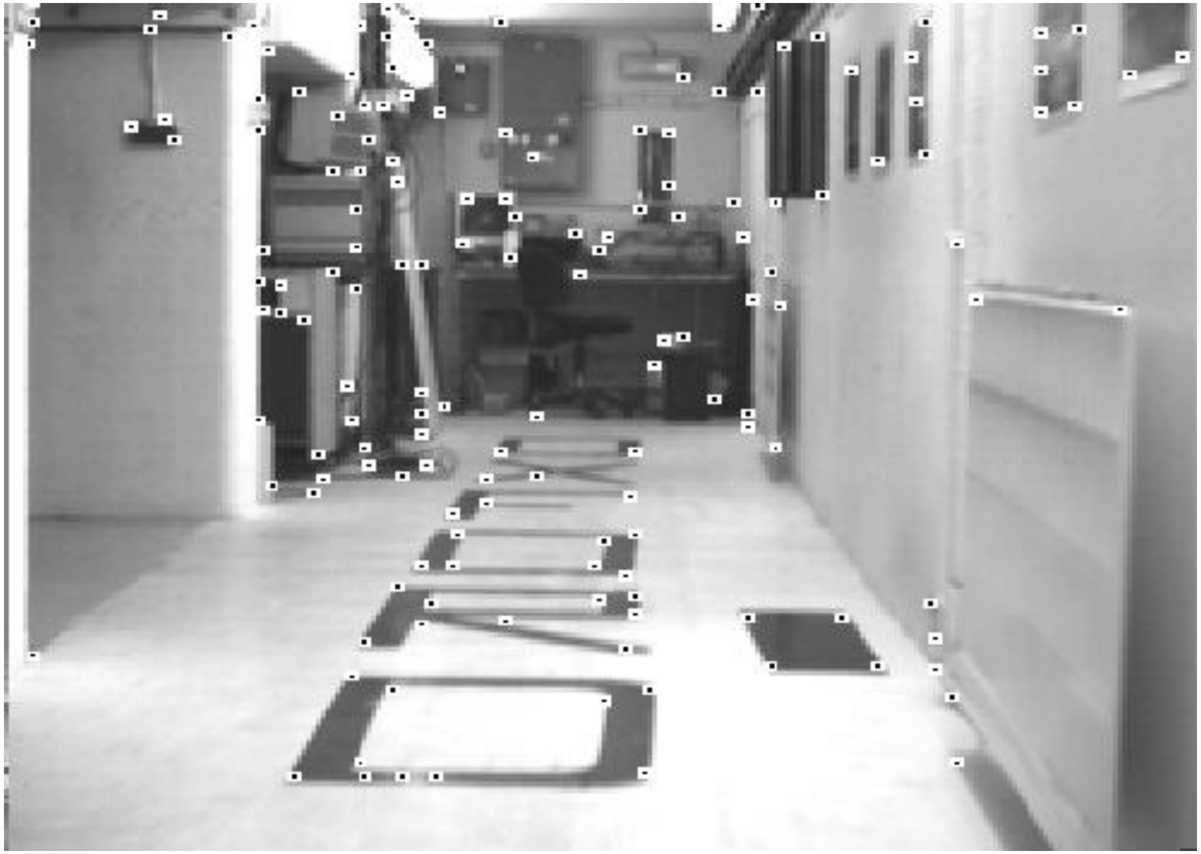
Detail of image with gradient covar-
iance ellipses for 3 x 3 windows

Example: Corner Detection (for camera calibration)



courtesy of B. Wilburn

Example: Corner Detection



courtesy of S. Smith

SUSAN corners

Harris Detector: Summary

Average intensity change in direction $[u,v]$ can be expressed as a bilinear form:

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

Describe a point in terms of eigenvalues of A :
measure of corner response

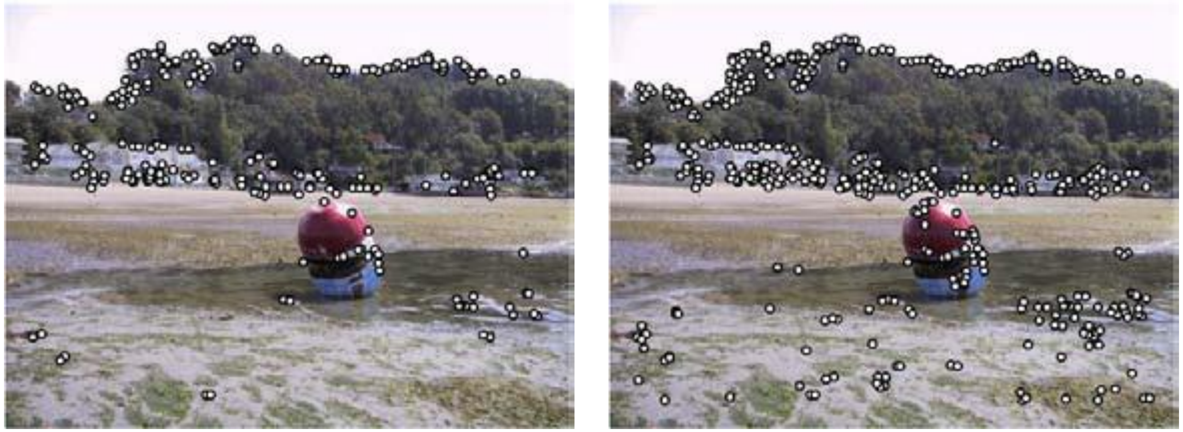
$$R = \lambda_1 \lambda_2 - k\left(\lambda_1 + \lambda_2\right)^2$$

A good (corner) point should have a *large intensity change* in *all directions*, i.e. R should be large positive

Outline of Feature Detection

1. Compute the horizontal and vertical derivatives of the image Ix and Iy by convolving the original image with derivatives of Gaussians
2. Compute the three images corresponding to the outer products of these gradients. (The matrix A is symmetric, so only three entries are needed.)
3. Convolve each of these images with a larger Gaussian.
4. Compute a scalar interest measure using one of the formulas discussed above.
5. Find local maxima above a certain threshold and report them as detected feature point locations.

Adaptive Non-Maximal Suppression (ANMS)



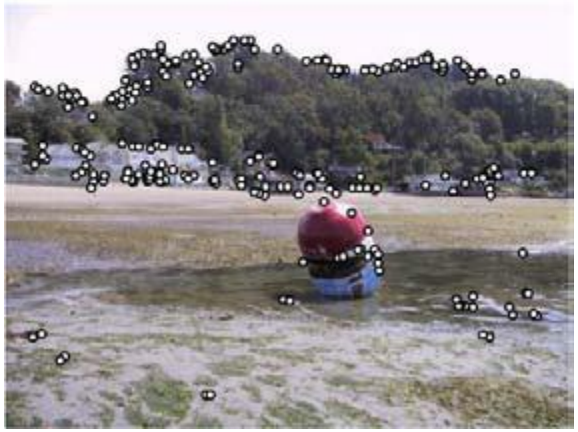
(a) Strongest 250 (b) Strongest 500

Uneven distribution

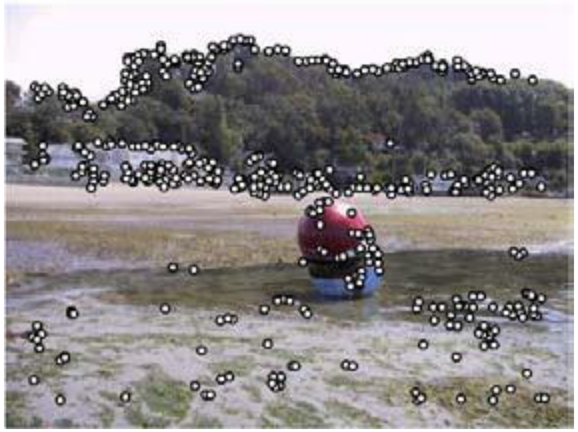
Local maxima & Response value should be significantly (10%) larger than all of its neighbors within a radius (r)

Adaptive suppression radius r

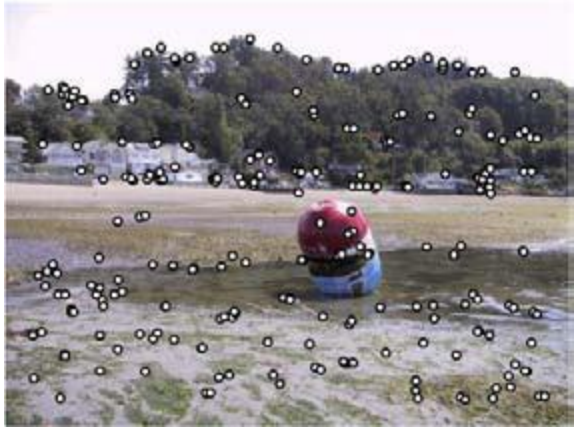
Adaptive Non-Maximal Suppression (ANMS)



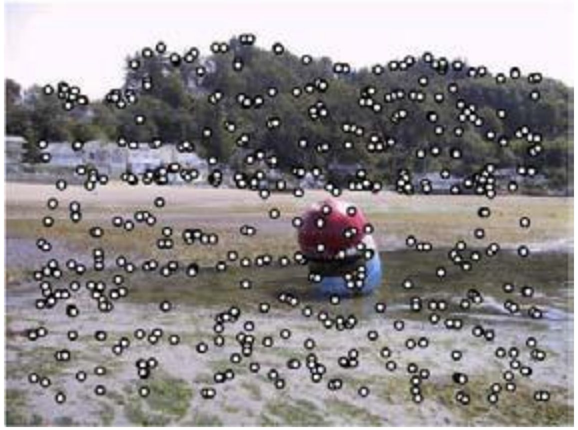
(a) Strongest 250



(b) Strongest 500



(c) ANMS 250, r = 24



(d) ANMS 500, r = 16

Invariance

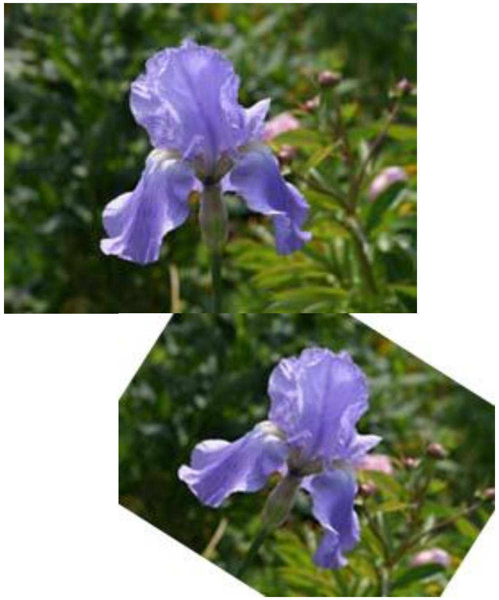
Suppose you **rotate** the image by some angle

- Will you still pick up the same features?

What if you change the brightness?

Scale?

Invariance



$$\boldsymbol{A} = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Rotation
Translation
Brightness

Repeatability of feature detector:

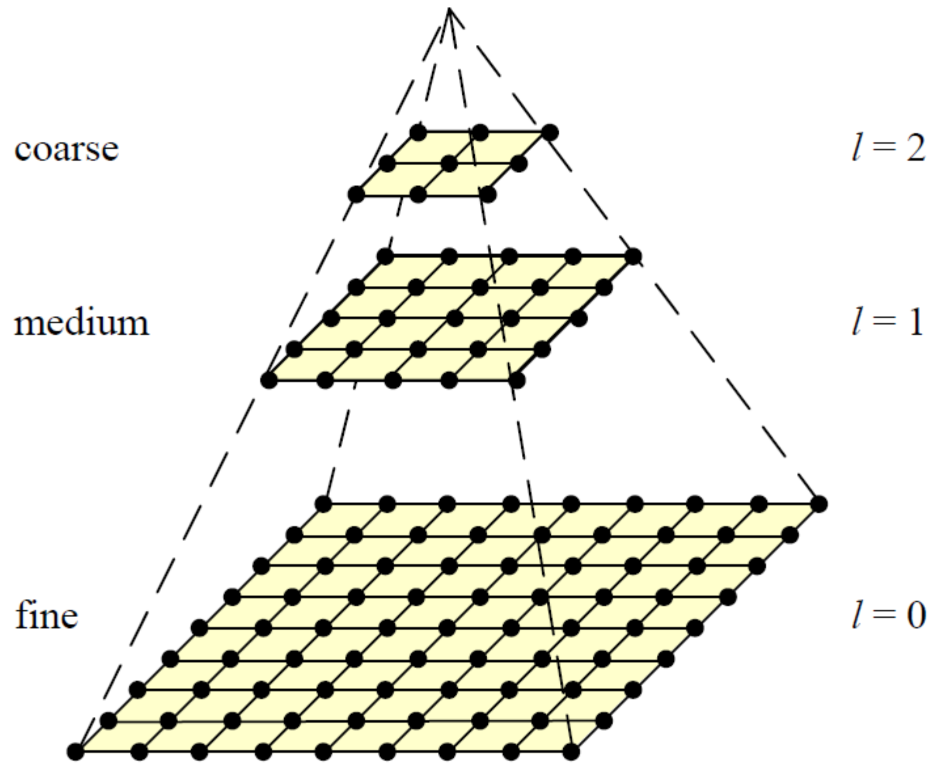
frequency with which keypoints are detected in one image are found within ε (ε=1.5) pixels of the corresponding location in a transformed image

Scale Invariance

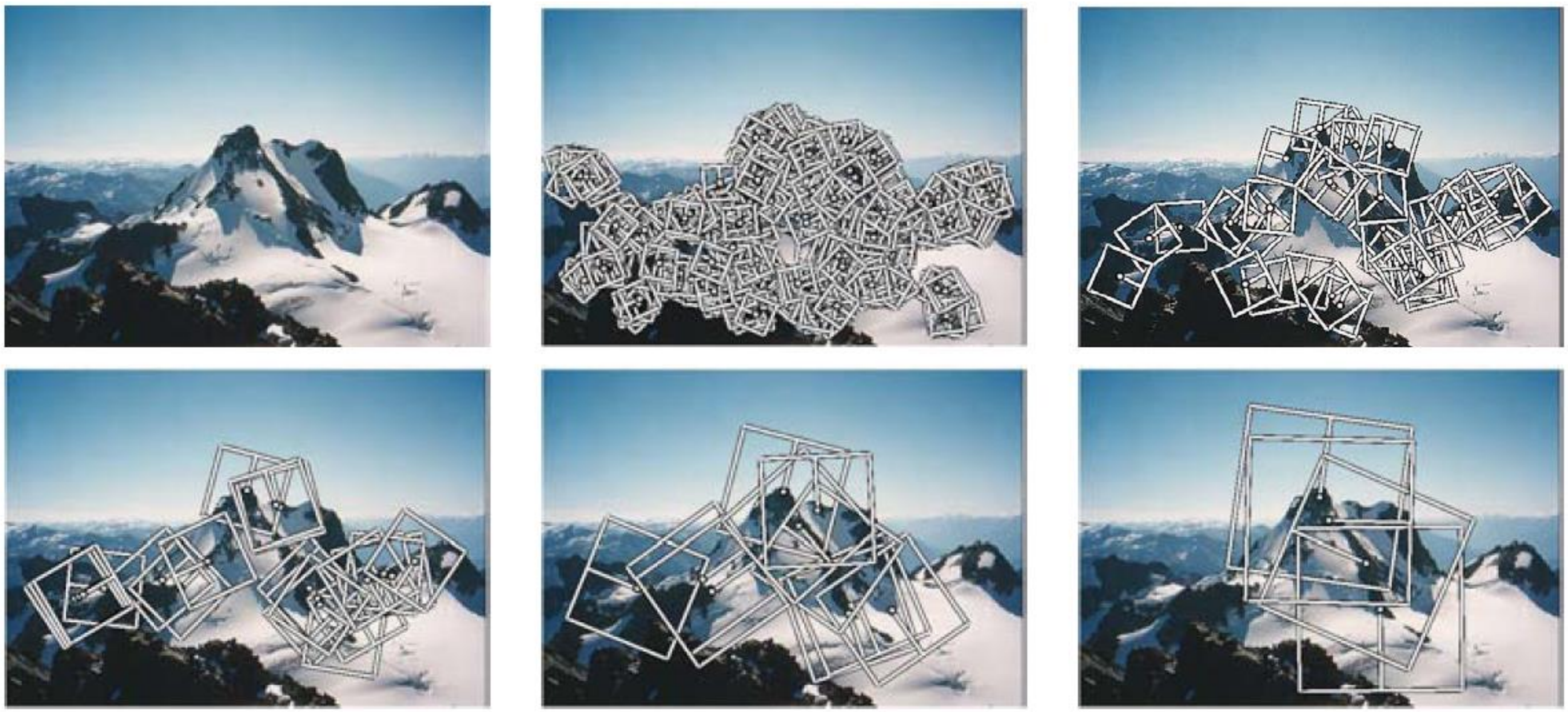
Detect features at a variety of scales

Multiple resolutions in a pyramid

Matching in all possible levels



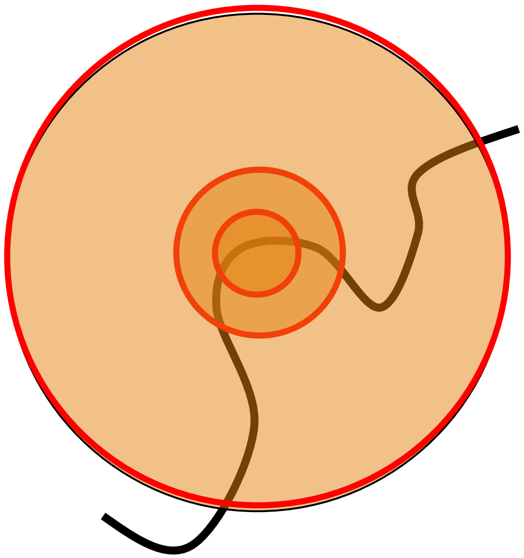
Multi-Scale Oriented Patches



A fixed number of scales

Scale invariant detection

Suppose you're looking for corners



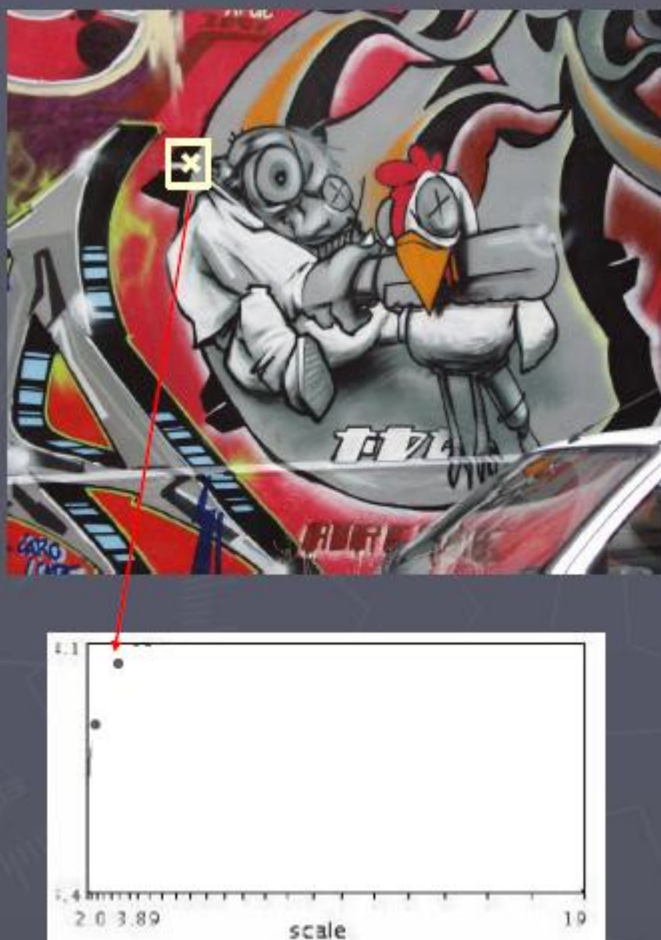
Automatic scale selection

Lindeberg et al., 1996

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

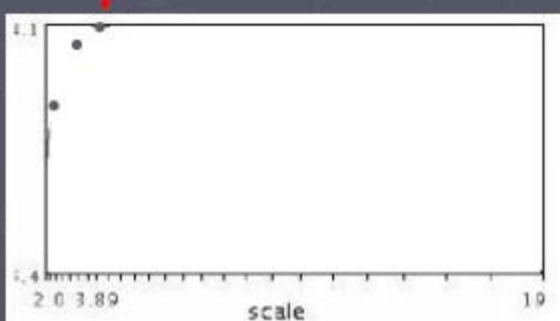
Slide from Tinne Tuytelaars

Automatic scale selection



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

Automatic scale selection

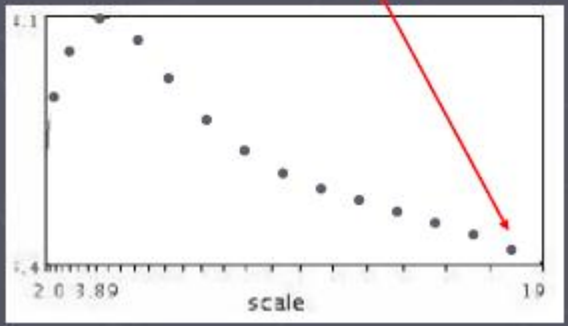


$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic scale selection

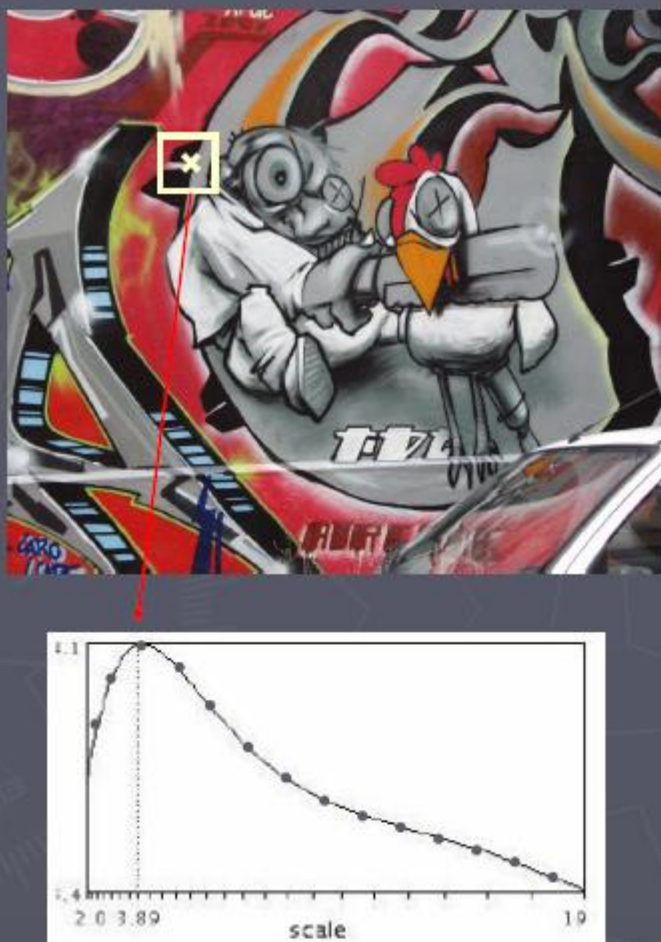
$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

Automatic scale selection



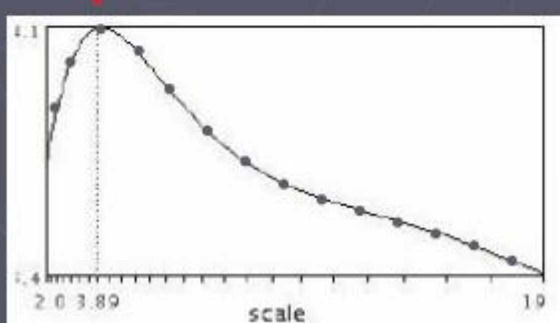
$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

Automatic scale selection

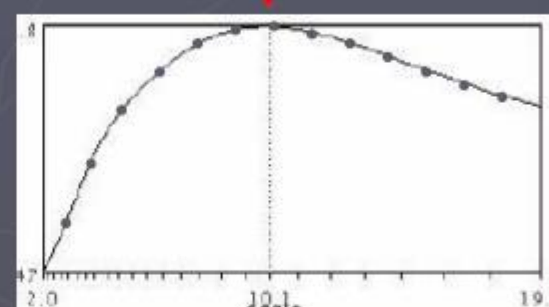


$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

Automatic scale selection



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



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

Automatic scale selection

Normalize: rescale to fixed size

SIFT Feature

Distinctive Image Features from Scale-Invariant Keypoints, David G. Lowe

Scale Invariant Feature Transform (SIFT)

Detect features that densely cover the image over the full range of scales and locations

Keypoint Detection and Matching

Four steps:

- Feature detection
- Feature description
- Feature matching
- Feature tracking

SIFT Background

Scale-invariant feature transform

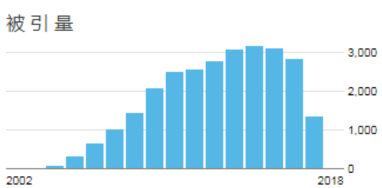
- **SIFT:** to detect and describe local features in an images.
- Proposed by *David Lowe* in ICCV1999.
- Refined in IJCV 2004.
- Wildly used in image search, object recognition, video tracking, gesture recognition, *etc*.

David Lowe
Professor in UBC

34128 times, 2016-3-1

43408 times, 2017-9-20

48404 times, 2018-9-19



[PDF] Distinctive Image Features from Scale-Invariant Keypoints
https://www.cs.ubc.ca/~**lowe**/papers/ijcv04.pdf ▾ 翻译此页
作者：DG Lowe - 2004 - 被引用次数：34128 - 相关文章
2004年1月5日 - David G. Lowe. Computer ... This approach has been named the Scale Invariant Feature Transform (SIFT), as it transforms image data into ...

Distinctive Image Features from Scale-Invariant Keypoints | SpringerLink
https://link.springer.com/article/10.1023/B:VISI.0000029664.99615.94 - 翻译此页
作者：DG Lowe - 2004 - 被引用次数：43408 - 相关文章
Abstract. This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different ...

Why SIFT is so popular?

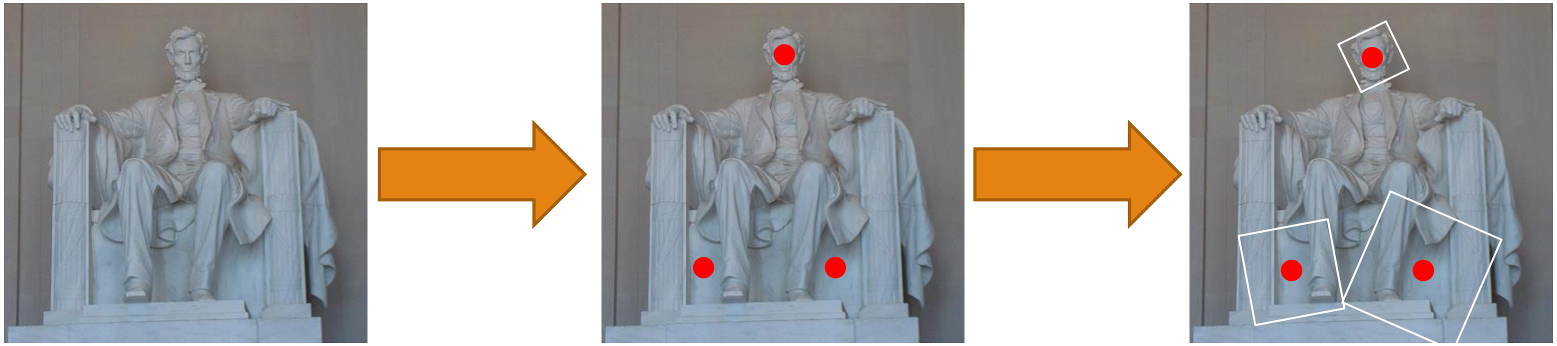
An instance of object matching

Why SIFT is so popular?

Desired property of SIFT

- Invariant to scale change
- Invariant to rotation change
- Invariant to illumination change
- Robust to addition of noise
- Robust to substantial range of affine transformation
- Robust to 3D view point
- Highly distinctive for discrimination

How to extract SIFT



Test image

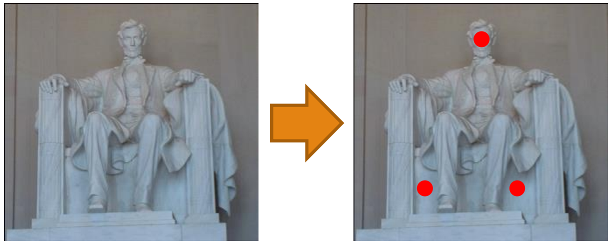
Detector: where are the local features?

Descriptor: how to describe them?

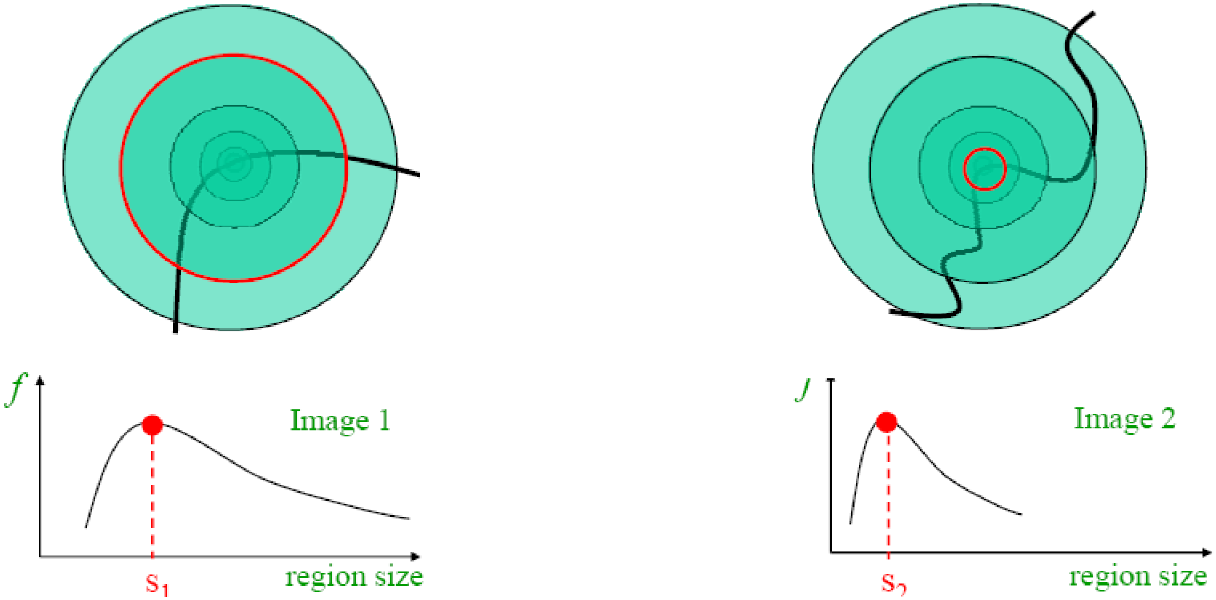
SIFT Detector

Desired properties for detector

- **Position**: Repeatable across different changes
- **Scale**: automatic scale estimation



Intuition: Find scale that gives local maxima of some function *f* in both position and scale.



What can be the “signature” function f?

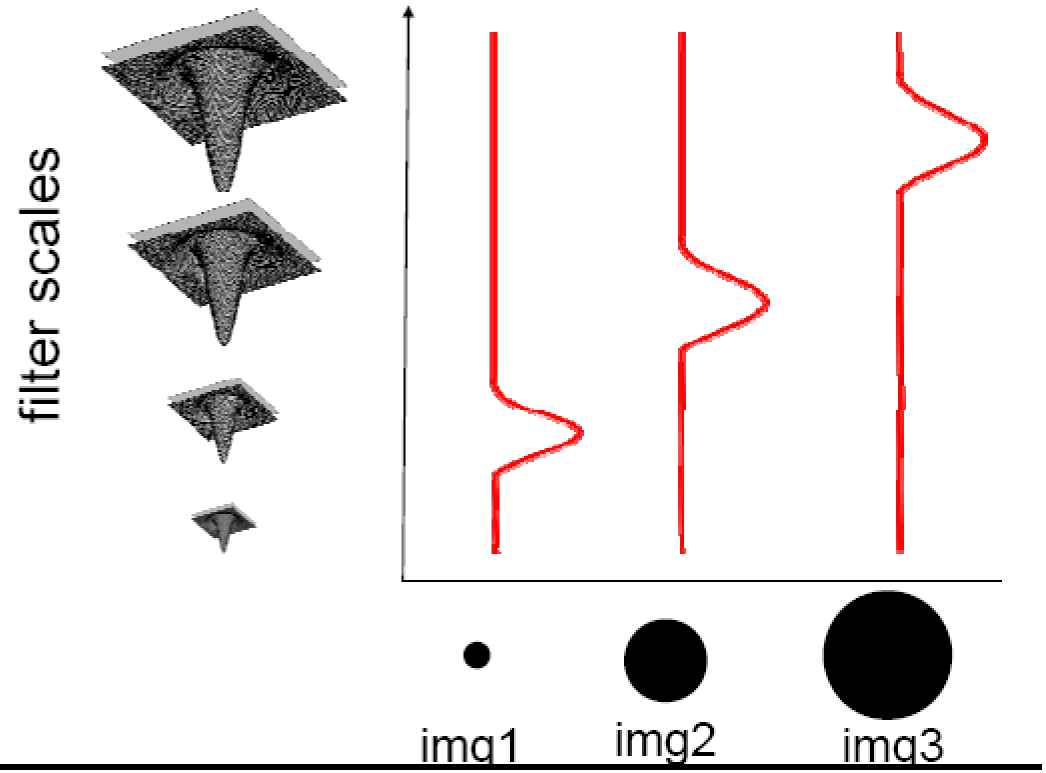
Scale-space kernel

$$f(x, y, \sigma)$$

What can be the “signature” function *f*?

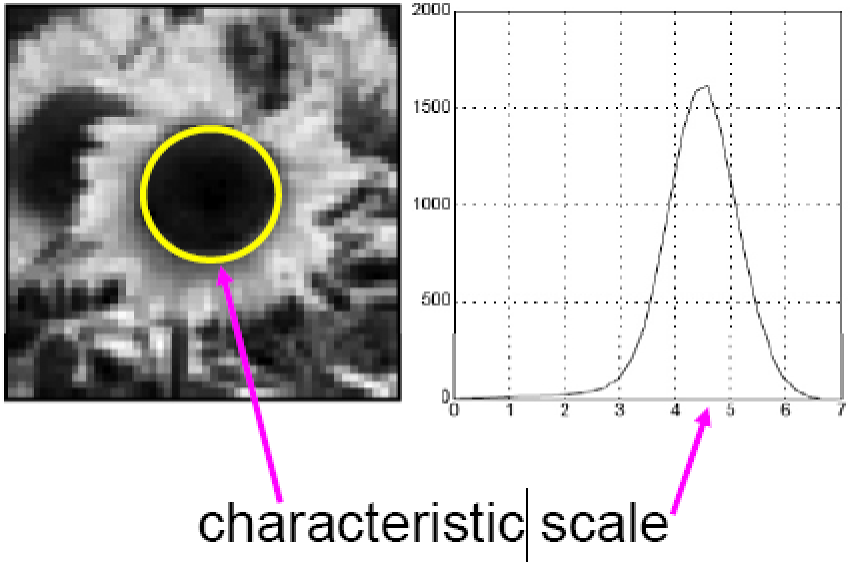
Laplacian-of-Gaussian = “***blob***” detector

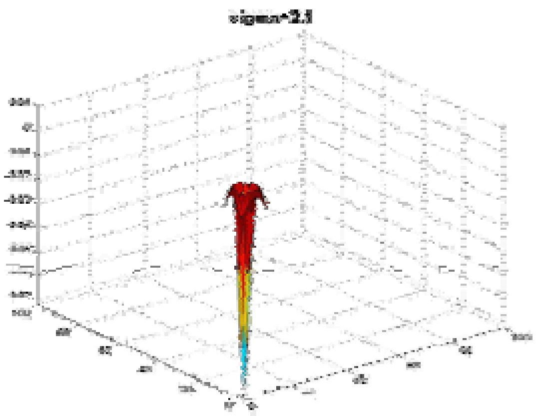
$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

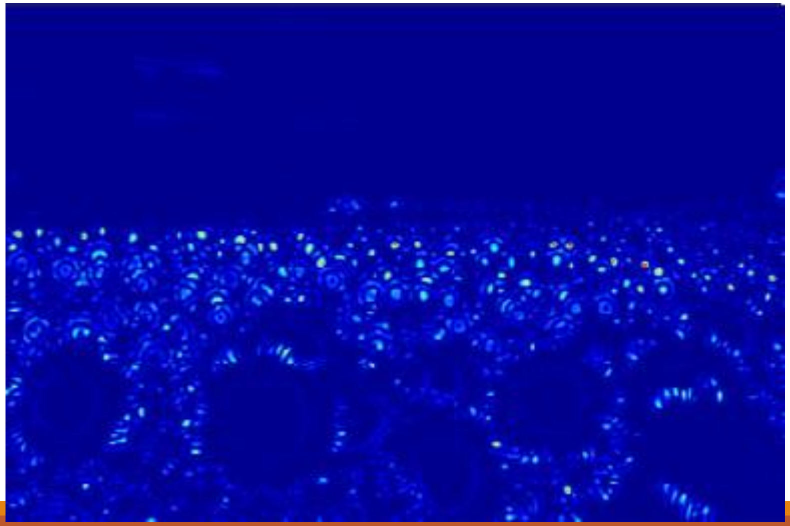


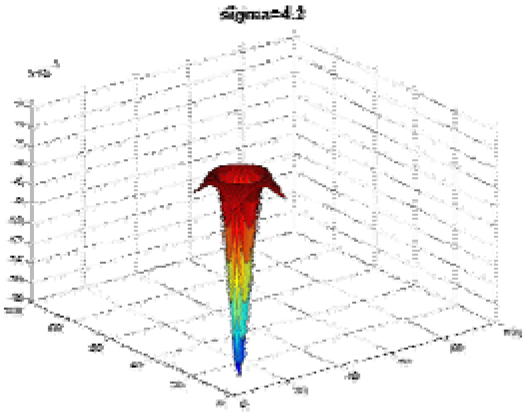
At a given point in the image:

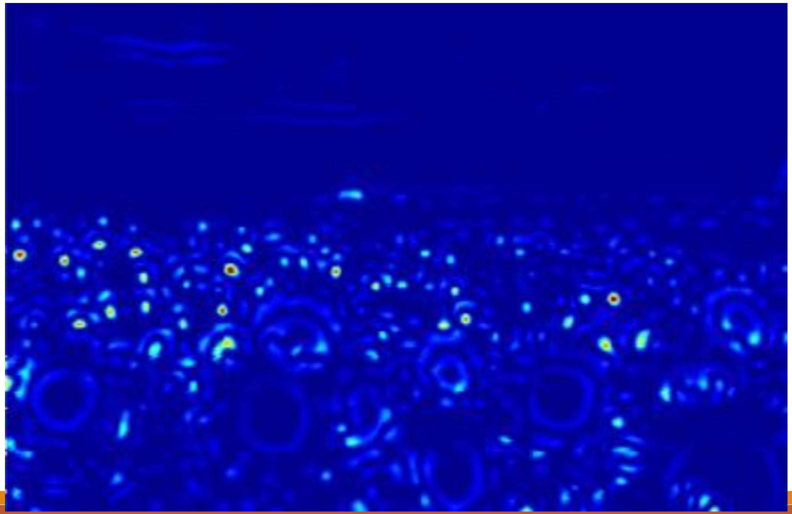
We define the *characteristic scale* as the scale that produces peak of Laplacian response

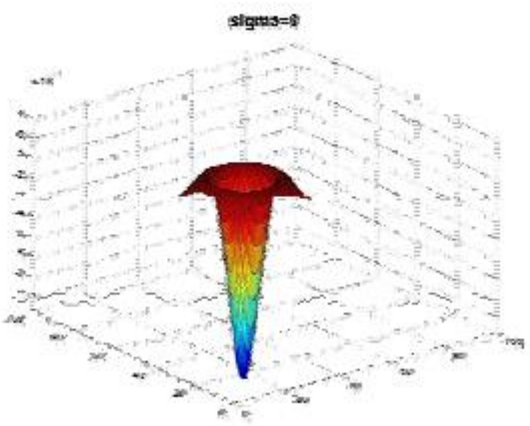


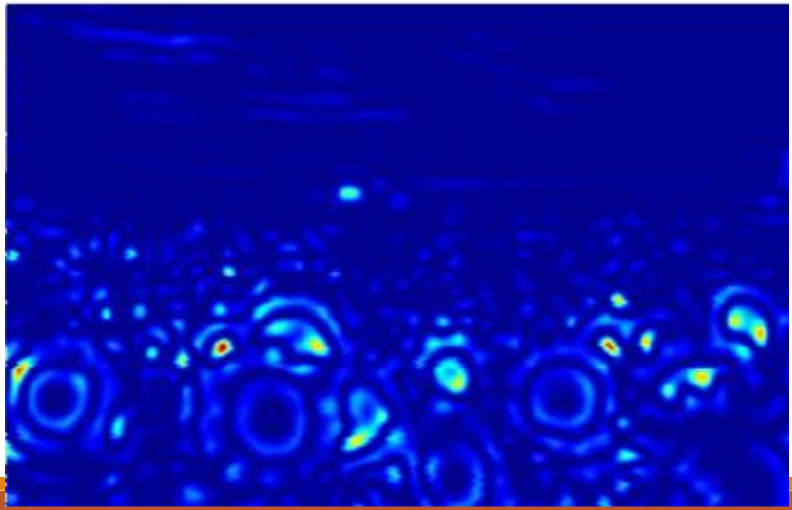


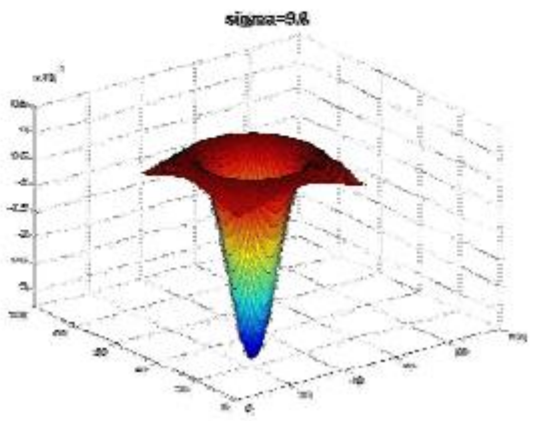


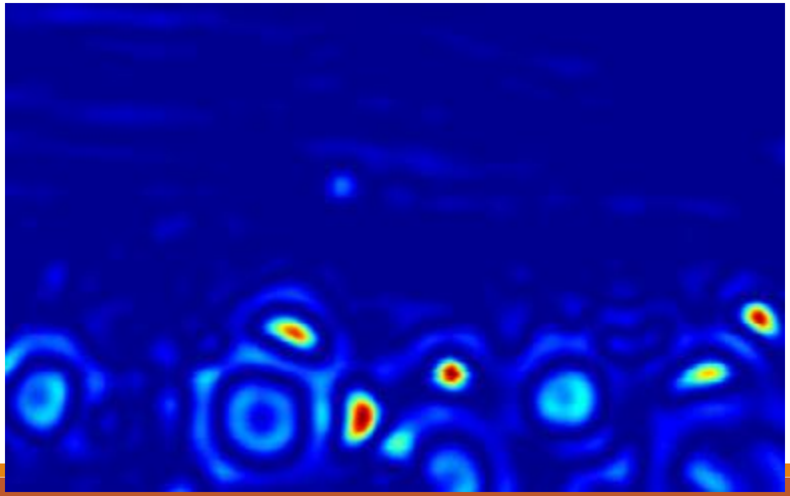

sigma=4.2

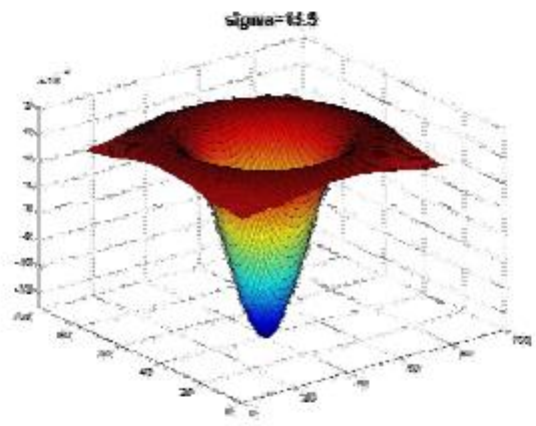


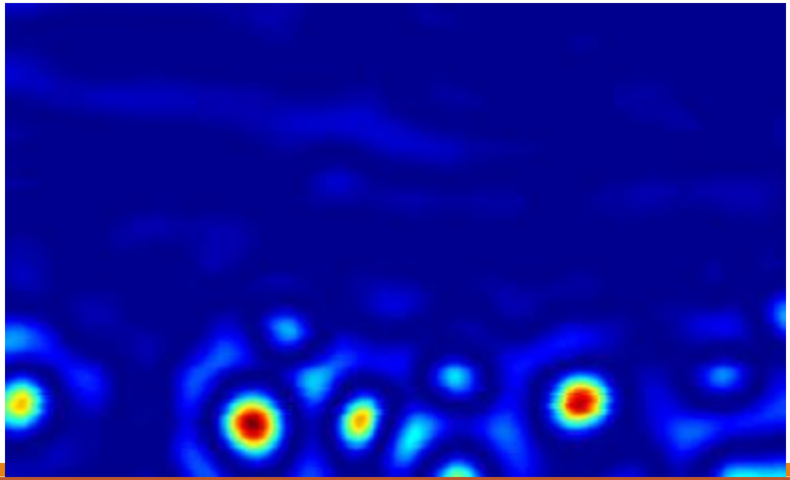


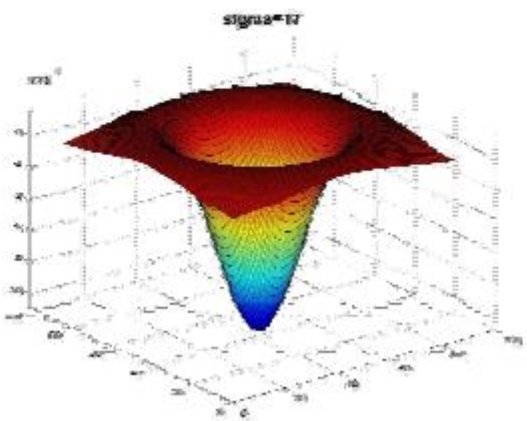


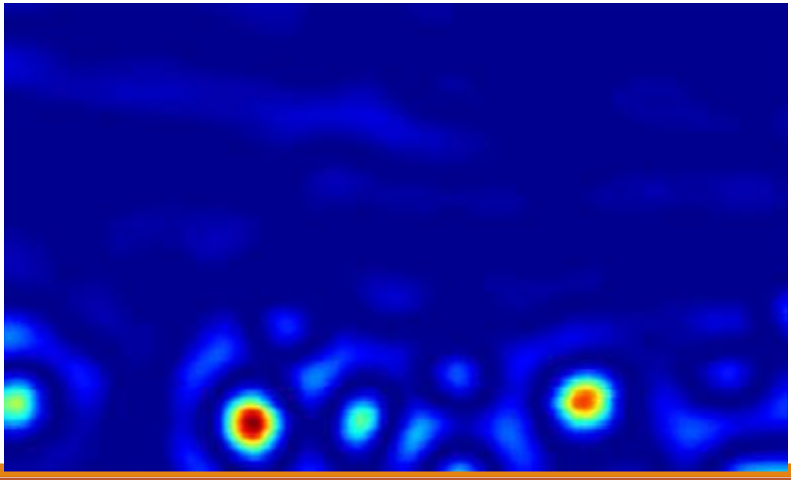

sigma=9.8



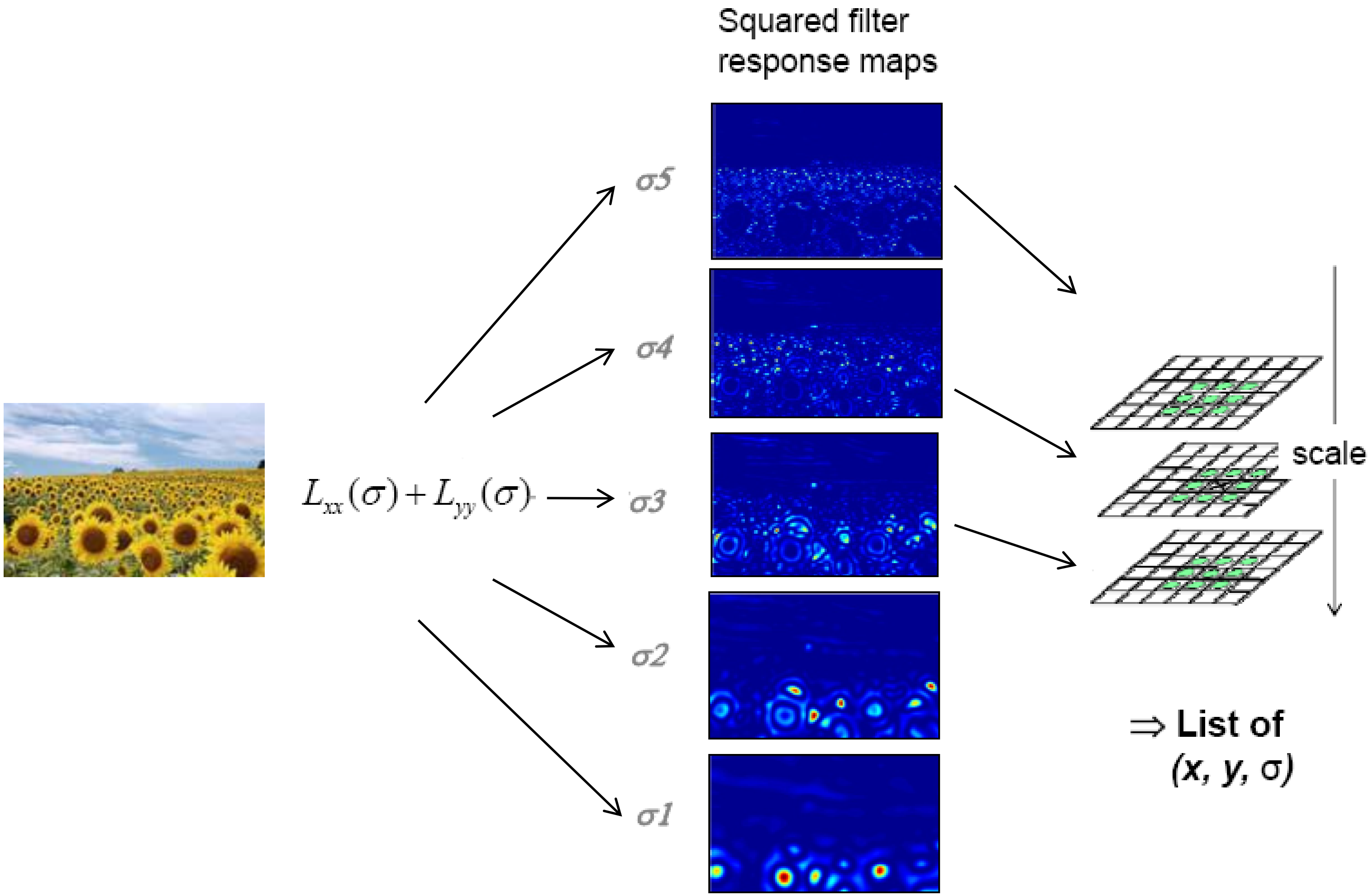




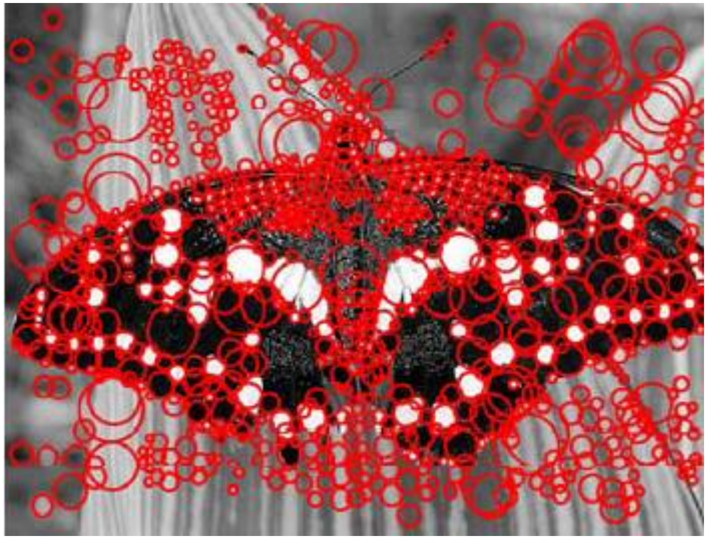




Scale-space blob detection



Scale-space blob detector: Example

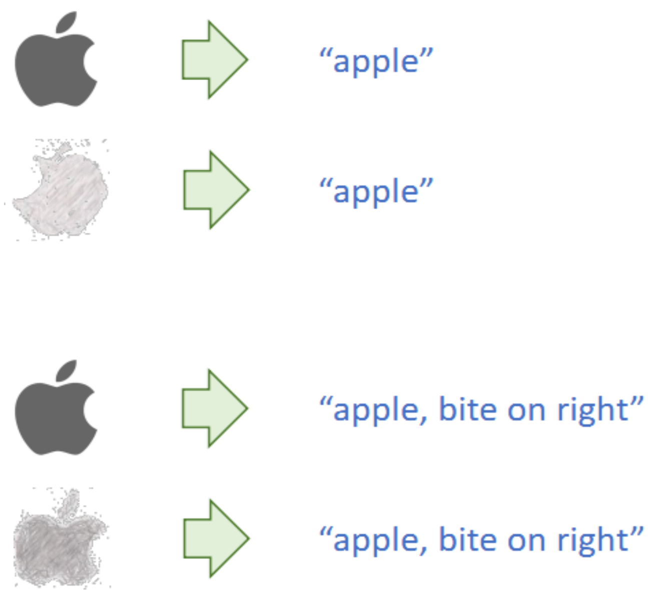


Invariant vs. Equivariant

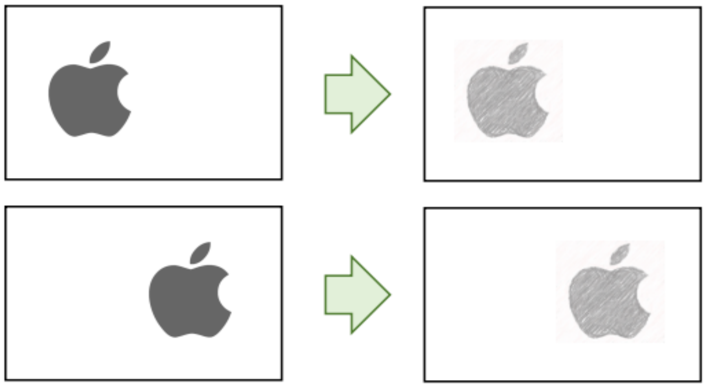
Invariance $f(g[I(x)]) = f\big(I(x)\big)$

Equivariance $f(g[I(x)]) = g[f(I(x))]$

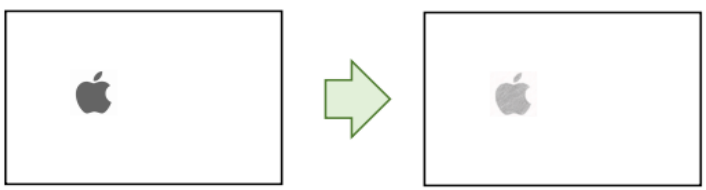
h: Image transformation
f: Image filter



Invariance



translation-equivariant



scale-equivariant

IS LOG scale-invariant?

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

The scale-space kernel
[Koenderink (1984)]
[Lindeberg (1994)]

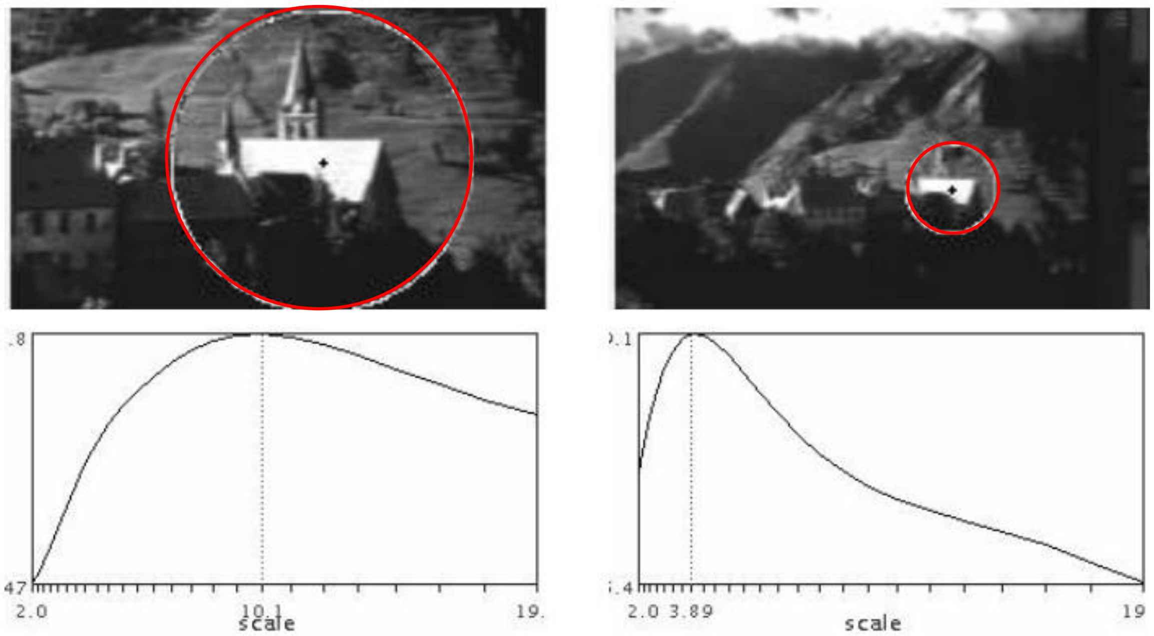
$$\nabla^2 G_\sigma(x,y) = \left(\frac{x^2+y^2}{\sigma^4} - \frac{2}{\sigma^2} \right) G_\sigma(x,y)$$

The normalization of the Laplacian with the factor σ^2 is required for true scale invariance. --Lindeberg (1994)

$$\sigma^2 \nabla^2 G_\sigma(x,y) = \frac{x^2+y^2-2\sigma^2}{\sigma^2} G_\sigma(x,y)$$

IS LOG scale-invariant?

I-right =resize(I-left, 2.5)



characteristic scales

3.89*2.5 = 9.725 ~10.1

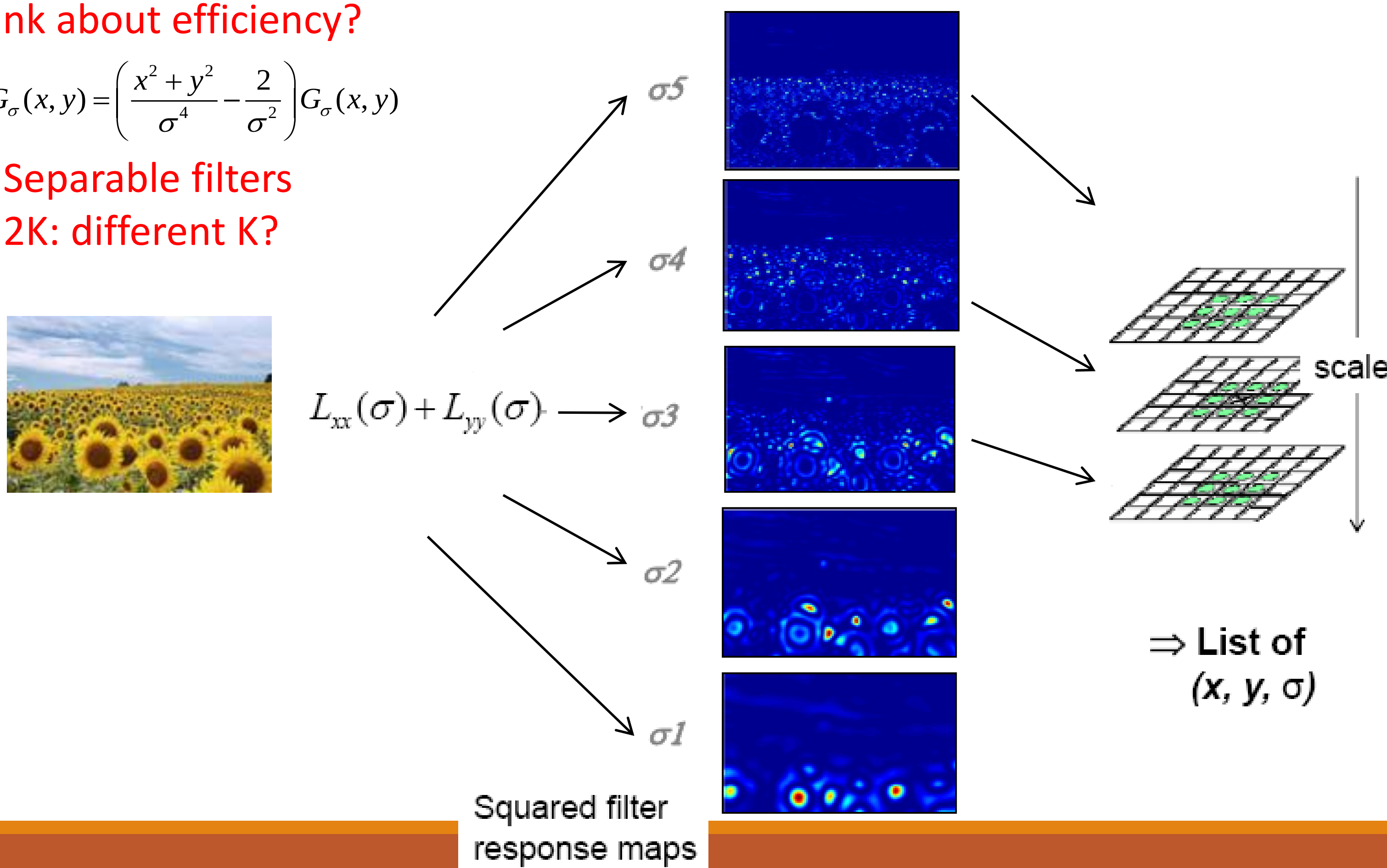
Mikolajczyk, 2002

Scale-space blob detection

Think about efficiency?

$$\nabla^2 G_\sigma(x,y) = \left(\frac{x^2+y^2}{\sigma^4} - \frac{2}{\sigma^2} \right) G_\sigma(x,y)$$

- Separable filters
- 2K: different K?



Separable Filter

Convolution of K-size kernel requires K^2 operations

Can be sped up to 2K operations by

- First performing a 1D horizontal convolution
- Followed by a 1D vertical convolution

$$K = vh^T$$

Separable Filter

$$\frac{1}{K^2}\begin{array}{|c|c|c|c|}\hline 1 & 1 & \cdots & 1 \\ \hline 1 & 1 & \cdots & 1 \\ \hline \vdots & \vdots & 1 & \vdots \\ \hline 1 & 1 & \cdots & 1 \\ \hline \end{array} \quad \frac{1}{16}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \quad \frac{1}{256}\begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 6 & 4 & 1 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 6 & 24 & 36 & 24 & 6 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array} \quad \frac{1}{8}\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \quad \frac{1}{4}\begin{array}{|c|c|c|}\hline 1 & -2 & 1 \\ \hline -2 & 4 & -2 \\ \hline 1 & -2 & 1 \\ \hline \end{array}$$

$$\frac{1}{K}\begin{array}{|c|c|c|c|}\hline 1 & 1 & \cdots & 1 \\ \hline \end{array} \quad \frac{1}{4}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline \end{array} \quad \frac{1}{16}\begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array} \quad \frac{1}{2}\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline \end{array} \quad \frac{1}{2}\begin{array}{|c|c|c|}\hline 1 & -2 & 1 \\ \hline \end{array}$$

(a) box, $K = 5$ (b) bilinear (c) “Gaussian” (d) Sobel (e) corner

Technical detail

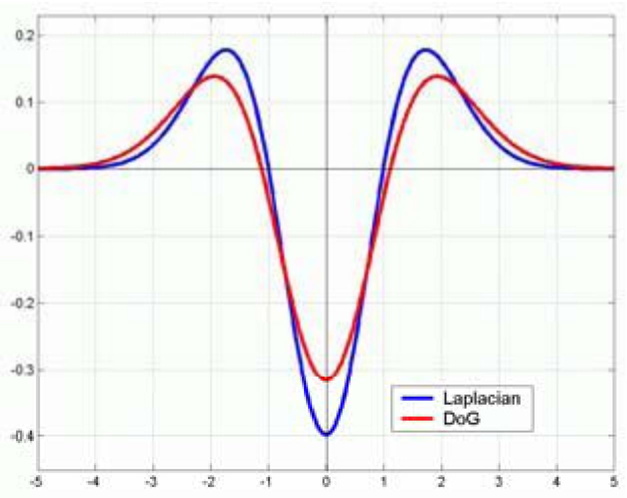
We can approximate the Laplacian with a difference of Gaussians; more efficient to implement.

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

(Laplacian)

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

(Difference of Gaussians)



Difference of Gaussian (DoG)

Gaussian $$G_{\sigma_1}(x,y)=\frac{1}{2\pi\sigma_1^2}e^{-\frac{x^2+y^2}{2\sigma_1^2}}$$

$$g_1(x,y)=G_{\sigma_1}(x,y)*f(x,y) \qquad g_2(x,y)=G_{\sigma_2}(x,y)*f(x,y)$$

DoG

$$\begin{aligned} g_1(x,y)-g_2(x,y) &= G_{\sigma_1}*f(x,y)-G_{\sigma_2}*f(x,y) \\ &= \boxed{(G_{\sigma_1}-G_{\sigma_2})}*f(x,y) \end{aligned}$$

$$DoG \triangleq G_{\sigma_1}-G_{\sigma_2}=\frac{1}{2\pi}\left(\frac{1}{\sigma_1^2}e^{-\frac{x^2+y^2}{2\sigma_1^2}}-\frac{1}{\sigma_2^2}e^{-\frac{x^2+y^2}{2\sigma_2^2}}\right)$$

Relationship between DoG and LoG

Heat diffusion equation

$$\frac{\partial G}{\partial \sigma} = -\frac{1}{\pi\sigma^3} e^{-\frac{x^2+y^2}{2\sigma^2}} + \frac{x^2+y^2}{2\pi\sigma^5} e^{-\frac{x^2+y^2}{2\sigma^2}} = \sigma\left(\frac{x^2+y^2}{\sigma^4} - \frac{2}{\sigma^2}\right)\frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} = \sigma\nabla^2 G$$

Finite difference

$$\sigma\nabla^2 \mathrm{G}_\sigma(x,y) = \frac{\partial \mathrm{G}}{\partial \sigma} \approx \frac{\mathrm{G}(x,y,k\sigma) - G(x,y,\sigma)}{k\sigma - \sigma}$$

$$\boxed{\mathrm{G}(x,y,k\sigma) - G(x,y,\sigma) \approx (k-1)\sigma^2\nabla^2 \mathrm{G}(x,y,\sigma)}$$

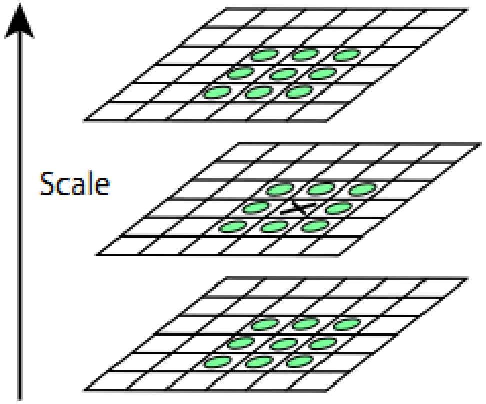
LoG V.S. DoG

$$\underbrace{G(x,y,k\sigma)-G(x,y,\sigma)}_{\text{DoG}} \approx (k-1)\sigma\frac{\partial G}{\partial \sigma} = (k-1)\sigma^2\underbrace{\nabla^2 G}_{\text{LoG}}$$

- Normalized LoG $\sigma^2\nabla^2\mathrm{G}$ for true scale invariance
- DoG has scales differing by a factor *k-1*
- Build scale-space with a constant *k* over all scales

Maxima and minima of DoG

Maxima and minima of the DoG images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales

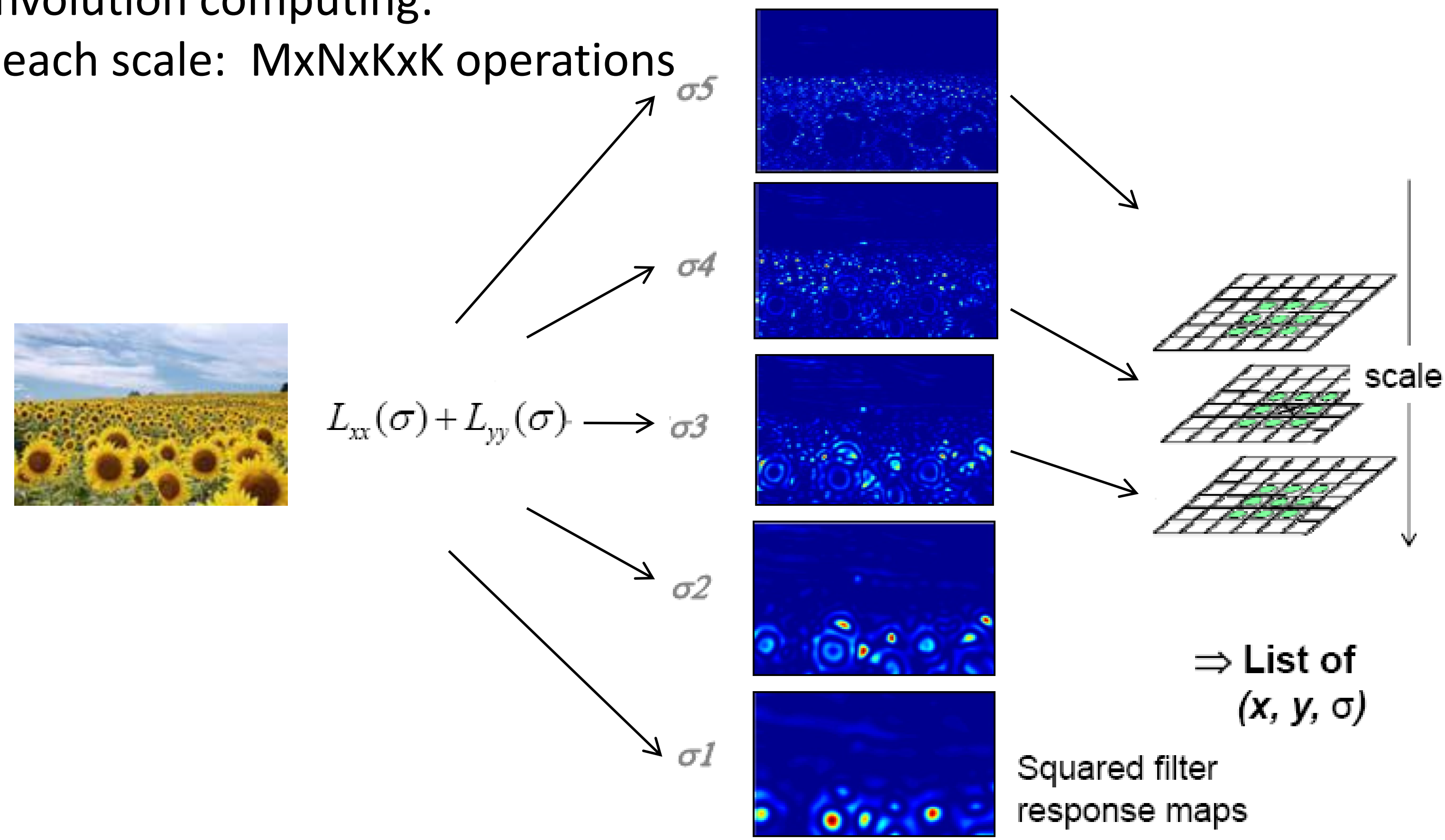


Sampling frequency

Trades off efficiency with completeness

convolution computing:

At each scale: MxNxKxK operations



DoG Image Pyramid

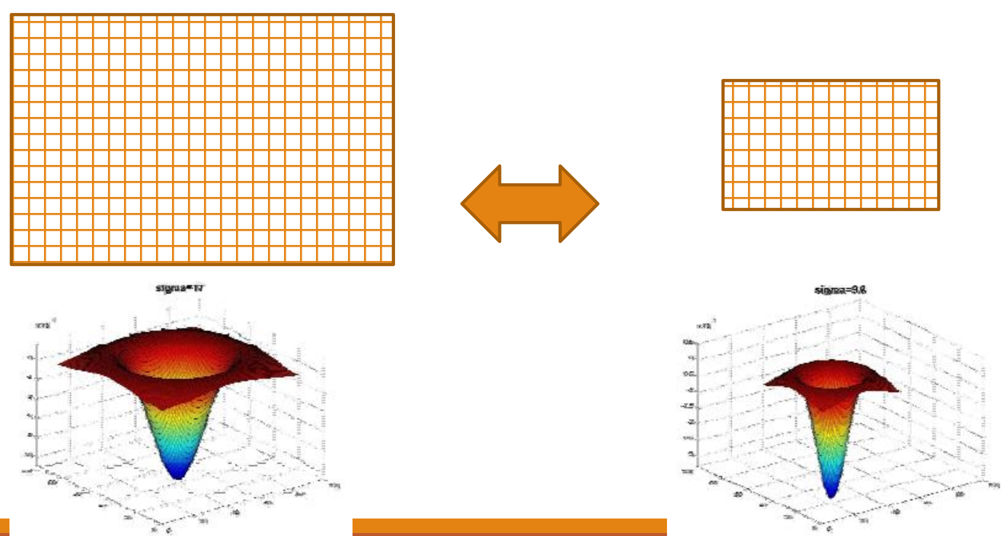
$$\sigma_0, k\sigma_0, k^2\sigma_0, k^3\sigma_0, k^4\sigma_0, k^5\sigma_0, k^6\sigma_0, \ldots$$

$$\sigma_0 \rightarrow 2\sigma_0$$

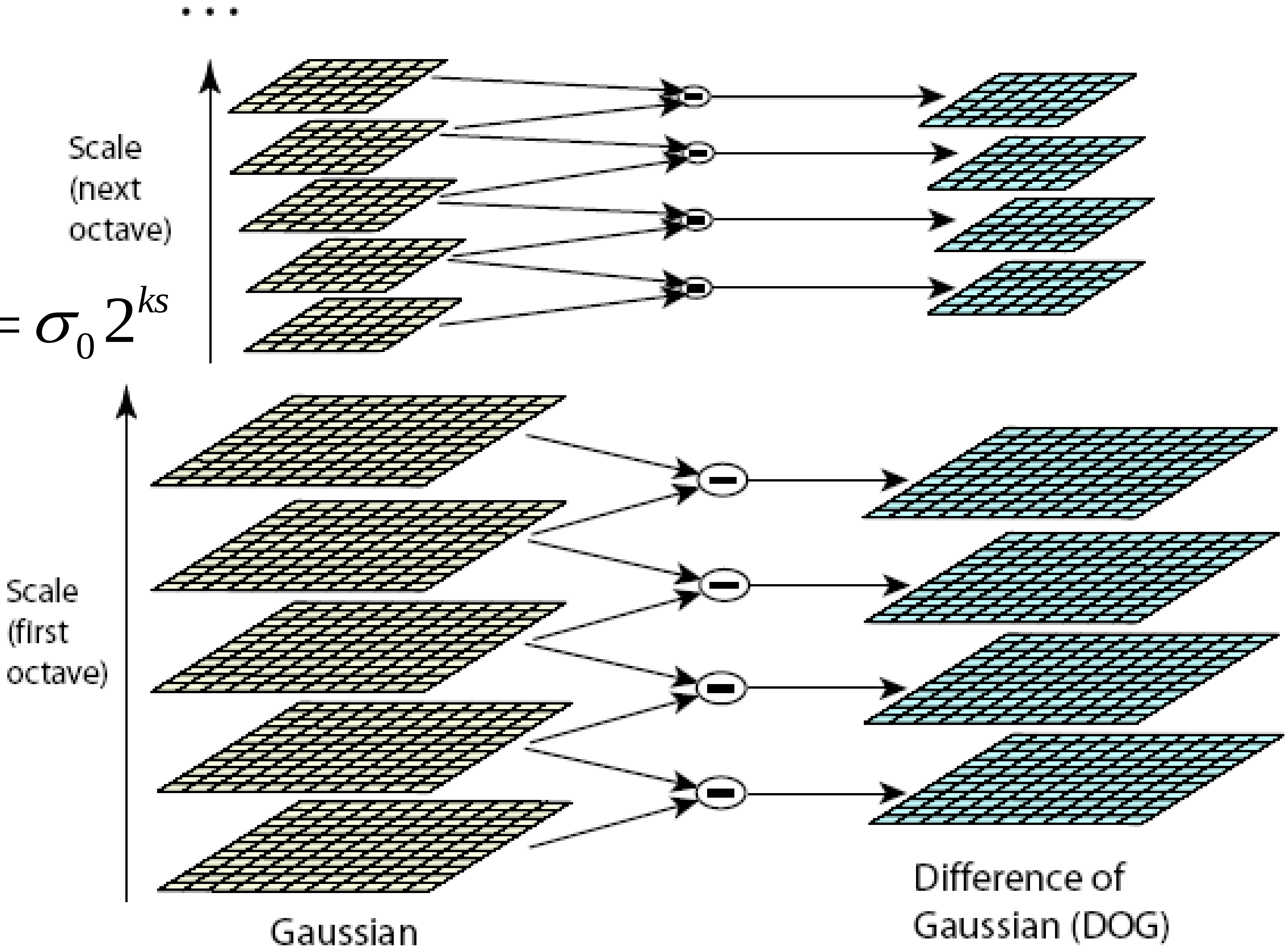
image MxN, filter 2Kx2K

$\downarrow$

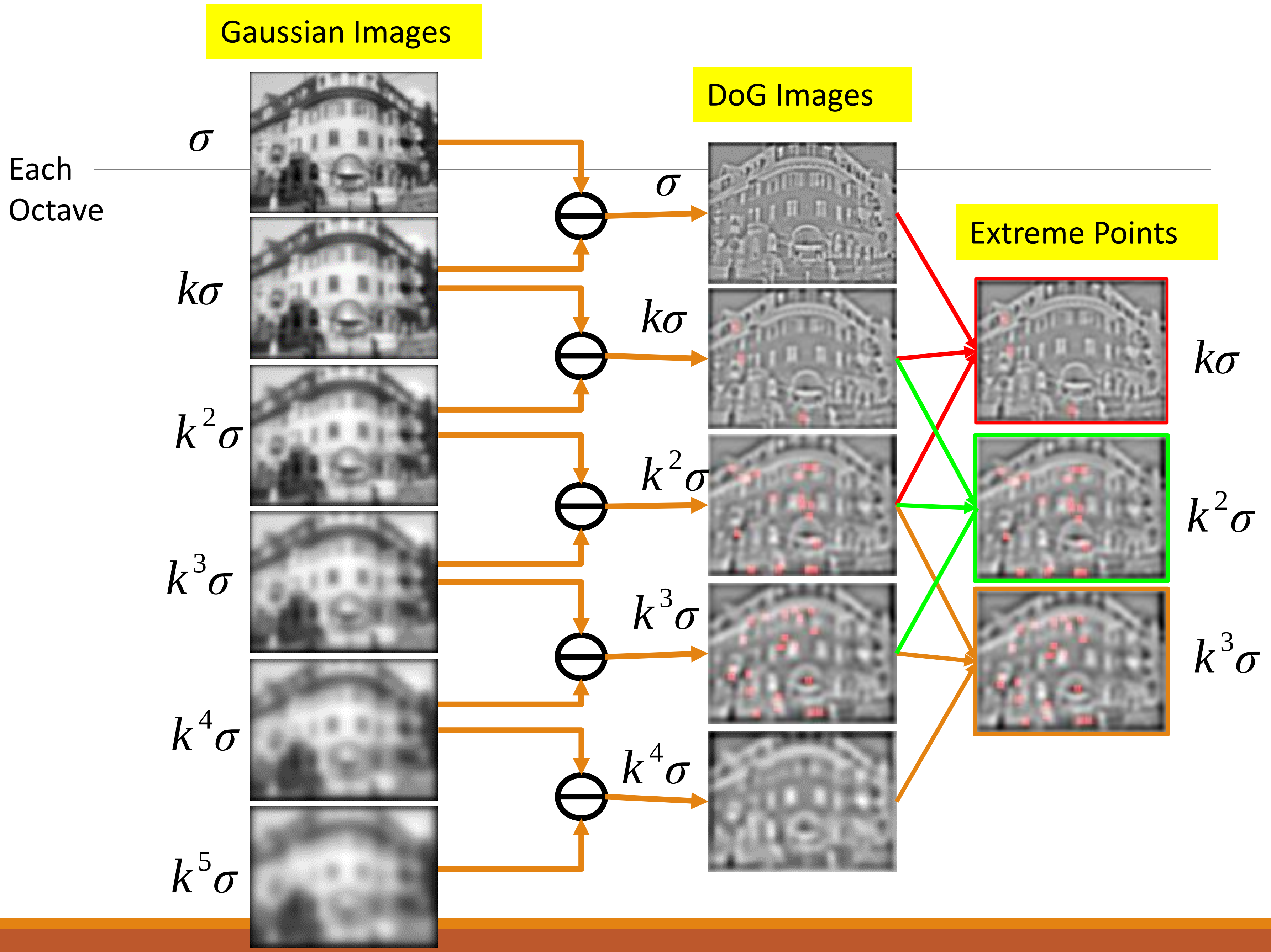
image M/2xN/2, filter, KxK



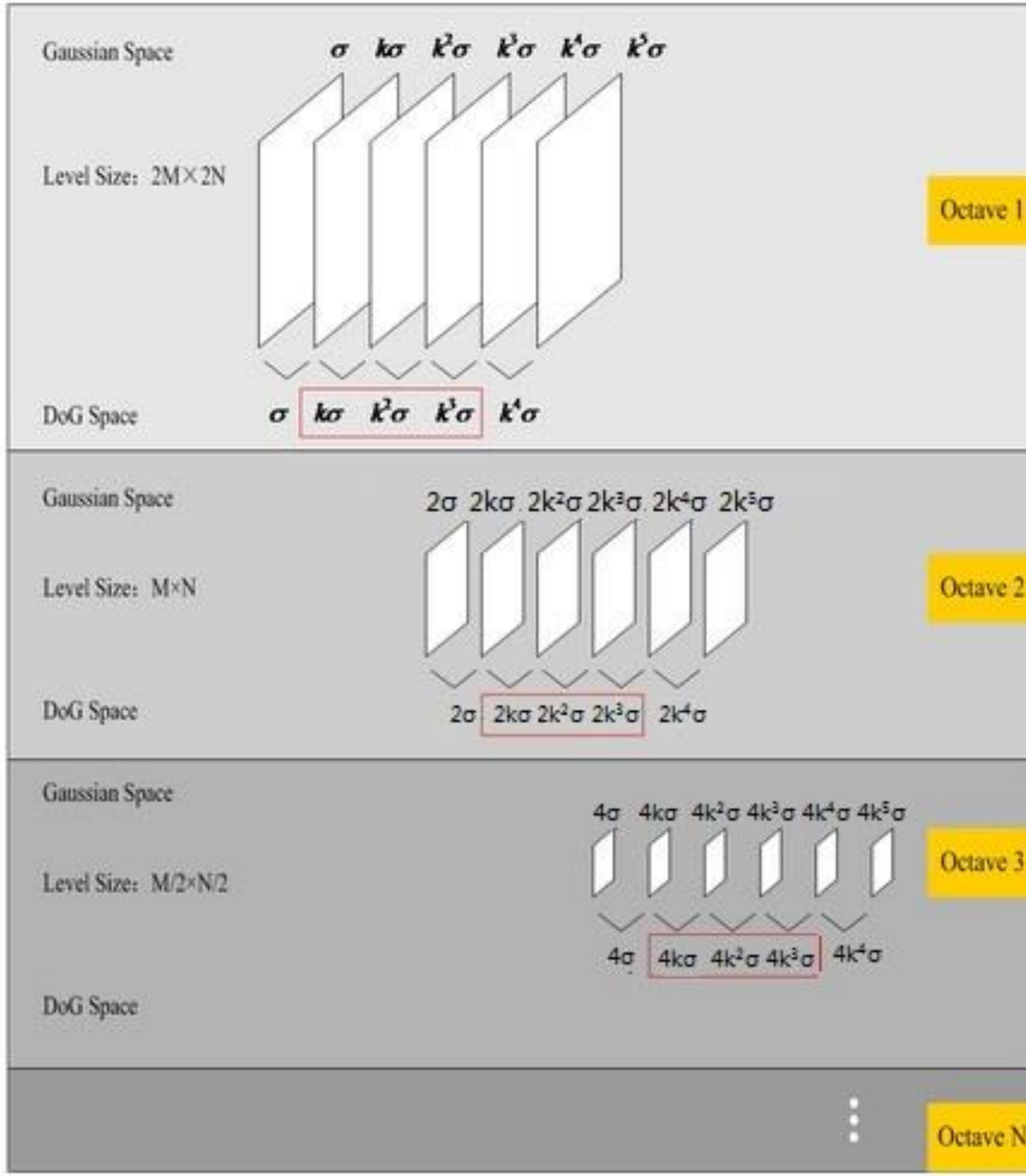
$$\sigma_{o,s} = \sigma_0 2^{ks}$$



$$\sigma_{o,s} = \sigma_0 2^{o+ks}$$

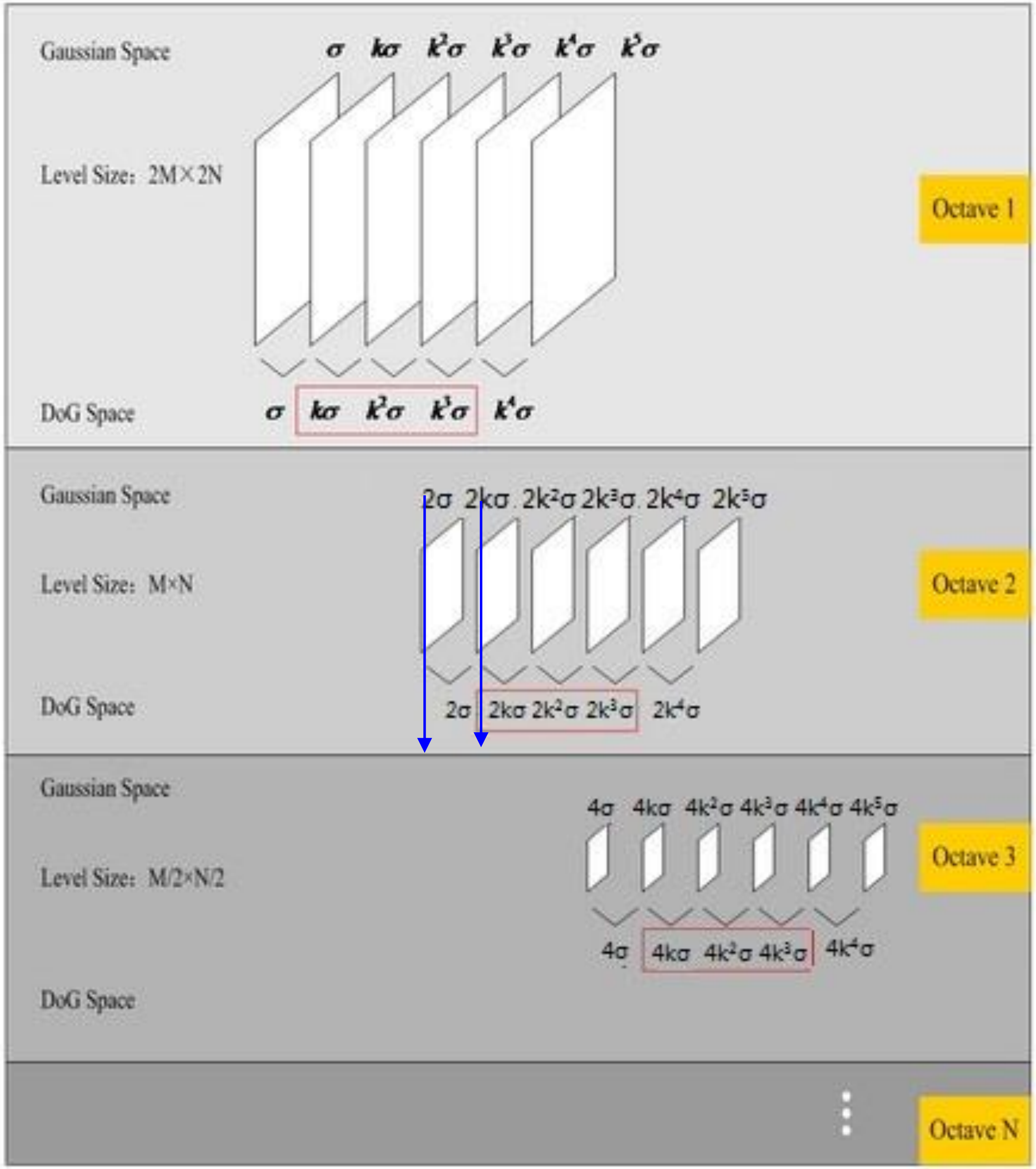


Gaussian Images
DoG Images
Extreme Points
Each Octave
σ
$k\sigma$
$k^2\sigma$
$k^3\sigma$
$k^4\sigma$
$k^5\sigma$
σ
$k\sigma$
$k^2\sigma$
$k^3\sigma$
$k^4\sigma$
$k\sigma$
$k^2\sigma$
$k^3\sigma$



Gaussian Space
σ $k\sigma$ $k^2\sigma$ $k^3\sigma$ $k^4\sigma$ $k^5\sigma$
Level Size：2M×2N
Octave 1
DoG Space
σ $k\sigma$ $k^2\sigma$ $k^3\sigma$ $k^4\sigma$
Gaussian Space
2σ $2k\sigma$ $2k^2\sigma$ $2k^3\sigma$ $2k^4\sigma$ $2k^5\sigma$
Level Size：M×N
Octave 2
DoG Space
2σ $2k\sigma$ $2k^2\sigma$ $2k^3\sigma$ $2k^4\sigma$
Gaussian Space
4σ $4k\sigma$ $4k^2\sigma$ $4k^3\sigma$ $4k^4\sigma$ $4k^5\sigma$
Level Size：M/2×N/2
Octave 3
4σ $4k\sigma$ $4k^2\sigma$ $4k^3\sigma$ $4k^4\sigma$
DoG Space
Octave N

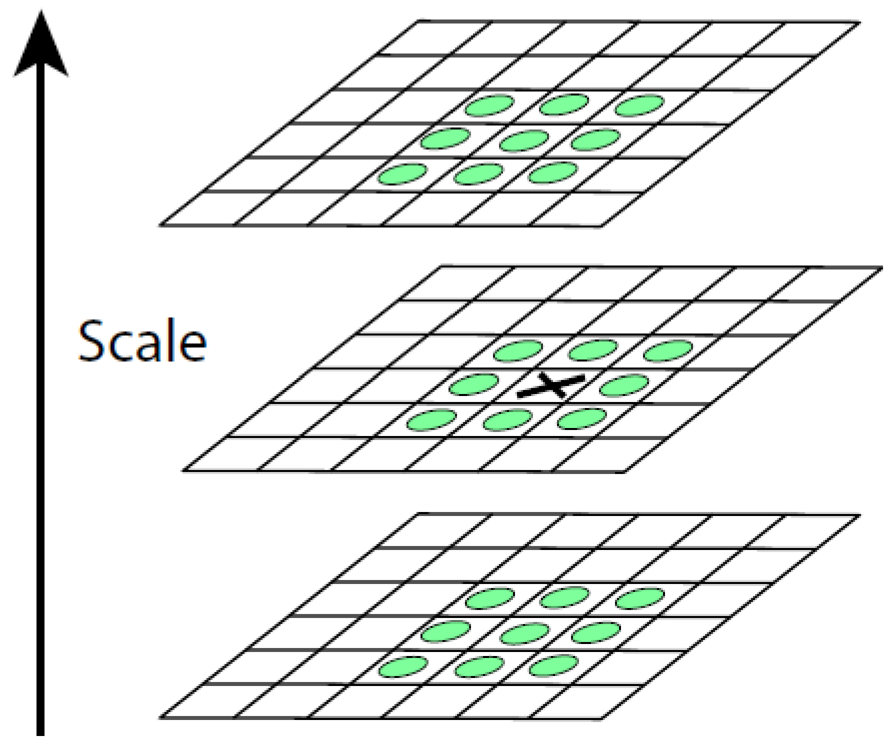
Resample the Gaussian image that has twice the initial value of σ by taking every second pixel in each row and column.



Local Extrema Detection

Maxima and minima

Compare x with its 26 neighbors at 3 scales



Frequency of sampling in scale

s: intervals in each octave of scale space ($\sigma_0 \rightarrow 2\sigma_0$)

- k=2^{1/s}

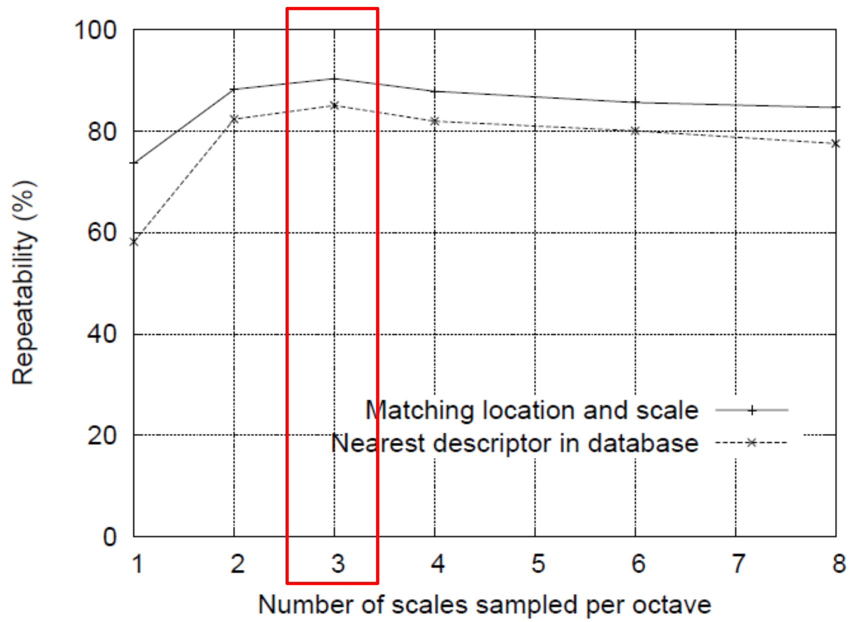
$$\sigma_{o,s} = \sigma_0 2^o k^s$$

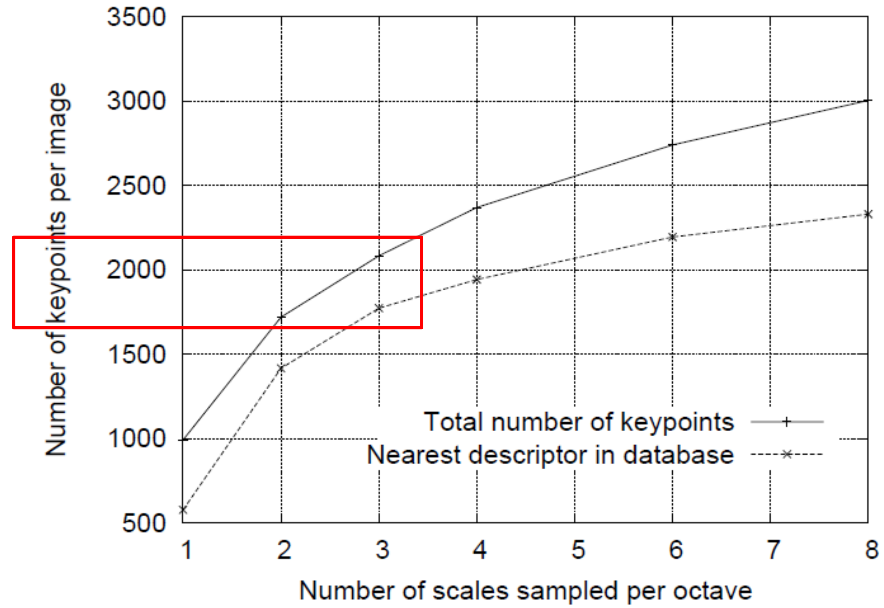
In order to cover a complete octave for extrema detection

- S = s+3 Gaussian images are produced for each octave
  - s: {-1,S+1}
- s+2 DoG images
- s scales for extrema detection

Frequency of Sampling in Scale

s=3

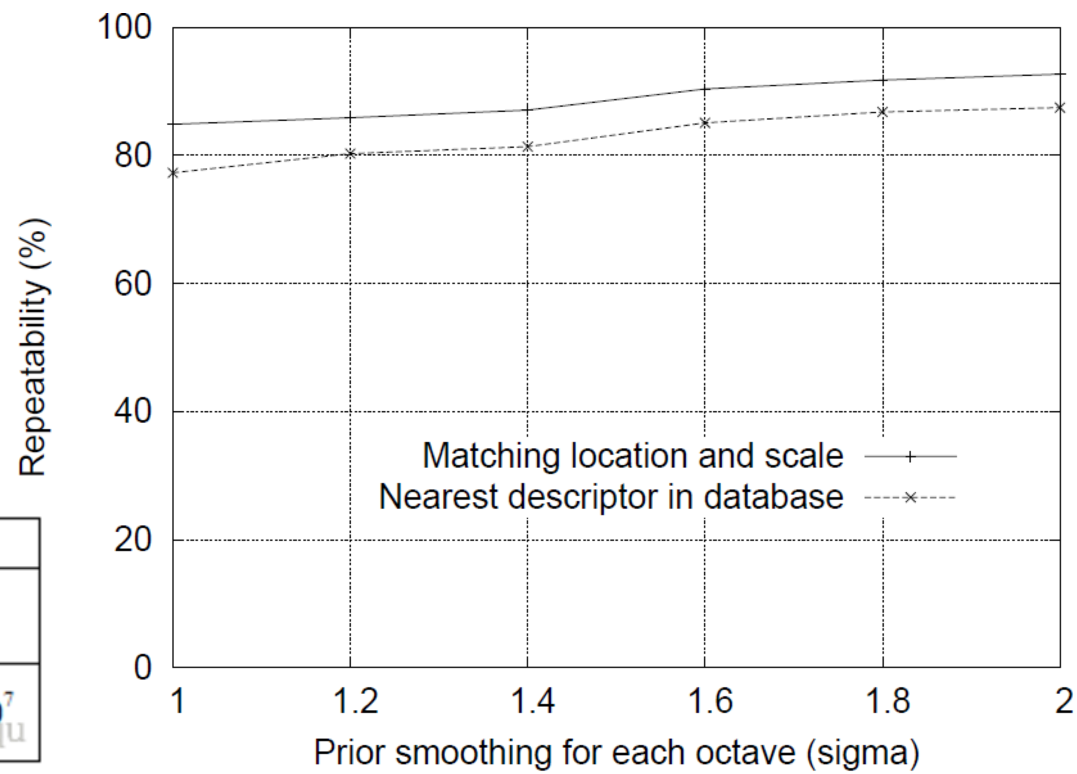




Frequency of Sampling in Domain

Trade-off between sampling frequency and rate of detection

σ=1.6



	LoG	DoG
乘法次数	$1.773936*10^8$	$5.5452*10^7$
加法次数	$1.751424*10^8$	$5.27916*10^7$

Frequency of Sampling in Domain

While pre-smooth image, discarding the highest spatial frequencies

Double the size of input image using linear interpolation as the first level of the pyramid

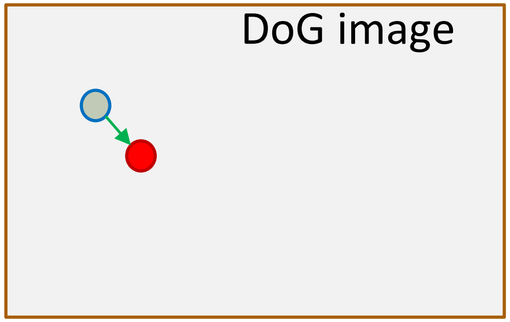
- Blur the original image at least with sigma=0.5 to prevent significant aliasing
- Increasing the number of stable keypoints by a factor of~4

Accurate Keypoint Localization

Derivatives D at the sample point (x,y,sigma) with offset x

Location of $D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^\mathbf{T} \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}$

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}$$



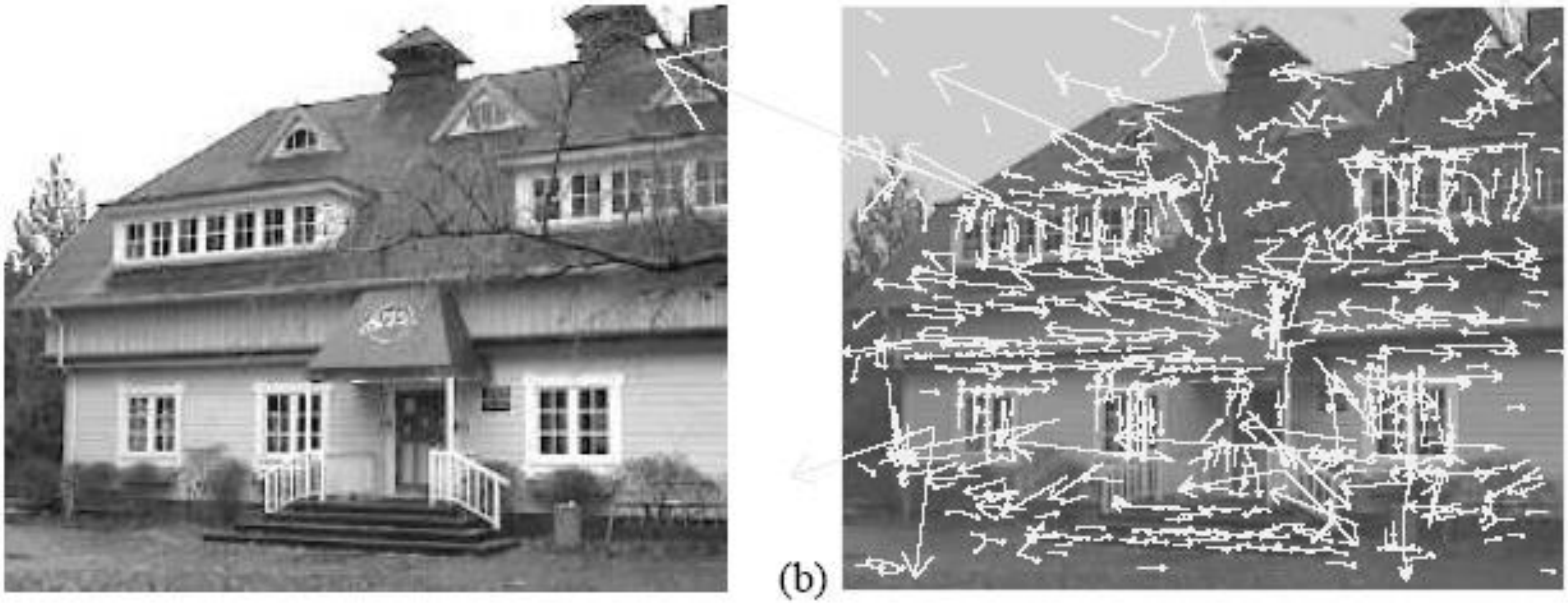
Eliminating unstable keypoint

If x^ > o.5 in any dimension, closer to a different sample point

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D}{\partial \mathbf{x}}^T \hat{\mathbf{x}}$$

Discard extremum that $\left|D(\hat{\mathbf{x}})\right| < 0.03$

Eliminating unstable keypoint



(a) (b)

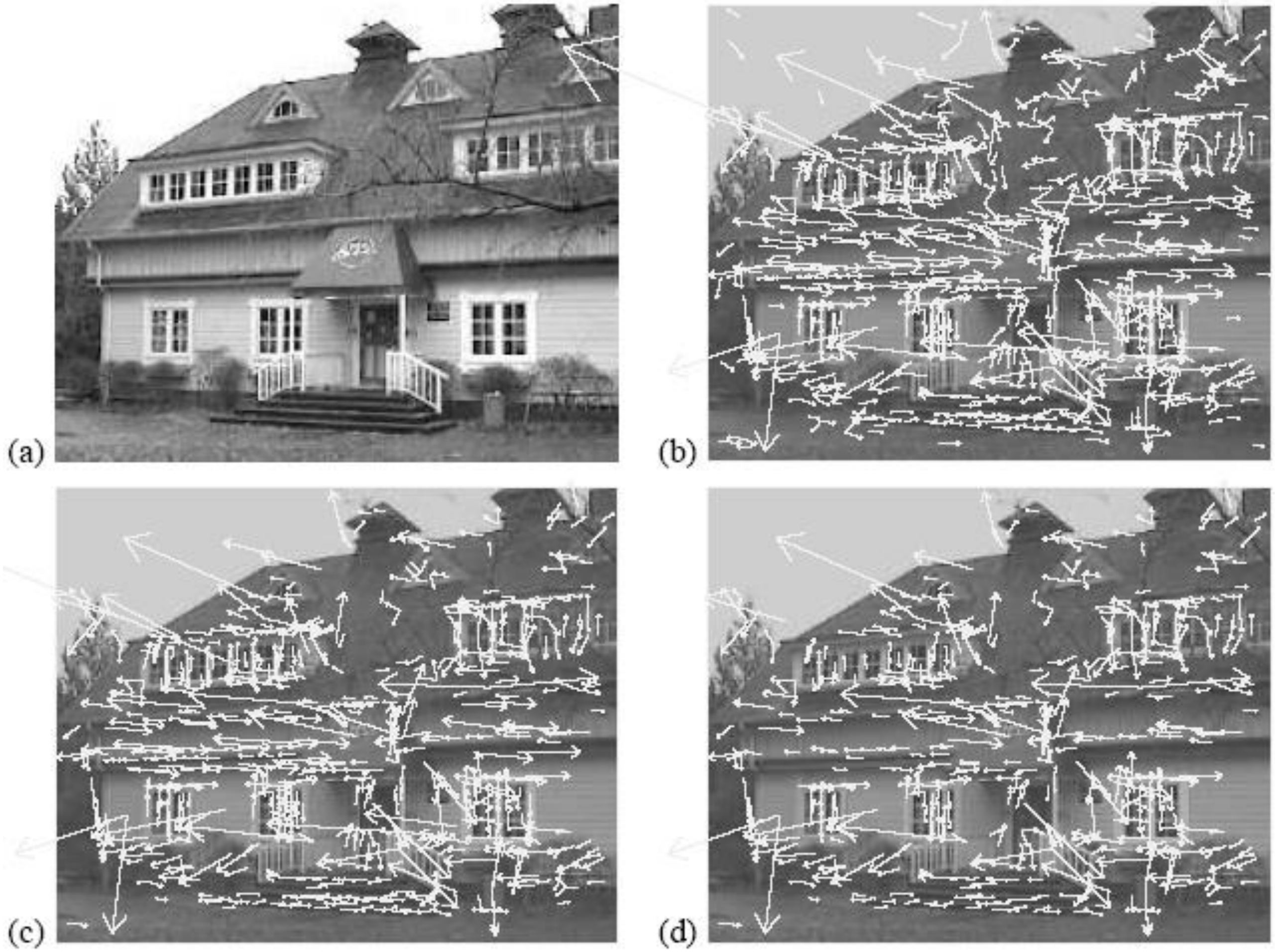


Figure 5: This fi gure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The fi nal 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

Eliminating Edge Responses

Motivation

- DoG aims to detect “blob”.
- DoG function have a strong response along edges.
- Remove such key points by Hessian Matrix analysis

Hessian matrix

- Formulation H=A

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \qquad A = \sum_i \omega(\mathbf{x}_i) \begin{bmatrix} I_x^2(\mathbf{x}_i) & I_x(\mathbf{x}_i) I_y(\mathbf{x}_i) \\ I_x(\mathbf{x}_i) I_y(\mathbf{x}_i) & I_y^2(\mathbf{x}_i) \end{bmatrix}$$

Eliminating Edge Responses

$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$

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

$$\alpha = r\beta$$

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

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}.$$

r=10

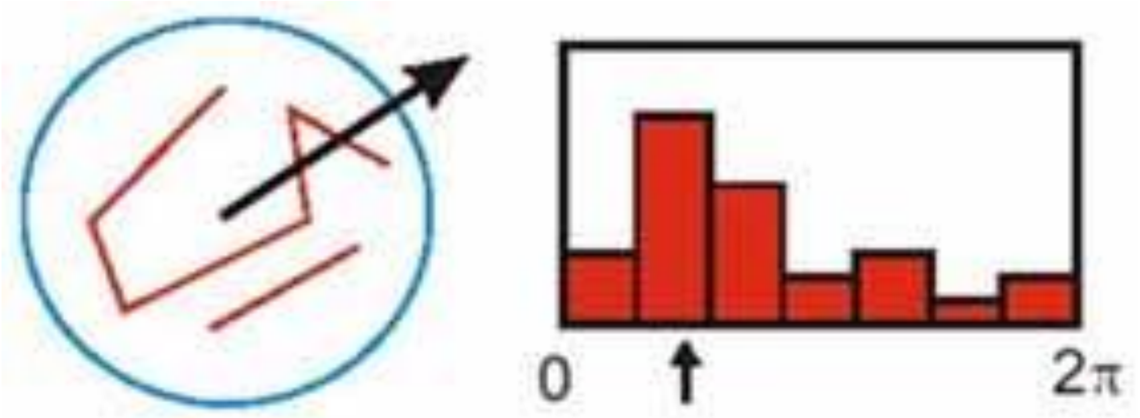
Orientation

Gradient and angle:

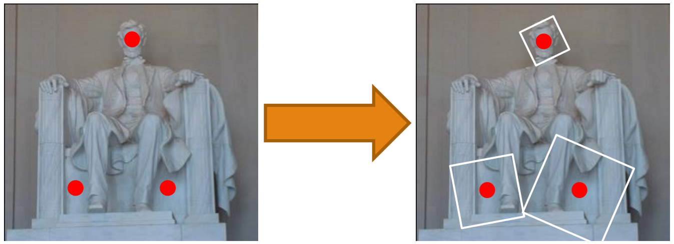
$$m(x,y)=\sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2}$$

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

Orientation selection

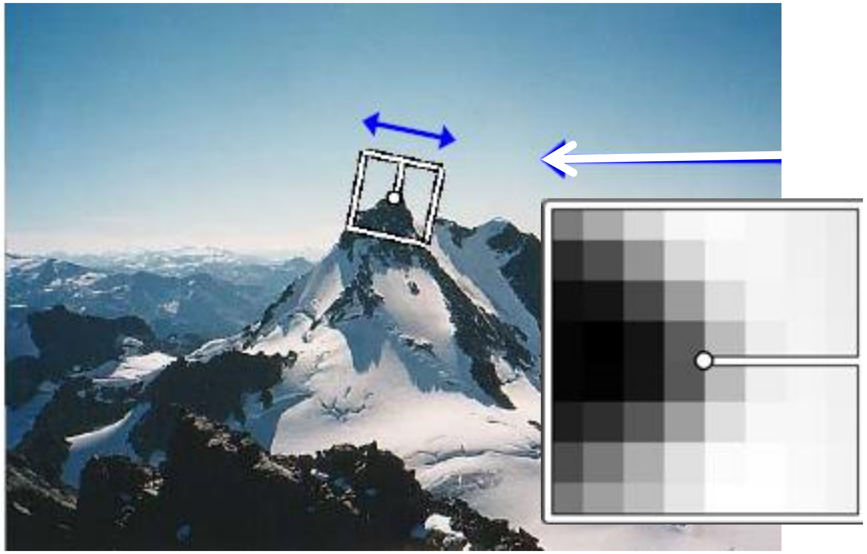


SIFT Descriptor



SIFT Descriptor

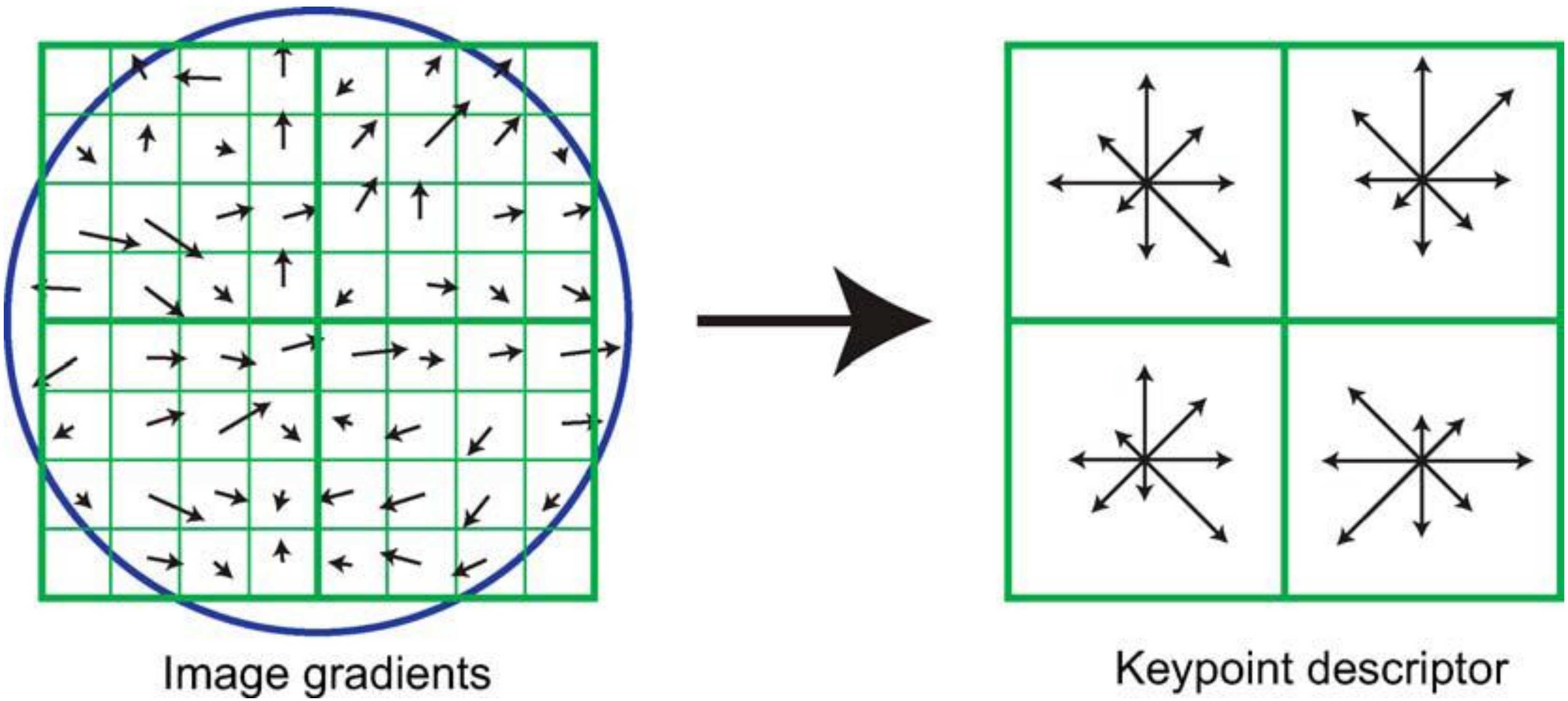
Making descriptor rotation invariant



- Rotate patch according to its dominant gradient orientation
- This puts the patches into a canonical orientation.

SIFT Descriptor

Use histograms to bin pixels within sub-patches according to their orientation.



Summary of SIFT Feature

Descriptor: 128-D

- 4 by 4 patches, each with 8-D gradient angle histogram:

  $4 \times 4 \times 8 = 128$

- Normalized to reduce the effects of illumination change.

Position: (*x*, *y*)

- Where the feature is located at.

Scale

- Control the region size for descriptor extraction.

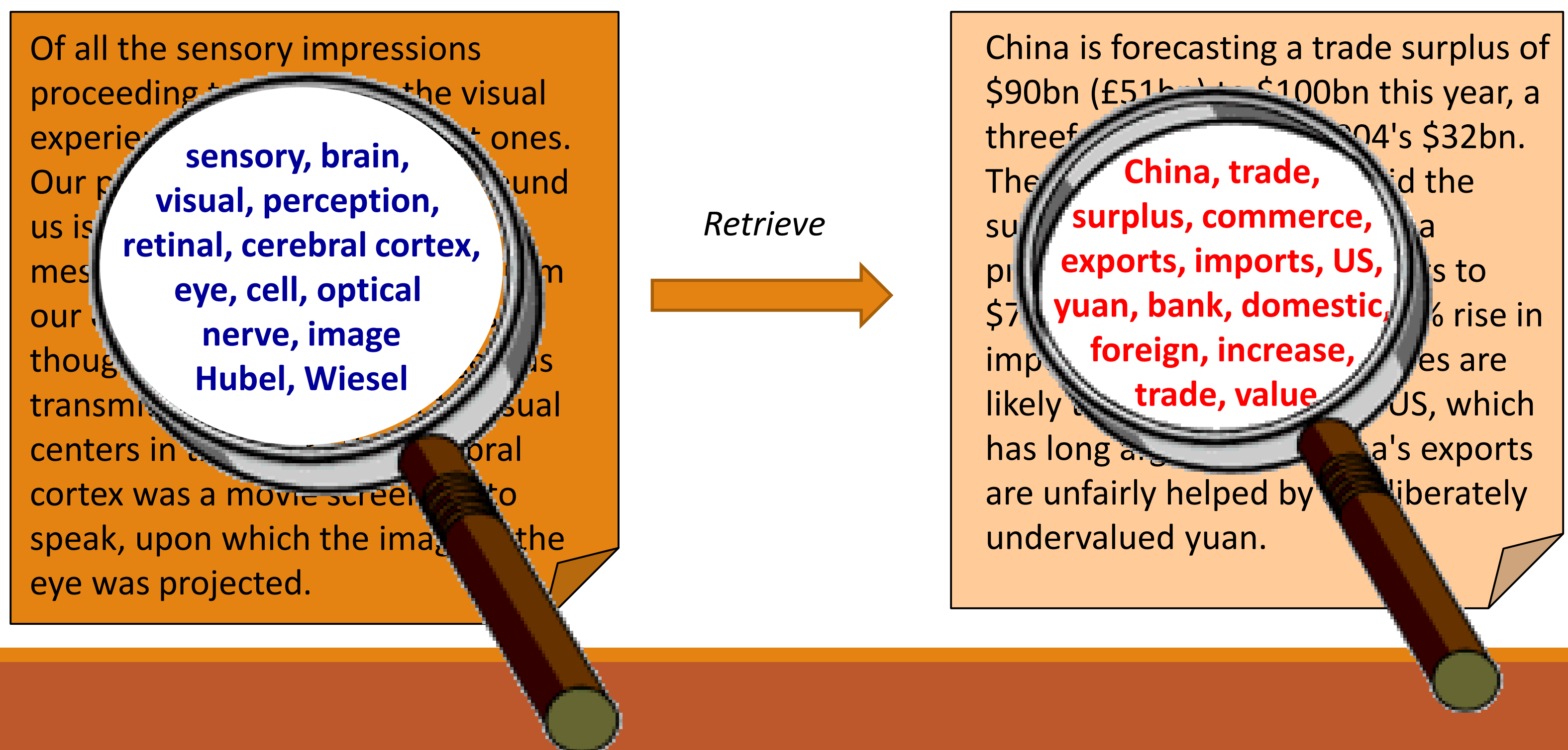
Orientation

- To achieve rotation-invariant descriptor.

Application in Image Search

- Text Words in Information Retrieval (IR)
  - ***Compactness***
  - ***Descriptiveness***

Bag-of-Word model



Conclusion of SIFT

Merit

- Desired property in invariance in changes of scale, rotation, illumination, *etc*.
- Highly distinctive and descriptive in local patch.
- Especially effective in rigid object representation.

Drawback

- Time consuming for extraction
  - About one second in average for an image with size of 400 by 400.
- Poor performance for un-rigid object.
  - Such as human face, animal, *etc*.
- May fail to work in severe affine distortion.
  - The local patch is a circle, instead of an ellipse adjusted to the affine distortion.