

SIFT: Detector and Descriptor

SIFT: David Lowe, UBC



D. Lowe. Distinctive image features from scale-invariant key points ., International Journal of Computer Vision 2004.

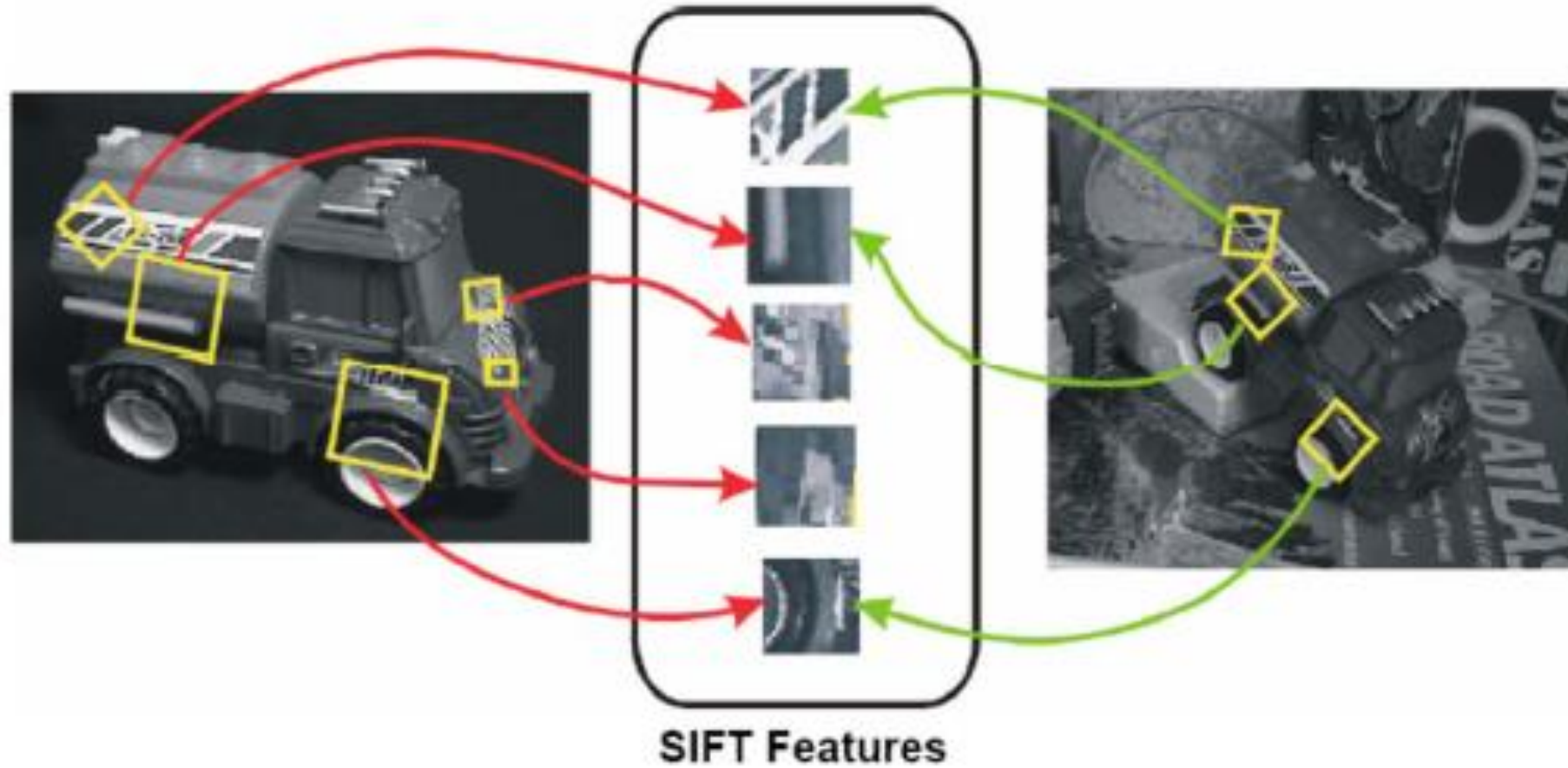
SIFT - Key Point Extraction

- Stands for Scale Invariant Feature Transform
- Patented by university of British Columbia
- Transforms image data into scale invariant coordinates

Goal

- Extract distinctive invariant features
 - Correctly matched against a large database of features from many images
- Invariance to image scale and rotation
- Robustness to
 - Affine (rotation, scale, shear) distortion,
 - Change in 3D viewpoint,
 - Addition of noise,
 - Change in illumination.

Invariant Local Features



Steps for Extracting Key Points

Scale space peak selection

- Potential locations for finding features

Key point localization

- Accurately locating the feature key points

Orientation Assignment

- Assigning orientation to the key points

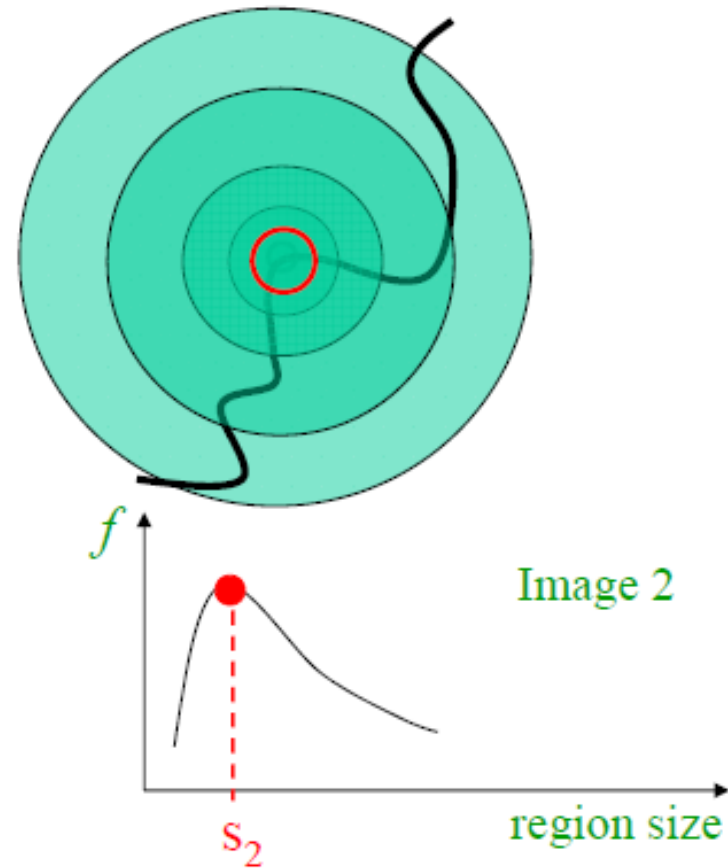
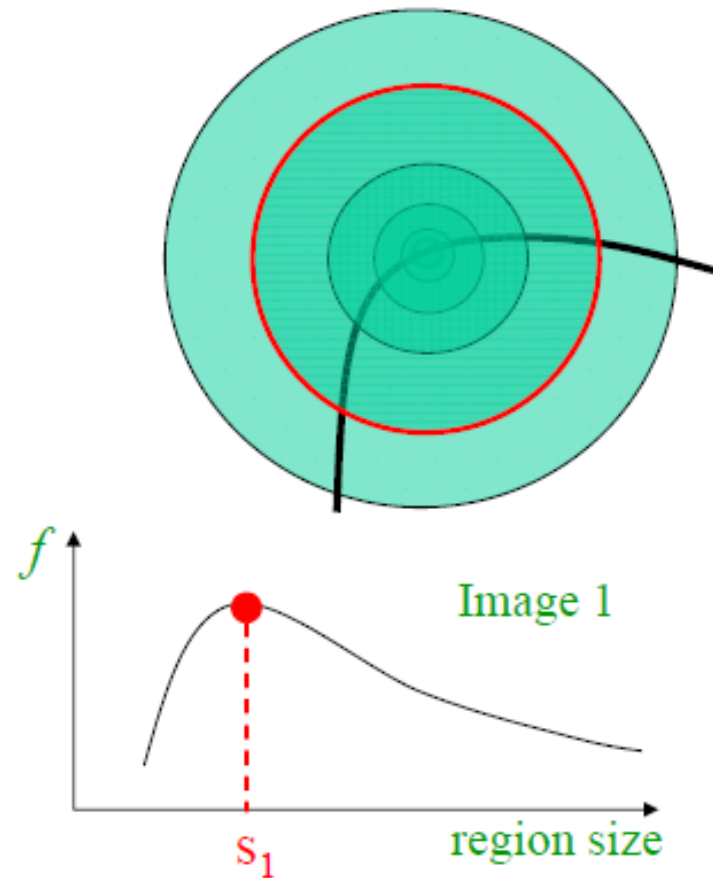
Key point descriptor

- Describing the key point as a vector of size 128 (SIFT Descriptor)

Automatic scale selection

Intuition:

- Find scale that gives local maxima of some function f in both position and scale.



Automatic Scale Selection

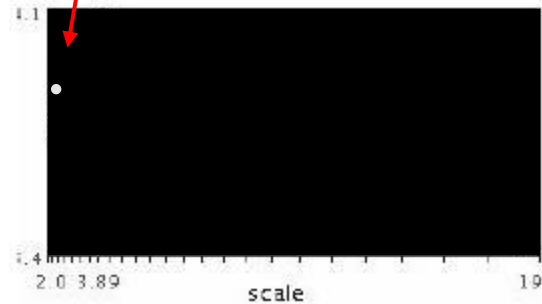


$$f(I_{i_1 \dots i_m}(x, \sigma)) = f(I_{i_1 \dots i_m}(x', \sigma'))$$

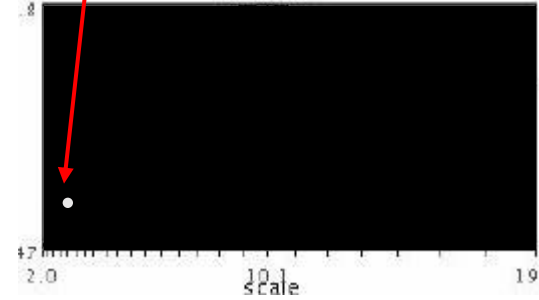
How to find corresponding patch sizes?

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



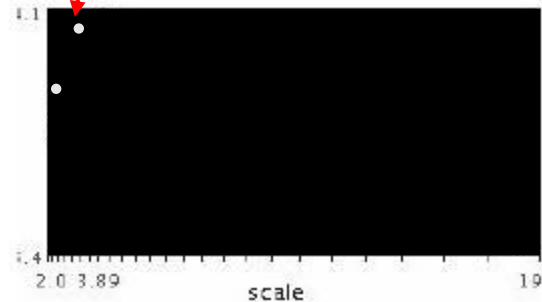
$$f(I_{i_1...i_m}(x, \sigma))$$



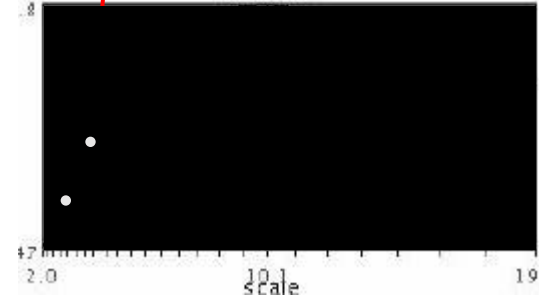
$$f(I_{i_1...i_m}(x', \sigma'))$$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



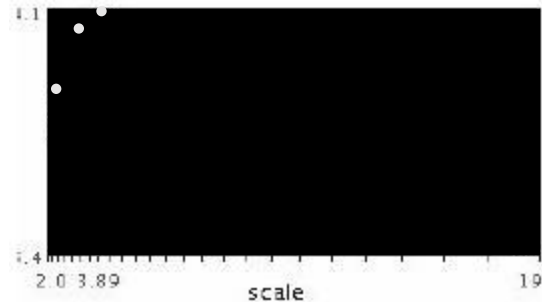
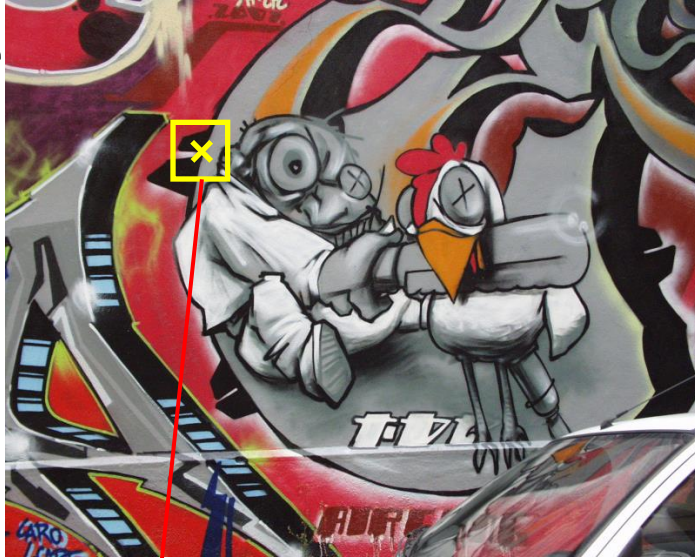
$$f(I_{i_1...i_m}(x, \sigma))$$



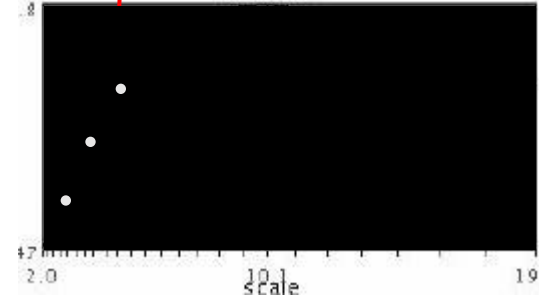
$$f(I_{i_1...i_m}(x', \sigma'))$$

Automatic Scale Selection

- Function re (scale



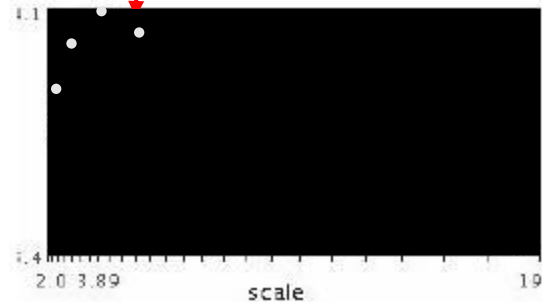
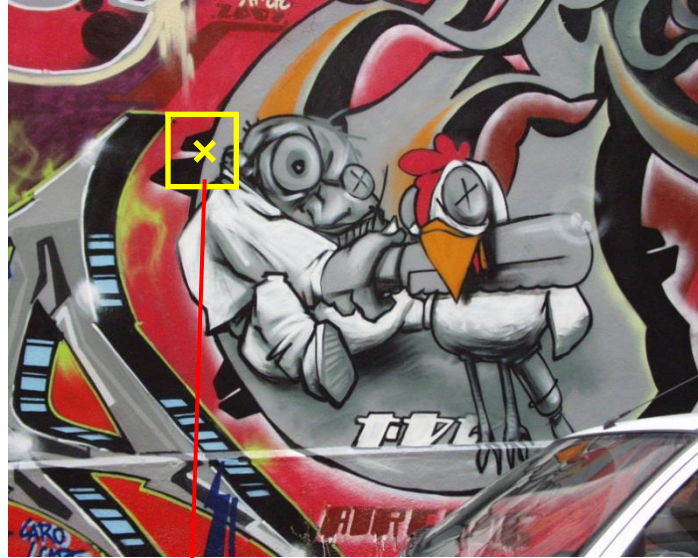
$$f(I_{i_1...i_m}(x, \sigma))$$



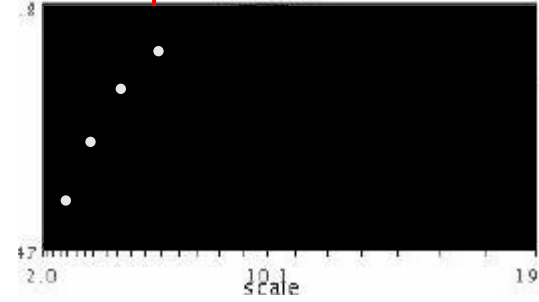
$$f(I_{i_1...i_m}(x', \sigma'))$$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



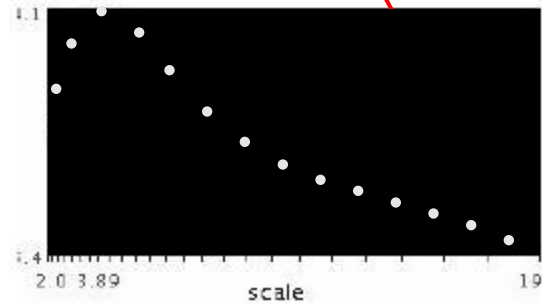
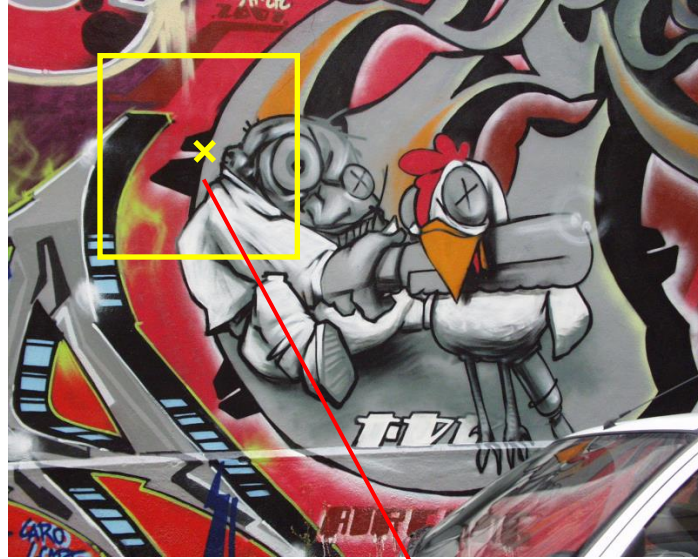
$$f(I_{i_1...i_m}(x, \sigma))$$



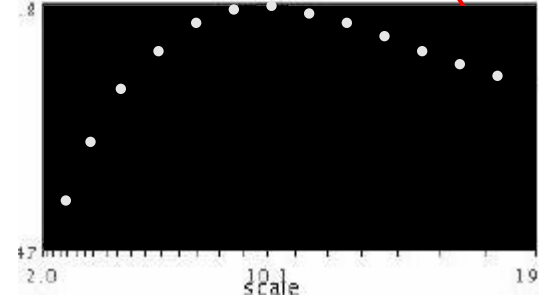
$$f(I_{i_1...i_m}(x', \sigma'))$$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



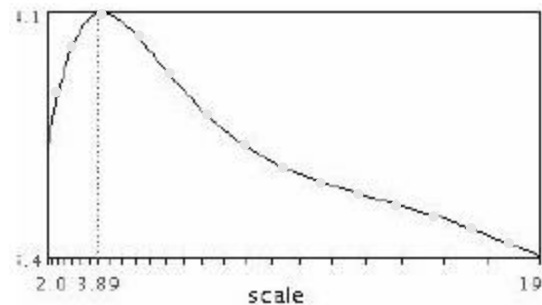
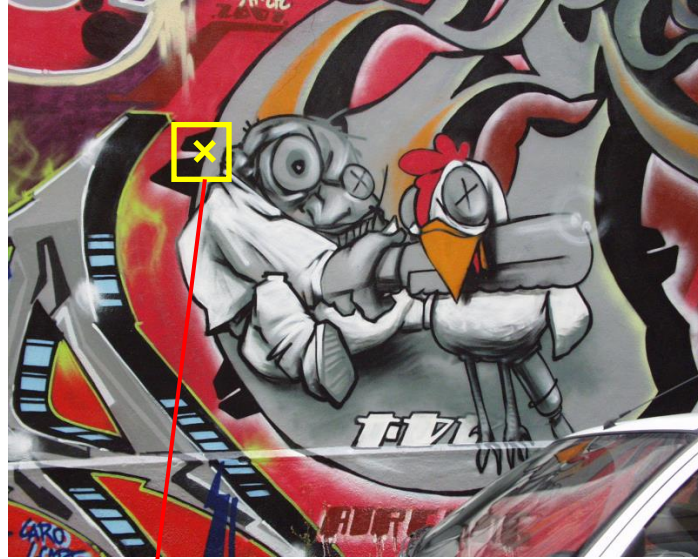
$$f(I_{i_1...i_m}(x, \sigma))$$



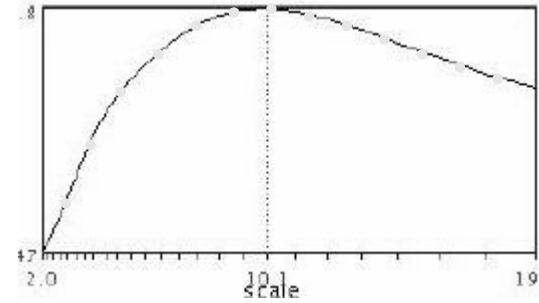
$$f(I_{i_1...i_m}(x', \sigma'))$$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



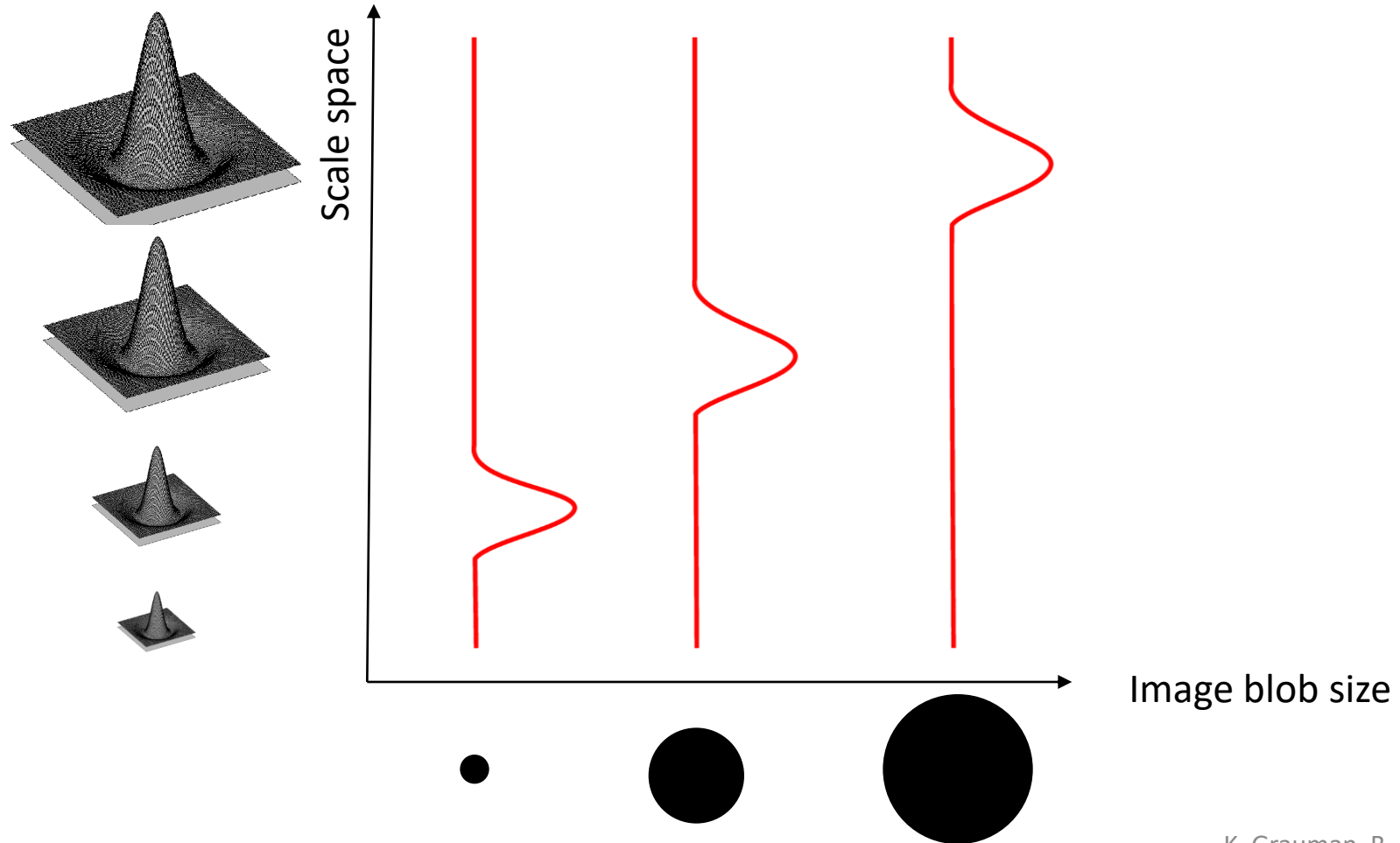
$$f(I_{i_1 \dots i_m}(x, \sigma))$$



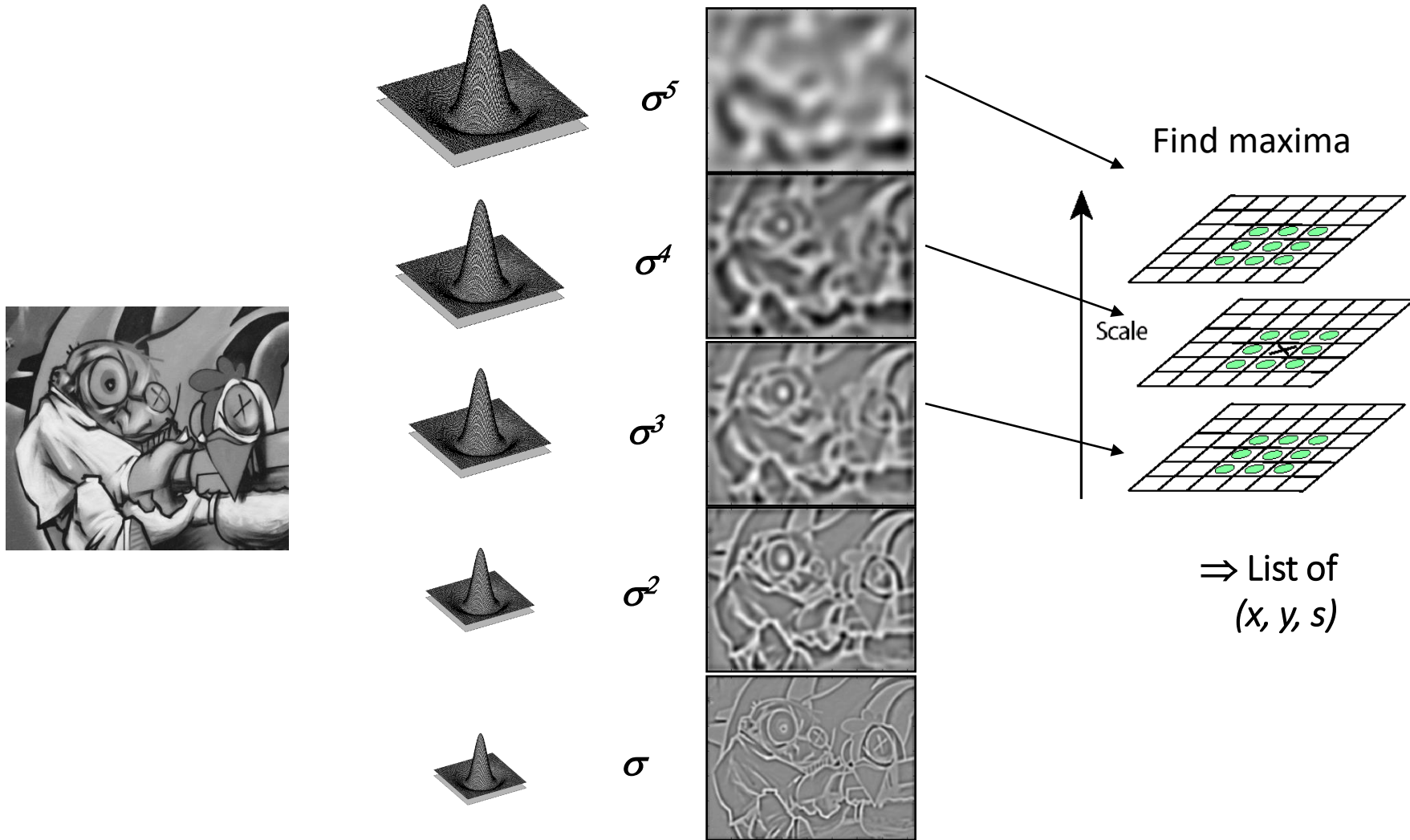
$$f(I_{i_1 \dots i_m}(x', \sigma'))$$

Blob detector

- Laplacian (2^{nd} derivative) of Gaussian (LoG)



Find local maxima in scale space of Laplacian of Gaussian (LoG)



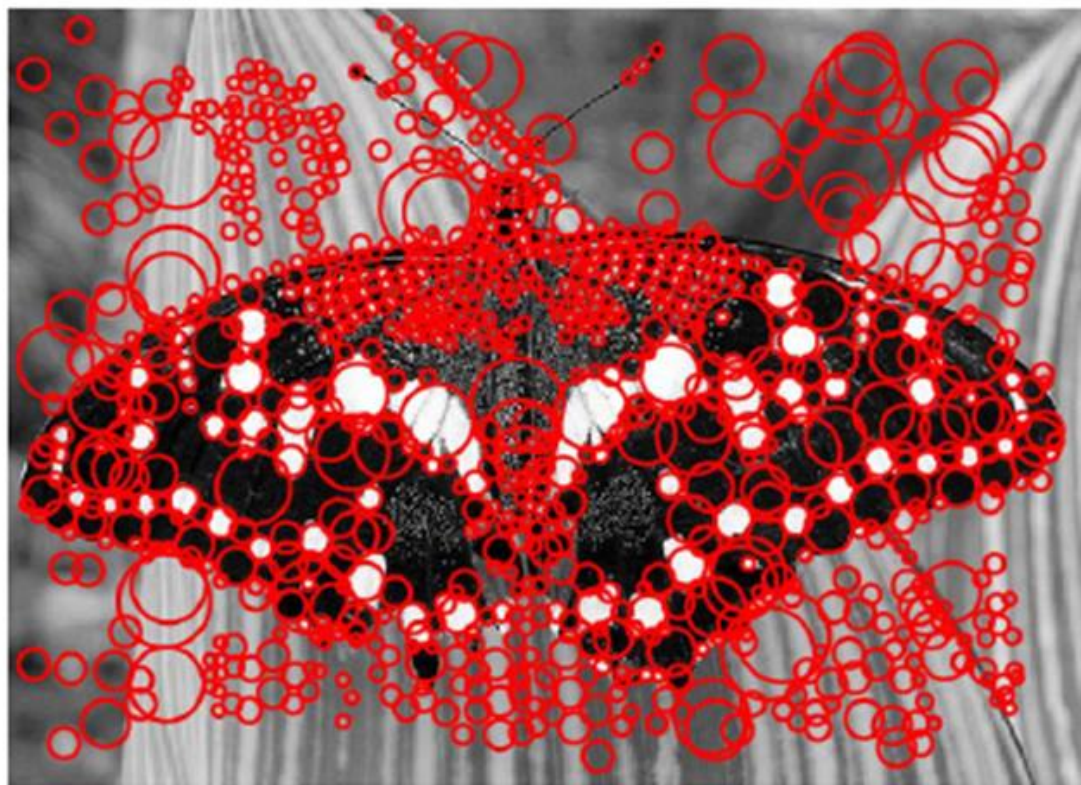
Scale-space blob detector: Example



Scale-space blob detector: Example



sigma = 11.9912



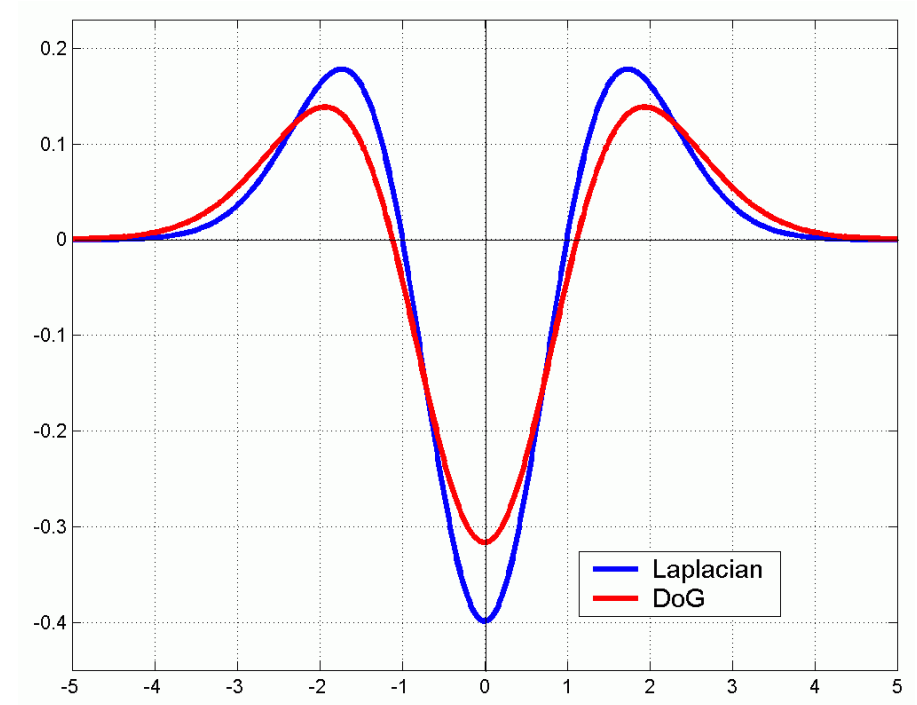
Approximation of LoG by Difference of Gaussians

$$\frac{\partial G}{\partial \sigma} = \sigma \Delta^2 G \quad \text{Heat Equation}$$

$$\sigma \Delta^2 G = \frac{\partial G}{\partial \sigma} = \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma}$$

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

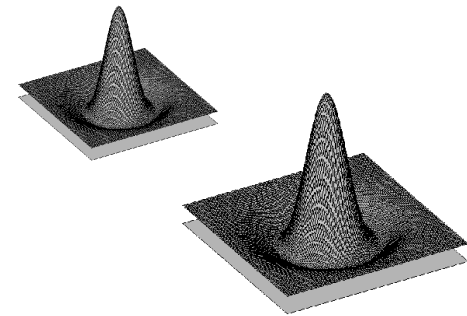
Typical values : $\sigma = 1.6$; $k = \sqrt{2}$



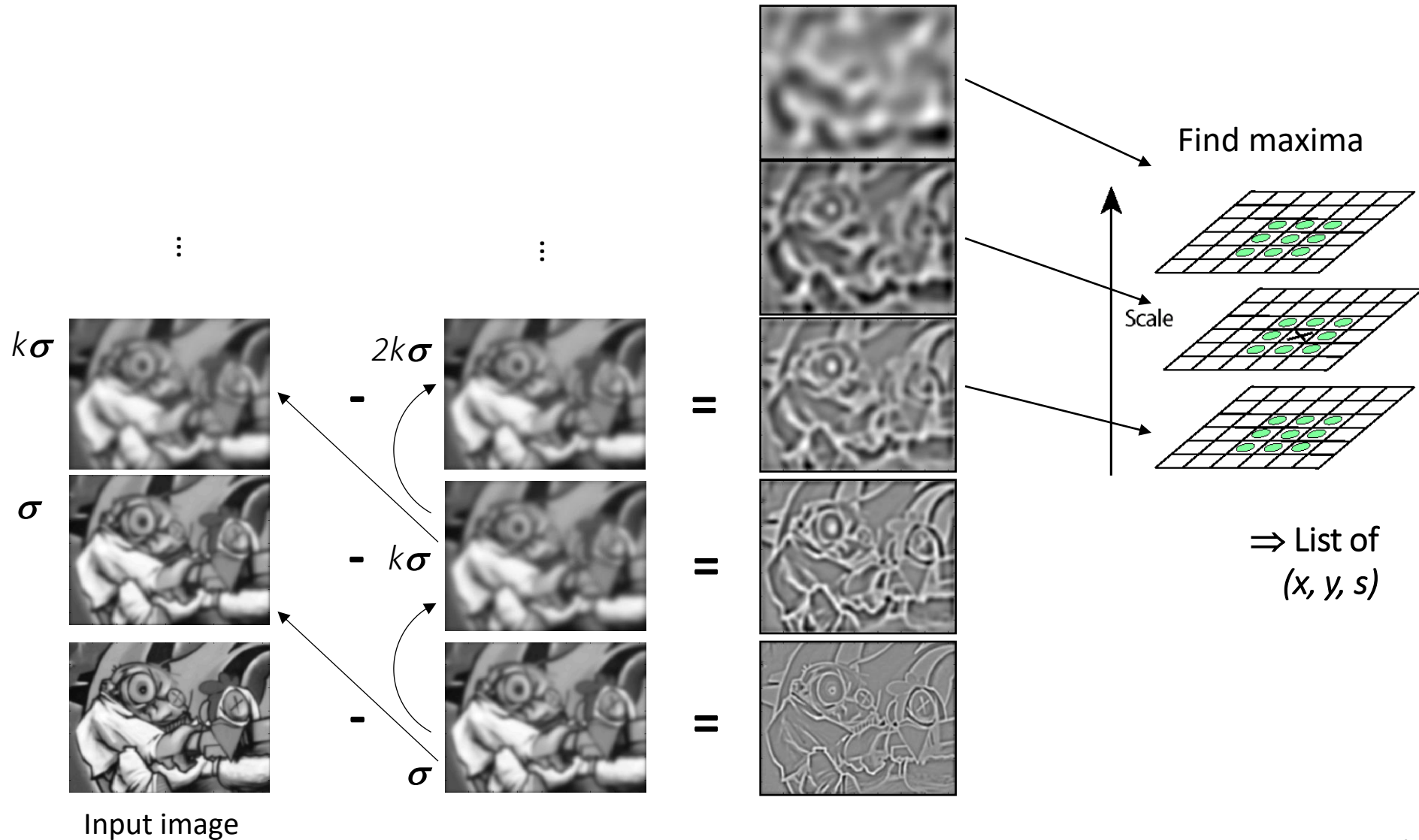
Difference-of-Gaussian (DoG)

Approximate LoG with DoG

1. Blur image with σ Gaussian kernel
2. Blur image with $k\sigma$ Gaussian kernel
3. Subtract 2. from 1.

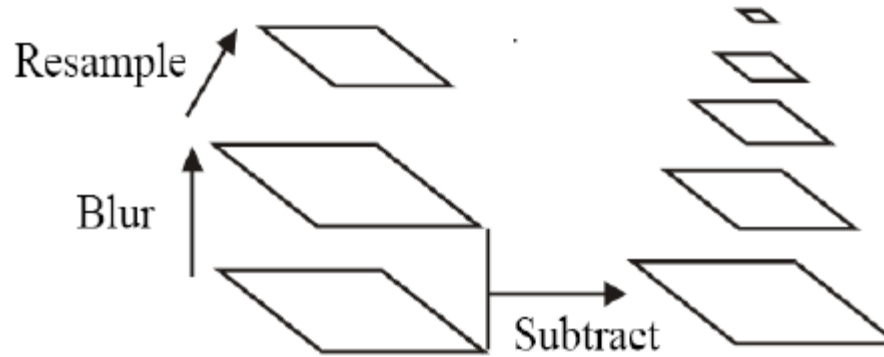


Find local maxima in scale space of Difference of Gaussian



Building a Scale Space

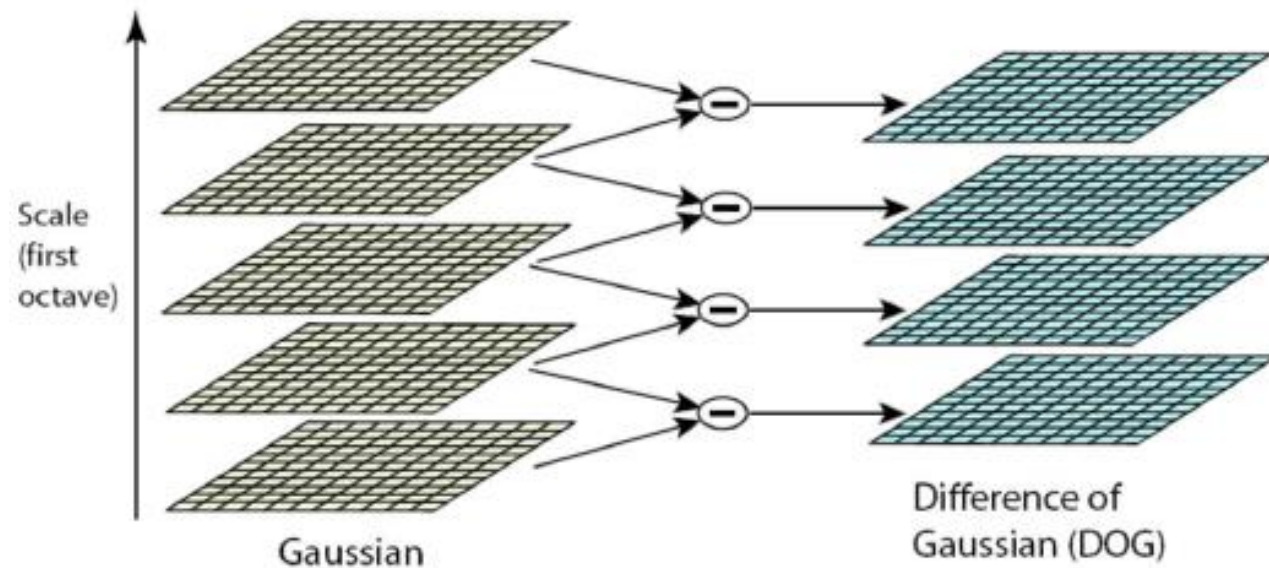
- All scales must be examined to identify scale invariant features
- An efficient function is to compute the Laplacian Pyramid (Difference of Gaussian)



Building a Scale Space

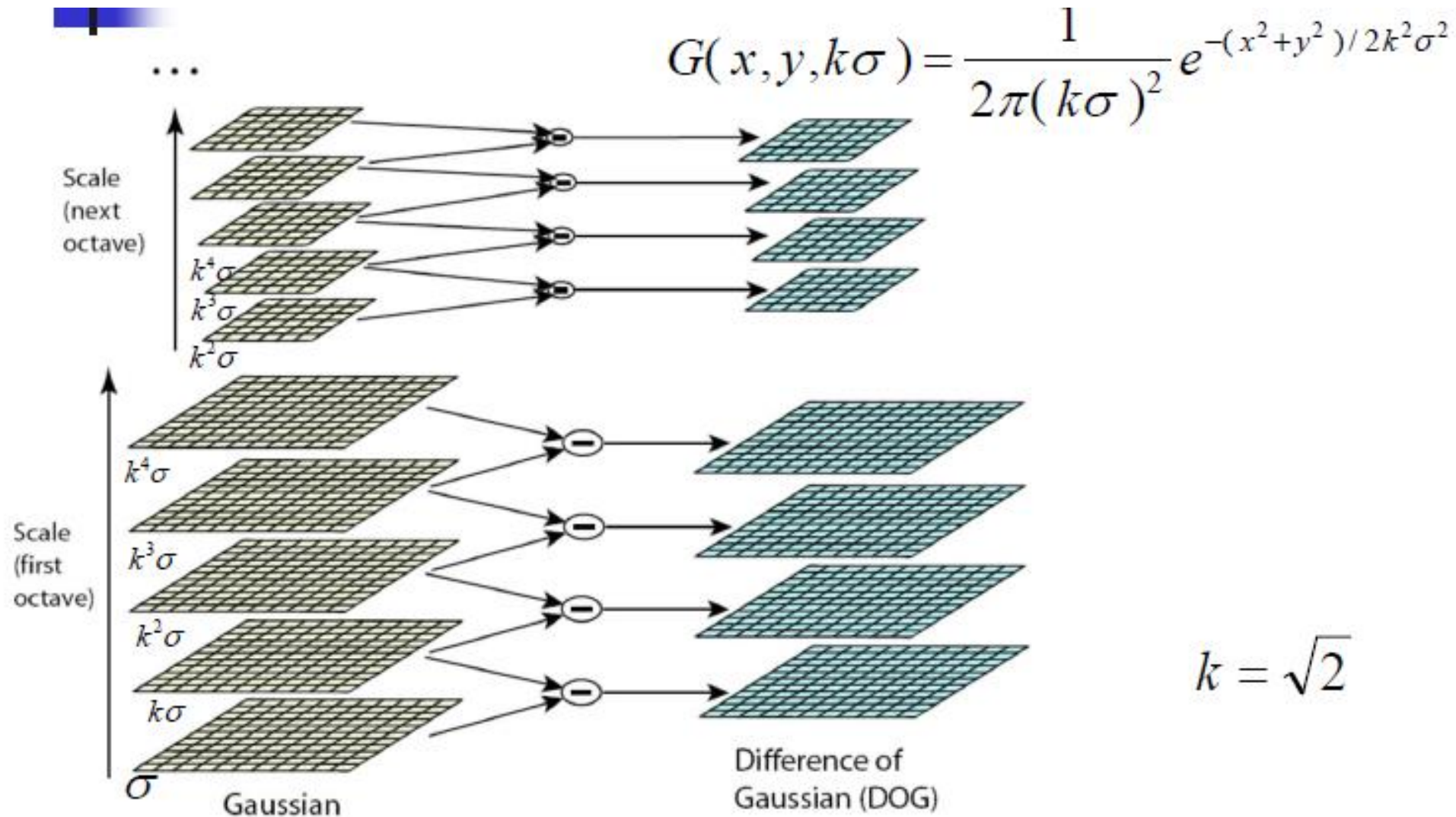


$$G(x, y, k\sigma) = \frac{1}{2\pi(k\sigma)^2} e^{-(x^2+y^2)/2k^2\sigma^2}$$




$$k = \sqrt{2}$$

Building a Scale Space



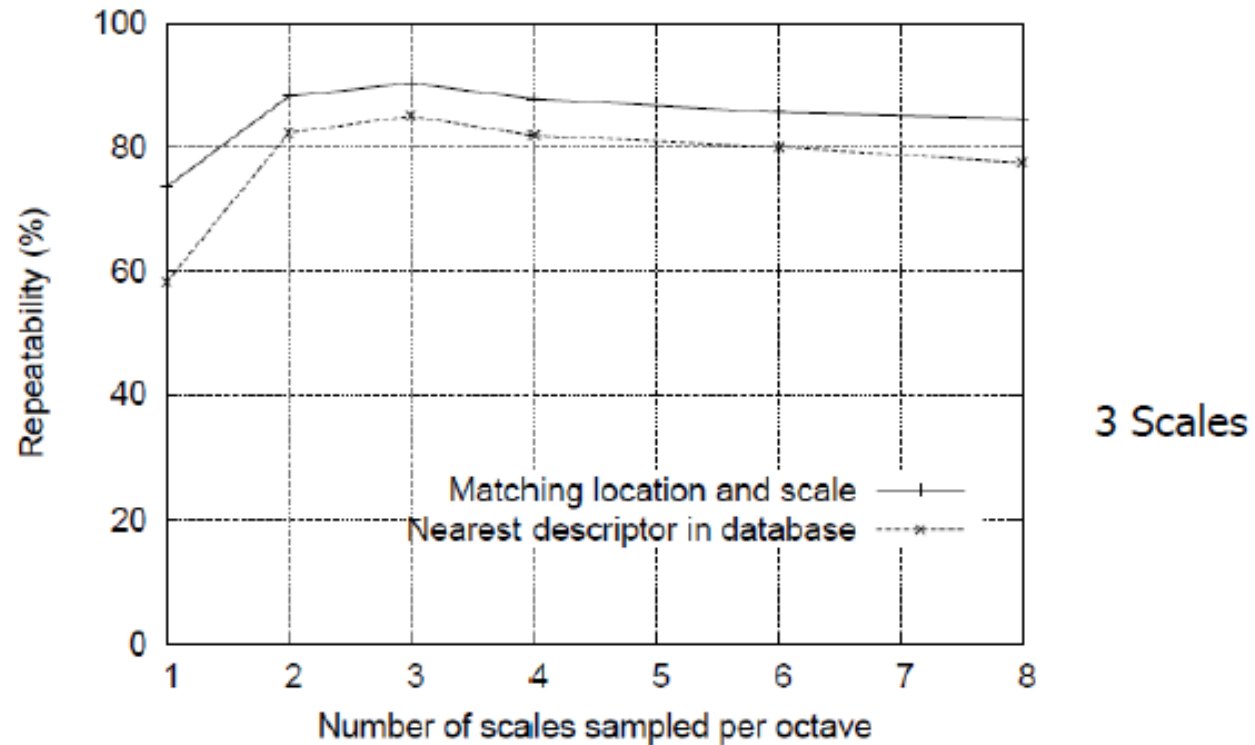
Building a Scale Space

	scale 				
octave	0.707107	1.000000	1.414214	2.000000	2.828427
	1.414214	2.000000	2.828427	4.000000	5.656854
	2.828427	4.000000	5.656854	8.000000	11.313708
	5.656854	8.000000	11.313708	16.000000	22.627417

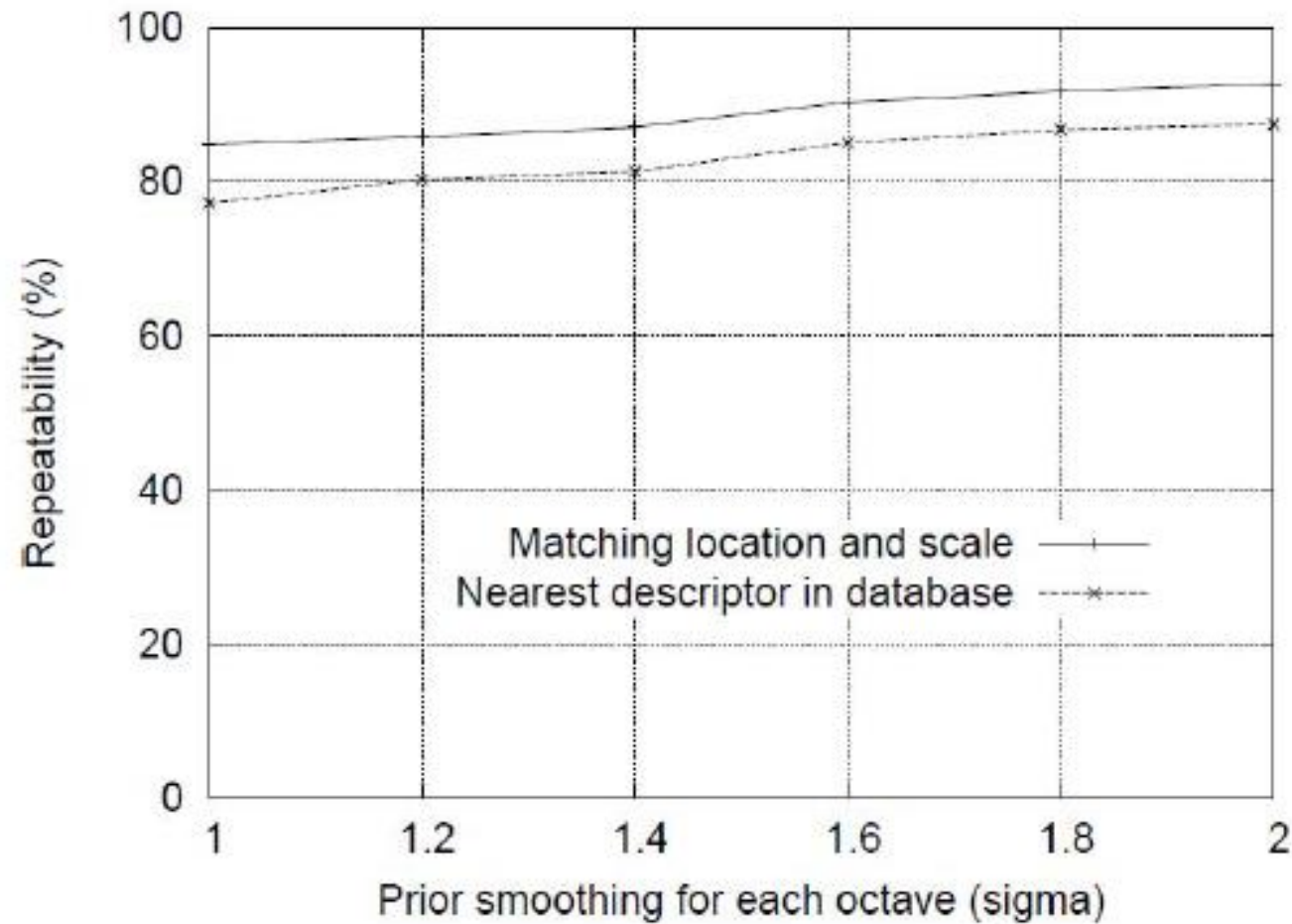
$$\sigma = .707187.6; k = \sqrt{2}$$

How many scales per octave?

- A collection of 32 real images drawn from a diverse range, including
 - outdoor scenes, human faces, aerial photographs, and industrial
- Each image was then subject to a range of transformations:
 - rotation, scaling, affine stretch, change in brightness and addition of image noise.



Initial value of sigma



1.6

Steps for Extracting Key Points

Scale space peak selection

- Potential locations for finding features

Key point localization

- Accurately locating the feature key points

Orientation Assignment

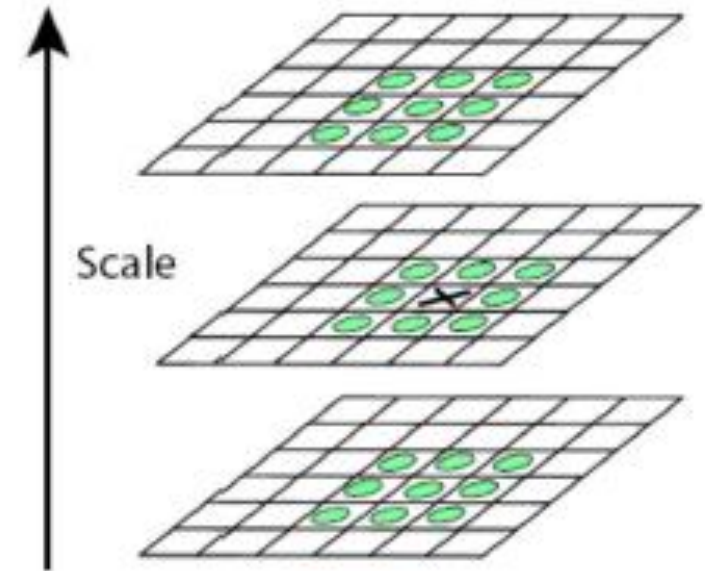
- Assigning orientation to the key points

Key point descriptor

- Describing the key point as a vector of size 128 (SIFT Descriptor)

Scale Space Peak Detection

- Compare a pixel (**X**) with 26 pixels in current and adjacent scales (Green Circles)
- Select a pixel (**X**) if larger/smaller than all 26 pixels
- Large number of extrema, computationally expensive
 - Detect the most stable subset of scales



Key Point Localization

- Candidates are chosen from extrema detection
- There are lot of points, some of them are not good enough
- The locations of keypoints may be not accurate



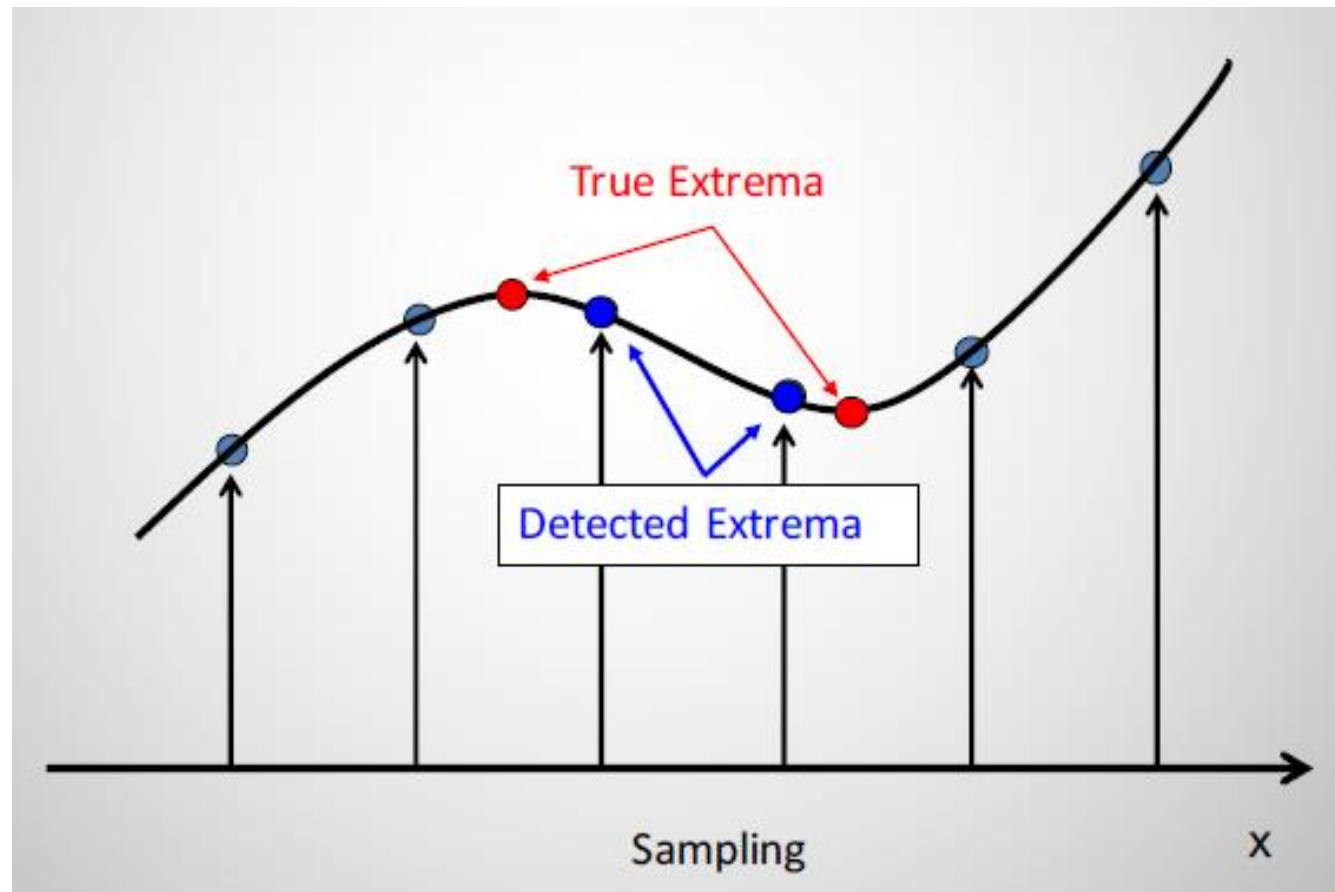
original image



extrema locations

Inaccurate Keypoint Localization

- Inaccurate Keypoint localization due to poor contrast



Inaccurate Keypoint Localization

- The Solution:

- Taylor expansion:

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

- Minimize to find accurate extrema:

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

- If offset from sampling point is larger than 0.5 - Keypoint should be in a different sampling point.

Initial Outlier Rejection



from 832 key points to 729 key points, $th=0.03$.

Further Outlier Rejection

- DOG has strong response along edge
- Assume DOG as a surface
 - Compute principal curvatures (PC)
 - Poorly defined peak will have a large principal curvature across the edge.

Further Outlier Rejection

- Compute Hessian of D (principal curvature)

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad \begin{aligned} \text{Tr}(\mathbf{H}) &= D_{xx} + D_{yy} = \lambda_1 + \lambda_2 \\ \text{Det}(\mathbf{H}) &= D_{xx}D_{yy} - D_{xy}^2 = \lambda_1\lambda_2 \end{aligned}$$

■ Remove outliers by evaluating

$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} = \frac{(r+1)^2}{r} \quad r = \frac{\lambda_1}{\lambda_2}$$

$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} = \frac{(\lambda_1 + \lambda_2)^2}{\lambda_1\lambda_2} = \frac{(r\lambda_2 + \lambda_2)^2}{r\lambda_2^2} = \frac{(r+1)^2}{r}$$

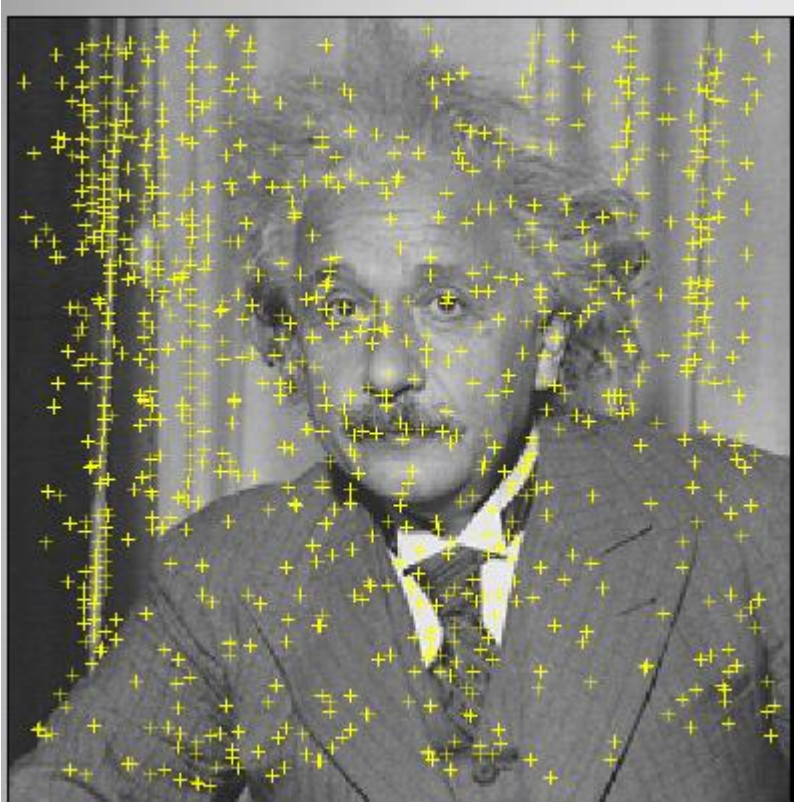
Eliminate key points if

$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} \prec \frac{(r+1)^2}{r} \quad r > 10$$

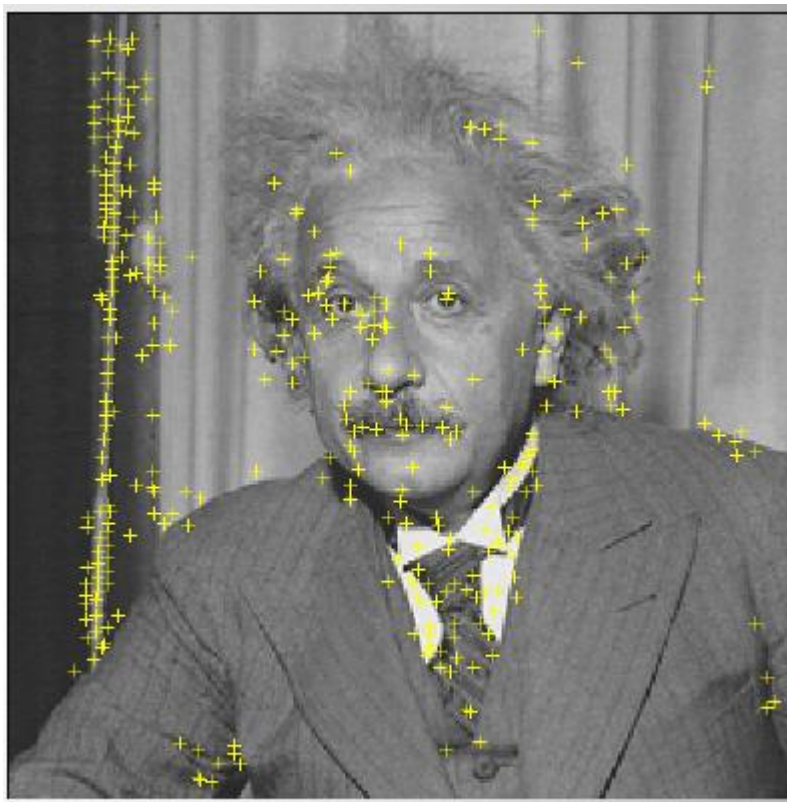
Further Outlier Rejection



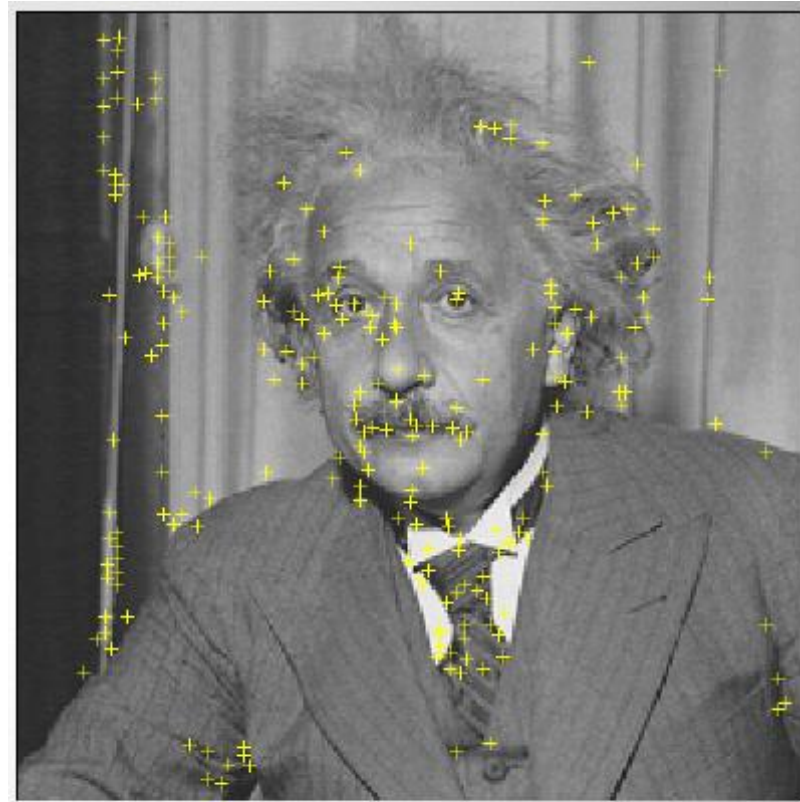
from 729 key points to 536 key points.



Local extremas



Remove low contrast features



Remove low edges

Orientation Assignment

- Use scale of point to choose correct image:

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

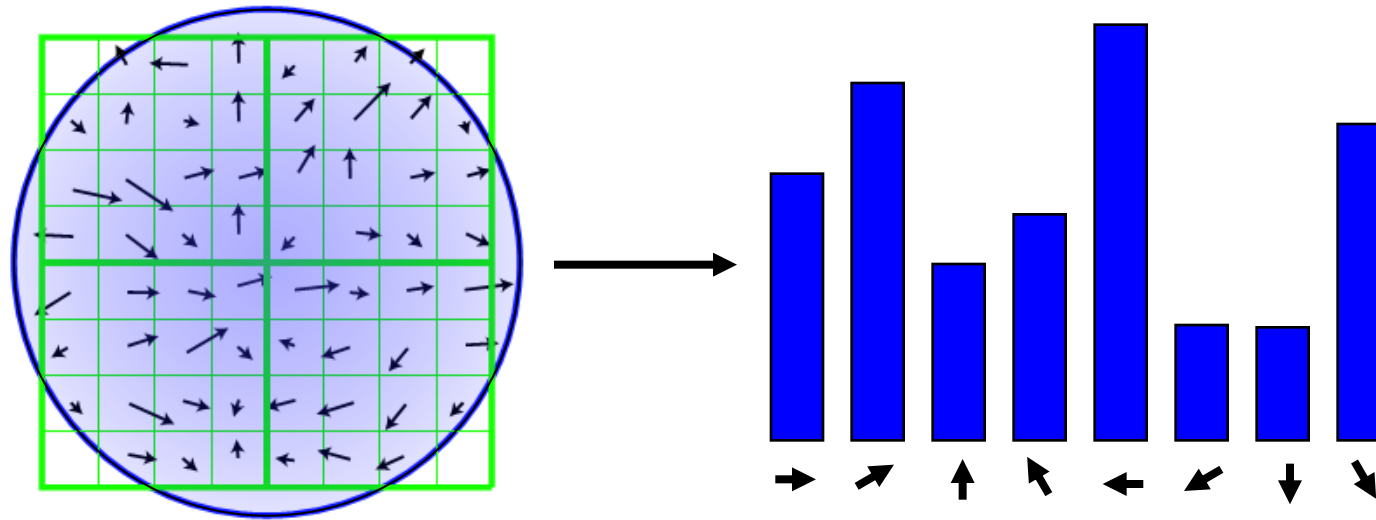
- Compute gradient magnitude and orientation using finite differences:

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

$$\theta(x, y) = \tan^{-1} \left(\frac{(L(x, y+1) - L(x, y-1))}{(L(x+1, y) - L(x-1, y))} \right)$$

Orientation Assignment

- Create gradient histogram (36 bins)
 - Weighted by magnitude and Gaussian window ($\sigma = 1.5$ times that of the scale of a keypoint)

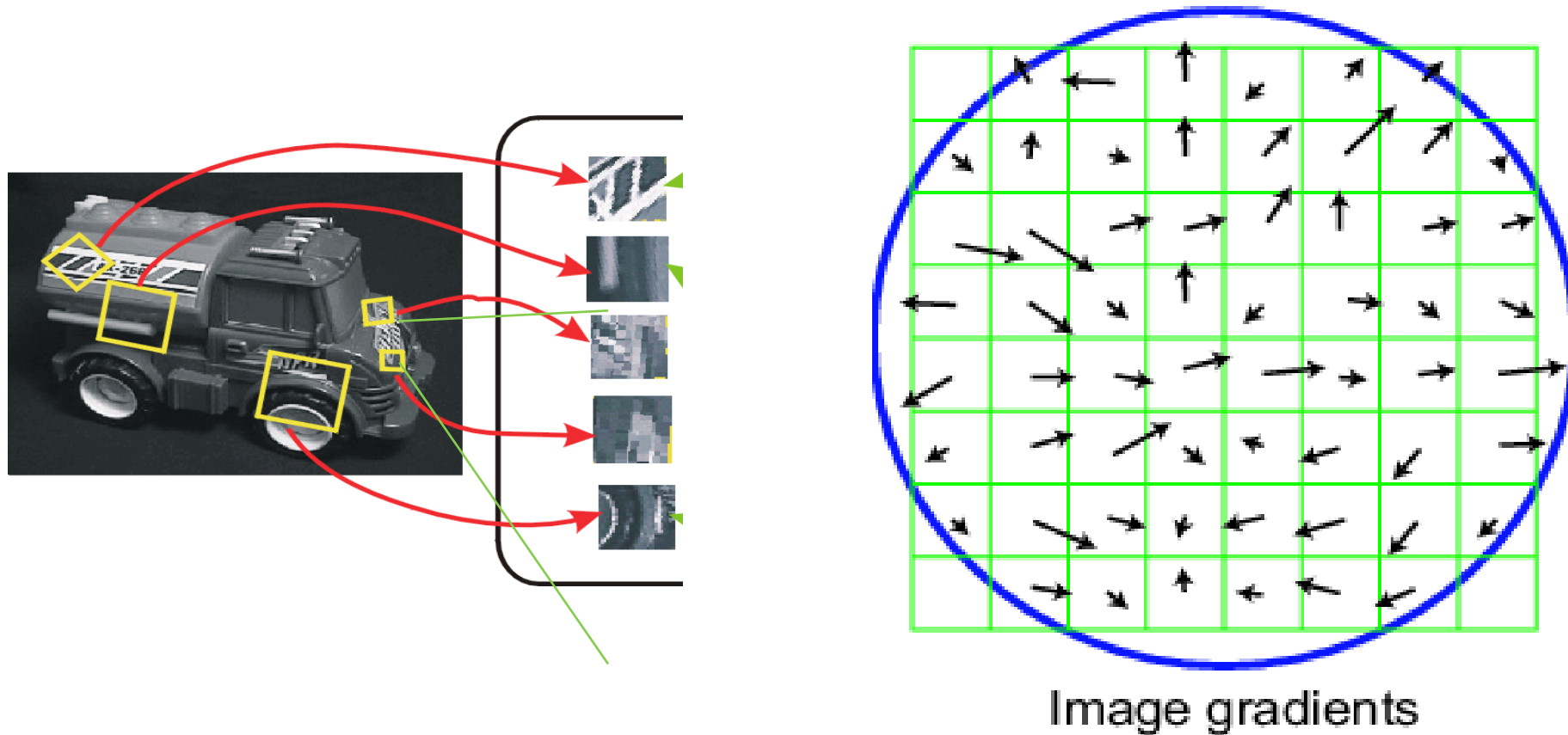


Orientation Assignment

- Any peak within 80% of the highest peak is used to create a keypoint with that orientation

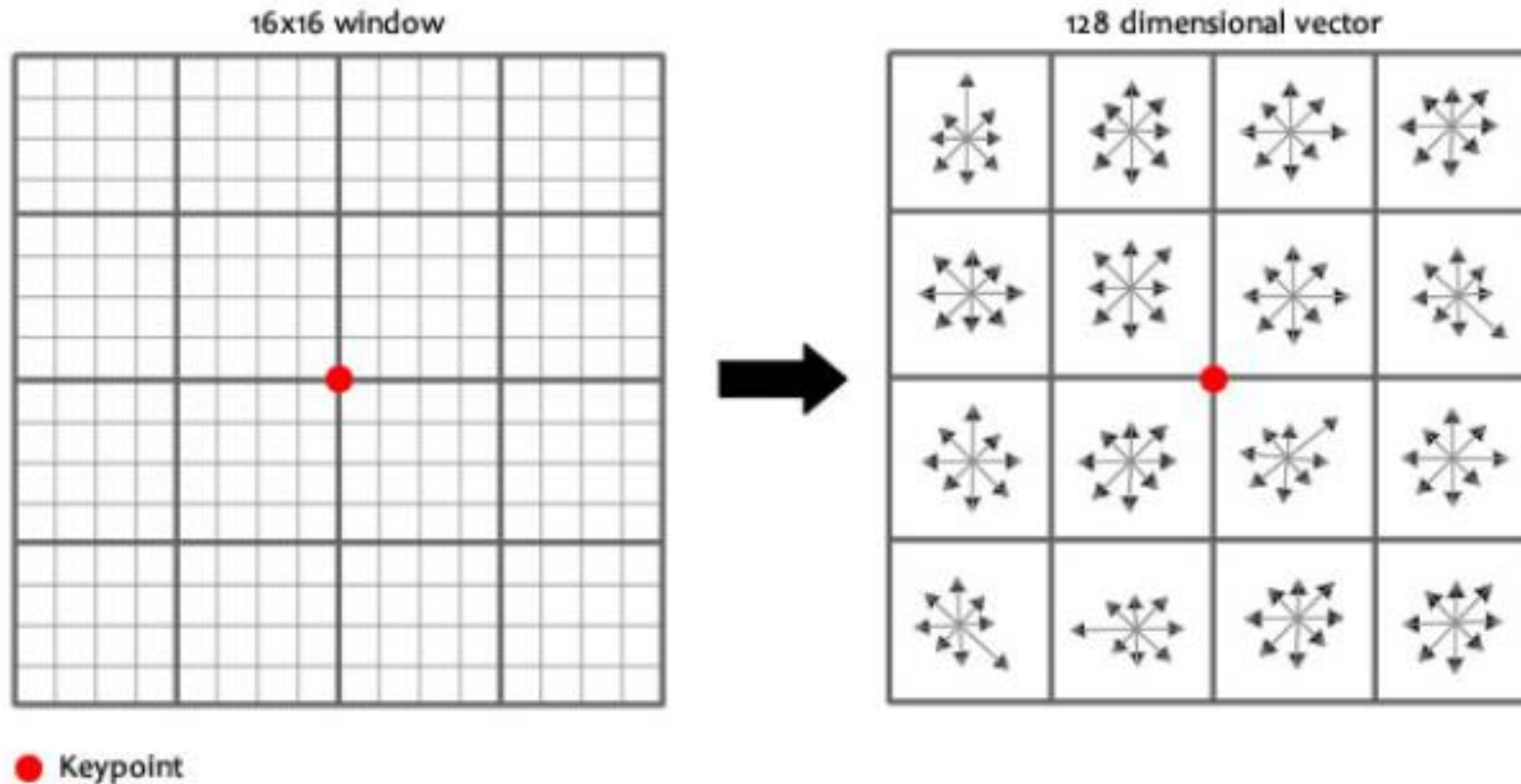
SIFT descriptor formation

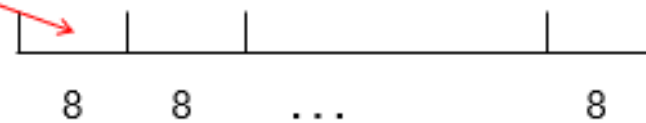
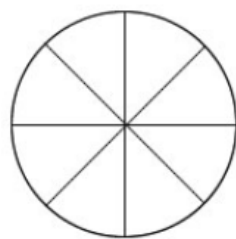
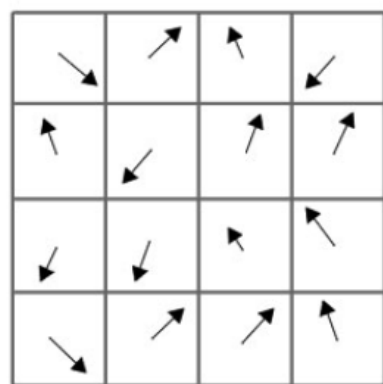
- Compute on local 16 x 16 window around detection.



Actually 16x16, only showing 8x8

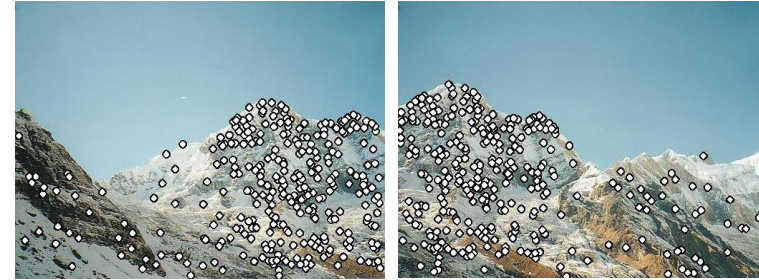
- Compute Divide the **16x16 window** into a 4x4 grid of cells
- an **orientation histogram** for each cell
- 16 cells * 8 orientations = **128 dimensional descriptor**



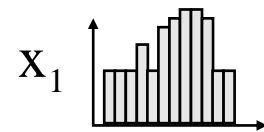


Local features: main components

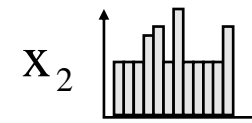
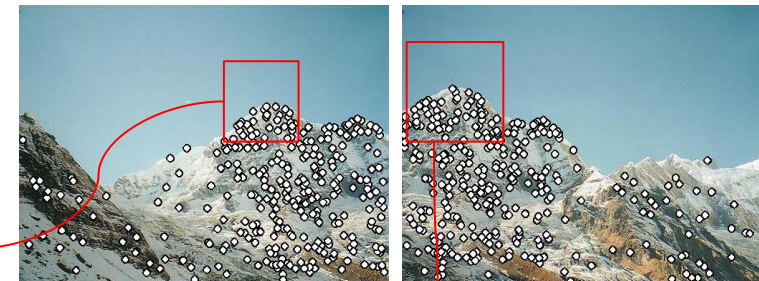
- 1) **Detection:**
Find a set of distinctive key points.



- 2) **Description:**
Extract feature descriptor around each interest point as vector.



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



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

- 3) **Matching:**
Compute distance between feature vectors to find correspondence.

$$d(\mathbf{x}_1, \mathbf{x}_2) < T$$



Object Recognition

- **For training images:**

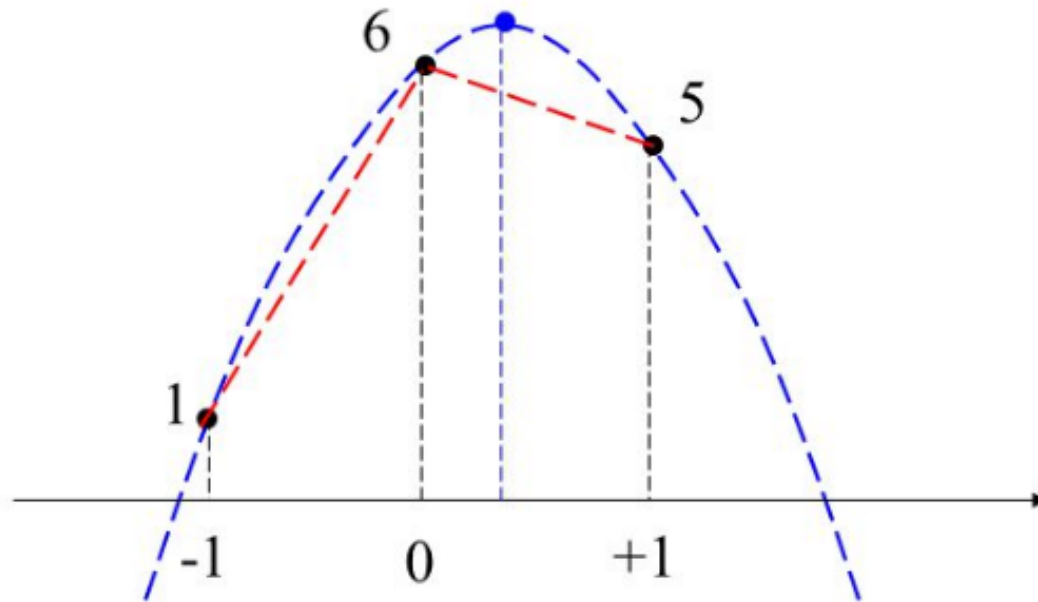
- Extracting keypoints by SIFT.
- Creating descriptors database.

- **For query images:**

- Extracting keypoints by SIFT.
- For each descriptor - finding nearest neighbor in DB.
- Finding cluster of at-least 3 keypoints.
- Performing detailed geometric fit check for each cluster.

2. Accurate keypoint localization

- Reject points with low contrast (flat) and poorly localized along an edge (edge)
- Fit a 3D quadratic function for sub-pixel maxima



2. Accurate keypoint localization

- Taylor series of several variables

$$T(x_1, \dots, x_d) = \sum_{n_1=0}^{\infty} \cdots \sum_{n_d=0}^{\infty} \frac{\partial^{n_1}}{\partial x_1^{n_1}} \cdots \frac{\partial^{n_d}}{\partial x_d^{n_d}} \frac{f(a_1, \dots, a_d)}{n_1! \cdots n_d!} (x_1 - a_1)^{n_1} \cdots (x_d - a_d)^{n_d}$$

- Two variables

$$f(x, y) \approx f(0, 0) + \left(\frac{\partial f}{\partial x} x + \frac{\partial f}{\partial y} y \right) + \frac{1}{2} \left(\frac{\partial^2 f}{\partial x \partial x} x^2 + 2 \frac{\partial^2 f}{\partial x \partial y} xy + \frac{\partial^2 f}{\partial y \partial y} y^2 \right)$$

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) \approx f\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}\right) + \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \frac{1}{2} \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} \frac{\partial^2 f}{\partial x \partial x} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y \partial y} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$f(\mathbf{x}) \approx f(\mathbf{0}) + \frac{\partial f}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 f}{\partial \mathbf{x}^2} \mathbf{x}$$

Accurate keypoint localization

- Taylor expansion in a matrix form, \mathbf{x} is a vector, f maps \mathbf{x} to a scalar

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

Hessian matrix
(often symmetric)

gradient

$$\begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

$$\begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

2D illustration

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

$f_{-1,1}$	$f_{0,1}$	$f_{1,1}$
$f_{-1,0}$	$f_{0,0}$	$f_{1,0}$
$f_{-1,-1}$	$f_{0,-1}$	$f_{1,-1}$

$$\frac{\partial f}{\partial x} = (f_{1,0} - f_{-1,0})/2$$

$$\frac{\partial f}{\partial y} = (f_{0,1} - f_{0,-1})/2$$

$$\frac{\partial^2 f}{\partial x^2} = f_{1,0} - 2f_{0,0} + f_{-1,0}$$

$$\frac{\partial^2 f}{\partial y^2} = f_{0,1} - 2f_{0,0} + f_{0,-1}$$

$$\frac{\partial^2 f}{\partial x \partial y} = (f_{-1,-1} - f_{-1,1} - f_{1,-1} + f_{1,1})/4$$

