Scale Invariant Feature Transform
: SIFT

Han Sol Kang

ISL Lab Seminar

Contents

Introduction

illumination

illumination + Scale

illumination + Scale + Rotation

illumination + Scale Rotation + Affine

Introduction

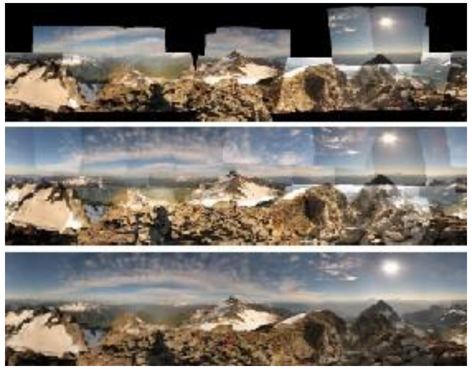
David G Lowe

A senior research scientist at Google (Seattle) in the Machine Intelligence Group.

99: *Object recognition from local scale-invariant features* [1]

04: *Distinctive Image Features from Scale-Invariant Keypoints* [2]

[Overview of Research Projects]



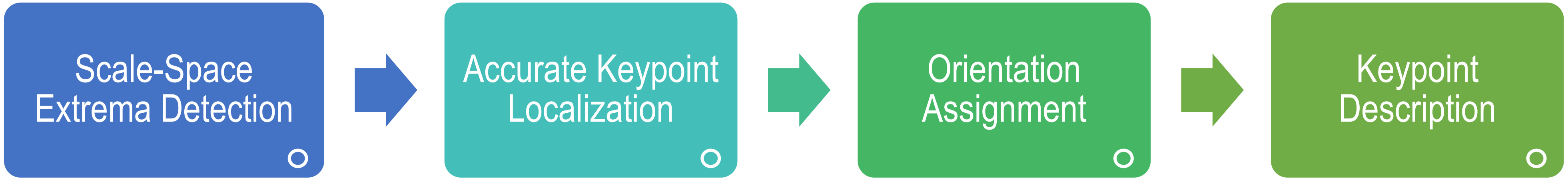
Autostich
: Atuomated paranoma creation

SIFT
: Matching with local invariant features

Augmented reality in natural scenes

[1] *Lowe, David G. "Object recognition from local scale-invariant features." Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Vol. 2. Ieee, 1999.*
[2] *Lowe, David G. "Distinctive image features from scale-invariant keypoints." International journal of computer vision 60.2 (2004): 91-110.*

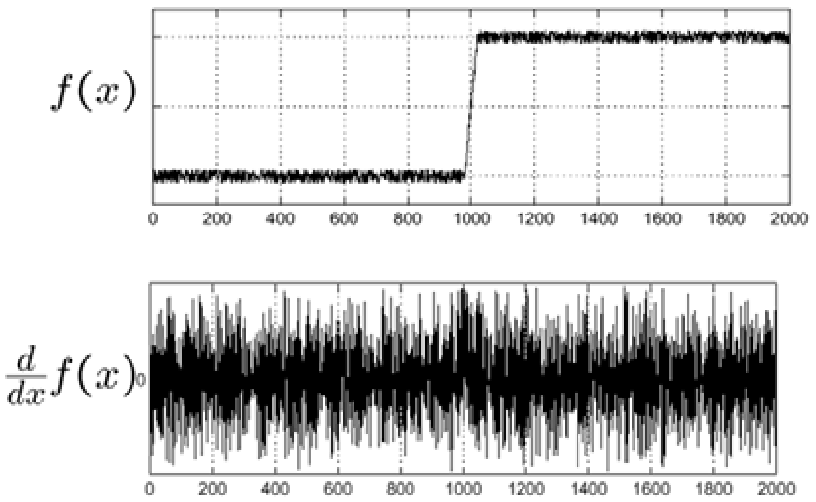
Introduction

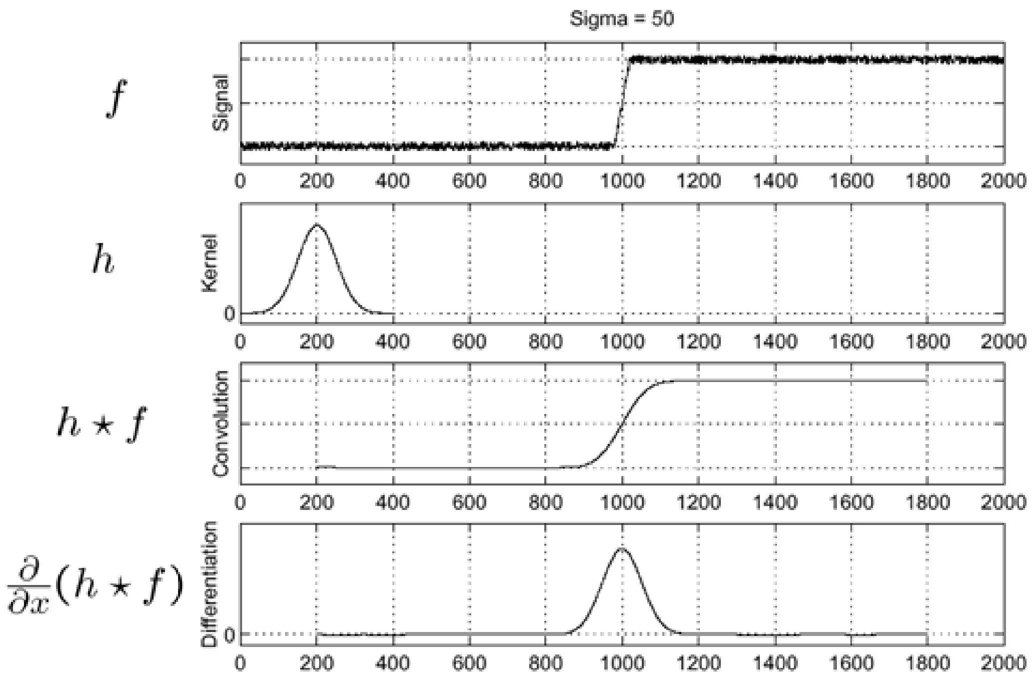


- Search over multiple scales and Image locations.
- Select keypoints based on a measure of stability.
- Compute best orientation(s) for each keypoint region.
- Use local image gradients at selected scale and rotation to describe each keypoint region.

Fundamental theory

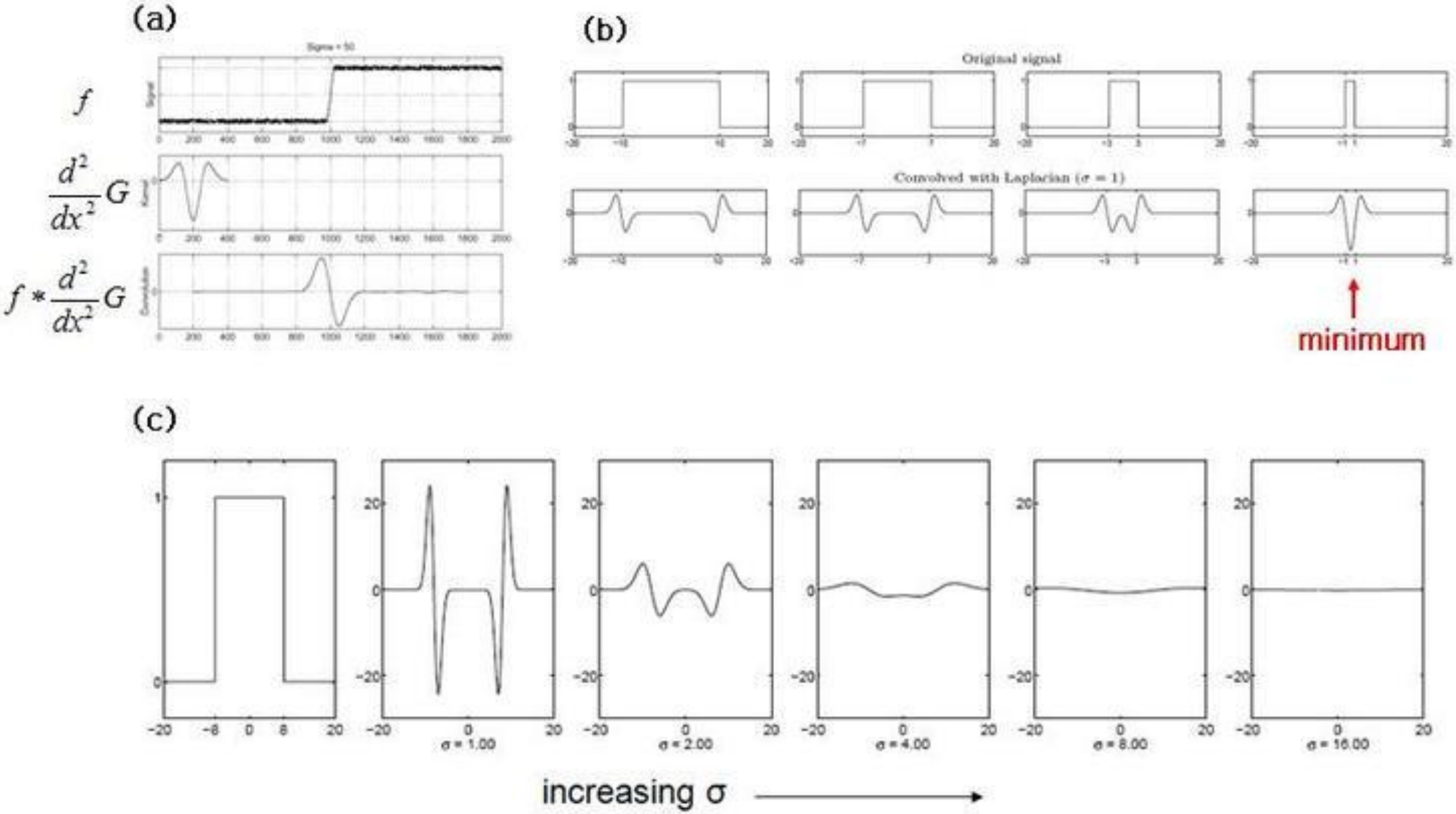
- DOG (edge)





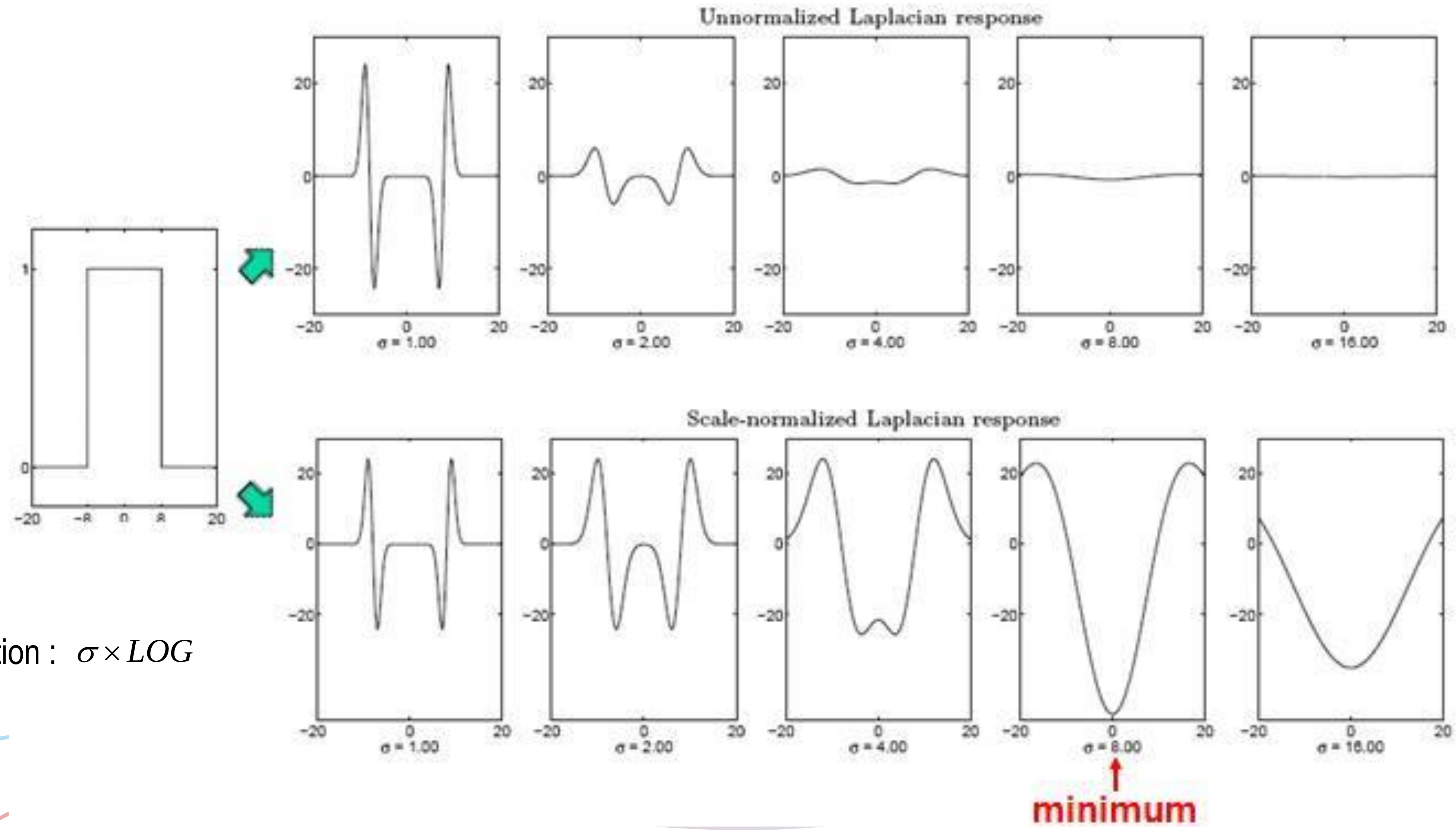
Fundamental theory

❖ LOG (blob)



Fundamental theory

❖ Normalized LOG



Normalization : $\sigma \times LOG$

Fundamental theory

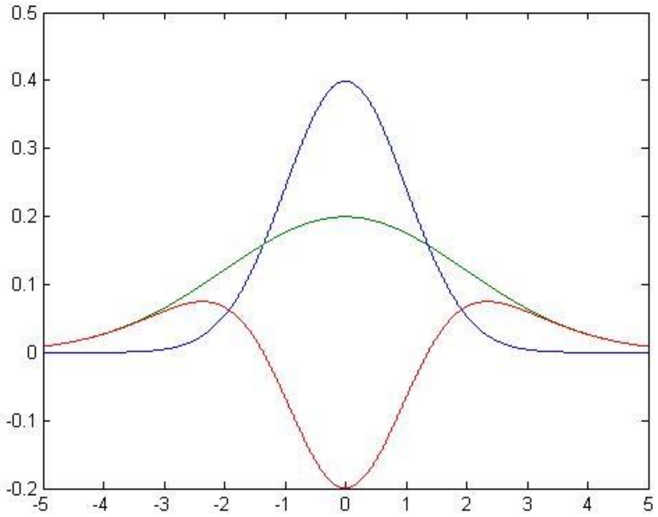
❖ DOG & LOG

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G \longrightarrow \text{Heat Diffusion Equation}$$

$$\frac{\partial G}{\partial \sigma} \approx \frac{G(x,y,k\sigma) - G(x,y,\sigma)}{k\sigma - \sigma}$$

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

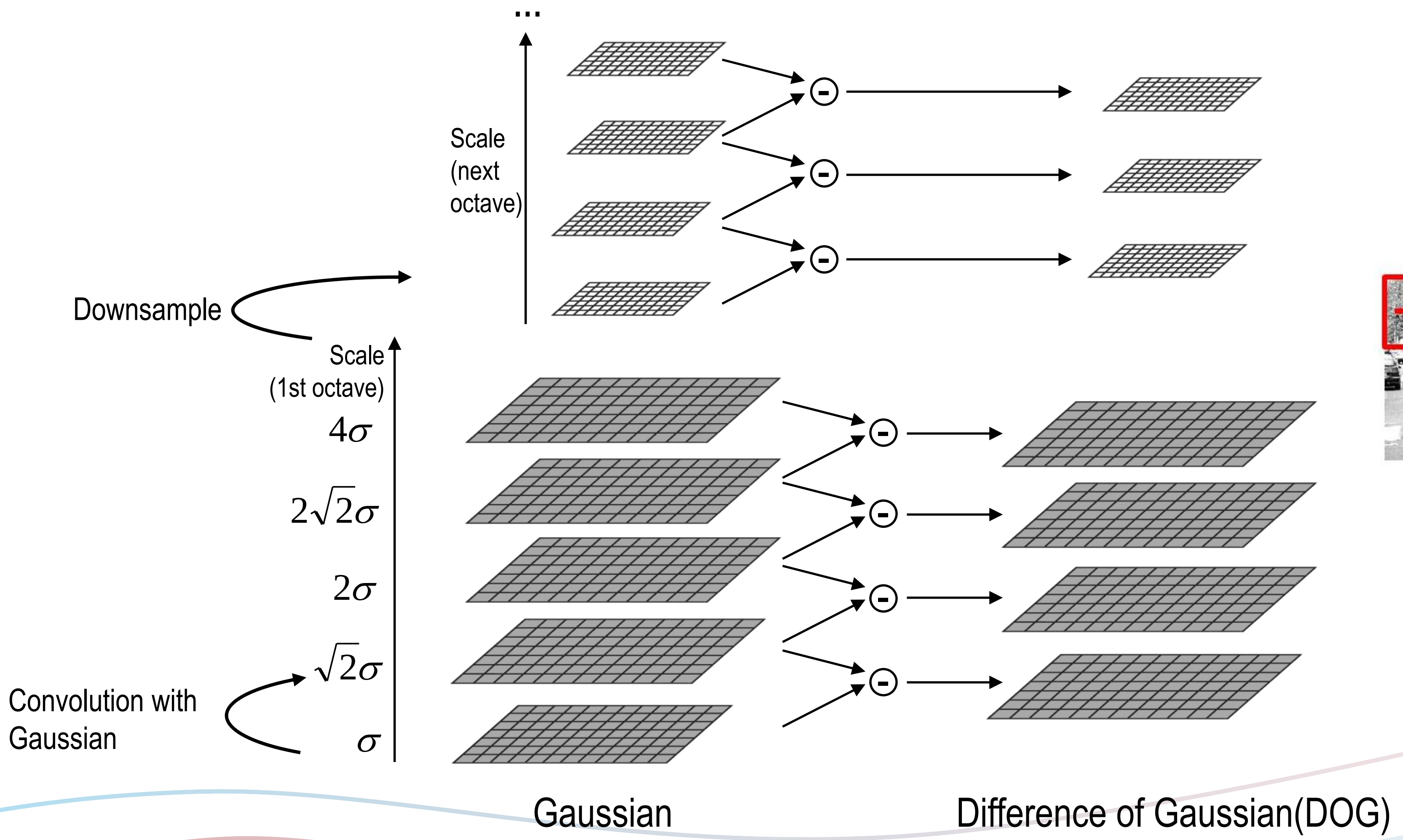
$$DOG \approx (k-1)\sigma^2 \nabla^2 G = (k-1)\sigma \times NLOG(Normalized\ \ Laplacian\ of\ \ Gaussian)$$

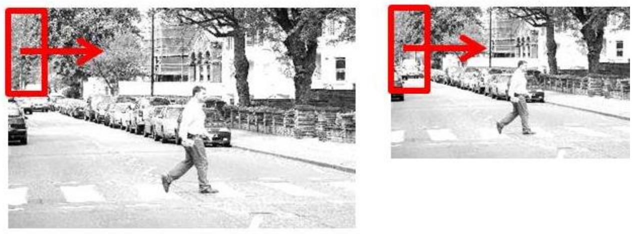


: $\sigma = 1$
: $\sigma = 2$
: DOG

Fundamental theory

❖ Gaussian Pyramid

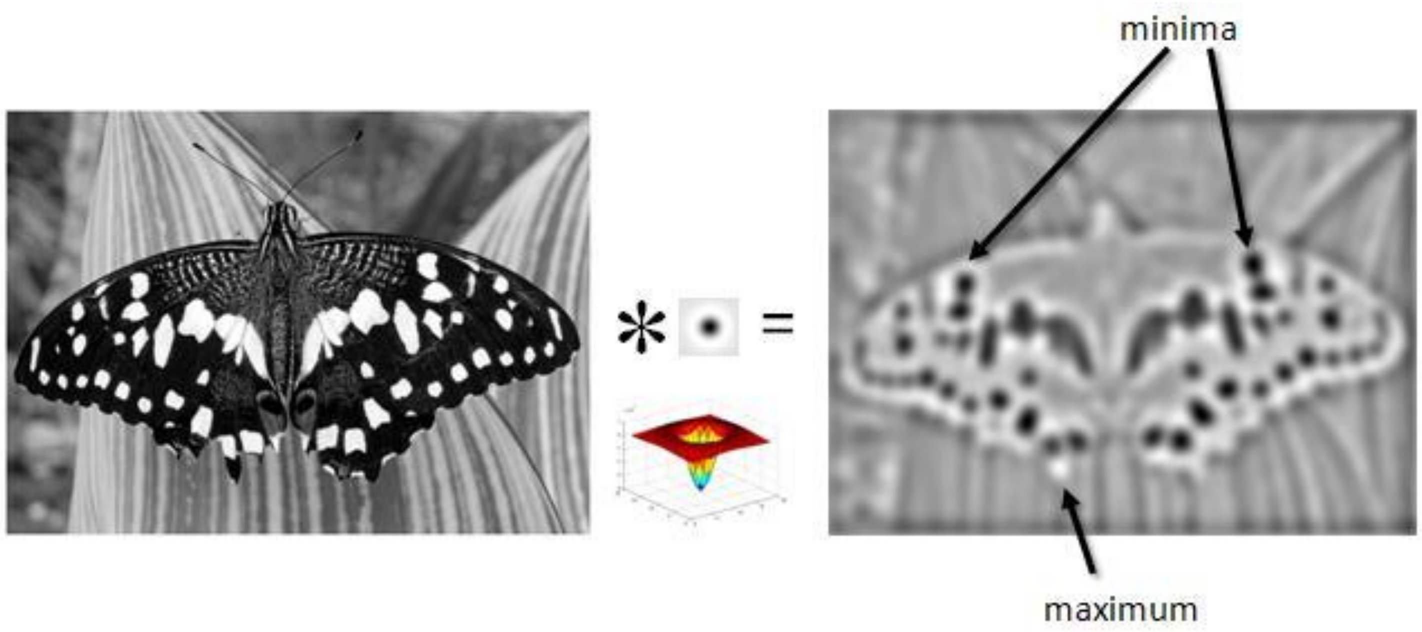




SIFT

❖ Detection of Scale-Space Extrema

Extrema : maxima & minima



SIFT

❖ Detection of Scale-Space Extrema

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

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

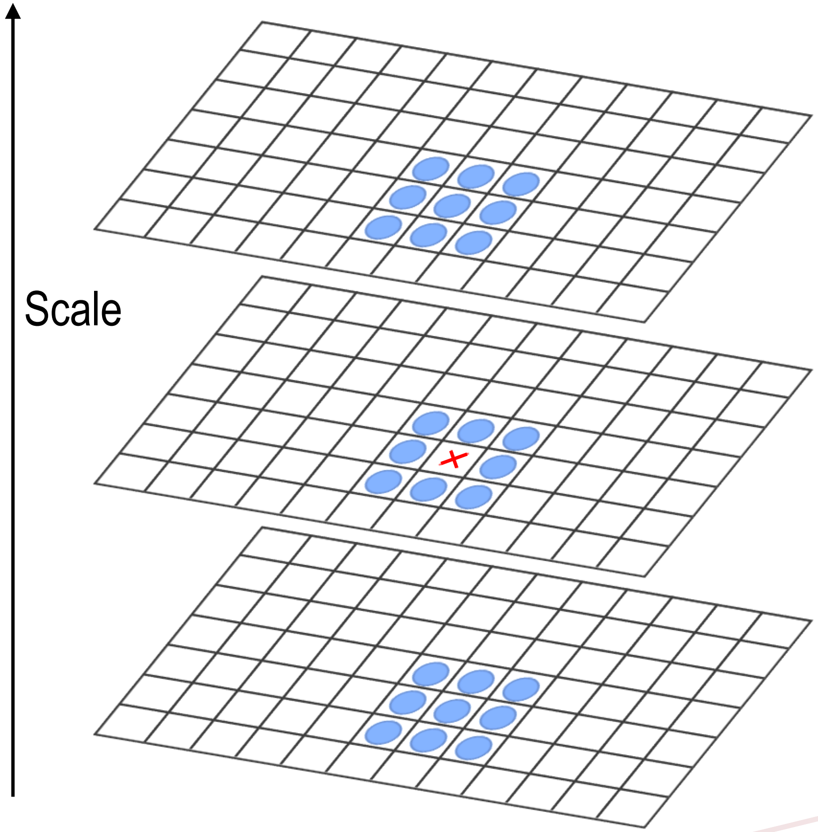
$$\begin{aligned} D(x,y,\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma) \end{aligned}$$

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

interval : s

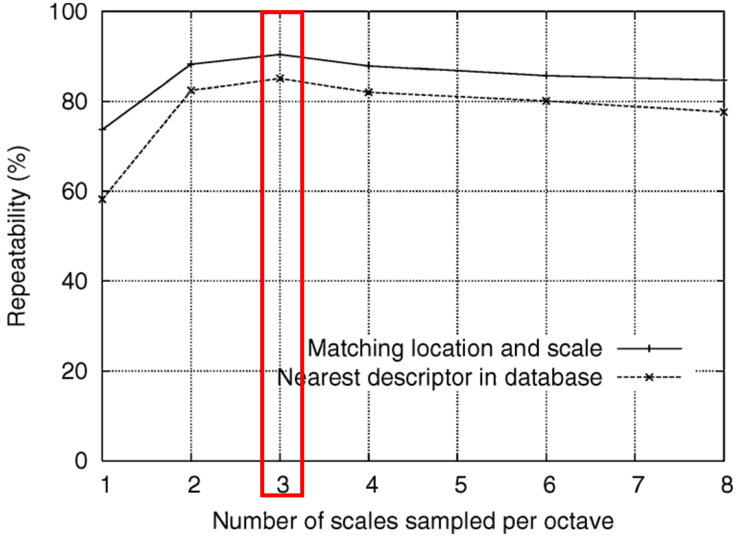
the number of Gaussian Image : $s+3$

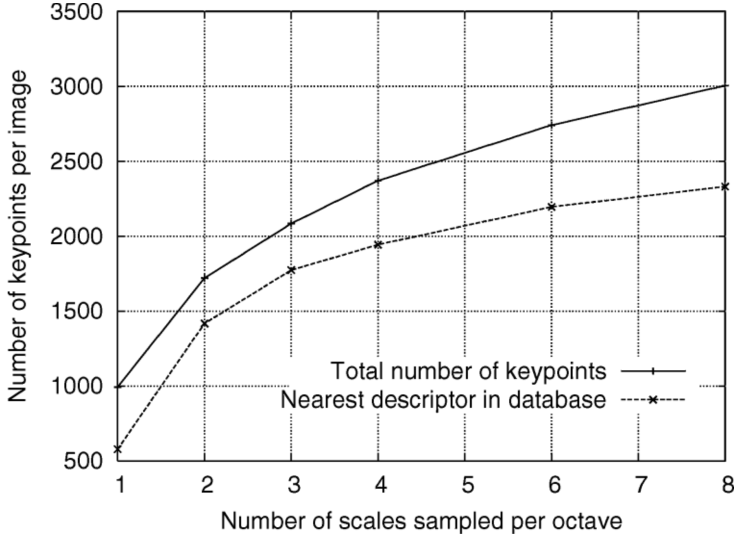
scaling ratio : $k = 2^{1/s}$

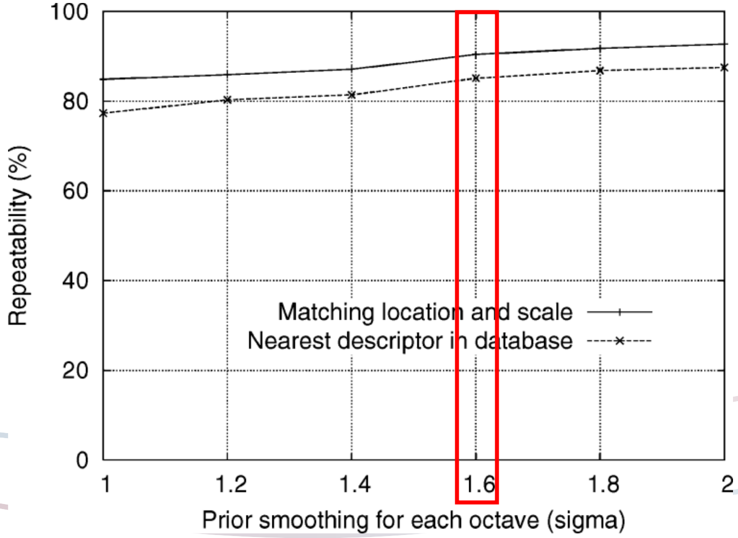


SIFT

Detection of Scale-Space Extrema







SIFT

❖ Accurate Keypoint Localization (low contrast)

$$D(\mathrm{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathrm{T}}\frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \longrightarrow \text{Taylor Expansion}$$

$$D'(\mathrm{x}) = 0 + \frac{\partial D^T}{\partial \mathbf{x}} + \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \qquad (\mathbf{x} = (x,y,\sigma)^{\mathrm{T}})$$

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

$$\mathbf{x} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2}\frac{\partial D^T}{\partial \mathbf{x}} \longrightarrow \hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2}\frac{\partial D}{\partial \mathbf{x}} \qquad (if\ \ \hat{\mathbf{x}} > 0.5)$$

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

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

$$= D + \frac{1}{2}\frac{\partial D^T}{\partial \hat{\mathbf{x}}}\hat{\mathbf{x}} \qquad \therefore D(\hat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D^T}{\partial \mathbf{x}}\hat{\mathbf{x}} \qquad (if\ \left|D(\hat{\mathbf{x}})\right| < 0.03)$$

SIFT

❖ Accurate Keypoint Localization (edge)

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

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

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

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r} \qquad (r = 10)$$

SIFT

❖ Accurate Keypoint Localization

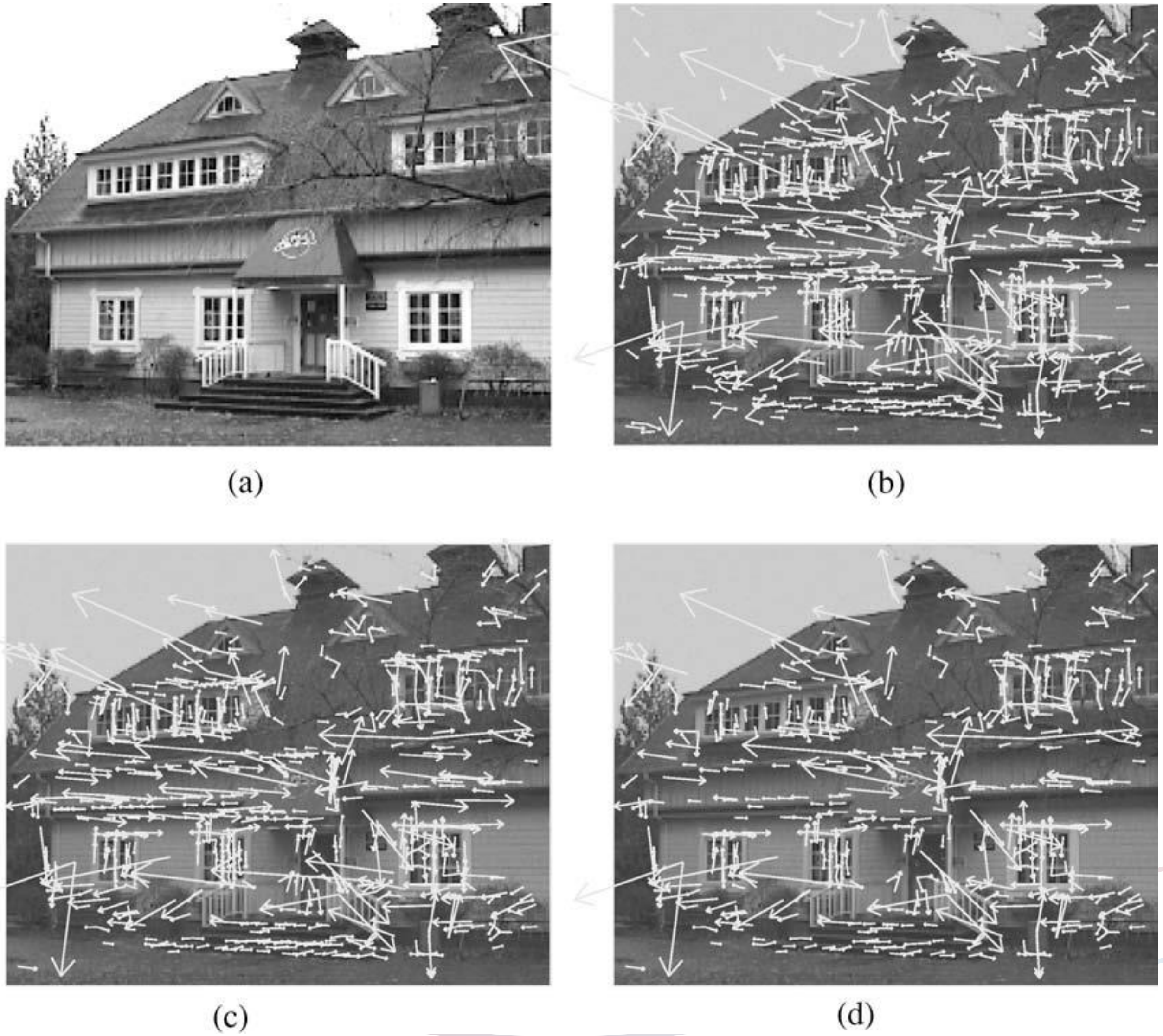
(a) 233x189 pixel original image

(b) 832 keypoints location

(c) 729 keypoints location
(threshold on minimum contrast)

(d) 536 keypoints location

(threshold on ratio of principal curvatures)

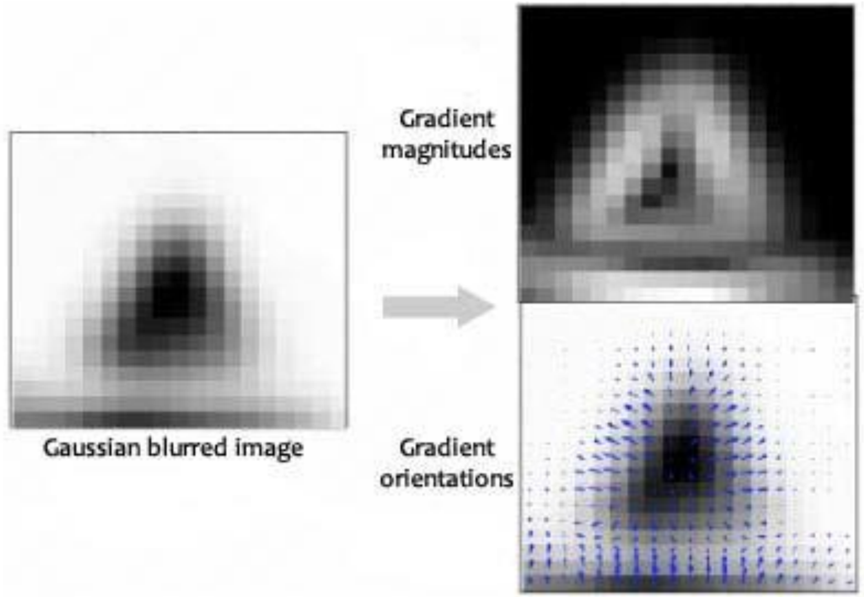


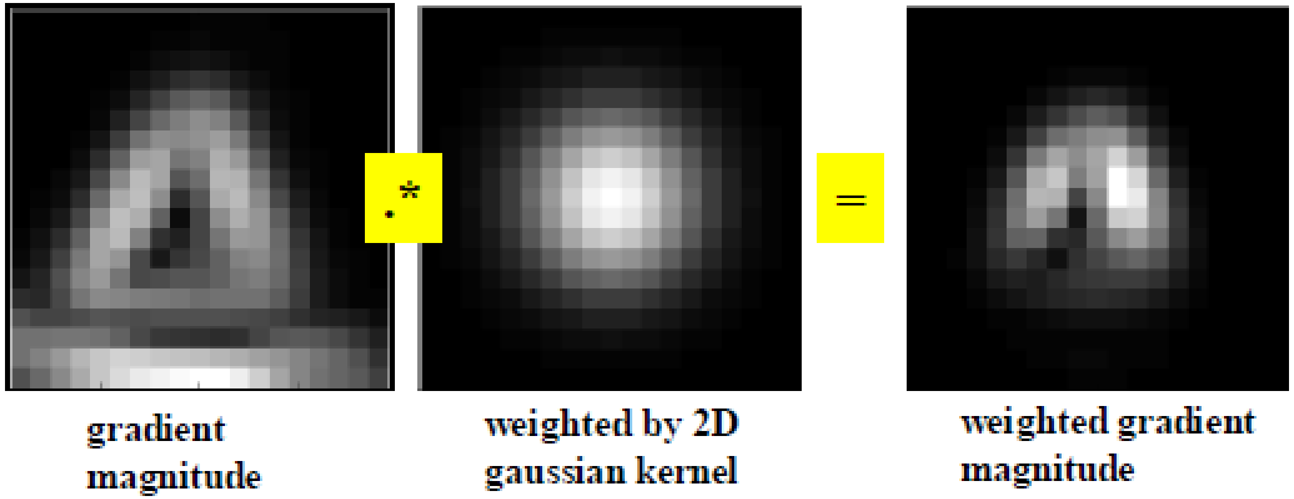
SIFT

❖ Orientation Assignment

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

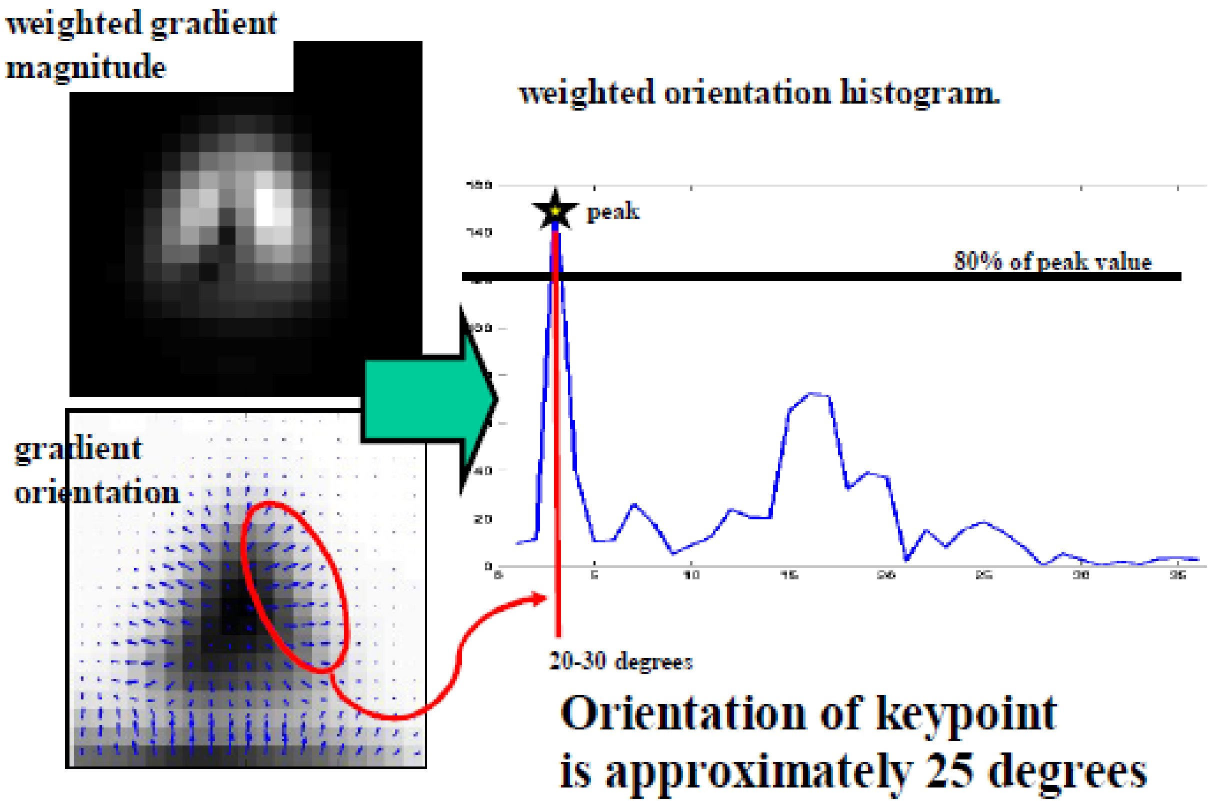
$$\theta(x,y)=\tan^{-1}((L(x,y+1)-L(x,y-1))/(L(x+1,y)-L(x-1,y)))$$

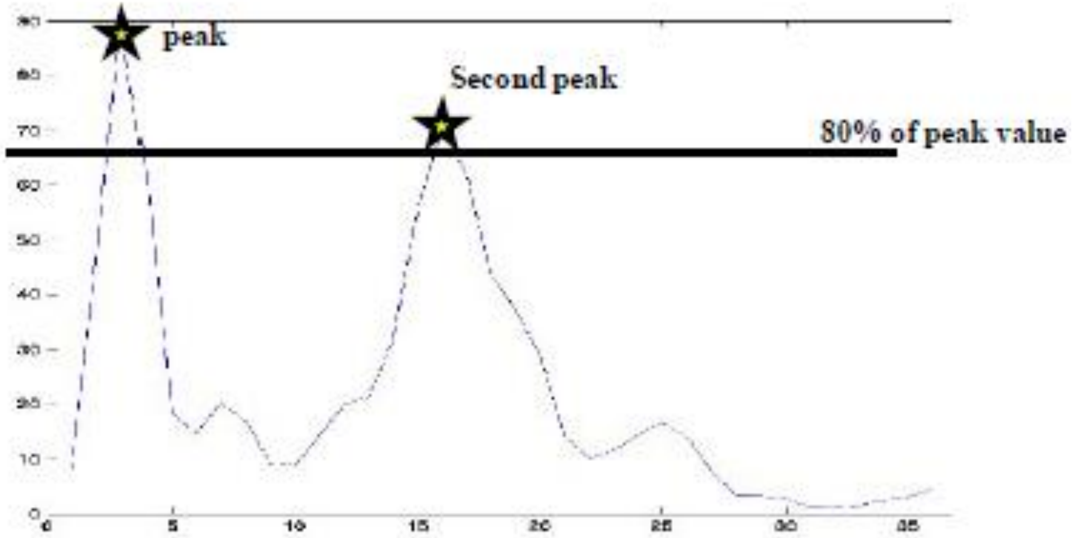




SIFT

❖ Orientation Assignment

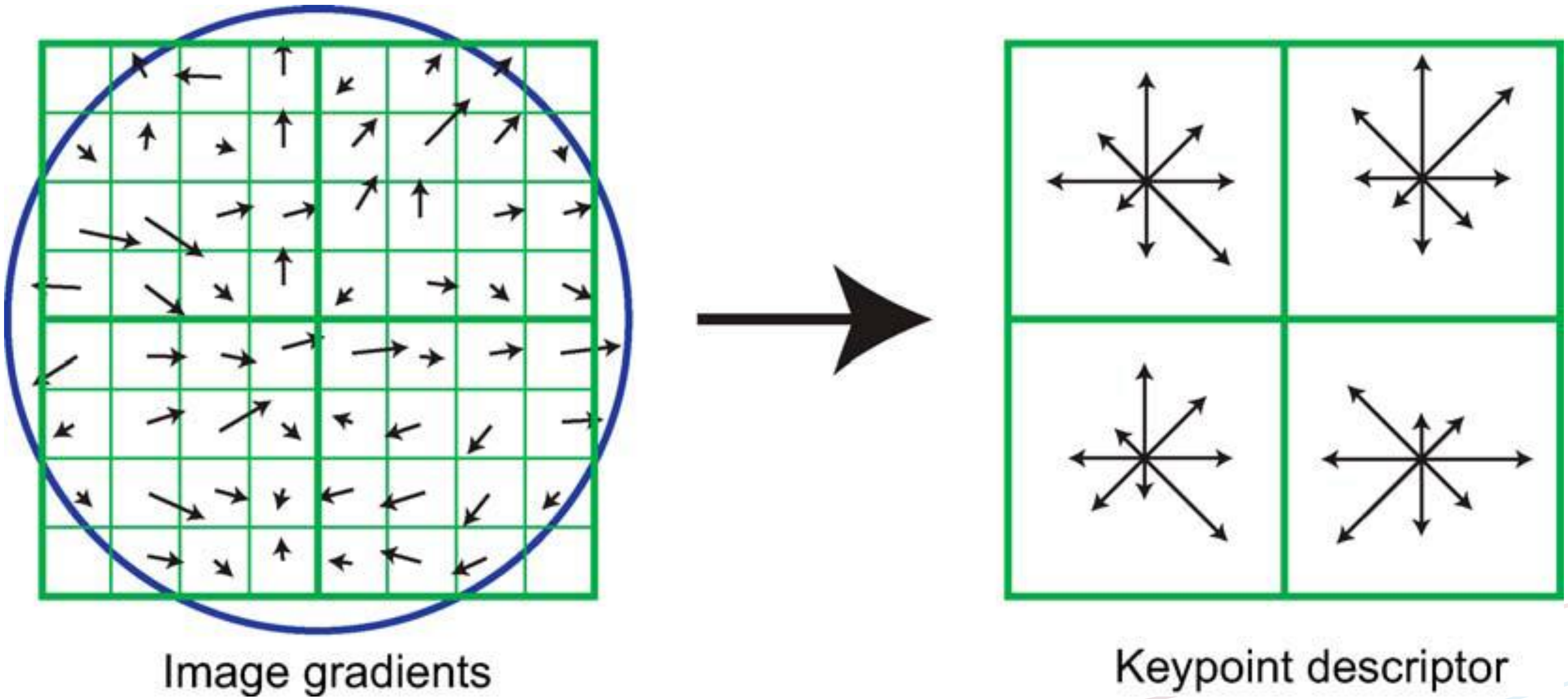




Histogram : Using 36bins

SIFT

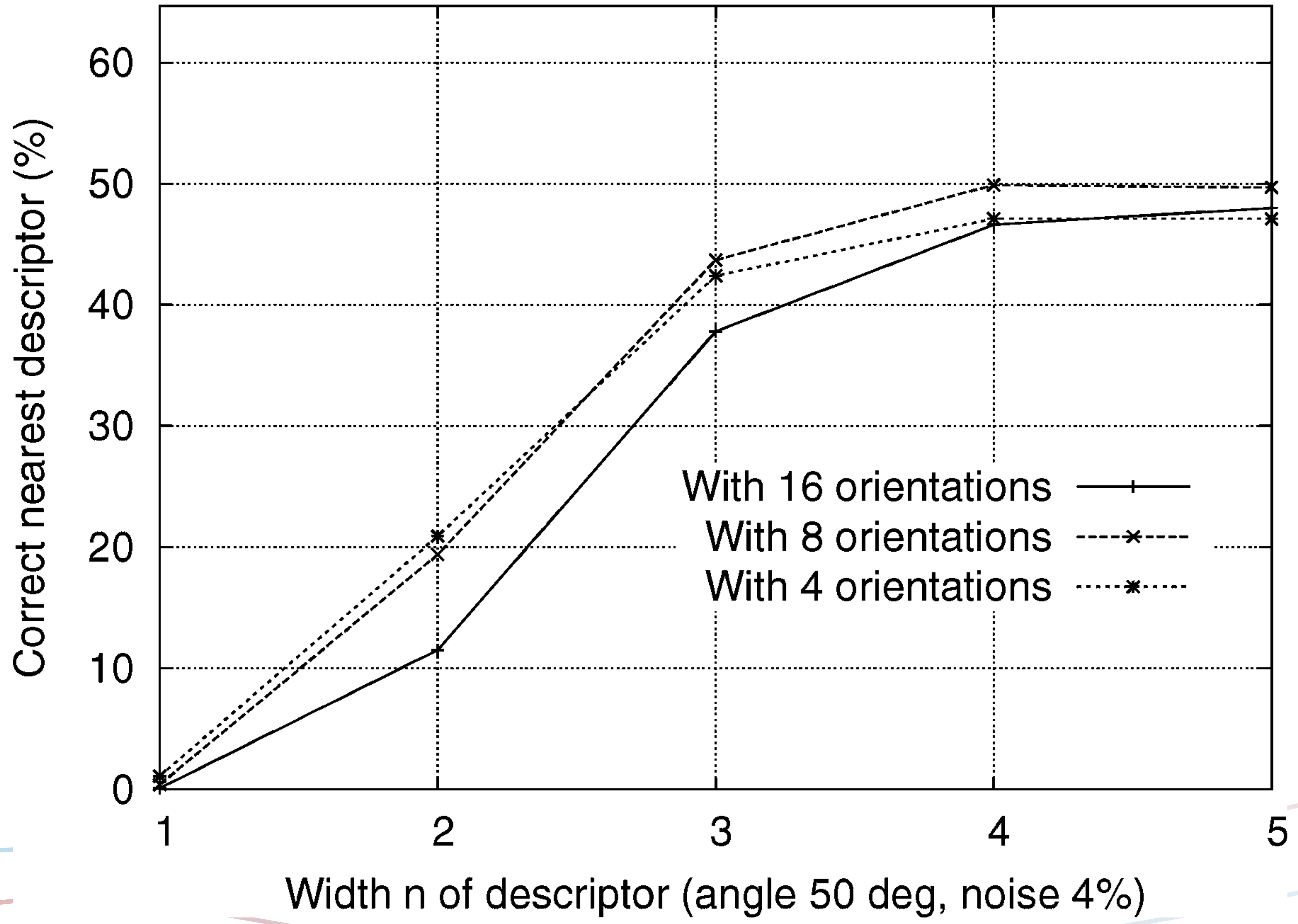
❖ The Local Images Descriptor



illumination : normalization vector
(Feature vector < 0.2)

SIFT

❖ **The Local Images Descriptor**

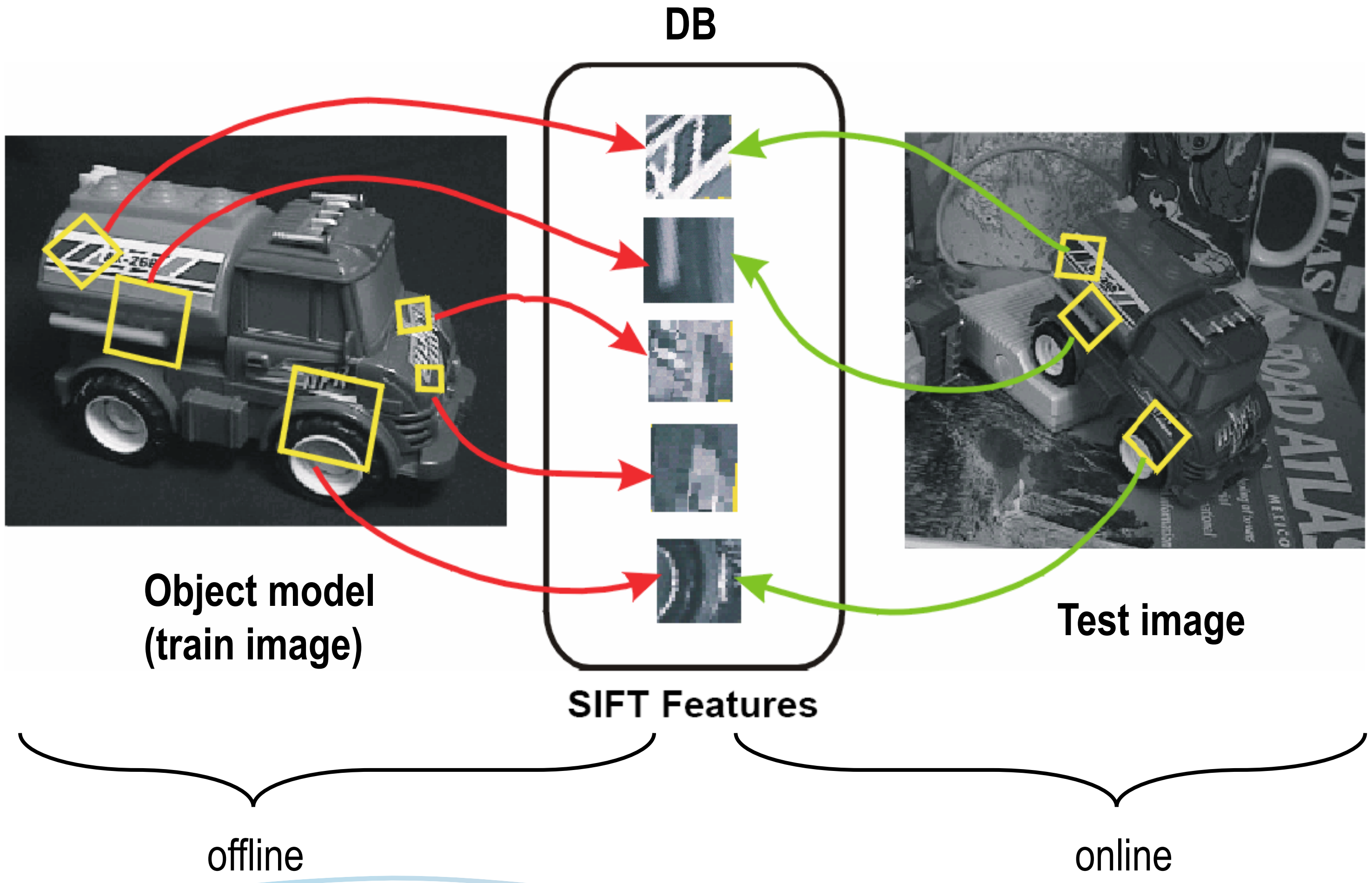


r : the number of orientations
n : the width

The size of the resulting descriptor vector is rn^2

SIFT

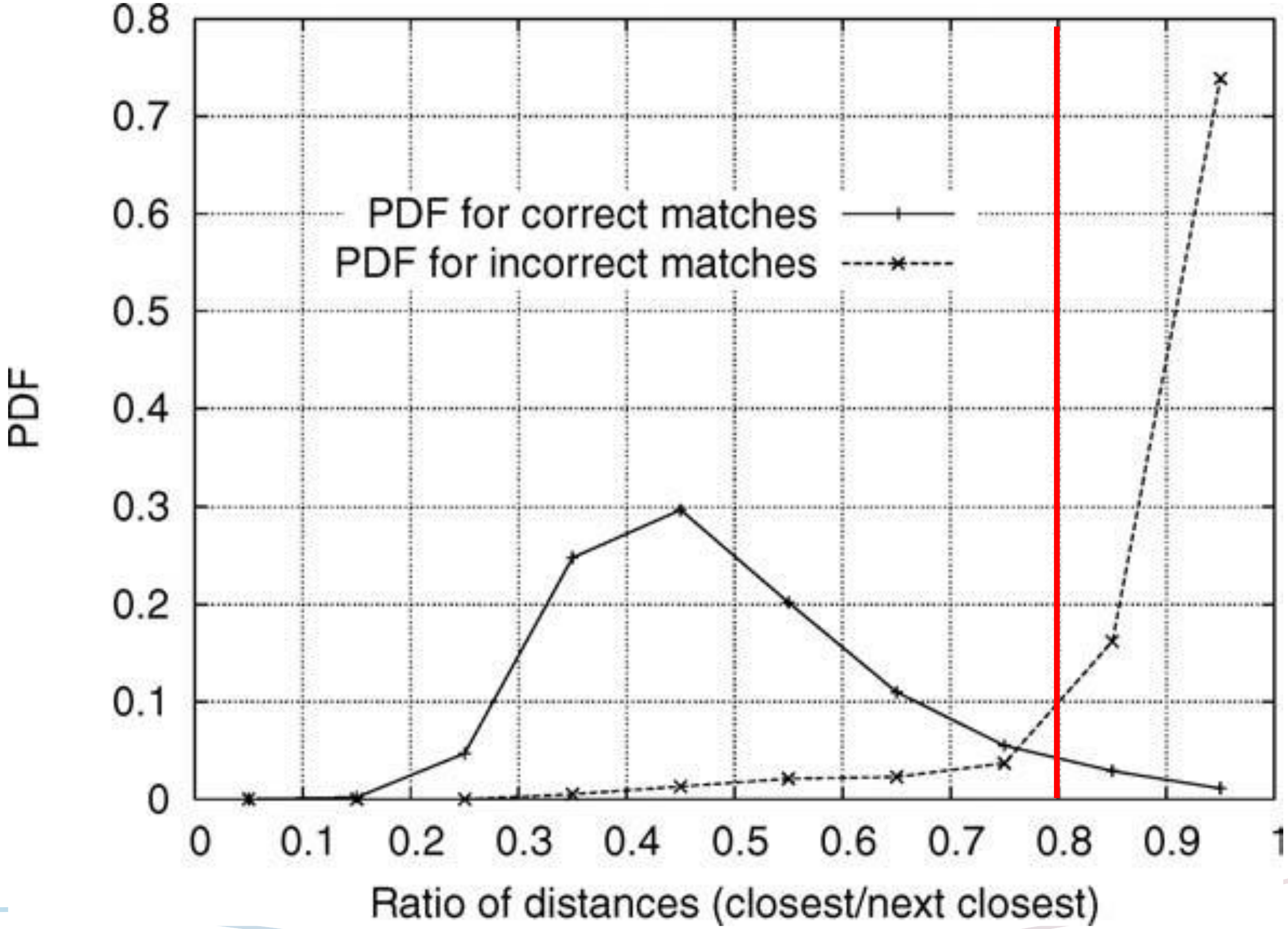
❖ Keypoint Matching



1) Nearest-neighbor search
   : Euclidean distance, K-D tree, BBF(Best-Bin-First)
2) Cluster identification by Hough transform voting
3) Model verification by linear least squares
4) Outlier detection

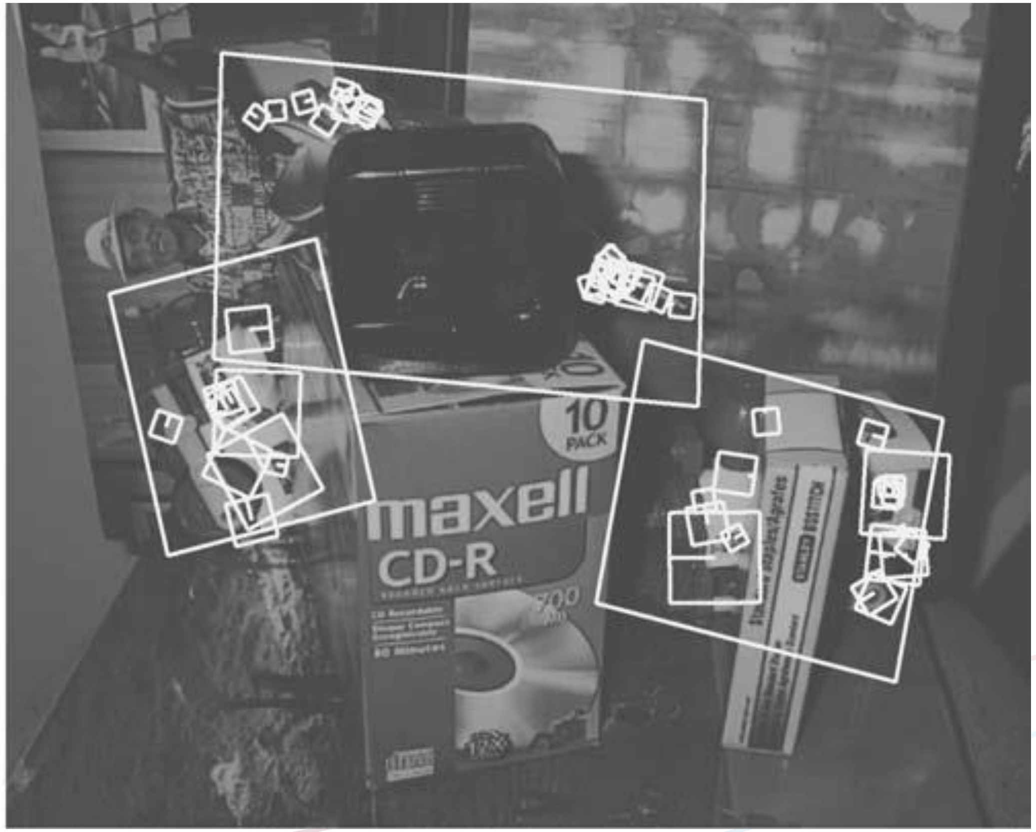
SIFT

❖ Keypoint Matching



Example

- Recognition



Example

- Recognition

Q & A

Thank You!!

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m1 & m2 \\ m3 & m4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & ... & & & \\ & & ... & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix}$$

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{x} = [\mathbf{A}^{\mathrm{T}}\mathbf{A}]^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}$$