Computer Vision Processing

Scale Invariant Feature Transform

YOON. H C

Content

- Introduction to SIFT
- Detection of Scale-Space Extrema
- Accurate Keypoint Localization
- Orientation Assignment
- The Local Image Descriptor
- Application to object Recognition

Introduction to SIFT

David Lowe invent SIFT at 1999

- Point Matching
- Scale Invariant
- Luminance Invariant
- Orientation Invariant
- Affine Transformation Invariant

Detection of Scale-Space Extrema

- Difference of Gaussian

$$D(x, y, \sigma) = \big(G(x, y, k\sigma) - G(x, y, \sigma)\big) * I(x, y)$$
$$= L(x, y, k\sigma) - L(x, y, \sigma)$$

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

- Laplacian of Gaussian : $\nabla^2 L(x, y, \sigma) = L_{xx} + L_{yy}$
- Heat diffusion equation : $\frac{\partial L}{\partial \sigma} = \sigma \nabla^2 L$

Detection of Scale-Space Extrema

$$\sigma\nabla^2 L = \frac{\partial L}{\partial \sigma} = \frac{\Delta L}{\Delta \sigma} = \lim_{k\to 1} \frac{L(x,y,k\sigma) - L(x,y,\sigma)}{k\sigma - \sigma}$$

$$\approx \frac{L(x,y,k\sigma) - L(x,y,\sigma)}{(k-1)\sigma}$$

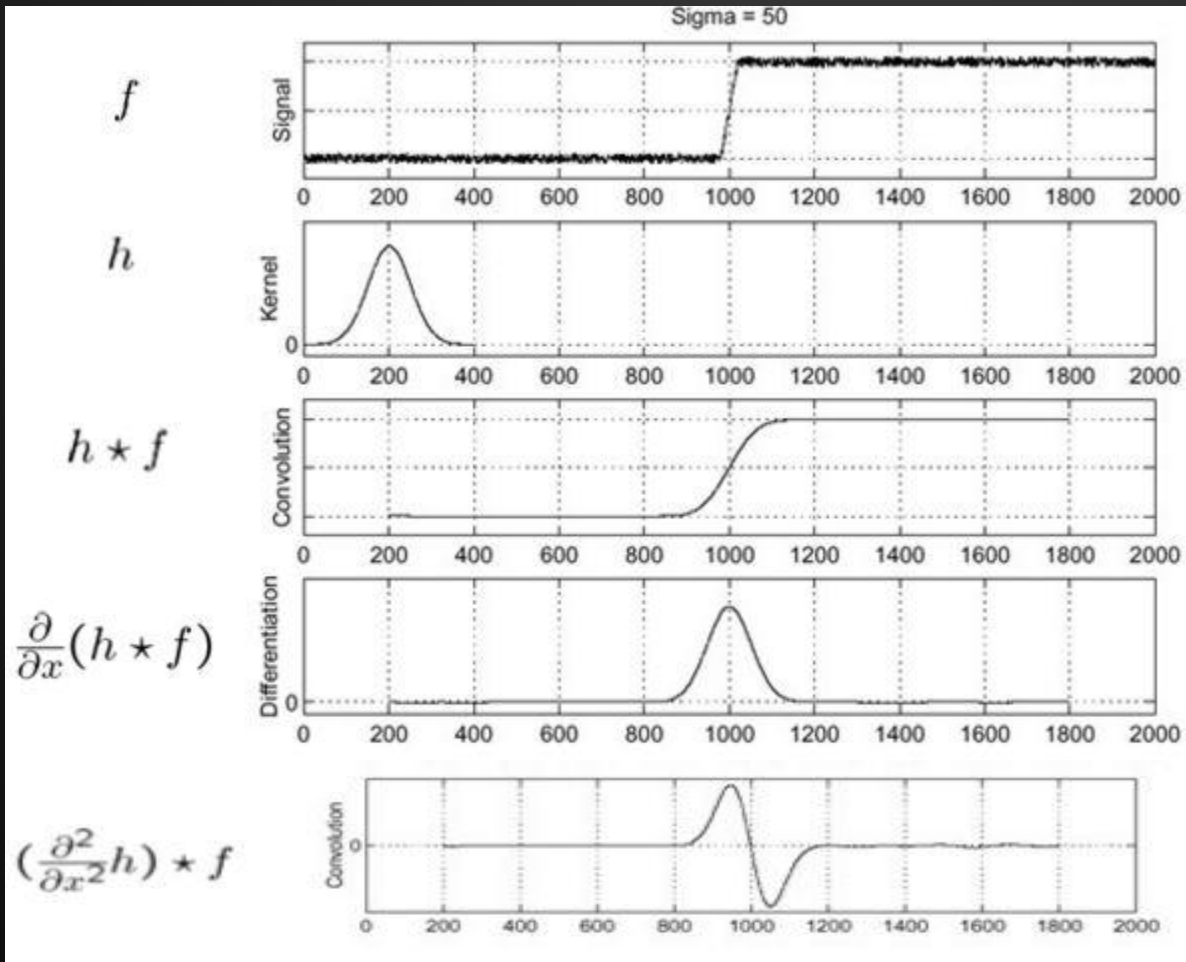
$$(k-1)\sigma \cdot \sigma\nabla^2 L = L(x,y,k\sigma) - L(x,y,\sigma)$$

$$(k-1)\sigma^2\nabla^2 L = L(x,y,k\sigma) - L(x,y,\sigma)$$

$$(k-1)\sigma^2 LoG = L(x,y,k\sigma) - L(x,y,\sigma)$$

$$(k-1)LoG_{nomalized} = DoG$$

Detection of Scale-Space Extrema



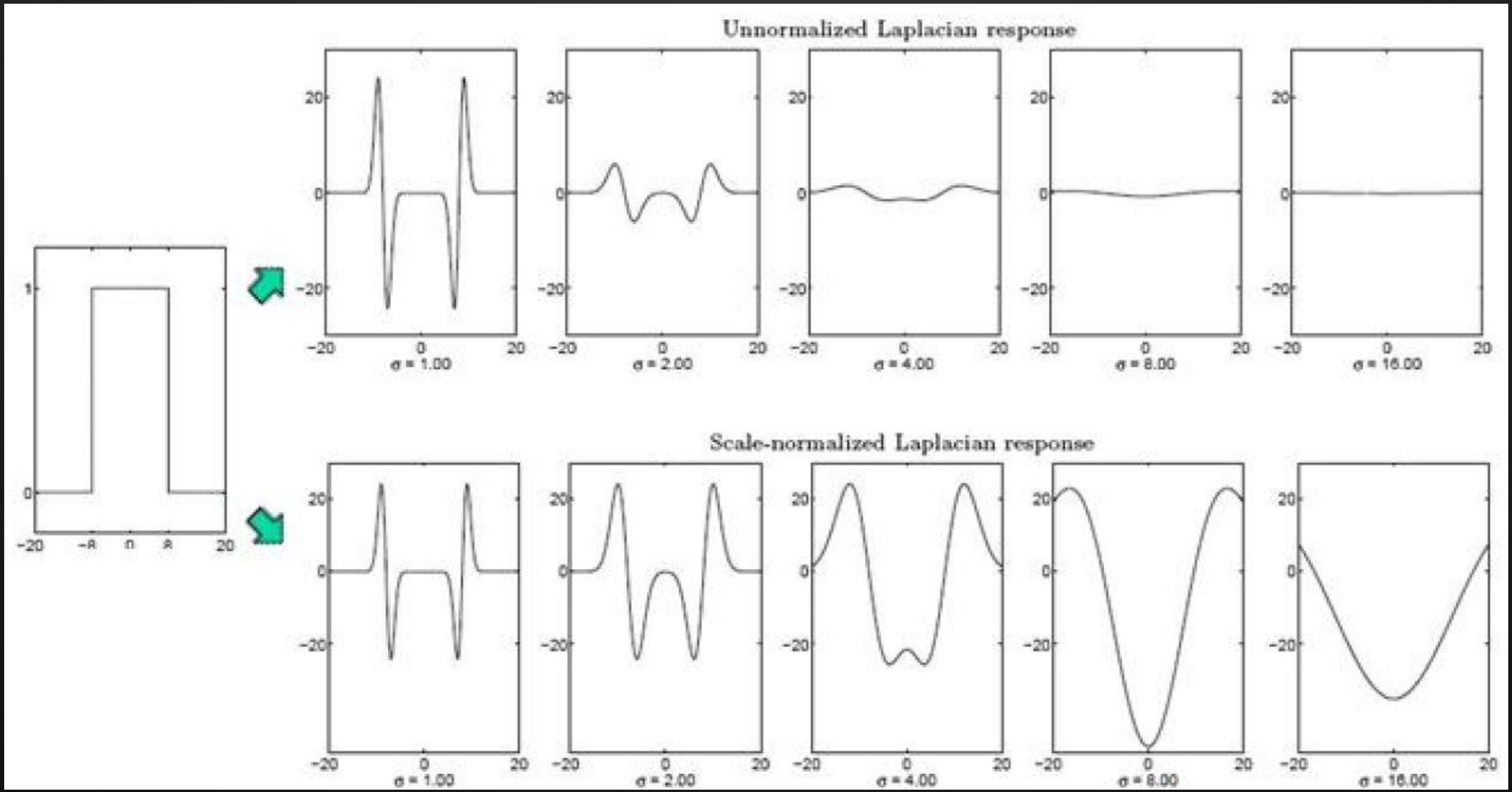
$\rightarrow I$

$\rightarrow g$

$\rightarrow g * I$

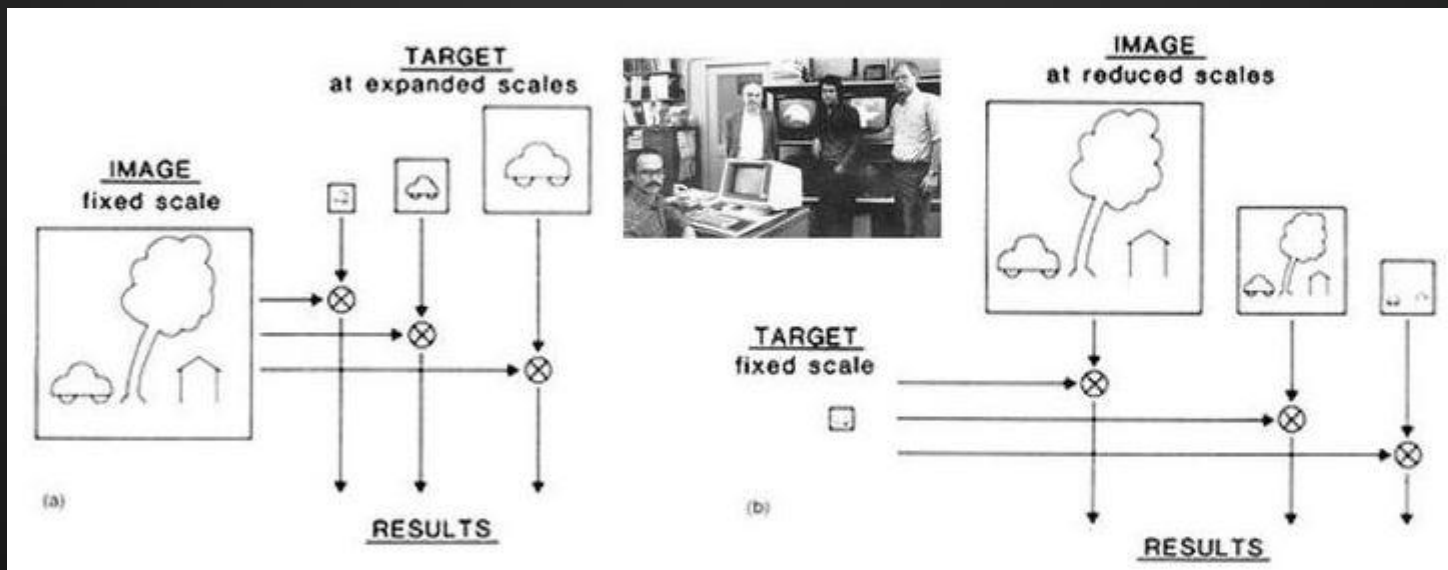
$\rightarrow G * I = L$

Detection of Scale-Space Extrema



Detection of Scale-Space Extrema

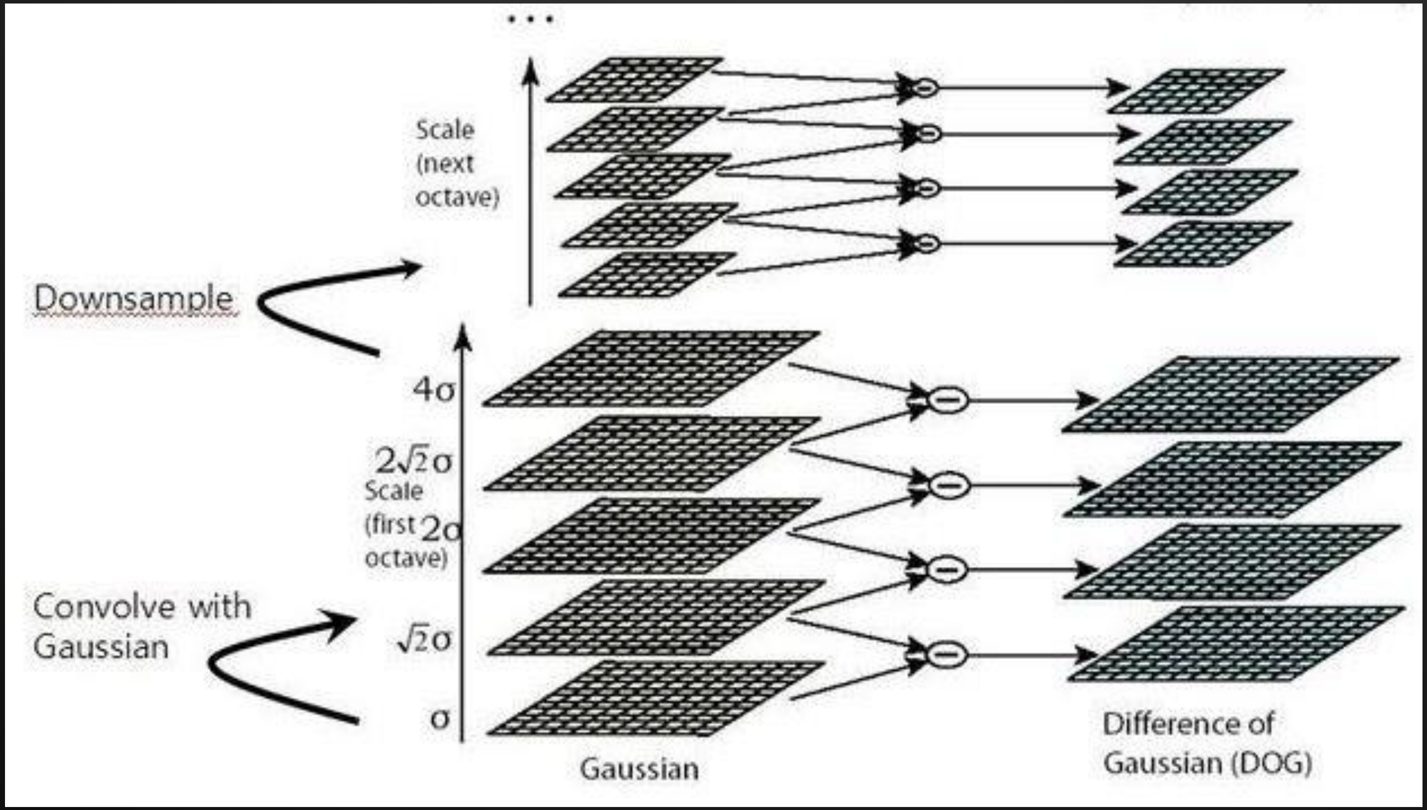
Gaussian Pyramid

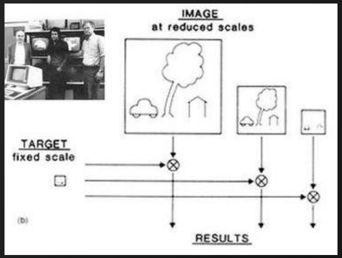


Detection of Scale-Space Extrema

Gaussian Pyramid

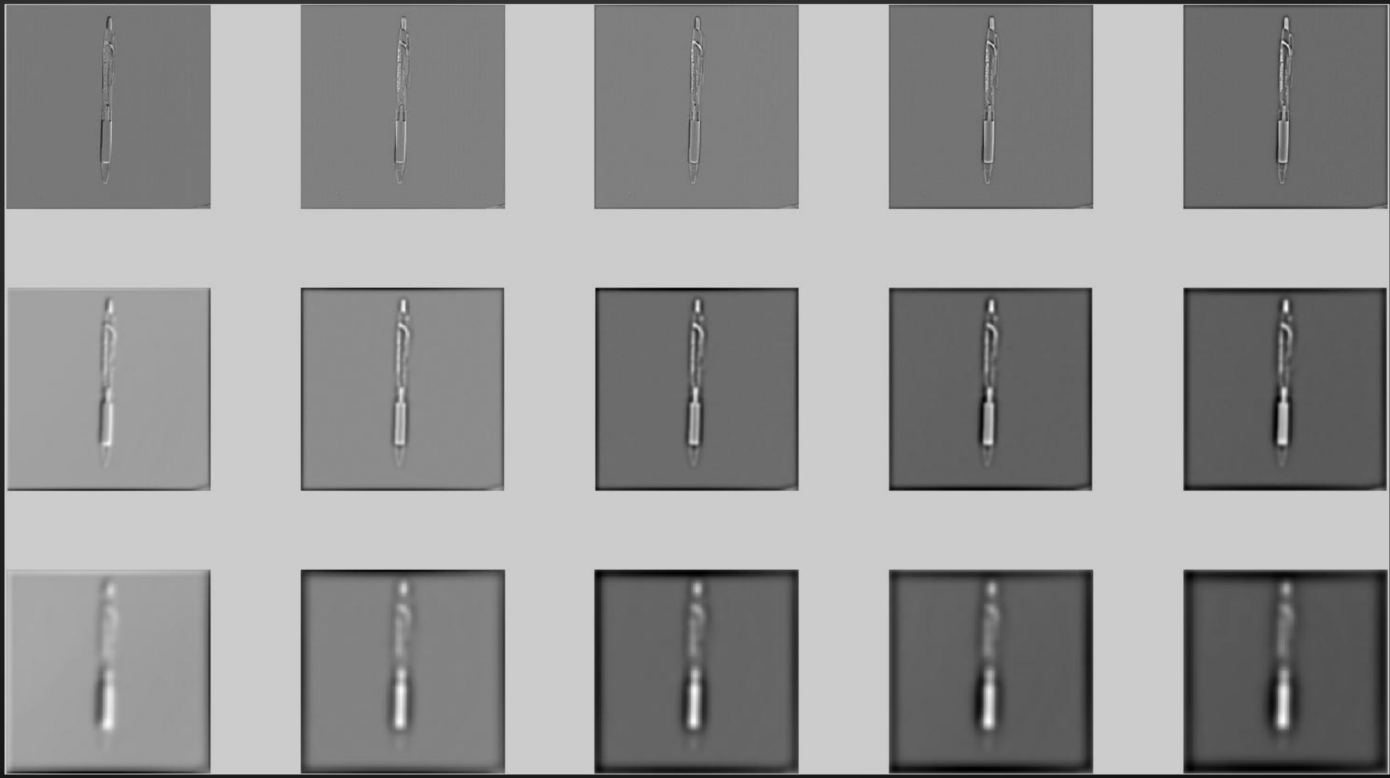
$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$





Detection of Scale-Space Extrema

Gaussian Pyramid



Detection of Scale-Space Extrema

Gaussian Pyramid

- Cascade property of Gaussian Kernerl

$$g(x, y, s1 + s2) = g(x, y, s1) * g(x, y, s2)$$

$$g(x, y, s1 + s2) * I(x, y) = g(x, y, s1) * g(x, y, s2) * I(x, y)$$

$$L(x, y, s1 + s2) = g(x, y, s1) * L(x, y, s2)$$

$$L(x, y, s1 - s1 + s2) = g(x, y, s2 - s1) * L(x, y, s1)$$

$$L(x, y, s2) = g(x, y, s2 - s1) * L(x, y, s1)$$

$$L(x, y, s2) = g(x, y, s2 - s1) * L(x, y, s1)$$

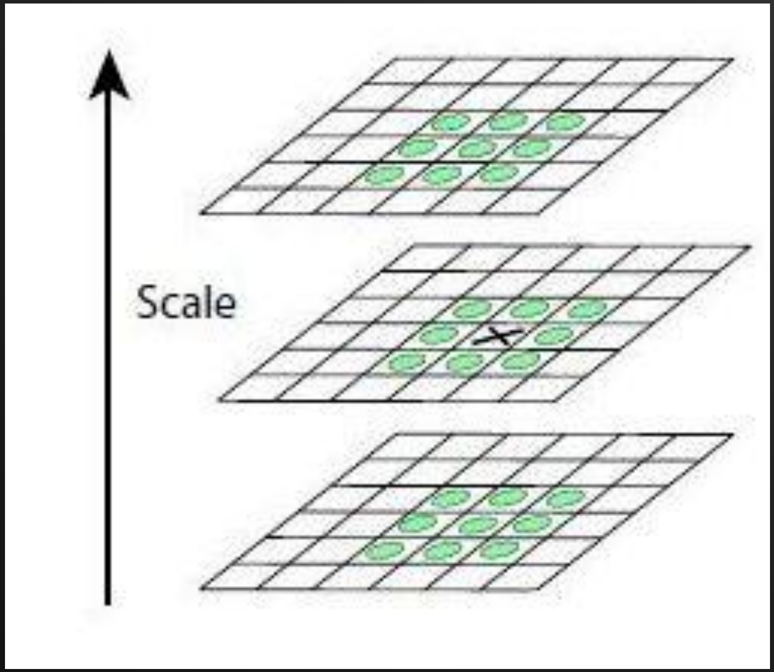
$$sigma_1 = \sigma, \qquad sigma_2 = \sqrt{2}\sigma$$

$$Sigma^2 = \left(\sqrt{2}\sigma\right)^2 - (\sigma)^2$$

$$Sigma = \sqrt{\left(\sqrt{2}\sigma\right)^2 - (\sigma)^2} = \sigma$$

Detection of Scale-Space Extrema

Extrema Detection(candidate)



- Compare to 28 pixel.
- Detect the local maxima and minima.

Accurate Keypoint Localization

Difference of Gaussian function D(X)

$$X = (x, y, \sigma)$$

Taylor series

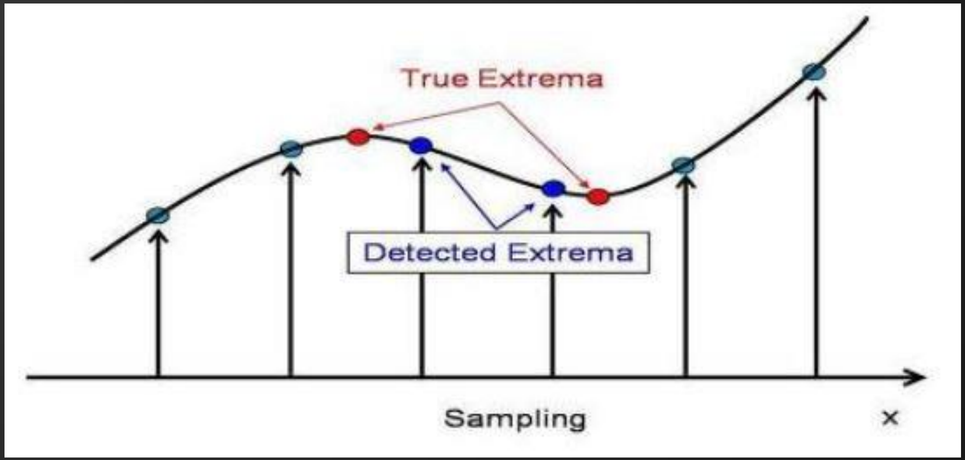
$$D(X) = D + \left(\frac{\partial D}{\partial X}\right)^T X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X$$

Extremum point

$$D'(X) = 0 + \left(\frac{\partial D}{\partial X}\right)^T + \frac{\partial^2 D}{\partial X^2} X$$

$$\frac{\partial^2 D}{\partial X^2} X = -\frac{\partial D}{\partial X}$$

$$X = -\left(\frac{\partial^2 D}{\partial X^2}\right)^{-1} \frac{\partial D}{\partial X}$$

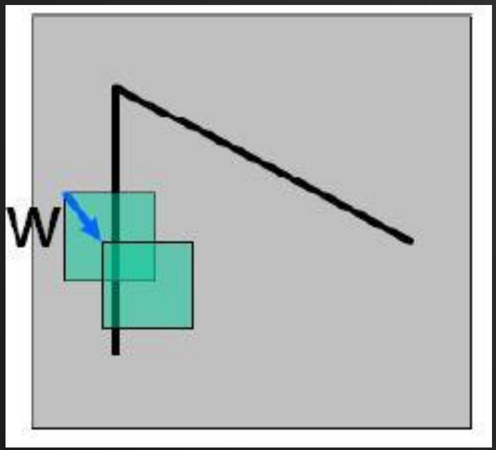


$$D(X) = D + \frac{1}{2}\left(\frac{\partial D}{\partial X}\right)^T X$$

$|D(X)| < 0.3$ discard

Accurate Keypoint Localization

Harris Edge Detection



Base Idea

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

$I(x+u, y+v)$ *Taylor series*

$$I(x+u, y+v) = I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + hight\ order$$

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

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

Accurate Keypoint Localization

Harris Edge Detection

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

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

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

$$= [u \quad v] H \begin{bmatrix} u \\ v \end{bmatrix}$$

Accurate Keypoint Localization

Harris Edge Detection

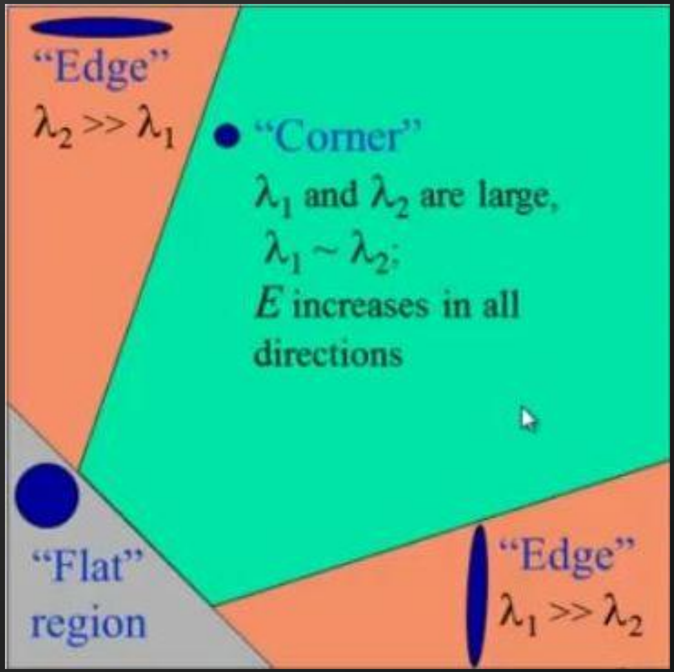
$H's\ Eigenvalue\ is\ \lambda_1, \lambda_2\ (\lambda_1 > \lambda_2)$

$Det(H) = \lambda_1 \lambda_2$

$Trace(H) = \lambda_1 + \lambda_2$

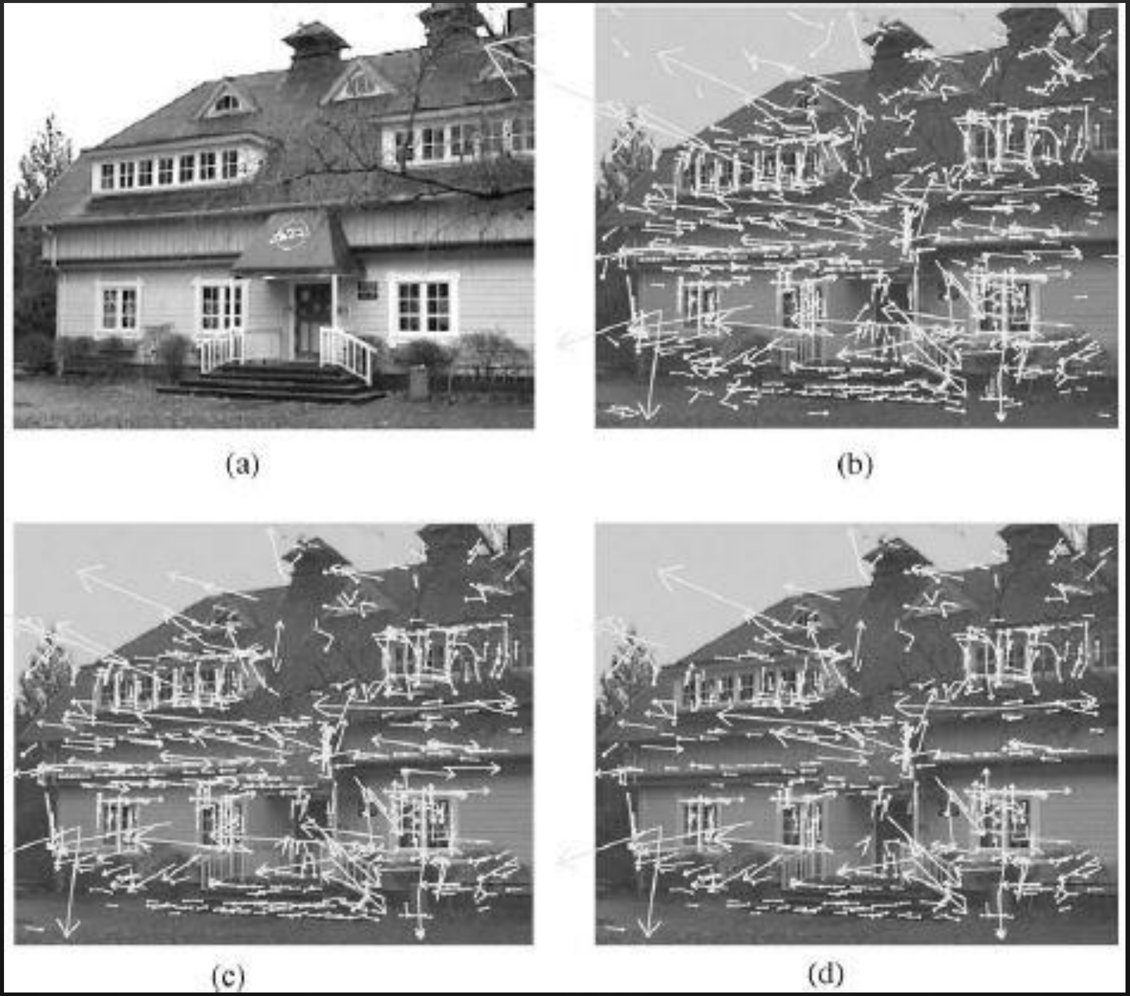
$$\frac{Tr(H)^2}{Det(H)} = \frac{(\lambda_1 + \lambda_2)^2}{\lambda_1 \lambda_2} = \frac{(r+1)^2}{r} \quad at, (\lambda_1 = r\lambda_2)$$

$$\therefore \frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \quad at, (r = 10)$$



Why remove the Edge?

Accurate Keypoint Localization



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

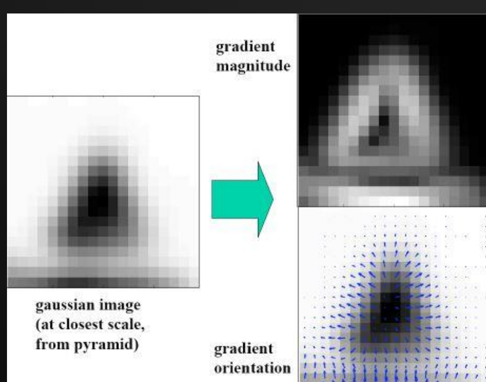
Orientation Assignment

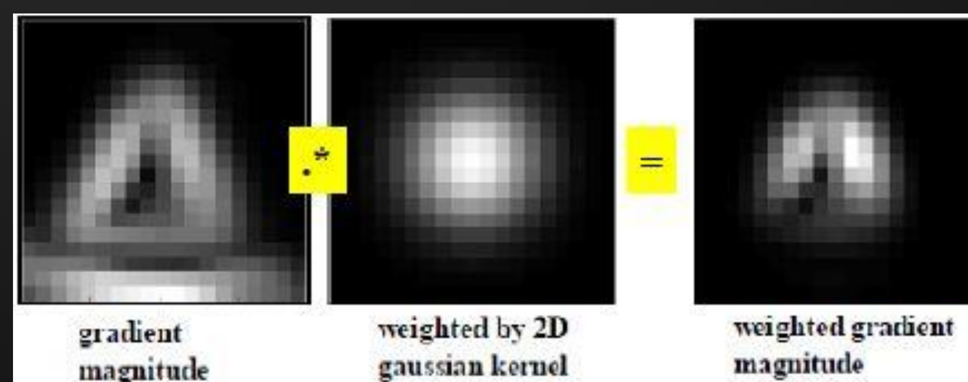
For 16x16 Pixel

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

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

In addition, each sample added to the histogram is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a that is 1.5 scale of the keypoint



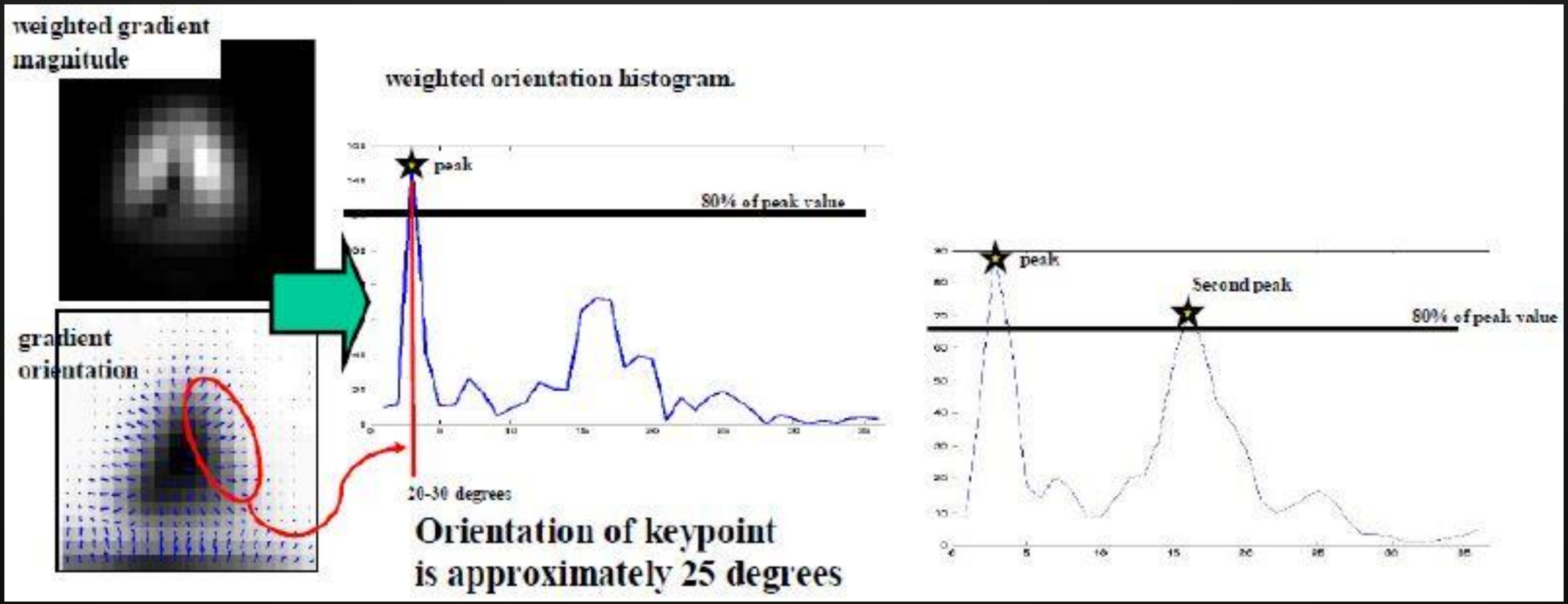


Orientation Assignment

Make Histogram graph

The orientation histogram has 36 bins covering the 360 degree range of orientation.

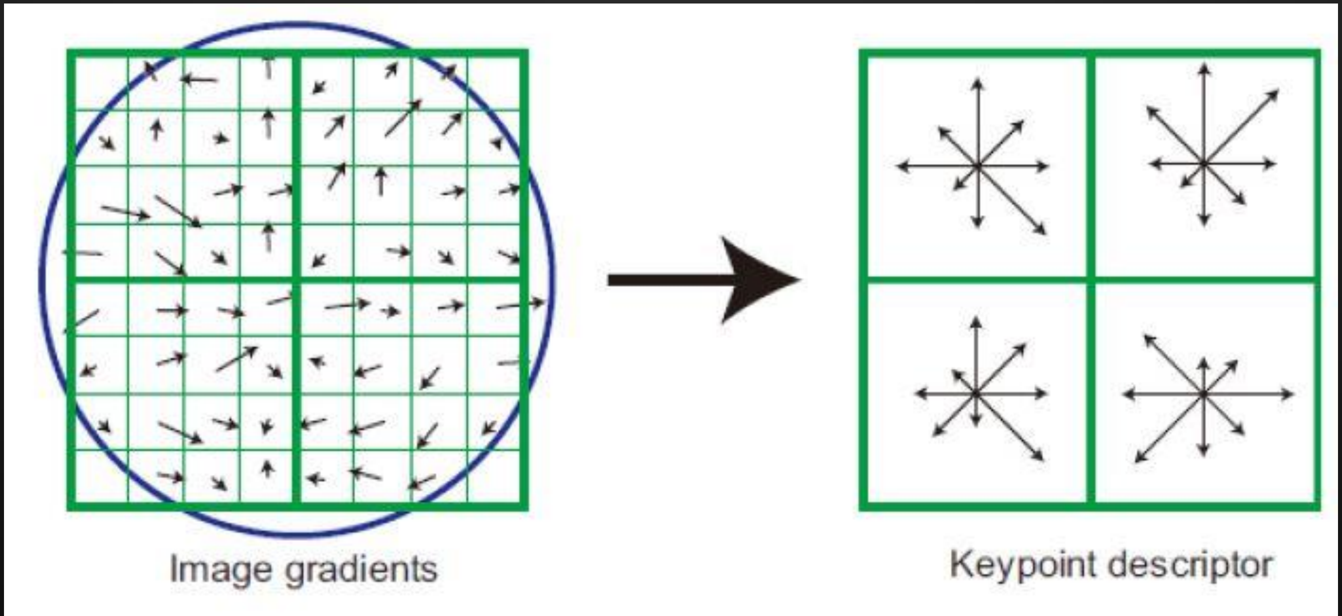
The highest peak in the histogram is detected, and then any other local peak that is within 80% of the highest peak is used to also create a keypoint with that orientation.



The local image Descriptor

Keypoint Descriptor

We know the Magnitude, orientation of Keypoint.
But this is not special feature.
This case, σ of Gaussian weighted function is half of Descriptor window size. In addition orientation is subtract the orientation of previous session.
Finally create the histogram for the 4x4 pixel.



Application to Object Recognition

Keypoint Matching

Successful keypoint matching is very small Euclidean distance.

Euclidean distance

$$A = (a_1, a_2, a_3, \dots, a_n)$$

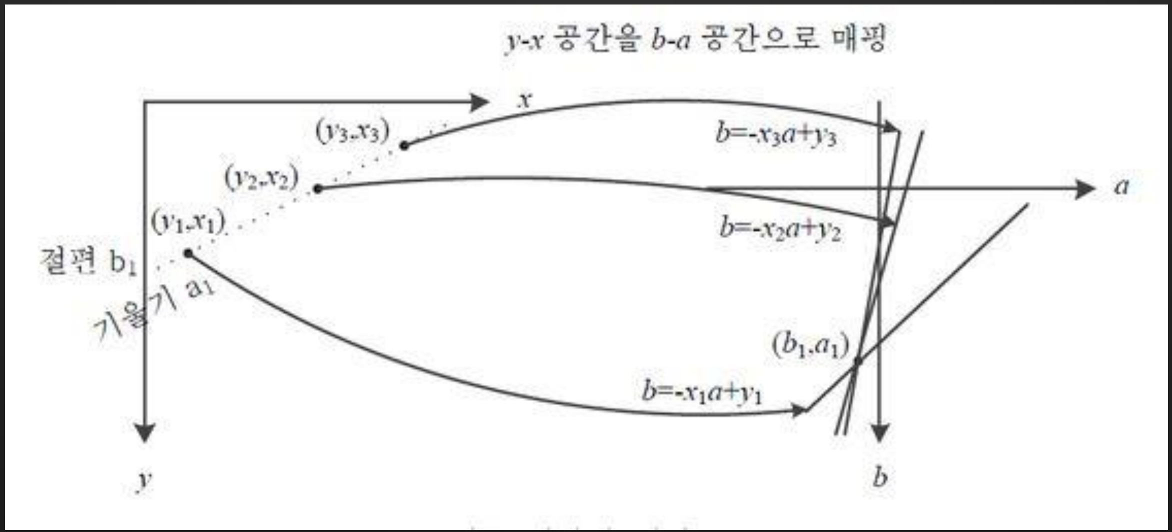
$$B = (b_1, b_2, b_3, \dots, b_n)$$

$$D = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$

However, many features from an image will not have any correct match in the DB. Thus a global threshold on distance to the closest feature does not perform well.
A more effective measure is obtained by comparing the distance of the closest neighbor to that of the second-closest neighbor.

Application to Object Recognition

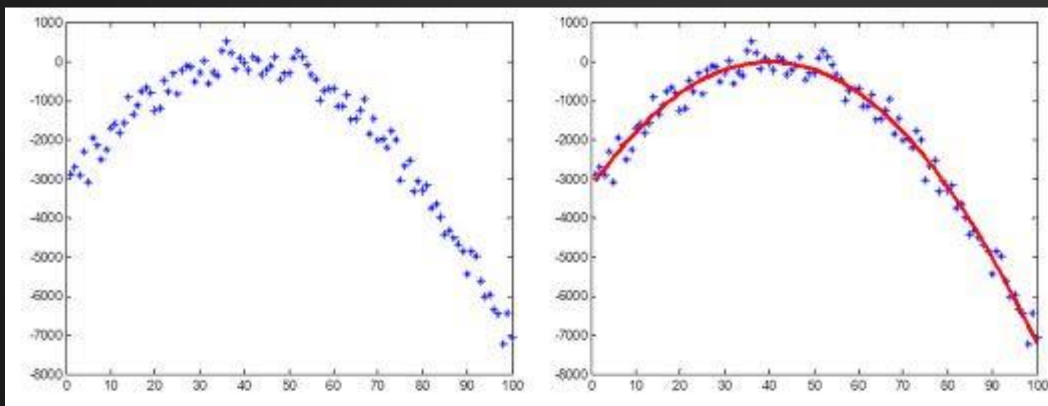
Hough Transform or RANdom Sample Consensus(RANSAC)

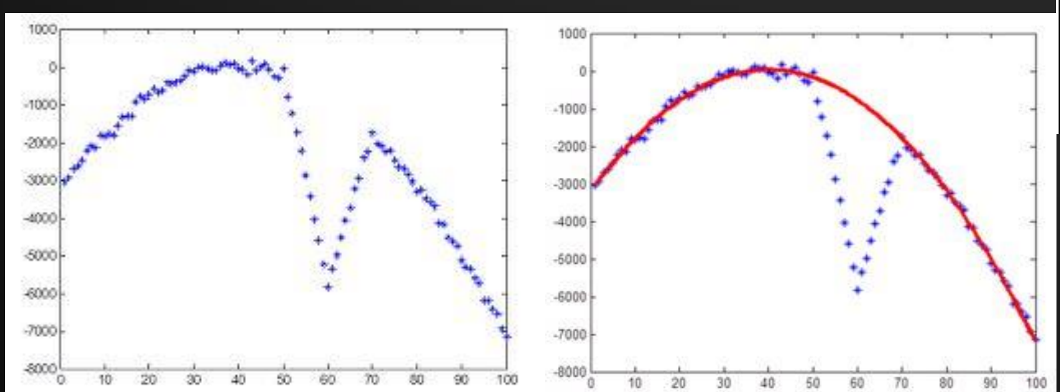


$y_i = ax_i + b$ $b = -x_i a + y_i$

Application to Object Recognition

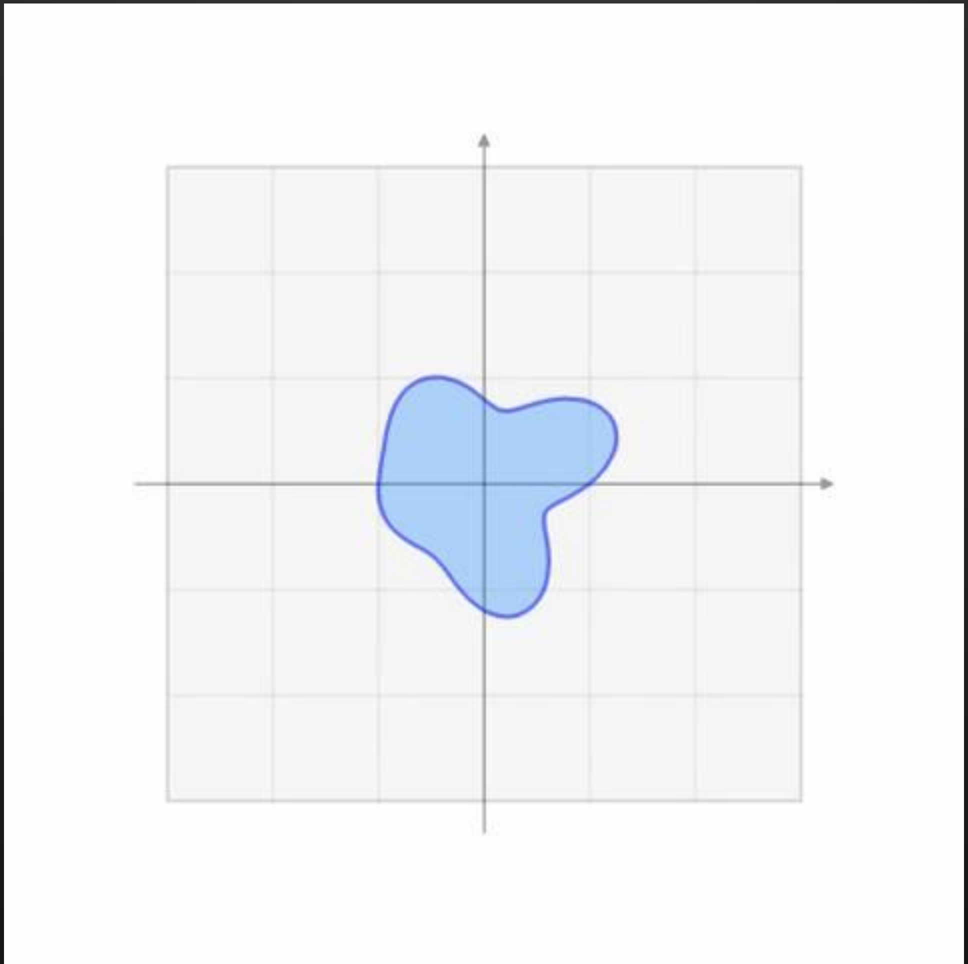
Hough Transform or RANdom Sample Consensus(RANSAC)





Application to Object Recognition

Affine transform



Application to Object Recognition

Affine transform

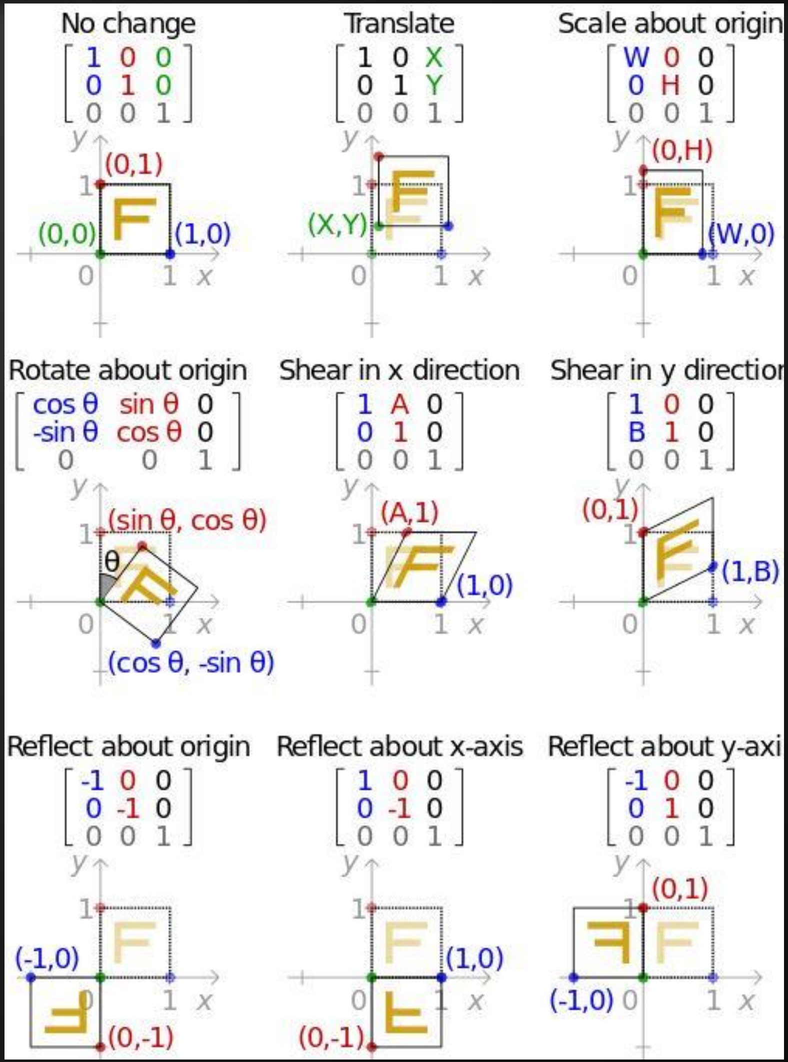
$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m1 & m2 \\ m3 & m4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \qquad \begin{aligned} Ax &= b \\ x &= A^{-1}b \end{aligned}$$

Unknown variable

$$m1, m2, m3, m4, t_x, t_y$$

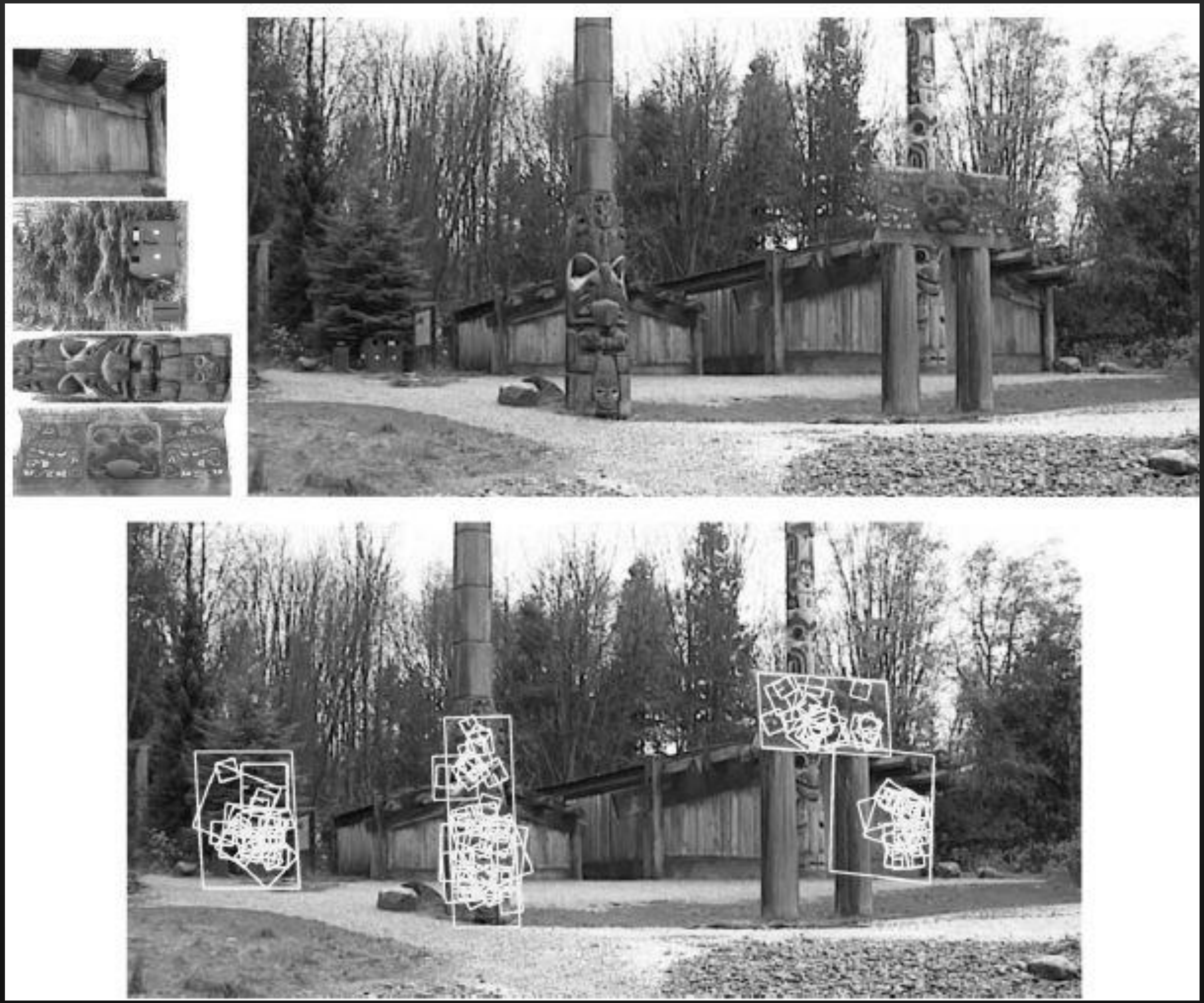
One matching has two equations.
So requires three matching.

$$\begin{bmatrix} x_1 & y_1 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_1 & y_1 & 0 & 1 \\ x_2 & y_2 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_2 & y_2 & 0 & 1 \\ x_3 & y_3 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_3 & y_3 & 0 & 1 \end{bmatrix} \begin{bmatrix} m1 \\ m2 \\ m3 \\ m4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix}$$



Result IMG

Result IMG



Q & A

YOON. H C

YOON. H C