ABV-Indian Institute of Information Technology and Management Gwalior

==========

Scale Invariant Feature Transform

(Detector-cum-Descriptor)

(ITIT-9507)

==========

Instructor – Dr. Sunil Kumar

Office – 206, F-Block (V), Tel No – 0751-2449710 (O), Email - snk@iiitm.ac.in

Mob - 8472842090

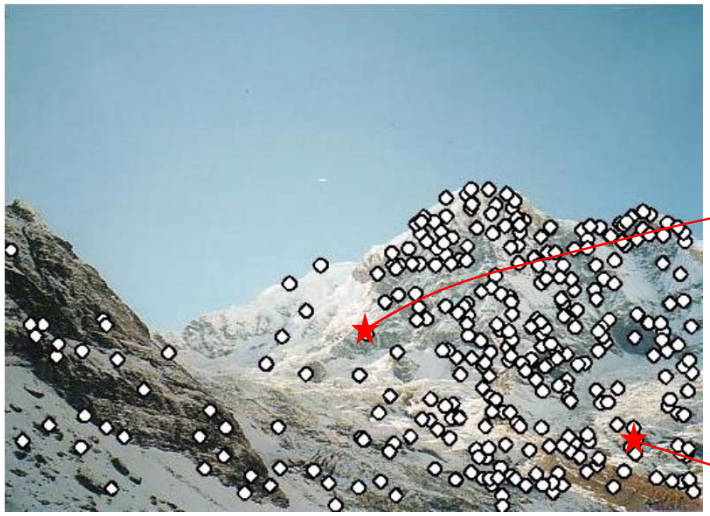
Uses of Interest Points

- Image correspondence
- Object tracking
- Object recognition
- Point matching for computing disparity
- 3D reconstruction
- Image retrieval and indexing
- Robot navigation
- Computing camera parameters (Stereo calibration)
- Image stitching
- … and so on.

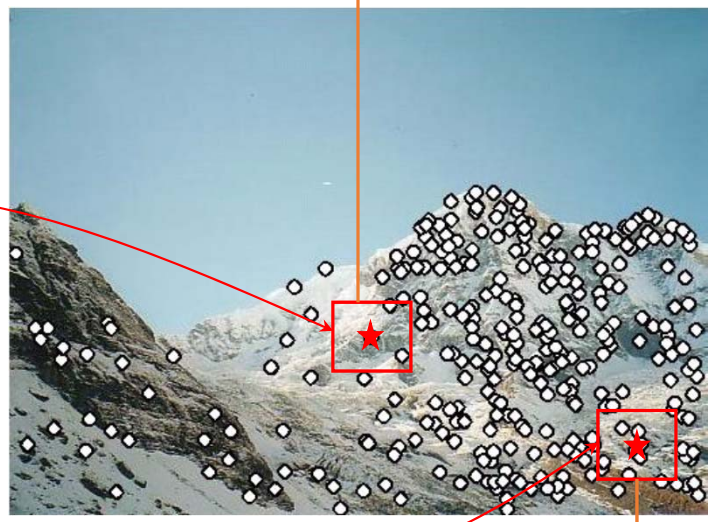
Applications of Interest Points : Image Matching

- Problem – Given two or more images of a scene captured at different angles, the goal is to match points from one image to the corresponding points in the images of other views.

$$\mathbf{x}_m = [x_{m1}, x_{m2}, x_{m3}, \cdots, x_{mn}]'$$



Interest Point Detection



Interest Point Detection

$$\mathbf{x}_k = [x_{k1}, x_{k2}, x_{k3}, \cdots, x_{kn}]'$$

Image Credit : David G. Lowe

Requirements

Robust “Interest Point Detector” and robust “Descriptor” around each interest point.

What to do?

So, there is a need of an Interest Point Detector that is Scale invariant as well.

SIFT

Detector-cum-Descriptor

Scale Invariant Feature Transform

Distinctive Image Features from Scale-Invariant Keypoints

Computer Science Departme *, Vancouver, B.C., Canada*

Received January 10, 20 *epted January 22, 2004*

Abstract. This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features from many images. This paper also describes an approach to using these features for object recognition. The recognition proceeds by matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm, followed by a Hough transform to identify clusters belonging to a single object, and finally performing verification through least-squares solution for consistent pose parameters. This approach to recognition can robustly identify objects among clutter and occlusion while achieving near real-time performance.

Descriptor

- Constructed based on local features around each interest point.
- Properties of a Descriptor:
  - ✓ Repeatability
    - The same feature can be found in several images despite geometric and photometric transformations
  - ✓ Saliency
    - Each feature has a distinctive description
  - ✓ Compactness and efficiency
    - Many fewer features than image pixels
  - ✓ Locality
    - Should capture local properties, and so robust to clutter and occlusion

Slide Credit – Mubarak Shah & Kristen Grauman

Major Contributions of SIFT

I. Extracting distinctive Invariant Feature

- Feature that can be correctly matched with high probability against a large database of features from many images.

II. Invariant to image scaling and rotation

III. Features are Robust to

- Affine Transformation
- Change in 3D view points
- Addition of Noise
- Change in illumination

The Core Idea of SIFT is to "Transform Image to Scale Space Pyramid"

Invariant to Scale and View-Point

- Intuitive example from SIFT :
  - Image – 1 and Image – 2 : change in scale and change in view-point

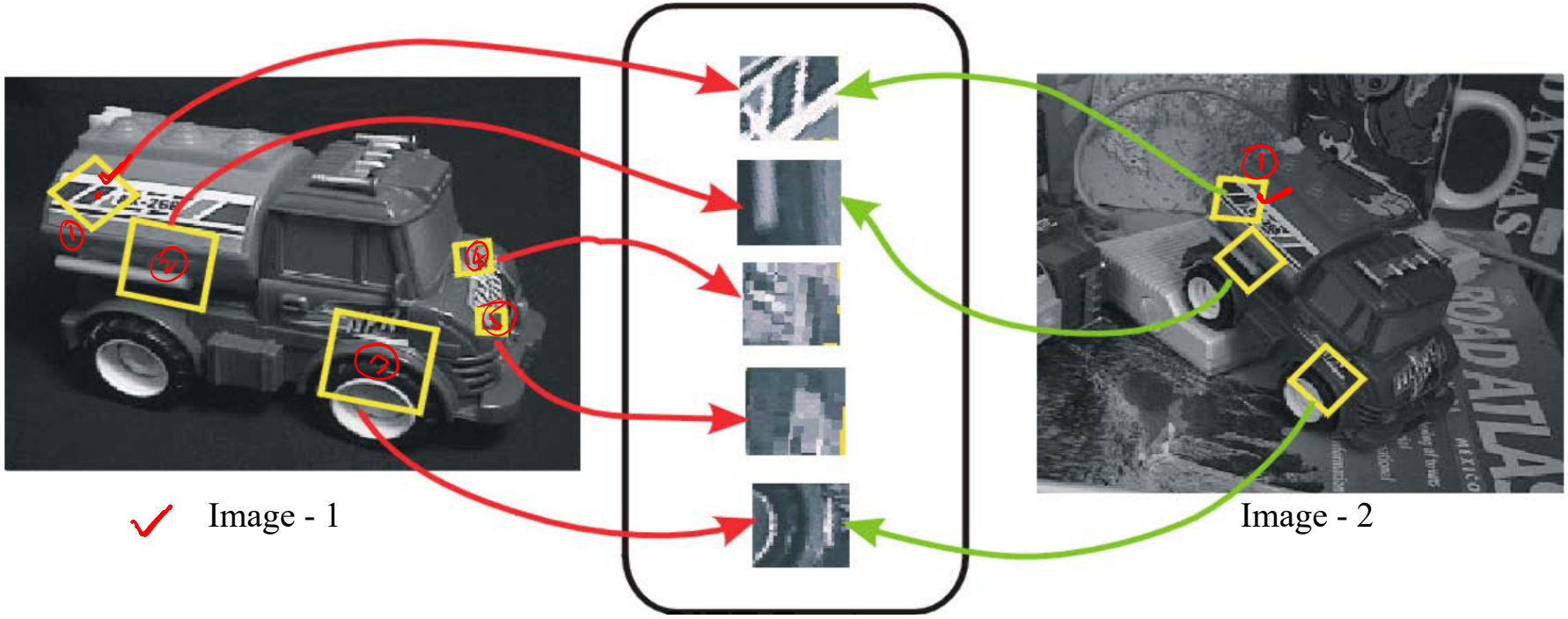


Image - 1

Image - 2

SIFT Features

Major Stages in SIFT for Extracting Distinctive Image Features

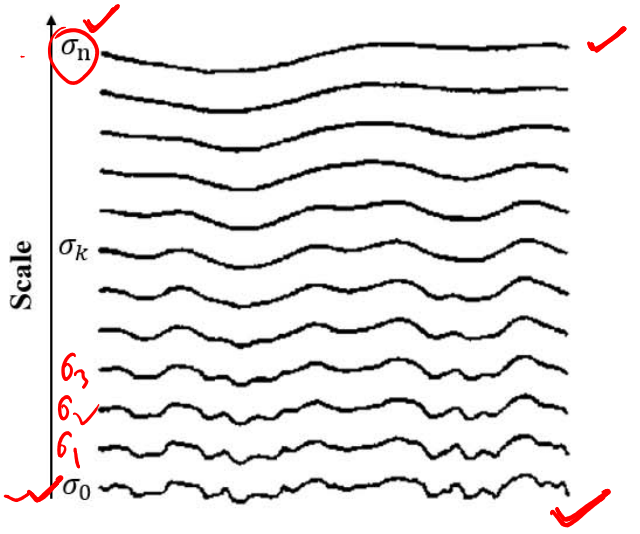
- Stage – 1 : Scale-Space Extrema Detection:
  - Transform image to scale space and search for potential interest point in all scale
- Stage – 2 : Key-point Localization :
  - Detecting accurate interest point based on their stability
- Stage – 3 : Orientation Assignment :
  - Assign orientation (direction) to each of the accurately detected interest points
- Stage – 4 : Key-point Descriptor
  - Describe the key point with a high-dimensional feature vector

Selection of Scale "Sigma"

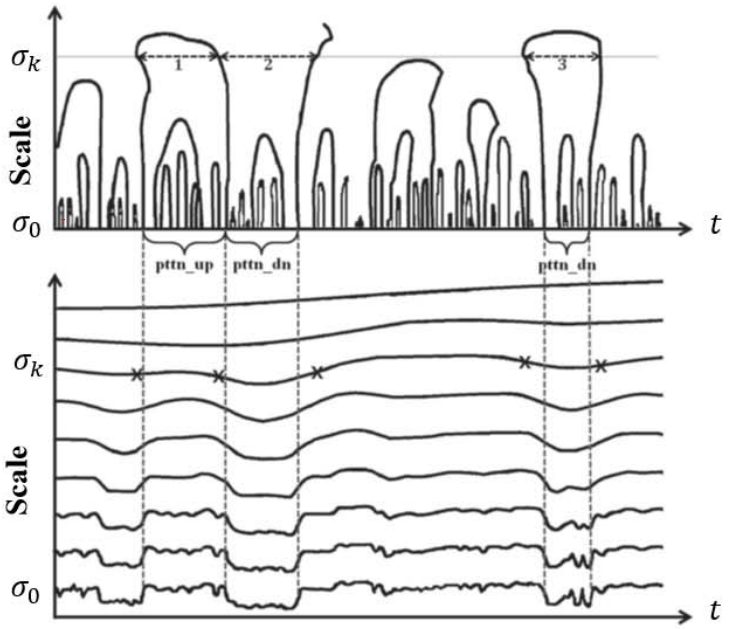
- How do we choose "sigma" (scale) value for Canny and Laplacian of Gaussian Edge Detector?
  - Marr-Hildreth Edge Detector
    - Laplacian of Gaussian - $\nabla^2(I_o * g) = I_o * \nabla^2 g$
  - Canny Edge Detector
    - Gradient of Gaussian - $\nabla I_s = \nabla(I_o * g) = I_o * \nabla g$
  - Possible solution – Use multiple sigma values
    - Issue – How do we combine multiple edge maps to one edge map (each sigma, we will get one edge map)
    - Marr-Hildreth – Zero-crossings that coincide at many scales may be the potential edge points.

Scale-Space-Filtering

- Andrew P. Witkin, addressed this issue in his paper titled "Scale-Space Filtering":
  - Consider the whole spectrum and look for the stable curvature zero-crossing



Multiple smoothen version of original 1D-signal

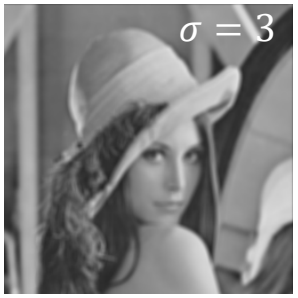


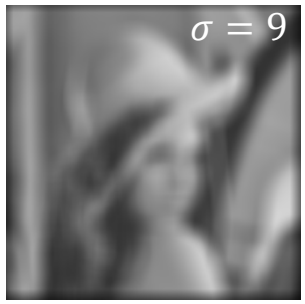
- Stability of a node is the range of scale over which interval exits

Stage – 1 : Scale-Space-Extrema Detection

- David G. Lowe : - Let us use all possible scales to identify scale invariant features
- Scale space : $I_o * G(x, y, \sigma_k) \ \forall k$

Original image





Stage – 1 : Scale-Space-Extrema Detection

- SIFT utilizes "Difference of Gaussian (DOG)" pyramid ≡ Laplacian pyramid
  - Produce the most stable image features compared to a range of other possible image functions, such as the gradient, Hessian, or Harris corner function.

Stage – 1 : Scale-Space-Extrema Detection

- SIFT utilizes "Difference of Gaussian (DOG)" pyramid ≡ Laplacian pyramid

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

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

- Difference of Gaussian (DOG) :

$$\begin{aligned} D(x, y, \sigma) &= L(x, y, k\sigma) - L(x, y, \sigma) \\ &= G(x, y, k\sigma) * I(x, y) - G(x, y, \sigma) * I(x, y) \\ &= [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y) \end{aligned} \quad \rightarrow (1)$$

- Let $\frac{dG}{d\sigma} = \sigma \nabla^2 G$: Heat diffusion Equation

$$\Rightarrow \sigma \nabla^2 G = \frac{dG}{d\sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma}$$

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

$$\left[\frac{dH}{dk} = k \nabla^2 H\right]$$

Stage – 1 : Scale-Space-Extrema Detection

- SIFT utilizes "Difference of Gaussian (DOG)" pyramid ≡ Laplacian pyramid
  - Difference of Gaussian (DOG) :

$$D(x, y, \sigma) = [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y)$$
$$\approx (k-1)\sigma^2 \nabla^2 G * I(x, y)$$

$L(x, y, k\sigma)$ $\quad$ $L(x, y, \sigma)$ $\quad$ $D(x, y, \sigma)$

Difference of Gaussian (DOG) ≡ Laplacian of Gaussian

Stage – 1 : Scale-Space-Extrema Detection

- SIFT utilizes "Difference of Gaussian (DOG)" pyramid ≡ Laplacian pyramid

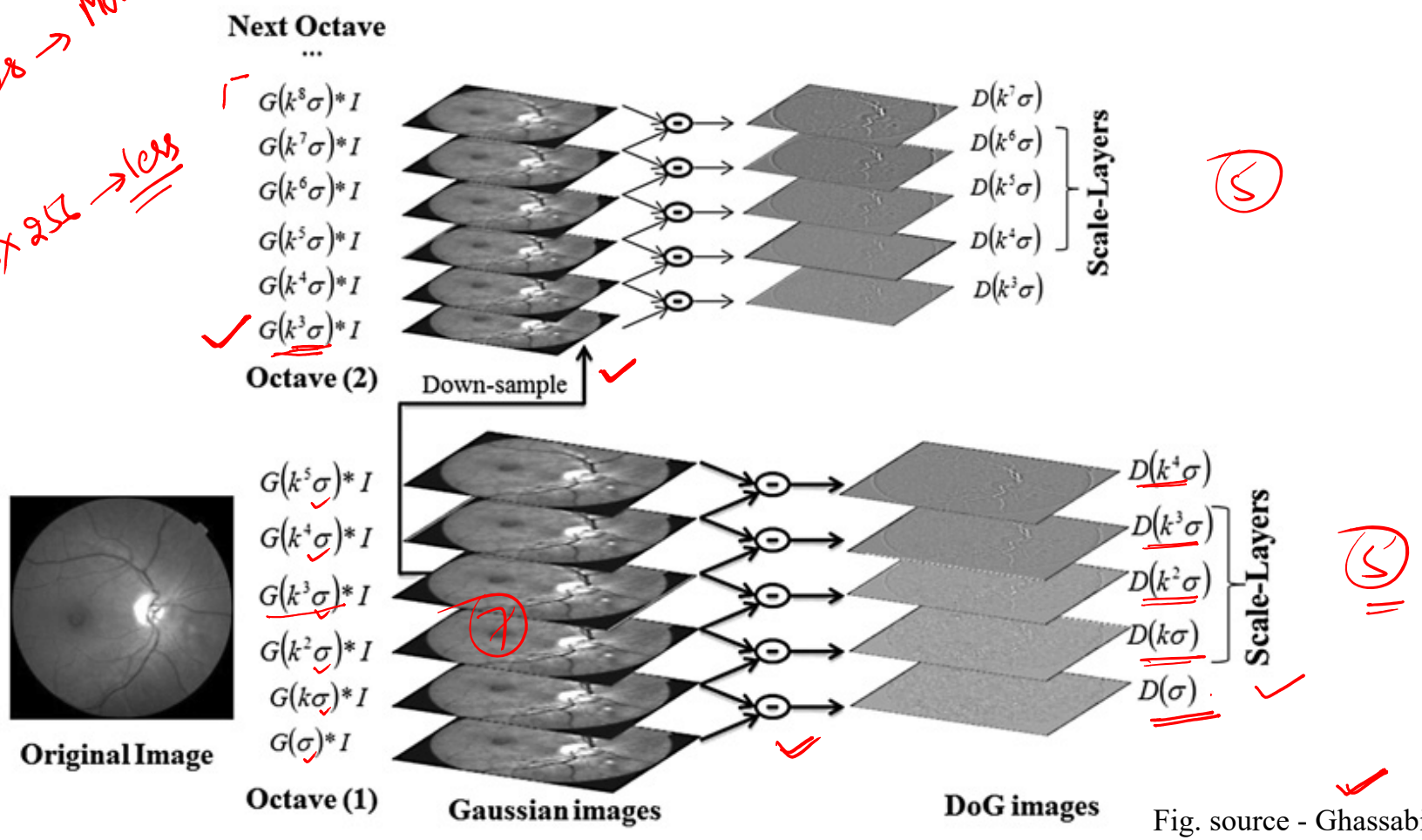


Fig. source - Ghassabi, Z.R. , IET

Stage – 1 : Scale-Space-Extrema Detection

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

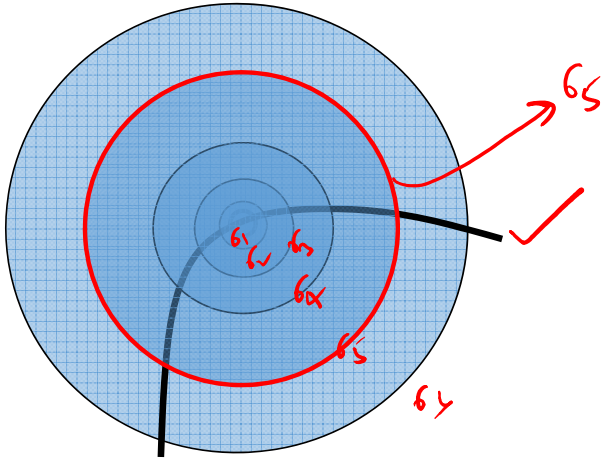
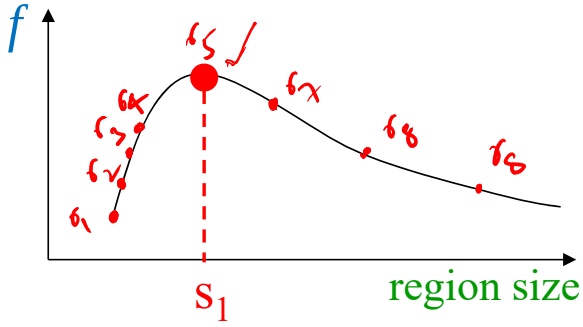


Image 1



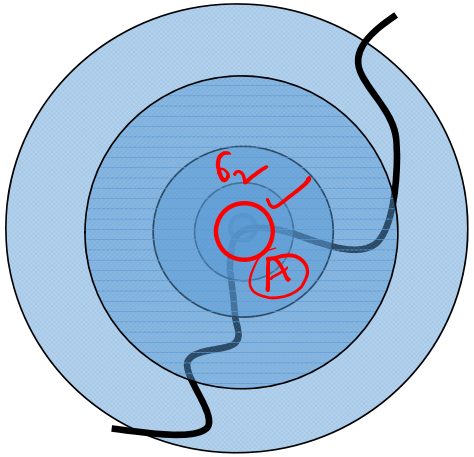
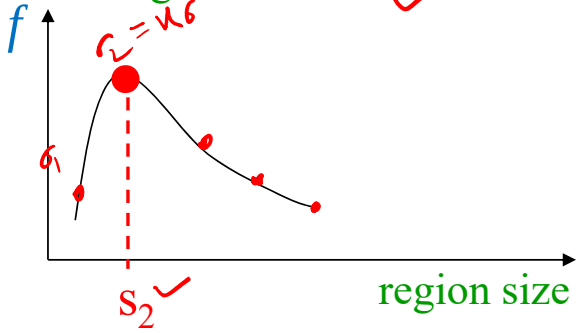
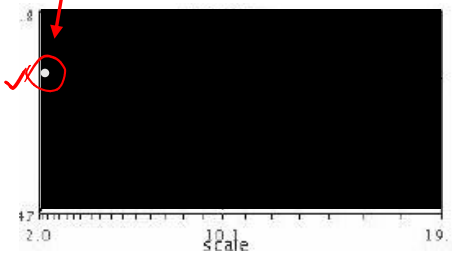


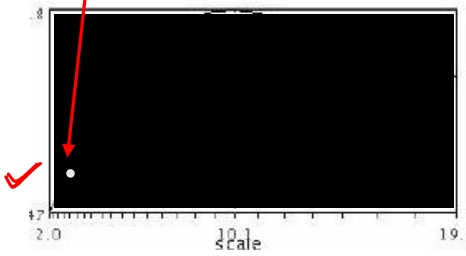
Image 2



Stage – 1 : Scale-Space-Extrema Detection

- Example on real image :

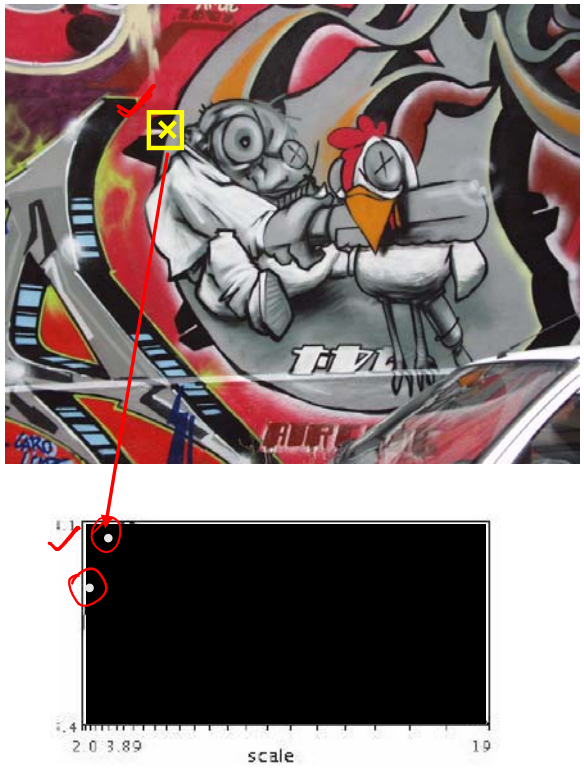




Slide credit -K. Grauman, B. Leibe

Stage – 1 : Scale-Space-Extrema Detection

- Example on real image : Response for increasing scale



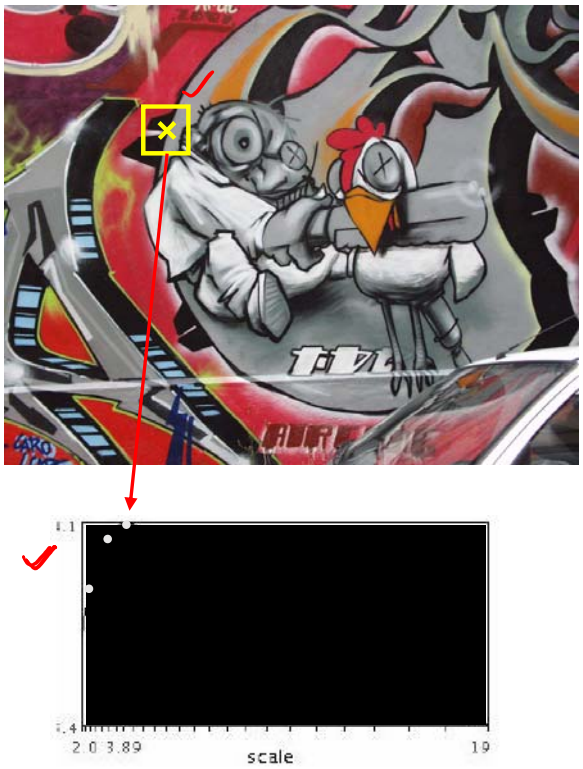
$f(I_{i_1 \ldots i_m}(x,\sigma))$

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

Slide credit -K. Grauman, B. Leibe

Stage – 1 : Scale-Space-Extrema Detection

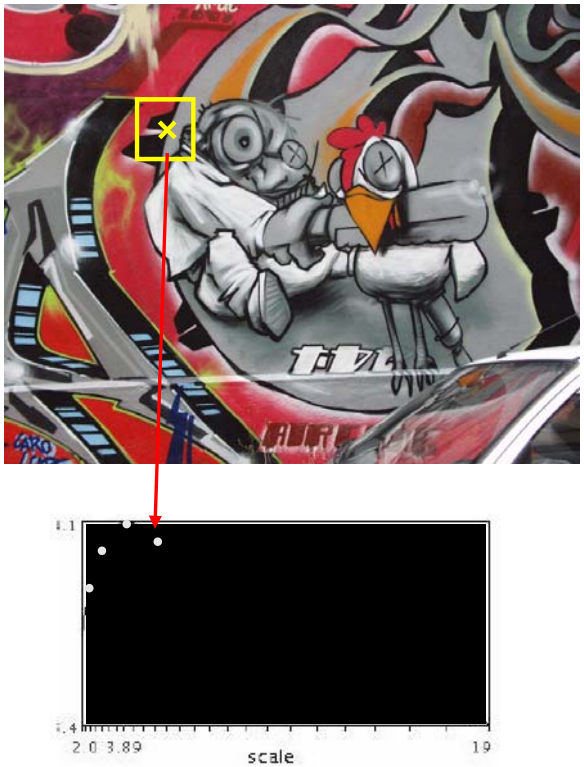
- Example on real image : Response for increasing scale



Slide credit -K. Grauman, B. Leibe

Stage – 1 : Scale-Space-Extrema Detection

- Example on real image : Response for increasing scale



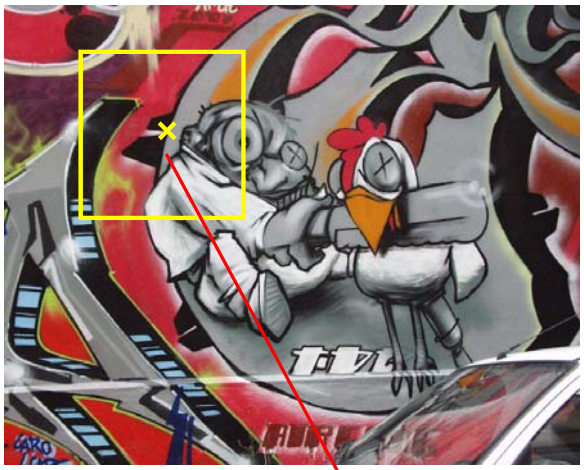
$f(I_{i_1 \ldots i_m}(x,\sigma))$

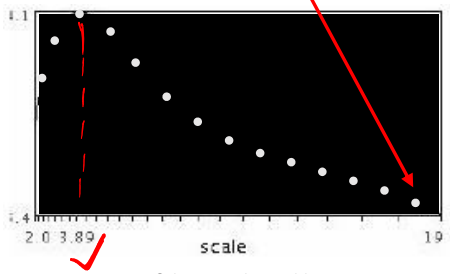
$f(I_{i_1 \ldots i_m}(x',\sigma))$

Slide credit -K. Grauman, B. Leibe

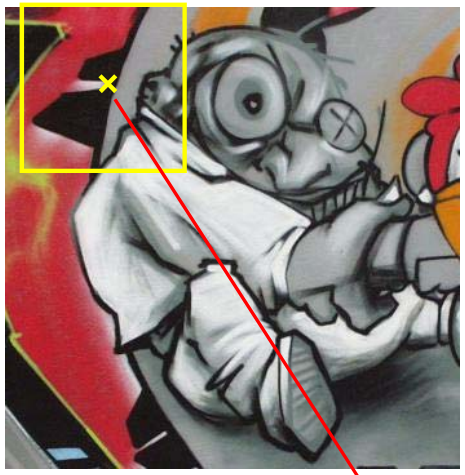
Stage – 1 : Scale-Space-Extrema Detection

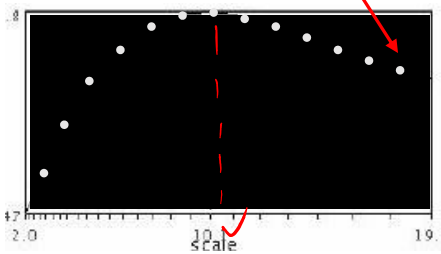
- Example on real image : Response for increasing scale





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



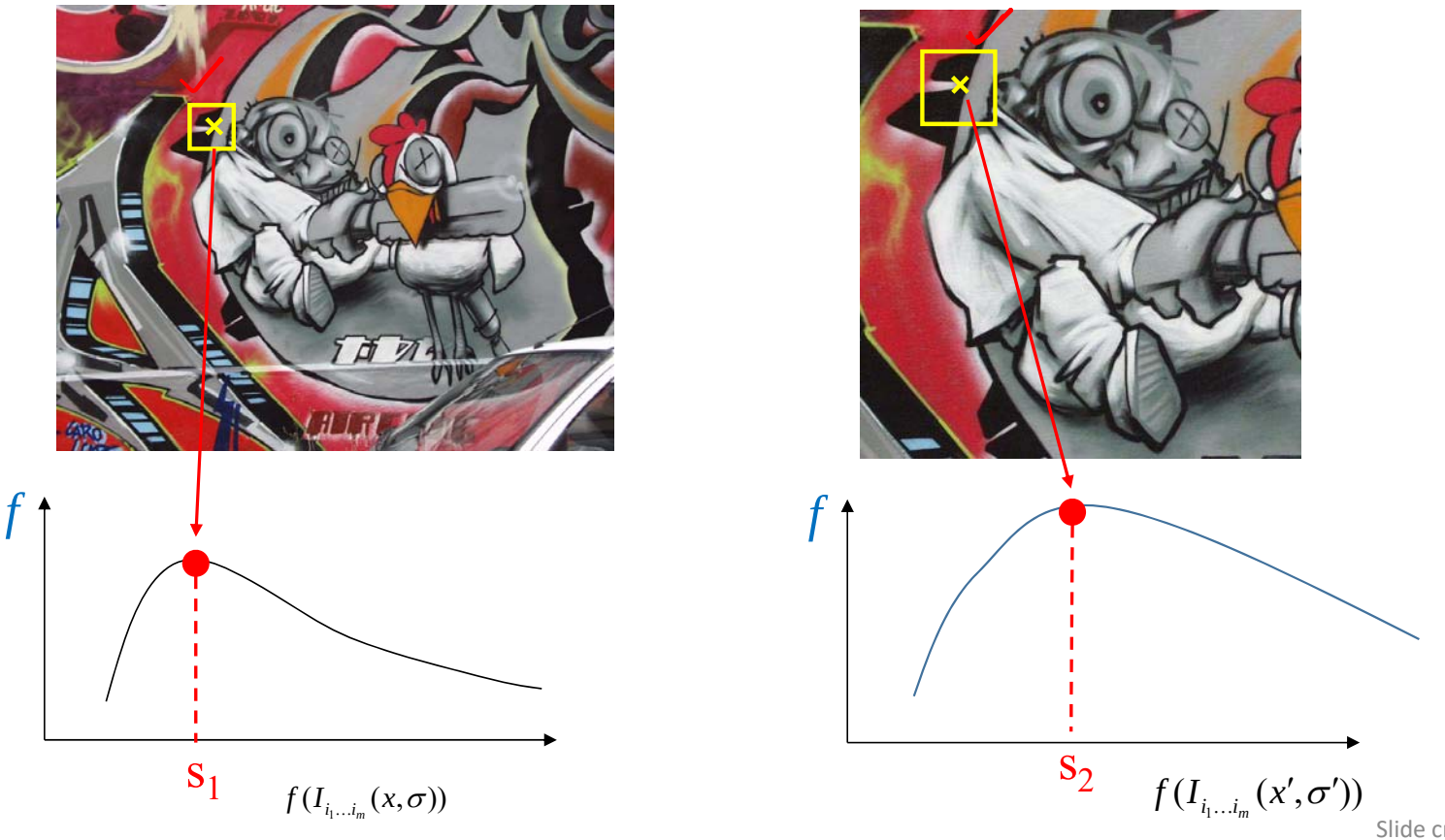


$f(I_{i_1 \ldots i_m}(x',\sigma))$

Slide credit -K. Grauman, B. Leibe

Stage – 1 : Scale-Space-Extrema Detection

- Example on real image : Response for increasing scale



Slide credit -K. Grauman, B. Leibe

Stage – 1 : Scale-Space-Extrema Detection

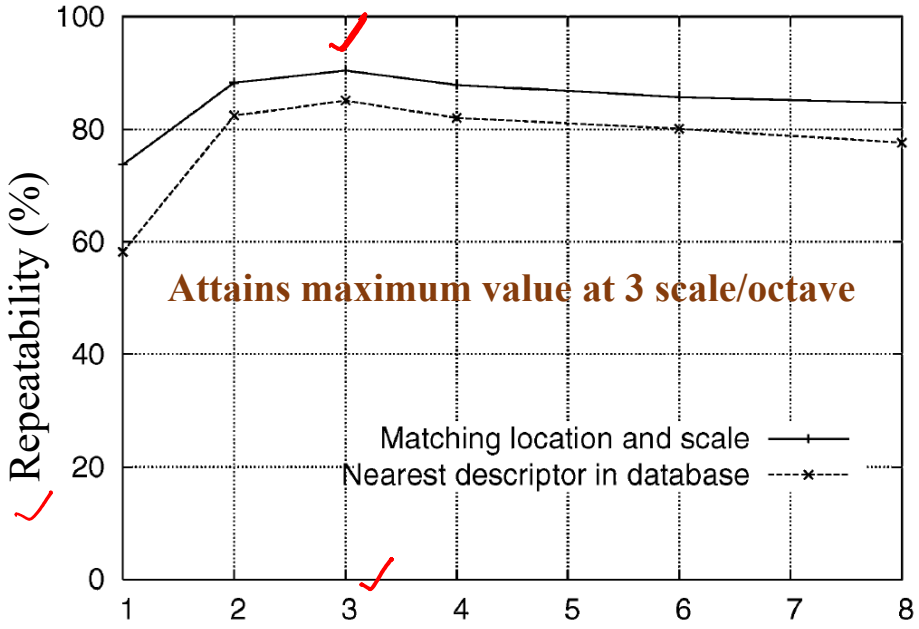
- Following questions comes in mind – Experimentally computed values
  - What should be the initial value of sigma (σ) : $\sigma_{initial} = 1.6$
  - What should be the no. of scale/octave : 3 scales/octave
  - What is optimal value of k : $k = \sqrt{2}$

- Difference of Gaussian (DOG) :

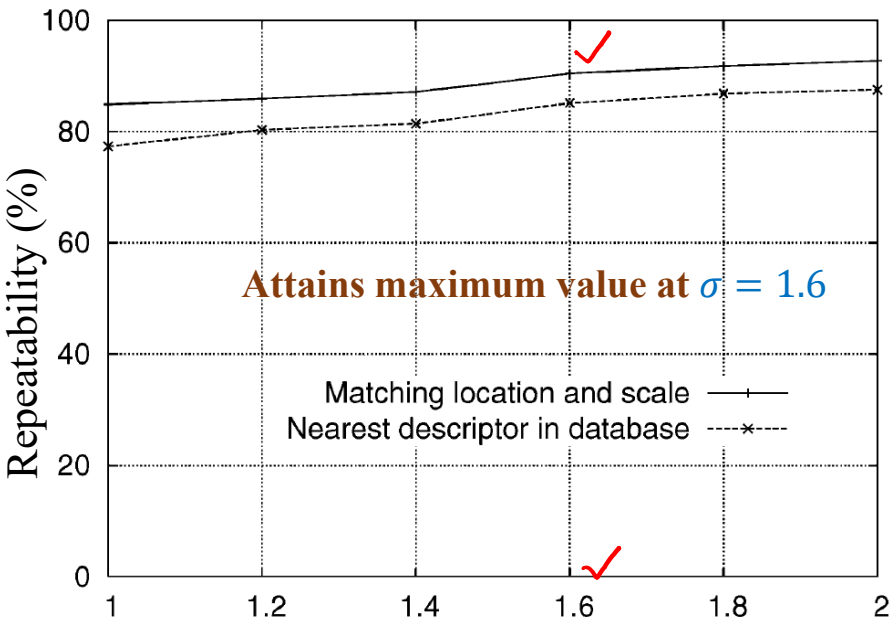
$$D(x, y, \sigma) = [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y)$$
$$\approx (k - 1)\sigma^2 \nabla^2 G * I(x, y)$$

Stage – 1 : Scale-Space-Extrema Detection

- Experimental values :



No. of scale/octave



Initial sigma (σ) for each octave

Stage – 1 : Scale-Space-Extrema Detection

- For different scales in each octave, do this step to detect potential interest point.

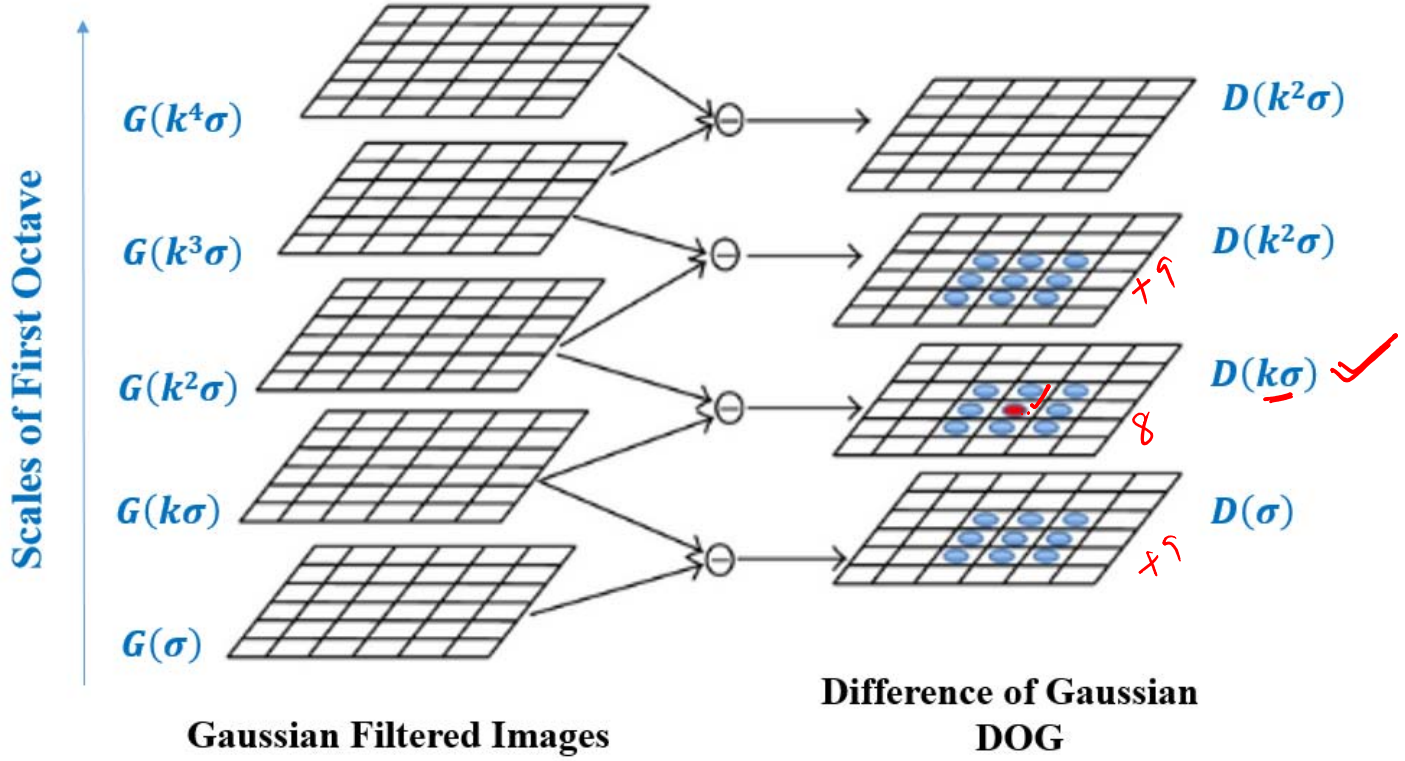
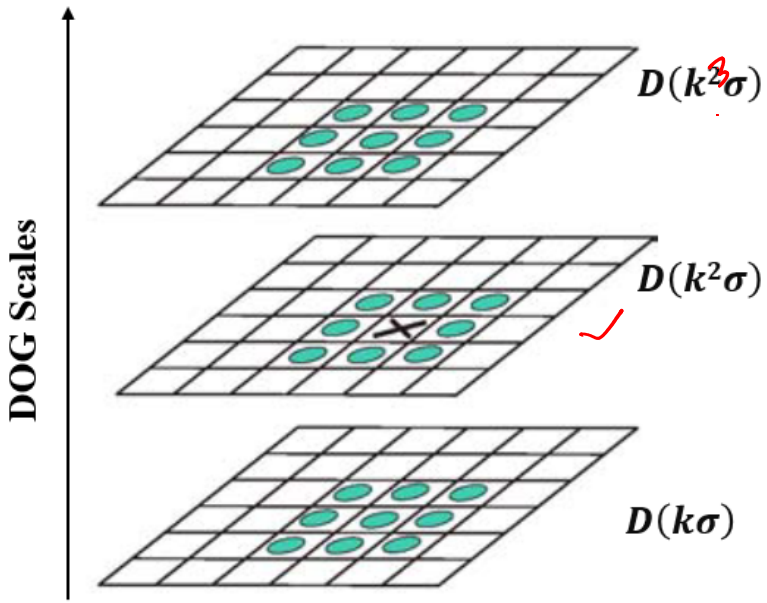


Fig. source – Mingming Huang

Stage – 1 : Scale-Space-Extrema Detection

- Way to detect potential interest point.



- Compare a pixel (say x) with 26 pixels in current and adjacent scales (Green circles : ●)
- Select pixel "x" as potential interest point if it greater / smaller than all 26 pixels.
- Do for all scales and for all octaves
- Lot of peaks may be qualified (computationally expensive)
- Need to find stable

Stage – 2 : Key-Point Localization

- Sub-stage-1 of key-point selection :

(a)

(b) : 832 key-points

Figure : Selection of key-point based on extrema detection. Key-points are displayed as vectors indicating scale, orientation, and location.

Stage – 2 : Key-Point Localization

- Sub-stage-2 : Removal of false-positive key-points
  - False-positive key points may be because of poor lighting condition, low-contrast, etc.
  - Poorly localized key-points along edges.

- Assume $DOG(x, y, \sigma)$ as a surface and then find the point where DOG is maximum/minimum compared to threshold Th.

- Taylor series of $DOG(x, y, \sigma)$ $about$ $(0,0,0)$:

$$DOG(x, y, \sigma) = D + \frac{dD^T}{d\mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\frac{d^2D}{d\mathbf{x}^2}\mathbf{x} + \cdots$$

Where, $\mathbf{x} = \begin{bmatrix} x \\ y \\ \sigma \end{bmatrix}$

Stage – 2 : Key-Point Localization

- Sub-stage-2 : Removal of false-positive key-points
  - For max or min : $\frac{dDOG(x,y,\sigma)}{d\mathbf{x}} = 0$

$$\Rightarrow \frac{dD}{d\mathbf{x}} + \frac{d^2D}{d\mathbf{x}^2}\mathbf{x} = \begin{bmatrix}0\\0\\0\end{bmatrix}$$

$$\Rightarrow \frac{d^2D}{d\mathbf{x}^2}\mathbf{x} = -\frac{dD}{d\mathbf{x}}$$

$$\Rightarrow A_{3\times3}\mathbf{x} = -B_{3\times1}$$

$$\therefore \mathbf{x} = -(A_{3\times3})^{-1}B_{3\times1} = -\frac{d^2D}{d\mathbf{x}^2}\frac{dD}{d\mathbf{x}} = \mathbf{p}\ (\mathbf{say})$$

Stage – 2 : Key-Point Localization

- Sub-stage-2 : Removal of false-positive key-points
  - If $|D(\mathbf{p})| > Th$ then retain the pixel (key-point) corresponding to $D(\mathbf{p})$.

(b) : 832 key-points

(c) : 729 key-points remains with $Th = 0.03$

Stage – 2 : Key-Point Localization

- Sub-stage-3 : Removal of false-positive key-points along Edges
- Core idea : Along the edge, one of the principal curvatures (λ_1 or λ_2) will be high compared to others.
- Analogous to Harris Detector
- Steps : -
  - For each of the detected key-points in sub-stage -2 :
    - Compute Hessian matrix: $H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$
    - Compute : $Tr(H) = D_{xx} + D_{yy} = \lambda_1 + \lambda_2$
    - Compute : $Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \lambda_1\lambda_2$

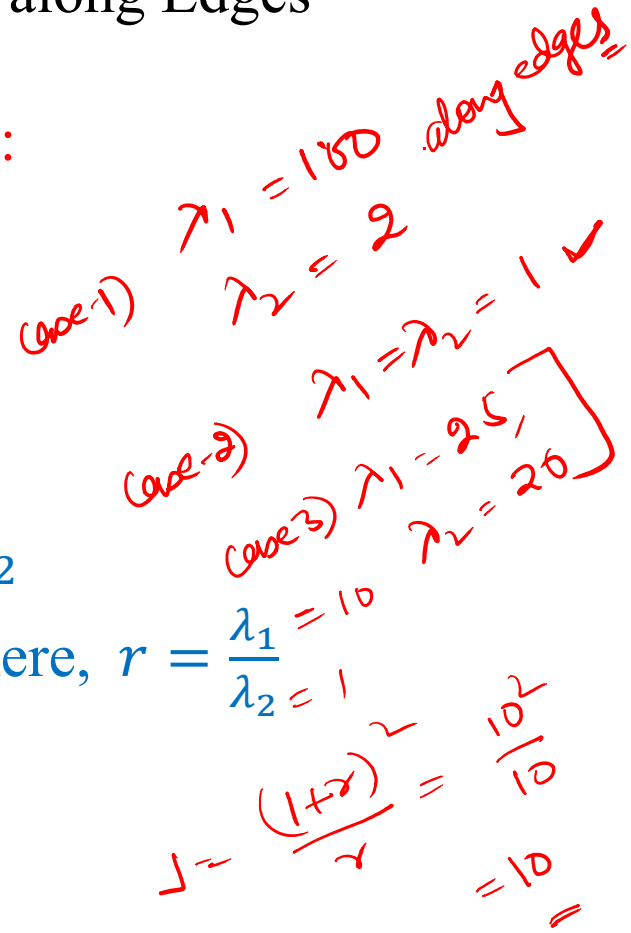
Stage – 2 : Key-Point Localization

- Sub-stage-3 : Removal of false-positive key-points along Edges

- For each of the detected key-points in sub-stage -2 :

  - Compute Hessian matrix: $H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$
  - Compute : $Tr(H) = D_{xx} + D_{yy} = \lambda_1 + \lambda_2$
  - Compute : $Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \lambda_1\lambda_2$
  - Compute : $J = \frac{[Tr(H)]^2}{Det(H)} = \frac{(\lambda_1+\lambda_2)^2}{\lambda_1\lambda_2} = \frac{(1+r)^2}{r}$; where, $r = \frac{\lambda_1}{\lambda_2}$

  - If $J > 10$, then retain key-point else discard it.



Stage – 2 : Key-Point Localization

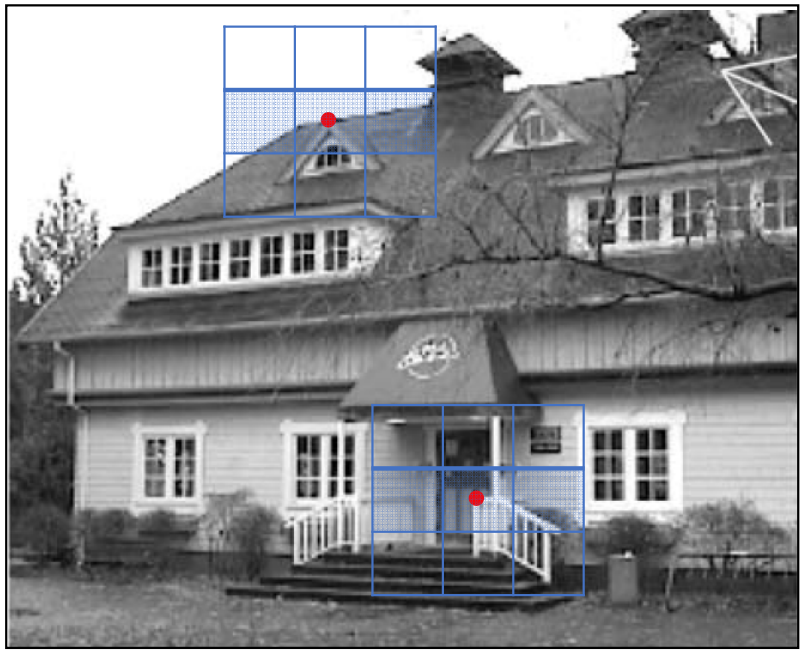
- Sub-stage-3 : Removal of false-positive key-points along Edges
  - If $J = \frac{[Tr(H)]^2}{Det(H)} = \frac{(\lambda_1+\lambda_2)^2}{\lambda_1\lambda_2} = \frac{(1+r)^2}{r} > 10$

(c) : 729 key-points remains with $Th = 0.03$

(d) : 536 key-points remains for $J > 10$

Stage – 3 : Orientation Assignment

- Sub-stage-3 : This stage assigns orientation to each of the detected "key point or interest point".
  - Find Gradient magnitude and Orientation at interest point -



$$\frac{\delta L}{\delta x} = L(x+1, y) - L(x-1, y)$$

$$\frac{\delta L}{\delta y} = L(x, y+1) - L(x, y-1)$$

$$M(x, y) = \sqrt{\left(\frac{\delta L}{\delta x}\right)^2 + \left(\frac{\delta L}{\delta y}\right)^2}$$

$$\theta(x, y) = \tan^{-1}\left[\left(\frac{\delta L}{\delta y}\right) / \left(\frac{\delta L}{\delta x}\right)\right]$$

Stage – 3 : Orientation Assignment

- Sub-stage-3 : This stage assigns orientation to each of the detected "key point or interest point".

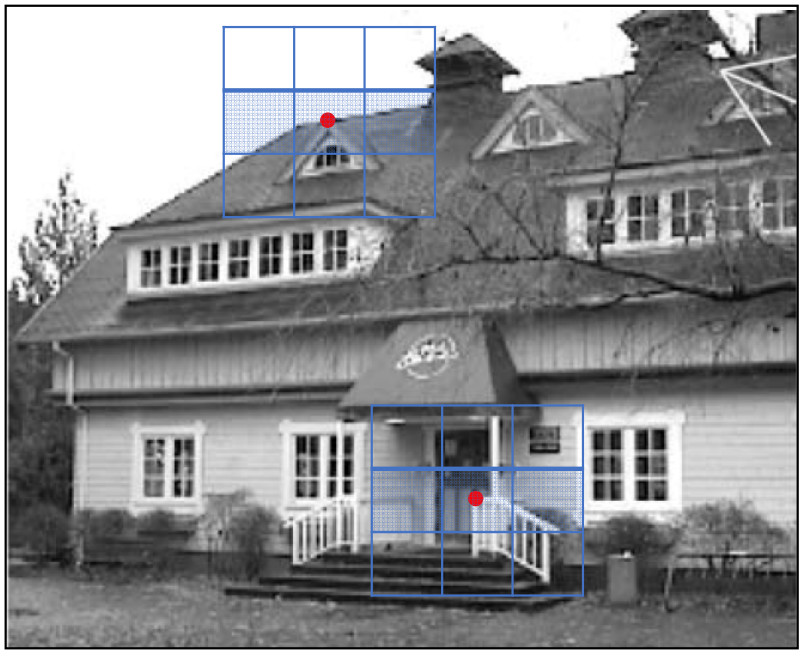


Fig-1 : Localized Key-points

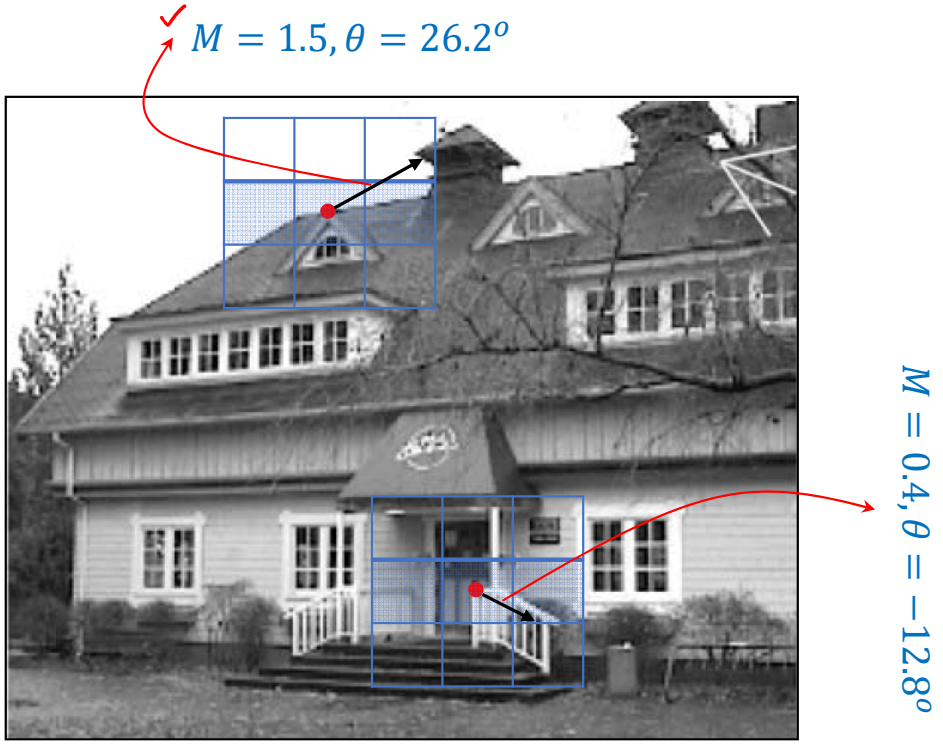


Fig-2 : Gradient magnitude and direction at key-points. The direction of line-segment indicates orientation of interest point whereas the length indicates their magnitude.

Stage – 3 : Orientation Assignment

- Sub-stage-3 : Dominant orientation at key-point

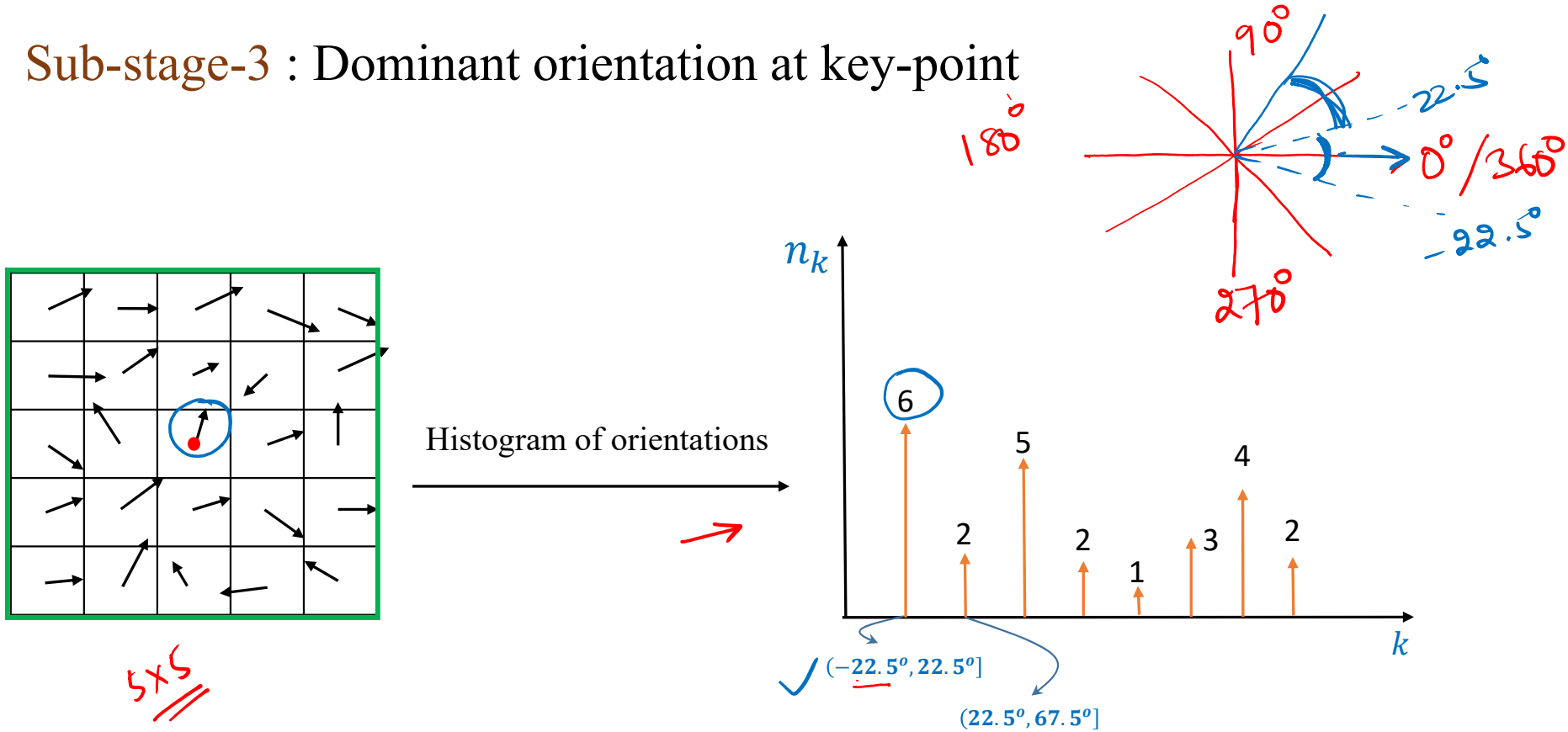


Fig-1 : Histogram of gradients used to calculate dominant direction at a key-point.

Stage – 3 : Orientation Assignment

- Sub-stage-3 : Dominant orientations at each of the selected key-points .

Fig-2 : **536 key-point**s . The direction of each of the line-segments indicates orientation of interest point whereas the length indicates their magnitude.

Stage – 4 : Key-point Descriptor

- Sub-stage-4 : SIFT descriptor is constituted by using gradient orientation of the interest points.
- For each key-point – the steps are as follows:

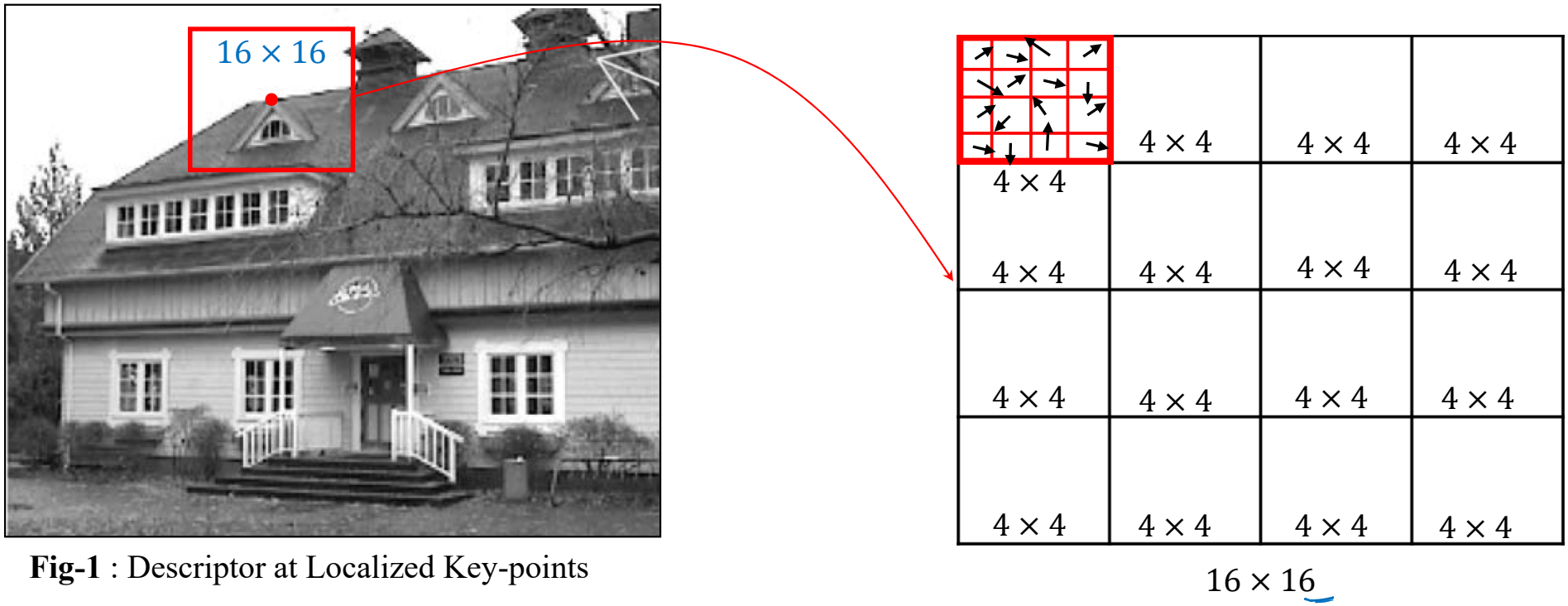
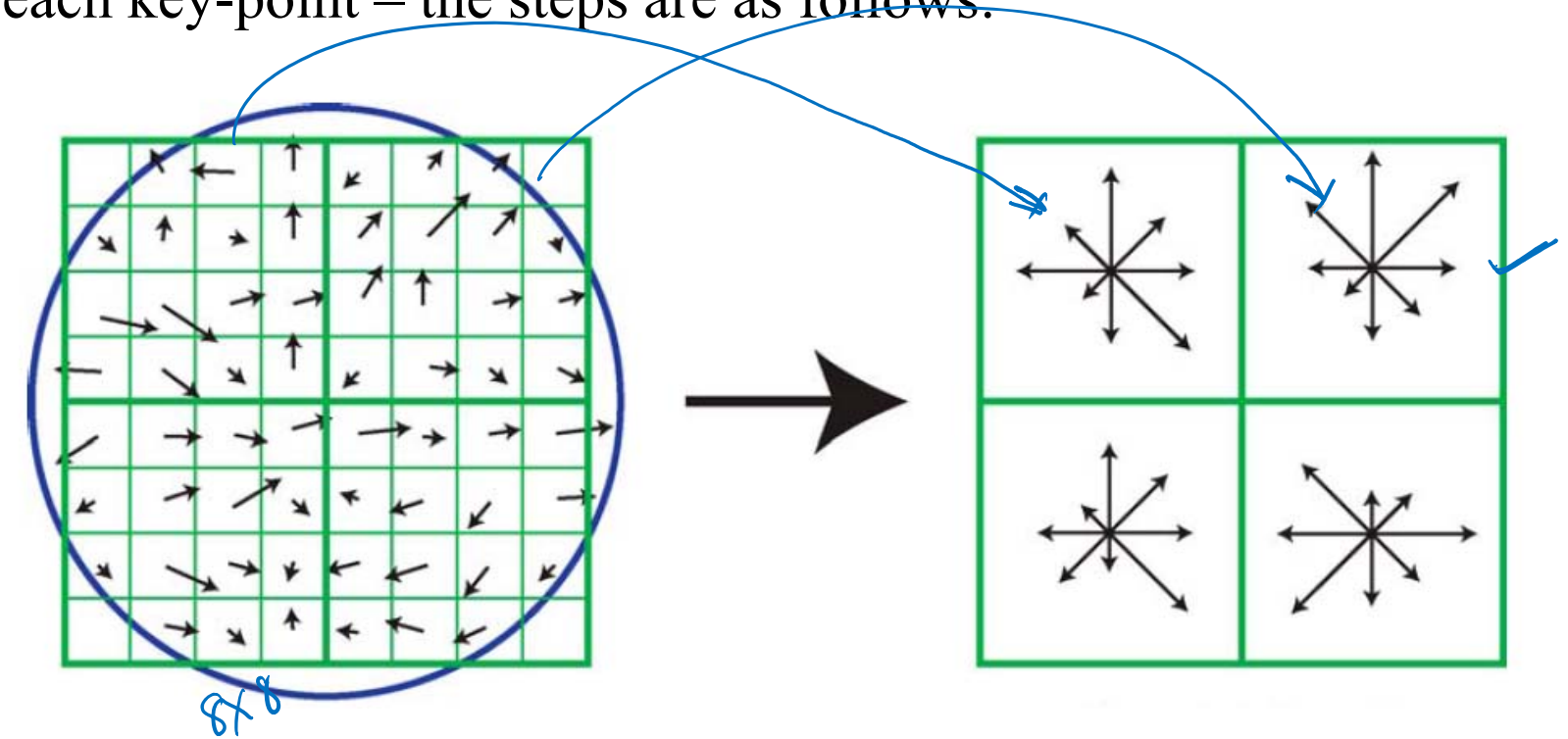


Fig-1 : Descriptor at Localized Key-points

Stage – 4 : Key-point Descriptor

- Sub-stage-4 : SIFT descriptor is constituted by using gradient orientation of the interest points.
- For each key-point – the steps are as follows:



Example : 8 × 8 neighbour of an interest point and their relative orientations

Stage – 4 : Key-point Descriptor

- Sub-stage-4-b : 8 bin histogram is constructed for each 4×4 block in 16×16

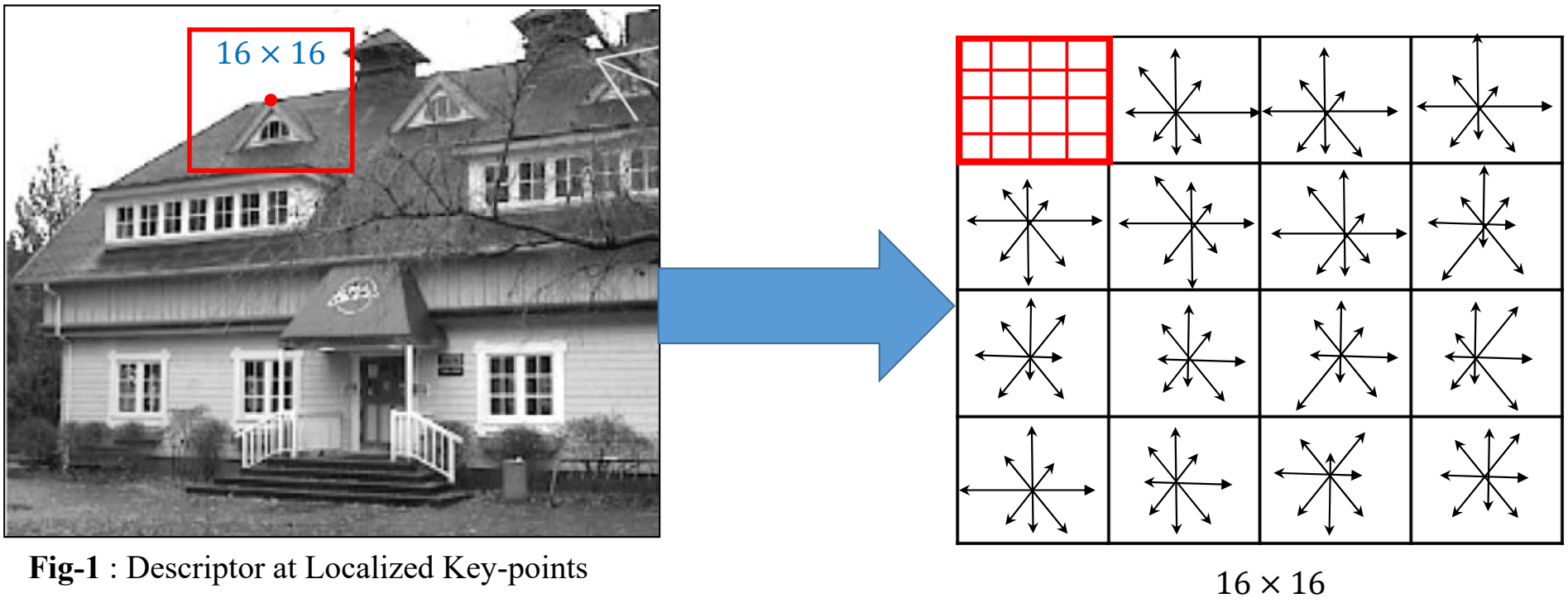


Fig-1 : Descriptor at Localized Key-points

Stage – 4 : Key-point Descriptor

- Sub-stage-4 : SIFT descriptor is constituted by using gradient orientation of the interest points.

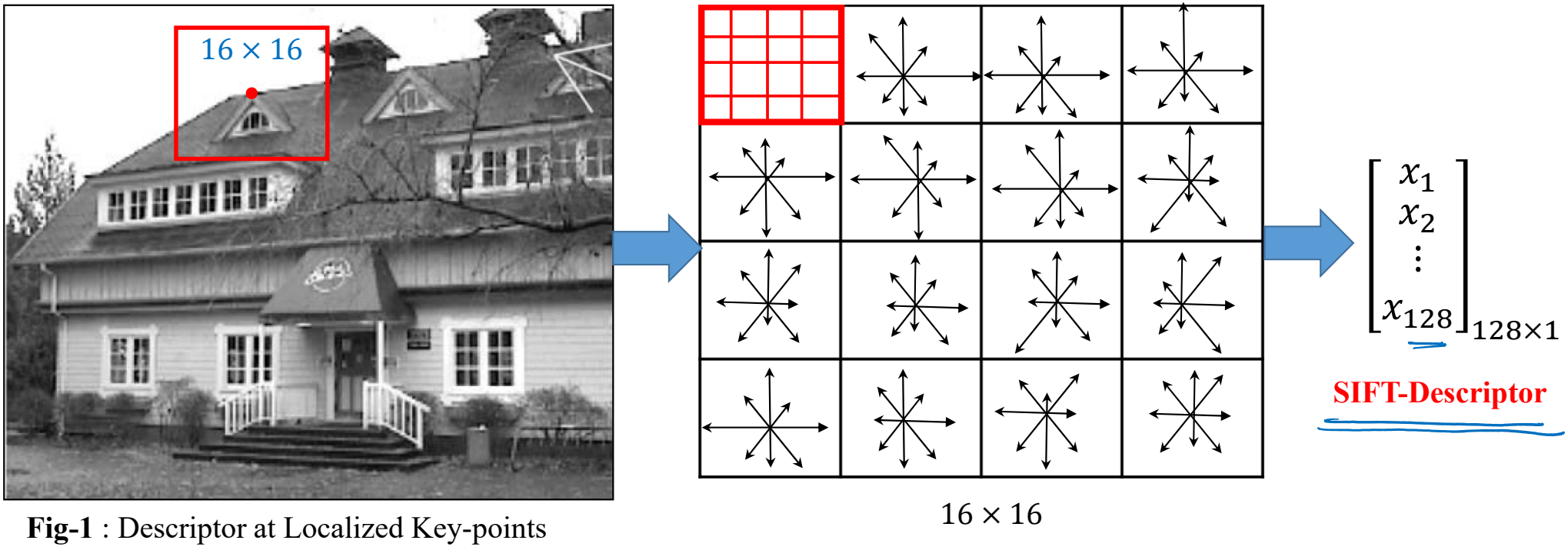
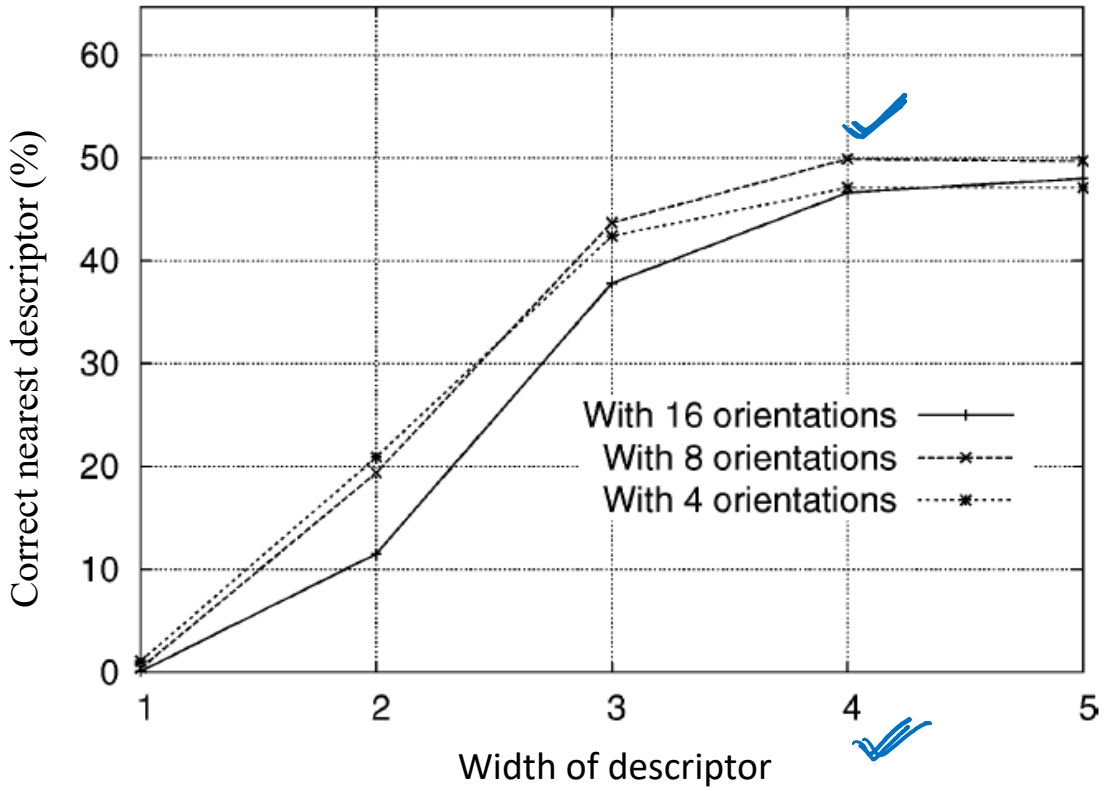


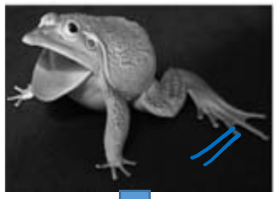
Fig-1 : Descriptor at Localized Key-points

Stage – 4 : Key-point Descriptor

- Sub-stage-4 : Width of descriptor

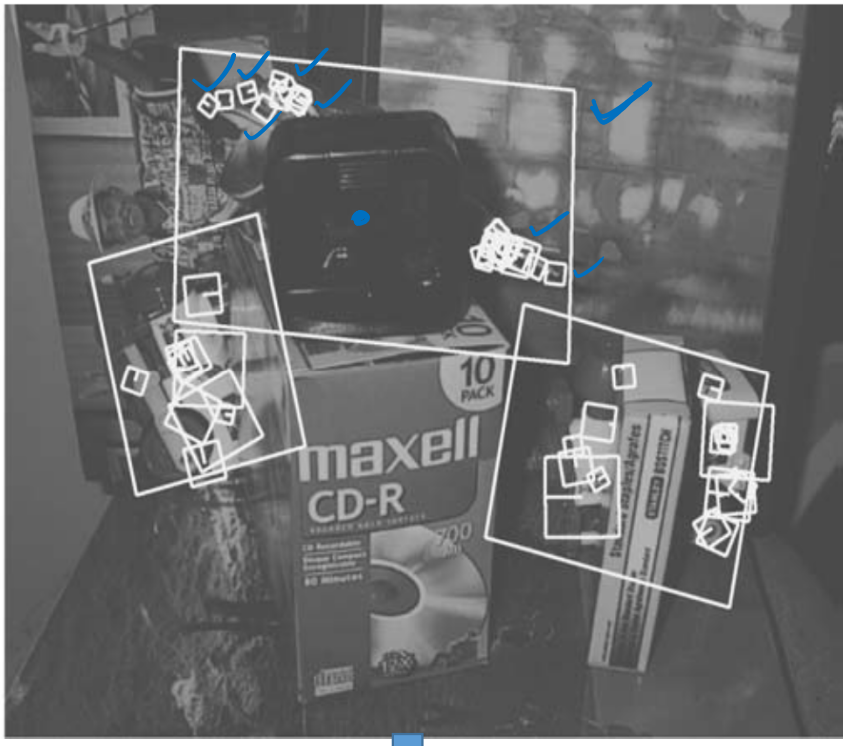


Matching using SIFT Descriptor : Object Recognition



Training objects

Query image with objects under complex environment



Query object found

Matching using SIFT Descriptor : Image Correspondence

View-1

View-2

Matching using SIFT Descriptor :

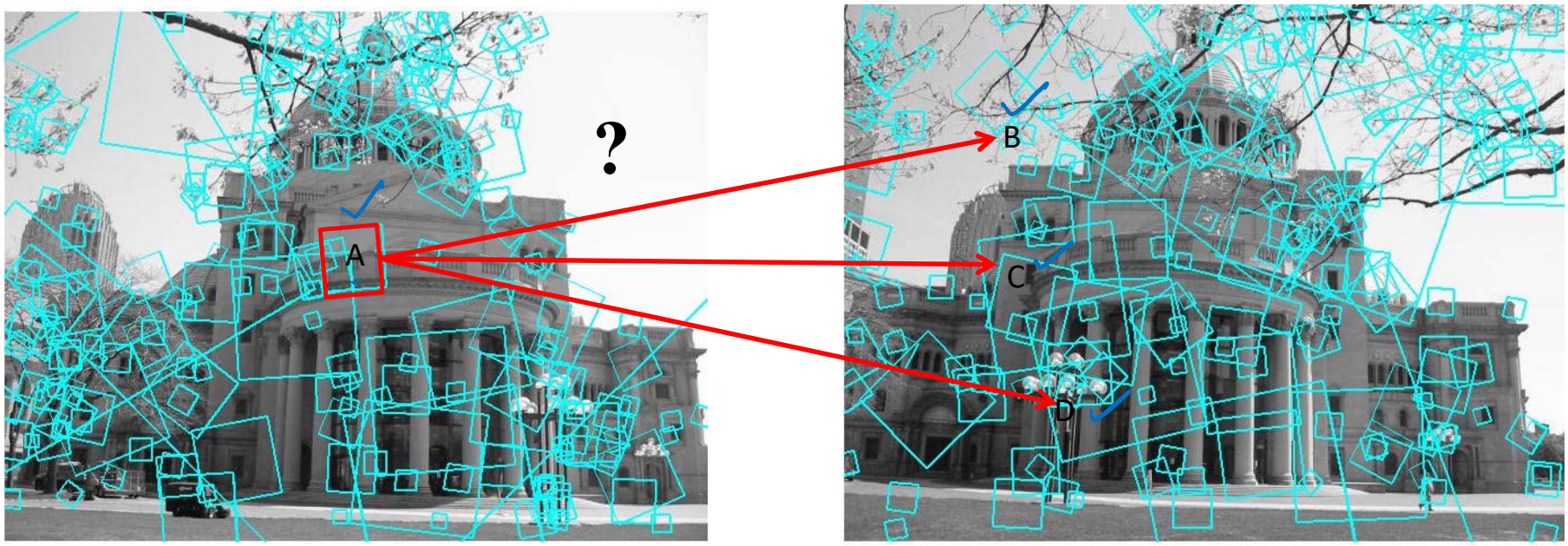


Image 1 Image 2

- Compute SIFT descriptors at “A” “B”, “C”, “D”, and so on. Find the patches that give the lowest SSD

Slide credit: Kristen Grauman

Resolving Ambiguous Matches

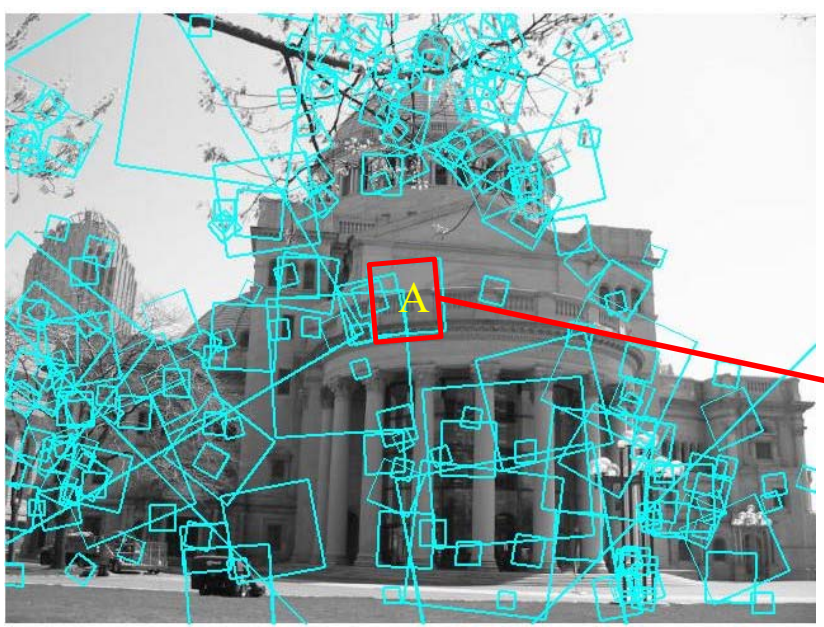


Image 1

?

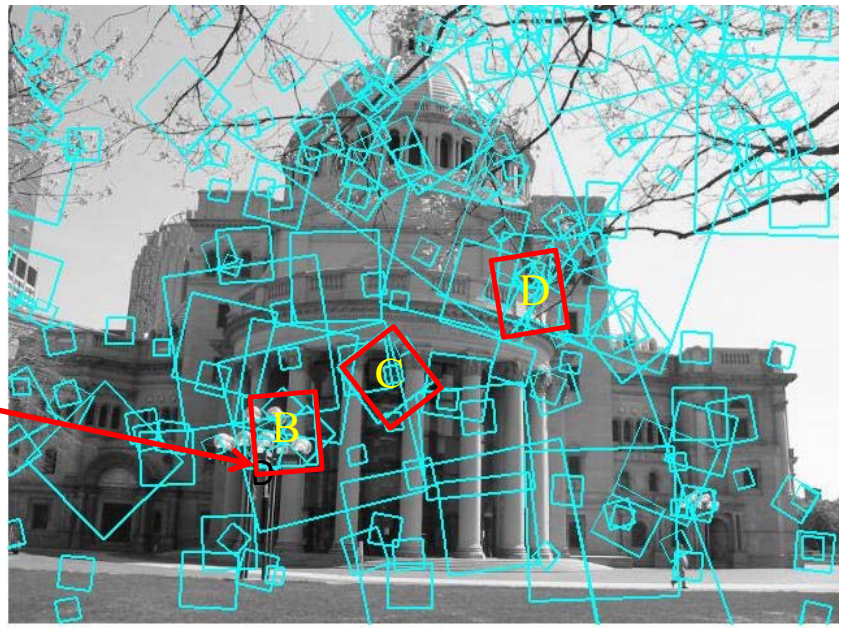


Image 2

$SSD(A, B) = 1.9$ $SSD(A, C) = 1.7$ $SSD(A, D) = 2.1$ ... $SSD(A, X) = 2.1$

Resolving Ambiguous Matches

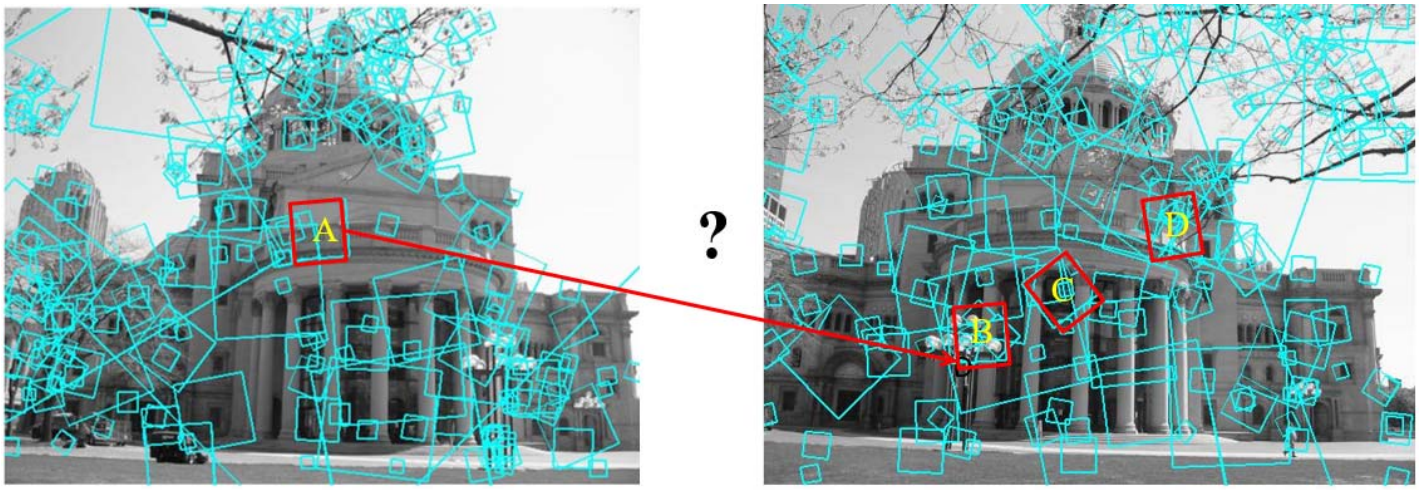


Image 1 Image 2

- For robust match : compute ratio between best match to the 2nd best match

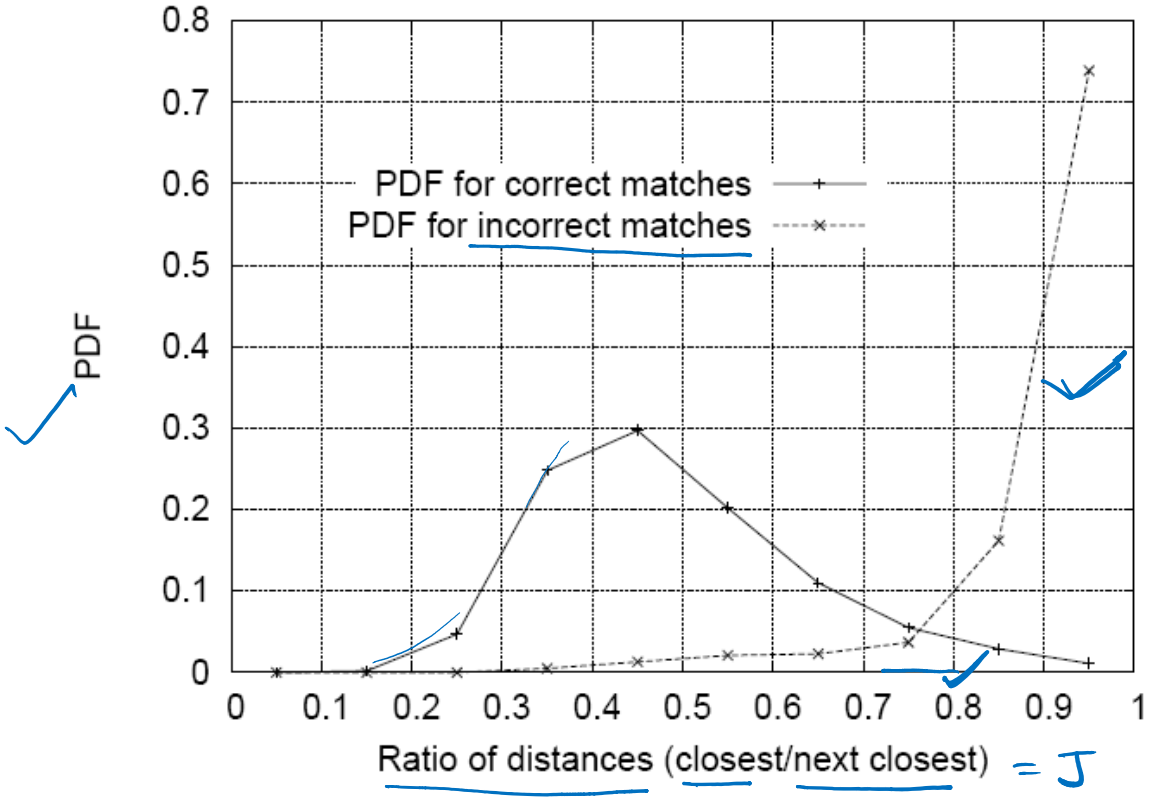
$$J = \frac{\text{Best match distance}}{\text{Second best match distance}}$$

$\frac{SSD(A,D)}{SSD(A,C)} = 0.9$

- If J is low (<0.8), then first match is good else match is ambiguous.
- J > 0.8, not to consider such matches.

Resolving Ambiguous Matches

- If J is low (<0.8), then first match is good else match is ambiguous.
- J > 0.8, not to consider such matches.



Automatic Image Stitching (Mosaicing)

Matthew Brown

http://matthewalunbrown.com/autostitch/autostitch.html

Acknowledgement!

Lowe, David G. "Distinctive image features from scale-invariant keypoints." *International journal of computer vision* 60.2 (2004): 91-110.

Reference

- Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010 (online draft),
- Mubarak Shah, "Fundamentals of Computer Vision" (Online available)
- Ian Goodfellow, Yoshua Bengio and Aaron Courville, "Deep Learning" (Online available)

Acknowledgement!

Sources for this lecture include materials from works by Szeliski, Abhijit Mahalanobis, Sedat Ozer, Ulas Bagci, Mubarak Shah, Antonio Torralba, D. Hoiem, Justin Liang, and others. References are given for the source image contents.

Queries!