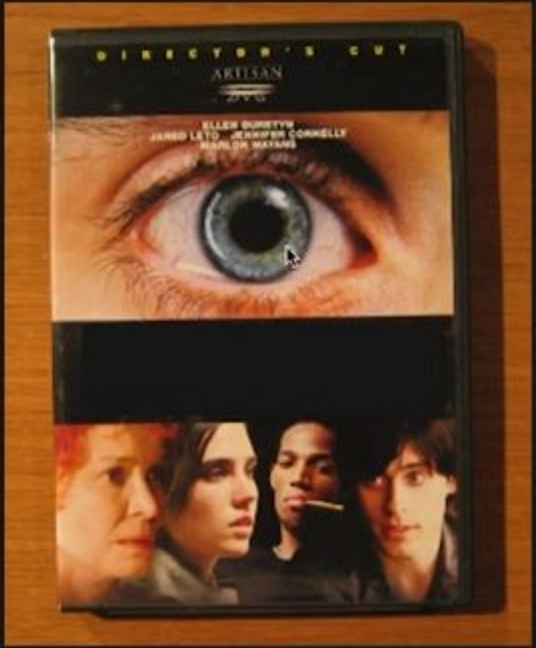
EE655: Computer Vision & Deep Learning

Lecture 06

Koteswar Rao Jerripothula, PhD
Department of Electrical Engineering
IIT Kanpur

CREDITS: Prof. Shree Nayar, Columbia University

How would you recognize the following types of objects?



Template

Rich 2D image

Find and Match "Interesting Points or Features"

Scale Invariant Feature Transform (SIFT) and its use for image alignment and 2D object recognition.

Topics:

(1) What is an Interest Point?

(2) Detecting Blobs

(3) SIFT Detector

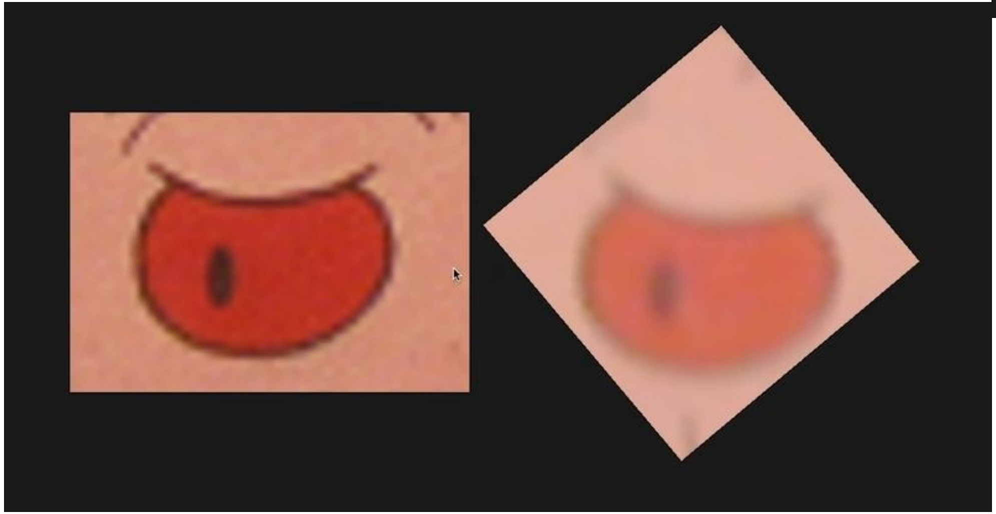
(4) SIFT Descriptor

Raw Images are Hard to Match

Different size, orientation, lighting, brightness, etc.

Removing Sources of Variation

Matching becomes easier if we can remove variations like size and orientation.

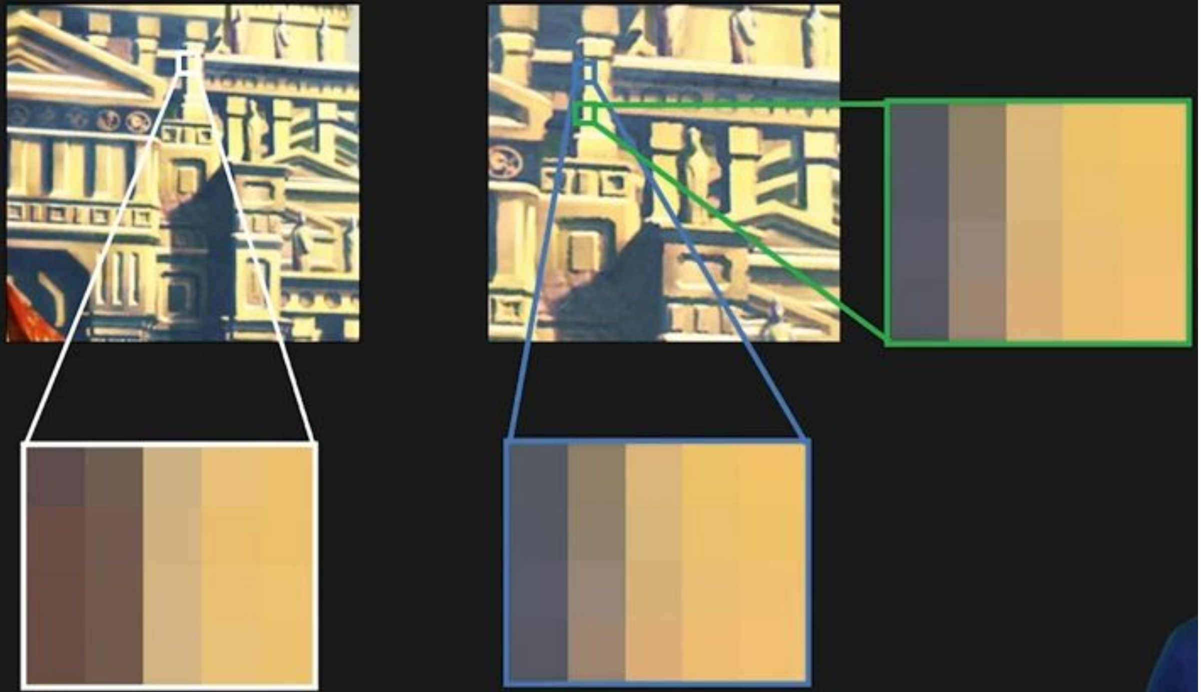


Some Patches are not "Interesting"

What is an Interesting Point/Feature?

- Has rich image content (brightness variation, color variation, etc.) within the local window
- Has well-defined representation (signature) for matching/comparing with other points
- Has a well-defined position in the image
- Should be invariant to image rotation and scaling
- Should be insensitive to lighting changes

Are Lines/Edges Interesting?



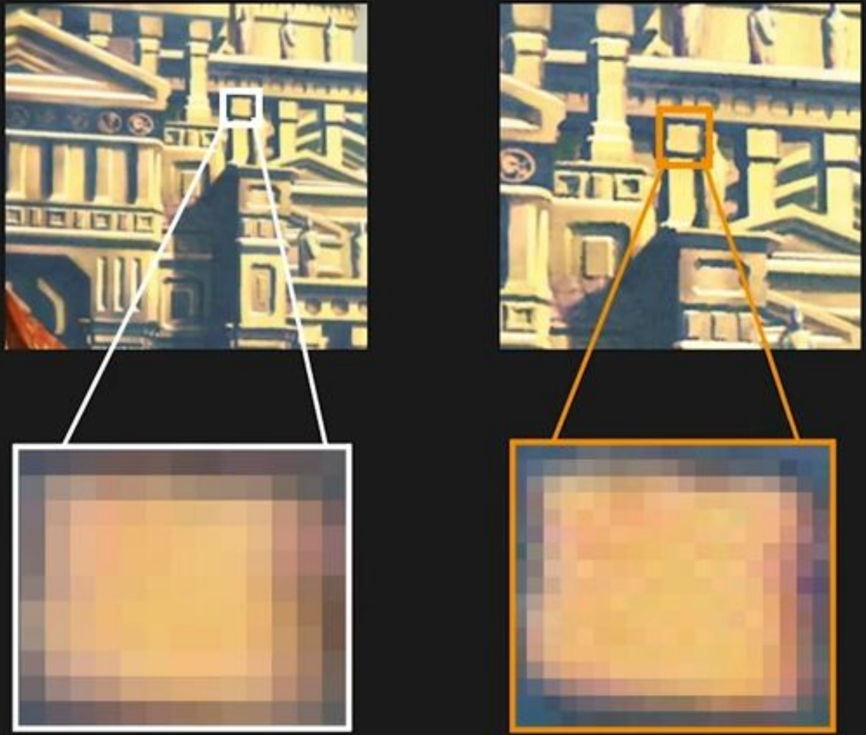
Cannot "Localize" an Edge

What about corners?

They are interesting, but:

- They don’t appear often enough
- Useful for simple applications
- Not unique enough

Are Blobs Interesting?



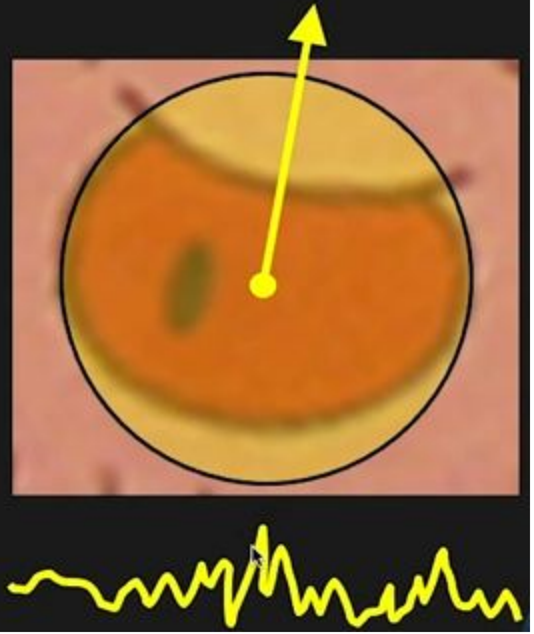
Yes! Blobs have fixed position and definite size.

Possible to define the appearance in a unique way

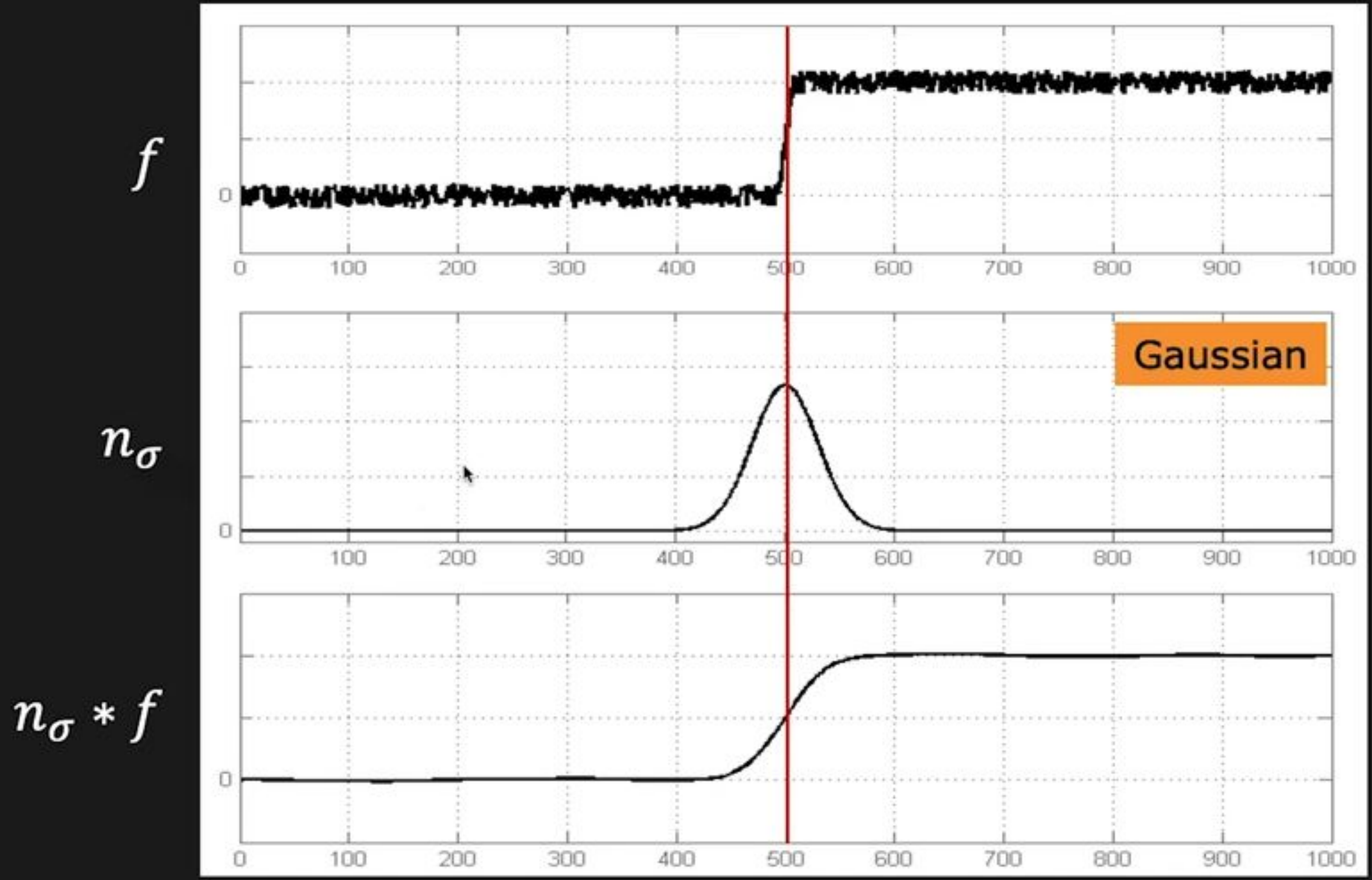
Blobs as Interest Points

For a Blob-like Feature to be useful, we need to:

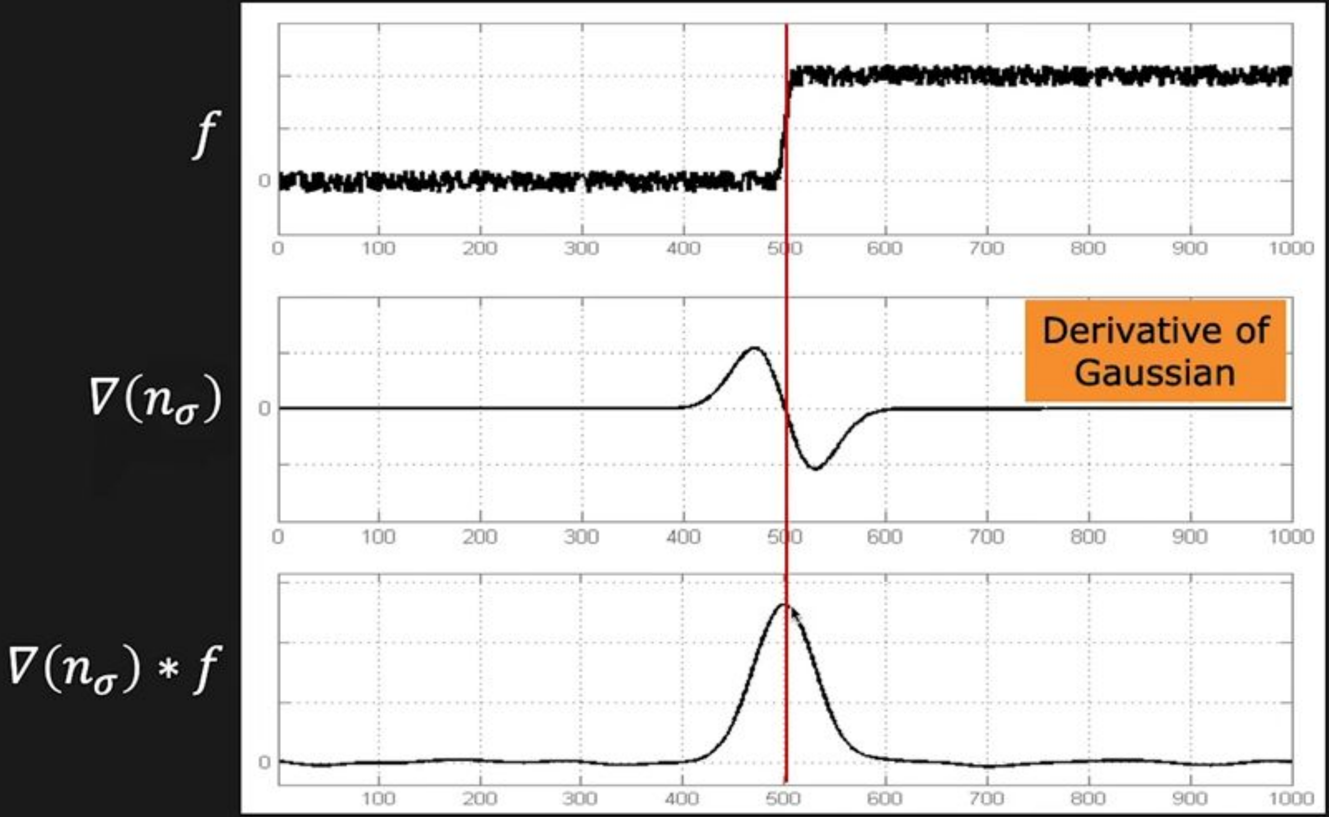
- Locate the blob
- Determine its size
- Determine its orientation
- Formulate a description or signature that is independent of size and orientation



Review: Gaussian Filter

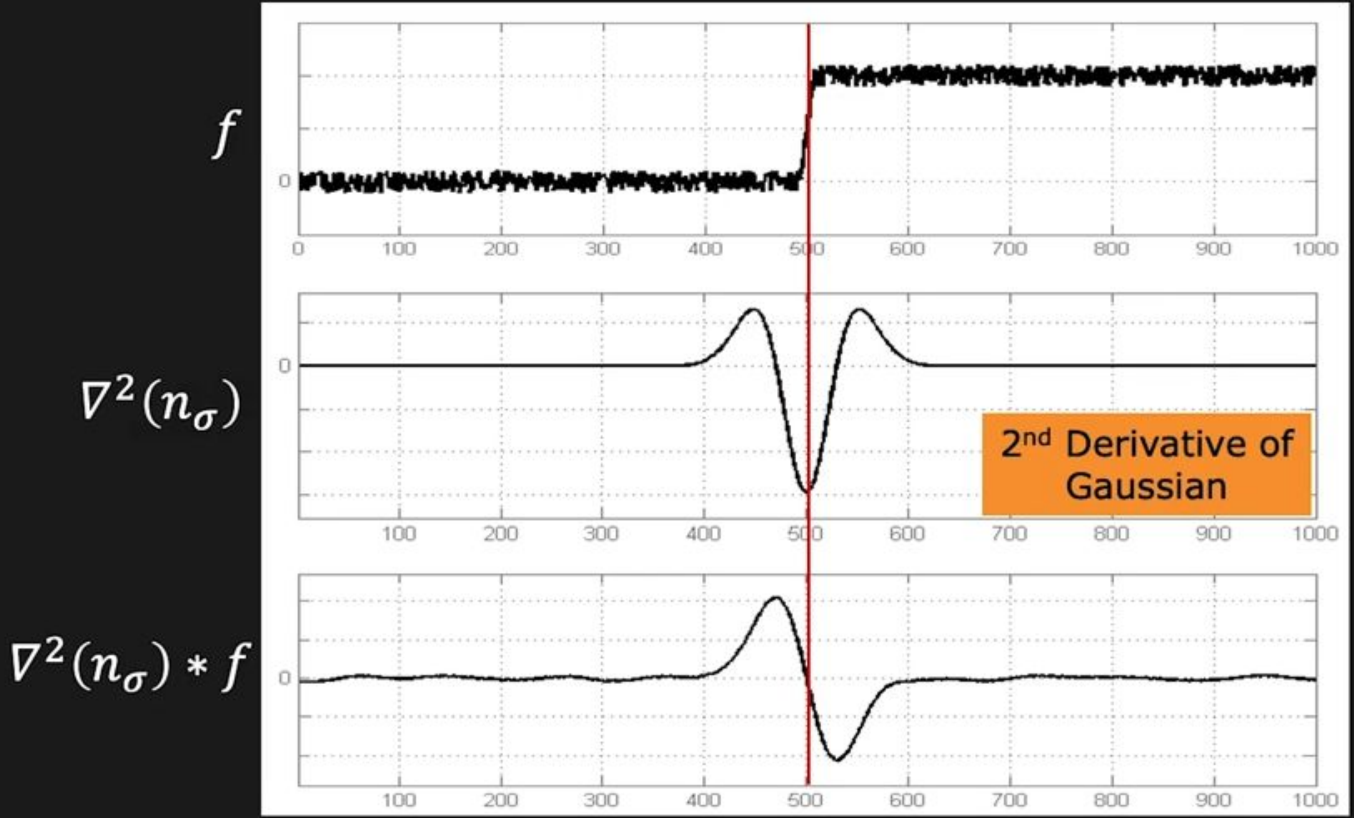


Review: Derivative of Gaussian



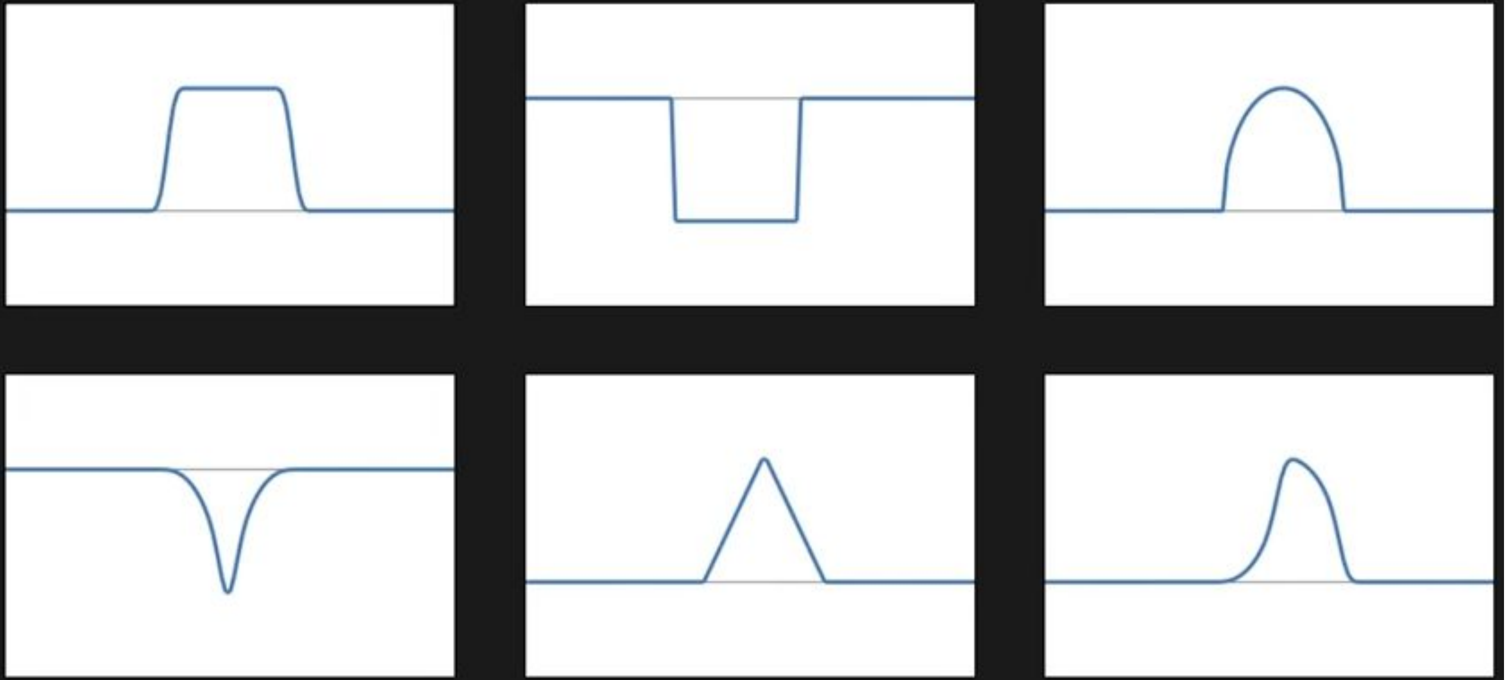
Extremum of Derivative of Gaussian denotes an Edge

Review: 2nd Derivative of Gaussian

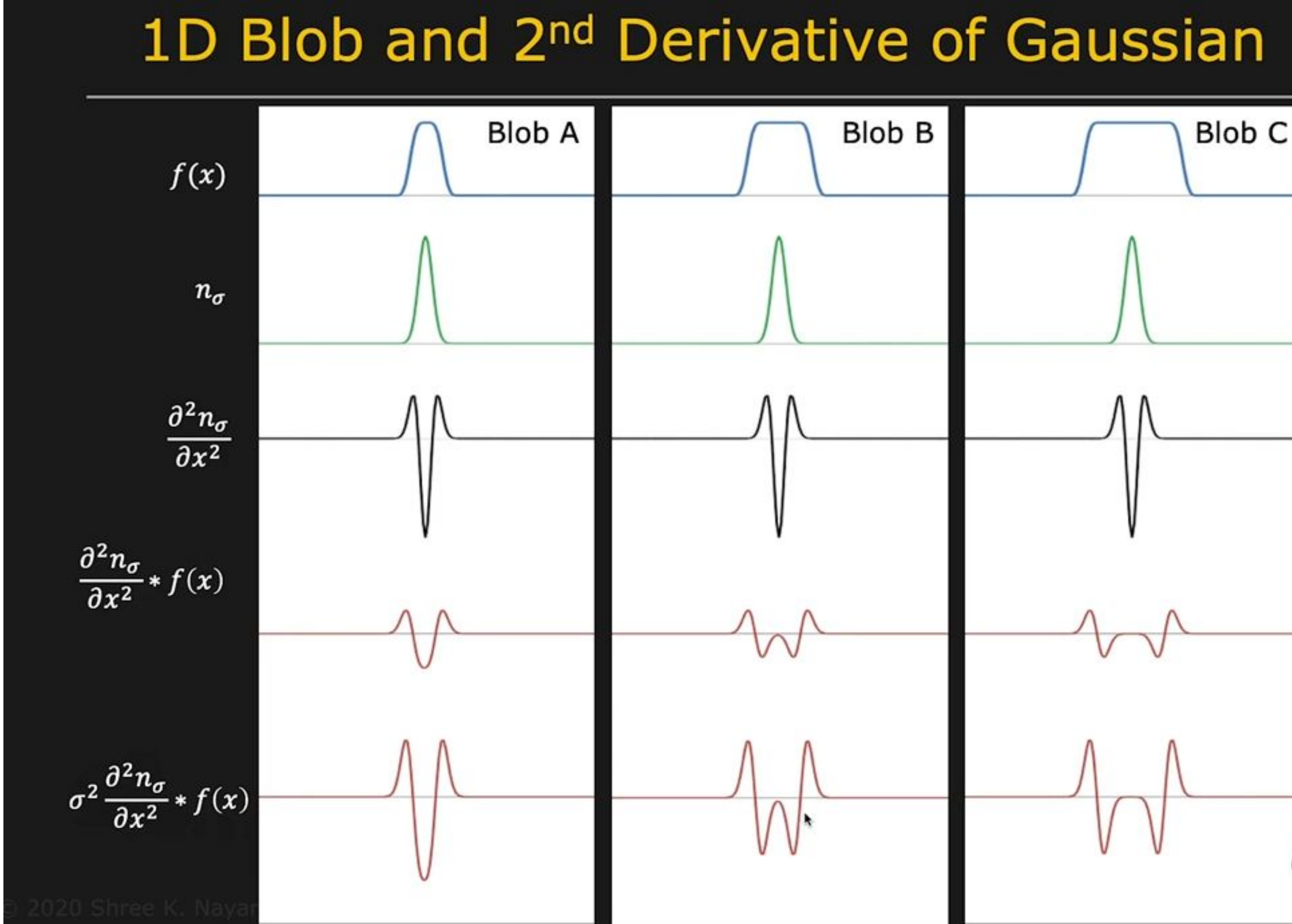


Zero Crossing in 2nd Derivative of Gaussian denote an Edge

1D Blobs



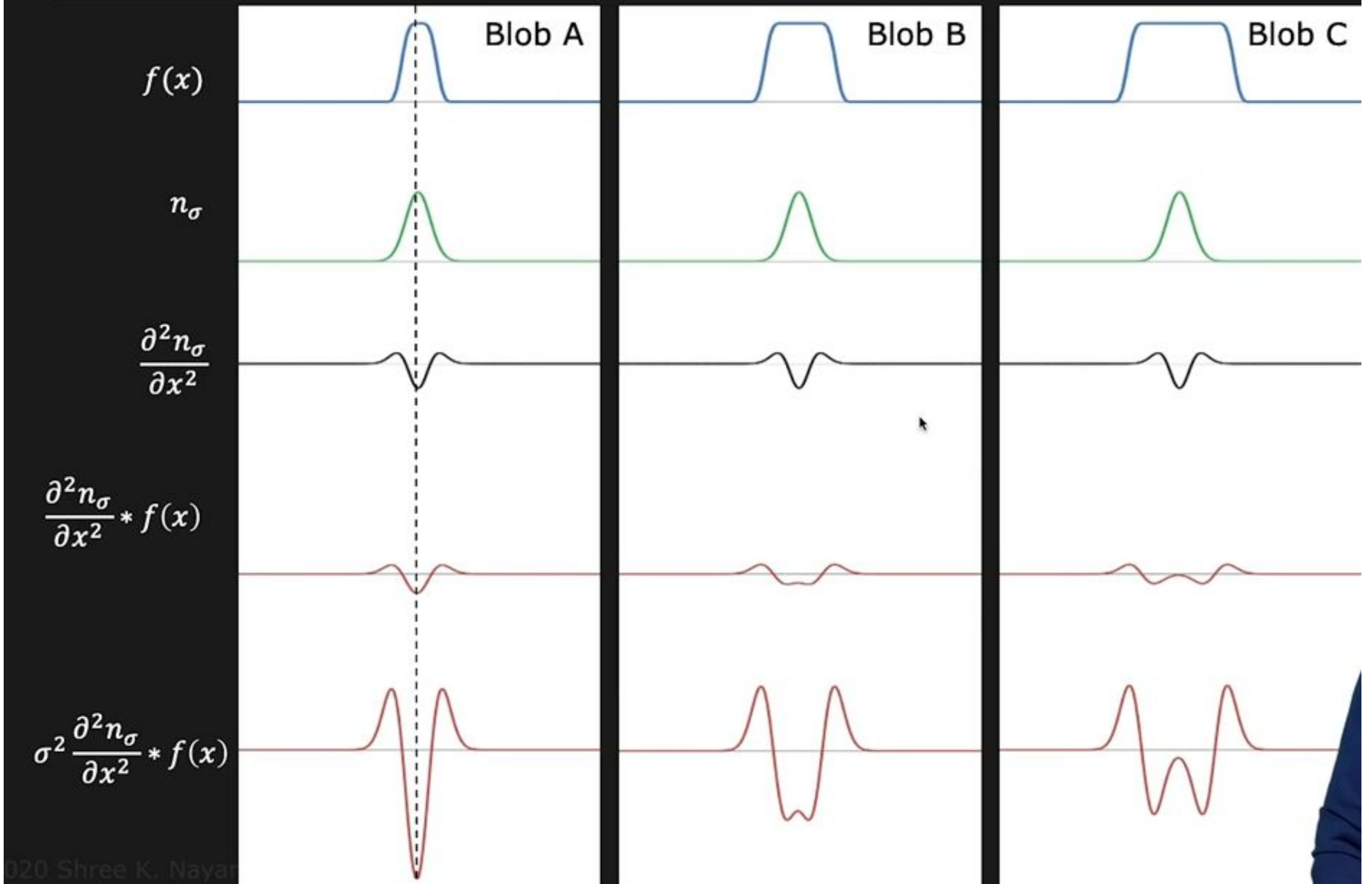
Examples of 1D Blob-like structures



Sigma Normalization

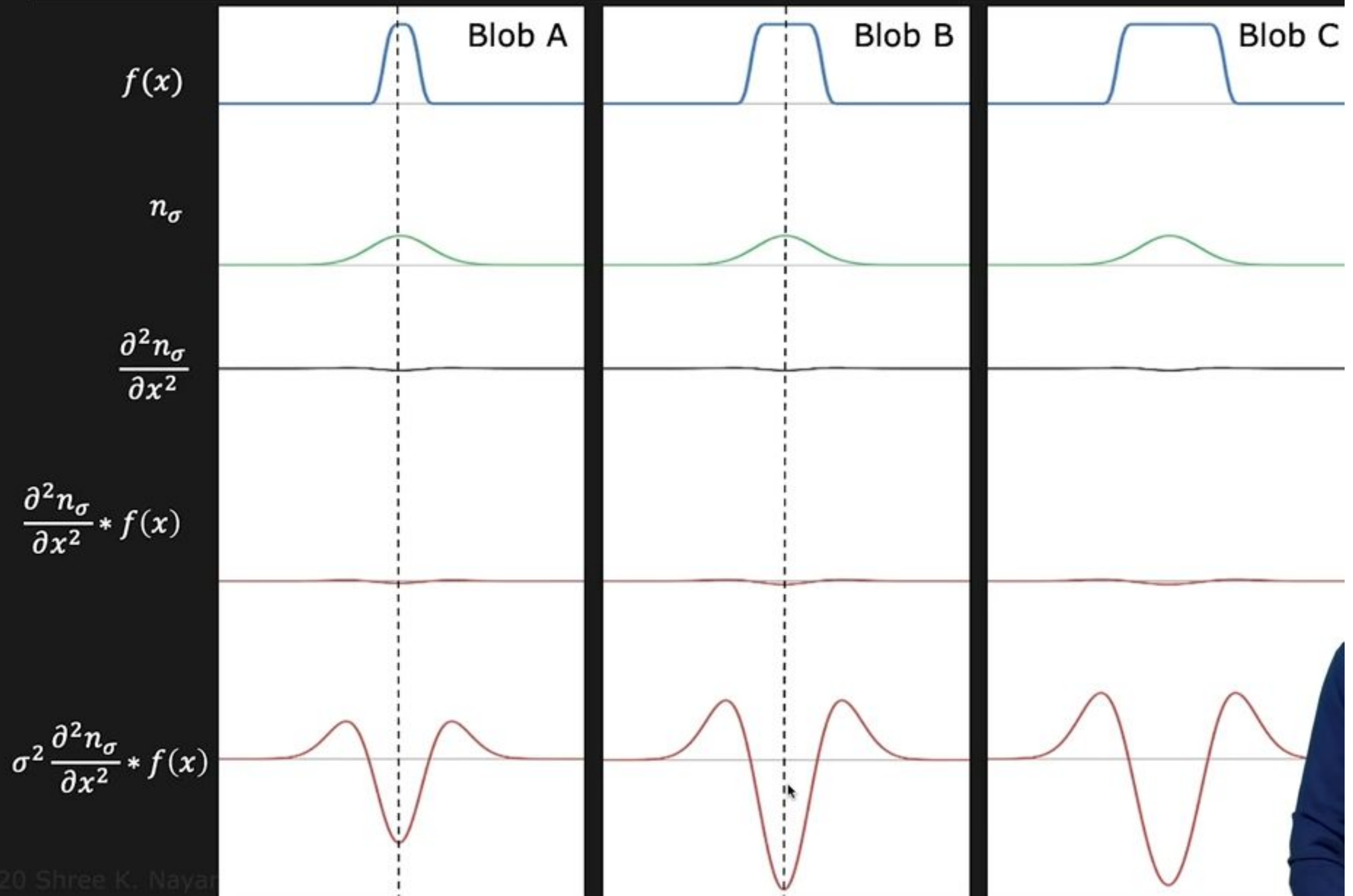
@SigmaA

1D Blob and 2nd Derivative of Gaussian

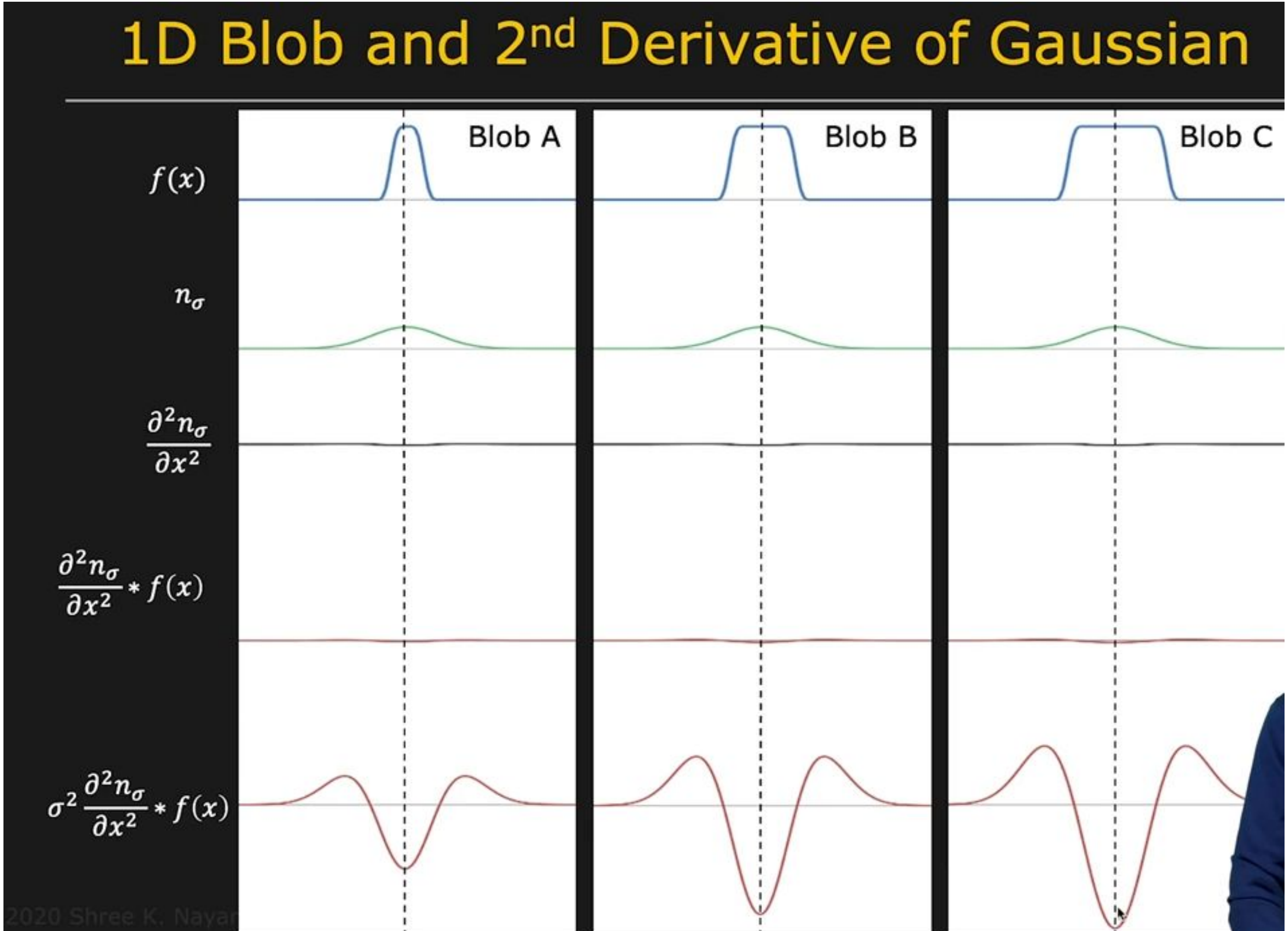


@SigmaB

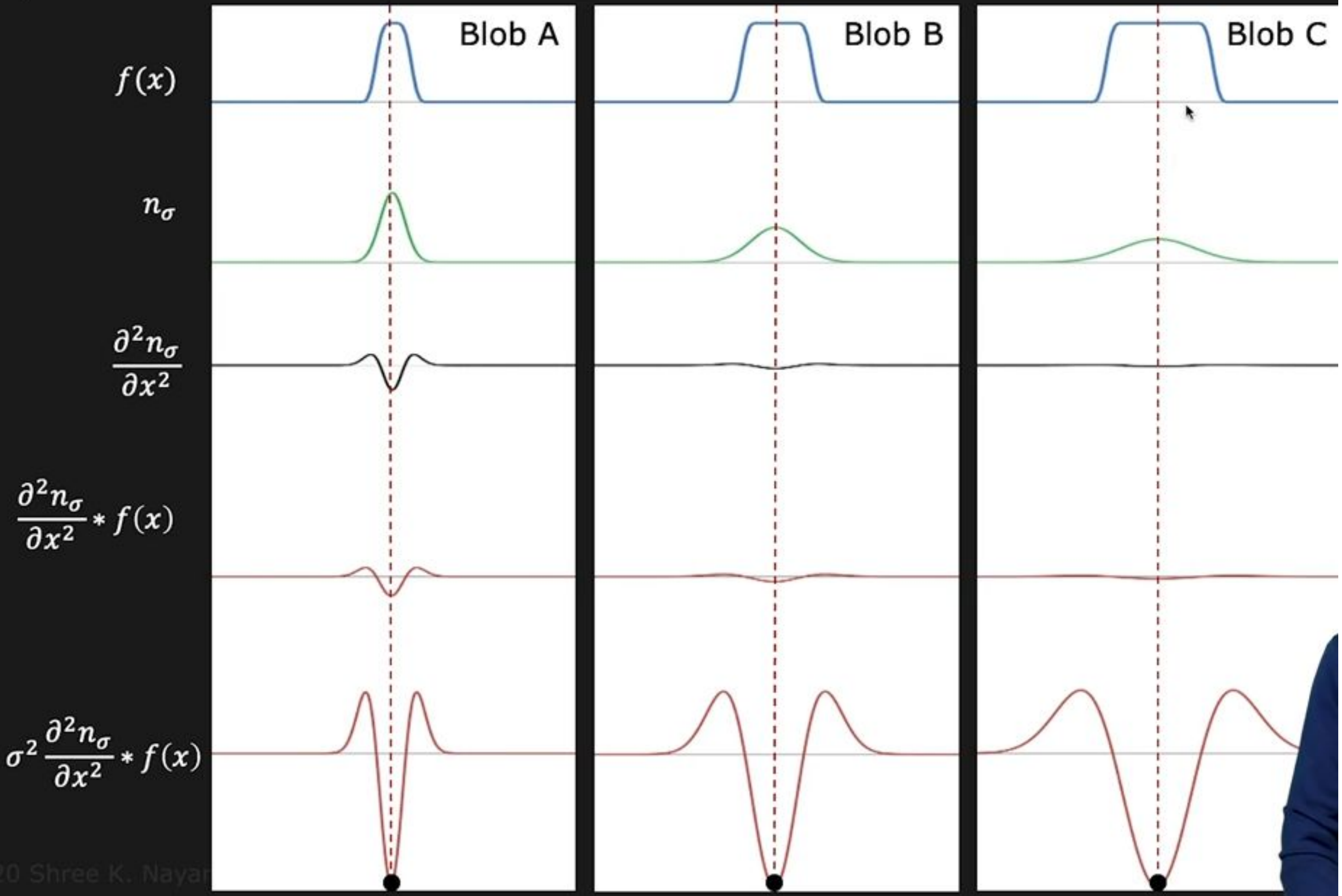
1D Blob and 2nd Derivative of Gaussian



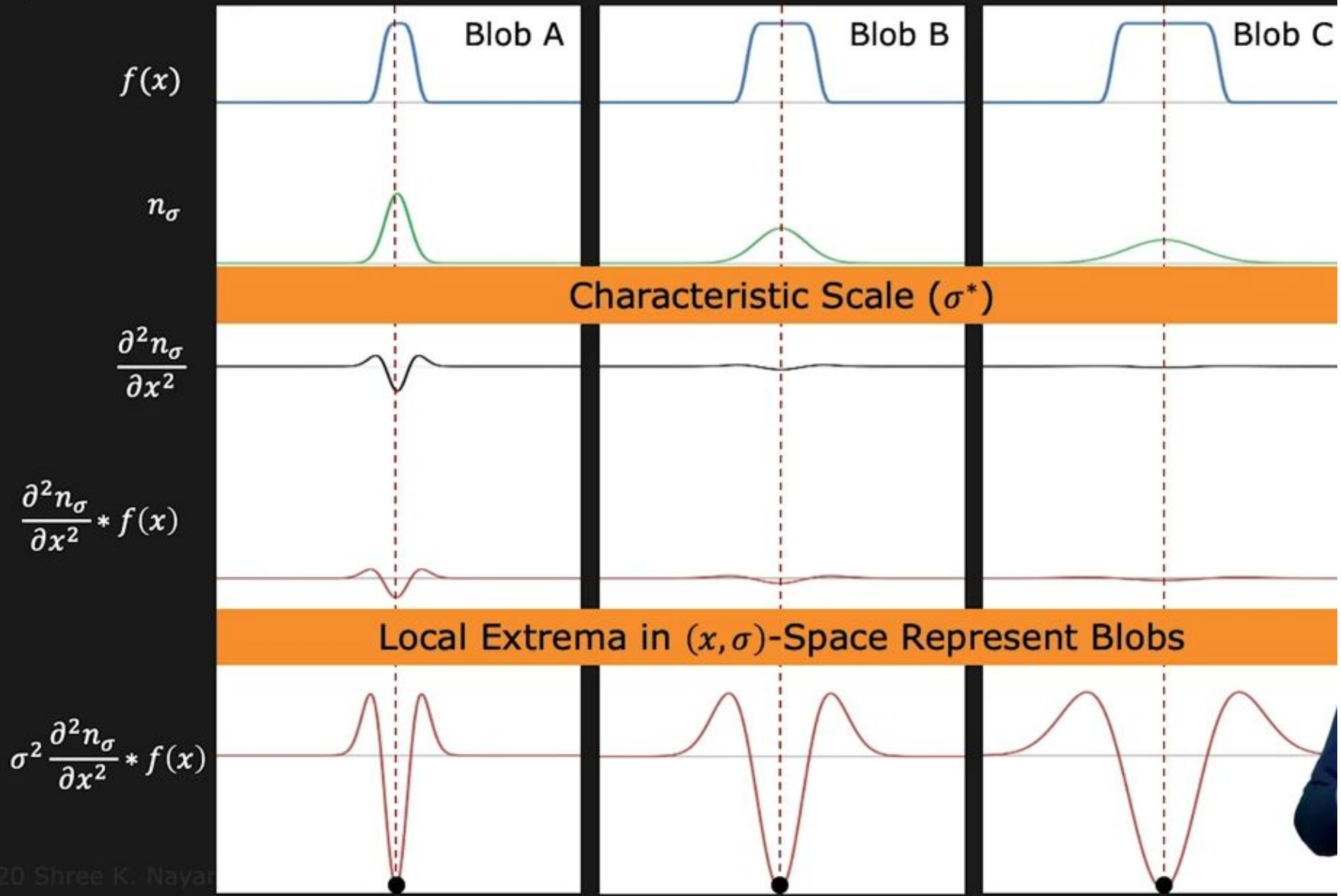
@SigmaC


1D Blob and 2nd Derivative of Gaussian
Blob A
Blob B
Blob C
$f(x)$
n_σ
$\frac{\partial^2 n_\sigma}{\partial x^2}$
$\frac{\partial^2 n_\sigma}{\partial x^2} * f(x)$
$\sigma^2 \frac{\partial^2 n_\sigma}{\partial x^2} * f(x)$
2020 Shree K. Nayar

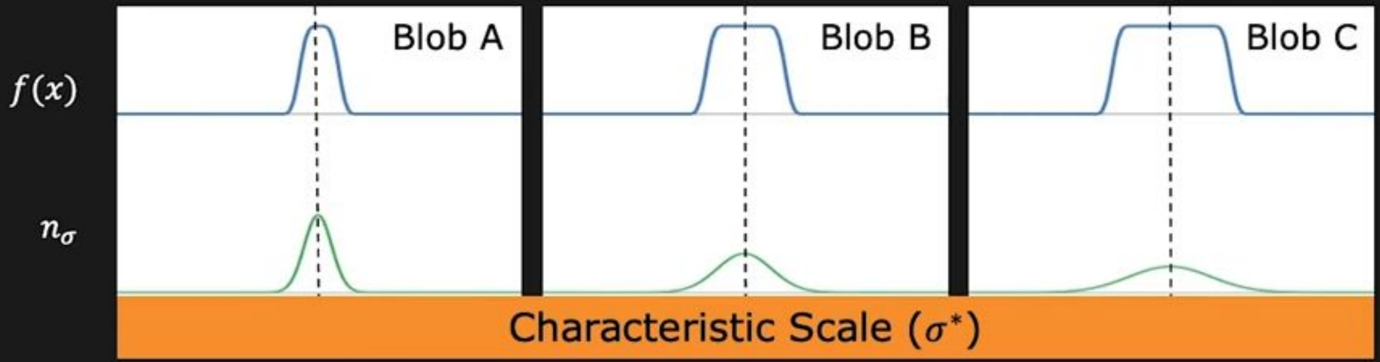
1D Blob and 2nd Derivative of Gaussian



1D Blob and 2nd Derivative of Gaussian



Characteristic Scale and Blob Size



$\sigma_A{}^* = \sigma_1$ $\sigma_B{}^* = 2\sigma_1$ $\sigma_C{}^* = 3\sigma_1$

Characteristic Scale: The σ at which σ-normalized 2nd derivative attains its extreme value.

Characteristic Scale $\propto$ Size of Blob

$$\frac{\text{Size of Blob A}}{\text{Size of Blob B}} = \frac{\sigma_A{}^*}{\sigma_B{}^*}; \quad \frac{\text{Size of Blob B}}{\text{Size of Blob C}} = \frac{\sigma_B{}^*}{\sigma_C{}^*}$$

1D Blob Detection Summary

Given: 1D signal $f(x)$

Compute: $\sigma^2 \frac{\partial^2 n_\sigma}{\partial x^2} * f(x)$ at many scales $(\sigma_0, \sigma_1, \sigma_2, \ldots, \sigma_k)$.

Find: $$\boxed{(x^*, \sigma^*) = \underset{(x,\sigma)}{\arg\max} \left| \sigma^2 \frac{\partial^2 n_\sigma}{\partial x^2} * f(x) \right|}$$

x^*: Blob Position

σ^*: Characteristic Scale (Blob Size)

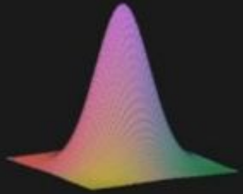
2D Blob Detector

Normalized Laplacian of Gaussian (NLoG) is used as the 2D equivalent for Blob Detection.

Laplacian

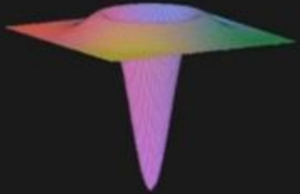
$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

Gaussian



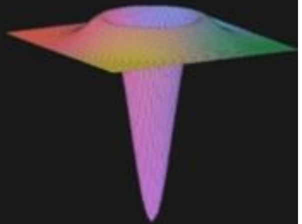
n_σ

LoG



$\nabla^2 n_\sigma$

NLoG



$\sigma^2 \nabla^2 n_\sigma$

Location of Blobs given by Local Extrema after applying Normalized Laplacian of Gaussian at many scales.

Scale-Space

$S(x,y,\sigma_0)$ $S(x,y,\sigma_1)$ $S(x,y,\sigma_2)$ $S(x,y,\sigma_3)$ ...

Increasing σ, Higher Scale, Lower Resolution

Scale Space: Stack created by filtering an image with Gaussians of different sigma (σ)

$$\boxed{S(x,y,\sigma) = n(x,y,\sigma) * I(x,y)}$$

Creating Scale-Space

$S(x, y, \sigma_0)$ $S(x, y, \sigma_1)$ $S(x, y, \sigma_2)$ $S(x, y, \sigma_3)$ ...

Increasing σ, Higher Scale, Lower Resolution

Selecting sigmas to generate the scale-space:

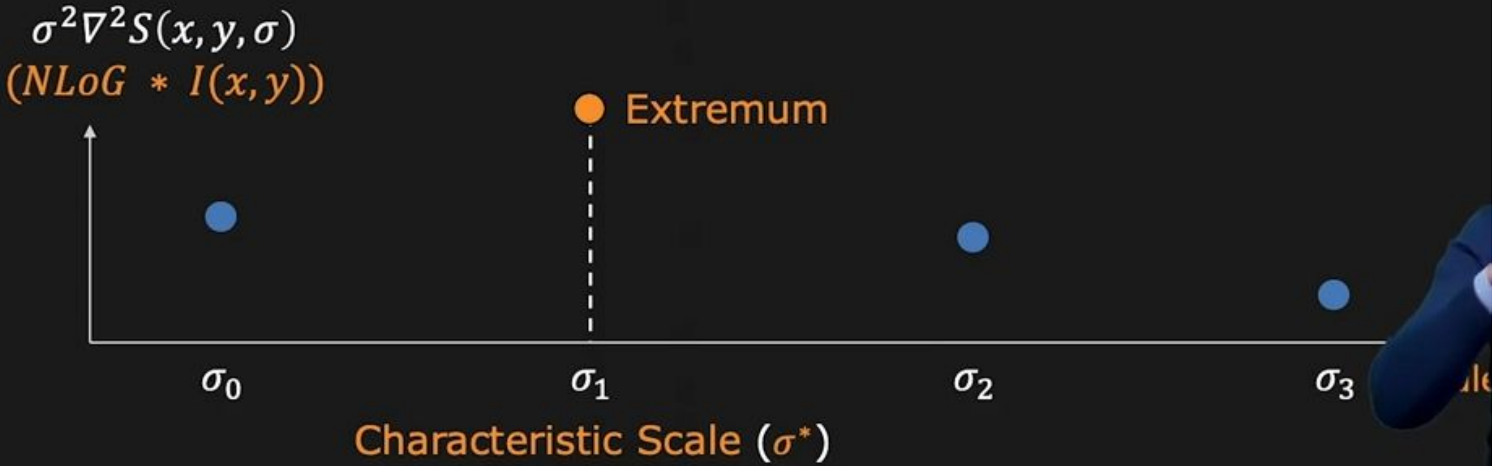
$$\boxed{\sigma_k = \sigma_0 s^k} \quad k = 0,1,2,3, \ldots$$

s: Constant multiplier

σ_0: Initial Scale

Blob Detection using Local Extrema

$S(x, y, \sigma_0)$ $S(x, y, \sigma_1)$ $S(x, y, \sigma_2)$ $S(x, y, \sigma_3)$ ...



Blob Detection using Local Extrema

$S(x, y, \sigma_0)$ $S(x, y, \sigma_1)$ $S(x, y, \sigma_2)$ $S(x, y, \sigma_3)$

$\sigma^2 \nabla^2 S(x, y, \sigma)$
$(NLoG * I(x, y))$

2D Blob Detection Summary

Given an image $I(x, y)$

Convolve the image using NLoG at many scales σ

Find:

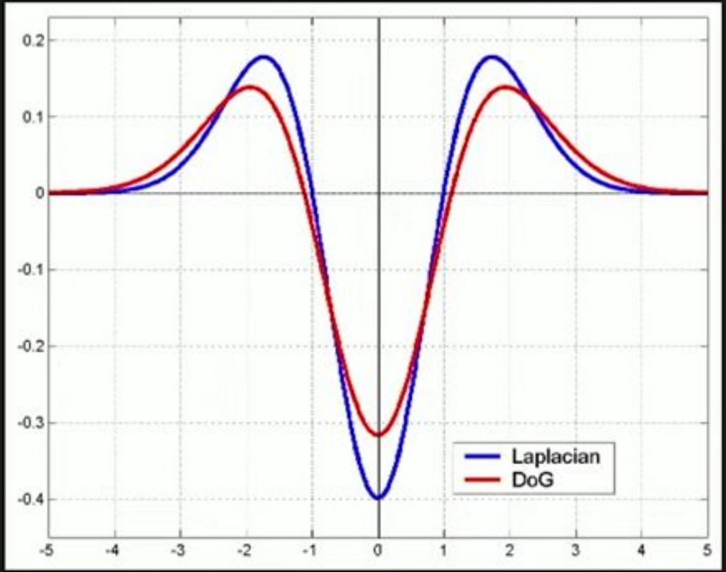
$$\boxed{(x^*, y^*, \sigma^*) = \underset{(x,y,\sigma)}{\arg\max} \left|\sigma^2 \nabla^2 n_\sigma * I(x, y)\right|}$$

(x^*, y^*): Position of the blob

σ^*: Size of the blob

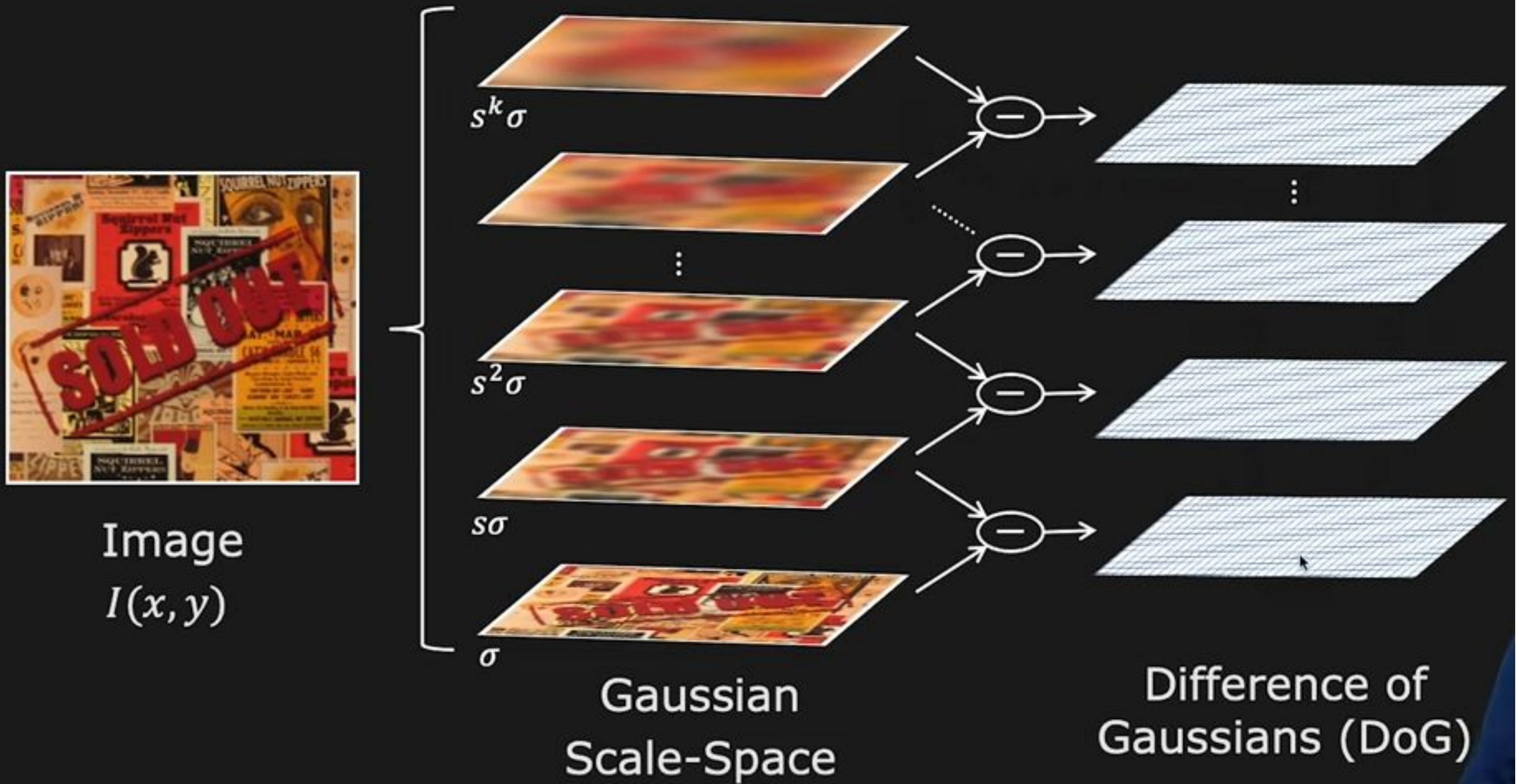
Fast NLoG Approximation: DoG

Difference of Gaussian (DoG) $= (n_{s\sigma} - n_{\sigma}) \approx (s-1)\underbrace{\sigma^2 \nabla^2 n_{\sigma}}_{\text{NLoG}}$

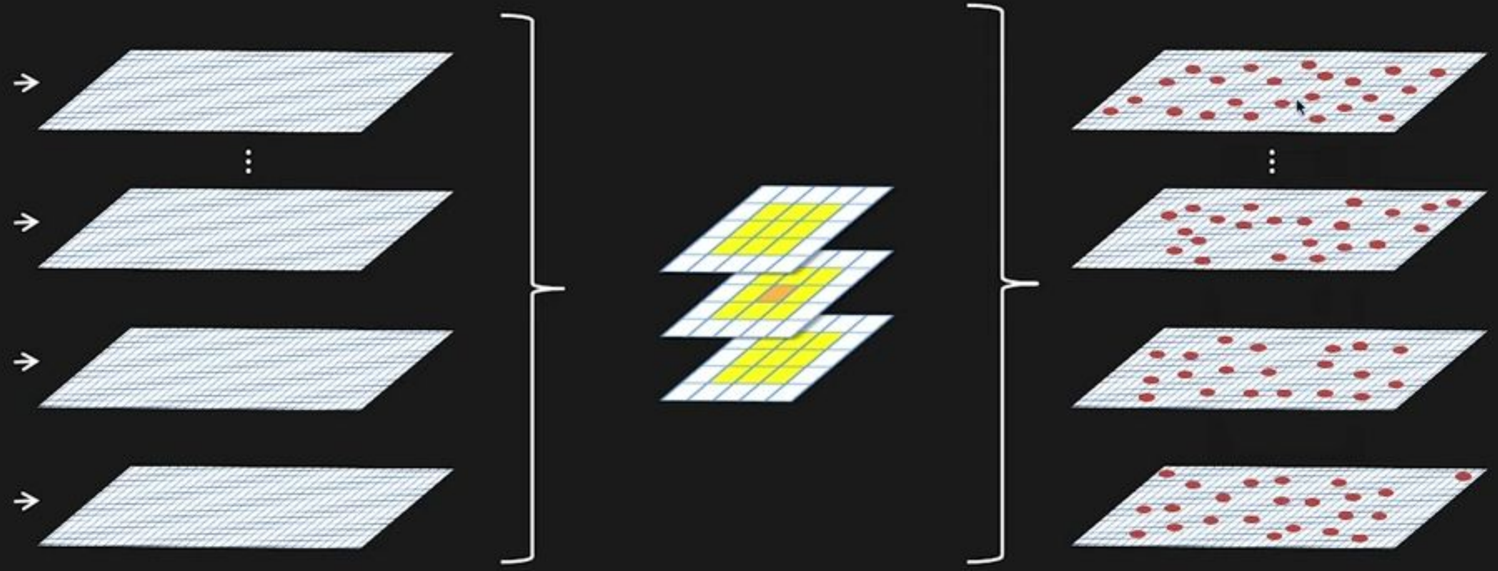


$$\boxed{\text{DoG} \approx (s-1)\ \text{NLoG}}$$

Extracting SIFT Interest Points



Extracting SIFT Interest Points

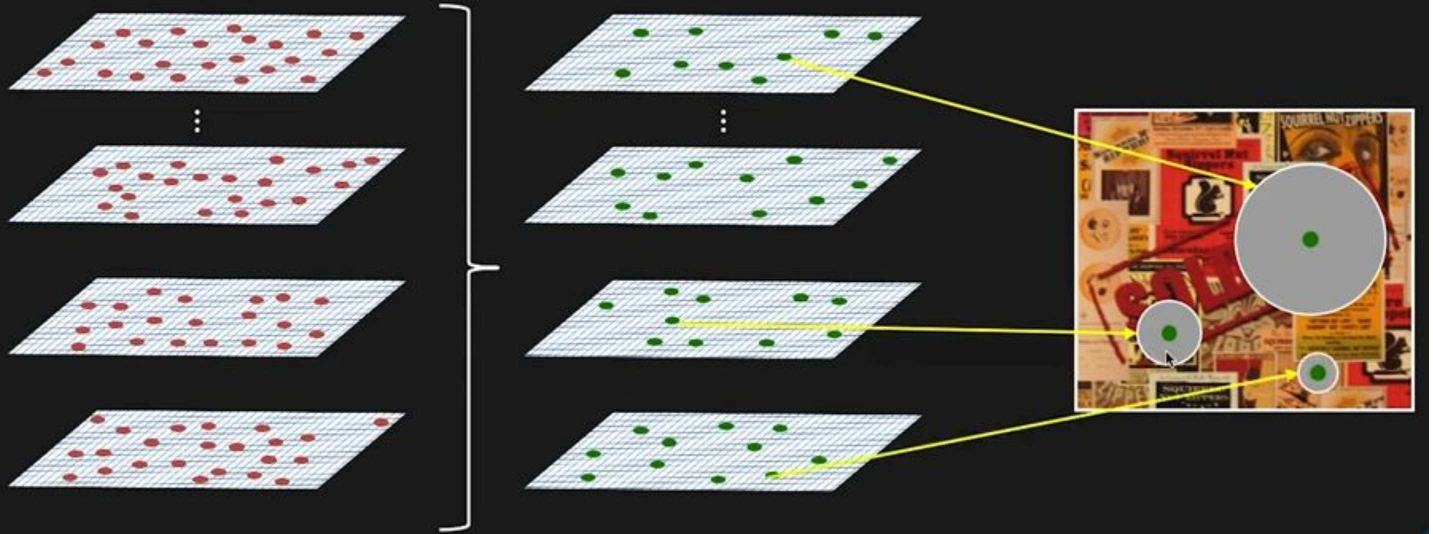


Difference of Gaussians (DoG)
$\approx (s-1)\sigma^2 \nabla^2 S(x, y, \sigma)$

Find Extremum in every 3x3x3 grid

Interest Point Candidates
(includes weak extrema)

Extracting SIFT Interest Points



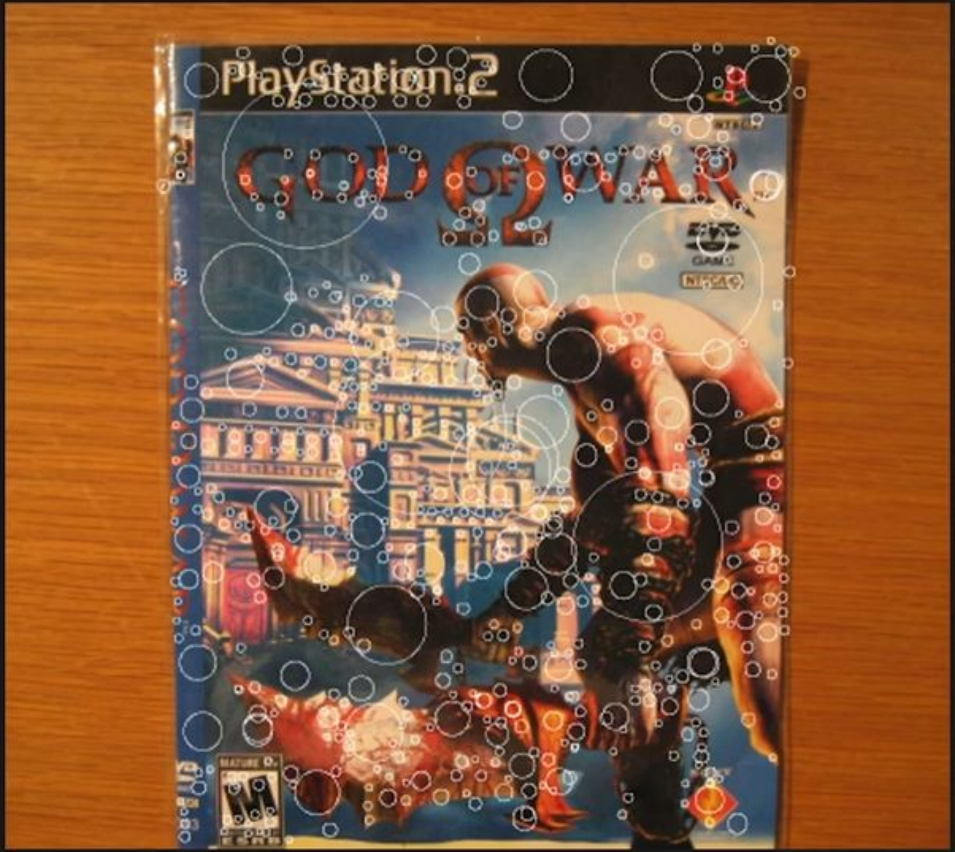
Interest Point Candidates

(includes weak extrema)

SIFT Interest Points

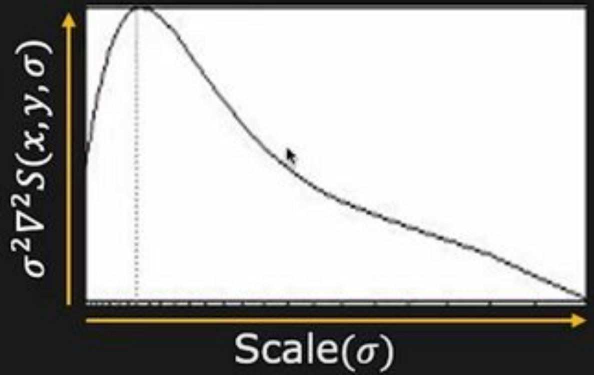
(after removing weak extrema)

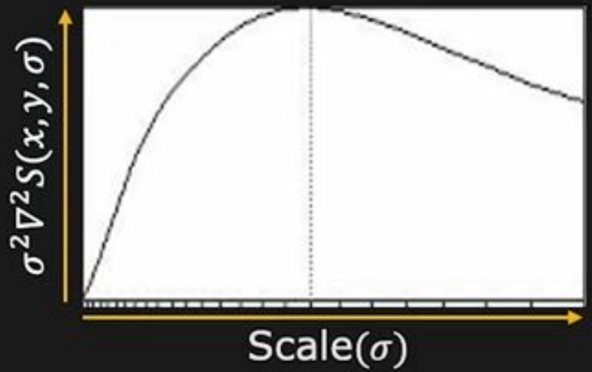
SIFT Detection Examples



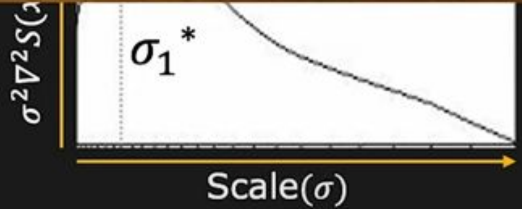
SIFT Detection Examples

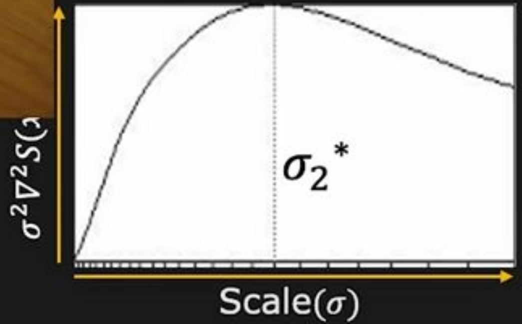
SIFT Scale Invariance





SIFT Scale Invariance





$\frac{\sigma_1^*}{\sigma_2^*}$: Ratio of Blob Sizes

Computing the Principal Orientation

Use the histogram of gradient directions

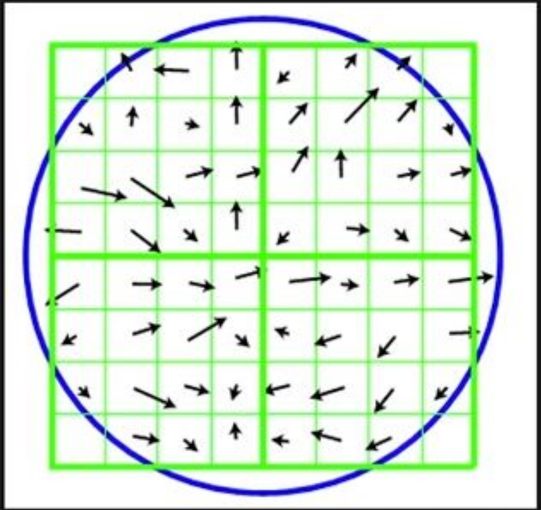


Image gradient directions

$$\theta = \tan^{-1}\left(\frac{\partial I}{\partial y} \Big/ \frac{\partial I}{\partial x}\right)$$

Computing the Principal Orientation

Use the histogram of gradient directions

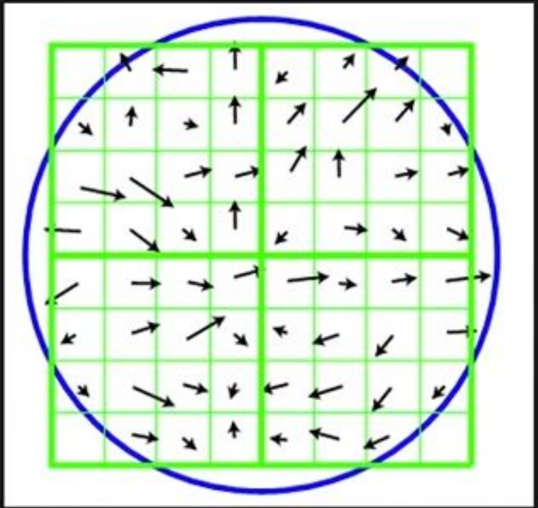
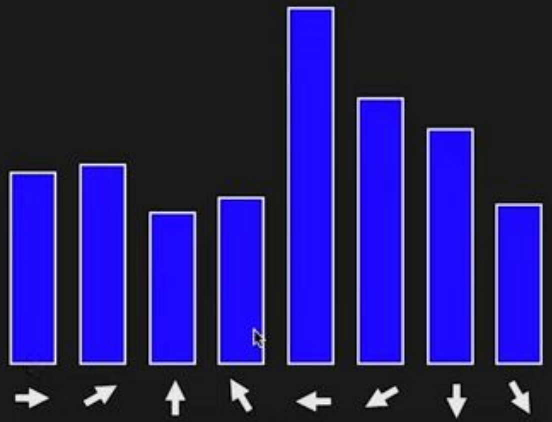


Image gradient directions

$$\theta = \tan^{-1}\left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x}\right)$$



Computing the Principal Orientation

Use the histogram of gradient directions

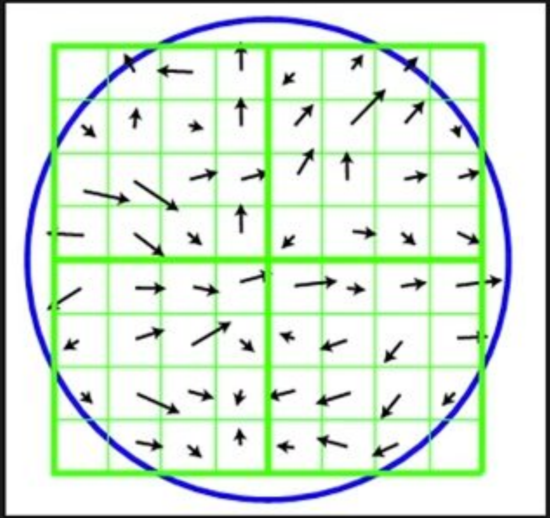
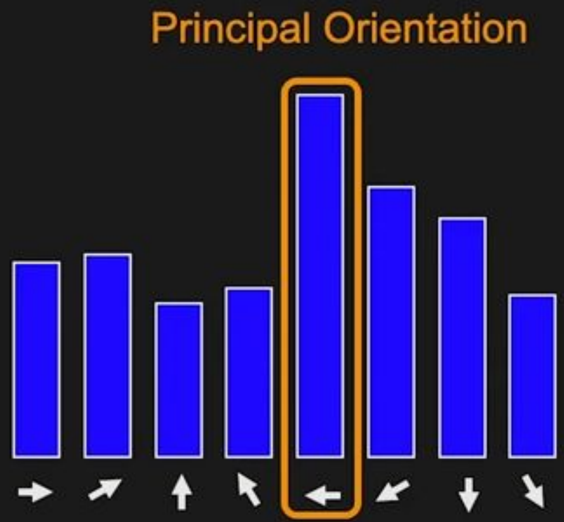


Image gradient directions

$$\theta = \tan^{-1}\left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x}\right)$$



Choose the most prominent gradient direction

SIFT Rotation Invariance

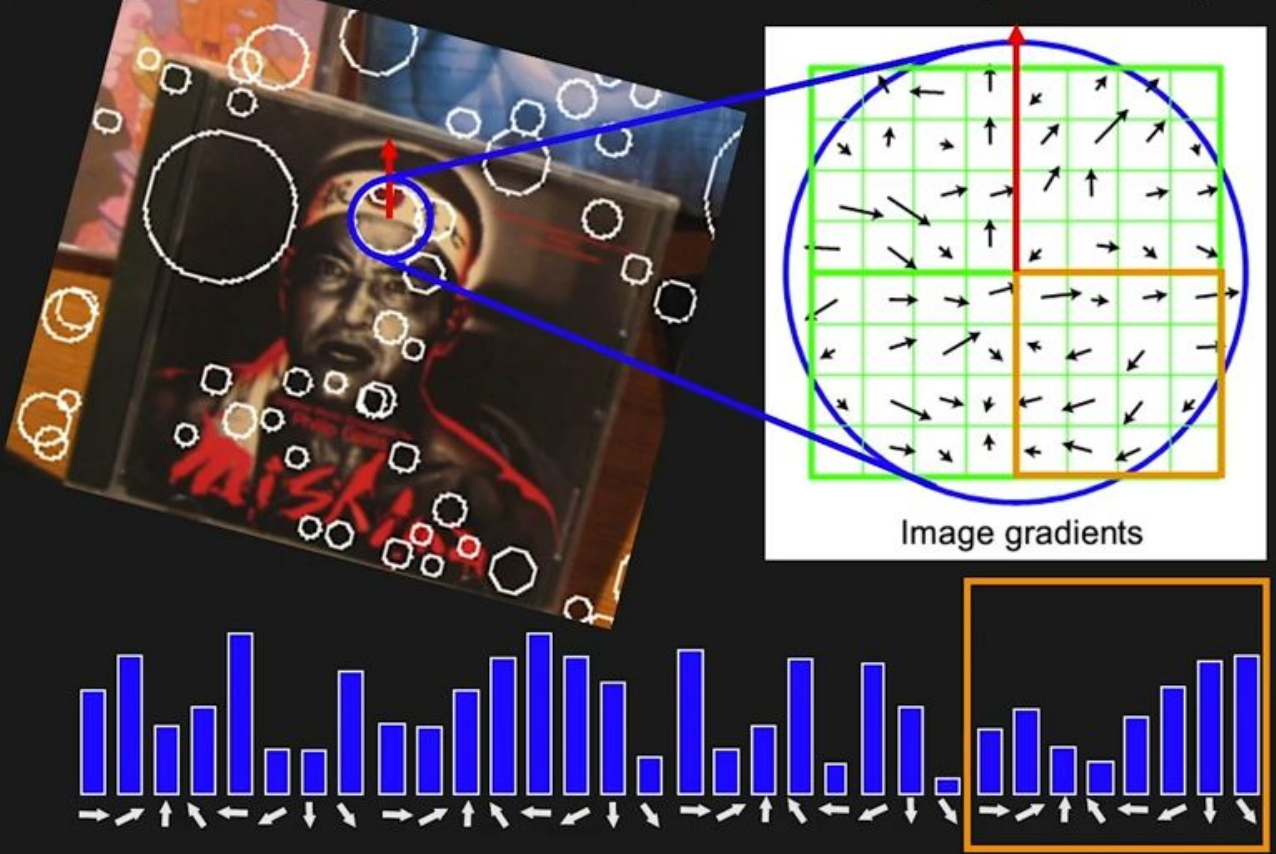
Use the principal orientation to undo rotation

SIFT Rotation Invariance

Use the principal orientation to undo rotation

SIFT Descriptor

Histograms of gradient directions over spatial regions



What about the brightness?

By neglecting the magnitude of the gradient, the SIFT process becomes invariant to brightness as well.

Comparing SIFT Descriptors

Essentially comparing two arrays of data.

Let $H_1(k)$ and $H_2(k)$ be two arrays of data of length N.

L2 Distance:

$$d(H_1, H_2) = \sqrt{\sum_k \left(H_1(k) - H_2(k)\right)^2}$$

Smaller the distance metric, better the match.

Perfect match when $d(H_1, H_2) = 0$

Comparing SIFT Descriptors

Essentially comparing two arrays of data.

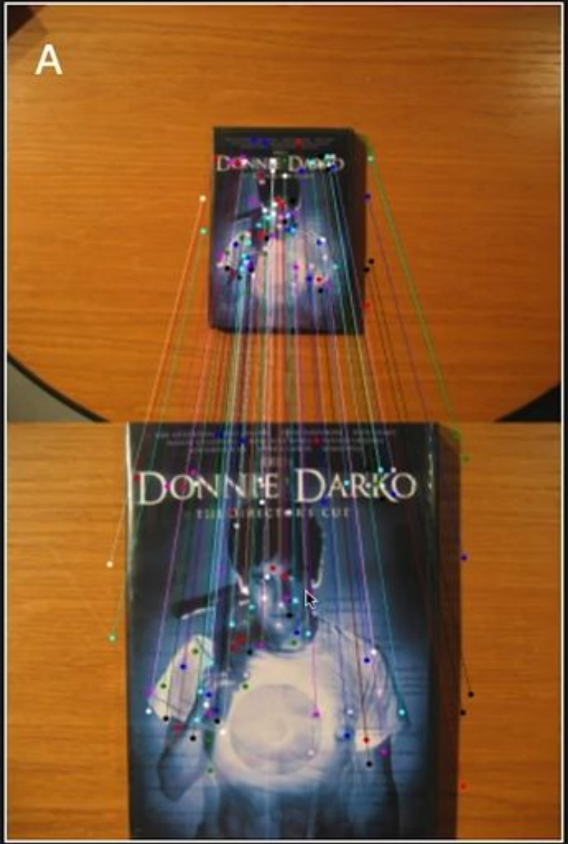
Let $H_1(k)$ and $H_2(k)$ be two arrays of data of length N.

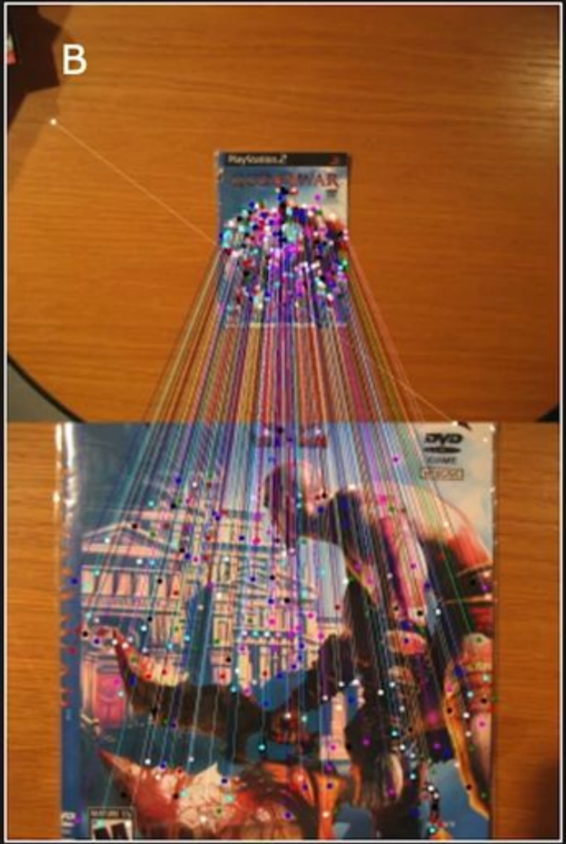
Intersection:

$$d(H_1, H_2) = \sum_k \min\big(H_1(k), H_2(k)\big)$$

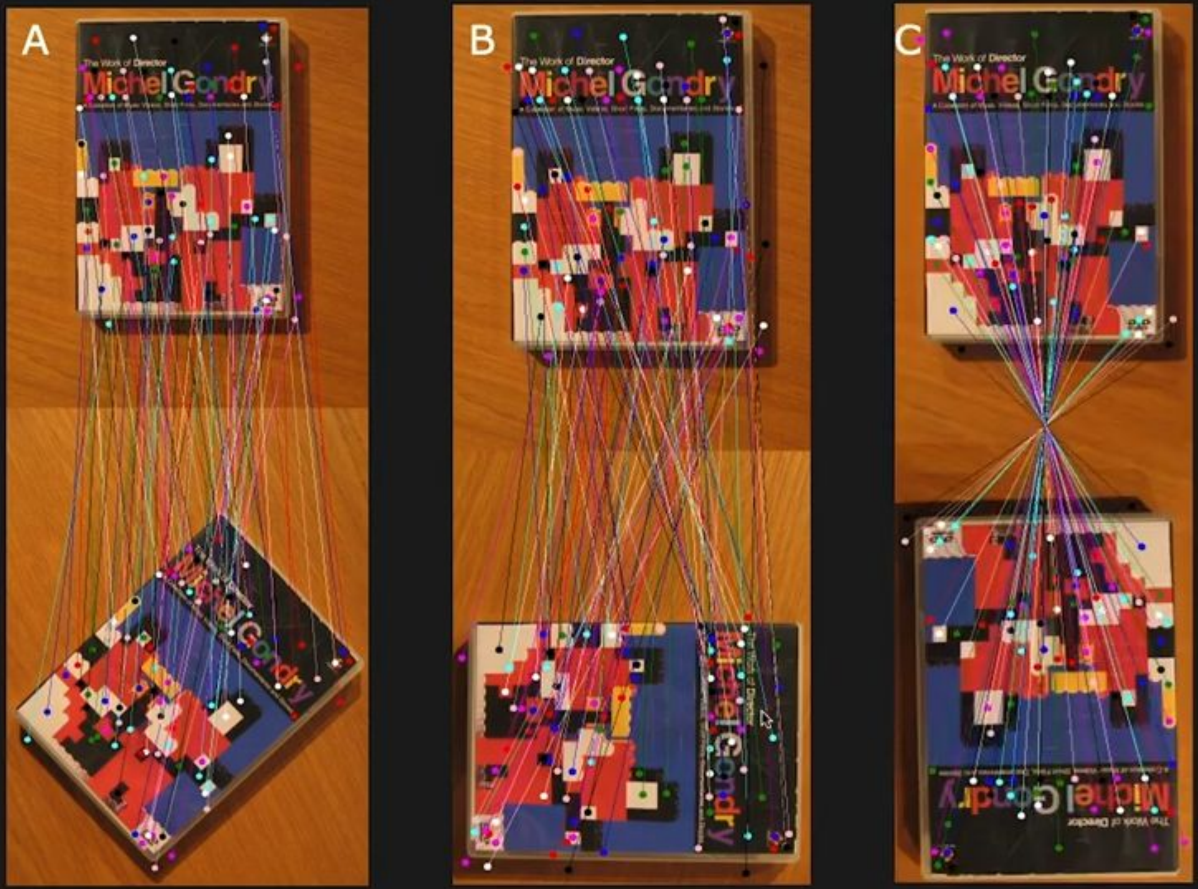
Larger the distance metric, better the match.

SIFT Results: Scale Invariance

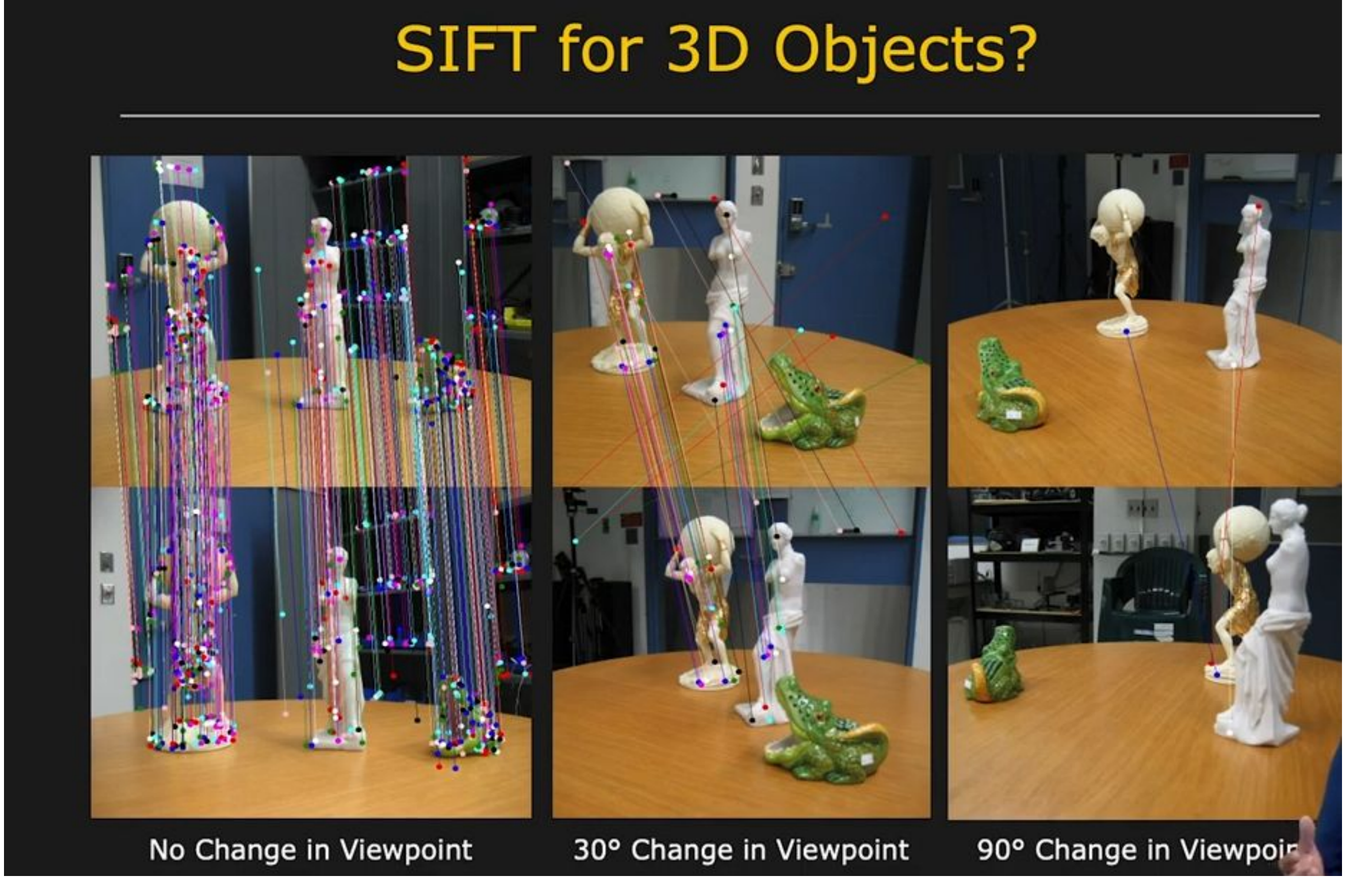




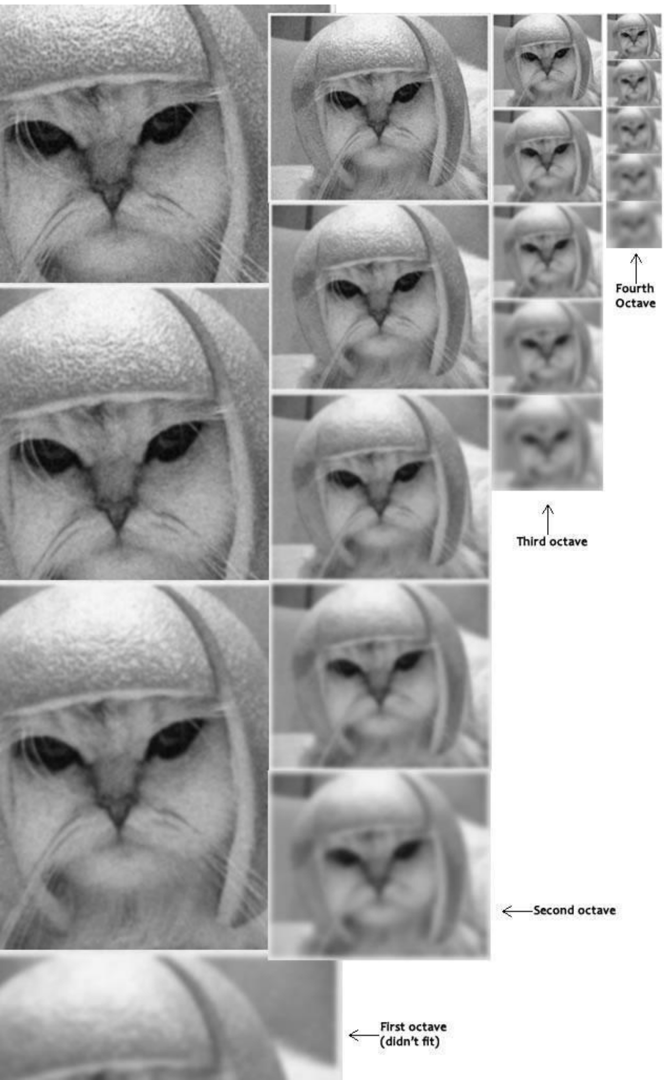
SIFT Results: Rotation Invariance



SIFT Results: Robustness to Clutter

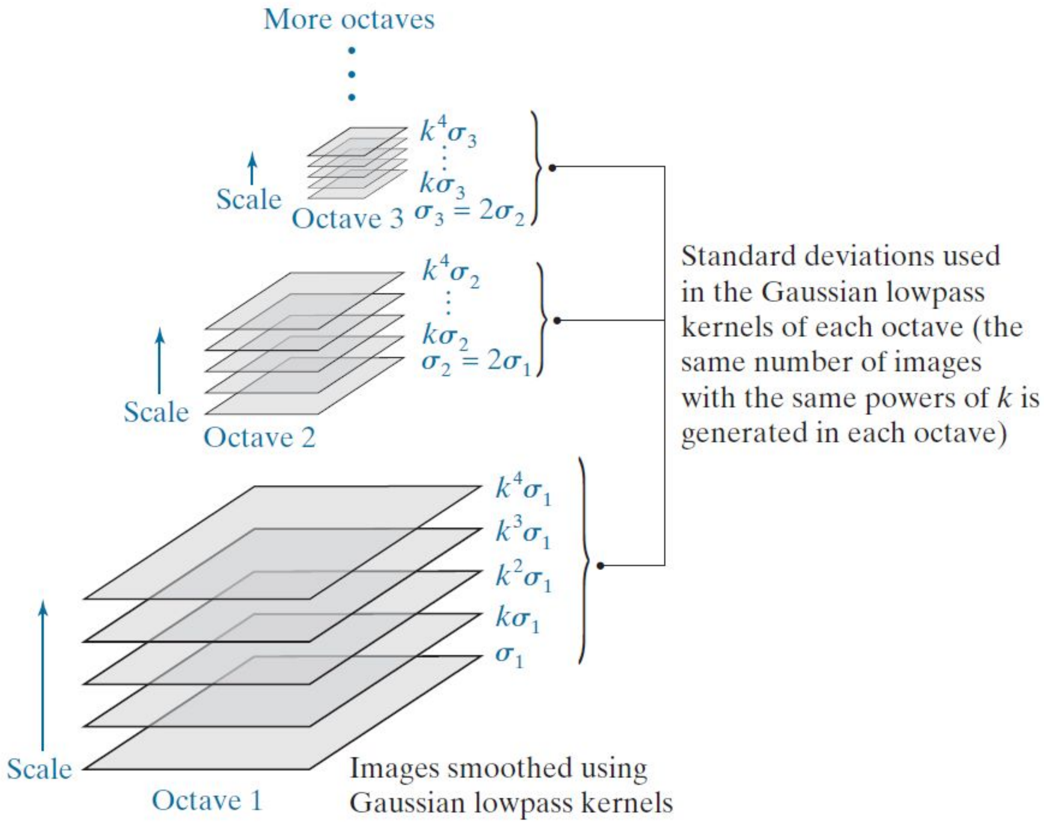


Reliable with only slight changes of viewpoints for 3D objects



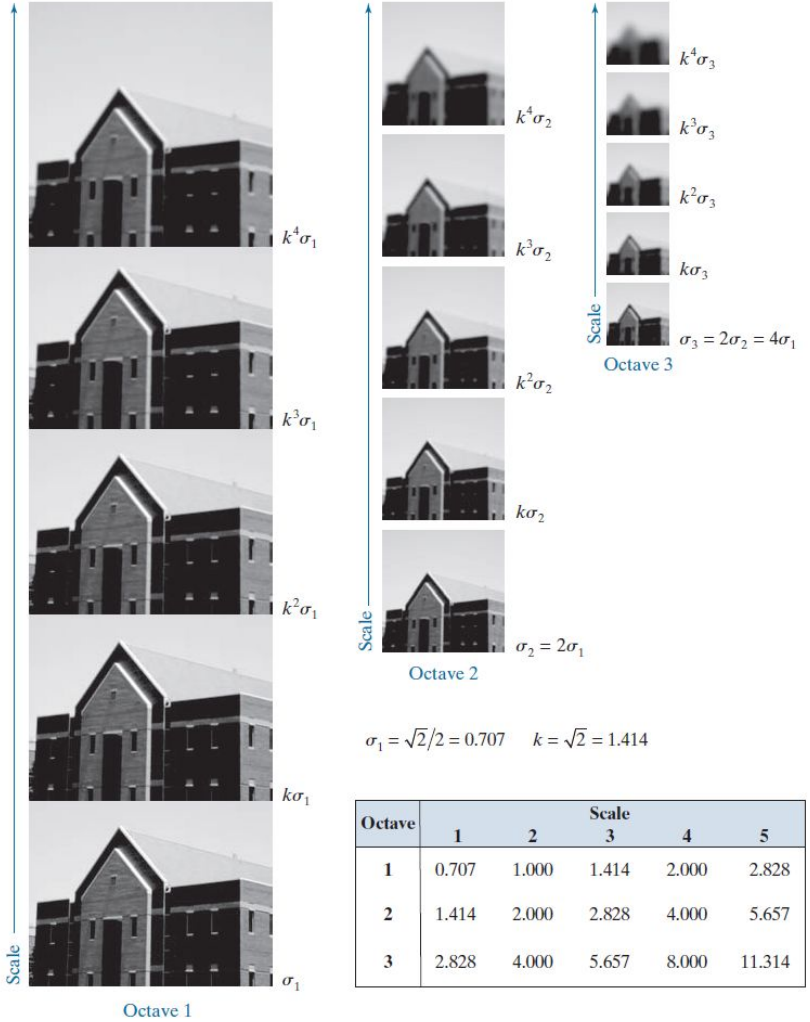
Implementation @ multiple sizes (octaves)

- Any octave is created by reducing the original size to half.
- We double the standard deviation as we create the next octaves.
- 5 blurred images in an octave.



More octaves
$k^4\sigma_3$
$k\sigma_3$
$\sigma_3 = 2\sigma_2$
Scale
Octave 3
$k^4\sigma_2$
$k\sigma_2$
$\sigma_2 = 2\sigma_1$
Scale
Octave 2
$k^4\sigma_1$
$k^3\sigma_1$
$k^2\sigma_1$
$k\sigma_1$
σ_1
Scale
Octave 1
Images smoothed using
Gaussian lowpass kernels
Standard deviations used
in the Gaussian lowpass
kernels of each octave (the
same number of images
with the same powers of k is
generated in each octave)

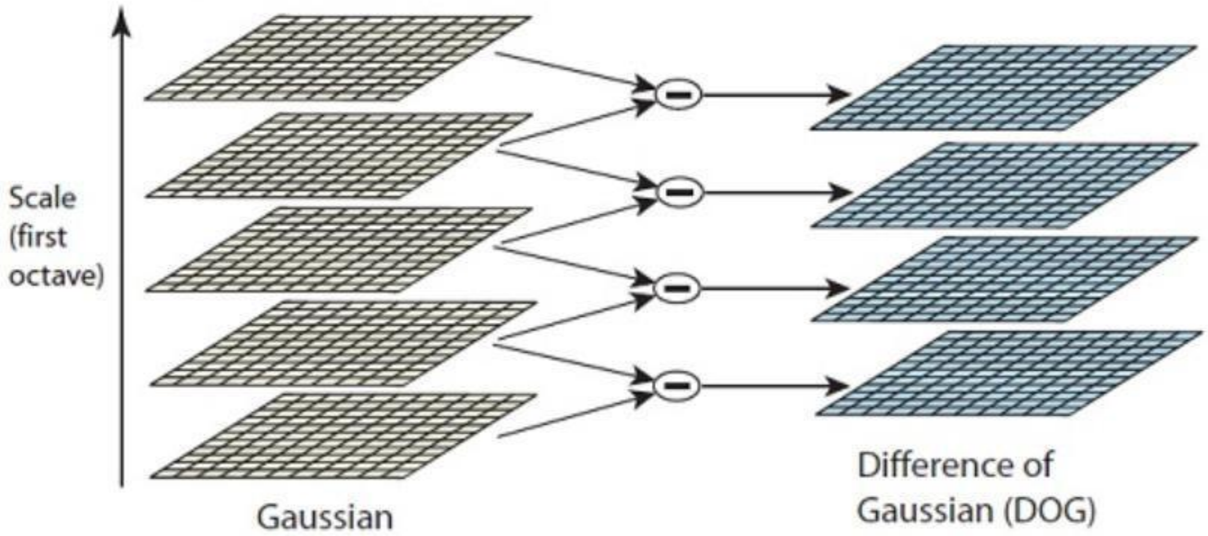
FIGURE 11.57 Illustration using images of the first three octaves of scale space in SIFT. The entries in the table are values of standard deviation used at each scale of each octave. For example the standard deviation used in scale 2 of octave 1 is $k\sigma_1$, which is equal to 1.0. (The images of octave 1 are shown slightly overlapped to fit in the figure space.)



$\sigma_1 = \sqrt{2}/2 = 0.707 \quad k = \sqrt{2} = 1.414$

Octave	Scale 1	2	3	4	5
1	0.707	1.000	1.414	2.000	2.828
2	1.414	2.000	2.828	4.000	5.657
3	2.828	4.000	5.657	8.000	11.314

DoG (for an octave)



Sample DOGs from each octave