Korea Institute of Energy Technology

Advanced Computer Vision
Week 07

Oct. 11, 2022

Seokju Lee

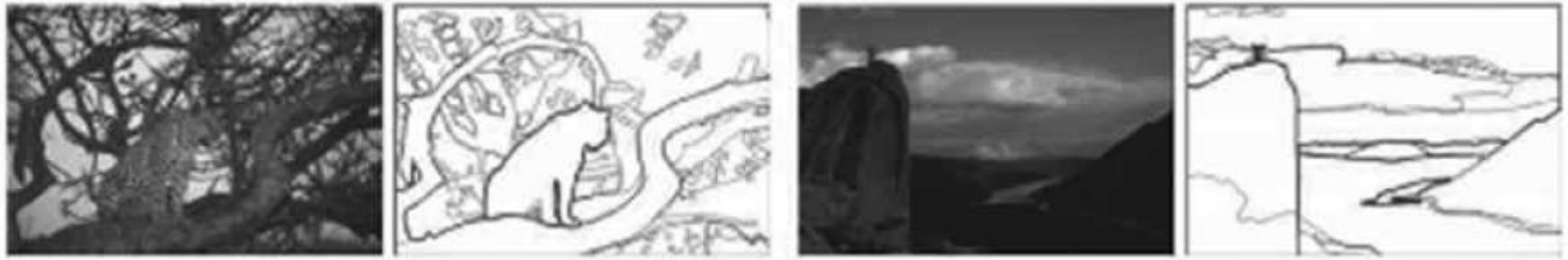
Parts of slides are by Prof. In So Kweon and Prof. Shree Nayar

KENTECH
Korea Institute of Energy Technology

Review: Edges & Corners

Review: Edges & Corners

1) Edge = **Gradient**



Gradient (partial derivatives) represents the direction of **most rapid change** in intensity.

$$\boxed{\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right]}$$

Pronounced as "Del I"

Gradient Magnitude $S = \|\nabla I\| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$

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

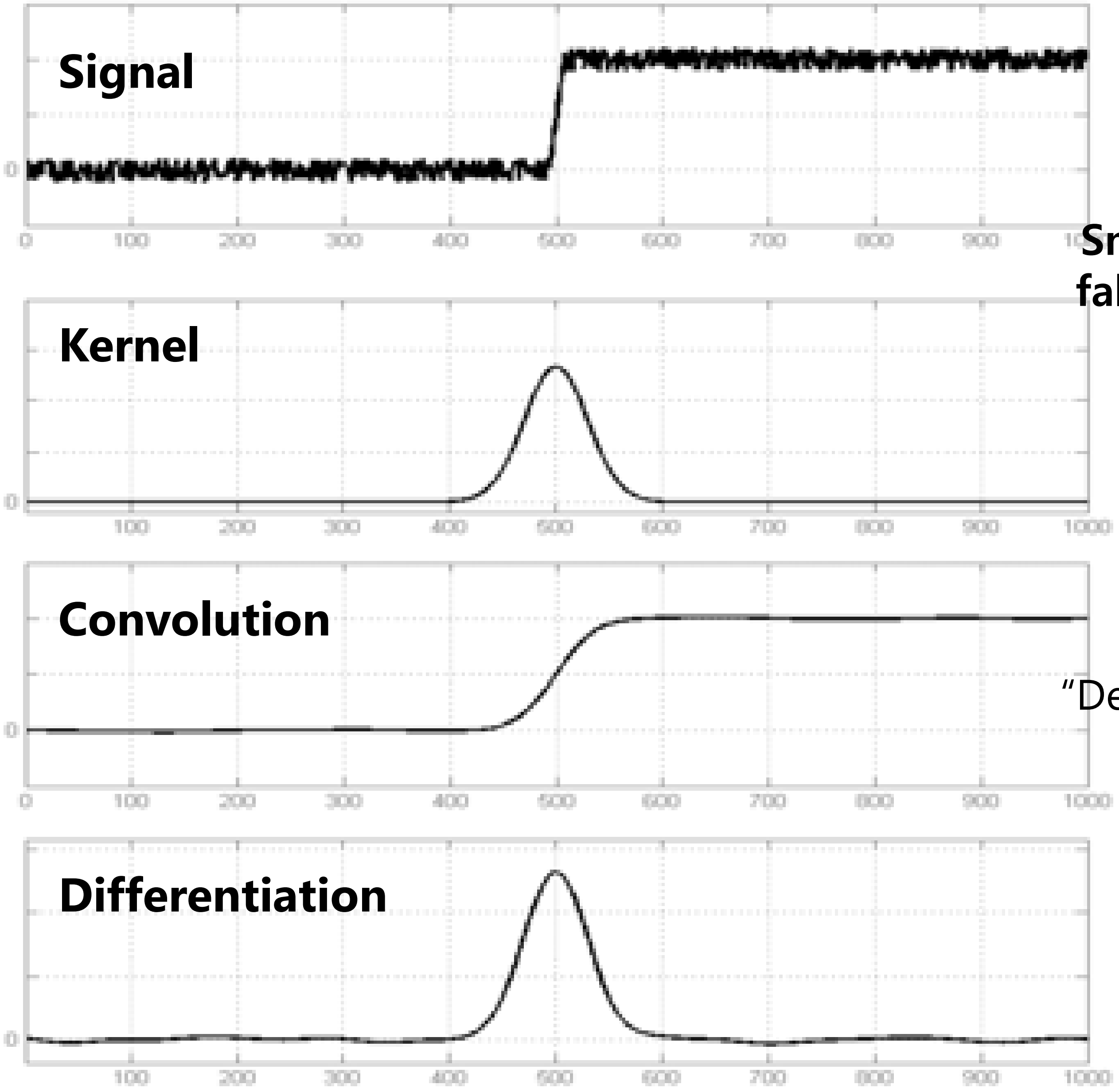
Edge Example

$$f$$

$$h$$

$$h * f$$

$$\frac{\partial}{\partial x}(h * f)$$



Smoothing the Image to prevent **false** edge detections = **Gaussian**

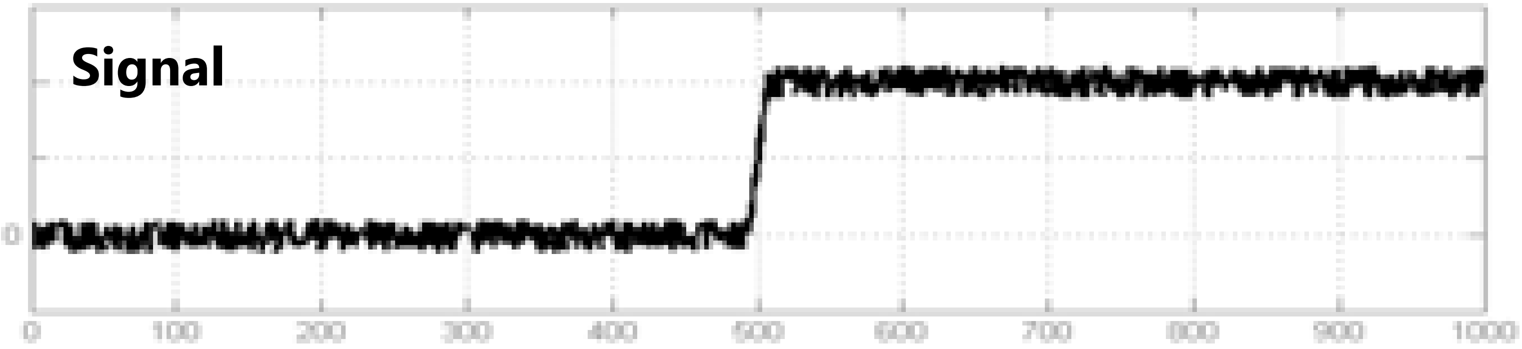
"Derivative Theorem of Convolution"

$$\frac{\partial}{\partial x}(h * f) = \left(\frac{\partial}{\partial x} h\right) * f$$

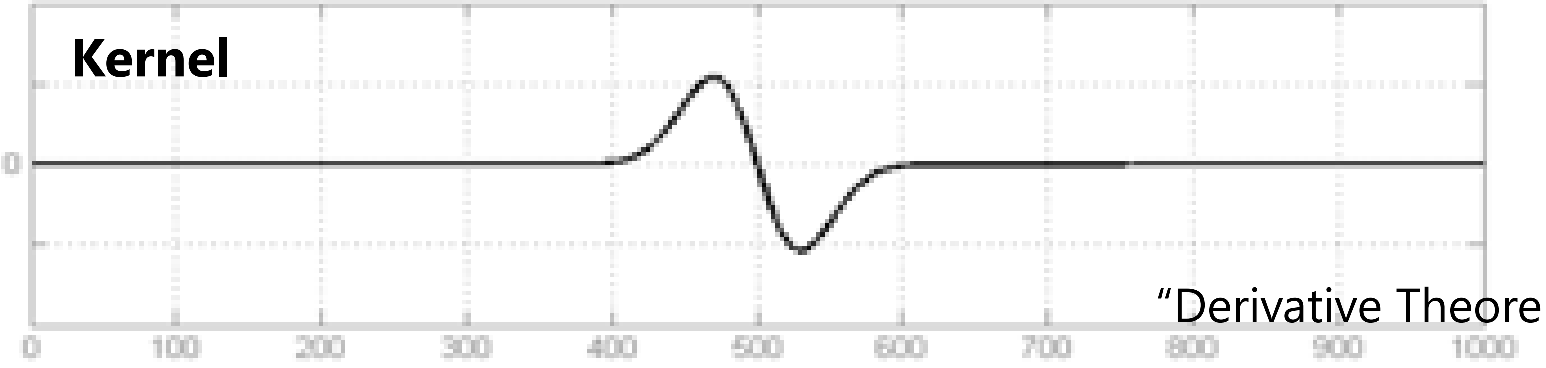
Slide from UC Davis

Edge Example

$$f$$



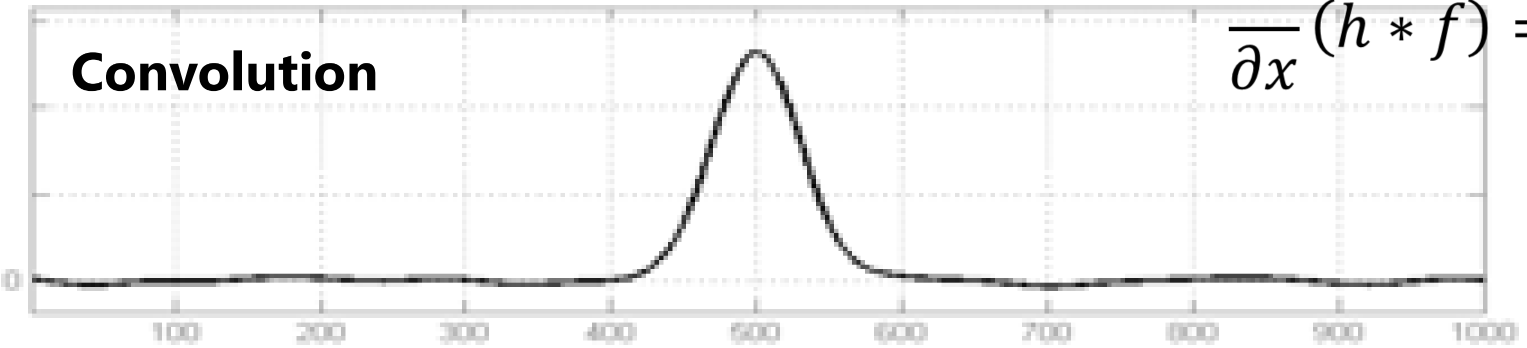
$$\frac{\partial}{\partial x}h$$



"Derivative Theorem of Convolution"

$$\frac{\partial}{\partial x}(h * f) = \left(\frac{\partial}{\partial x}h\right) * f$$

$$(\frac{\partial}{\partial x}h) * f$$

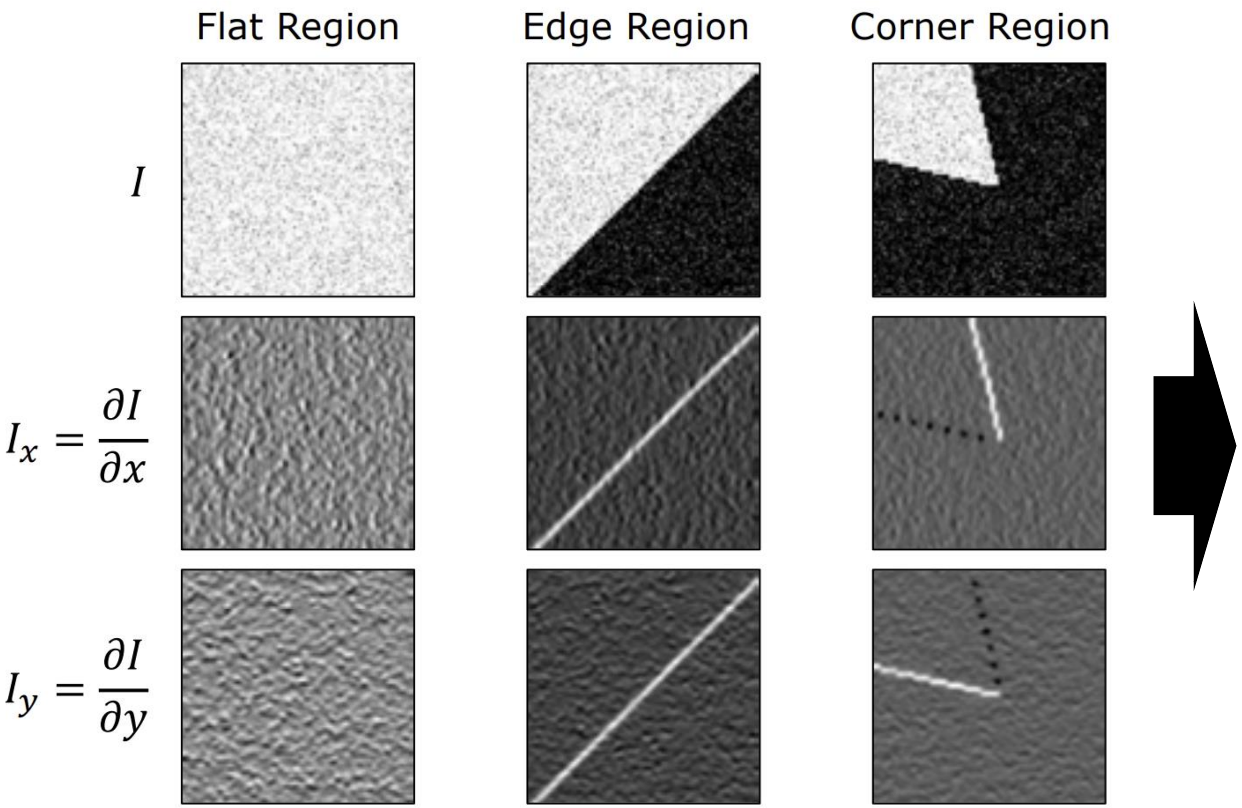


Slide from UC Davis

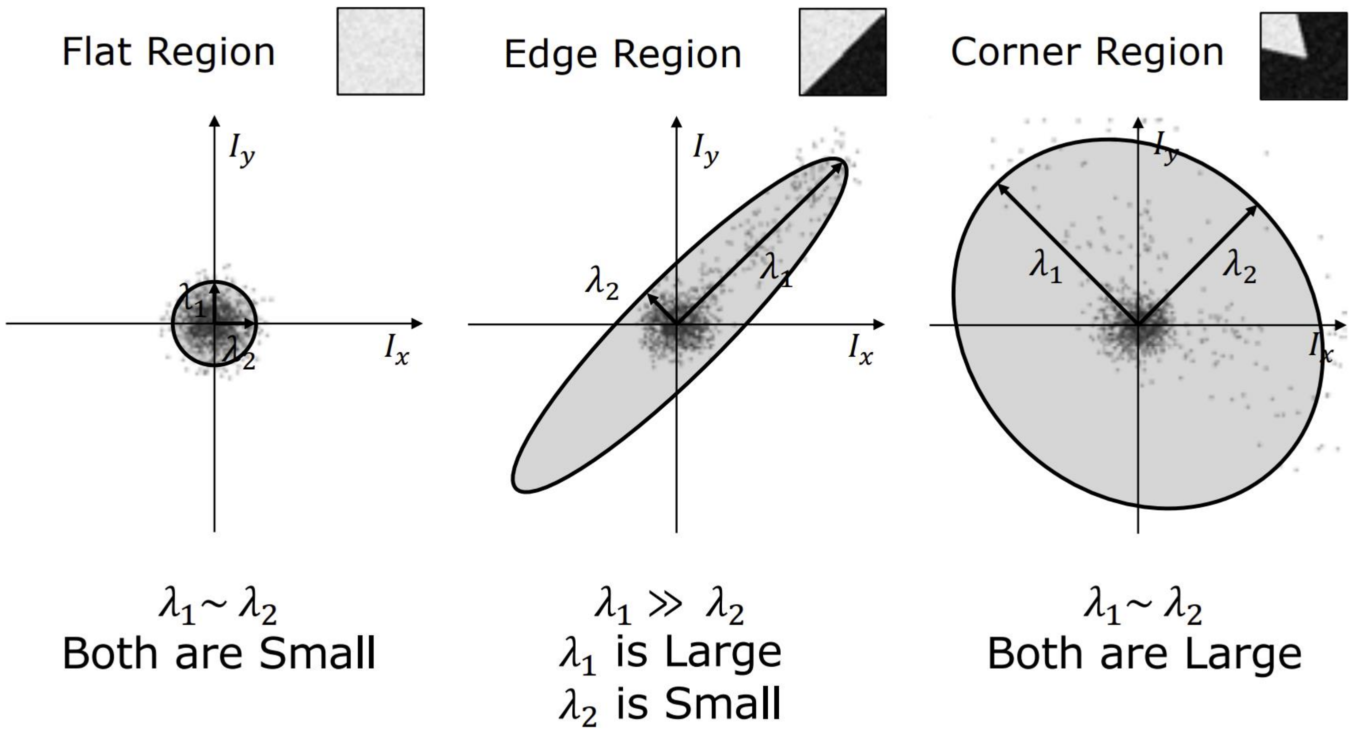
Review: Edges & Corners

2) Corner = **Point** where two edges meet

Rapid changes of image intensity in **two directions** within a small region.



Principal component ellipse



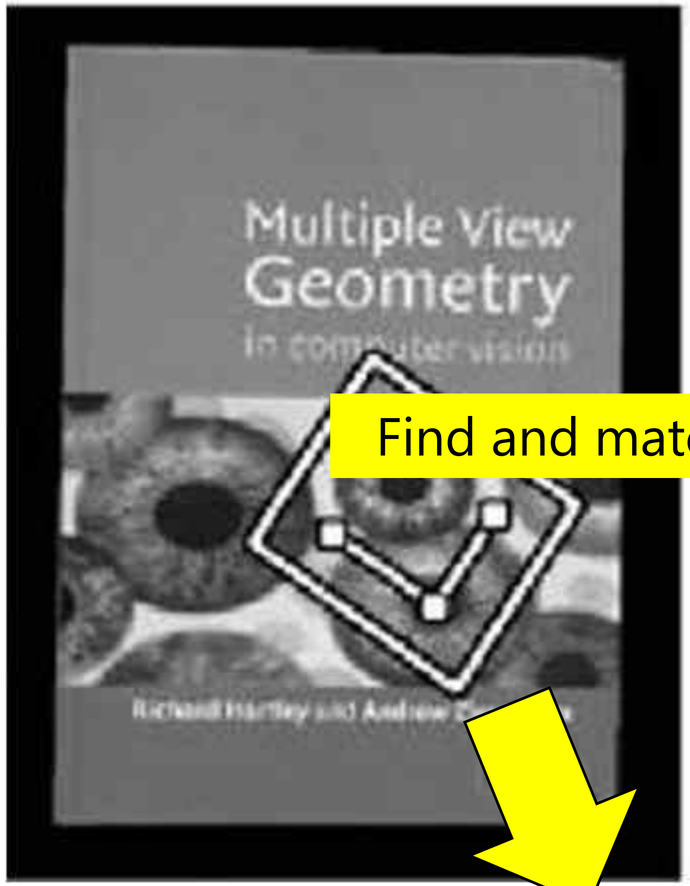
KENTECH
Korea Institute of Energy Technology

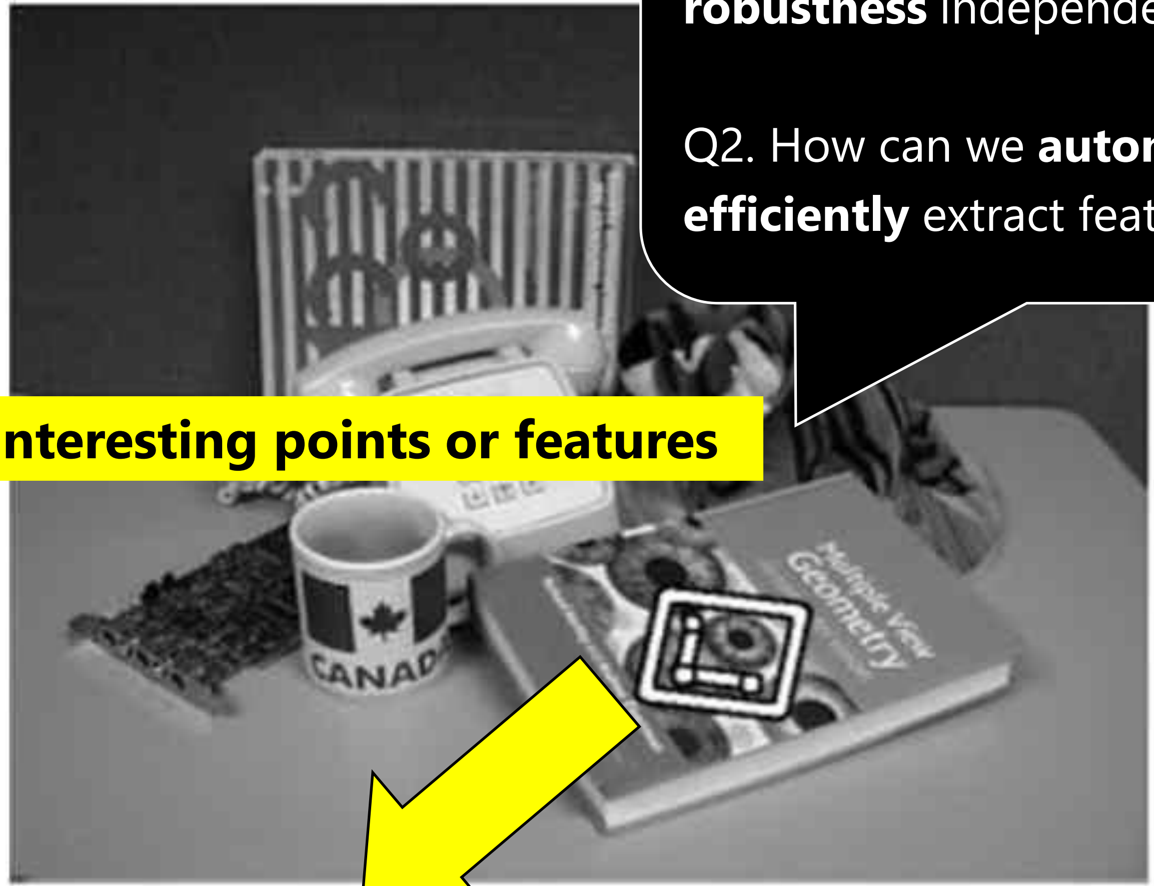
Scale Invariant Feature Transform (SIFT) Detector

Back to "Image Matching"

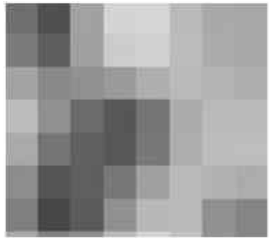
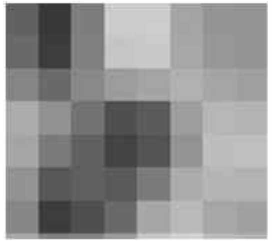
Q1. What good features that show **robustness** independent of **variations**?

Q2. How can we **automatically** and **efficiently** extract features in images?





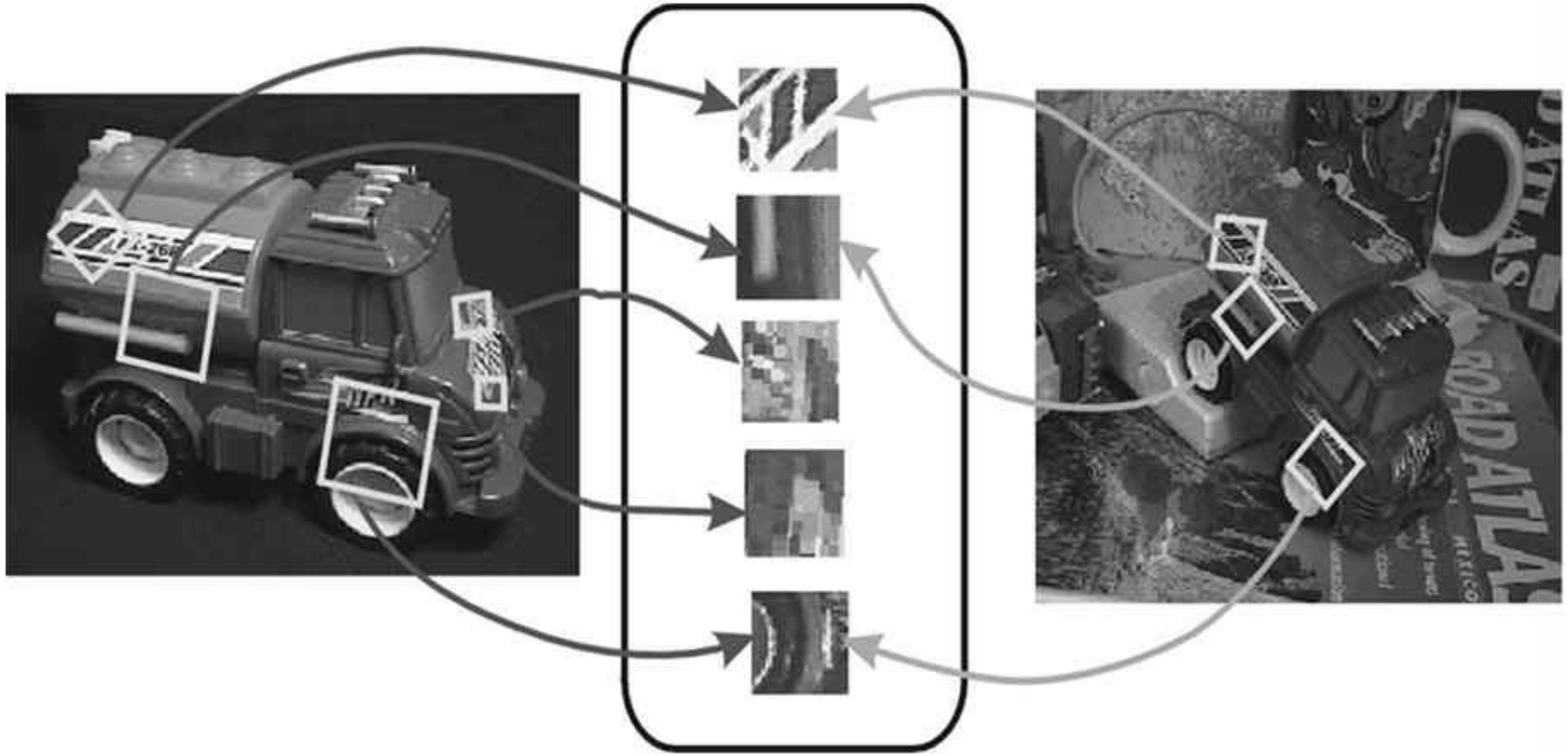
Find and match **interesting points or features**


≈


→ How to extract **generic** features?

Invariant Local Features

- ✓ Find features that are **invariant** to **transformations**:
  - **Geometric** invariance: translation, rotation, scale, …
  - **Photometric** invariance: brightness, exposure, …



Slide by D. Lowe.

Let's Learn SIFT Detector

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

Topics:

(1) What is an Interest Point?

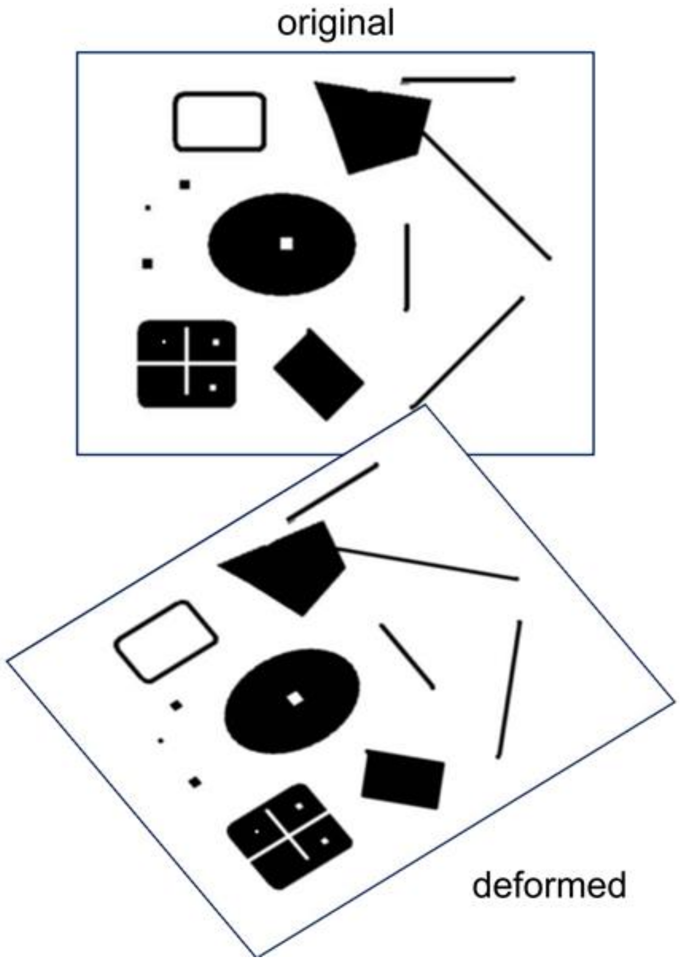
(2) Detecting Blobs

(3) SIFT Detector

(4) SIFT Descriptor

What is an Interest Point?

Interest points = **keypoints**, also sometimes called **features**.

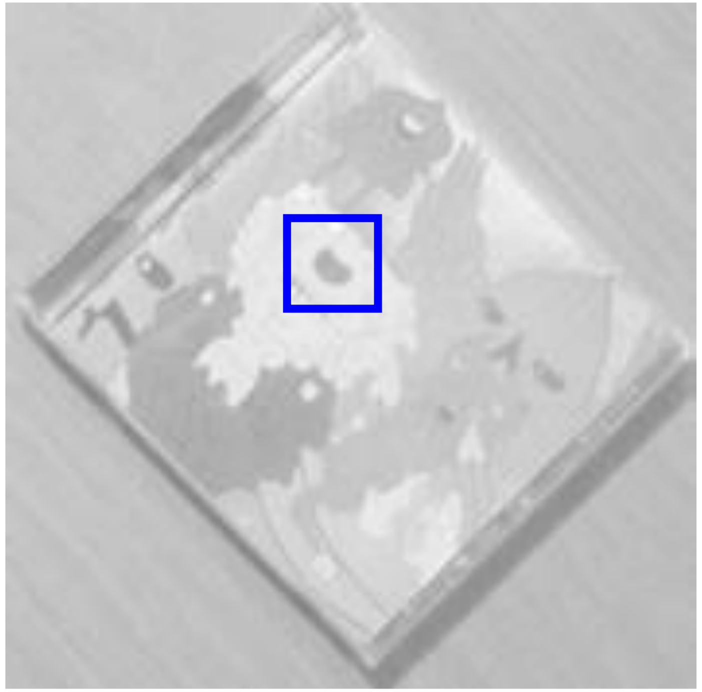


Suppose you have to click on some point, go away and come back after I **deform** the image, and click on the same points again.

→ Which points would you choose?

What is an Interest Point?

"Raw images are hard to match"



Different size, orientation, lighting, brightness, etc.

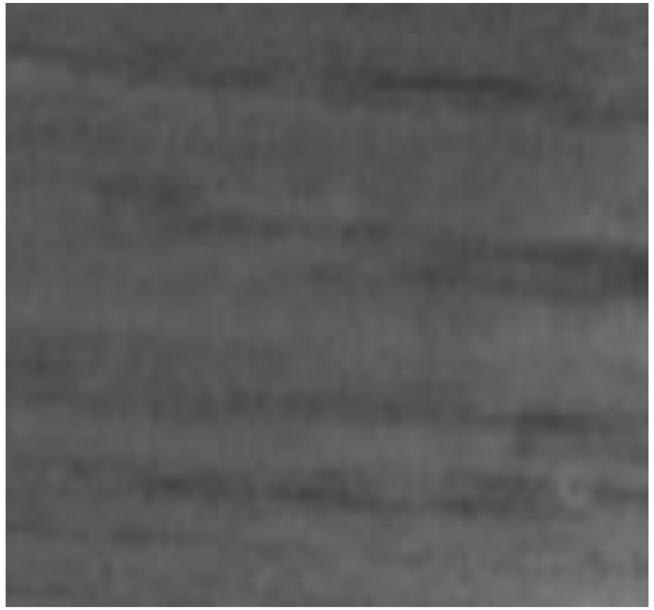
What is an Interest Point?

"Remove the variations"

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

What is an Interest Point?

"Some patches are **not** interesting"

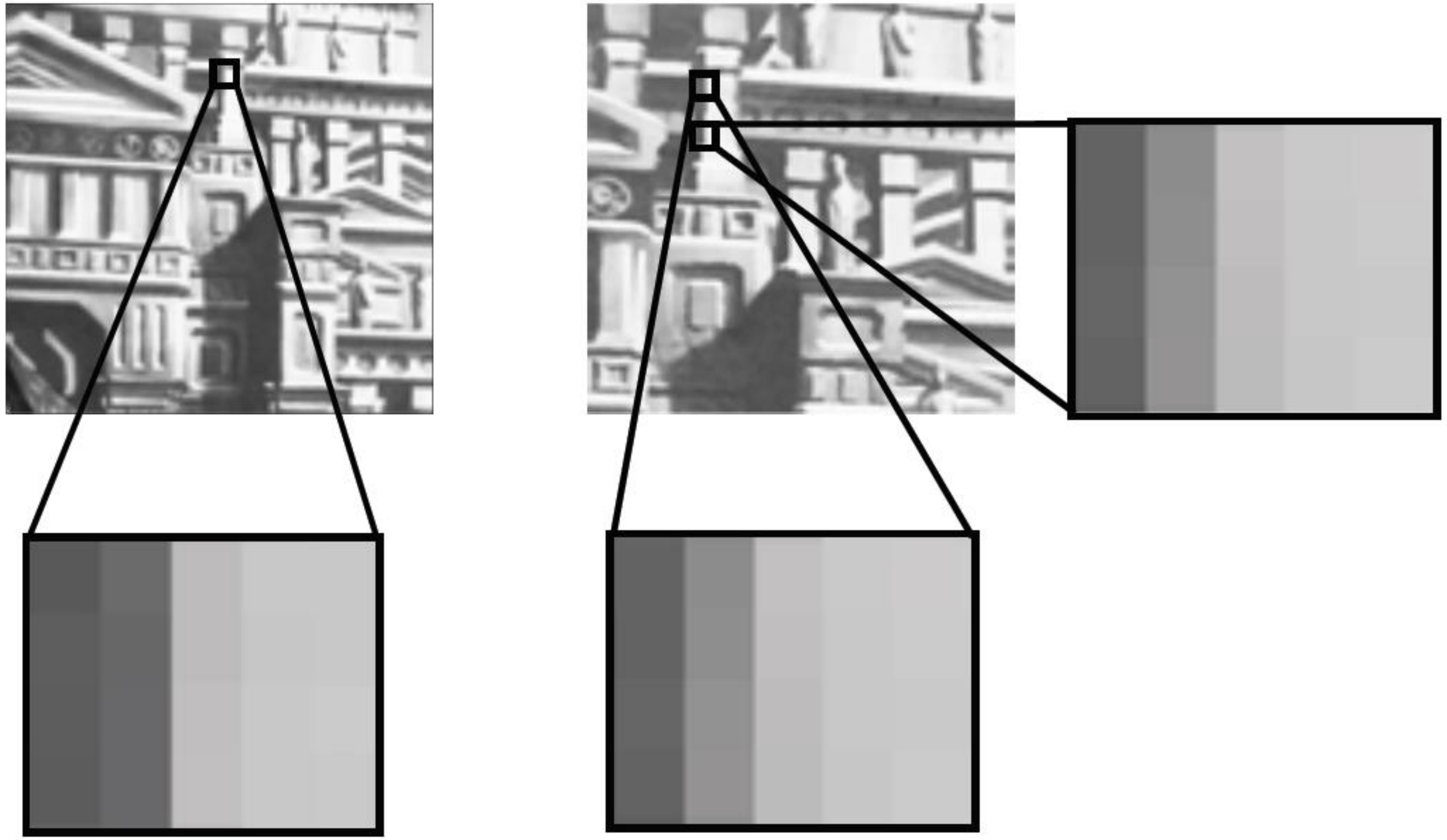


Background

What is an Interest Point?

- 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** specific position ☹

Are Blobs Interesting?

blob

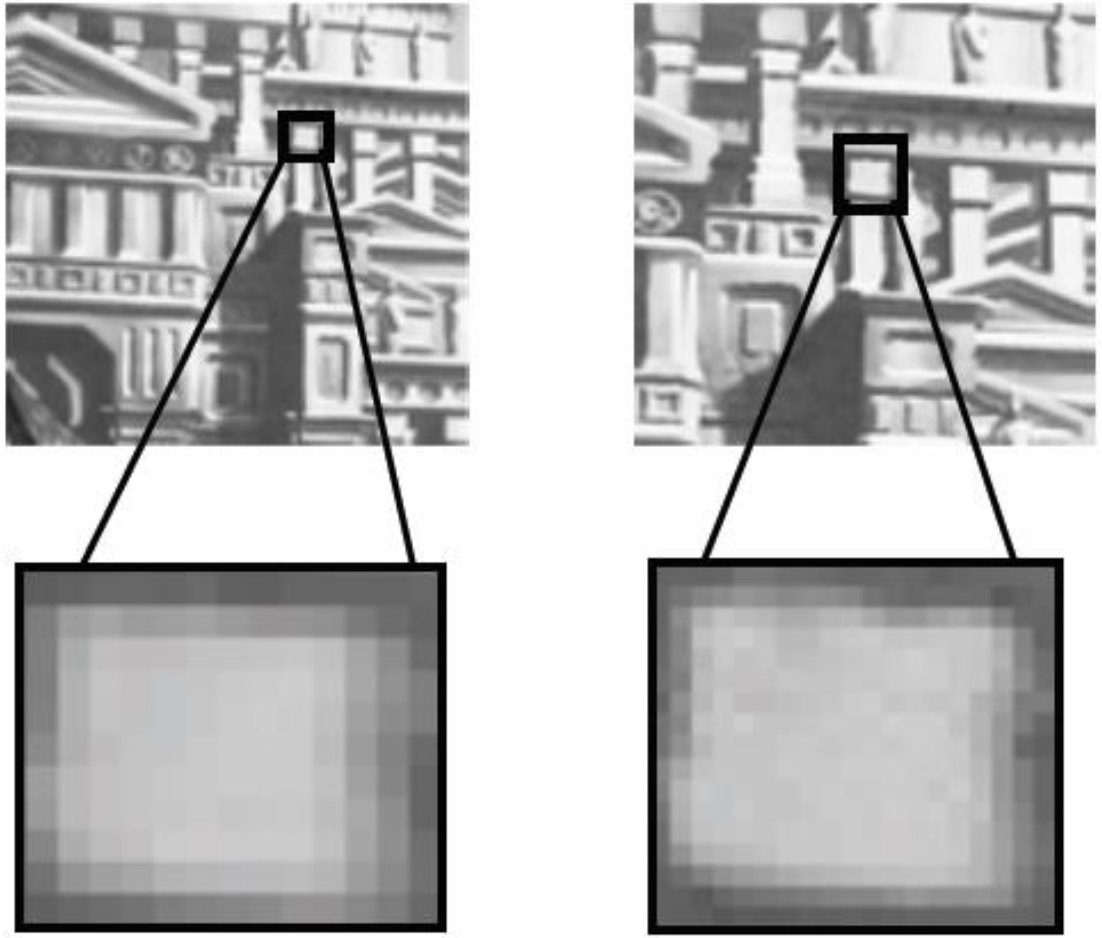
미국식[blɑːb] 영국식[blɒb]

명사

(작은) 방울, (작은) 색깔 부분

a **blob** of ink

잉크 한 방울



→ Yes! Blobs have **fixed position** and **definite size** ☺

Example Code for SIFT

Updated codes are uploaded in https://view.kentech.ac.kr/f088fa7f-874e-44bc-bd6d-6084b42dfdf7

```
$ cd OpenCV-Python-Tutorials/Src/FeatureDetectionAndDescription/SIFTAndSURF

$ python SIFT.py
```

Q. What does circle mean?

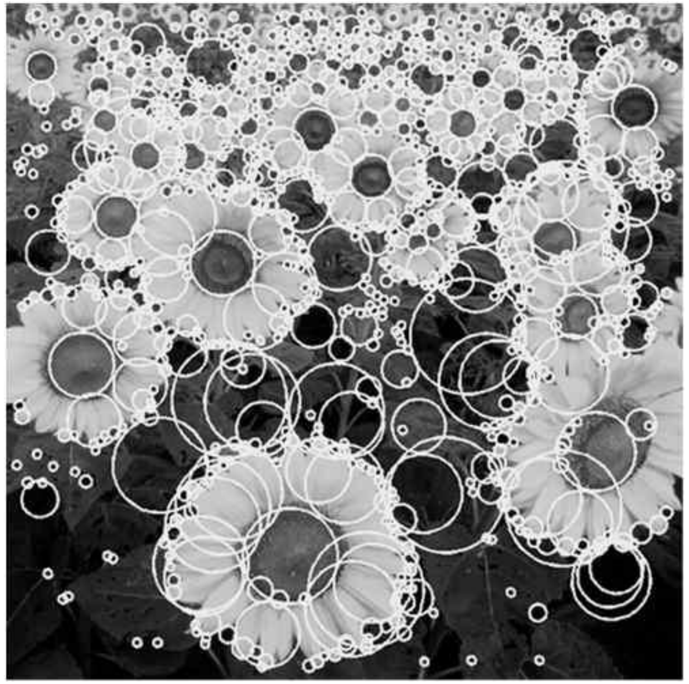
KENTECH
Korea Institute of Energy Technology

Blob Detector

Blob

- A **blob** is a **region** of an image where some properties (brightness or color, etc.) are **constant** or **approximately constant**.
- All the points in a blob can be considered to be **similar** to each other.

Corners



Blobs

Blobs as Interest Points

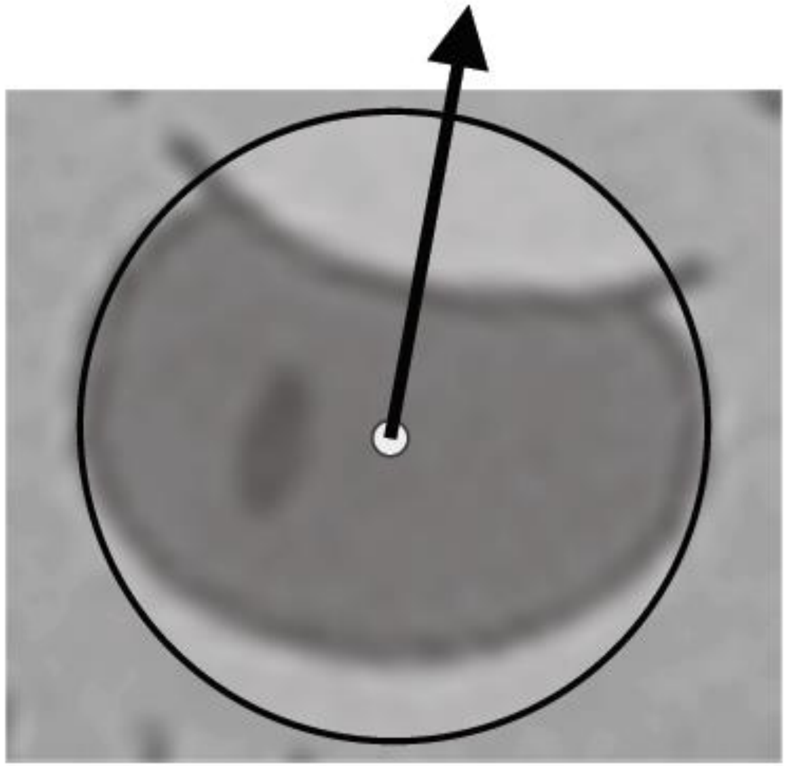
- To make a Blob-like feature useful, we need to:

  → **Locate** the blob

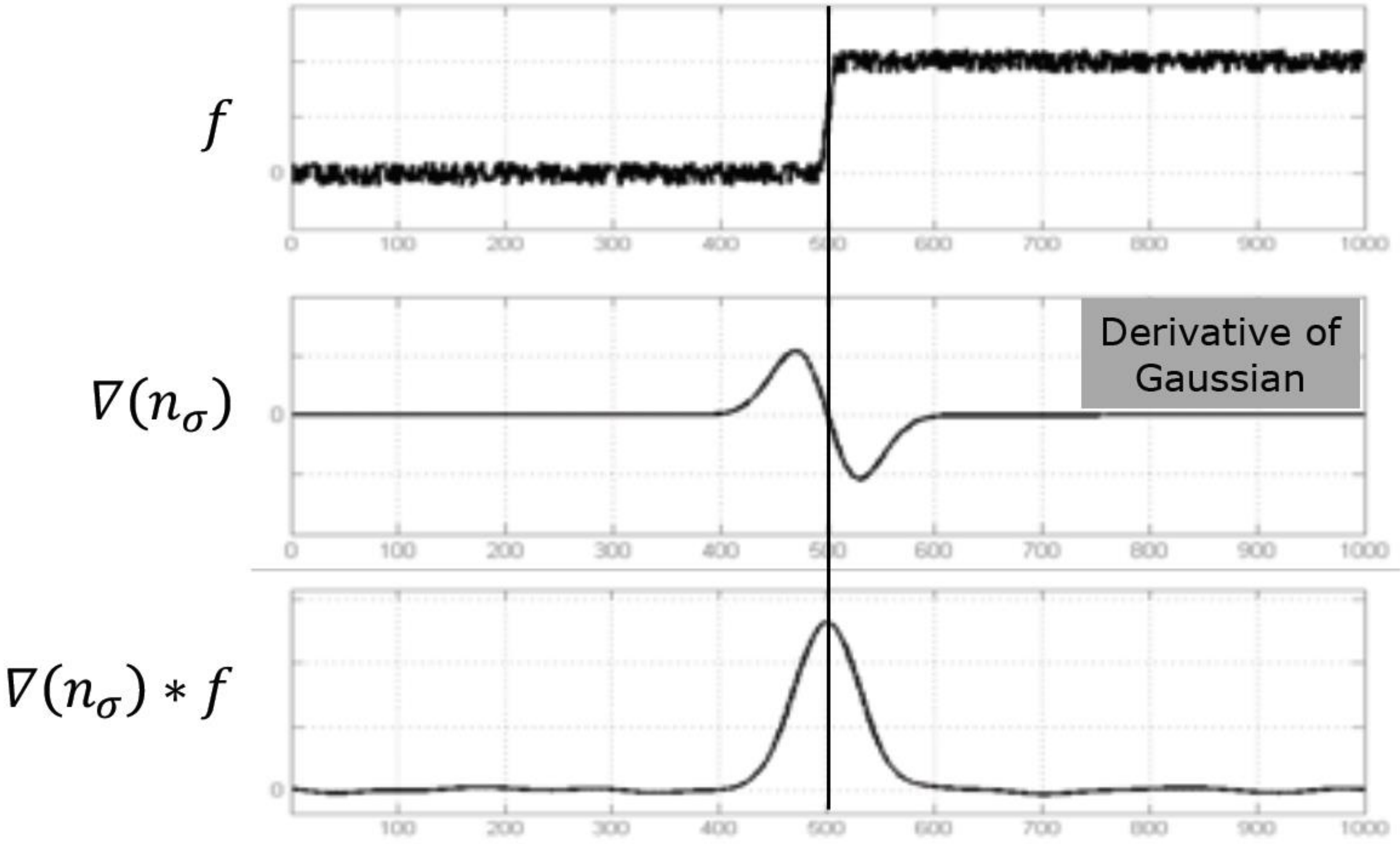
  → Determine its **size**

  → Determine its **orientation**

  → Formulate a **description** that is **independent** of size and orientation

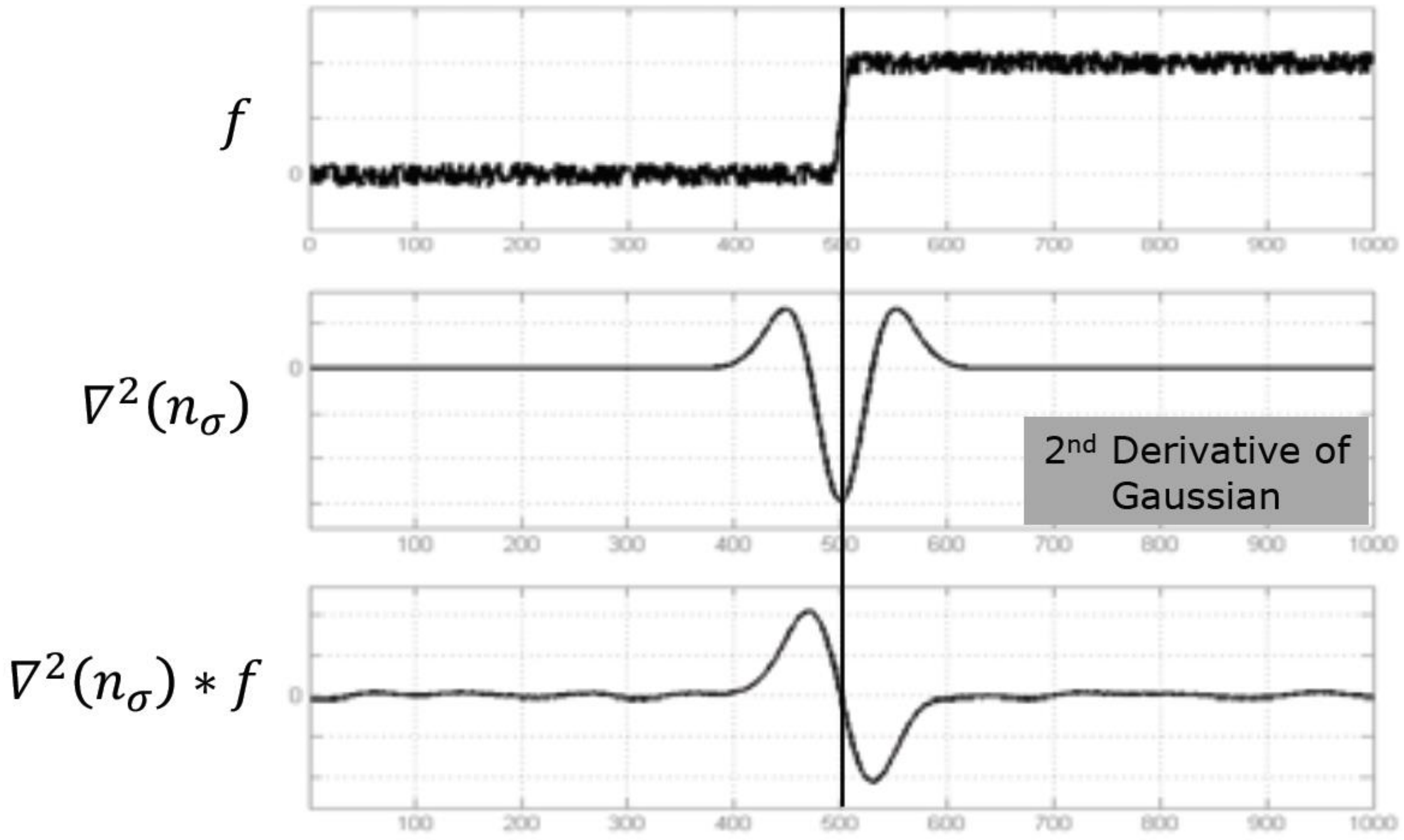


Review: Derivative of Gaussian



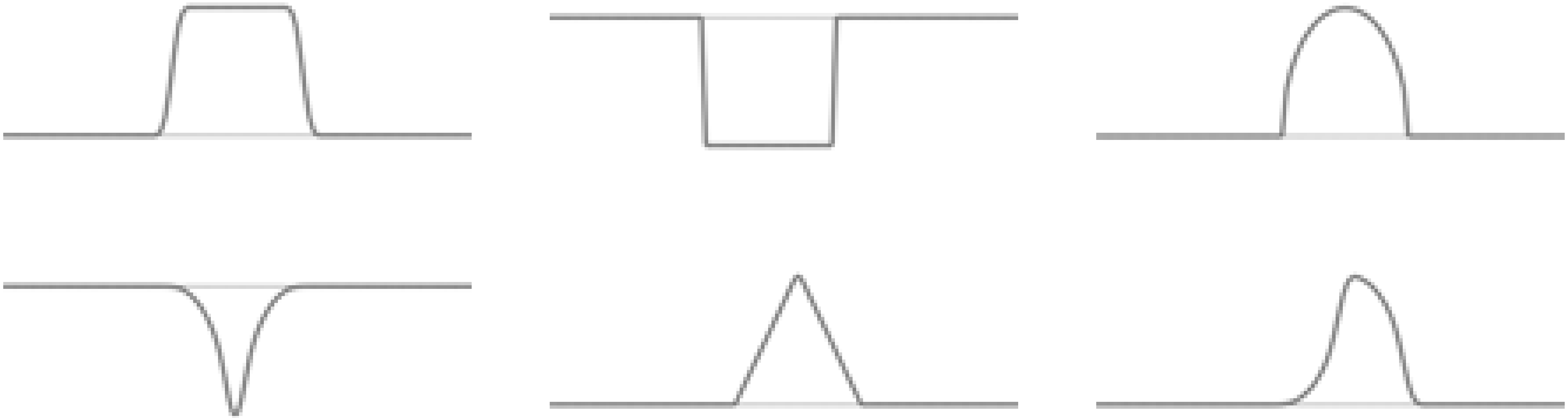
→ **Extremum** of derivative of Gaussian denotes an **edge**.

Review: 2nd Derivative of Gaussian



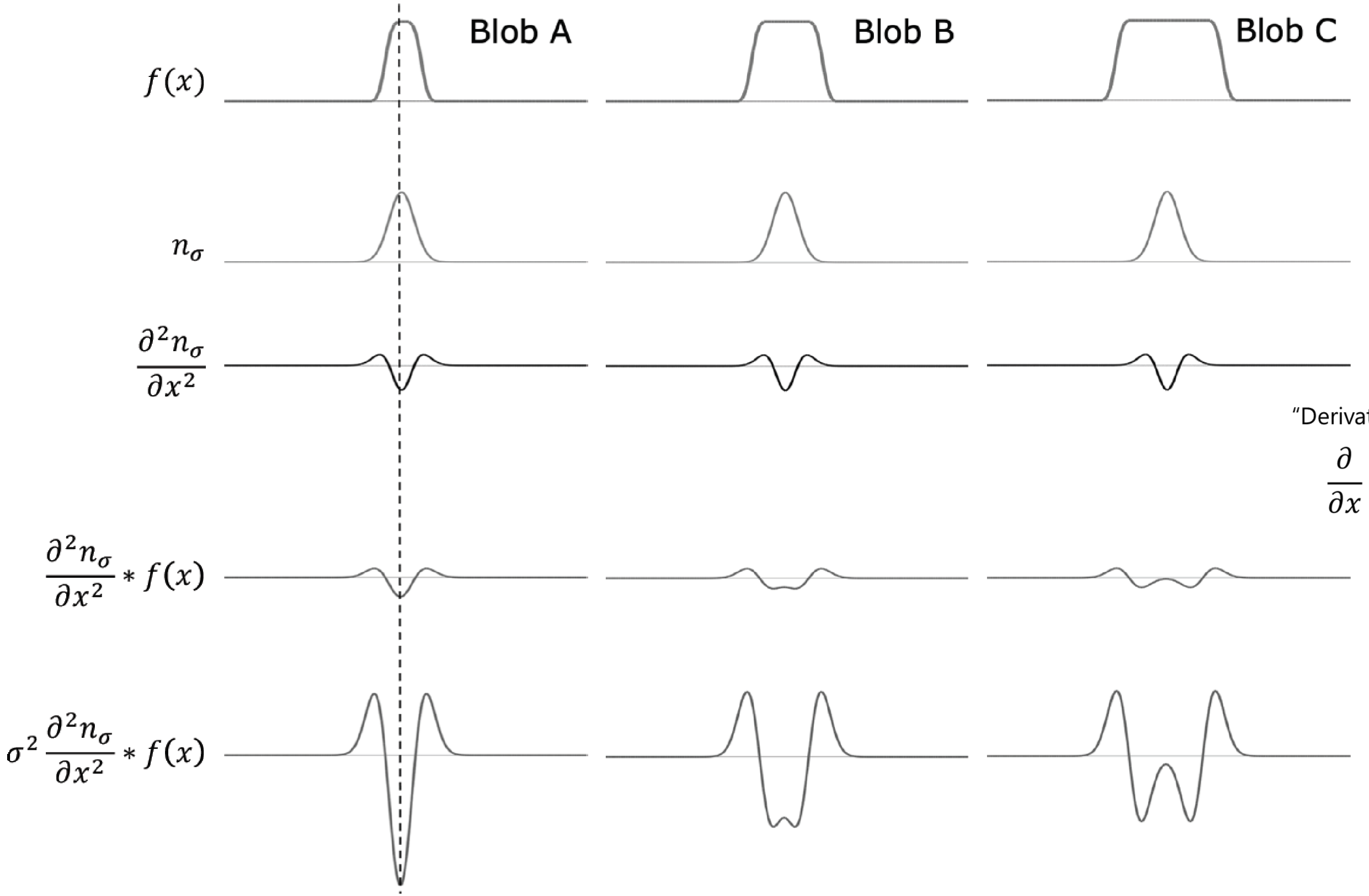
→ **Zero crossing** in **2nd** derivative of Gaussian denotes an **edge**.

1D Blobs



→ Examples of **1D** blob-like structures.

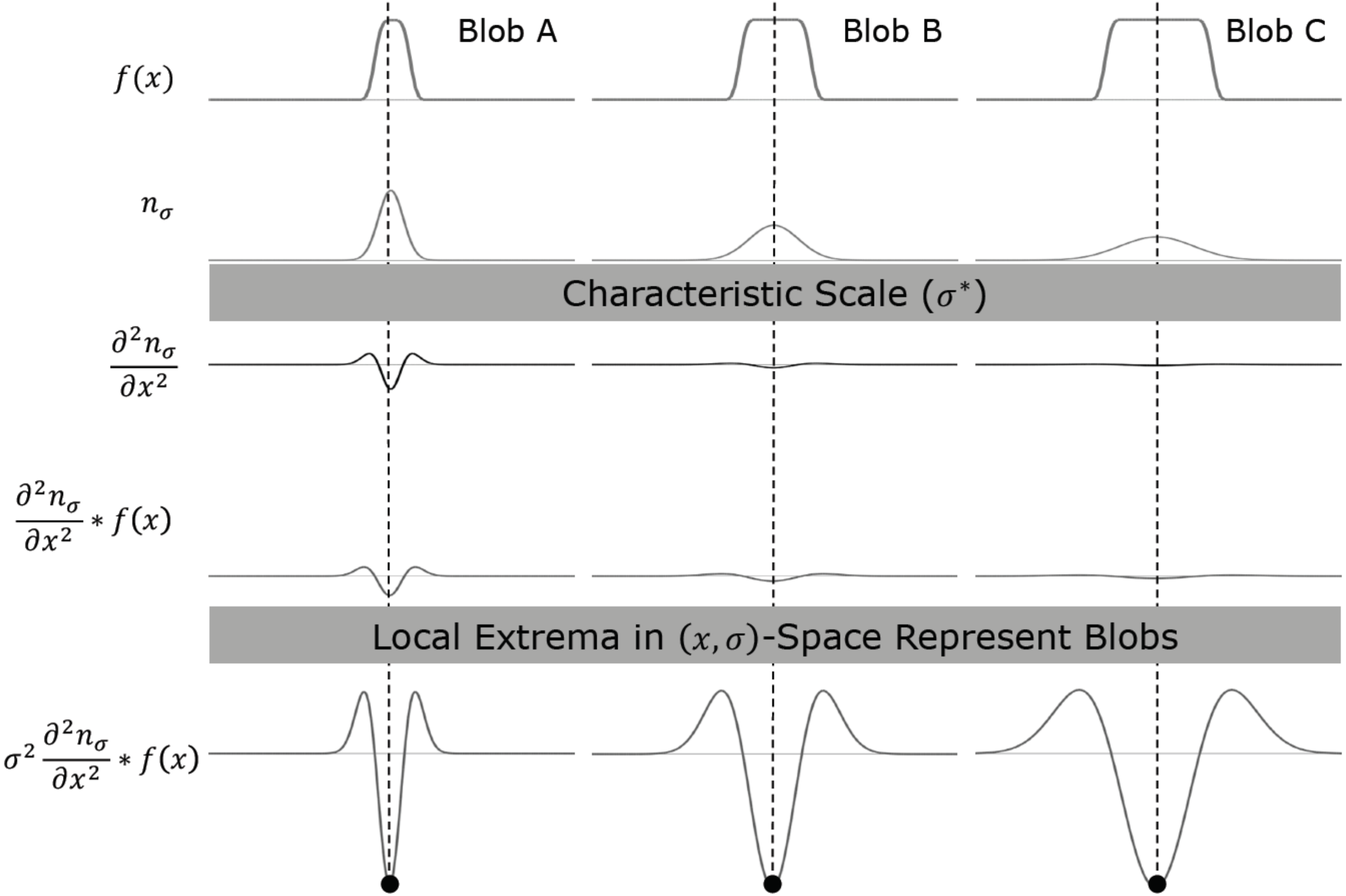
1D Blob and 2nd Derivative of Gaussian



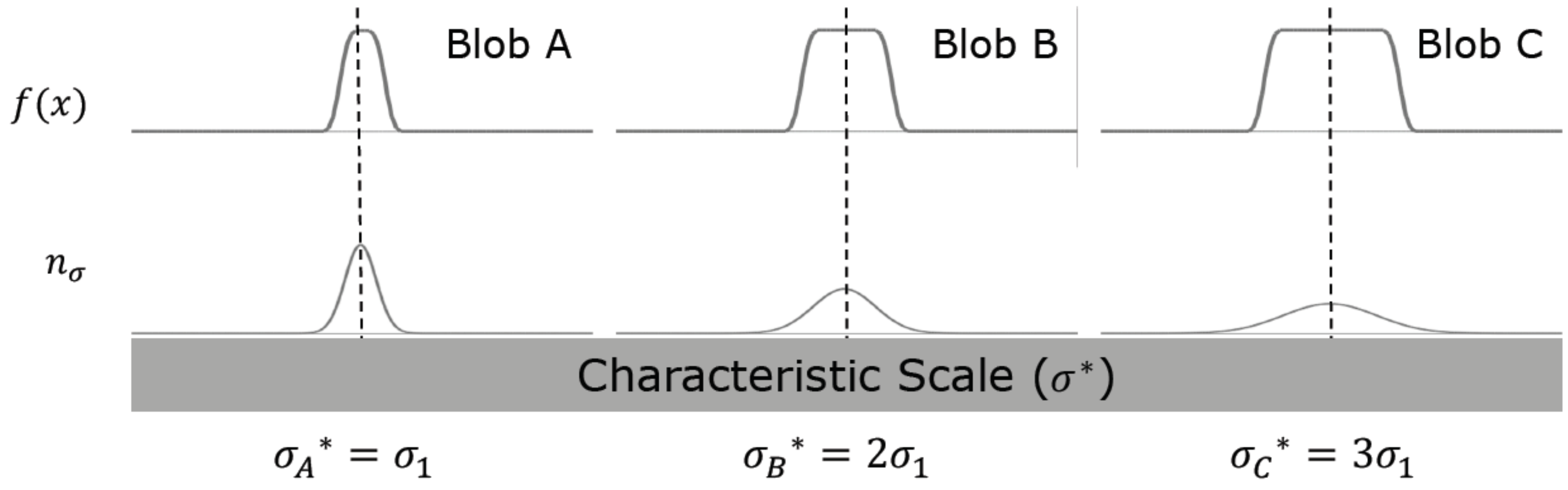
"Derivative Theorem of Convolution"

$$\frac{\partial}{\partial x}(h * f) = \left(\frac{\partial}{\partial x} h\right) * f$$

1D Blob and 2nd Derivative of Gaussian



1D Blob and 2nd Derivative of Gaussian



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

$$\boxed{\text{Characteristic Scale} \propto \text{Size of Blob}}$$

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

Summary of 1D Blob Detection

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 Blob Detection.

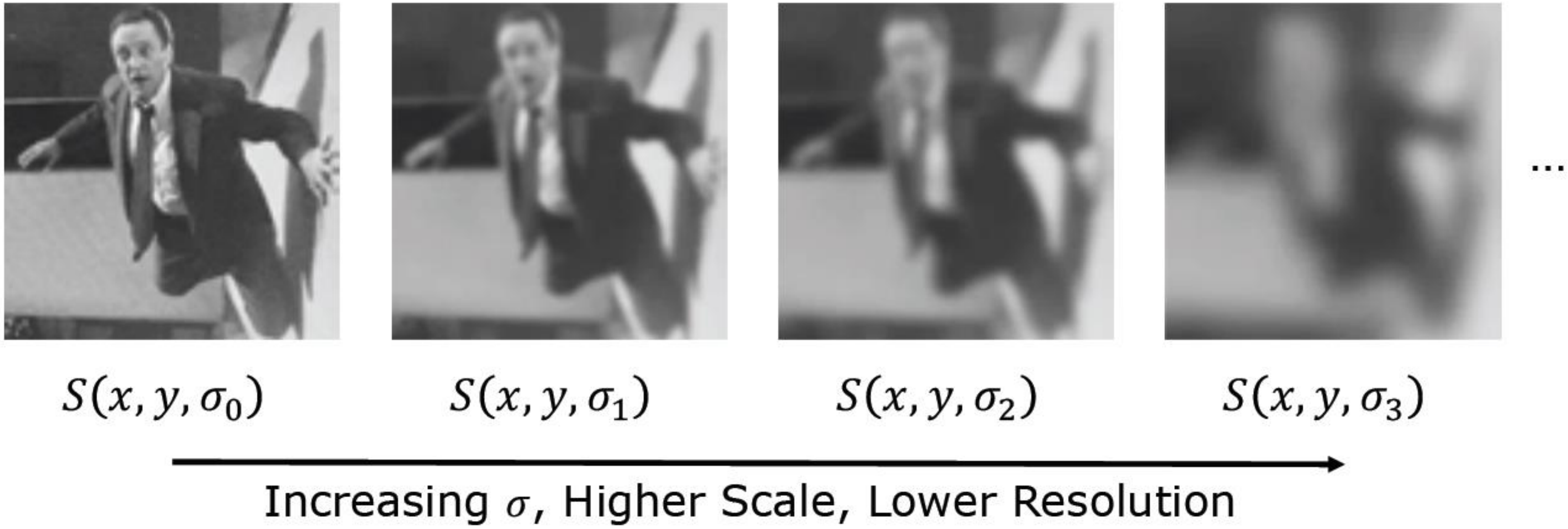
Laplacian | Gaussian | LoG | NLoG

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

n_σ | $\nabla^2 n_\sigma$ | $\sigma^2 \nabla^2 n_\sigma$

- Location of blobs given by **local extrema** after applying **NLoG** at **many scales**.

Scale Space



- Scale space: Stack created by filtering an image with Gaussian of different $\boldsymbol{\sigma}$.

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

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

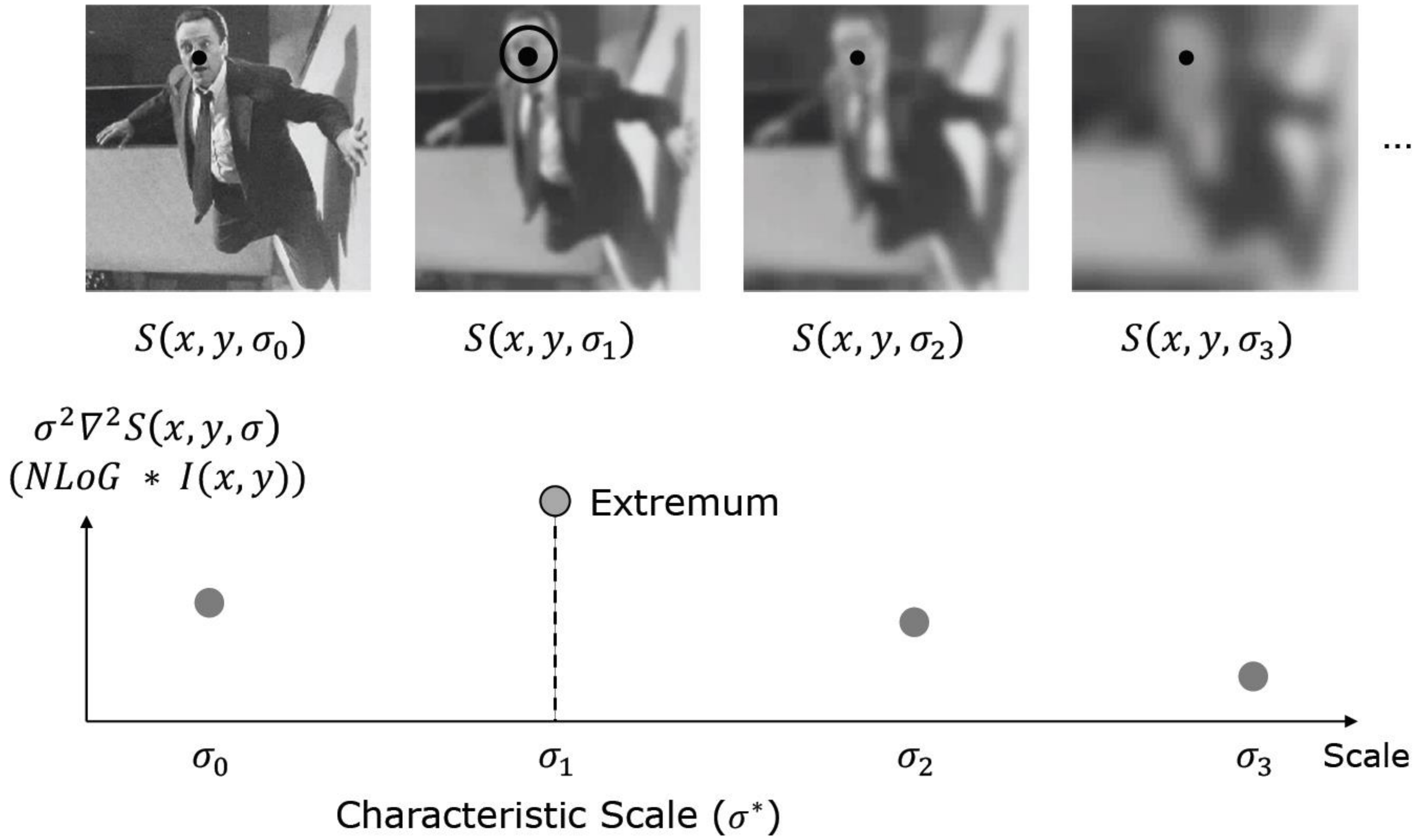
Selecting sigmas to generate the scale-space:

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

s: Constant multiplier

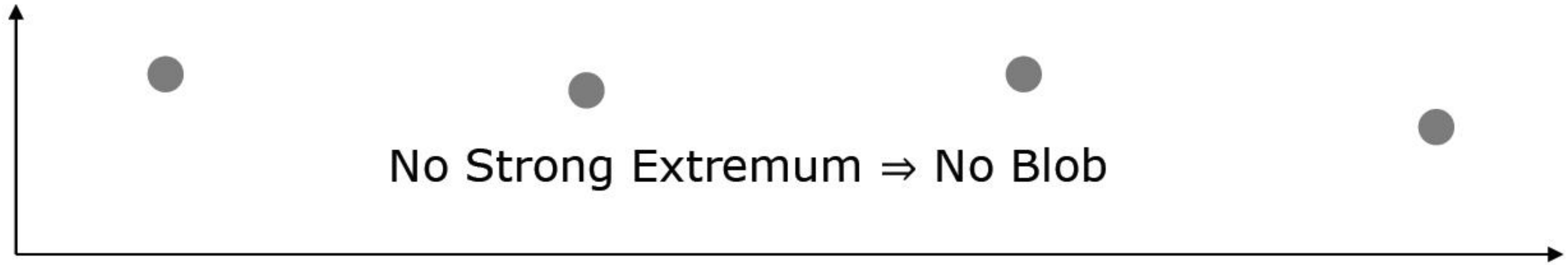
σ_0: Initial Scale

Blob Detection Using Local Extrema



Blob Detection Using Local Extrema

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



Summary of 2D Blob Detection

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

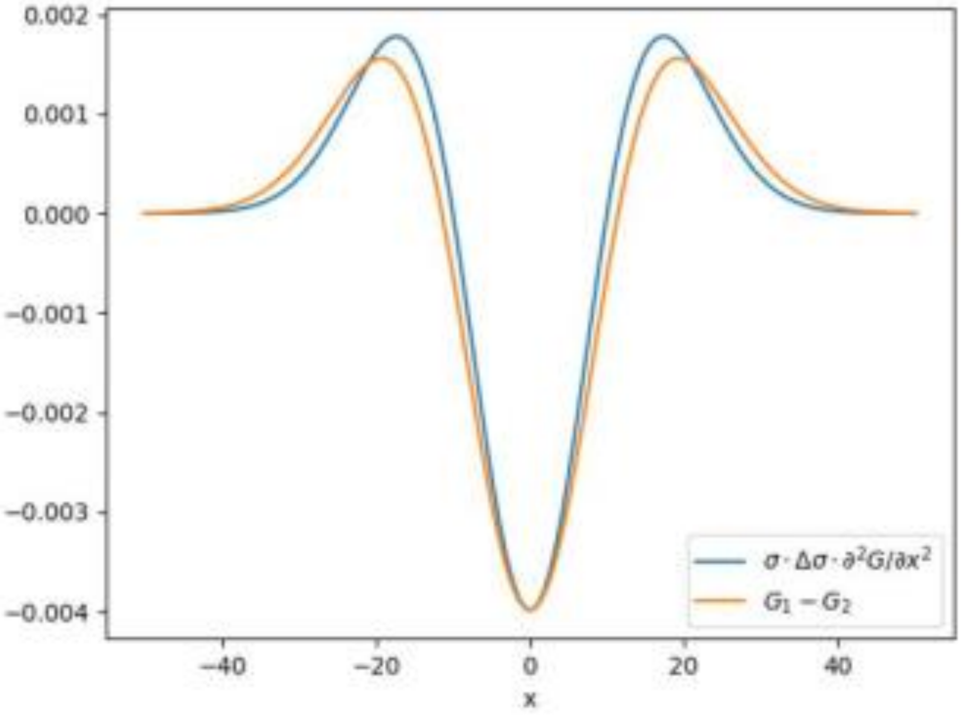
σ^*: Size of the blob

Korea Institute of Energy Technology

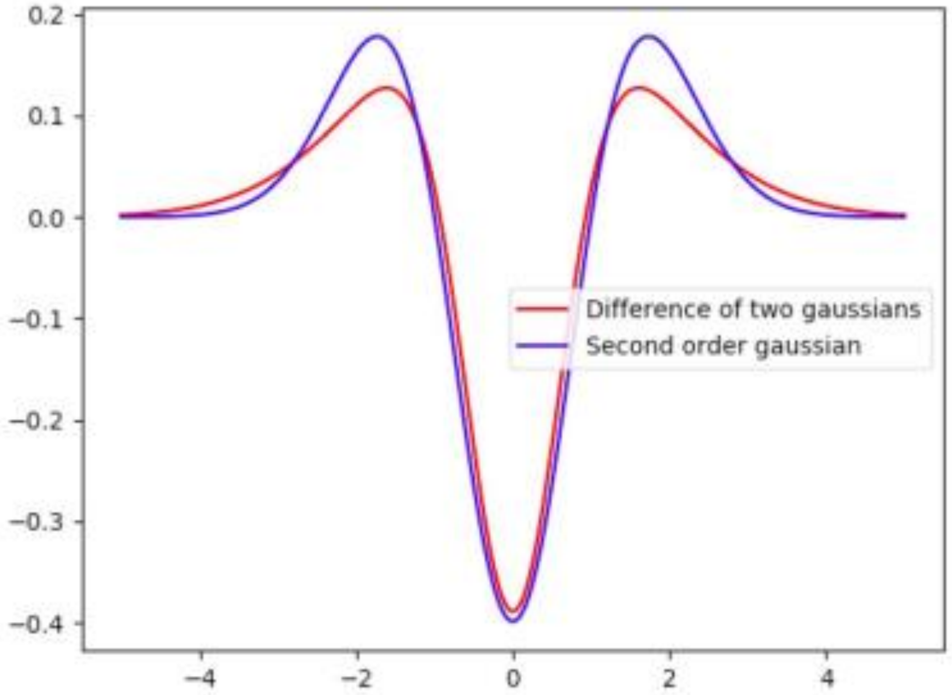
SIFT Detector

Fast NLoG Approximation: DoG

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

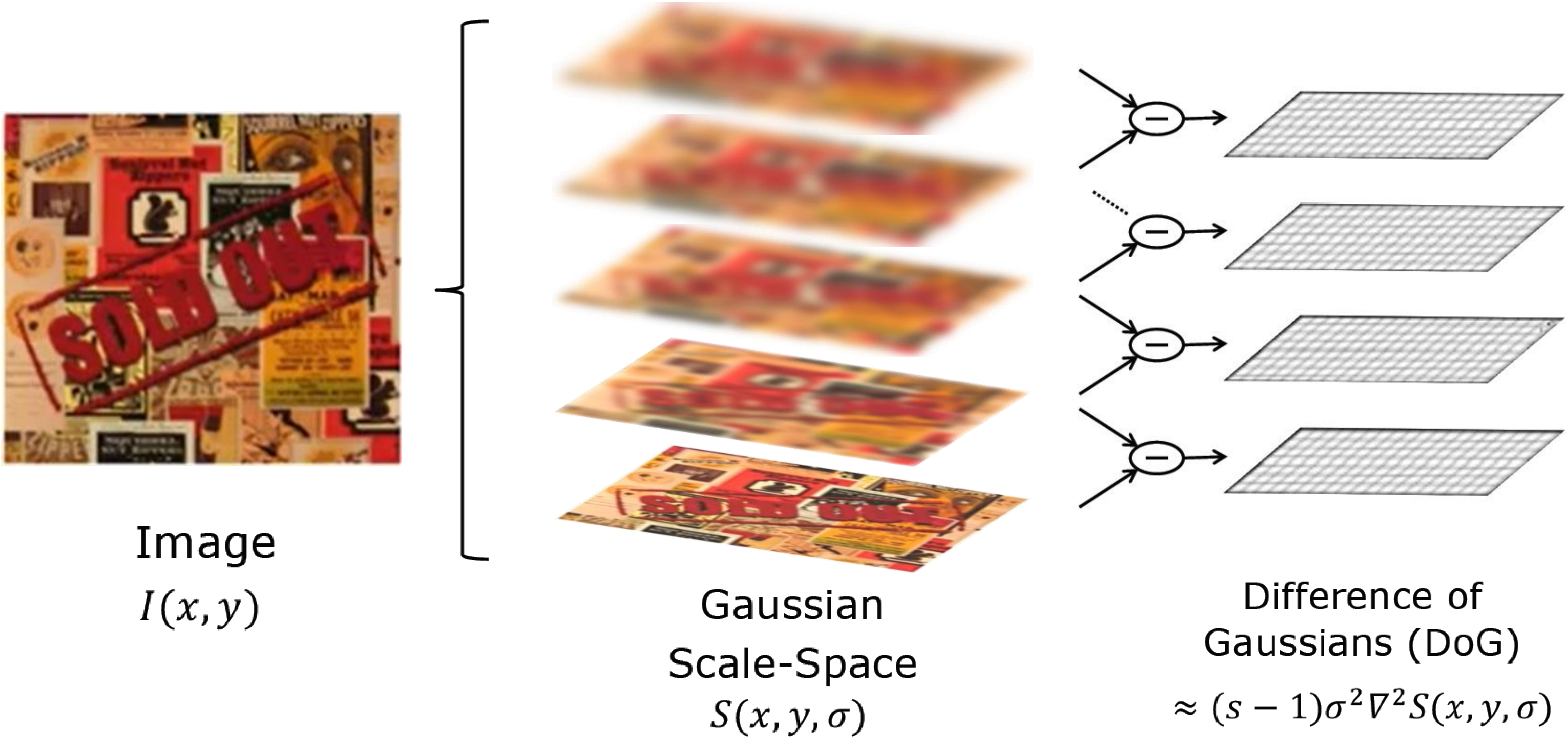


DoG approximation of LoG with $\sigma = 10$ and $\Delta\sigma = 1$

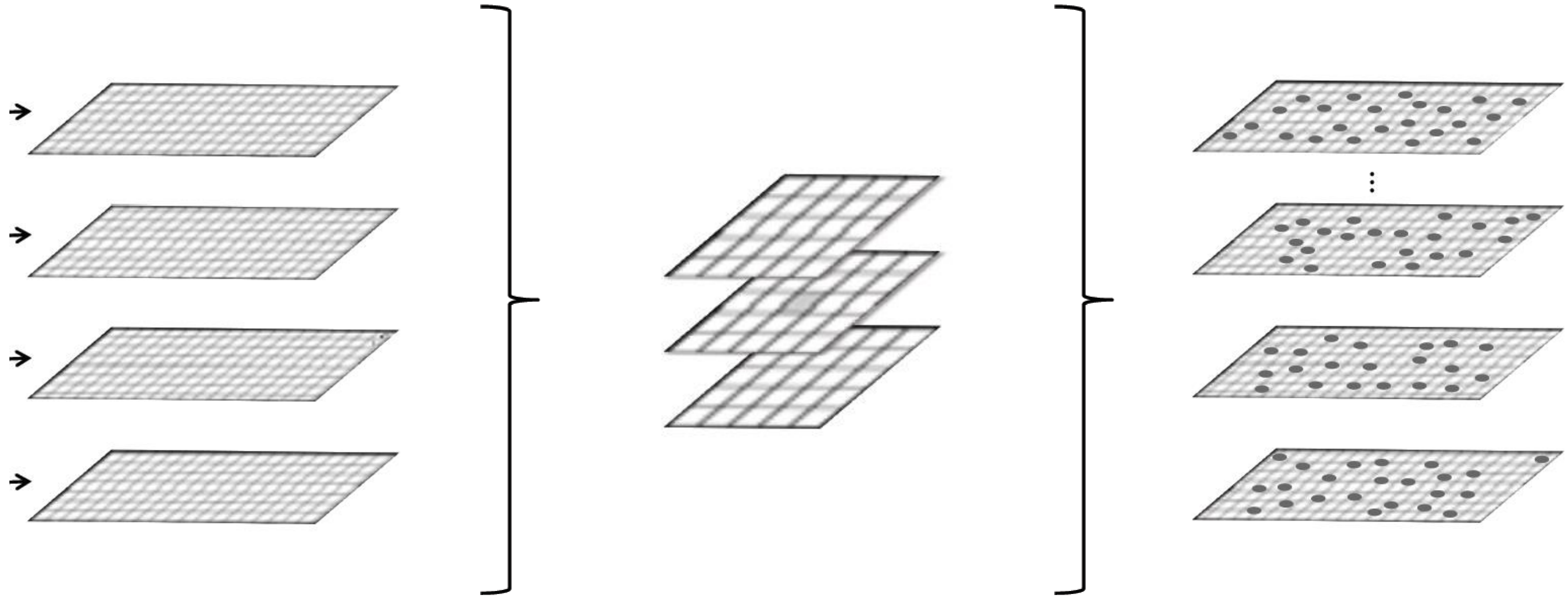


Sigma = 1; sigma1 = sigma / 1.6; sigma2 = sigma * 1.6

Extracting SIFT Interest Point



Extracting SIFT Interest Point



Difference of Gaussians (DoG)

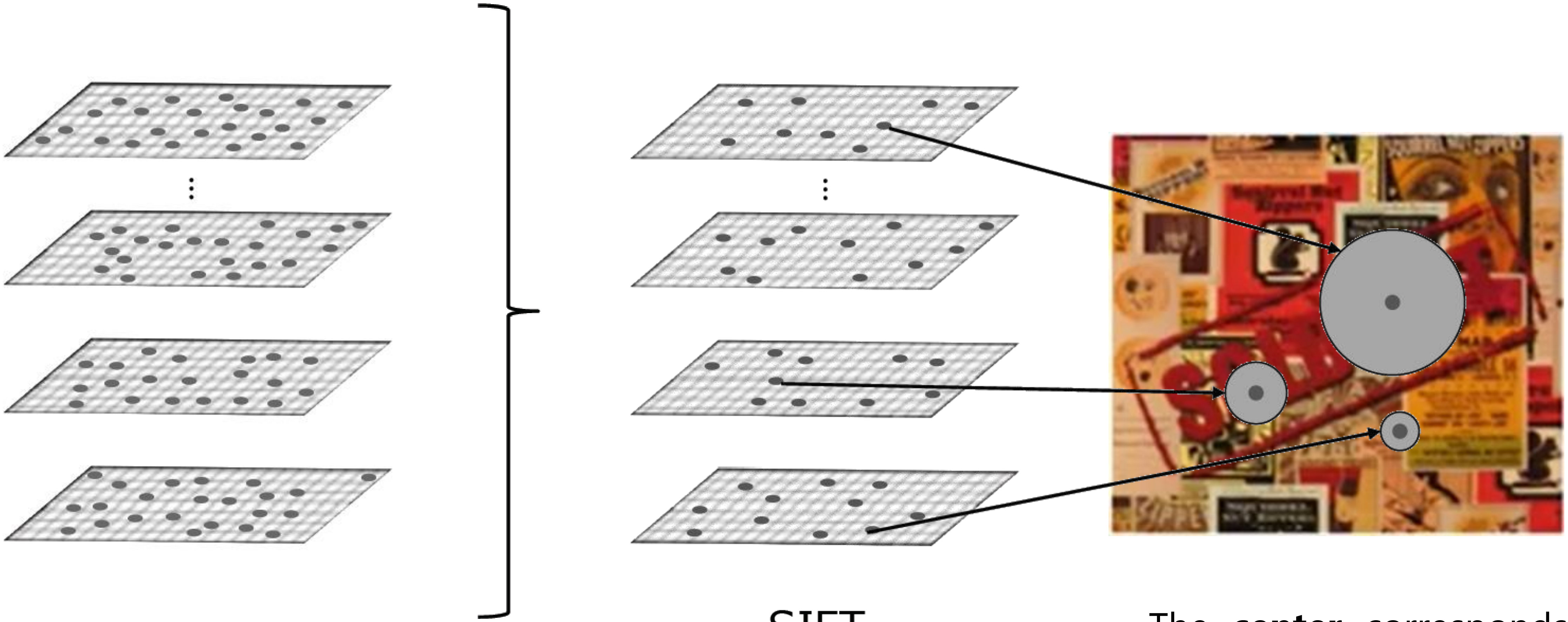
$\approx (s-1)\sigma^2 \nabla^2 S(x, y, \sigma)$

Find Extremum in every 3x3x3 grid

Interest Point Candidates

(includes weak extrema)

Extracting SIFT Interest Point



Interest Point Candidates

(includes weak extrema)

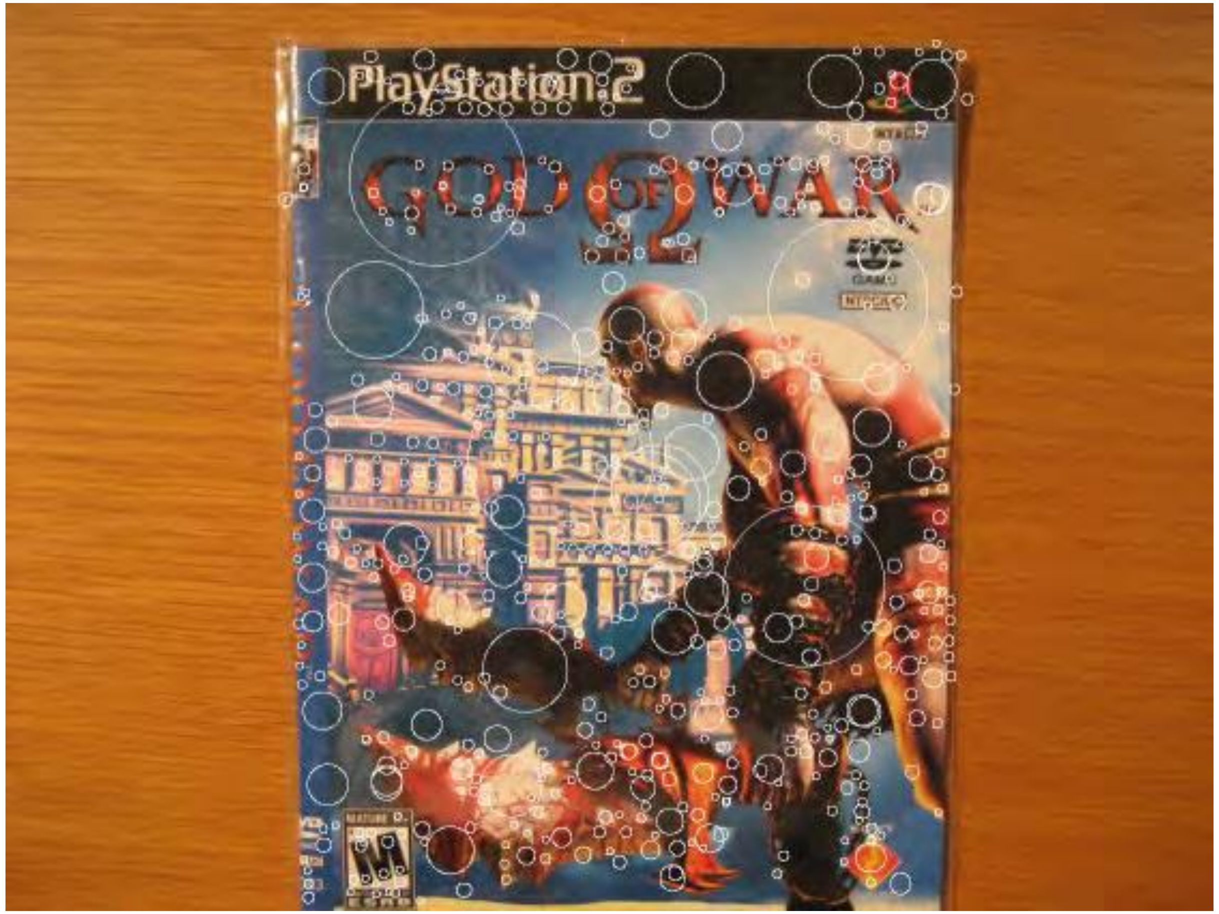
SIFT Interest Points

(after removing weak extrema)

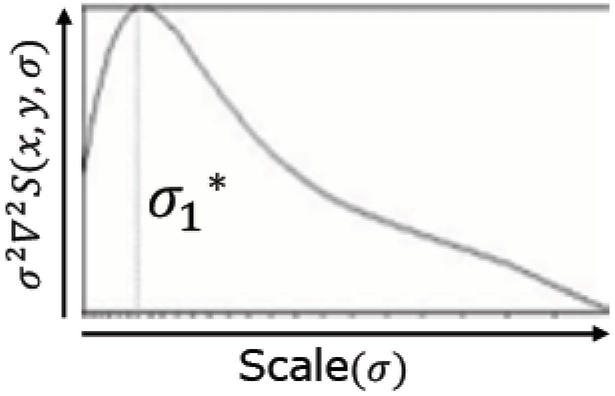
→ The **center** corresponds to the **location** of the feature.

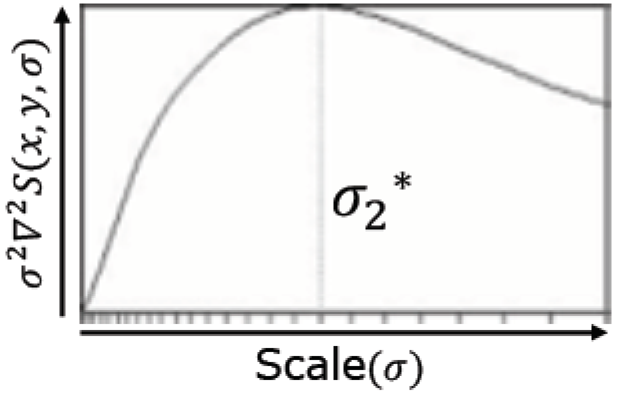
→ The **radius** is proportional to the **size** of the feature.

SIFT Detection Examples



SIFT Scale Invariance





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

Computing the Principal Orientation

Use the histogram of gradient directions

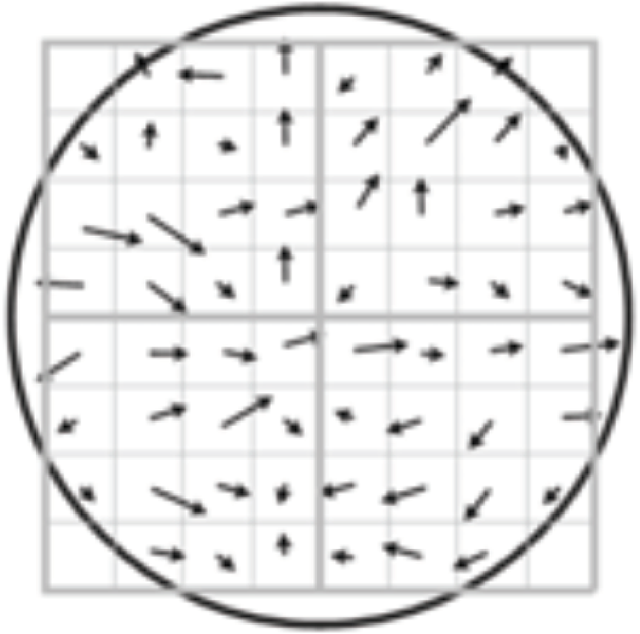
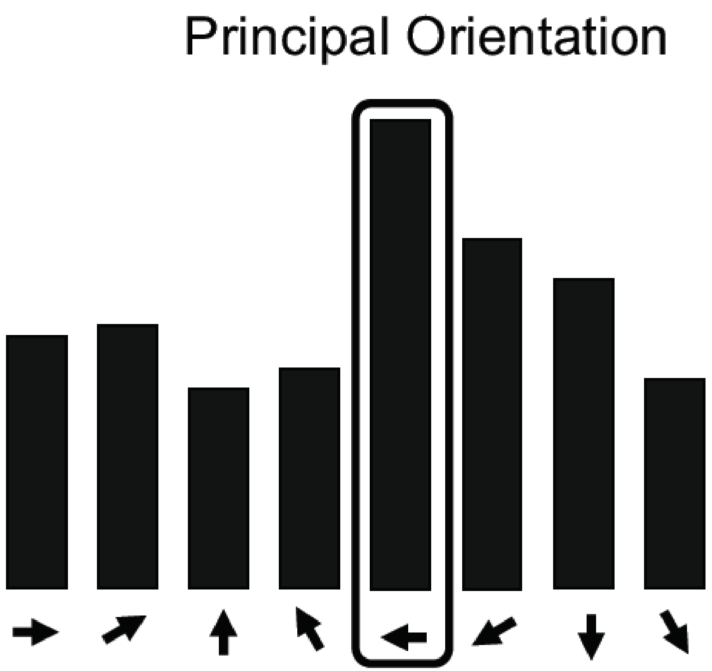


Image gradient directions

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



Choose the most
prominent gradient direction

SIFT Rotation Invariance

Use the principal orientation to undo rotation