KENTECH
Korea Institute of Energy Technology

Advanced Computer Vision
Week 05

Sep. 30, 2022
Seokju Lee

No Lectures Next Week (Oct. 4th & 7th)

1) **Instead**, we will have classes for reviewing **programming assignments**.

So far we have tried:

- `Image Processing Puzzle`
- `Camera Calibration (+ calibration of your mobile phone camera)`
- `Geometric Transformation`
- `Corner Detection (this week)`

Contents that were not possible due to lack of time will be covered next week.

2) Online Graduate Seminar (10/6) by **Dr. Hyojin Park (Qualcom AI Research @ San Diego)**

Title: "`Energy-Efficient AI for Image and Video Processing`"

Prepare at least **one question** and ask her. Summarize the Q&A and submit them to LMS.

Notice of Paper Reviews

Link: https://docs.google.com/spreadsheets/d/1S9z_QkqnSqy92P-fvhyggx-lJQwMPH1im3EvzNh3_cw/edit?usp=sharing

Rules:

1. 페이퍼는 최소 3편 이상을 리뷰(각 구역에서 1편 이상)
2. 페이퍼 포인트의 합은 최소 12포인트 이상을 원칙으로 함
3. 신청 가능 링크는 10/28 수요일 오후 5시 공지
4. 추가 페이퍼 리뷰를 원한다면 조율 가능(포인트 추가 가능)
5. 페이퍼 리스트는 업데이트 될 수 있음
6. 모든 페이퍼 리뷰는 발표 3일 전까지 개별 면담을 통해 검토 받는 것을 권장
7. 이해가 어려운 페이퍼의 경우 개별 면담 가능
8. 20분 영어발표 + 5분 한국어 Q&A

Notice of Paper Reviews

Topics	Papers	Conference	Point	Links	Reserve	Expected Date
CNN architectures	AlexNet	NIPS'12	3	https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf	jaiyong	10/17~
	VGG	ICLR'15	3	https://arxiv.org/abs/1409.1556	gwsur	
	ResNet	CVPR'16	3	https://arxiv.org/abs/1512.03385	yglee	
	Spatial Transformer Network	NIPS'15	3	https://arxiv.org/abs/1506.02025	sohee kim	
	Convolutional Bottleneck Attention Module	ECCV'18	4	https://arxiv.org/abs/1807.06521	Lee Tae Hong	
Semantic Segmentation	DeconvNet	CVPR'15	3	https://arxiv.org/abs/1505.04366	jaiyong	10/31~
	DeepLab-v2	T-PAMI'15	4	https://arxiv.org/abs/1606.00915		
	VPGNet	ICCV'17	4	https://github.com/SeokjuLee/VPGNet	yglee	
Deep generative models	Conditional Variational Auto-Encoder	NIPS'15	3	https://papers.nips.cc/paper/2015/hash/8d55a249e6baa5c06772297520da2051-Abstract.html	jaiyong	
	Pix2Pix	CVPR'17	4	https://phillipi.github.io/pix2pix/		
	Cycle-GAN	ICCV'17	4	https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix	Lee Tae Hong	
Representation learning	Context encoders	CVPR'16	4	https://arxiv.org/abs/1604.07379		
	Damaged puzzle	WACV'18	5	https://arxiv.org/abs/1802.01880	gwsur	
	Contrastive learning	ICML'20	4	https://arxiv.org/abs/2002.05709	sohee kim	
Optical flow	UnFlow	AAAI'18	4	https://arxiv.org/abs/1711.07837	jaiyong	11/21~
Stereo matching	PSMNet	CVPR'18	4	https://github.com/JiaRenChang/PSMNet		
Structure-from-Motion	MonoDepth2	ICCV'19	4	https://github.com/nianticlabs/monodepth2	gwsur	
	Insta-DM	AAAI'21	5	https://github.com/SeokjuLee/Insta-DM		
	Unsupervised SfM in Dynamic Scenes	CoRL'20	5	https://arxiv.org/abs/2010.16404	sohee kim	
	Neural Ray Surfaces	3DV'20	5	https://arxiv.org/abs/2008.06630		
NeRF	Neural Scene Flow Fields	CVPR'21	5	https://www.cs.cornell.edu/~zl548/NSFF/	yglee	
	Self-Calibrating NeRF	ICCV'21	5	https://github.com/POSTECH-CVLab/SCNeRF		

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

KENTECH
Korea Institute of Energy Technology

Image Features

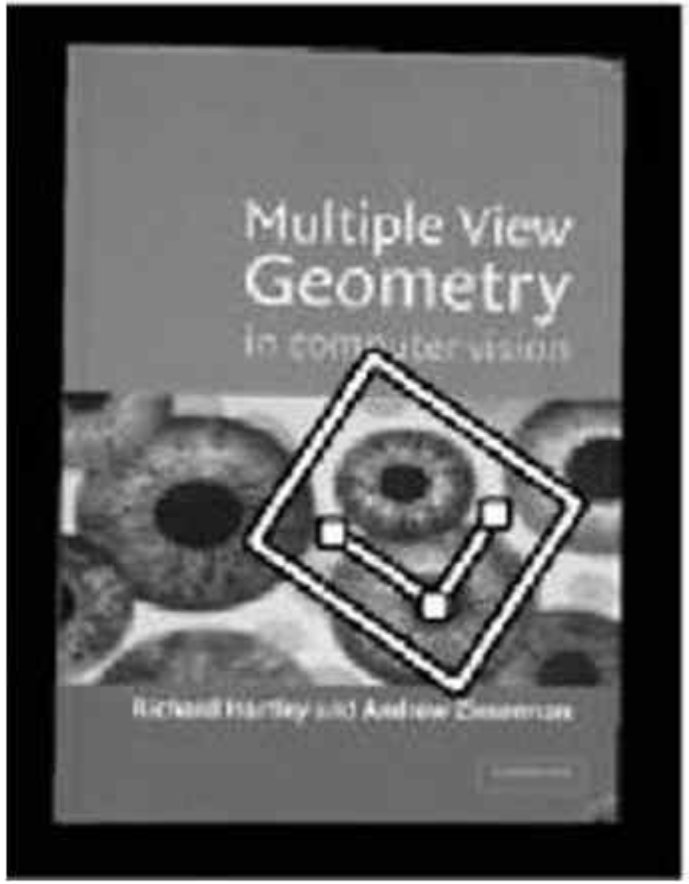
Why Detecting Image Features?

✓ Feature points are used for:

- Image alignment (*e.g.*, image stitching, video stabilization)
- 3D reconstruction
- Motion tracking
- Object recognition
- Indexing & database retrieval
- Robot navigation (*e.g.*, SLAM)
- etc.

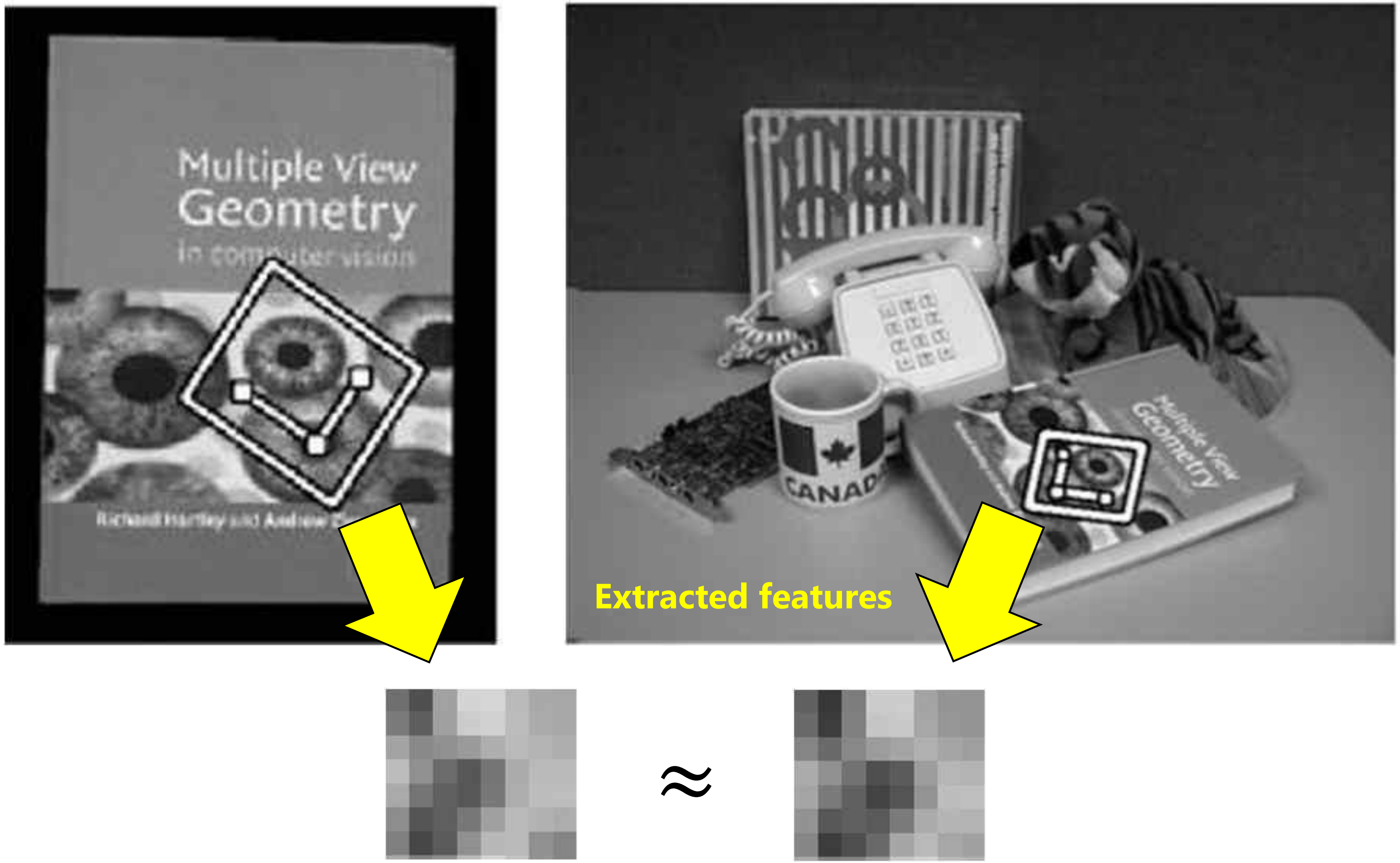
Pyimagesearch

Image Matching



Slide by R. Szeliski.

Image Matching



Slide by R. Szeliski.

Salient Features

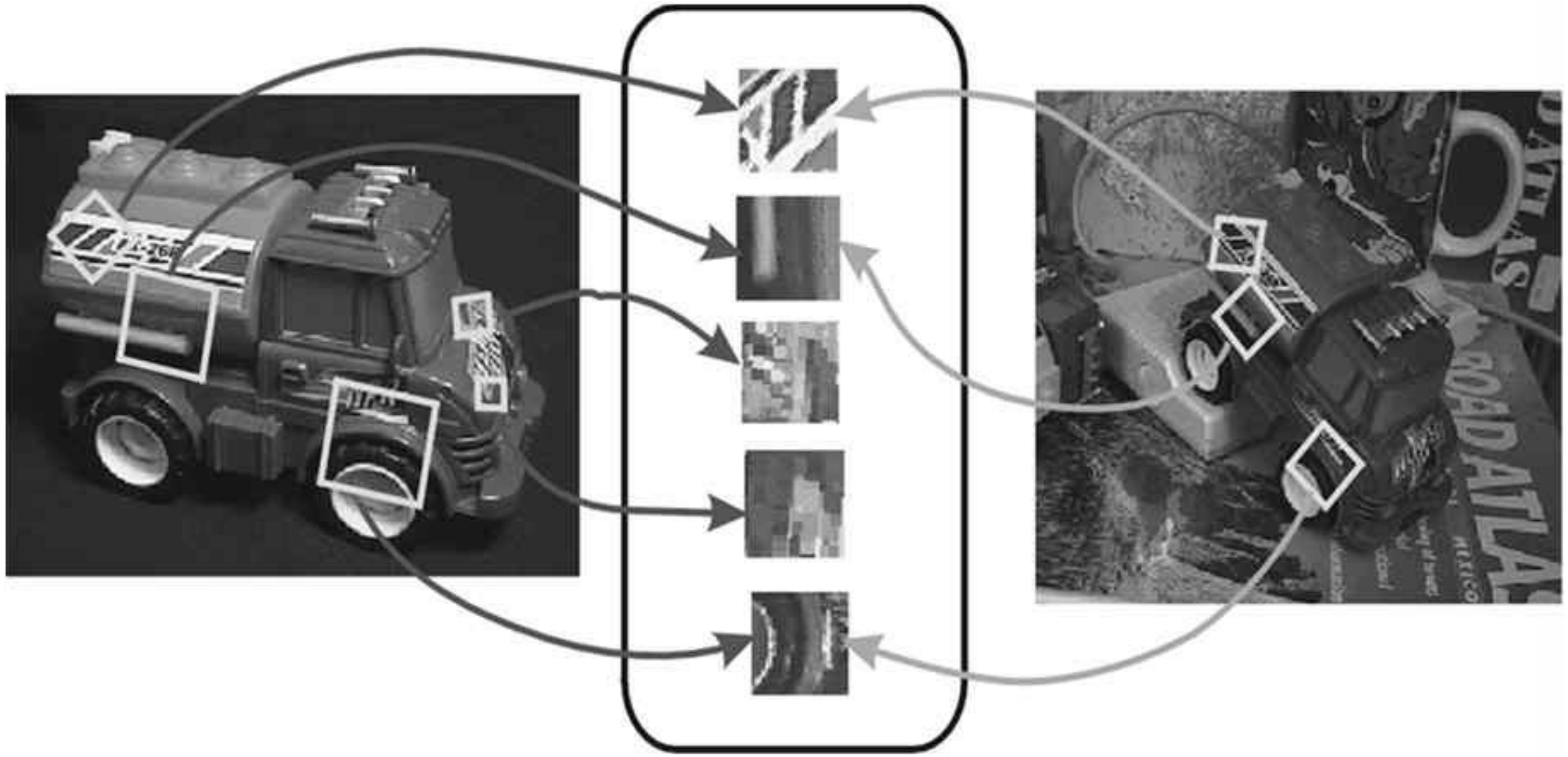
- ✓ **Generic** features:
  - Independent of the lens and the CCD.
  - Independent of the lighting conditions.
  - Independent of the pose and scale.

- ✓ The human visual system can interpret images using a **small amount** of feature (e.g., **edge** and **corner**) data.

- ✓ Two main issues:
  - What good features that show **robustness** independent of **variations**?
  - How can we **automatically** and **efficiently** extract features in images?

Invariant Local Features

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



Slide by D. Lowe.

KENTECH
Korea Institute of Energy Technology

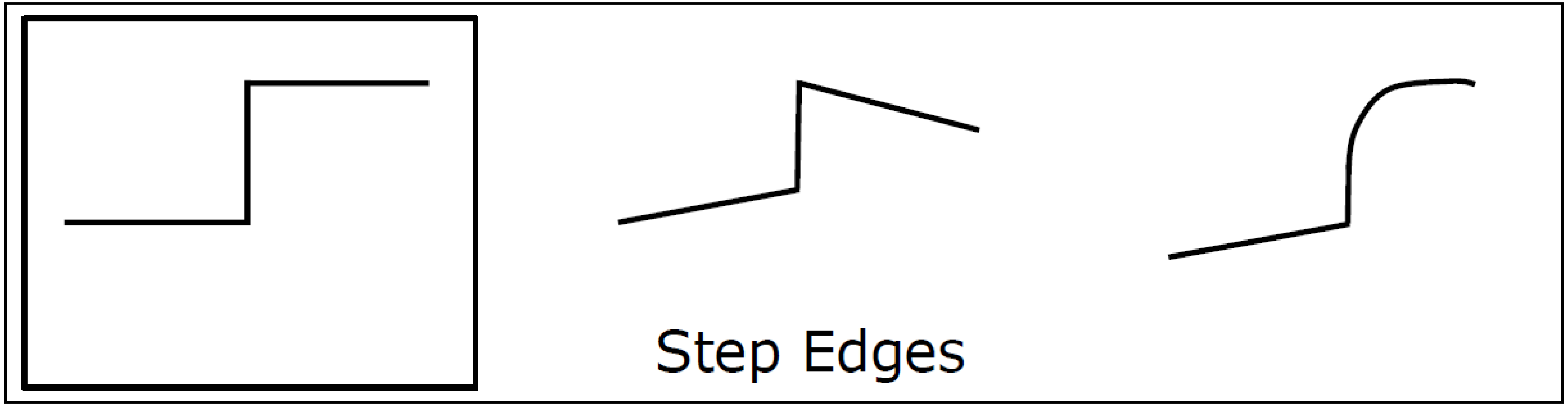
Edges

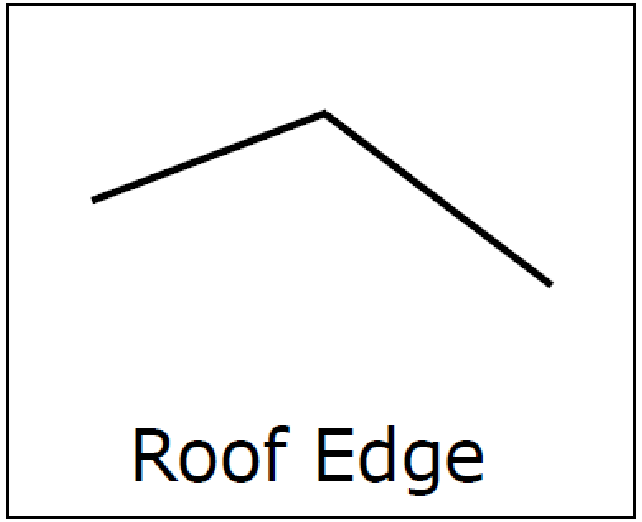
Edges

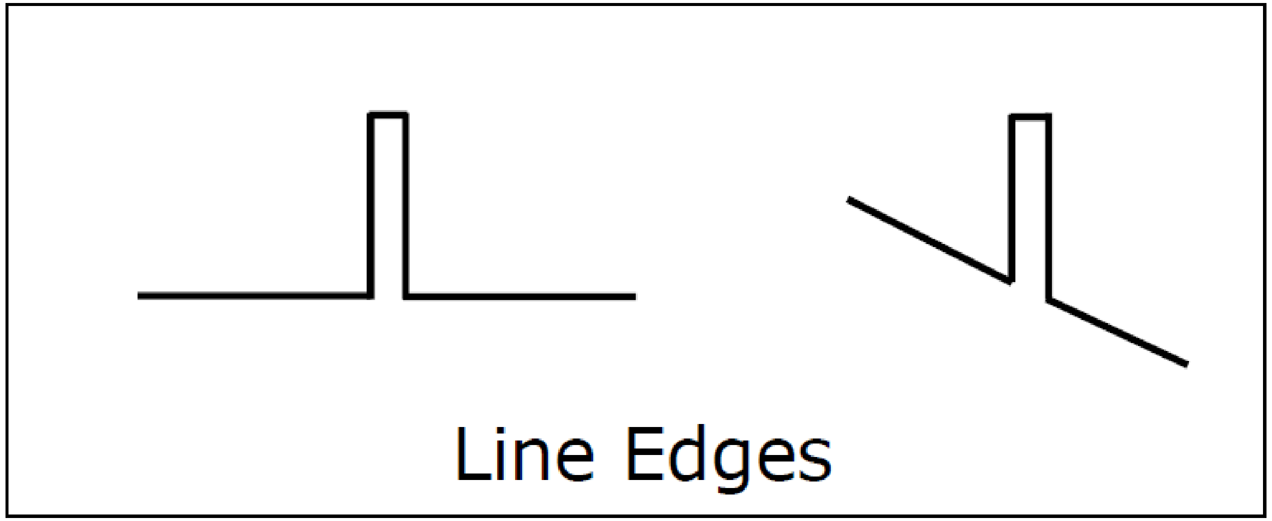
- ✓ Motivation
  - Sharp changes in image intensity (edges) are a **key** indicator of image content

- ✓ Why important?
  - Extracting meaningful edges from images amounts to a **dramatic reduction** in the amount of data.

Types of Edges





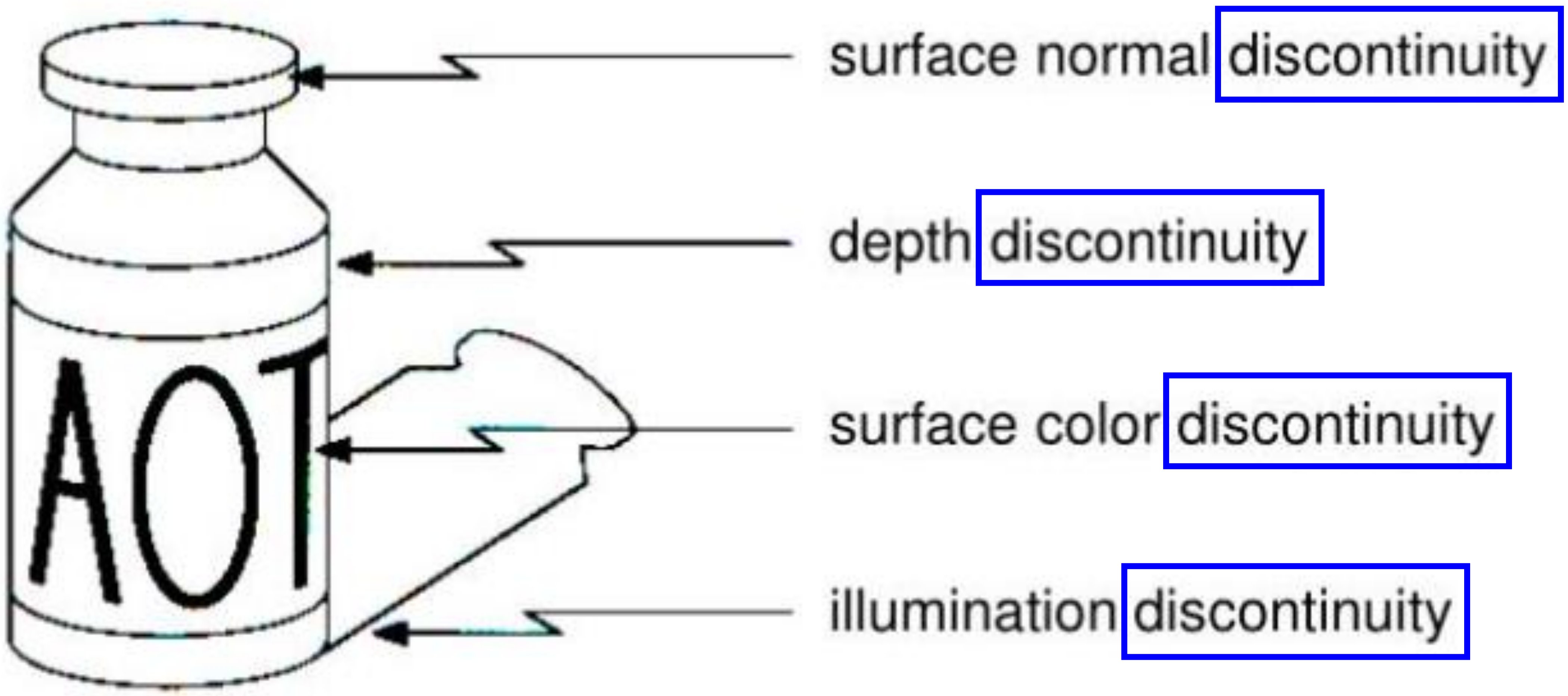


Edge Detector

- ✓ We want an edge operator that **produces**:
  - Edge position
  - Edge magnitude (strength)
  - Edge orientation (direction)

- ✓ Performance requirements:
  - High detection rate
  - Good localization
  - Low noise sensitivity

What Causes Edges?



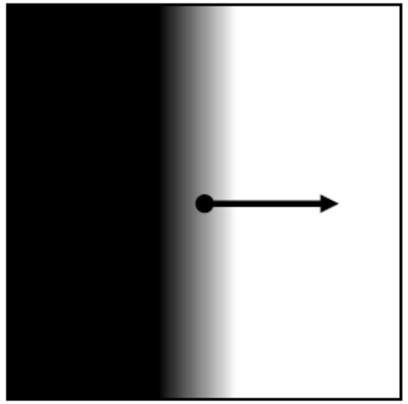
→ We need differentiation to find discontinuity!

Gradient (∇)

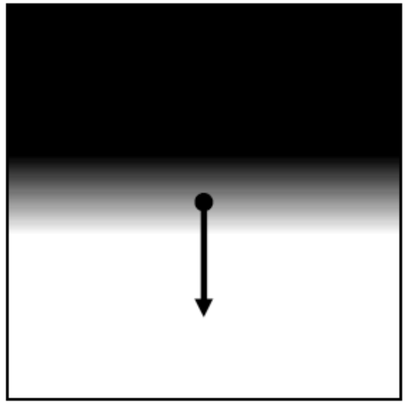
- ✓ **Gradient** (partial derivatives) represents the direction of most rapid change in intensity:

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

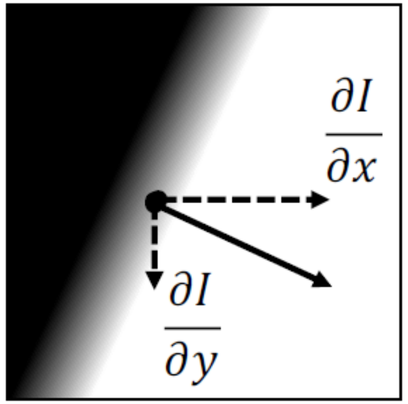
Pronounced as "Del I"



$$\nabla I = \left[\frac{\partial I}{\partial x}, 0\right]$$



$$\nabla I = \left[0, \frac{\partial I}{\partial y}\right]$$

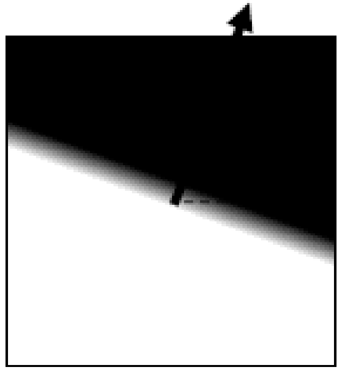


$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right]$$

Gradient (∇) as Edge Detector

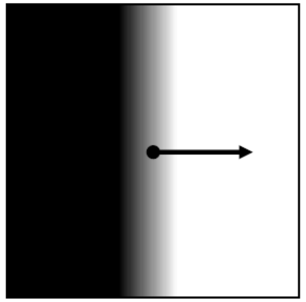
Gradient Magnitude $S = \|\nabla I\| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$

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

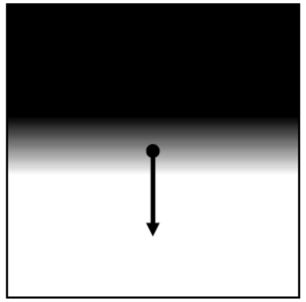


Differentiating the Image → Convolution

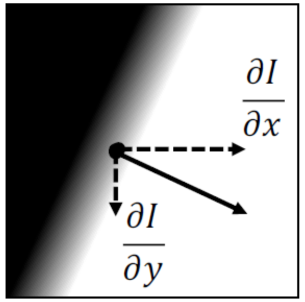
- ✓ Why **differentiate**? The derivative tells us about how **sharp** an edge is.



$$\nabla I = \left[\frac{\partial I}{\partial x}, 0\right]$$



$$\nabla I = \left[0, \frac{\partial I}{\partial y}\right]$$



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

- ✓ Differentiation can be approximated by **finite differences**…

- ✓ … which can be implemented as a **convolution** filter!

Slide from UC Davis

Differentiating the Image → Convolution

$$\frac{df(x)}{dx} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

$$\frac{df(x)}{dx} \cong \frac{f(x+1) - f(x-1)}{2}$$

→ Convolve with:

-1	**0**	**1**

Or:

$$\Delta_x I = I(x,y) - I(x-1,y)$$

$$\Delta_y I = I(x,y) - I(x,y-1)$$

Slide from UC Davis

Comparing Gradient Operators

Gradient	Roberts	Prewitt	Sobel (3x3)	Sobel (5x5)
$\frac{\partial I}{\partial x}$	$\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}$	$\begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix}$	$\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$	$\begin{matrix} -1 & -2 & 0 & 2 & 1 \\ -2 & -3 & 0 & 3 & 2 \\ -3 & -5 & 0 & 5 & 3 \\ -2 & -3 & 0 & 3 & 2 \\ -1 & -2 & 0 & 2 & 1 \end{matrix}$
$\frac{\partial I}{\partial y}$	$\begin{matrix} 1 & 0 \\ 0 & -1 \end{matrix}$	$\begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix}$	$\begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$	$\begin{matrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 3 & 5 & 3 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -3 & -5 & -3 & -2 \\ -1 & -2 & -3 & -2 & -1 \end{matrix}$

Good Localization
Noise Sensitive
Poor Detection

Poor Localization
Less Noise Sensitive
Good Detection

Gradient Using Sobel Filter

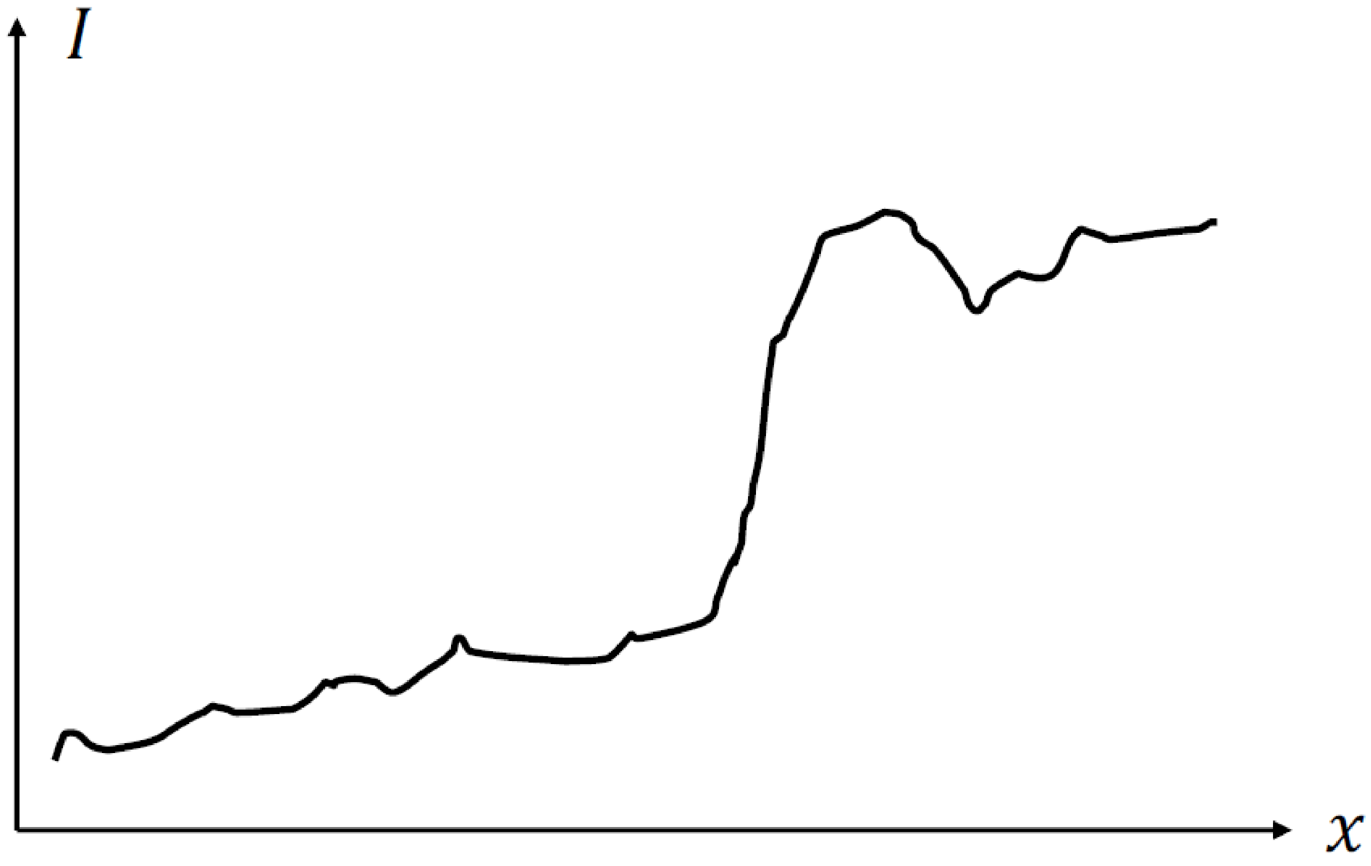
Image (I)

$\partial I / \partial x$

$\partial I / \partial y$

Gradient Magnitude

Real Edges

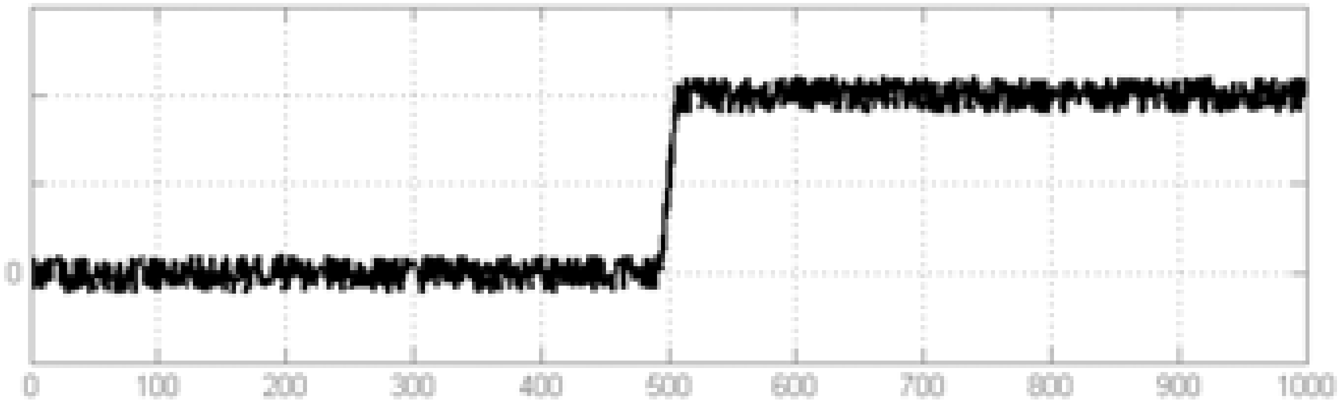


Problems: Noisy Images and Discrete Images

Smoothing the Image

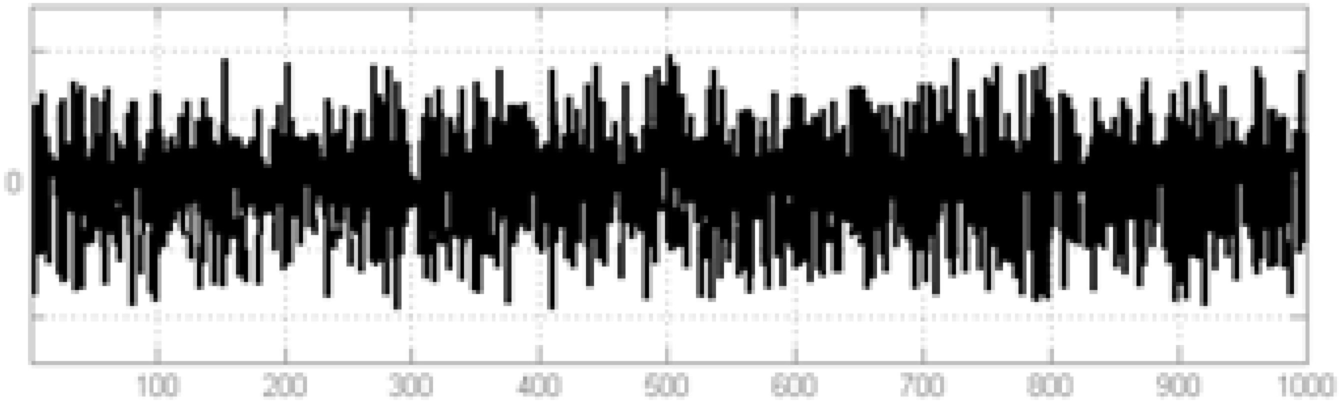
✓ We generally want to **smooth** the image to get rid of **noise** that causes **false** edge detections.

$f(x)$



← Noise

$\dfrac{df(x)}{dx}$

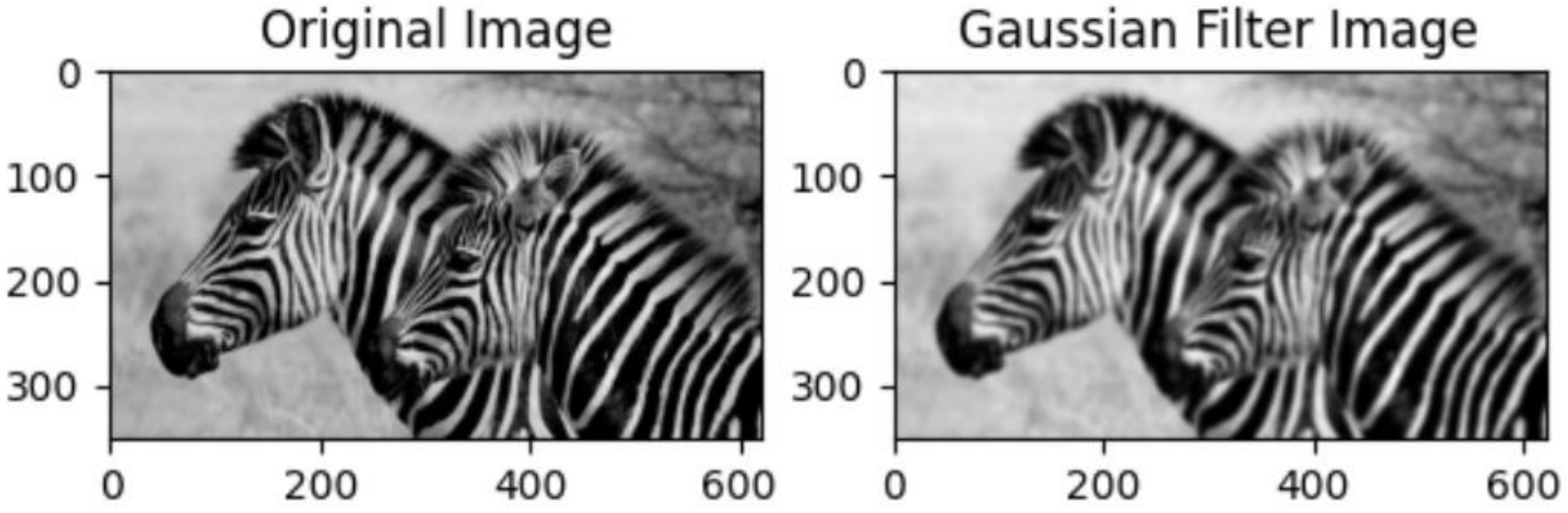


→ Where is the edge?

Slide from UC Davis

Smoothing the Image

✓ Gaussian filter for smoothing



Smoothing + Detecting Edge: Derivative of Gaussian Filter

- ✓ If you want to **smooth** with a gaussian, then **differentiate**, this is **equivalent** to <u>convolving with a derivative of gaussian</u>.

  "Derivative Theorem of Convolution"

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

Slide from UC Davis

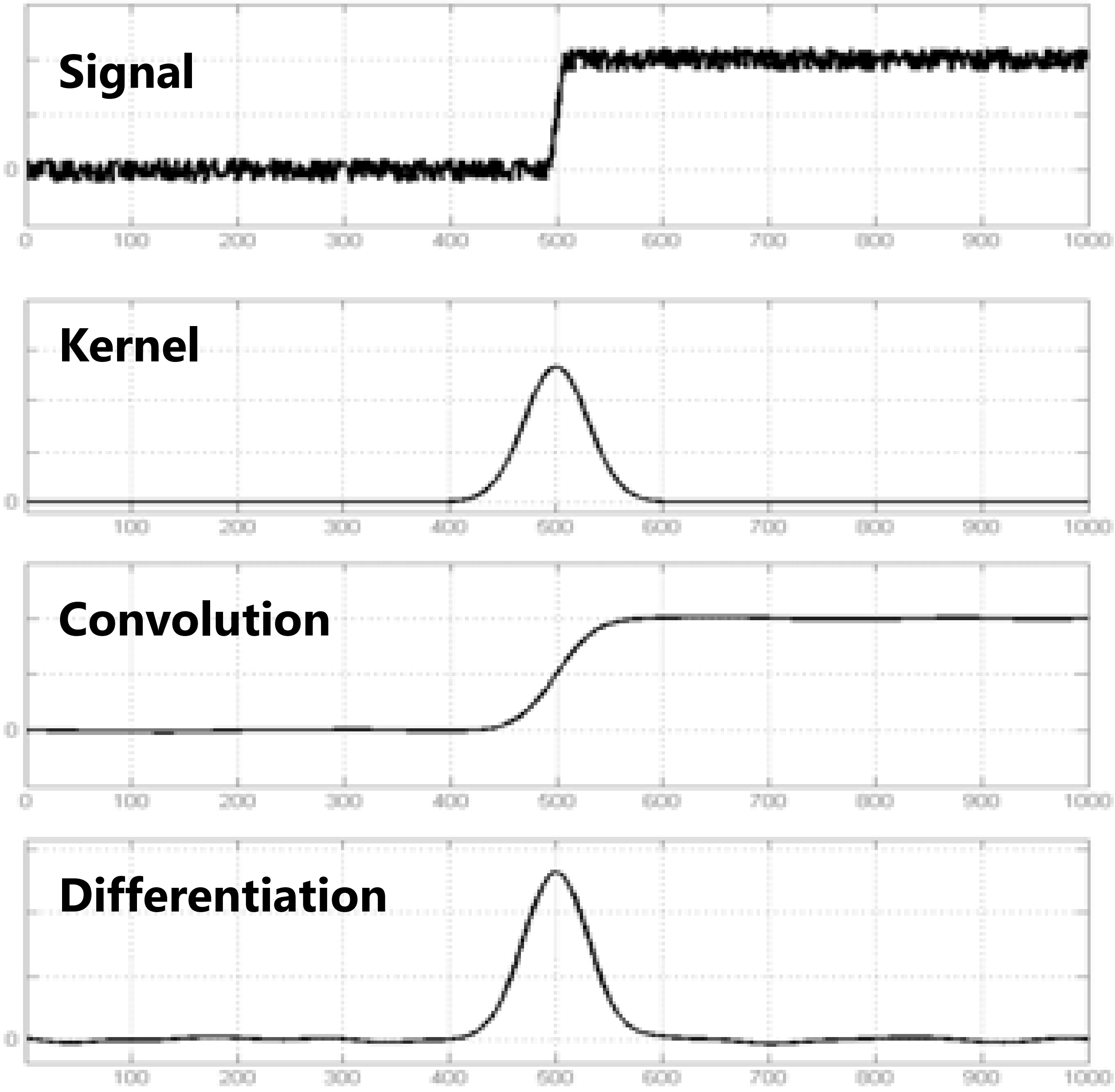
Example

f

h

$h * f$

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



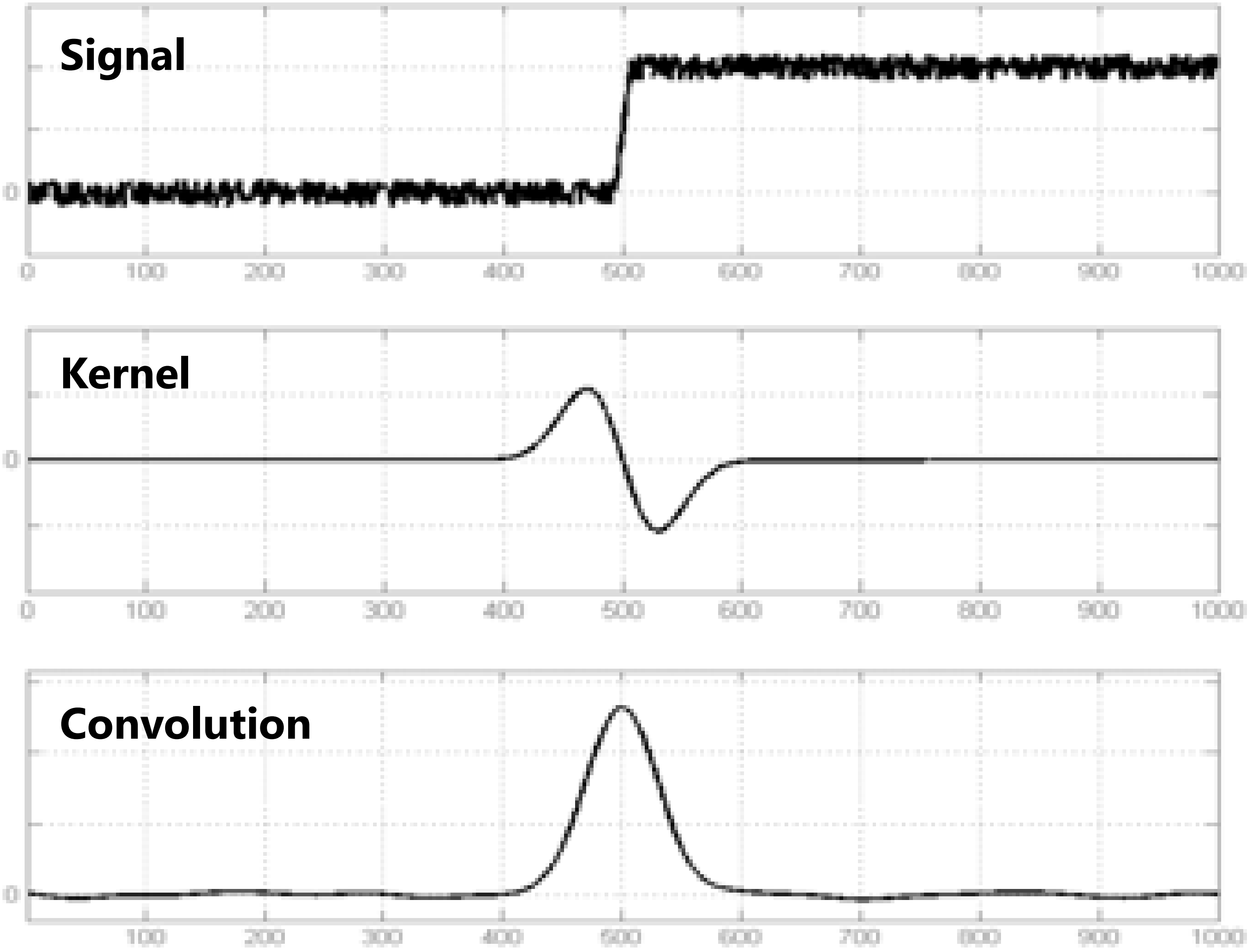
Slide from UC Davis

Example

$$f$$

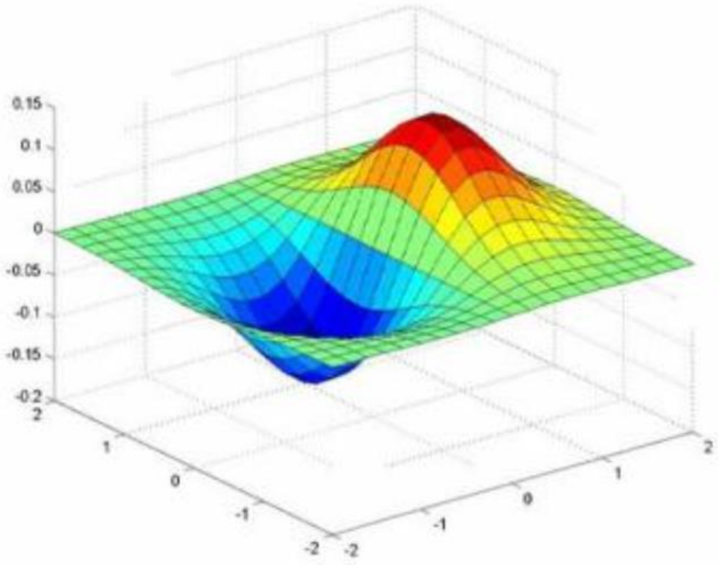
$$\frac{\partial}{\partial x}h$$

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

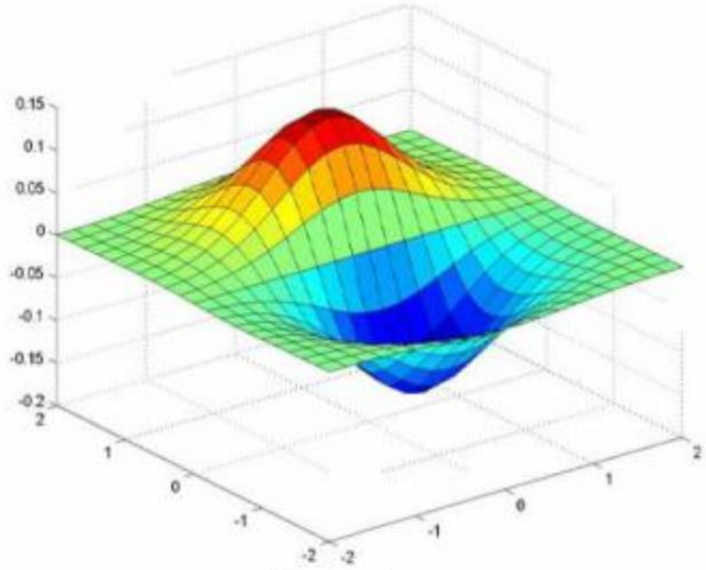


Slide from UC Davis

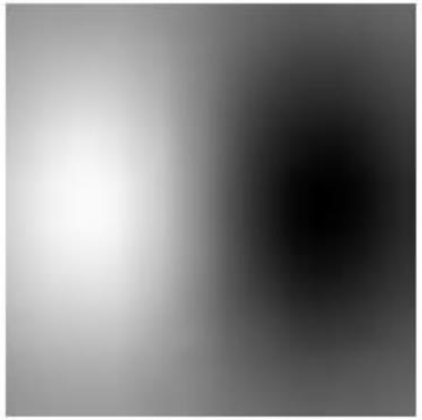
Derivative of Gaussian Filter: Visualization

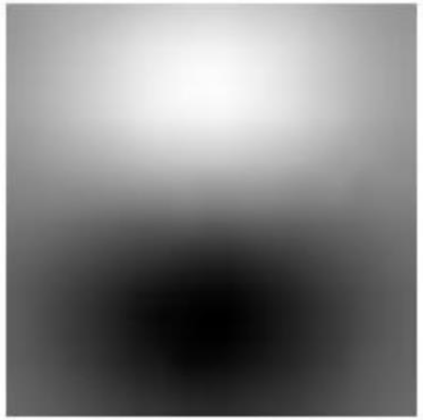


x-direction

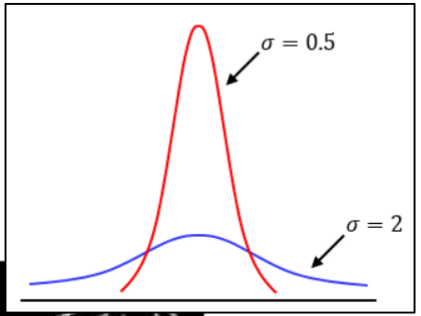


y-direction





Canny Edge Detector



✓ **The most widely used** edge detector in computer vision [1]

original

Canny with $\sigma = 1$

→ fine features

Canny with $\sigma = 2$

→ Large scale edges

✓ Effects of σ (Gaussian kernel spread/size) → 표준편차

[1] J. Canny, A Computational Approach to Edge Detection, IEEE T-PAMI 1986.

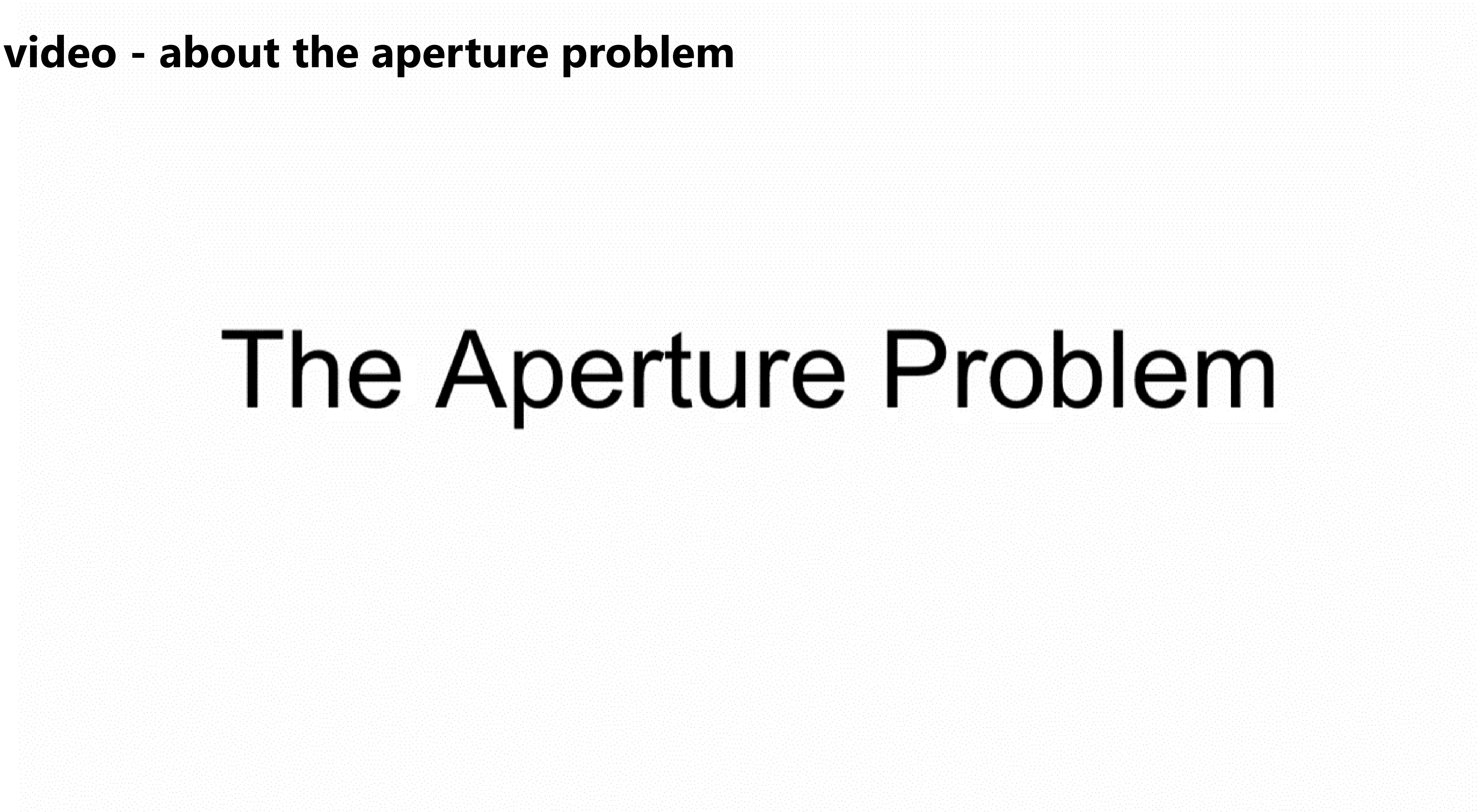
Parts of slides are by Prof. In So Kweon and Prof. Shree Nayar

KENTECH
Korea Institute of Energy Technology

Corners

Corners

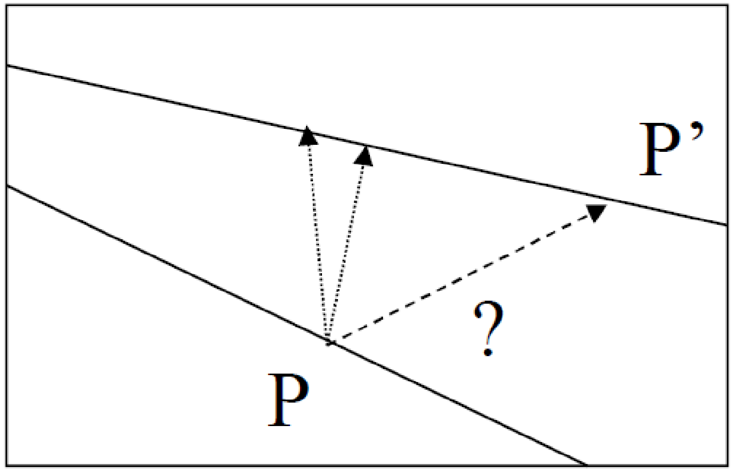
2 mins video - about the aperture problem

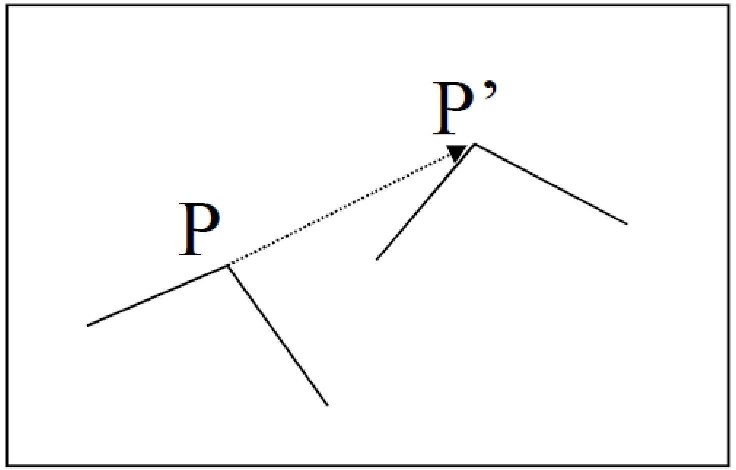


LearnOpenCV

Corners

- ✓ Aperture problem:





- ✓ **Corner** features are **useful** to compute the **correspondence**.

- ✓ Intensity discontinuities in **two** directions.

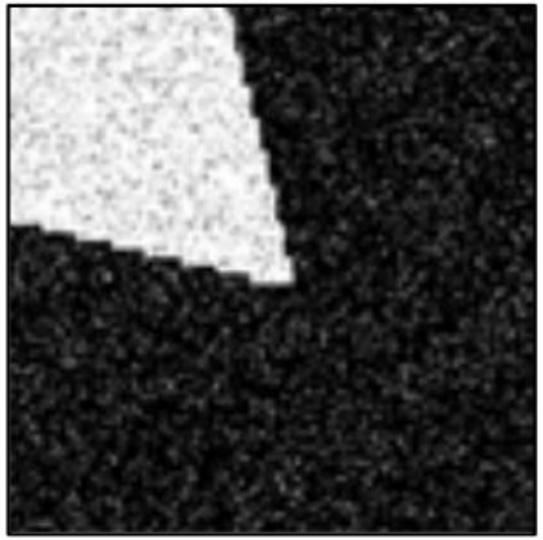
Corners

✓ **Point** where two edges meet.

i.e., rapid changes of image intensity in **two directions** within a **small region**

"Flat" Region

"Edge" Region



"Corner" Region

Why is it useful to find correspondence?

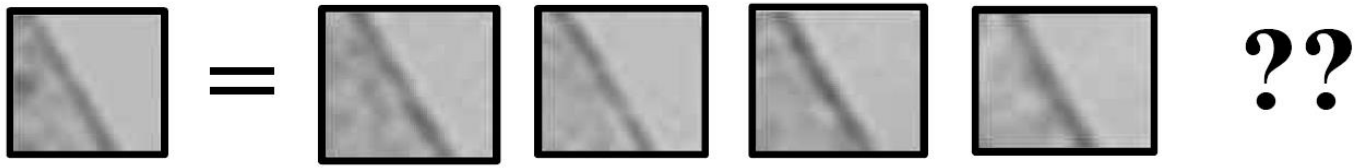
Corner Detection

✓ Intuitive understanding, let's try to use **flat** region to match two images.

Slide by F. Rameau.

Corner Detection

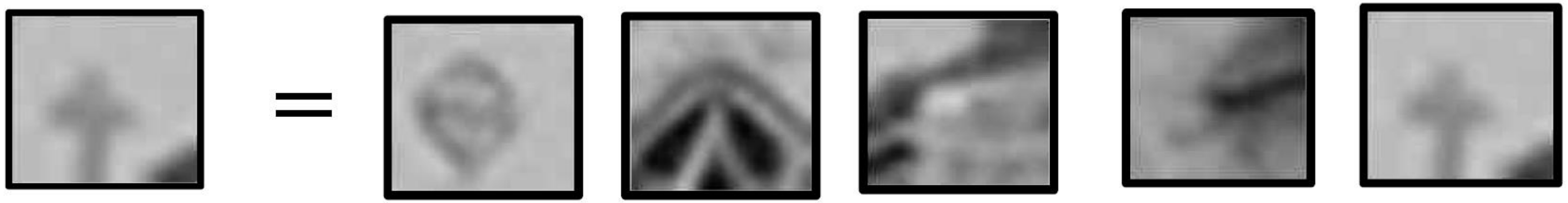
✓ Intuitive understanding, let's try to use **edge** region to match two images.



Slide by F. Rameau.

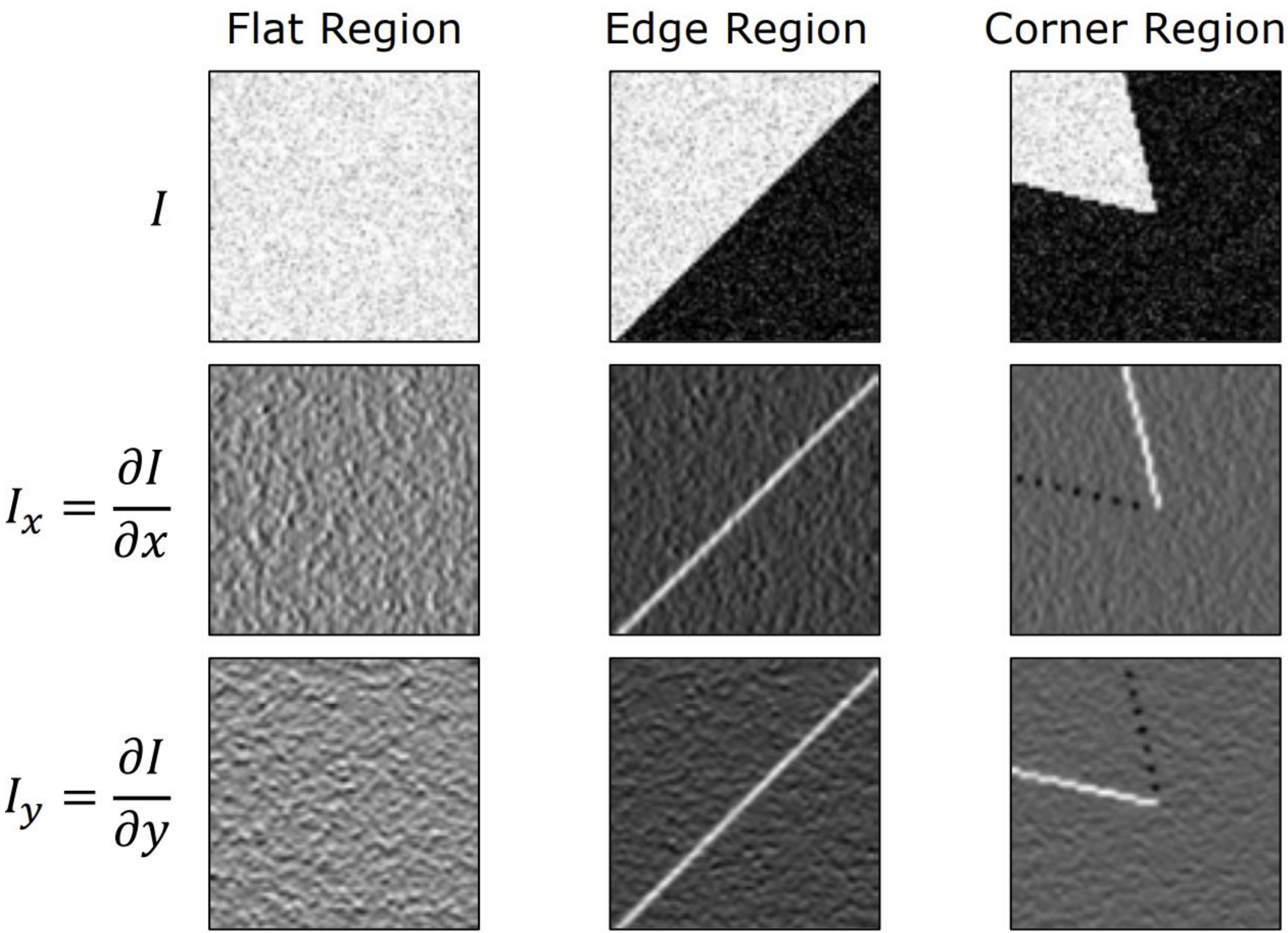
Corner Detection

✓ Intuitive understanding, let's try to use **corner** region to match two images.

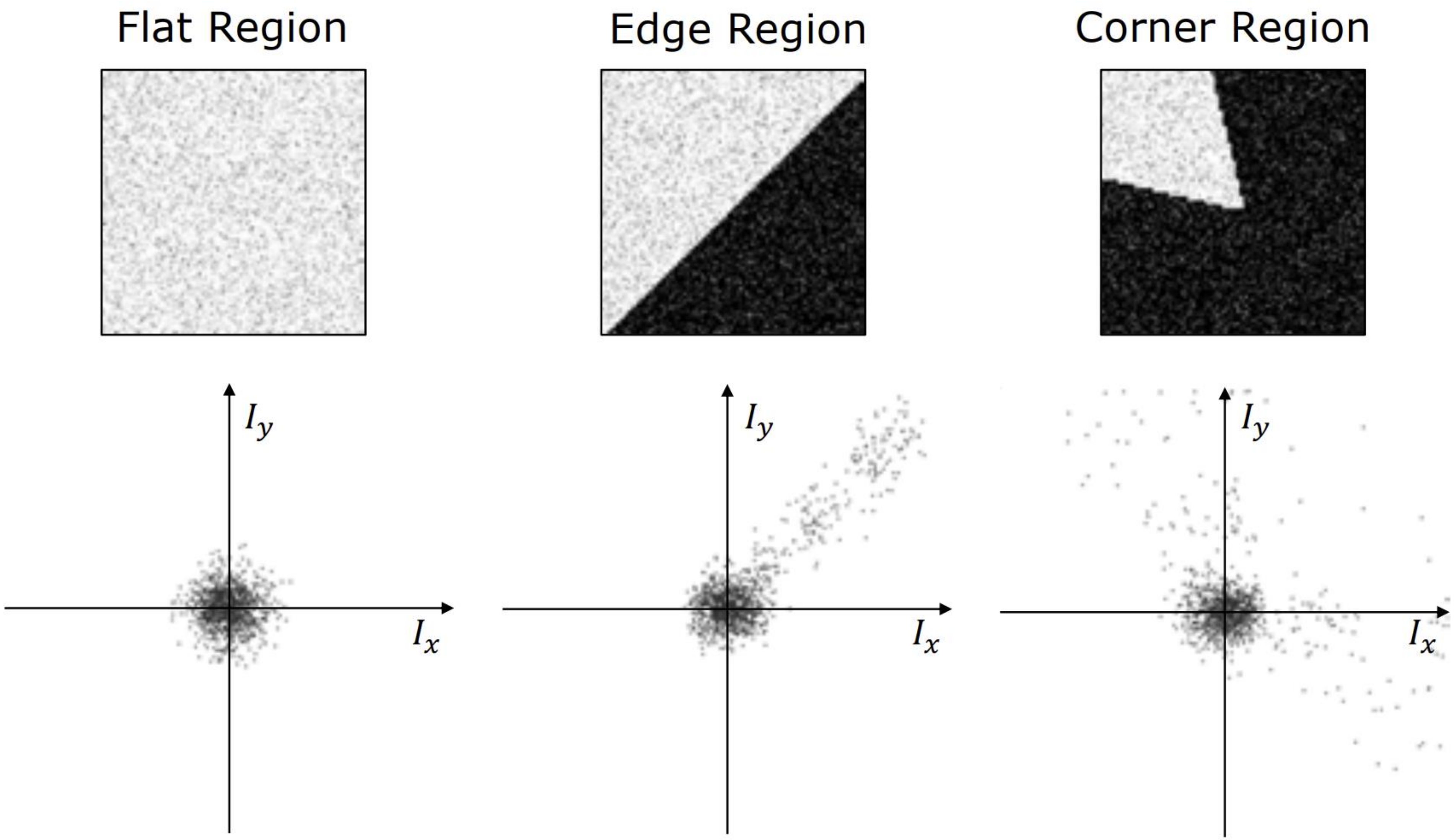


Slide by F. Rameau.

Image Gradients



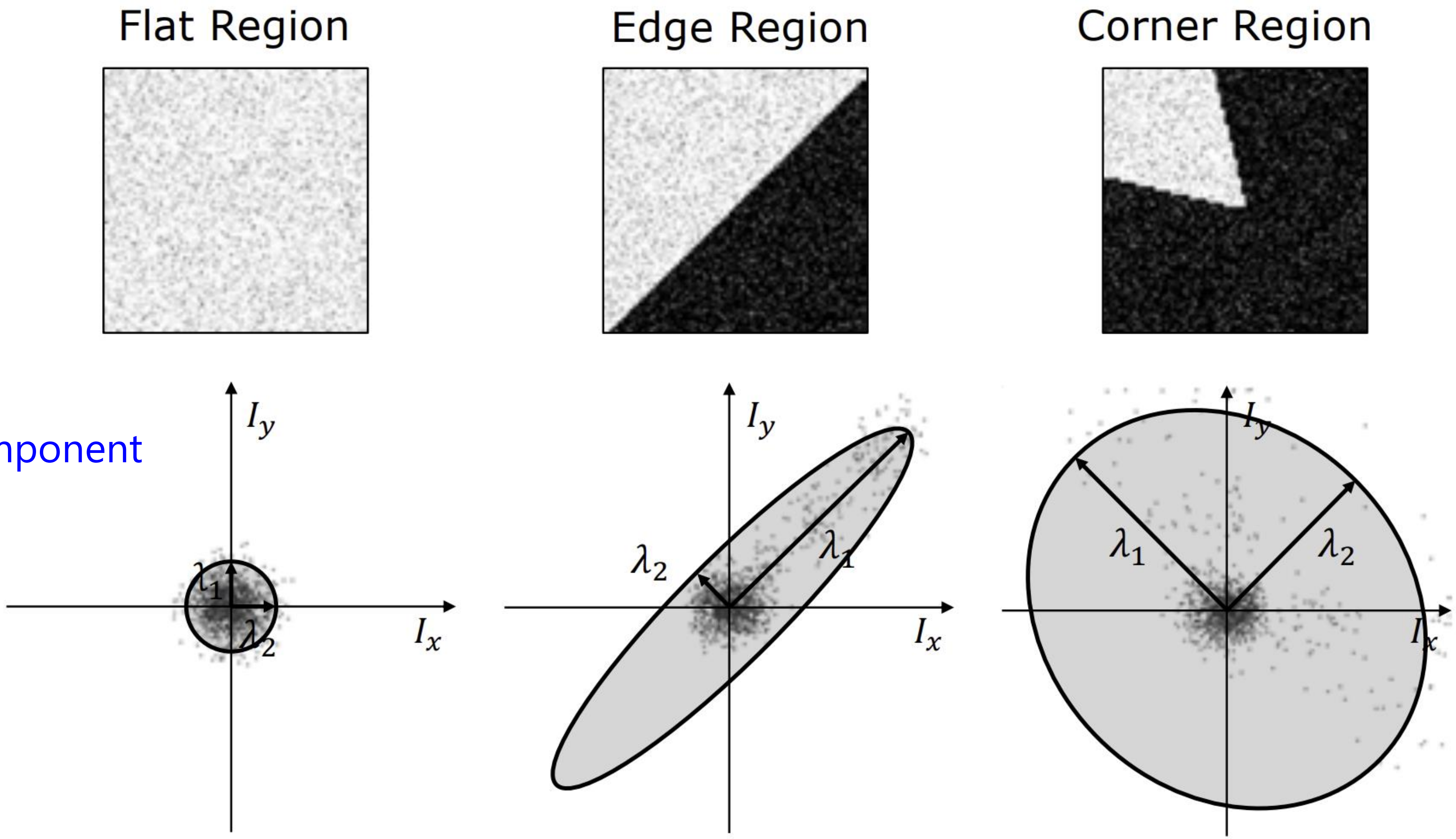
Distribution of Image Gradients



→ Distribution of I_x and I_y is different for all three regions.

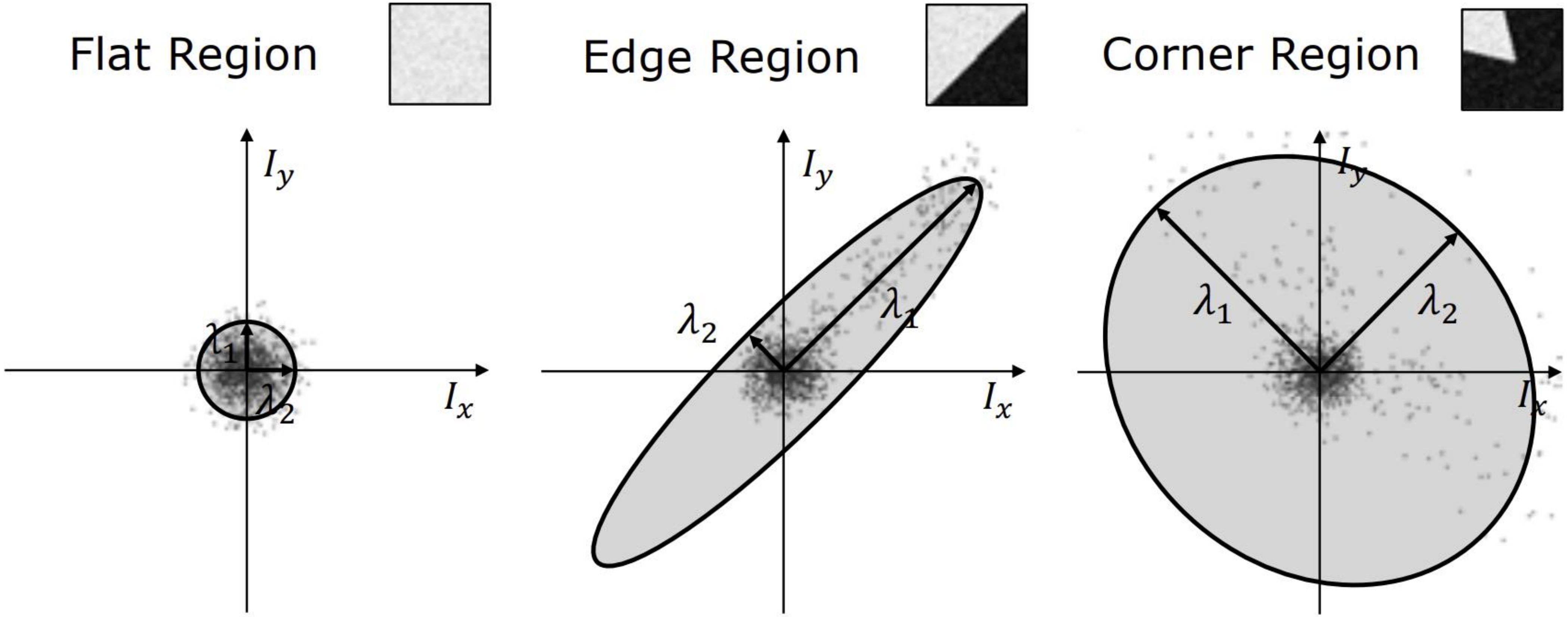
Fitting Elliptical Disk to Distribution

Principal component ellipse



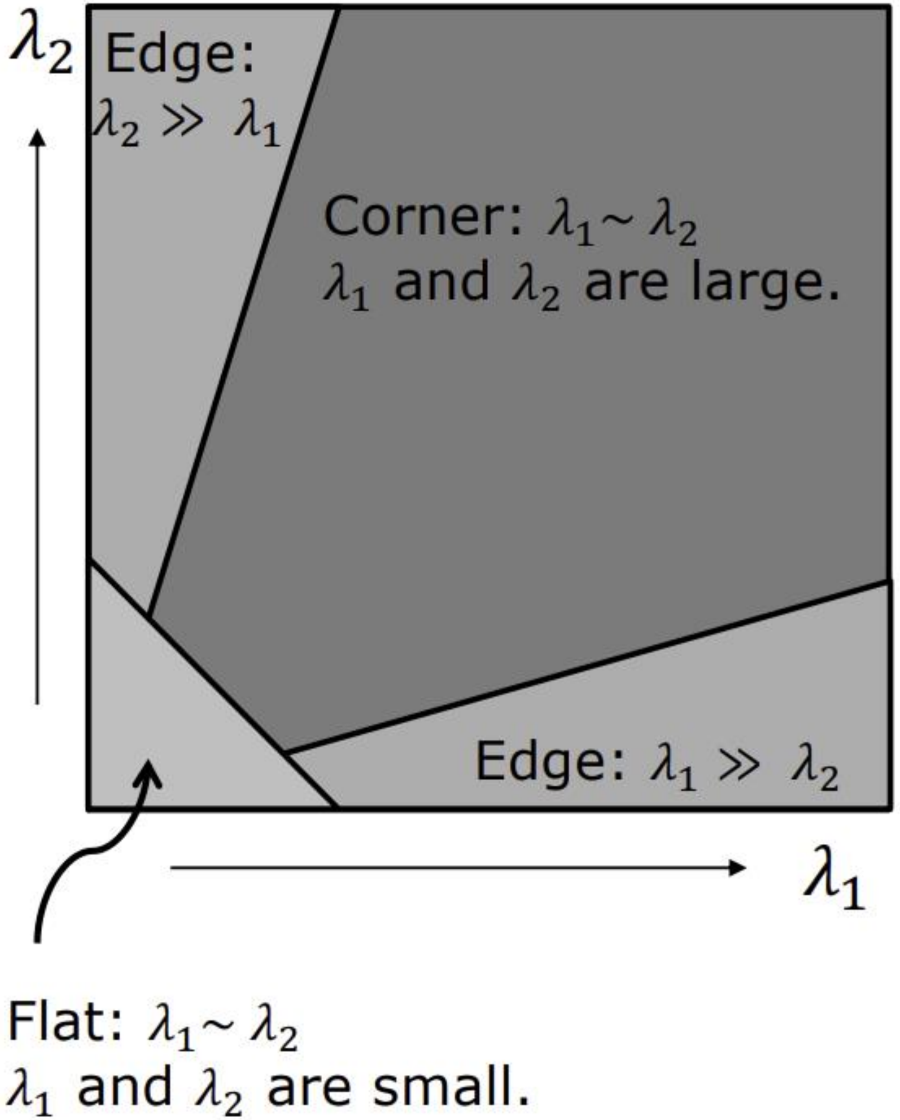
λ_1: Length of semi-major axis λ_2: Length of semi-minor axis

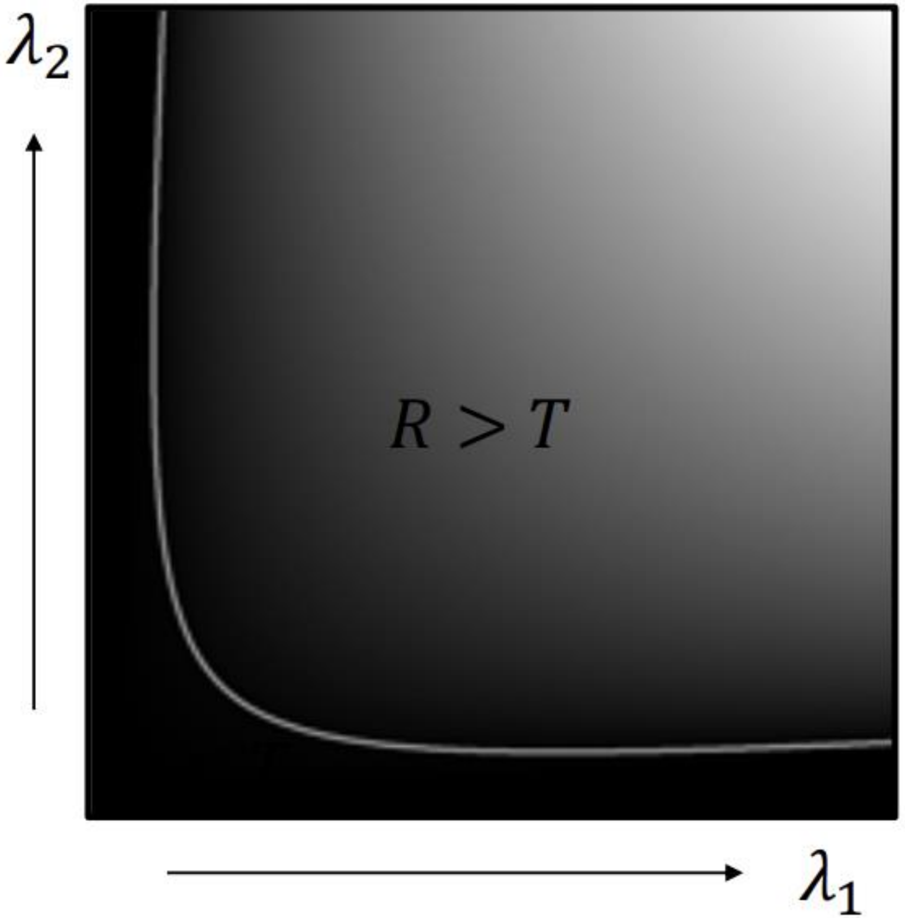
Interpretation of λ_1 and λ_2



$\lambda_1 \sim \lambda_2$	$\lambda_1 \gg \lambda_2$	$\lambda_1 \sim \lambda_2$
Both are Small	λ_1 is Large λ_2 is Small	Both are Large

Harris Corner Response Function





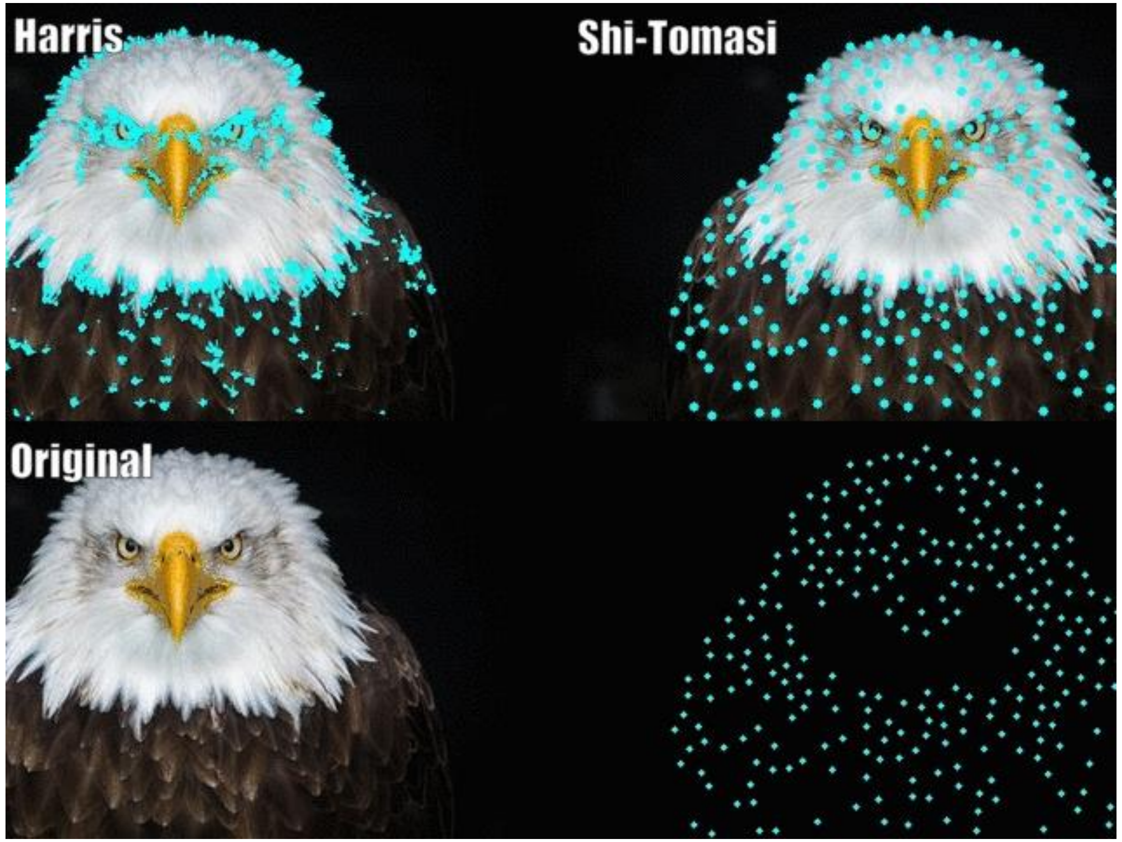
$$\boxed{R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2}$$

where: $0.04 \leq k \leq 0.06$

[1] Harris Corner Detection 1988.

Shi-Tomasi Corner Detector

- ✓ Only considers the smaller $\boldsymbol{\lambda}$
  - Key points are defined, where $min(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) > k$ is satisfied.



What are the differences between them?

Which one is better?

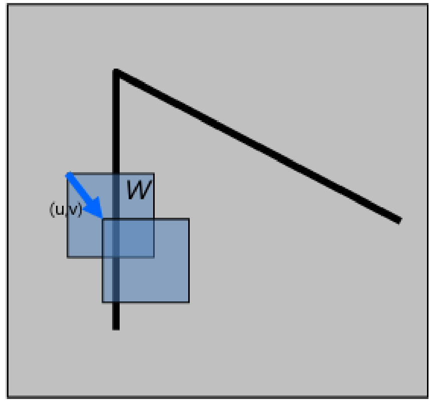
[1] J. Shi and C. Tomasi, "Good features to track", in CVPR 1994

Experiments: Harris Corner Detection

Updated codes (for python3) are uploaded in https://view.kentech.ac.kr/f088fa7f-874e-44bc-bd6d-6084b42dfdf7

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

```
$ python Harris.py
```

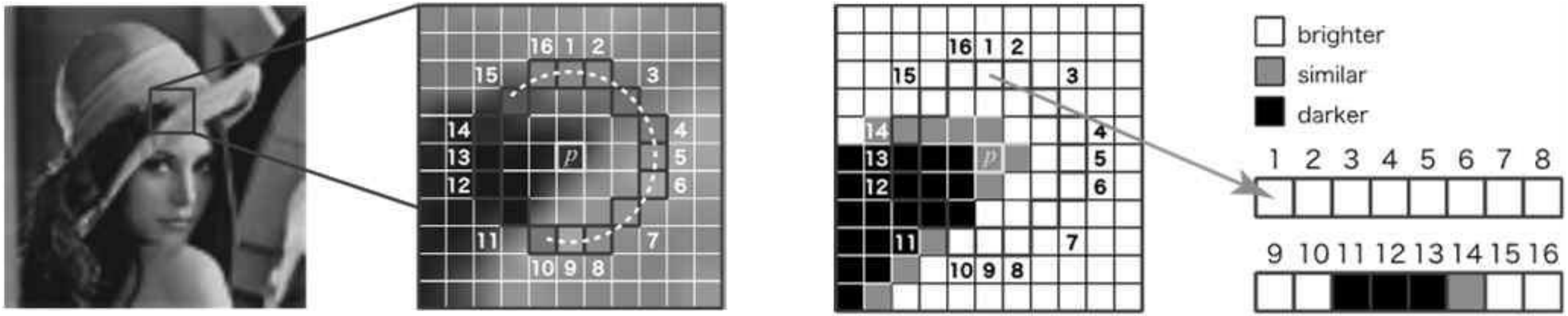


[Q] What are the **limitations** of Harris corner detector?

[A] we need to set **different threshold values** for every image in order to detect the most prominent interest points. **Slow** to compute exactly for each pixel and each offset (u, v).

FAST Corner Detector

- ✓ Harris corner detector is mathematically elegant, but is **not** the most efficient detector in term of **speed**.
- ✓ **F**eatures from **A**ccelerated **S**egment **T**est (**FAST**)
  - – FAST is a very **efficient** key point detector
- ✓ If *a set of **N** (e.g., 12) contiguous pixels in a ring* (of radius 3 around ***p***) are all brighter (darker) than the intensity of candidate pixel ***p*** ± ***t*** (*threshold*), then **p** is classified as a corner.



Viswanathan, Deepak. "Features from Accelerated Segment Test (FAST)." (2011).