KENTECH
Korea Institute of Energy Technology

Advanced Computer Vision
Week 05

Sep. 26, 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`
- `Feature Matching (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 (Qualcomm 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

Geometric Transformation

Codes are available at:

https://github.com/Leo-LiHao/OpenCV-Python-Tutorials

```
$ git clone https://github.com/Leo-LiHao/OpenCV-Python-Tutorials
$ cd OpenCV-Python-Tutorials/Src/ImageProcessing/GeometricTransform
```

Please try **four transformation codes** in the following order.

```
$ python GeometricTransform_rotateAndTrans.py
$ python GeometricTransform_resize.py
$ python GeometricTransform_affine.py
$ python GeometricTransform_perspective.py
```

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

Scaling, Rotation, Skew, Translation

$$\begin{bmatrix}\tilde{x}_2\\ \tilde{y}_2\\ \tilde{z}_2\end{bmatrix} = \begin{bmatrix} s_x & 0 & 0\\ 0 & s_y & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ y_1\\ 1\end{bmatrix}$$

Scaling

$$\begin{bmatrix}\tilde{x}_2\\ \tilde{y}_2\\ \tilde{z}_2\end{bmatrix} = \begin{bmatrix} 1 & m_x & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ y_1\\ 1\end{bmatrix}$$

Skew

$$\begin{bmatrix}\tilde{x}_2\\ \tilde{y}_2\\ \tilde{z}_2\end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x\\ 0 & 1 & t_y\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ y_1\\ 1\end{bmatrix}$$

Translation

$$\begin{bmatrix}\tilde{x}_2\\ \tilde{y}_2\\ \tilde{z}_2\end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta & 0\\ sin\theta & cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ y_1\\ 1\end{bmatrix}$$

Rotation

Composition of these transformations?

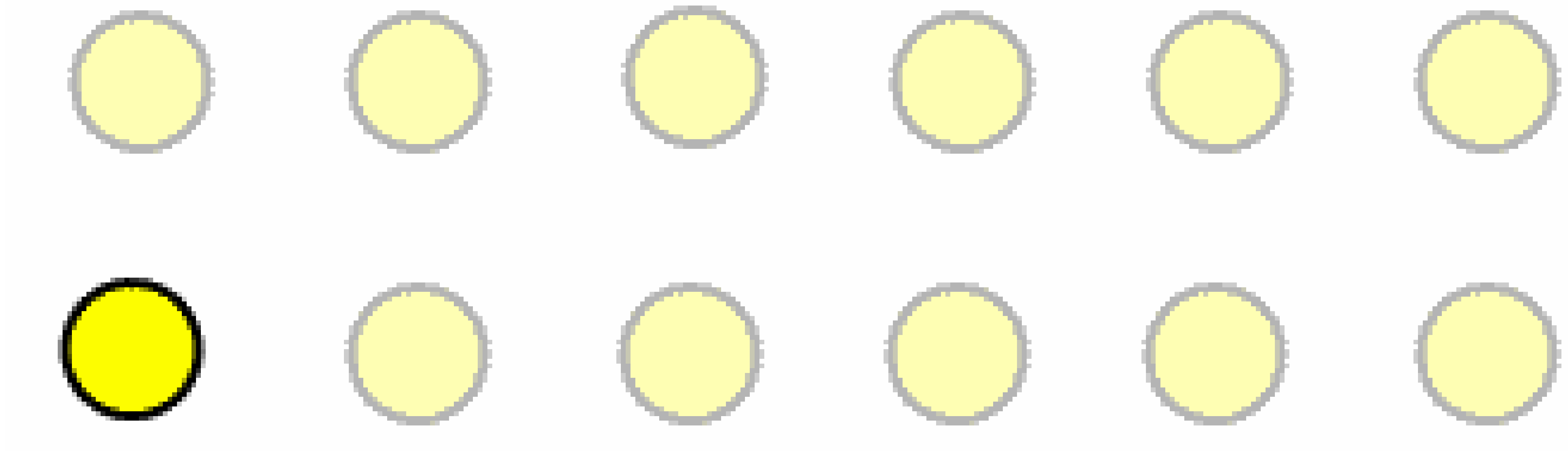
[1] Original slide by Shree Nayar

KENTECH
Korea Institute of Energy Technology

Visual Correspondence

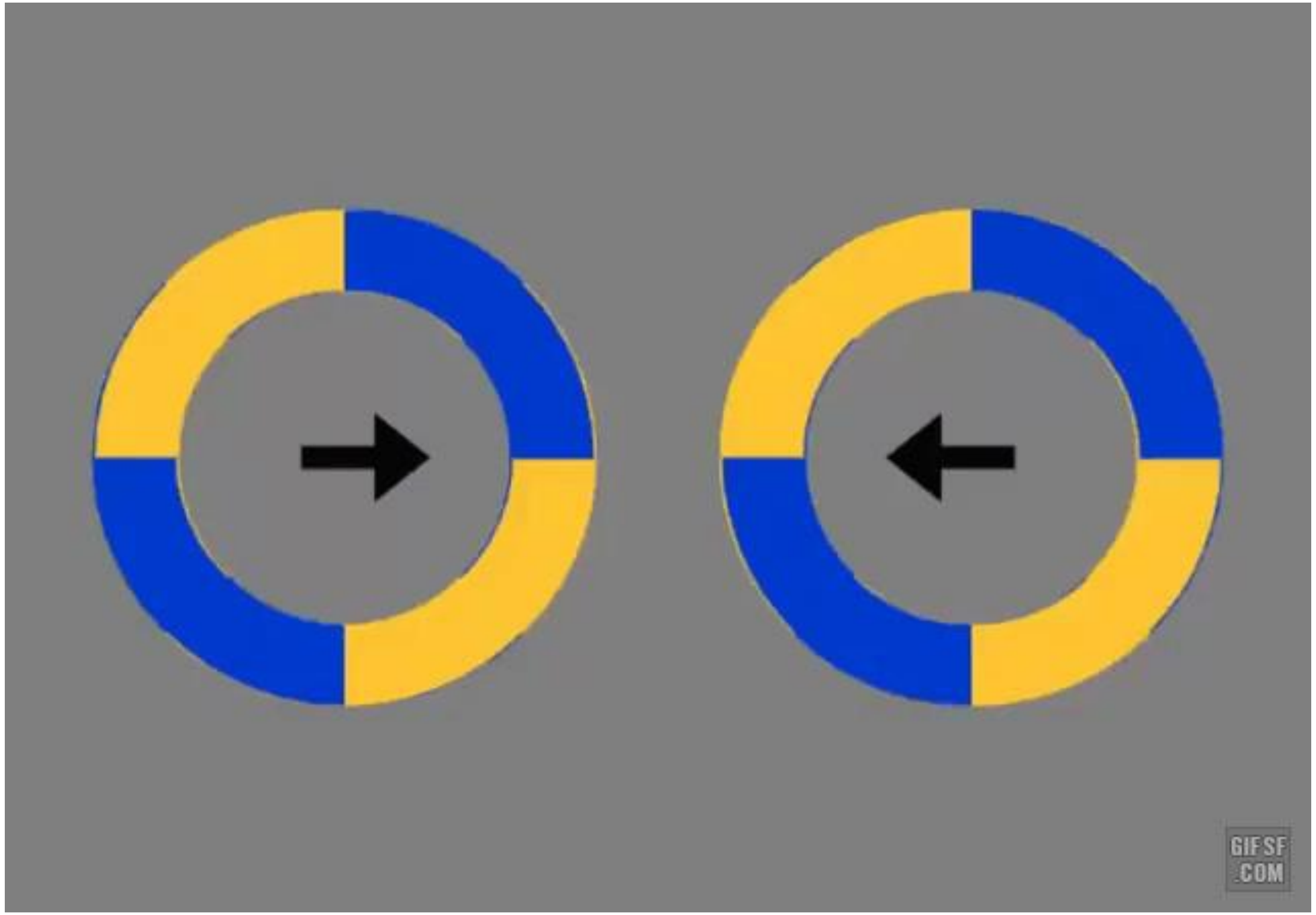
Basic Function of Human Brain: Motion Perception

✓ We always try to find **visual correspondence**!



Basic Function of Human Brain: Motion Perception

✓ We always try to find **visual correspondence**!

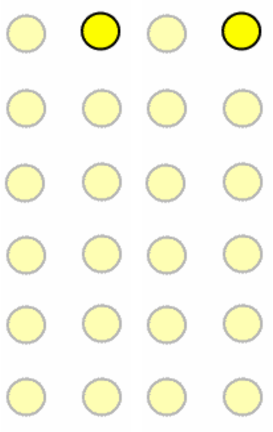


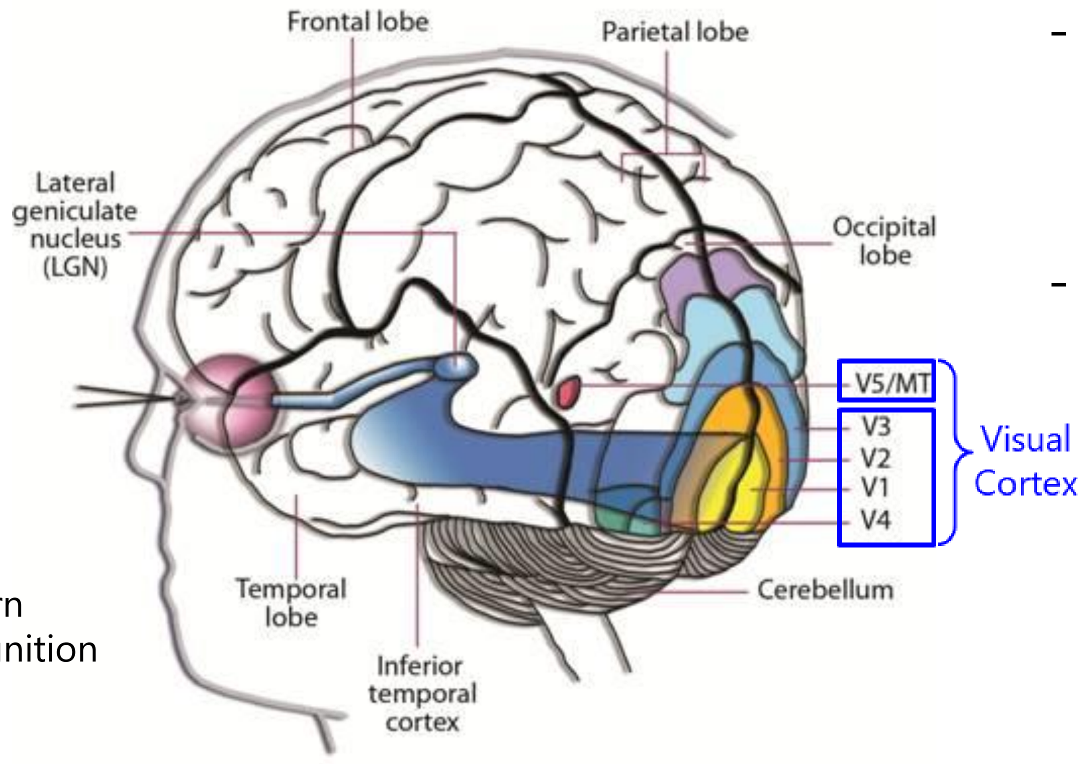
Due to the difference in thin borderlines outside the circle!

Basic Function of Human Brain: Motion Perception

- **Neurophysiological study**

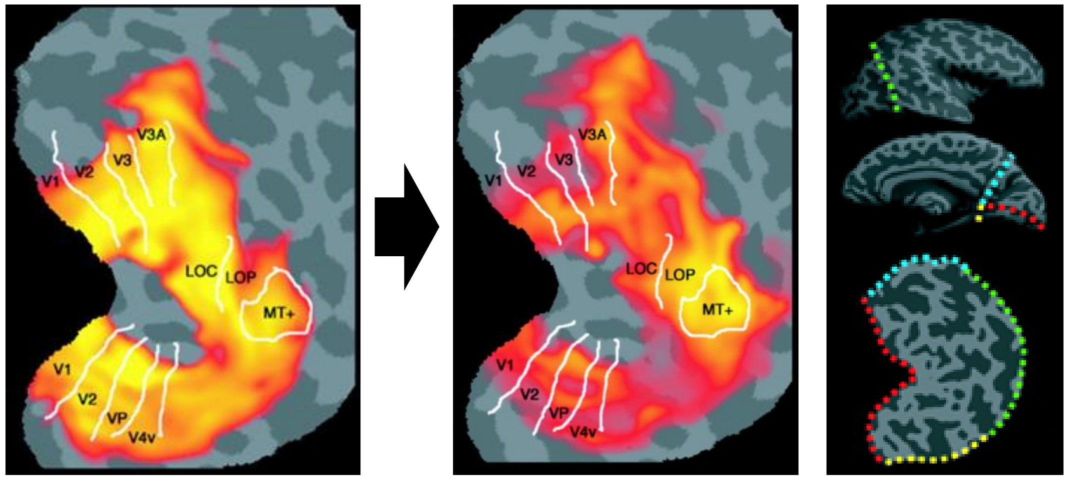
  : Visual association area [1]





Flickering pattern
→ Motion recognition

- **Visual motion perception in visual cortex**
- V1 ~ V4 (visual area)
  - → Retinotopic visual areas processing visual stimuli
- MT (Middle Temporal) / V5
  - → **Motion-sensitive** area
  - → High response to change in signal intensity, motion stimuli such as flickering
- Motion aftereffect
  - → Brain response after **adapted** to **motion stimuli** [2]

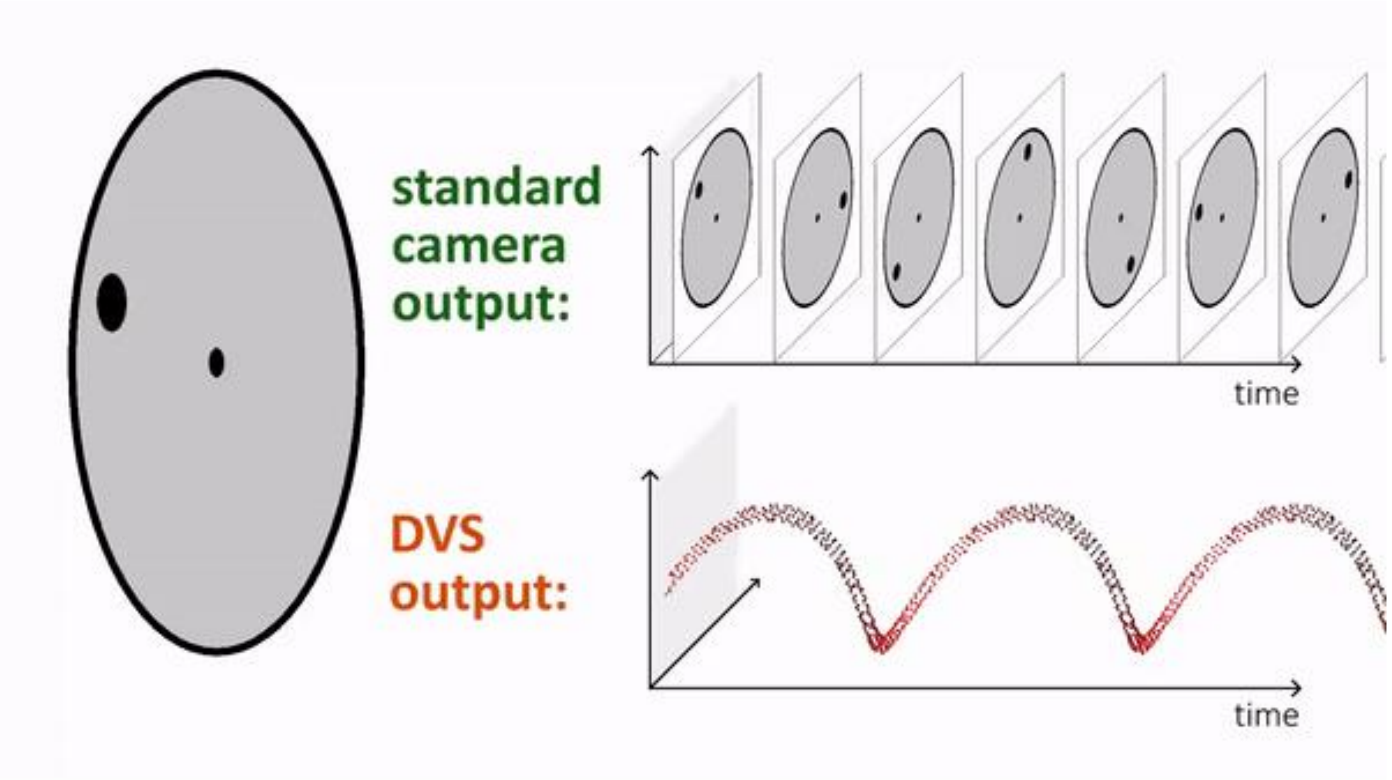


[1] Neuroscience of Imagination, http://mobilereference.com/mind/online/index.htm
[2] Seiffert et al., "Functional MRI studies of human visual motion perception: texture, luminance, attention and after-effects", *Cerebral Cortex* 2003

Bio-Inspired Vision Sensor

- **Dynamic Vision Sensor (DVS)**

  : Event-based camera [1]



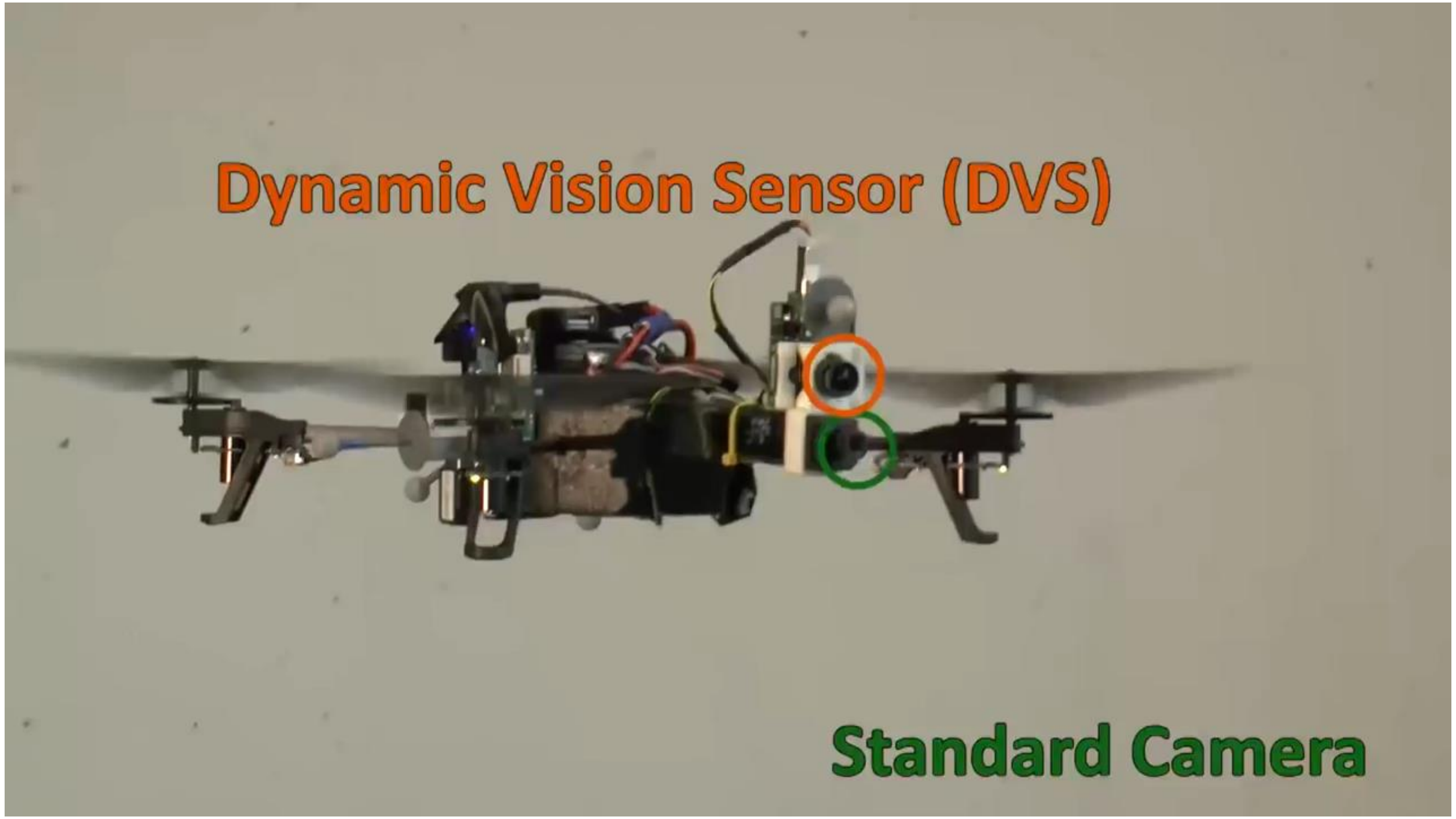
- **Standard camera**
  - **Pixels**: globally triggered
  - **Output**: full image frames at fixed frame rate

- **DVS**
  - **Pixels**: independent and asynchronous
  - **Output**: sequence of events
    (local brightness changes)
  - High temporal resolution (no motion blur)
  - Dynamic range that resembles human eye

[1] Mueggler et al., "Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers", IROS 2014

Bio-Inspired Vision Sensor: DVS

- Various **robotic** applications under **dynamic** environments



[1] Mueggler et al., "Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers", IROS 2014

Bio-Inspired Vision Sensor: DVS

- Various **robotic** applications under **dynamic** environments

[1] Dynamic Obstacle Avoidance for Quadrotors with Event Cameras (Science Robotics 2020)

Bio-Inspired Vision Sensor: DVS

- Various **robotic** applications under **dynamic** environments

GoPro vs Prophesee event-camera

[1] A gentle introduction to event cameras. 2020.

Bio-Inspired Vision Sensor: DVS

- Various **robotic** applications under **dynamic** environments

[1] G. Gallego, et al., "Event-Based Vision: A Survey", T-PAMI 2022.

KENTECH
Korea Institute of Energy Technology

Image Features

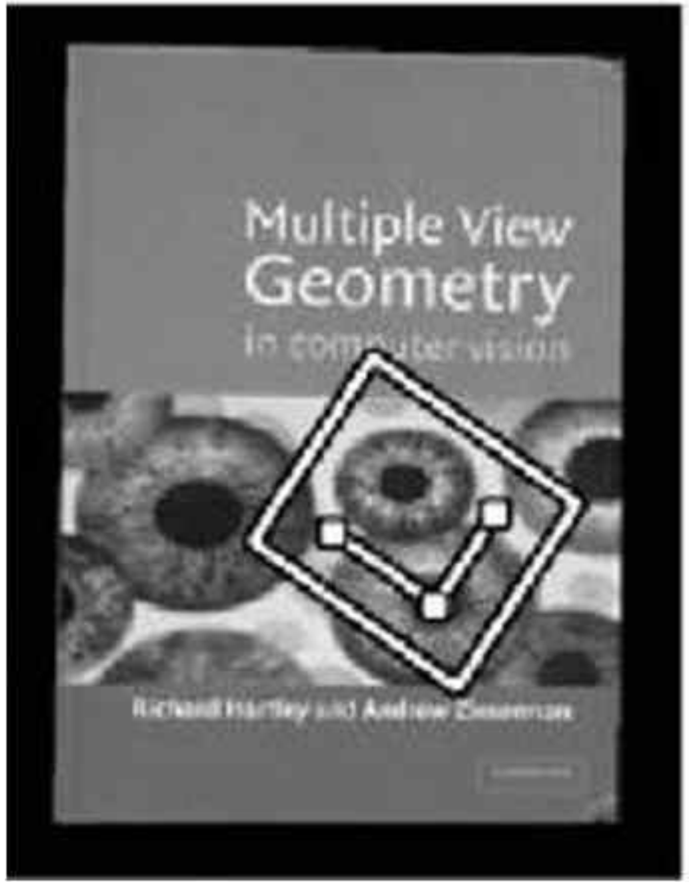
Why Detecting Image Features?

✓ 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.

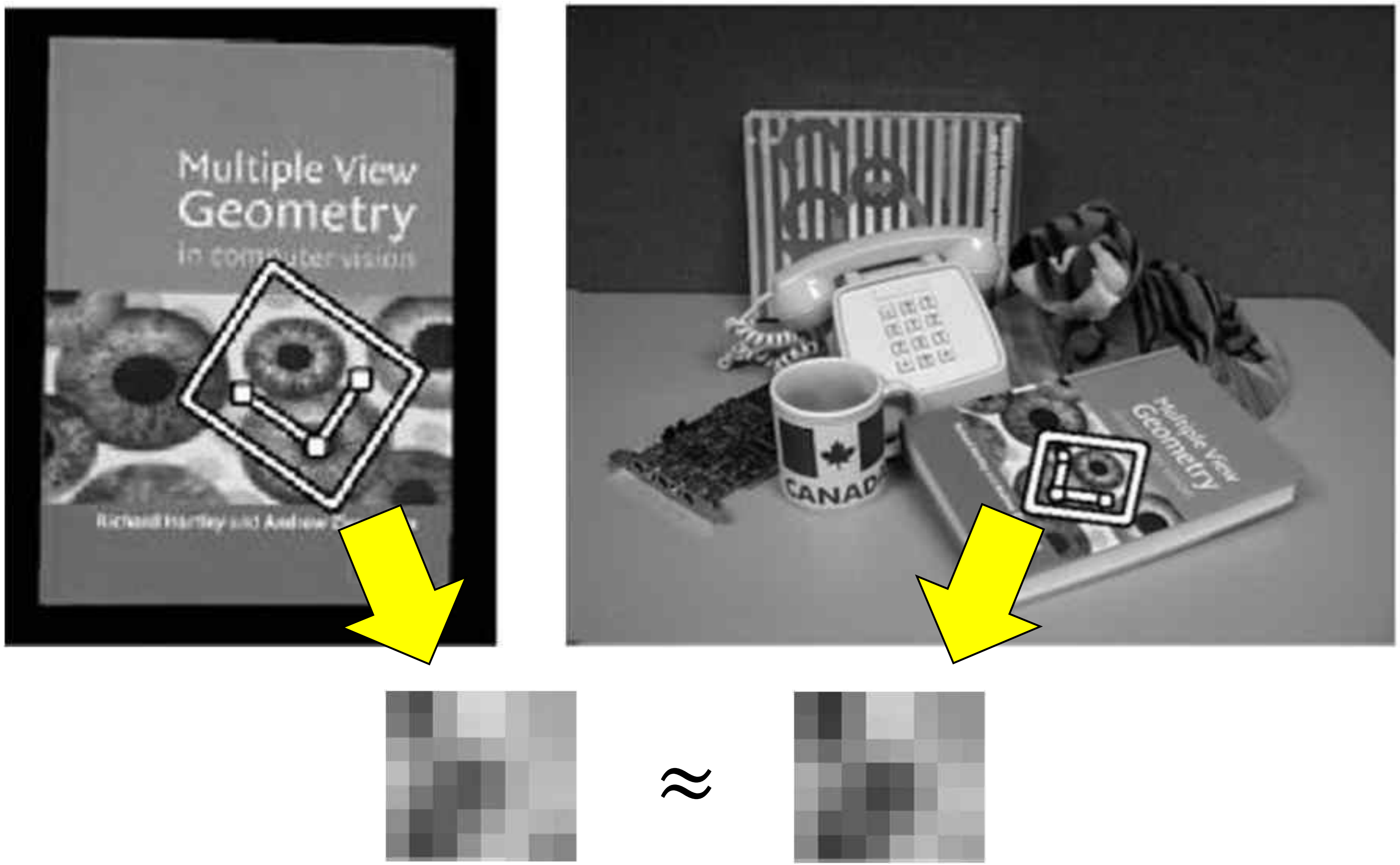
Pyimagesearch

Image Matching



Slide by R. Szeliski.

Image Matching



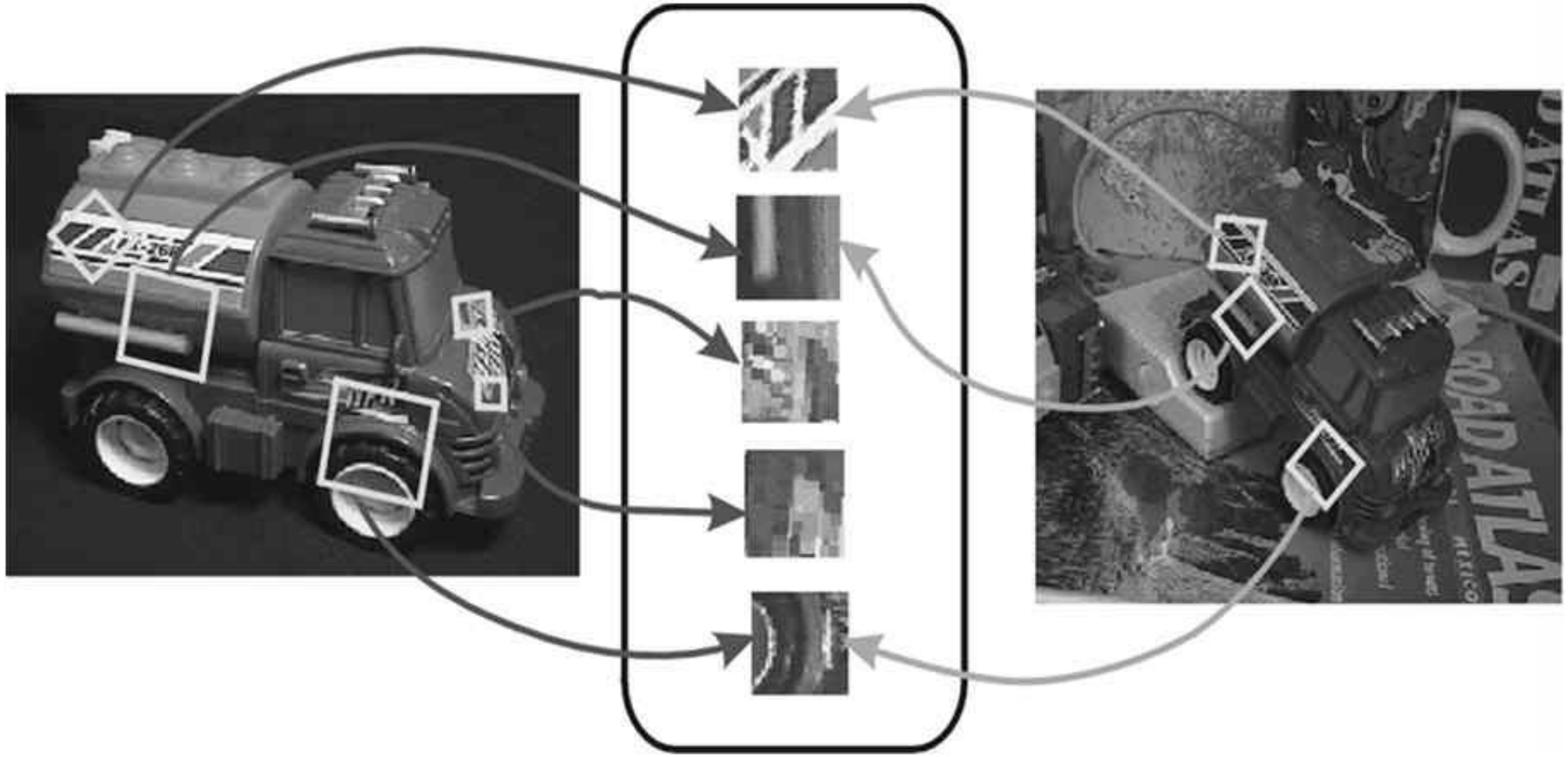
Slide by R. Szeliski

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.

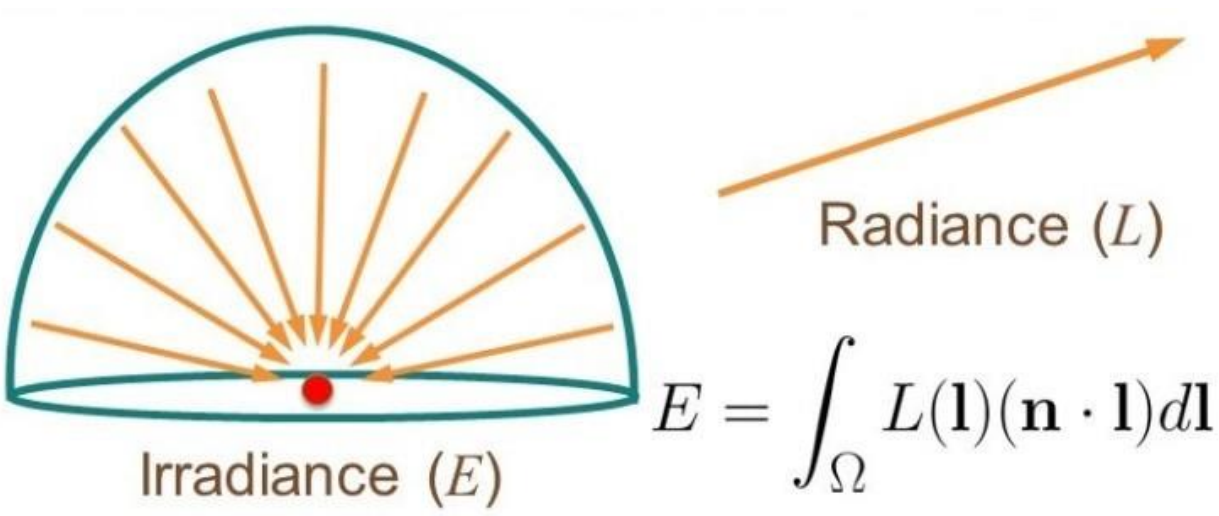
Intensity Discontinuities

✓ Changes in the **image irradiance** correspond to changes in the **scene radiance**.

복사조도(단위 평면에 입사되는 복사량) 복사휘도(단위 입체 각 당 복사량)

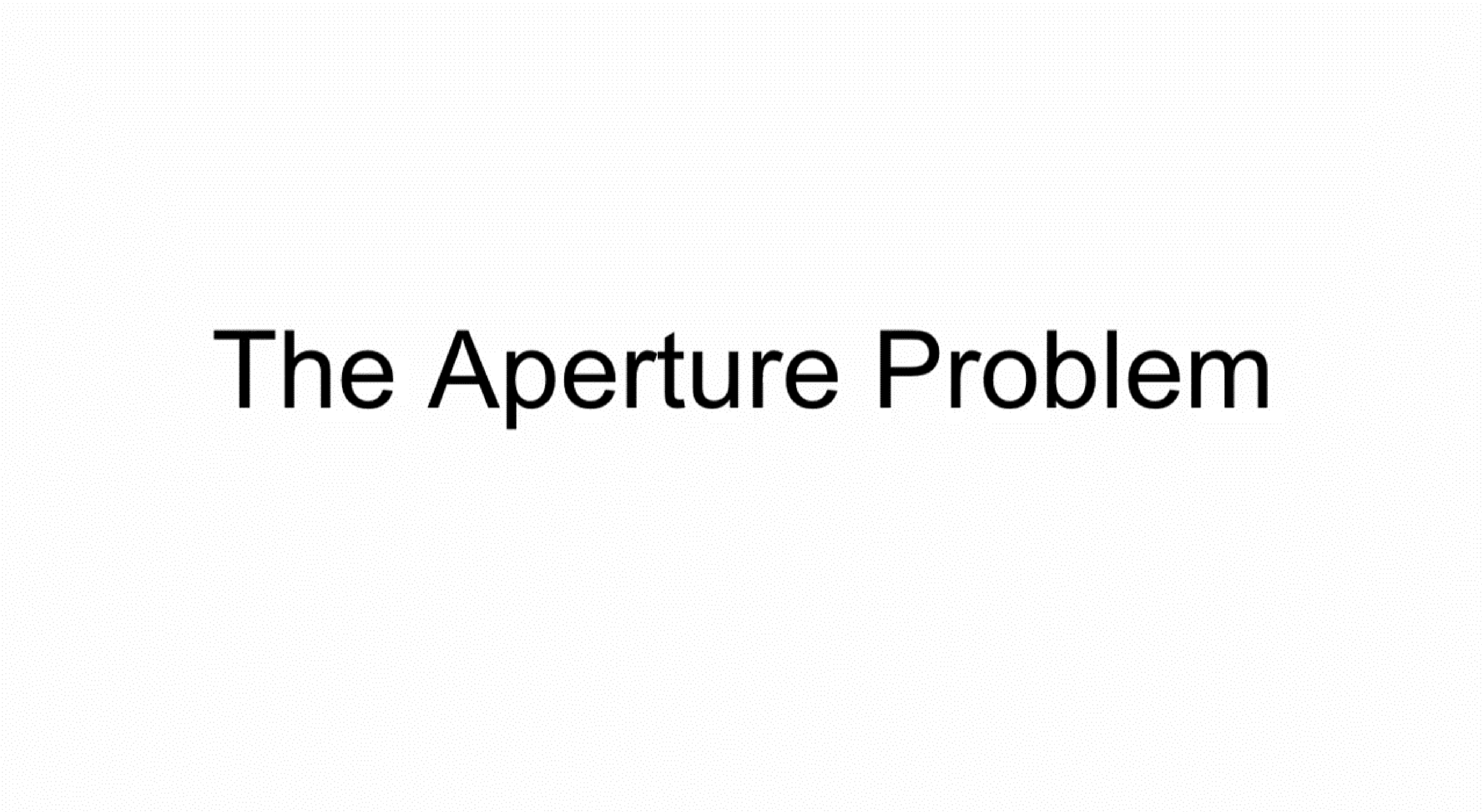
✓ Scene radiance changes provide **important cues** about the **scene events**:

- Surface normal discontinuity
- Depth discontinuity
- Reflectance discontinuity
- Illumination discontinuity.



$$E = \int_{\Omega} L(\mathbf{l})(\mathbf{n} \cdot \mathbf{l})d\mathbf{l}$$

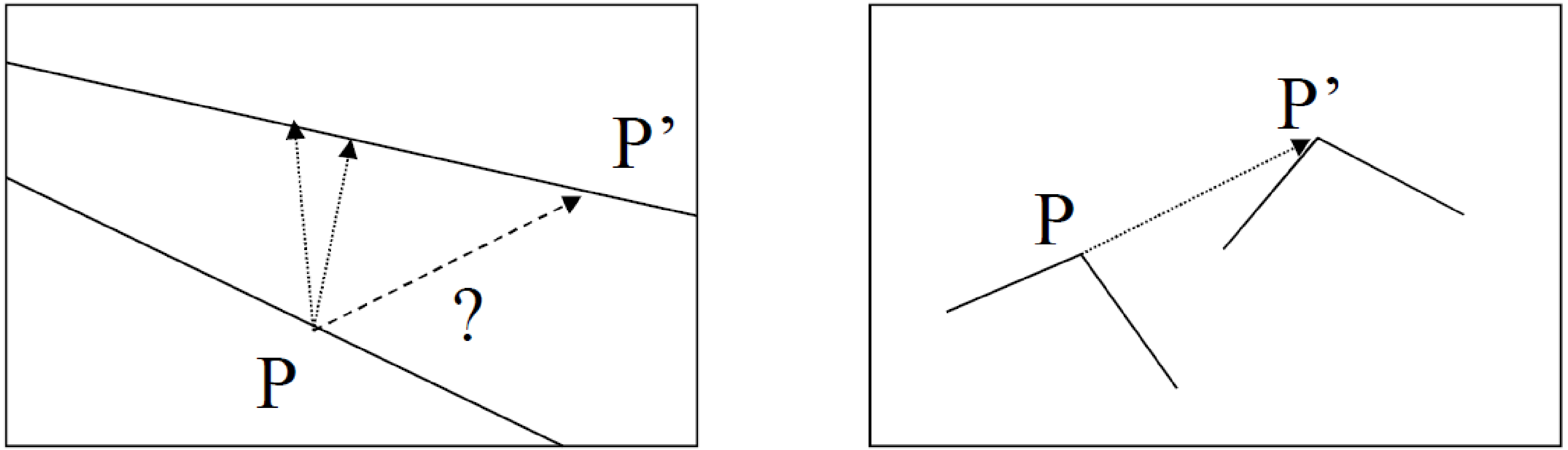
Corners



LearnOpenCV

Corners

- ✓ Aperture problem:



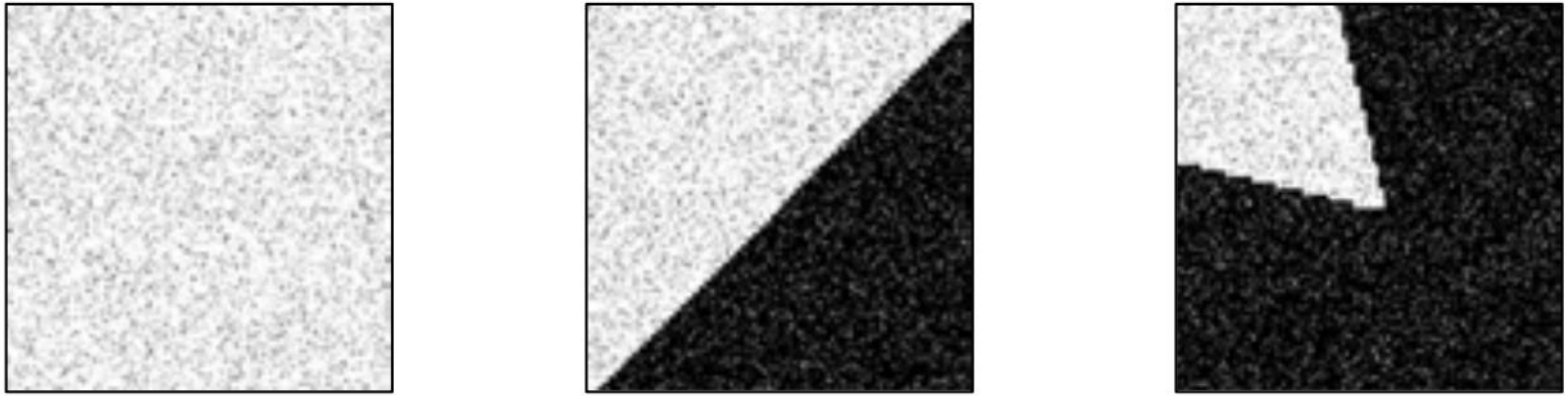
- ✓ **Corner** features are useful to compute the **correspondence**.

- ✓ 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

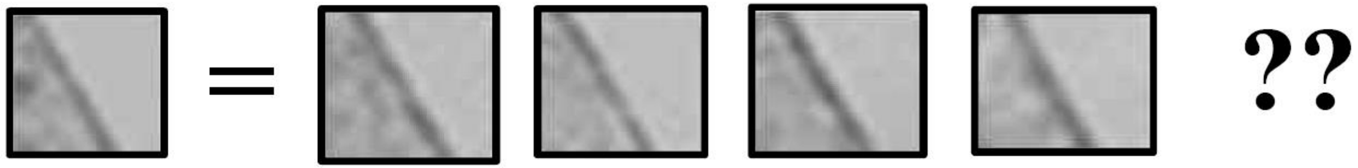
Corner Detection

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

Slide by F. Rameau.

Corner Detection

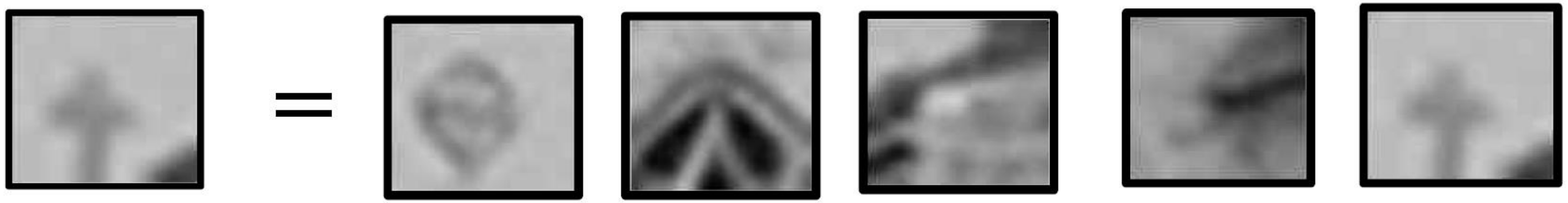
✓ Intuitive understanding, let's try to use **edge** region to match two images.



Slide by F. Rameau.

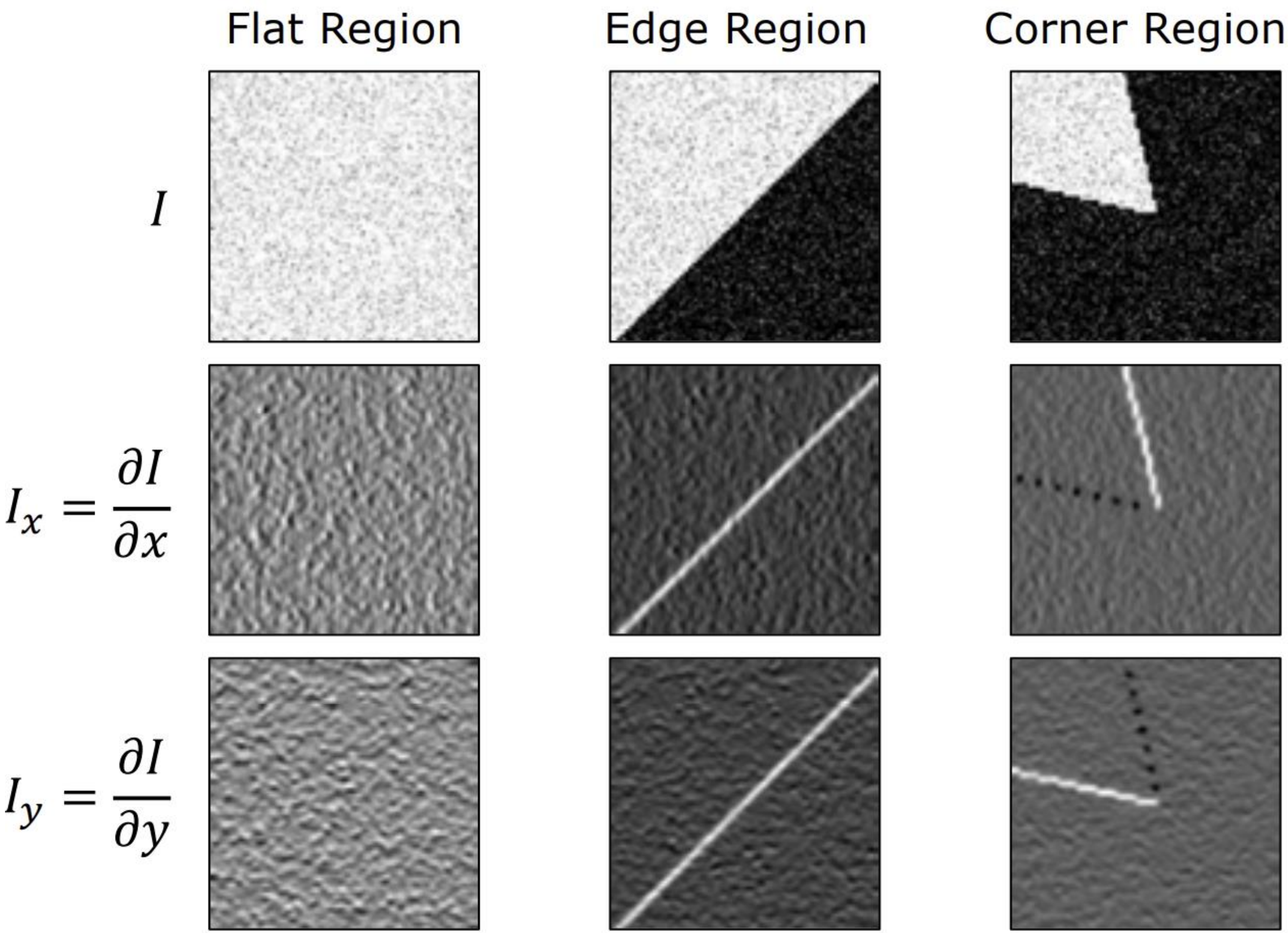
Corner Detection

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

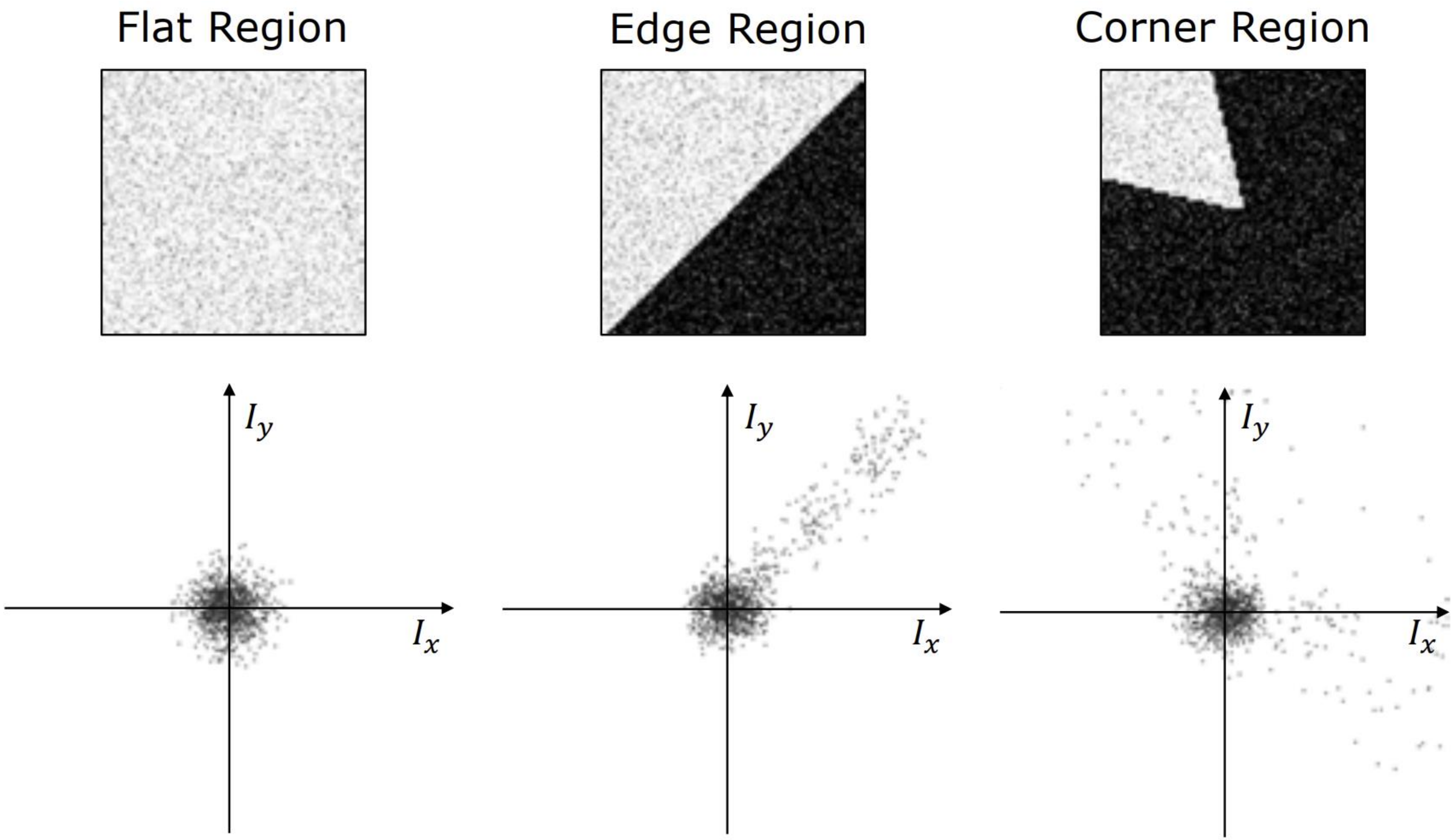


Slide by F. Rameau.

Image Gradients

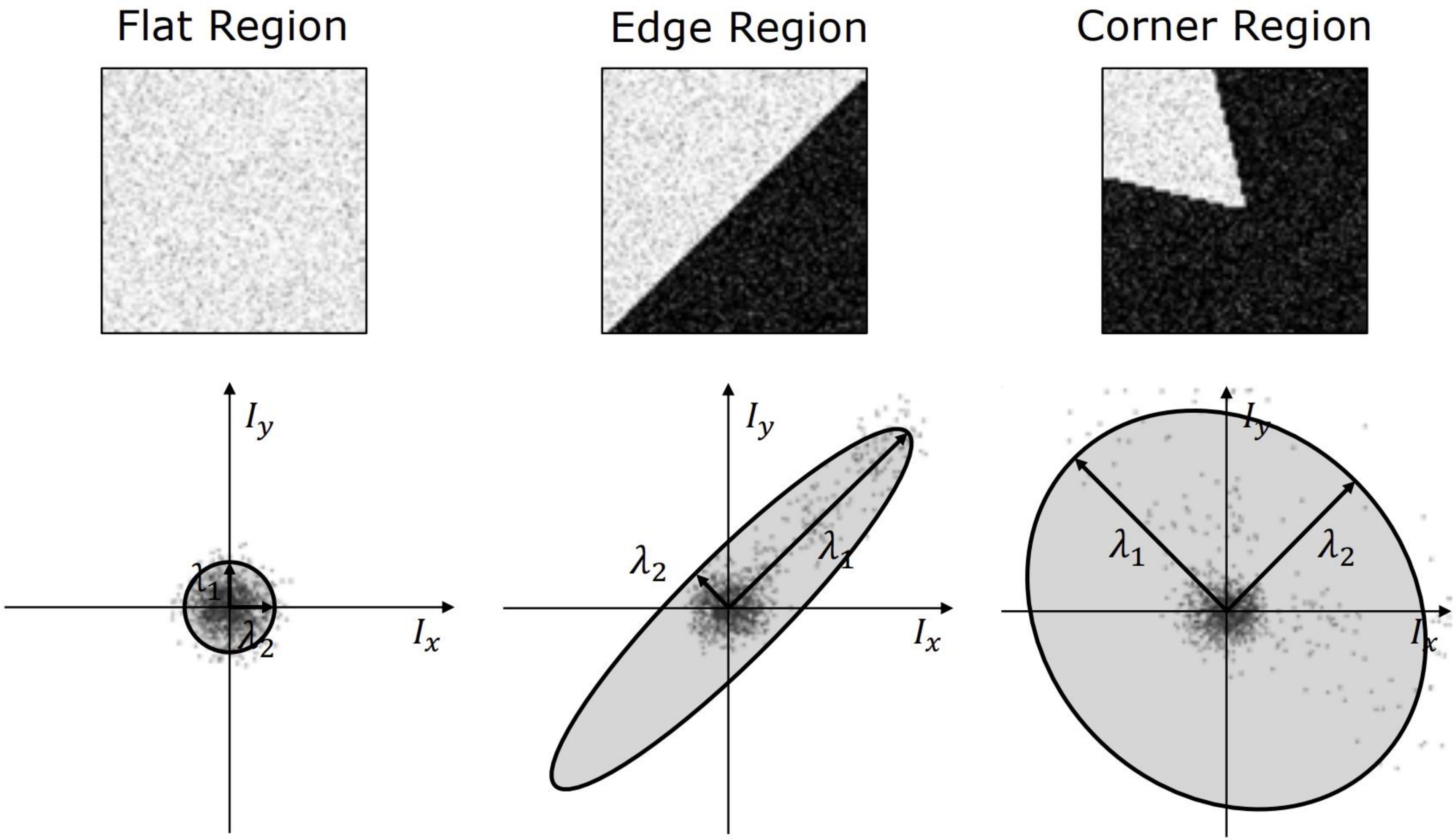


Distribution of Image Gradients



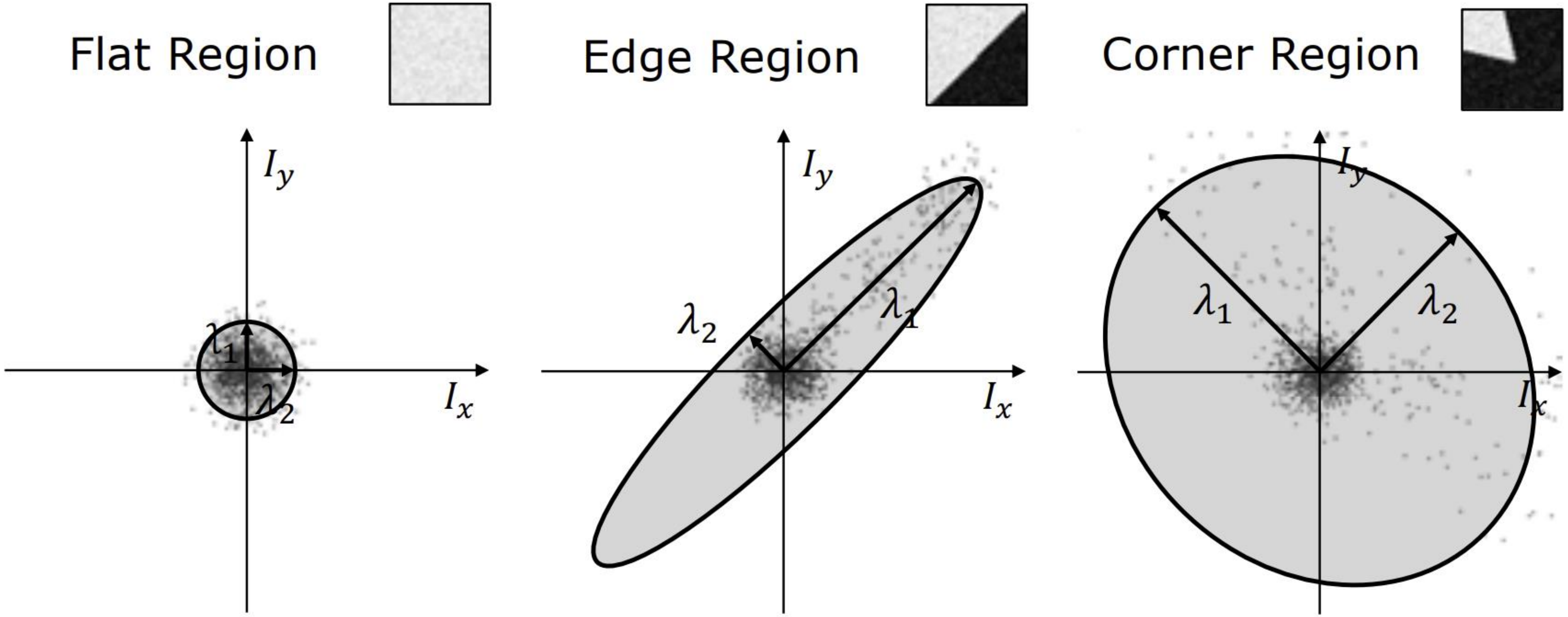
→ Distribution of I_x and I_y is different for all three regions.

Fitting Elliptical Disk to Distribution



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

Interpretation of λ_1 and λ_2

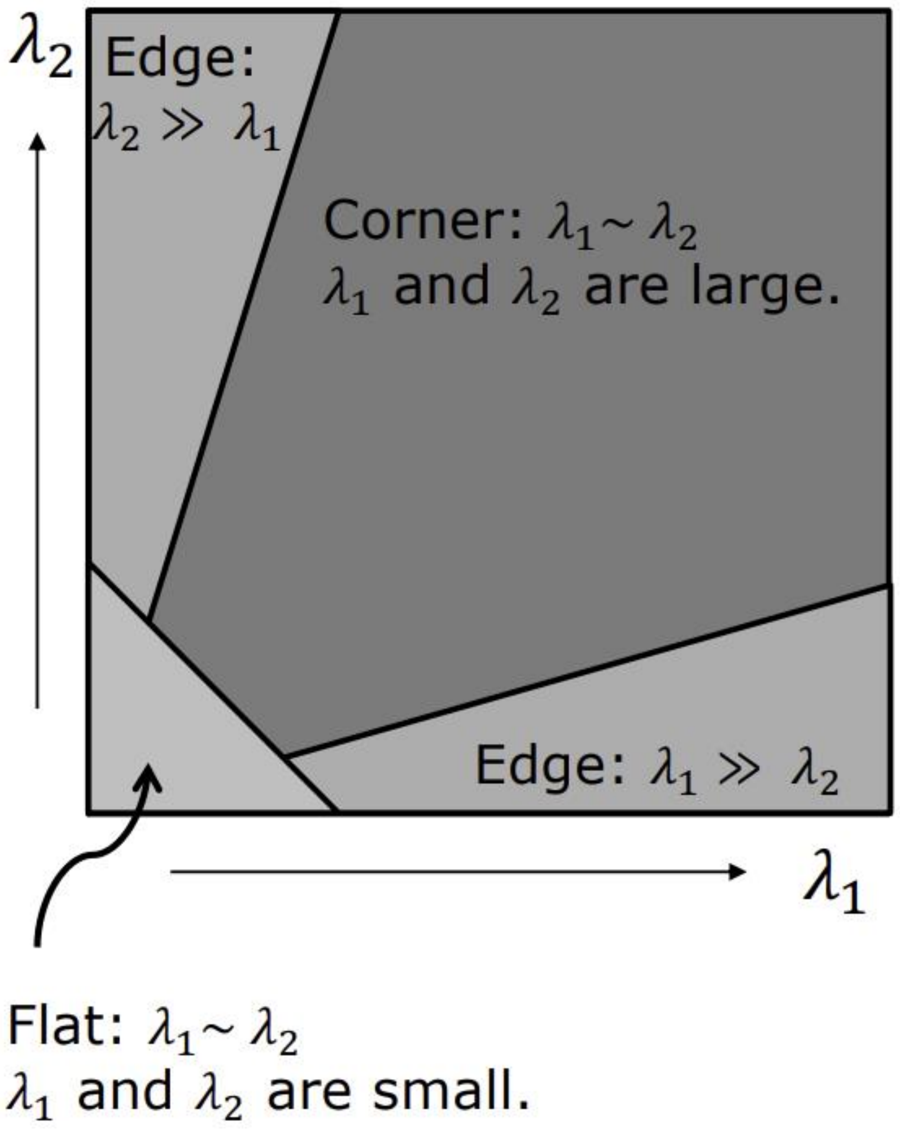


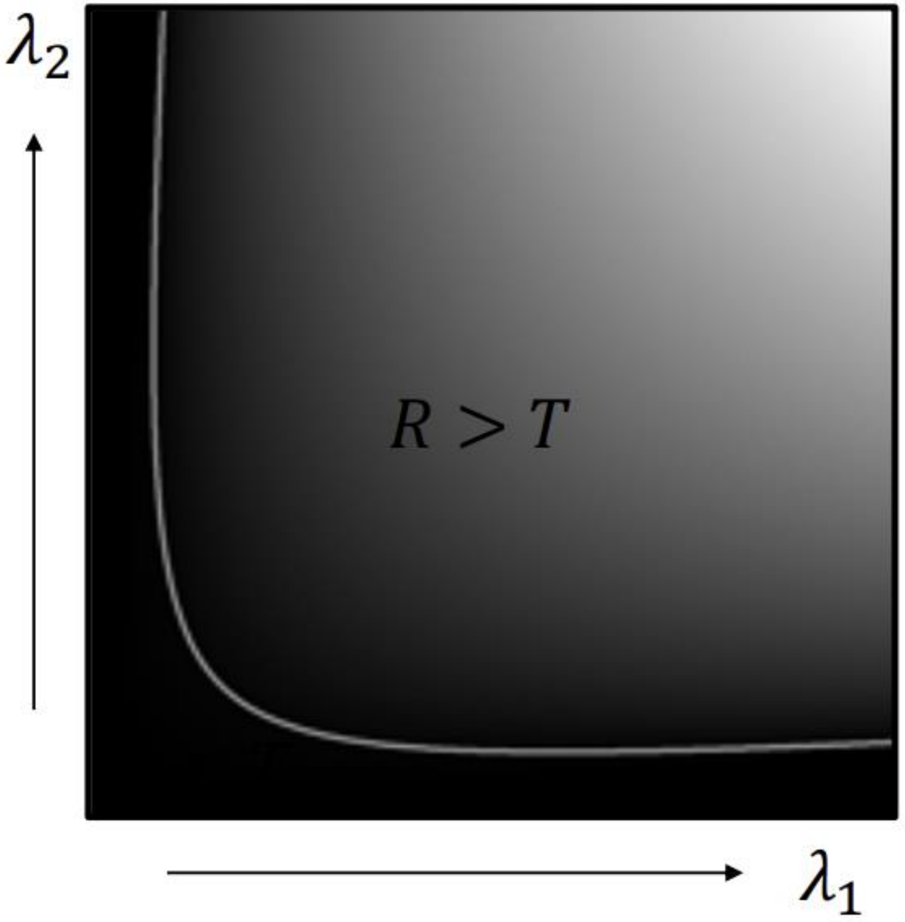
$\lambda_1 \sim \lambda_2$
Both are Small

$\lambda_1 \gg \lambda_2$
λ_1 is Large
λ_2 is Small

$\lambda_1 \sim \lambda_2$
Both are Large

Harris Corner Response Function





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

where: $0.04 \le k \le 0.06$

[1] Harris Corner Detection 1988.

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
```