KENTECH
Korea Institute of Energy Technology

Advanced Computer Vision
Week 03

Sep. 16, 2022
Seokju Lee

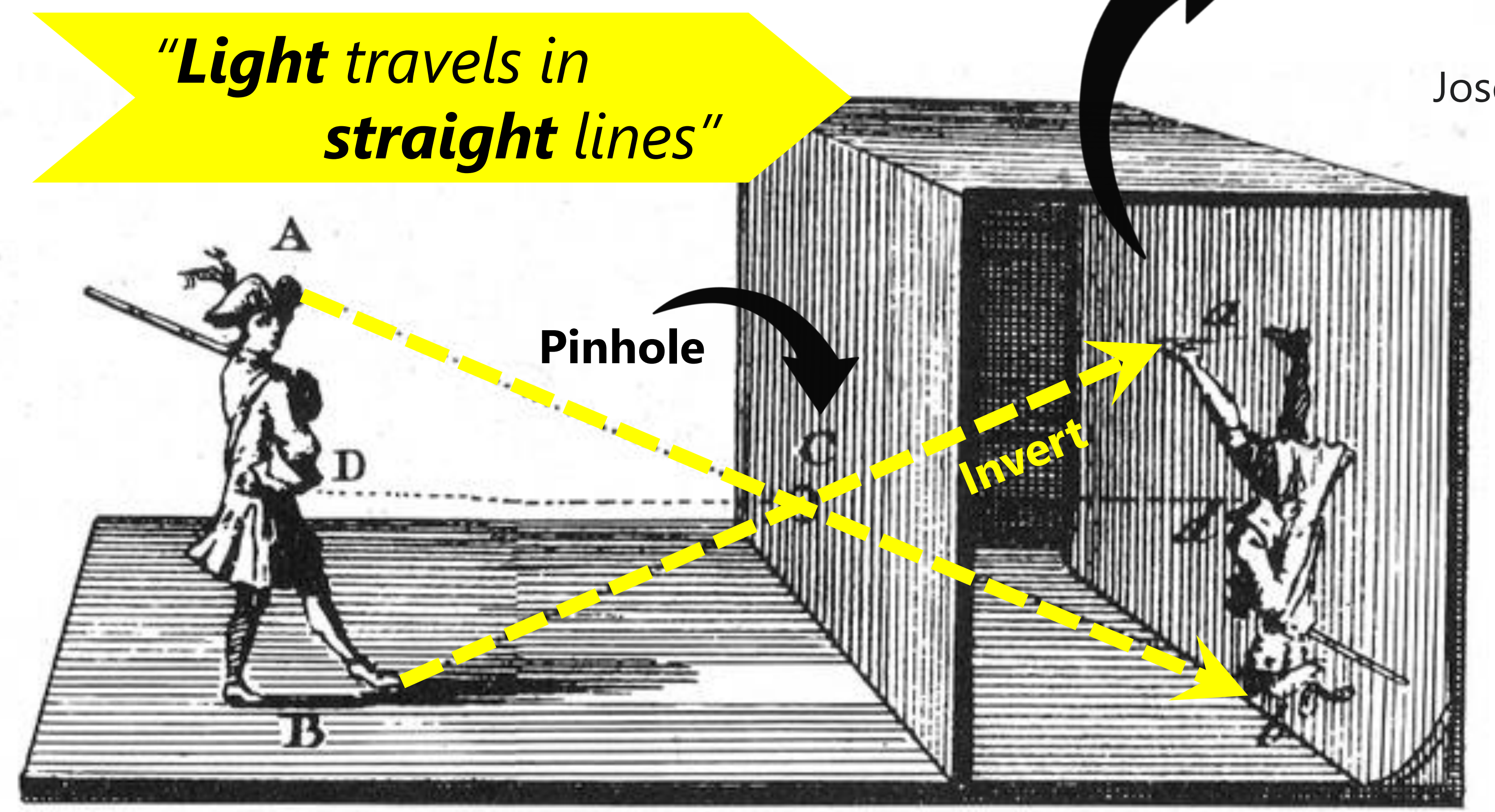
KENTECH
Korea Institute of Energy Technology

Brief Overview of Pinhole Camera Model

Camera Obscura – *Darken Room*

Camera (Latin) **= room or chamber**

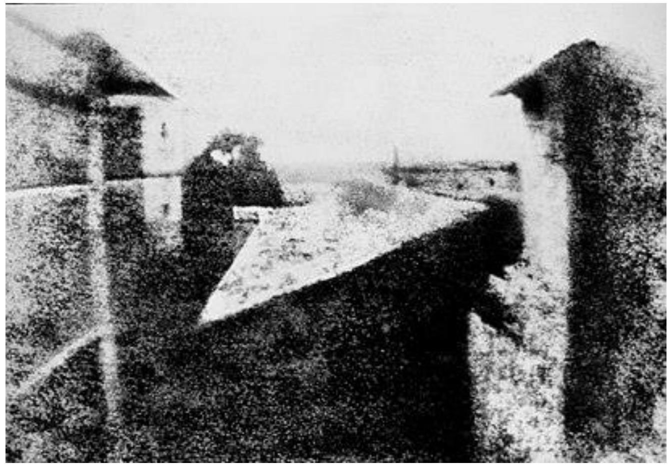
Obscura (Latin) **= dark**



Photosensitive Surface

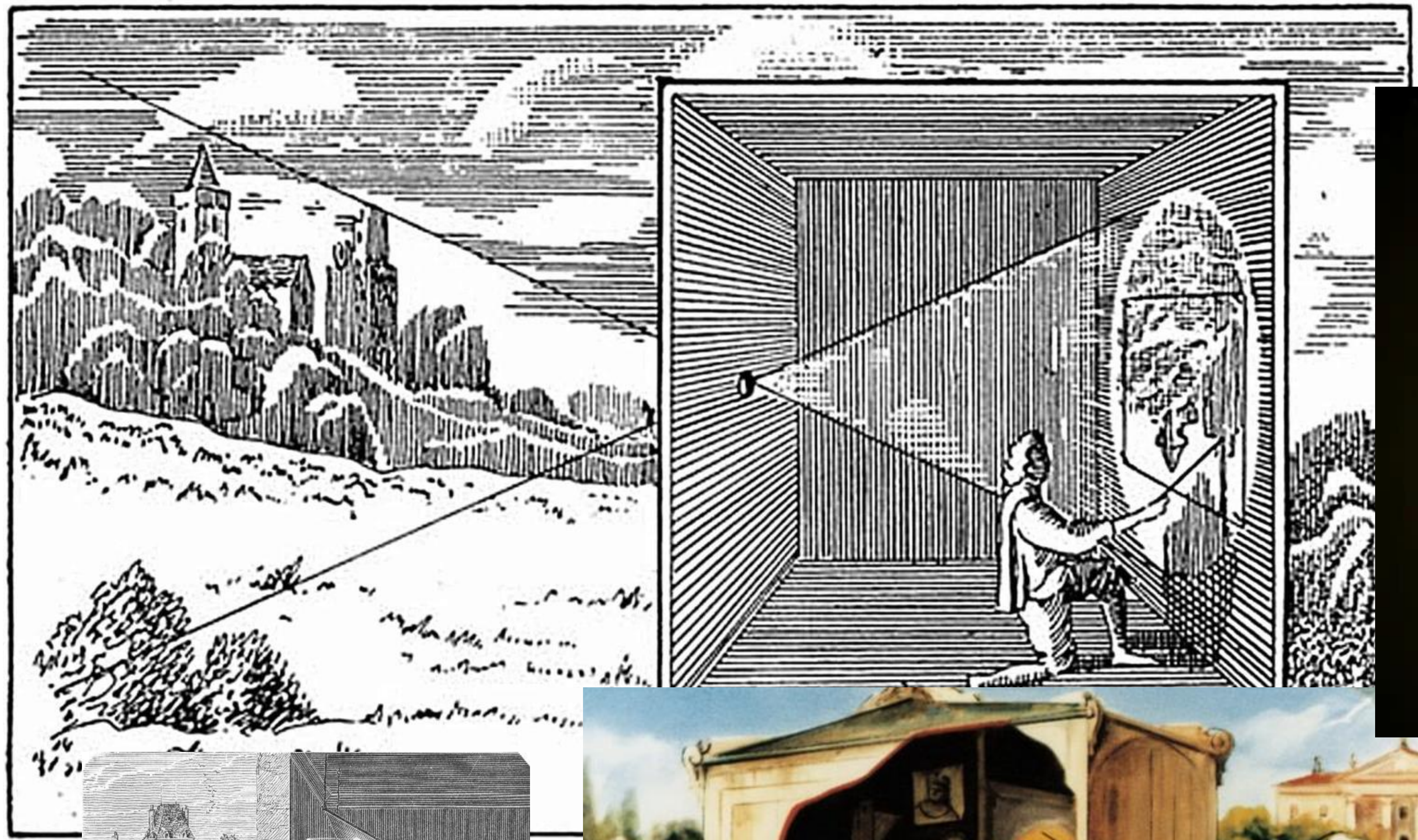
Joseph Nicéphore Niépce
(1765 – 1833)

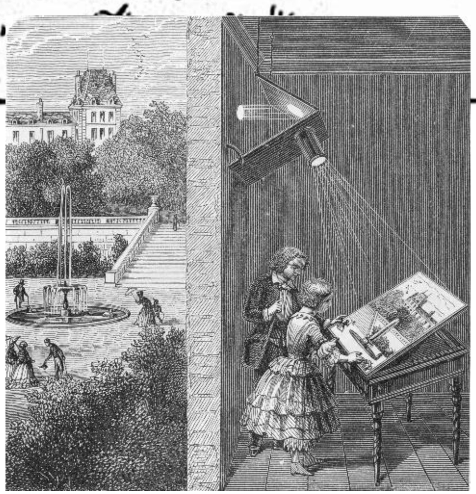
The inventor of photography



The first photograph
→ It took 8 hours

Camera Obscura – Based on *Art*





Amazing, cheap, fun way to experiment during COVID-19 quarantine

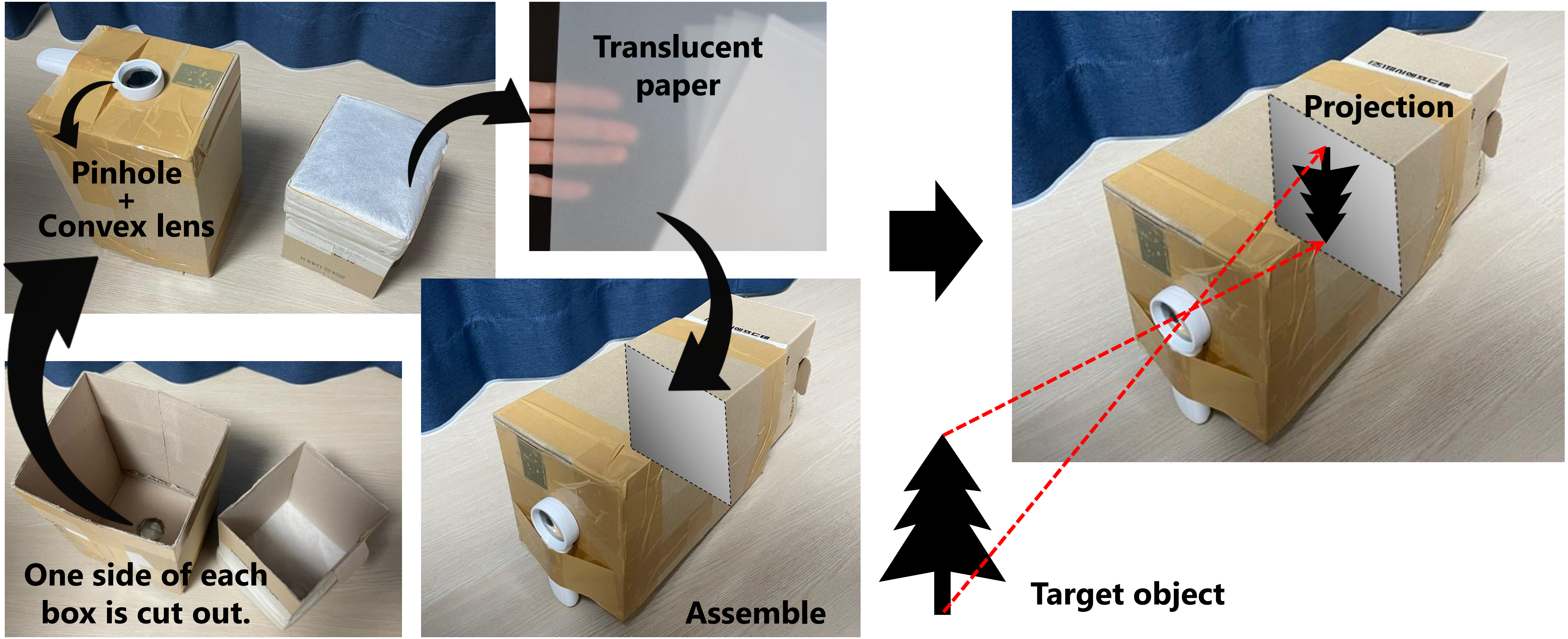
By Mathieu Stern

Inviting nature into your home!

Agents of Change: Camera Obscura (Art Land Magazine)

Camera Obscura – Making

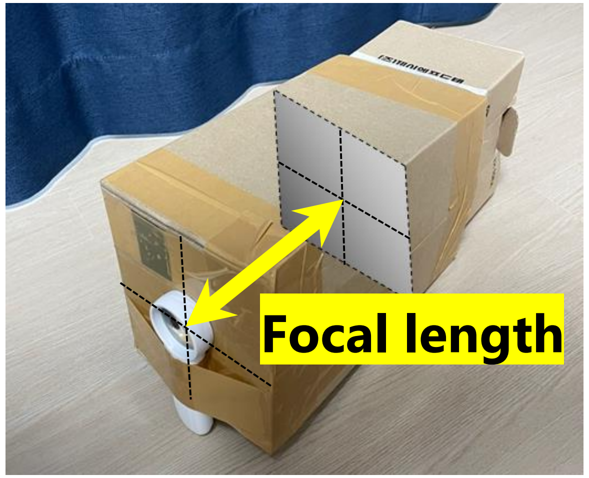
Preparation: two boxes, convex lens, translucent paper

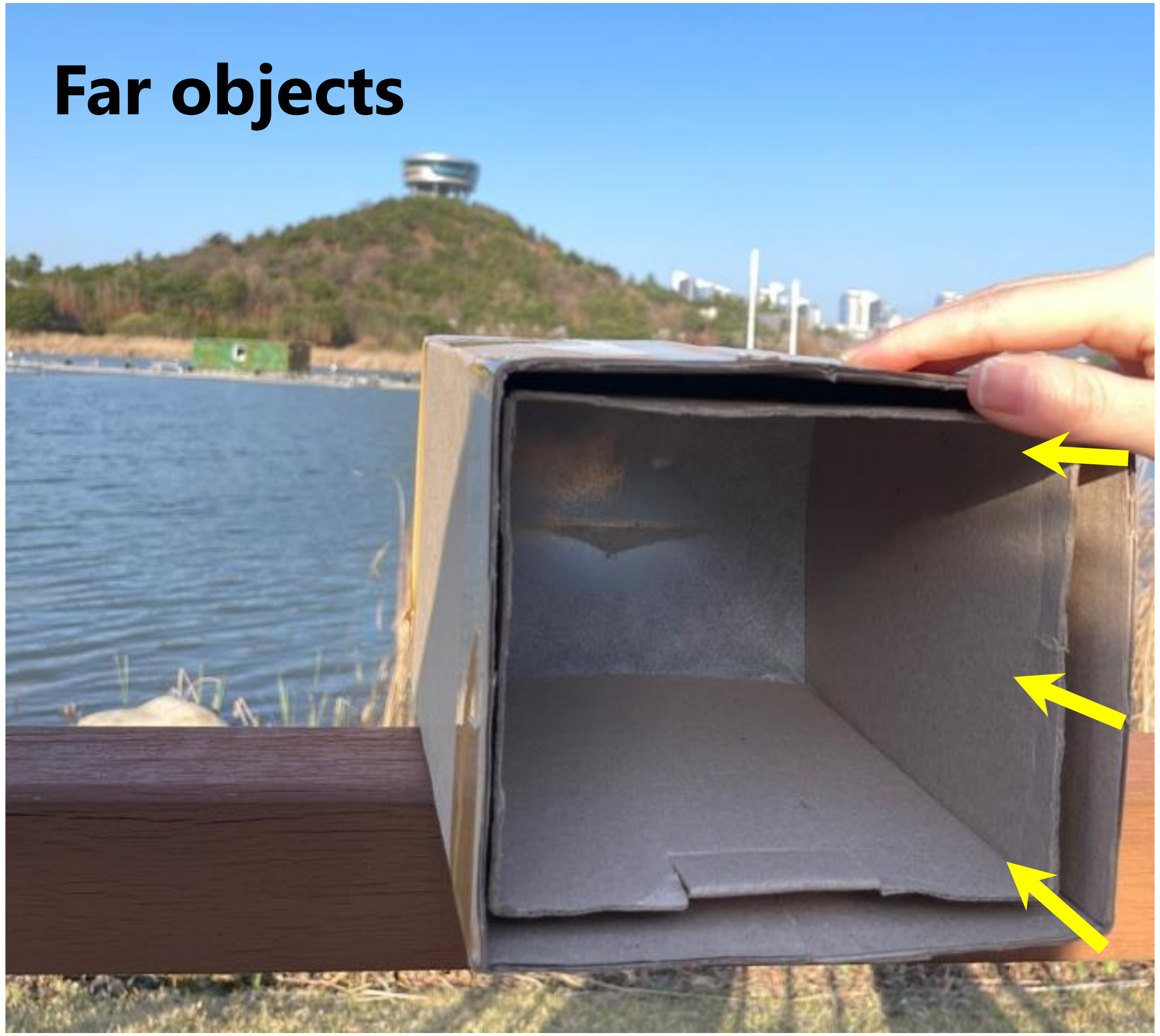


Camera Obscura – Experiments

Camera Obscura – About Focal Length

Near objects





→ **Near** objects: focal length ↑

→ **Far** objects: focal length ↓

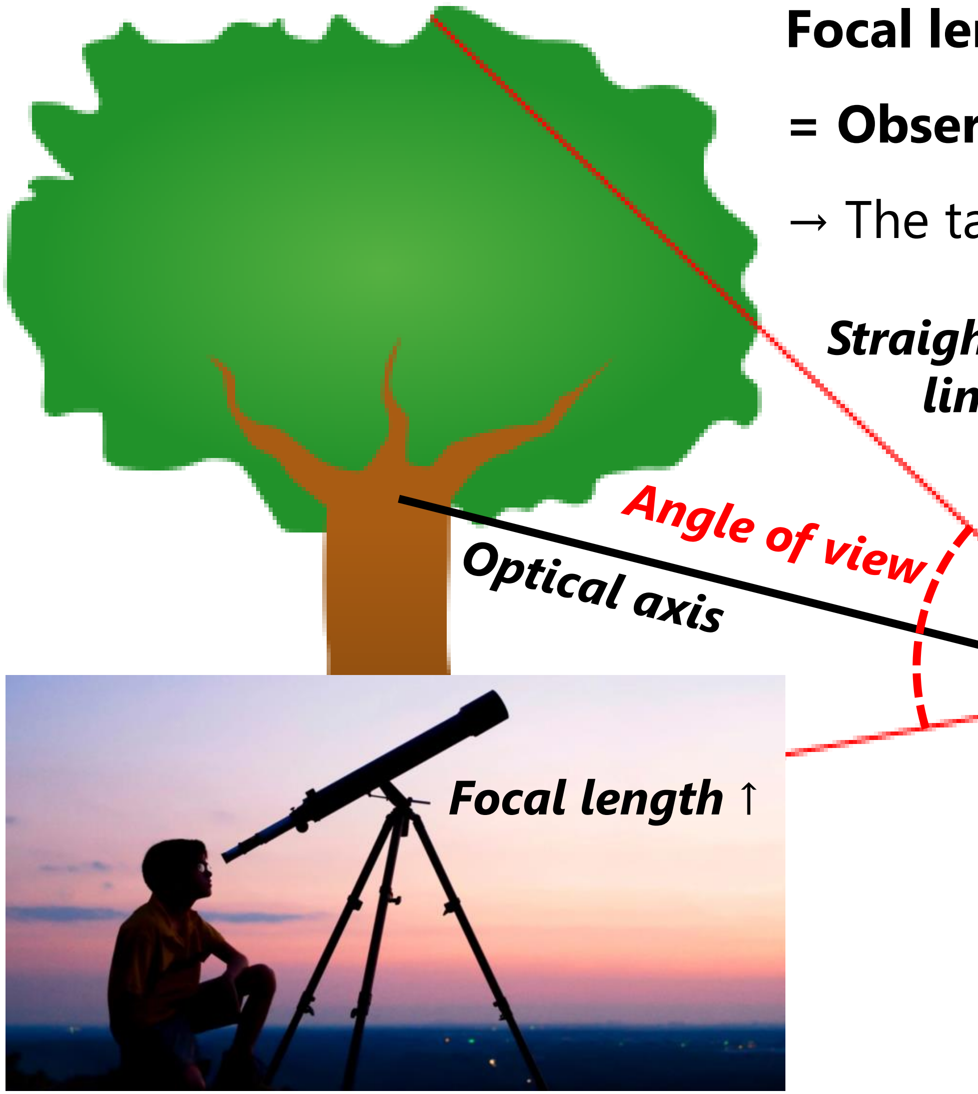
Pinhole Camera Model

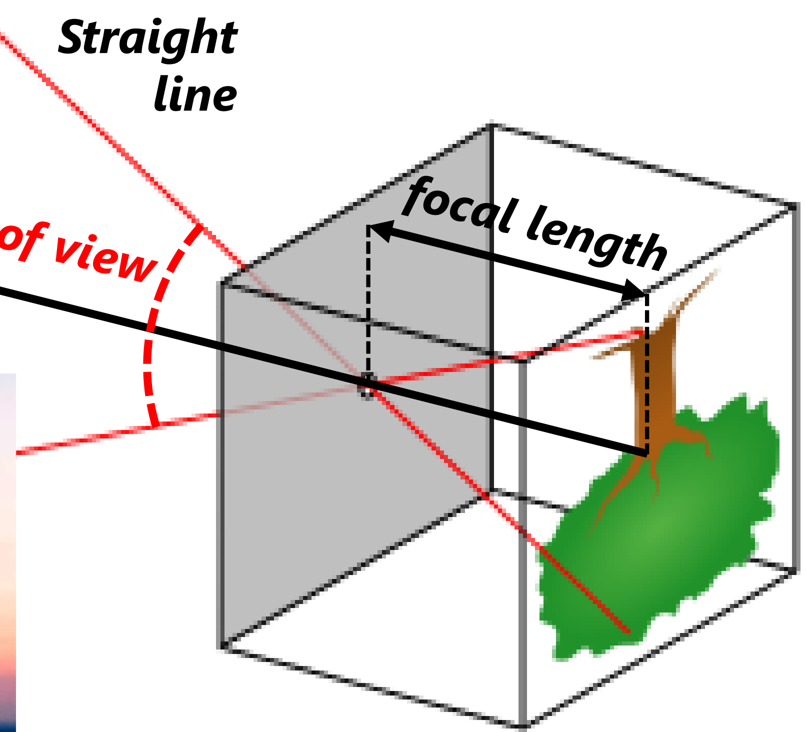
Simplified example of ray optics

Focal length ↑ = Angle of view ↓

= Observation of local areas

→ The target subject is magnified!





We will make it!

***Resolution** (for image, "spatial")

→ The number of pixels in each dimension

***Frame rate** (for video, "temporal")

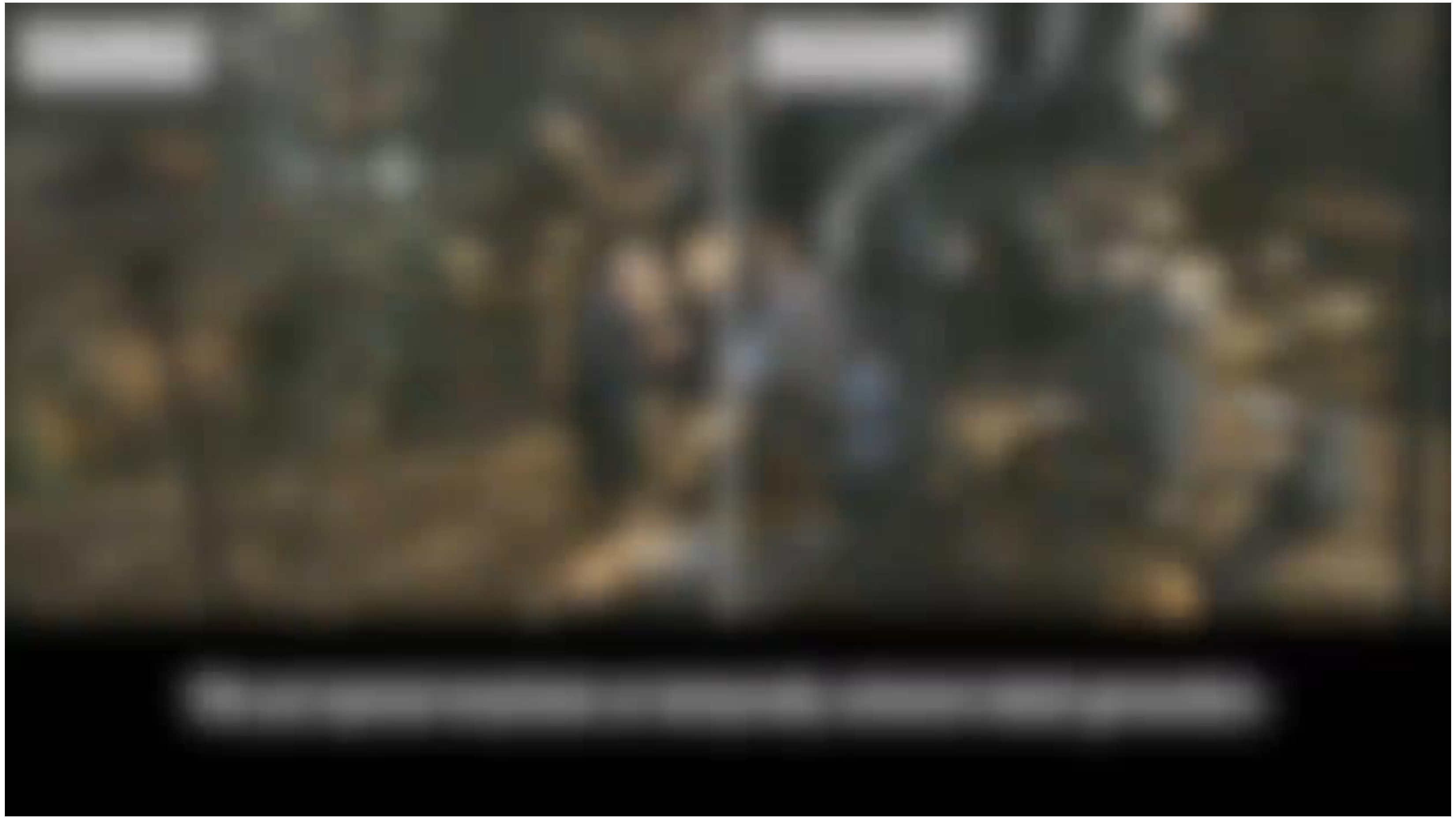
→ The number of image frames per second

Screen = Photographic film = CCD sensor

About Resolution

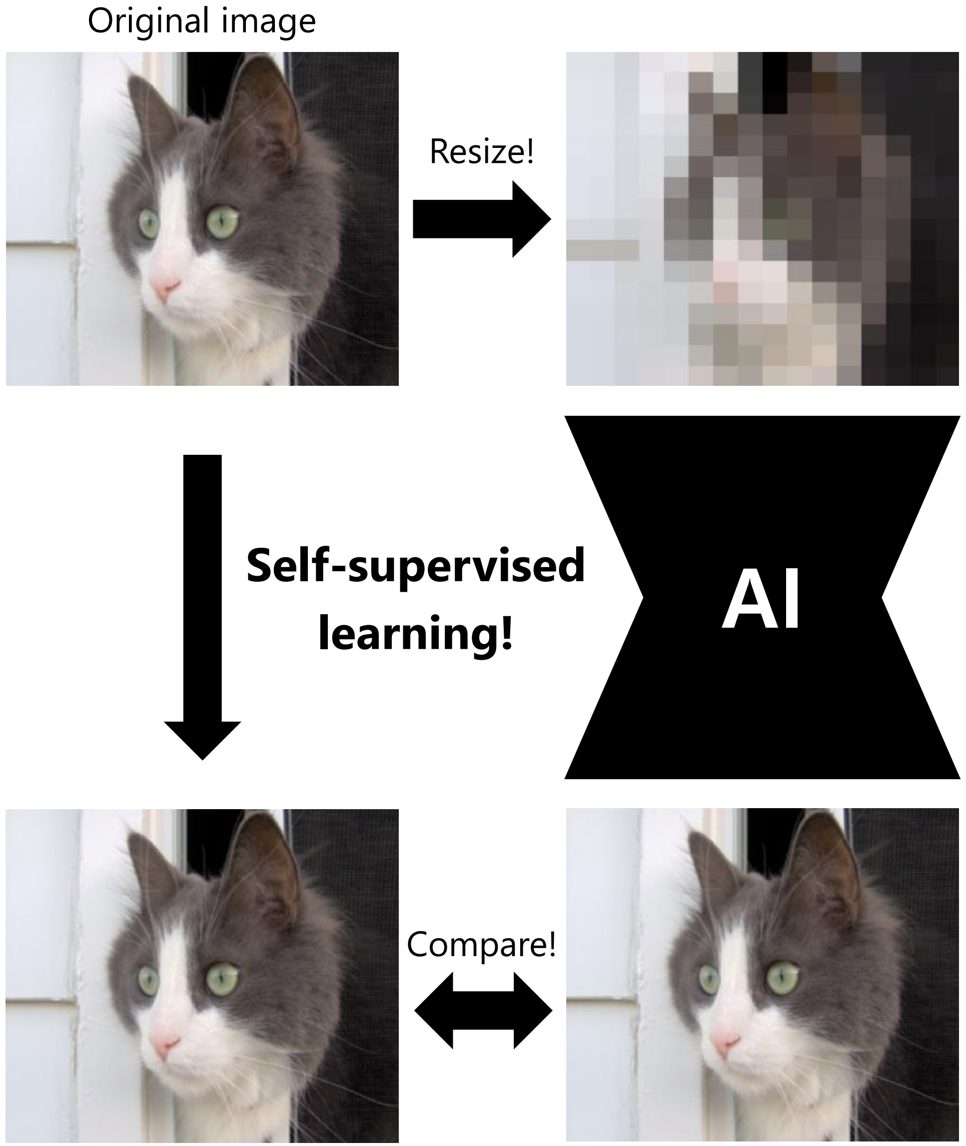
Example of deep learning application:

→ Image super resolution, "**spatial** processing"



TecoGAN (*SIGGRAPH'19*)

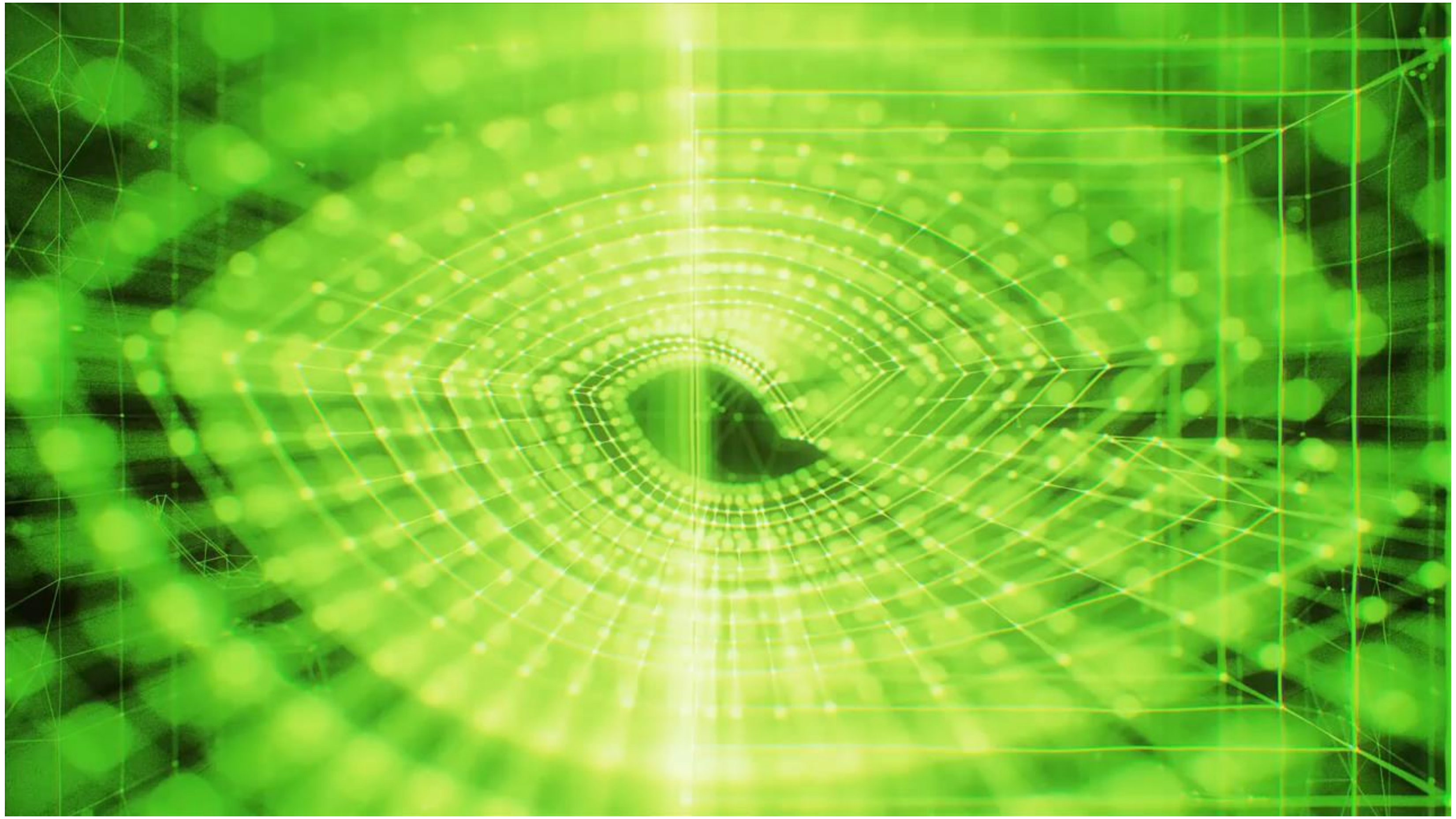
How to train?



About Frame Rate

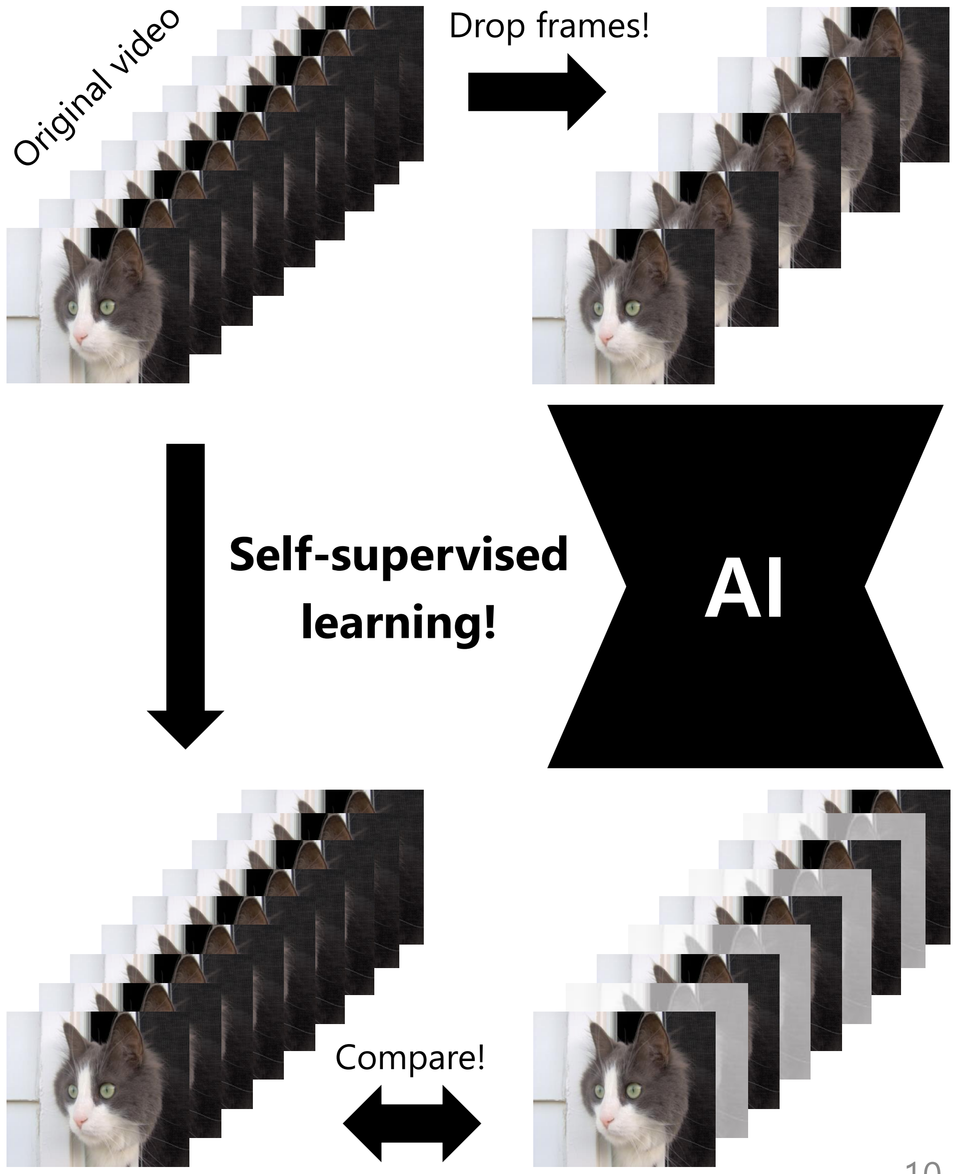
Example of deep learning application:

→ Video frame interpolation, “**temporal** processing”



Transforming Standard Video Into Slow Motion with AI
(Research at NVIDIA, 2018)

How to train?



About Shutter Speed

A.k.a., exposure time:

The length of time that the digital sensor inside the camera is exposed to light

→ **How shutter speed affects motion.**

"The faster your shutter speed,

the sharper your image will be."

mrs-cook.weebly.com

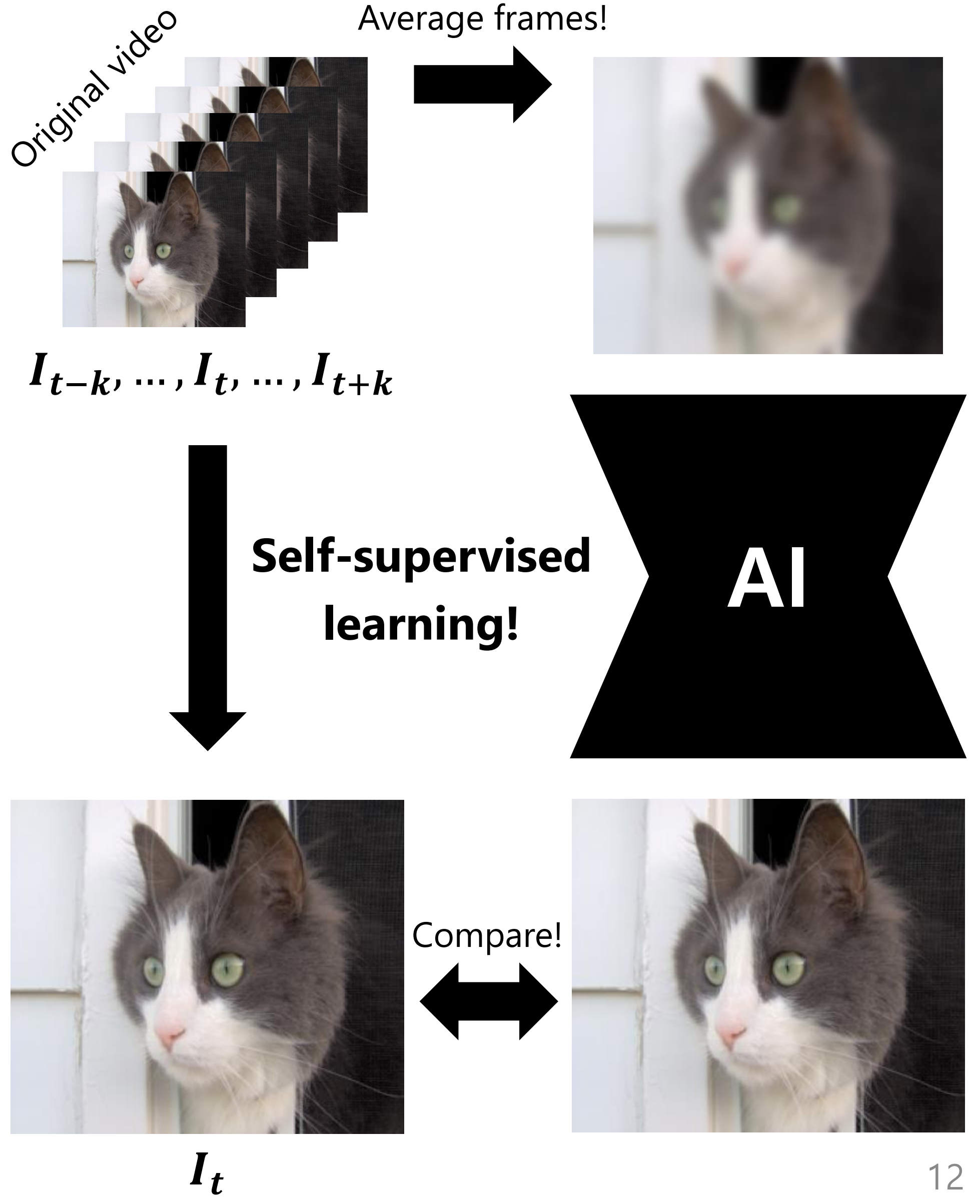
About Shutter Speed

Example of deep learning application:

→ Motion deblurring

Fix Blurry Photos with Motion Deblur AI (by AKVIS, 2021)

How to train?

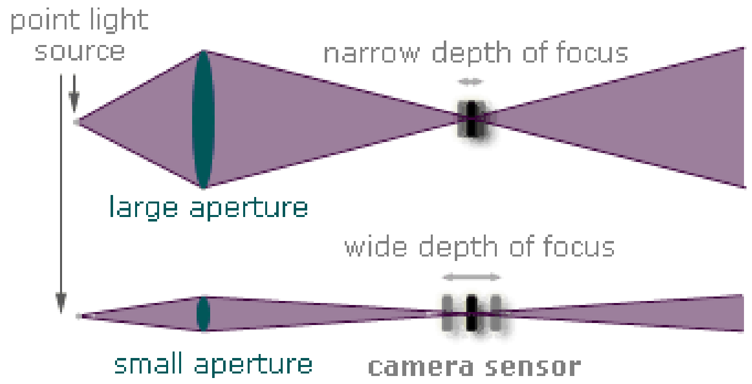


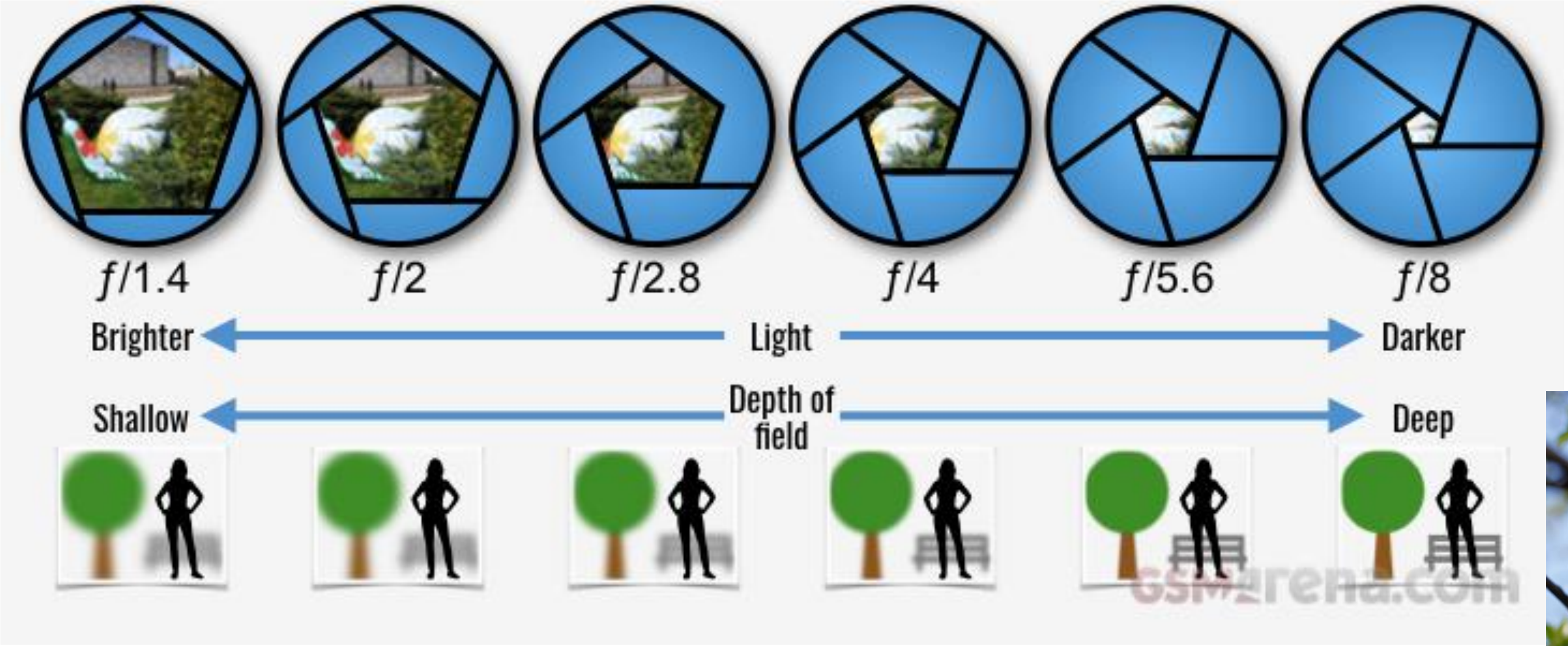
Aperture

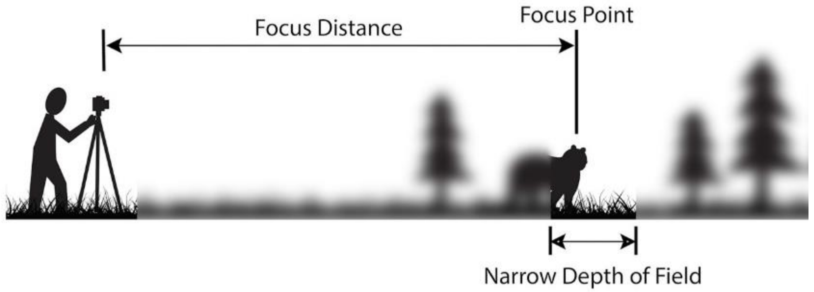
A hole through which light travels

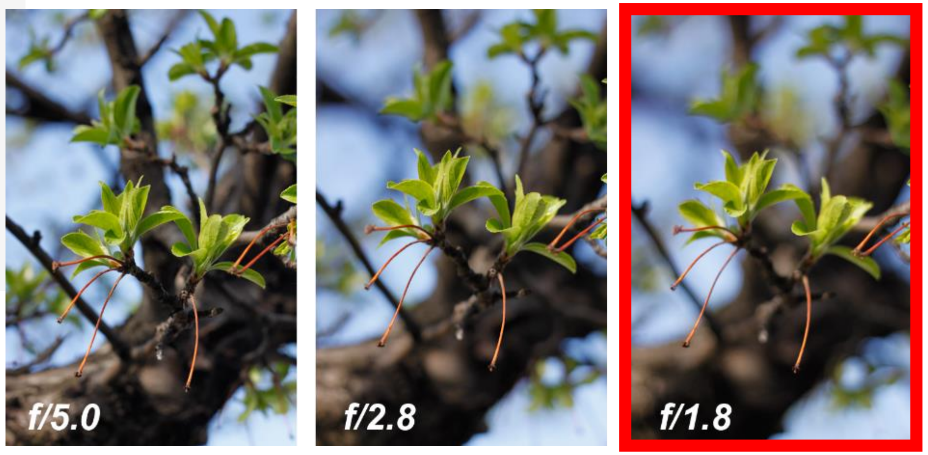
→ Controls the amount of the light

→ Controls depth-of-field (DoF)









Out focus
= **shallow** depth of field

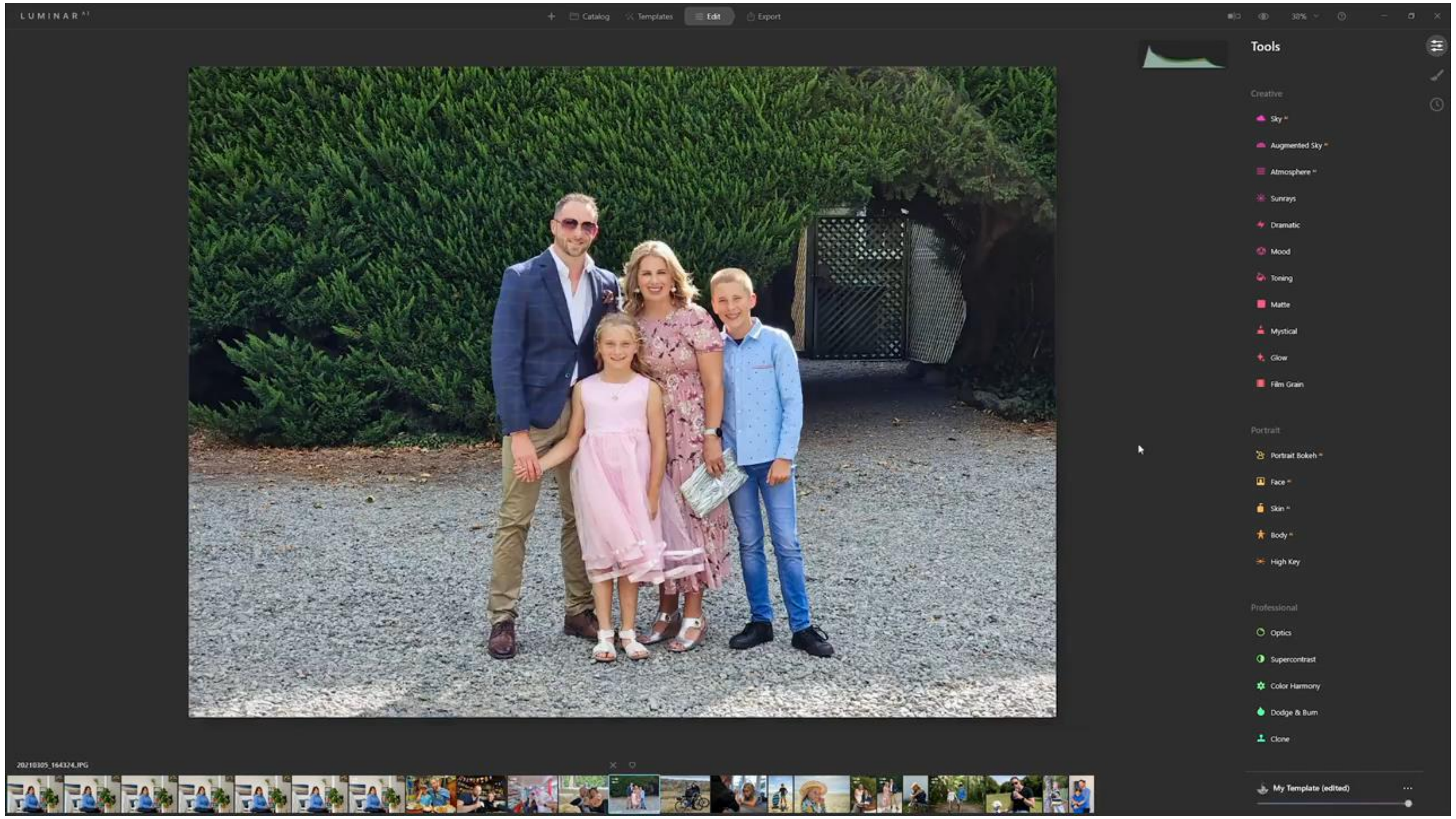
→ Helps concentrate on the subject!

[1] gsmarena.com [2] boostyourphotography.com [3] cambridgeincolour.com

Out Focusing

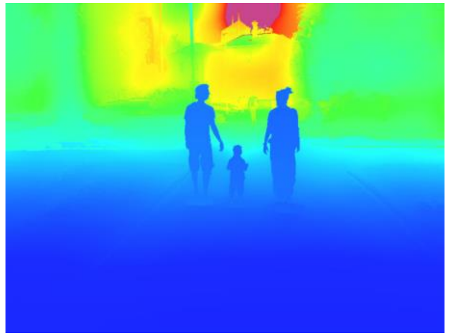
Technically difficult for smartphone cameras

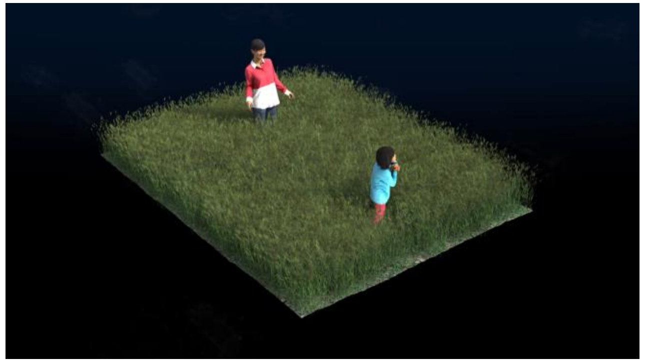
→ Computer vision & AI make this possible!



Automatic bokeh by Luminar AI (2021)

→ AI model for binary **segmentation**





→ All we need is just a **depth!**

[1] Liu, et al., “Bokeh Effects Based on Stereo Vision” (2015)

Summary of a Basic Camera Model

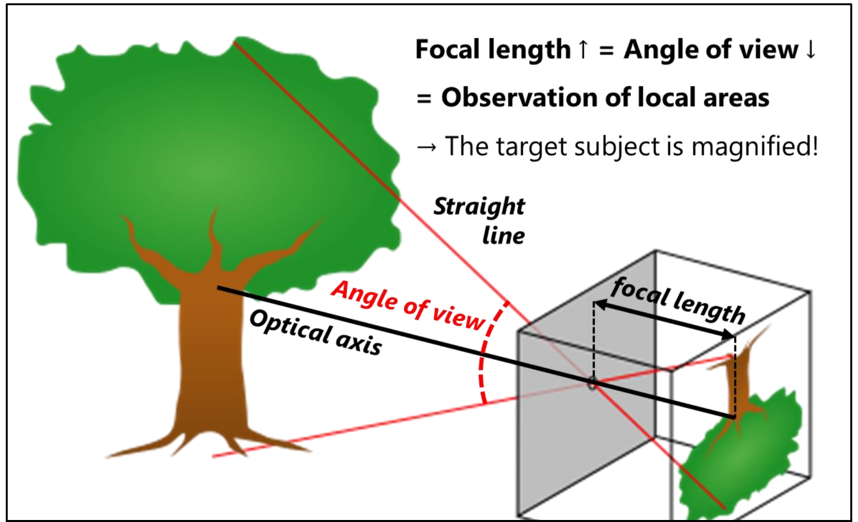
Characteristics of a pinhole camera

- Camera obscura
- Focal length
- Aperture & depth of field
- Shutter speed & exposure time

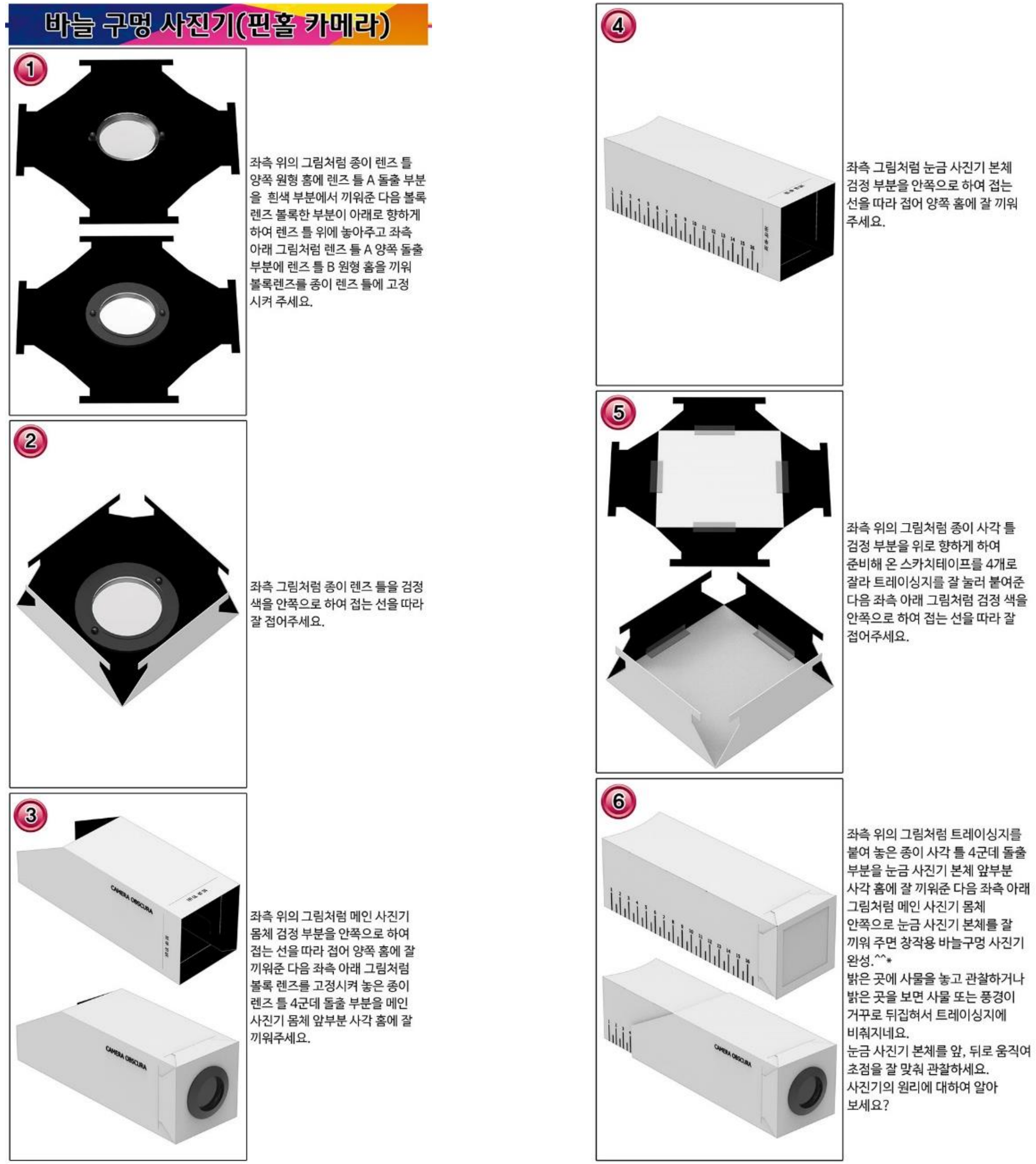
What AI can enhance images and videos

- Self-supervised learning
  - Image super resolution
  - Video frame interpolation
  - Motion deblurring
- Segmentation and depth
- 3D photography

Let's Make a Pinhole Camera



→ Experience the characteristics of a pinhole camera.



Discussion

<Pinhole Camera Quiz>
True or False?

Q1. Decreasing the focal length makes the object <u>smaller</u>. [True / False]

Q2. Increasing the focal length makes the field-of-view <u>smaller</u>. [True / False]

Q3. Increasing the size of aperture makes the photo <u>darker</u>. [True / False]

Q4. Increasing the size of aperture makes the depth-of-field <u>shallower</u>. [True / False]

Q5. Decreasing the shutter speed makes the photo <u>sharper</u>. [True / False]

Q6. Increasing the exposure time makes the photo <u>brighter</u>. [True / False]

KENTECH
Korea Institute of Energy Technology

Camera Calibration

Camera Calibration?

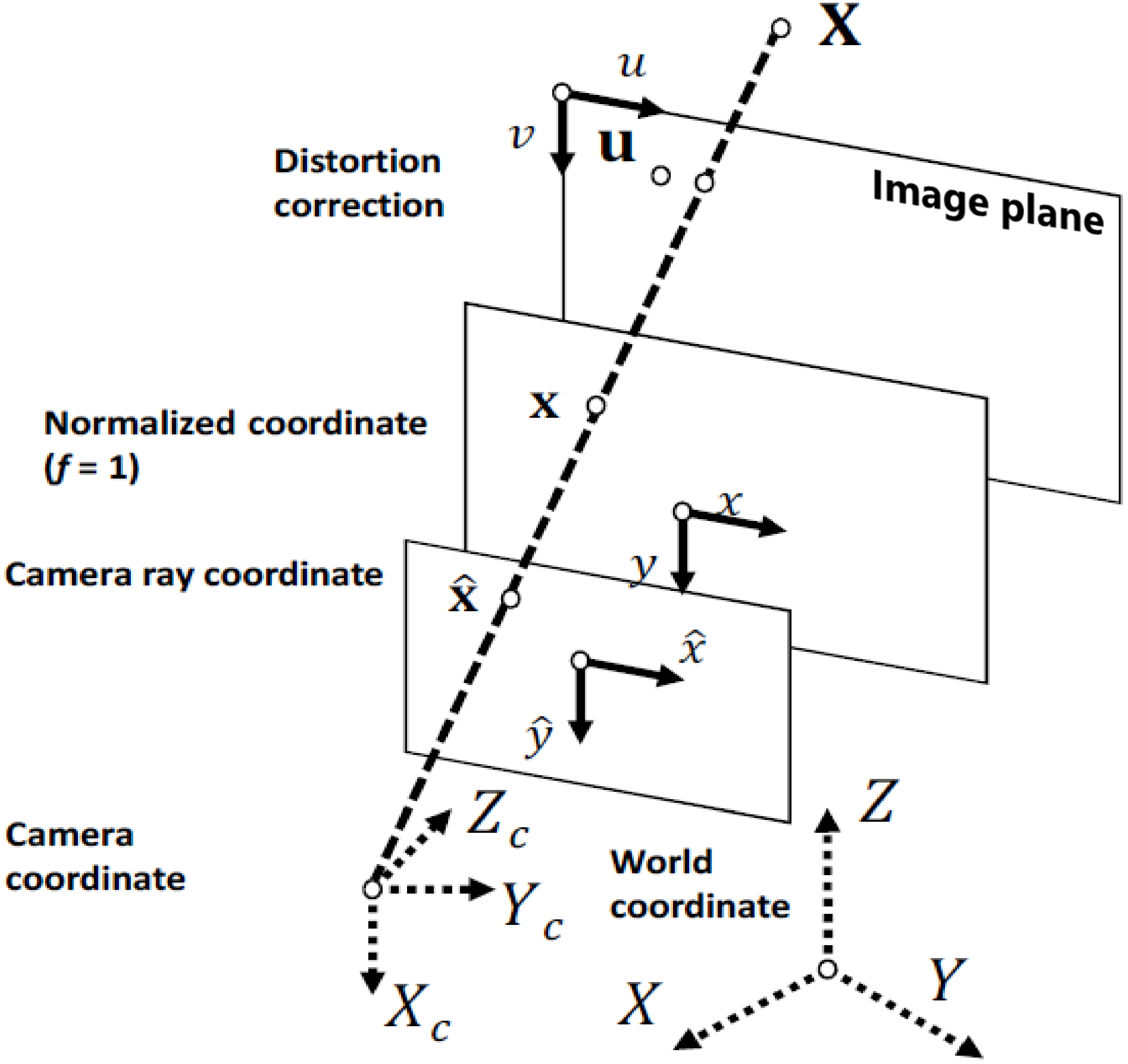
- ✓ Process of estimating the **parameters** of a pinhole camera model.
- ✓ Determines which **incoming light (3D)** is **associated with** each pixel on the resulting **image (2D)**.
- ✓ In an ideal pinhole camera, **a simple projection matrix** is enough to do this.
- ✓ The camera projection matrix is derived from the **intrinsic** and **extrinsic** parameters of the camera.

$$s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \text{skew\_c}f_x & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= K[R|t]\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})$: 3D point in the world coordinate
- $[\boldsymbol{R}|\boldsymbol{t}]$: extrinsic parameters to convert the world coordinate into the camera coordinate
- $\boldsymbol{K}$: intrinsic parameters to represent the camera characteristics
- $\boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]$: camera projection matrix
- $(\boldsymbol{x}, \boldsymbol{y})$: 2D pixel location in the image plane
- $\boldsymbol{s}$: scale factor

Camera Geometry: 3D → 2D



$$\mathrm{X}_C = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \mathrm{X}$$

$$(X_C, Y_C, Z_C)^T \mapsto (\hat{x}, \hat{y}) = \left(\frac{fX_C}{Z_C}, \frac{fY_C}{Z_C}\right)^T$$

$$\hat{\mathrm{x}} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ f \end{bmatrix} = \frac{1}{Z_C} \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & f & 0 \end{bmatrix} \begin{bmatrix} \mathrm{X}_C \\ 1 \end{bmatrix}$$

$$f = 1$$

$$\mathrm{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = [R \mid T]X$$

$$\mathbf{u} = \mathrm{K}\mathrm{x}$$

$$\mathbf{u} = \mathbf{K}[\mathbf{R}|\mathbf{T}]\mathbf{X}$$

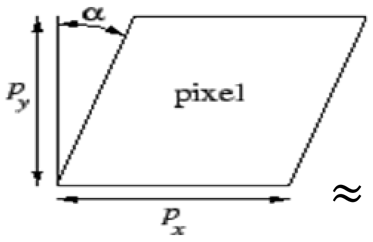
Intrinsic Parameters

✓ The intrinsic matrix ***K*** contains 5 intrinsic parameters of the specific camera model.

→ Focal length: f_x, f_y

→ Principal point: c_x, c_y

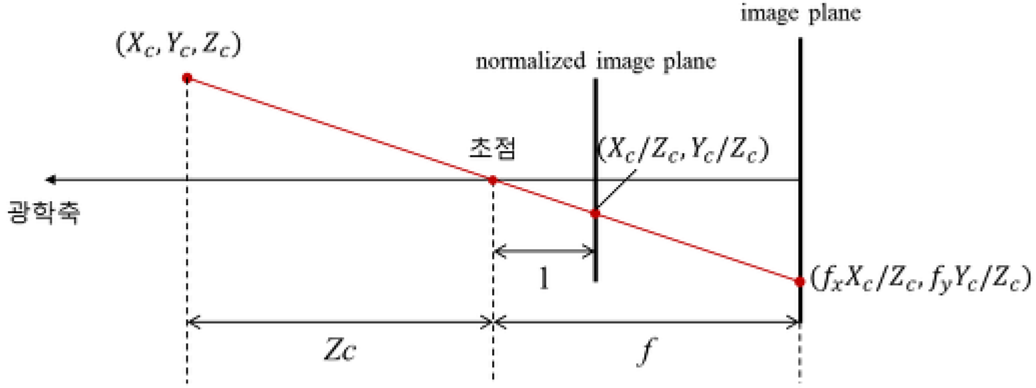
→ Skew coefficient (y-axis slope of the image sensor): $skew_c = \tan(\alpha)$



≈ 0

$$s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \mathrm{skew\_c}f_x & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= K[R|t]\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



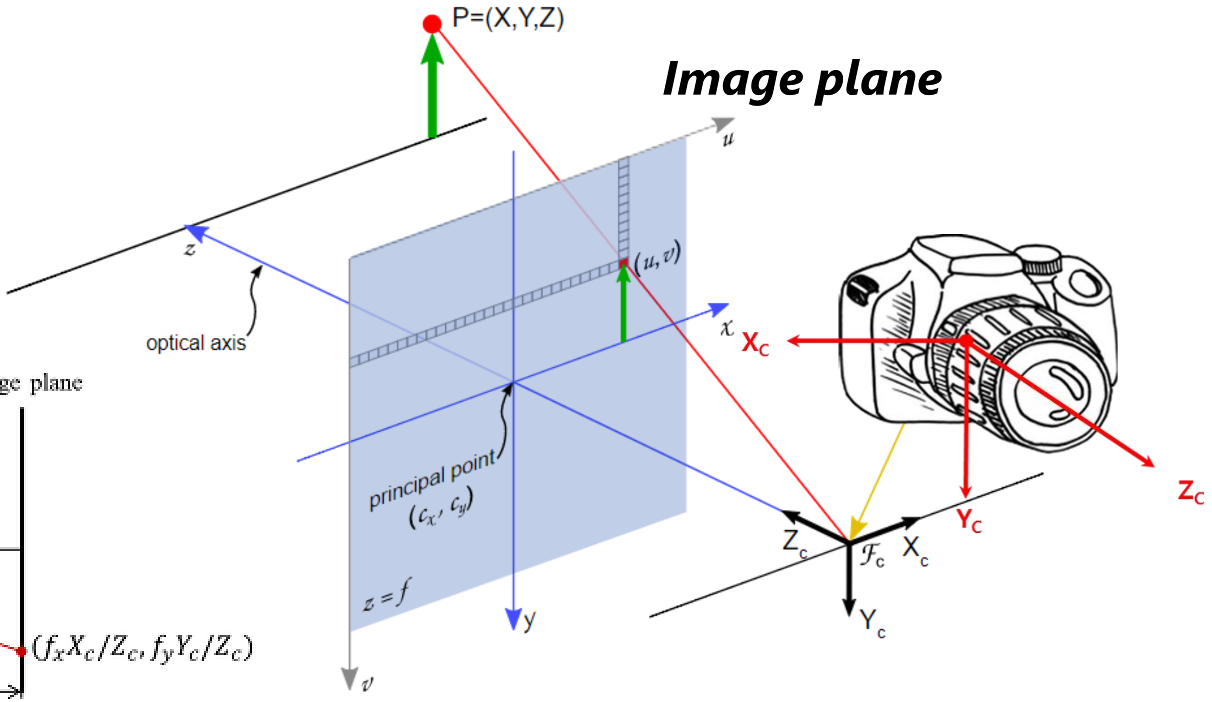


Figure by NVIDIA

Extrinsic Parameters

✓ Transformation matrix from 3D world coordinates to 3D camera coordinates

→ Rotation ($\boldsymbol{R}$, 3×3 matrix) + Translation ($\boldsymbol{t}$, 3×1 matrix)

$$s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \mathrm{skew\_c}f_x & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= K[R|t]\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

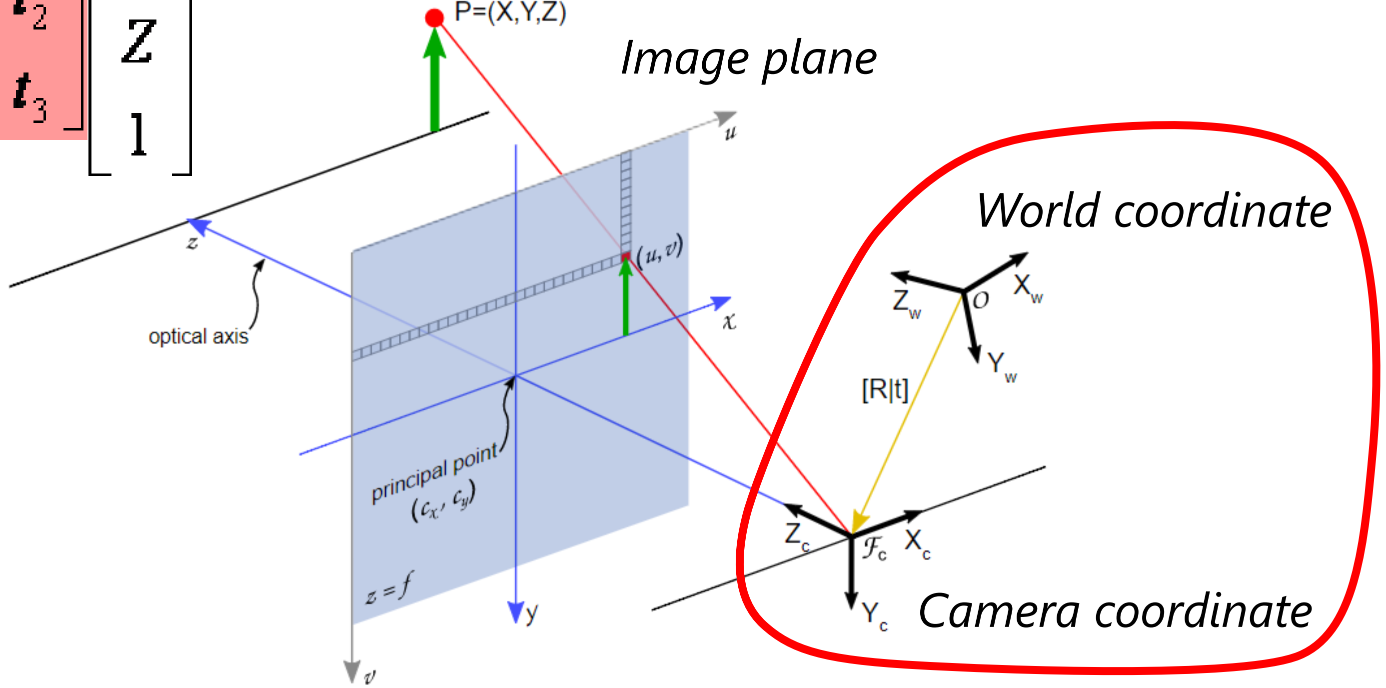
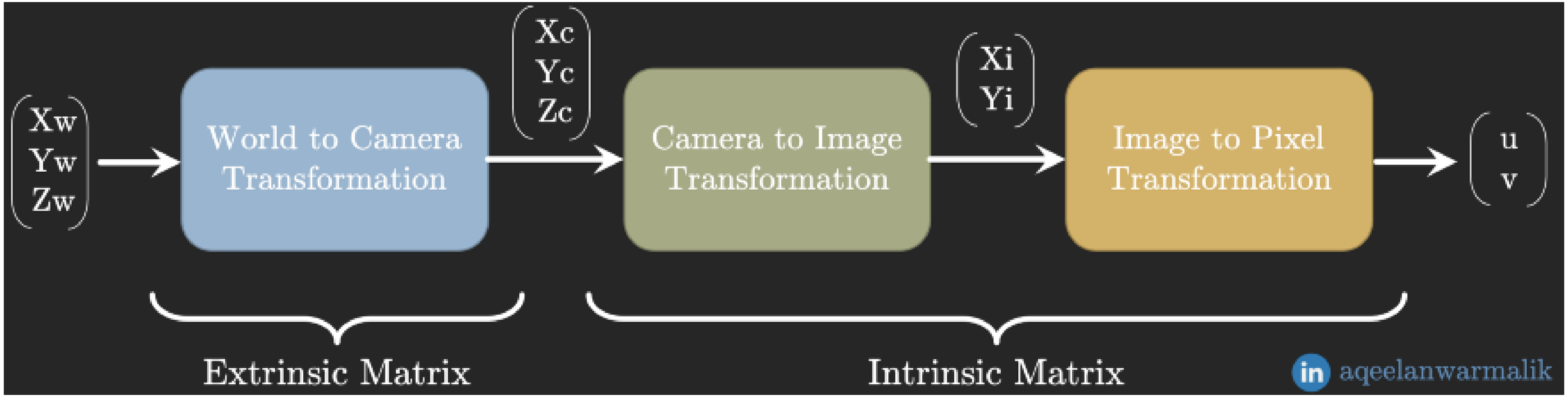


Figure by NVIDIA

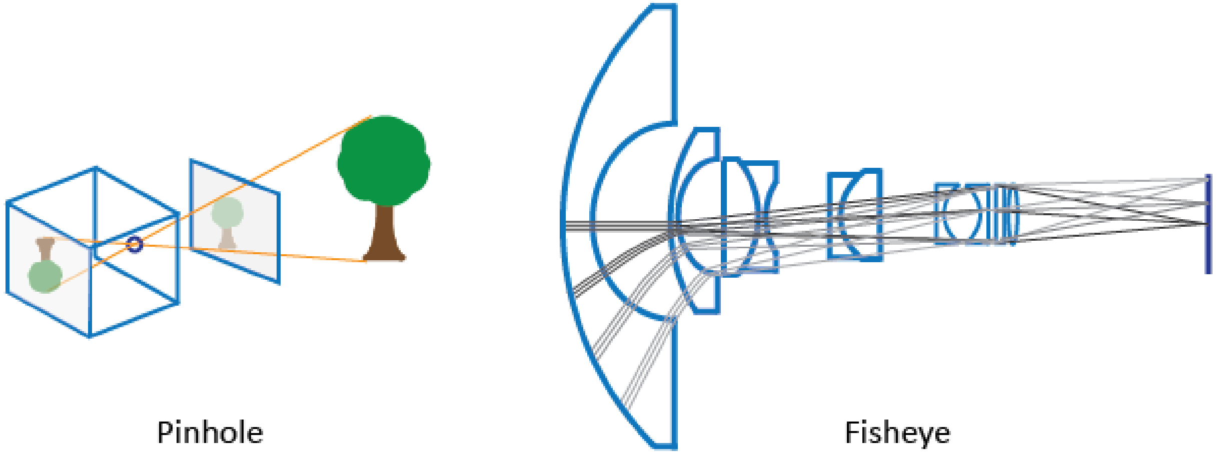
Summary: 3D to 2D Projection

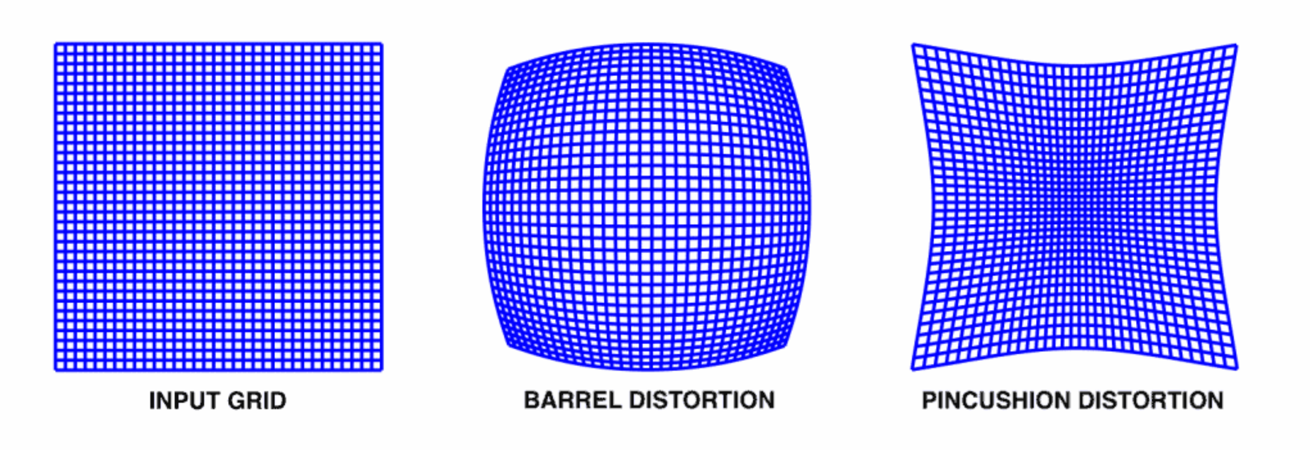
1. **World-to-Camera**: 3D-3D projection. Representation of Rotation + Scaling + Translation.
   → **Camera extrinsic matrix**: depends on the position and orientation of the camera.
2. **Camera-to-Image**: 3D-2D projection. Loss of information. Requires pinhole camera model and its parameters.
   → **Camera intrinsic matrix**: depends on camera properties (e.g., focal length, pixel dimensions, resolution, etc.)
3. **Image-to-Pixel**: 2D-2D projection. Continuous to discrete. Quantization + Origin shift.



+ Camera Lens Distortion

✓ For the wider Field-of-View (FoV), various types of lens are used.





Different types of the radial distortion

Distorted image with wide FoV

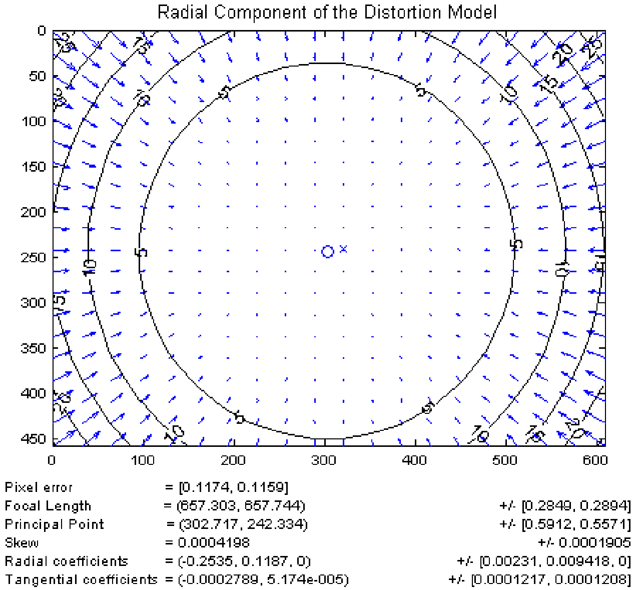
+ Camera Lens Distortion → Undistortion

✓ Before camera calibration, we need to correct the image distortion.

Undistortion

Radial distortion model

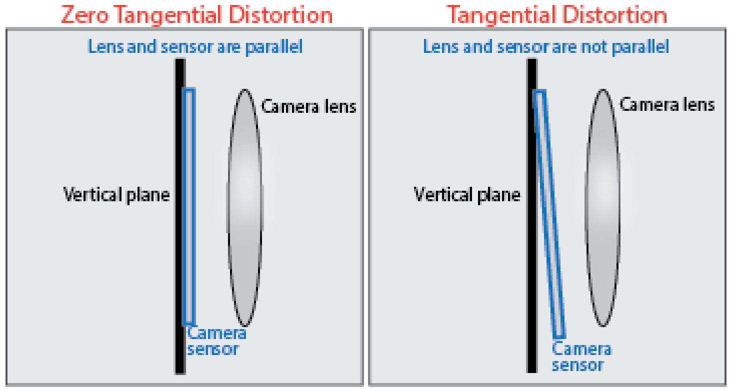
$$x_{\text{corrected}} = x\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right)$$
$$y_{\text{corrected}} = y\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right)$$



Tangential distortion model

→ caused by **unparallel** *lens* and *image plane*!

$$x_{\text{corrected}} = x + \left[2p_1 xy + p_2\left(r^2 + 2x^2\right)\right]$$
$$y_{\text{corrected}} = y + \left[p_1\left(r^2 + 2y^2\right) + 2p_2 xy\right]$$

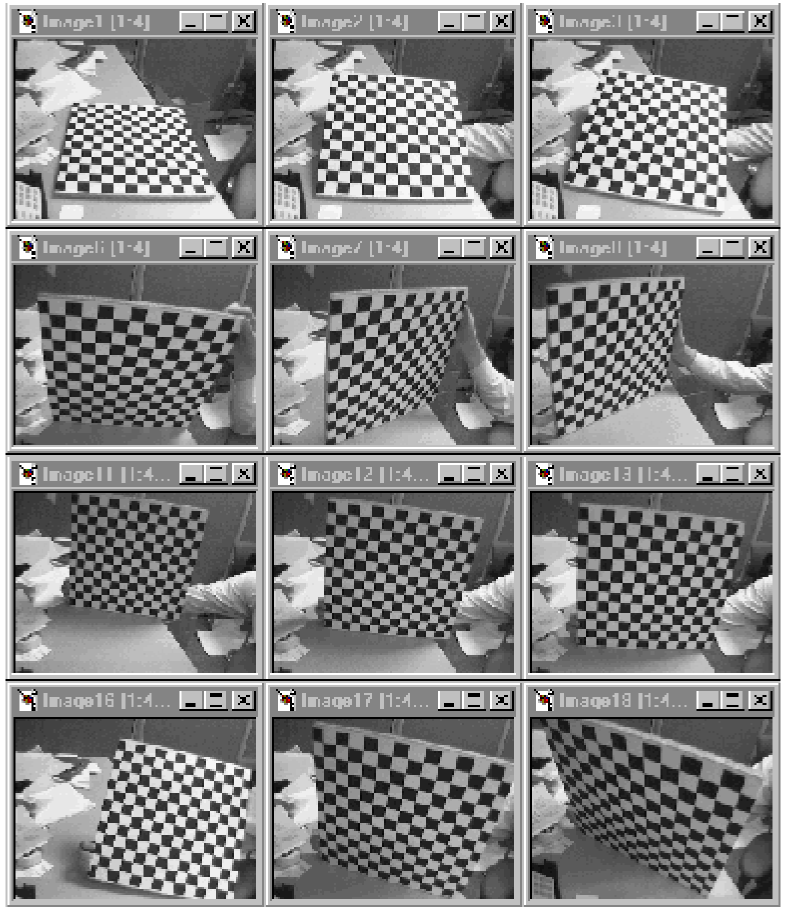


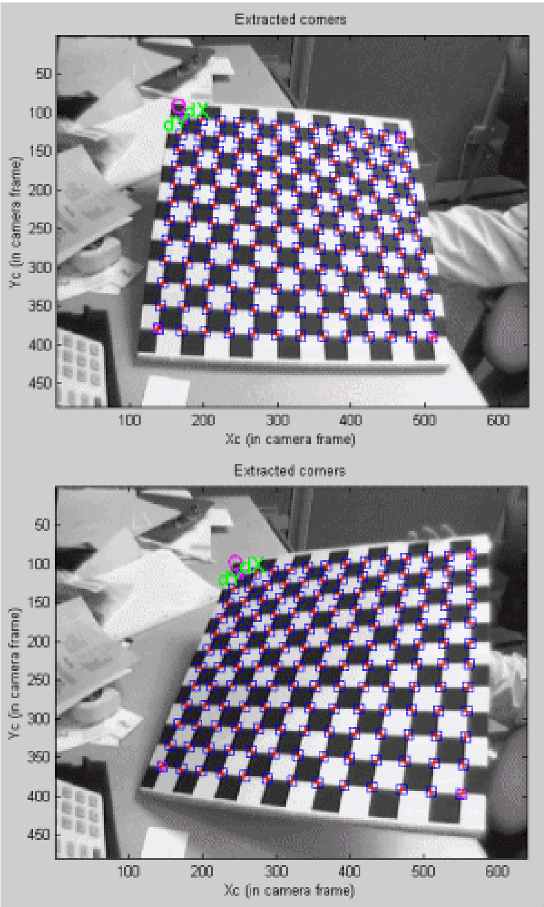
$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} x_u \\ y_u \end{bmatrix} + \left(k_1 r^2 + k_2 r^4 + k_3 r^6 + \cdots\right)\begin{bmatrix} x_u \\ y_u \end{bmatrix} + \begin{bmatrix} 2p_1 x_u y_u + p_2\left(r^2 + 2x_u^2\right) \\ p_1\left(r^2 + 2y_u^2\right) + 2p_2 x_u y_u \end{bmatrix},$$

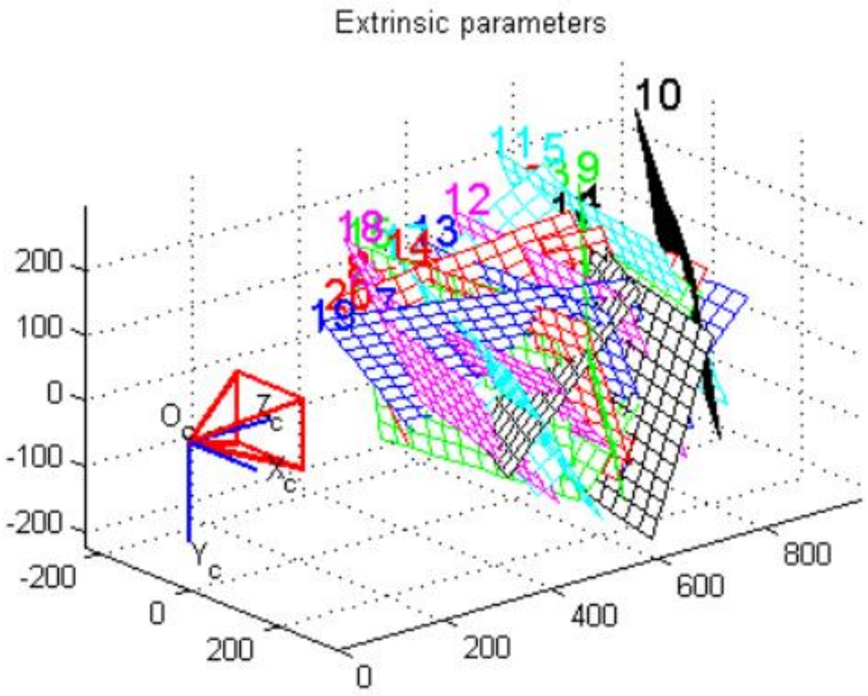
Let's Try Camera Calibration!

Camera calibration code is available at:
https://view.kentech.ac.kr/1d2a3167-672f-4456-a5b8-b8c56c15eefe

- ✓ Camera Calibration Toolbox [1]








```
Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 665.77415   665.77415 ] ± [ 2.42637   2.42637 ]
Principal point:       cc = [ 319.50000   239.50000 ] ± [ 0.00000   0.00000 ]
Skew:             alpha_c = [ 0.00000 ] ± [ 0.00000  ]   => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:            kc = [ 0.00000   0.00000   0.00000   0.00000  0.00000 ]
Pixel error:          err = [ 1.09184   0.99454 ]

Note: The numerical errors are approximately three times the standard deviations (for reference).
```

[1] http://robots.stanford.edu/cs223b04/JeanYvesCalib/htmls/example.html

Next Contents

- ✓ Python practice for basic camera geometry & image transformation
- ✓ Basic feature extraction & matching

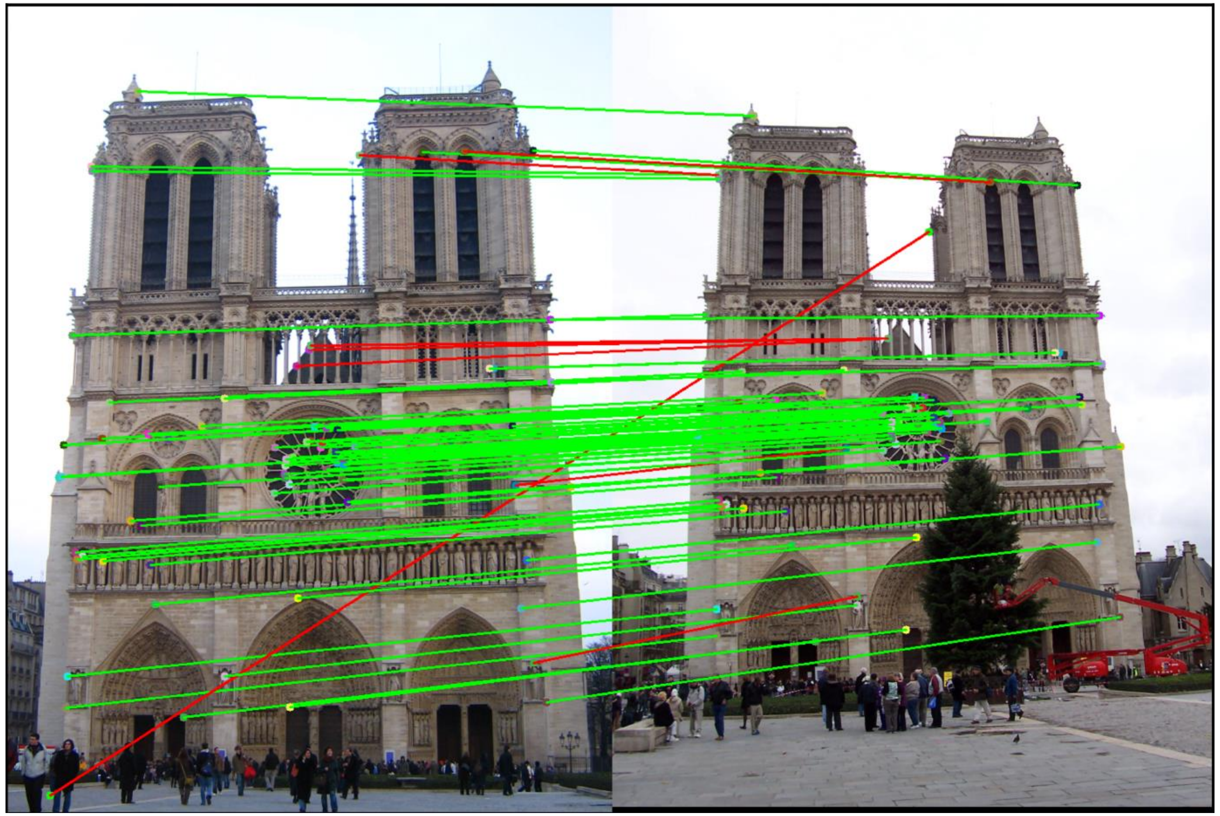


Figure by James Hays