



Camera Model

Graduate Course INTR-6000P

Week 2 - Lecture 4

Changhao Chen

Assistant Professor

HKUST (GZ)

Reading & Survey Task Release

Objective: This reading & survey assignment is designed to deepen your understanding of cutting-edge research in visual navigation. You will move beyond the foundational models and analyze how they are applied and advanced in real-world scenarios.

Select papers published in a top-tier conference or journal within the last 3-5 years.

Papers can fall into one of the following research areas:

- ✓ Visual SLAM
- ✓ Place Recognition
- ✓ Scene Representation
- ✓ LiDAR-Visual Navigation
- ✓ Visual-Inertial Navigation
- ✓ Embodied Navigation
- ✓ End-to-end Self-driving

Reading & Survey Task Release

Assignment: IEEE-Style Survey Paper

Formatting Requirements:

Template: Must use the official IEEE conference template. Please refer to the author guidelines and templates from recent IEEE International Conference on Robotics and Automation (ICRA) proceedings for examples.

Length: Maximum 8 pages, including all figures, tables, and references.

Layout: Double-column format, as standard for IEEE conferences.

Style: Follow all IEEE formatting guidelines for citations, figures, and sections.

Submission Deadline: Monday, Nov 10, 2025 (23:59 Anywhere on Earth)

Reading & Survey Task Release

Grading Rubric

- Understanding (40%): Accurate summary of the problem and method.
- Analysis (30%): Depth of critical thought in discussing metrics, results, and limitations.
- Clarity & Structure (20%): Organization, writing quality, and use of figures.
- Context (10%): Ability to relate the paper to the broader field of visual navigation.

Policy on the Use of Large Language Models (LLMs)

The generation of core content using LLMs is strictly prohibited. This includes the generation of original analysis, technical explanations, literature reviews, and conclusions.

Permitted use is limited to:

Proofreading and correcting grammar, spelling, and punctuation.

Polishing sentence structure and clarity for existing, self-written text.

Formatting assistance (e.g., ensuring citation style compliance).

Recap: Cameras

Monocular Camera:

Single lens, low-cost

Rich information

Limitation: lacks depth perception



Stereo Camera:

Mimics human eyes → depth from disparity

Useful for near-field obstacle detection

More expensive & calibration-sensitive



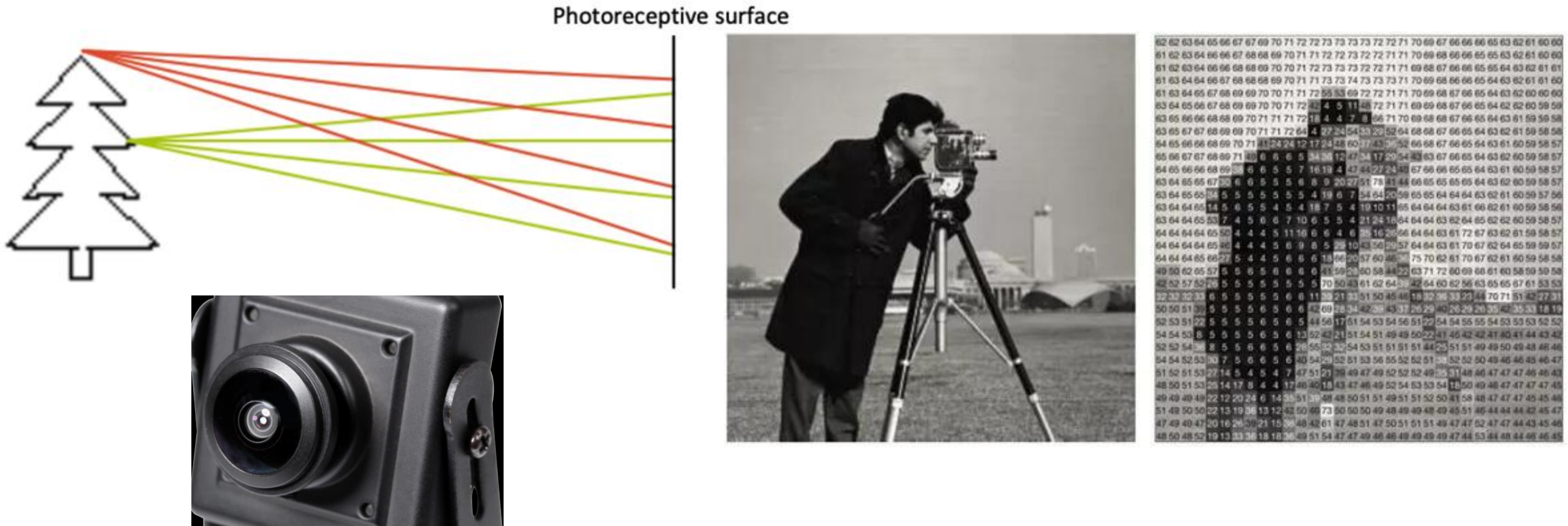
Sensor Placement:

Front-facing cameras - lane keeping, forward vision

Surround-view (4–8 cameras) - 360° coverage

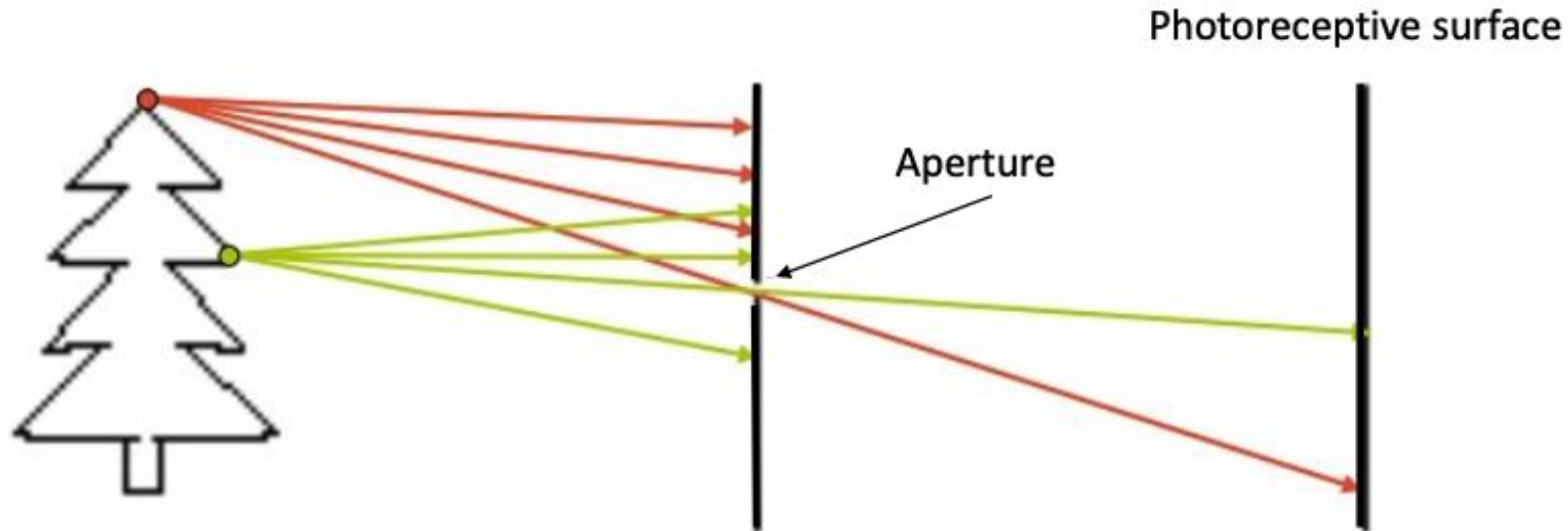
Cameras

- How to capture an image of the world
 - (Digital) Cameras -> capture light -> converted to digital image
 - Light is reflected by the object and scattered in all directions
 - if we simply add a photoreceptive surface, the captured image will be extremely blurred



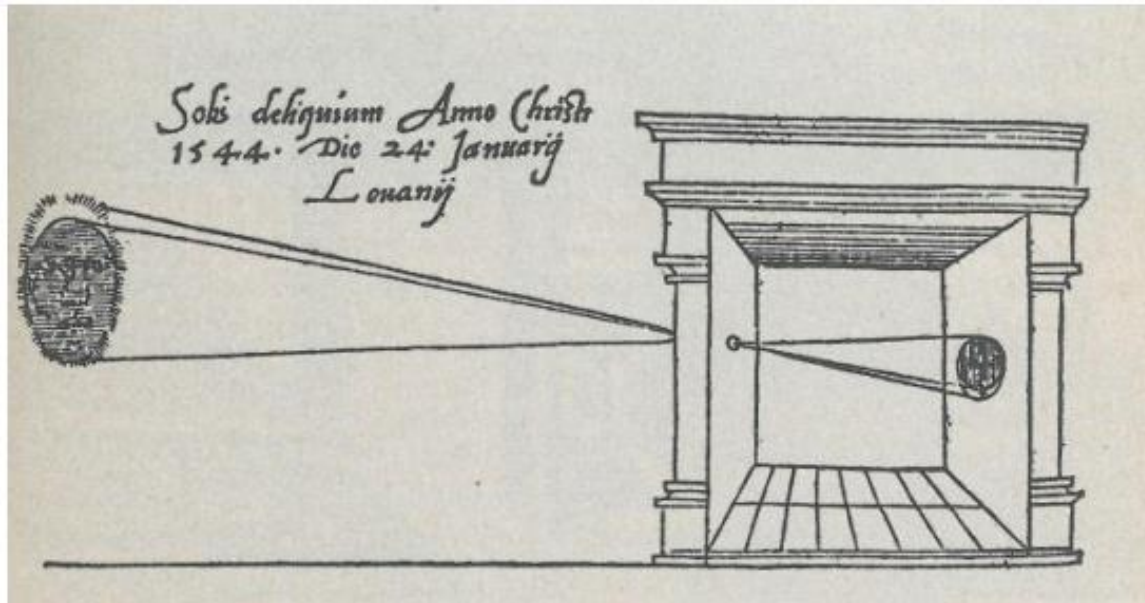
Pinhole Camera

- Place an opaque barrier with a tiny aperture (a pinhole) between the scene and the sensor.
- This barrier blocks most light rays, allowing only a narrow ray bundle from each point in the scene to pass through the pinhole.
- Pinhole camera: a camera without a lens but with a tiny aperture, a pinhole
- The light is spread on the surface

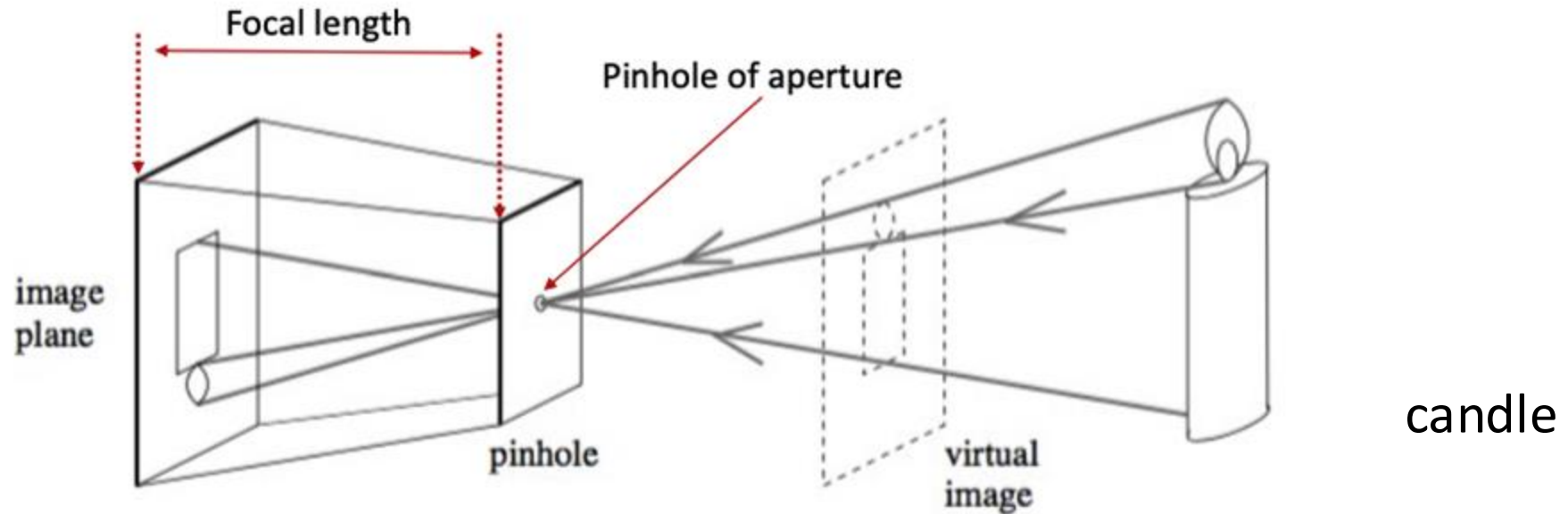


History of Pinhole Camera

- 1502: Leonardo da Vinci provides the first known clear description of the camera obscura (pinhole principle) in his notebooks.
- 1544: Gemma Frisius publishes the oldest known illustration of a camera obscura, used to observe a solar eclipse.

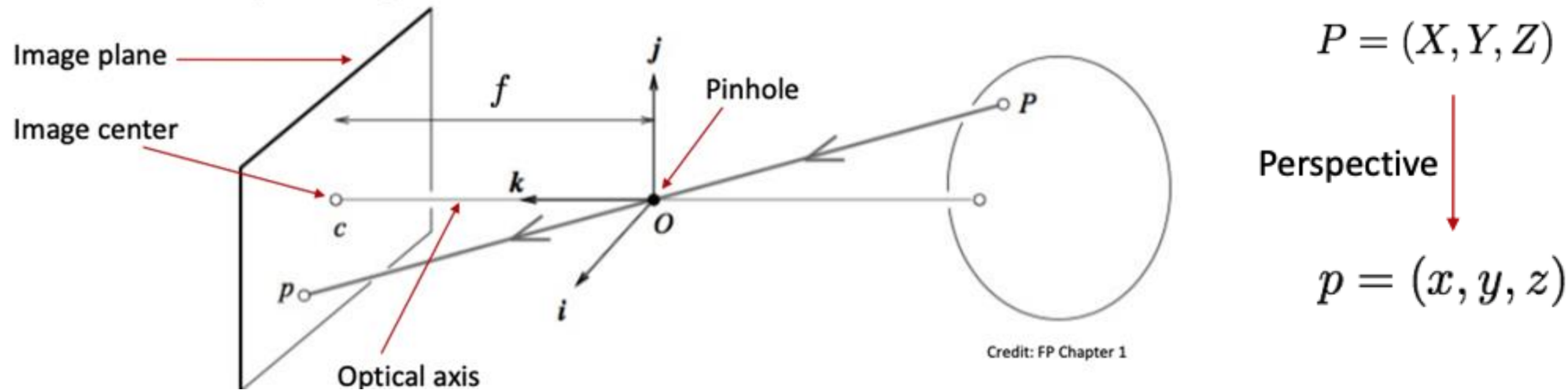


Pinhole Camera



- Perspective projection creates *inverted images*
- Sometimes it is convenient to consider a virtual image associated with a plane lying in front of the pinhole
- Virtual image is equivalent to the actual one except for on scale

From 3D to 2D: The Pinhole Camera Model



- How do we project a 3D point onto a 2D image?

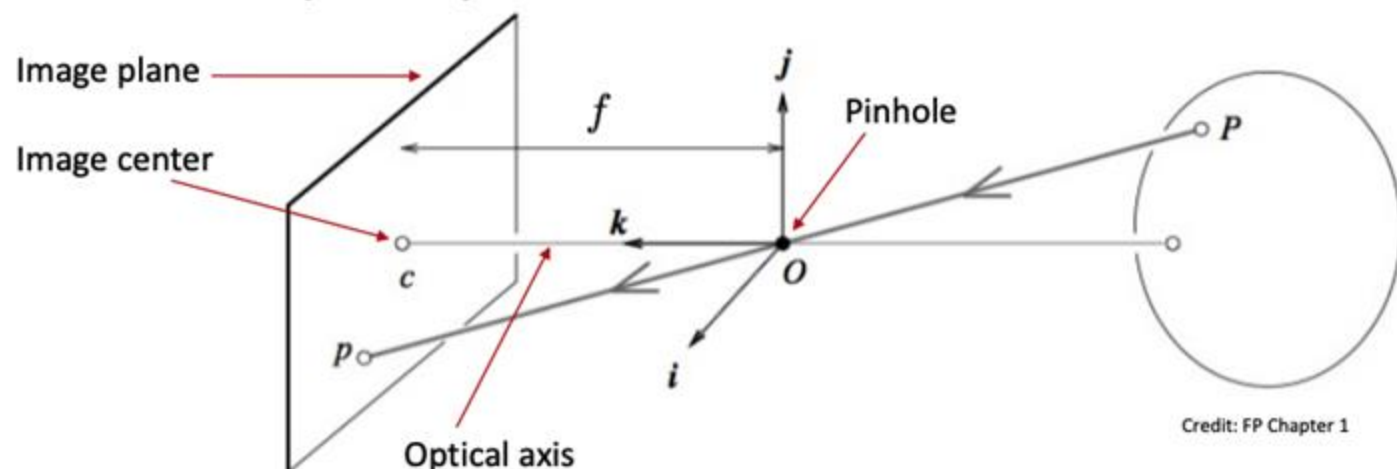
We transform a point in the world coordinate frame, $[X, Y, Z]$, into a pixel location on the image plane, $[u, v]$. This is accomplished through a series of coordinate transformations: from world, to camera, to image coordinates, and finally to pixels. The core of this process is a perspective projection, modeled by the camera's intrinsic and extrinsic parameters.

(X, Y, Z): 3D point in the Camera Coordinate System (origin at O).

(x, y) : 2D point in the Image Coordinate System (origin where the optical axis pierces the image plane).

f: Focal Length. The distance between the projection center O and the image plane.

The Geometry of Projection



$$P = (X, Y, Z)$$

Perspective

$$p = (x, y, z)$$

The Principle of Collinearity:

The world point P , the camera center (pinhole) O , and its projected image point p are always collinear. This is the fundamental geometric constraint that enables us to model perspective projection.

$$\overline{Op} = \lambda \overline{OP}$$

$$\begin{cases} x = \lambda X \\ y = \lambda Y \\ z = \lambda Z \end{cases} \Leftrightarrow \lambda = \frac{x}{X} = \frac{y}{Y} = \frac{z}{Z} \Rightarrow \begin{cases} x = f \frac{X}{Z} \\ y = f \frac{Y}{Z} \end{cases}$$

Using the principle of similar triangles, we can find the coordinates of the projected point p .

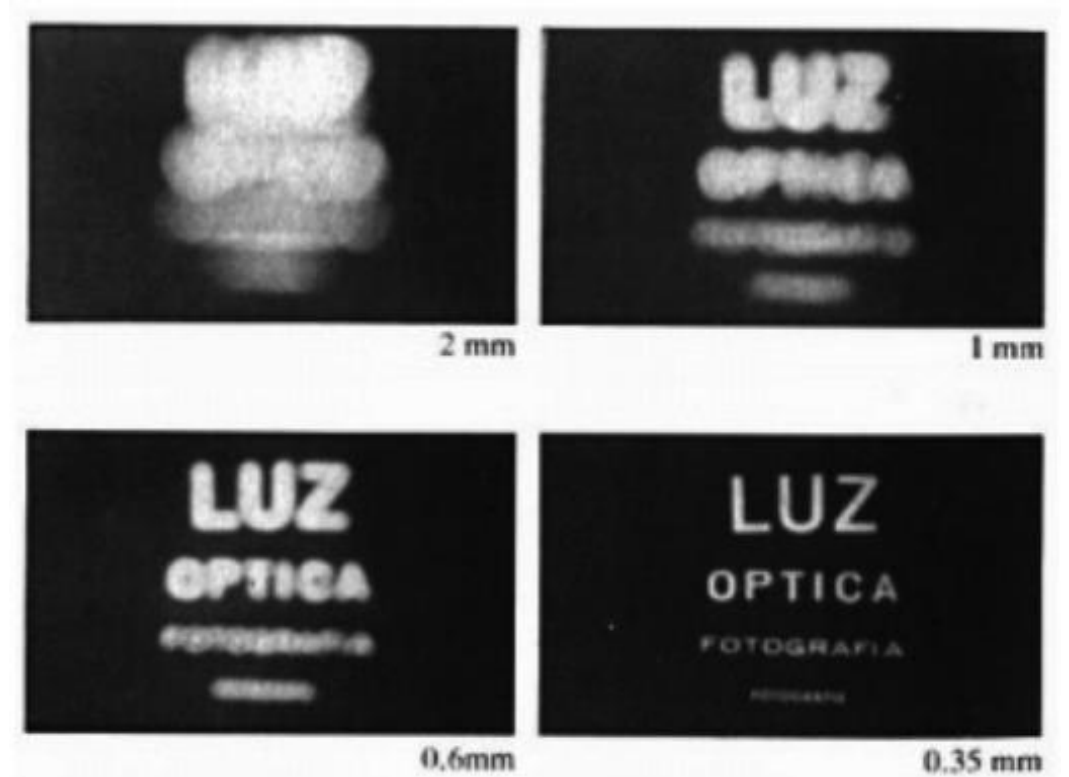
The Need for Lenses

The Pinhole Dilemma:

- A larger aperture allows more light, producing a brighter but blurrier image due to unfocused light rays.
- A smaller aperture creates a sharper image but is too dark for practical use.

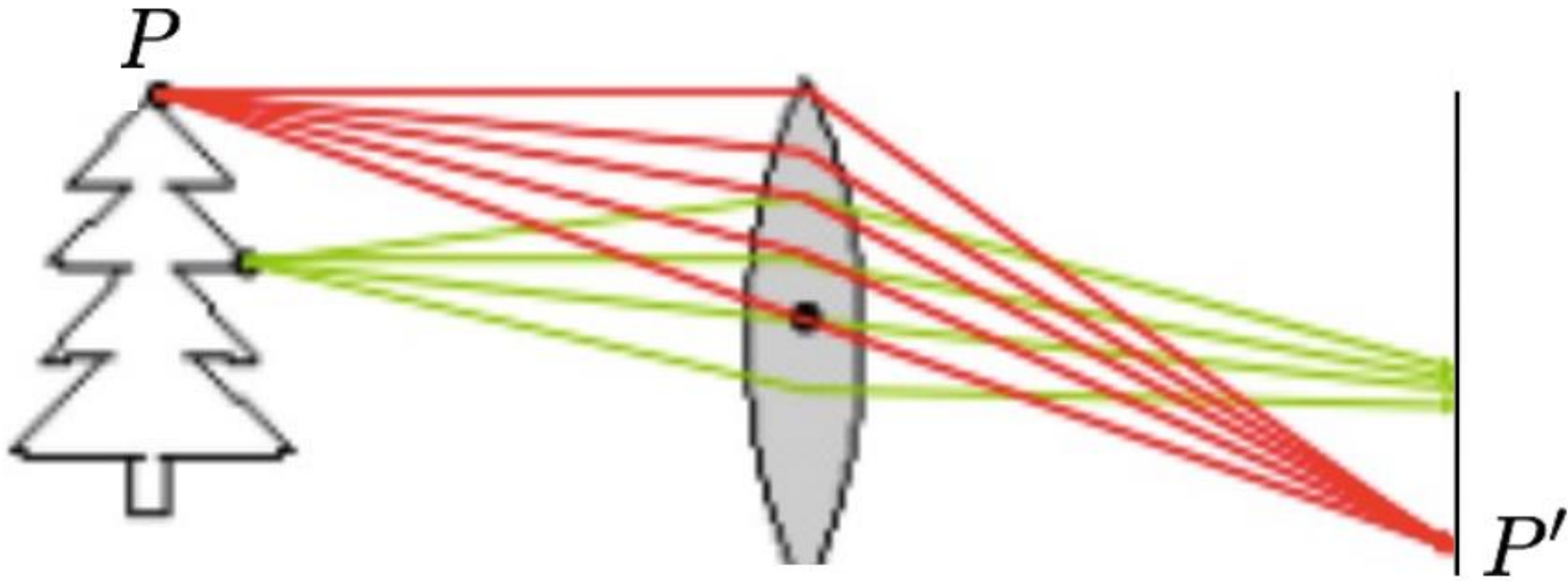
The Optical Solution:

- We replace the pinhole with a convex lens.
- The lens focuses light rays from a scene onto the image plane, providing both a bright and a sharp image.



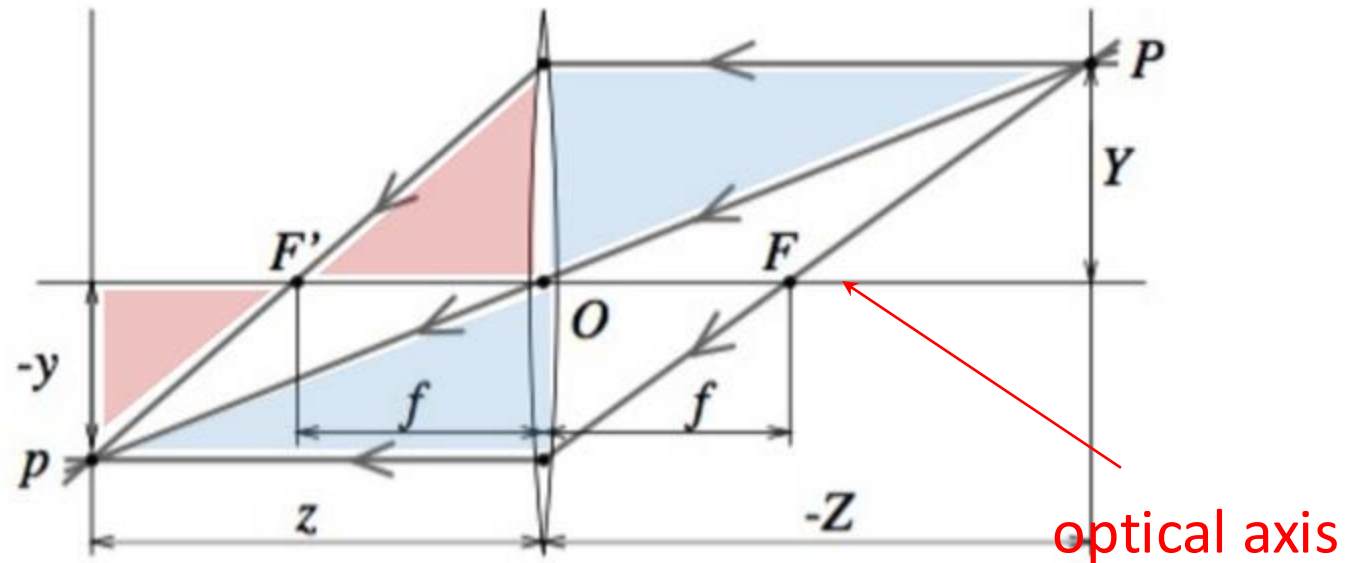
The Need for Lenses

- Lens: an optical element (glass) that focuses light by means of refraction



The Thin Lens Model

Unlike a pinhole, which allows only a single ray from P to reach p , a lens focuses a large bundle of rays from P onto the same point. This results in a brighter and clearer image.



A Practical Approximation:

The thin lens model simplifies complex lens optics into a set of predictable rules for focusing light.

Key Principles:

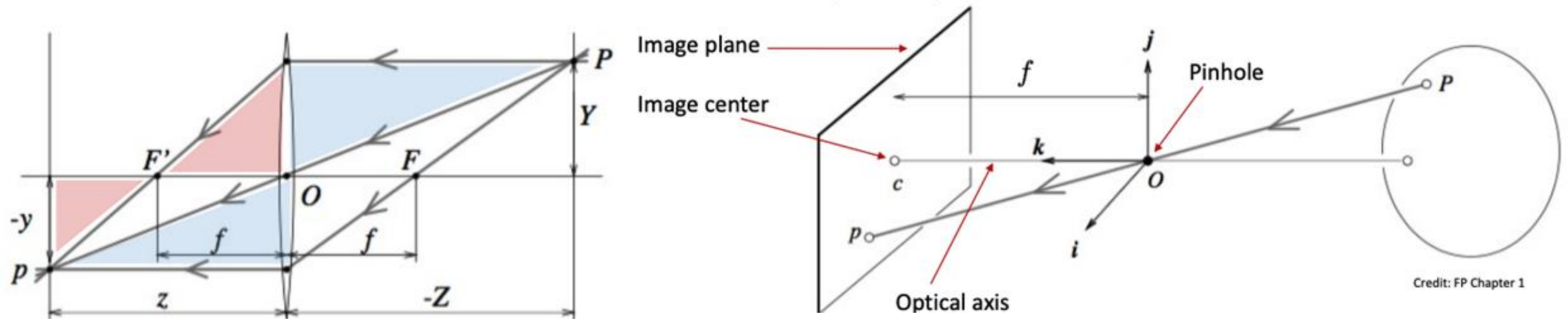
- **No Deviation at the Center:** Rays passing through the optical center (O) of the lens are not refracted and continue straight.
- **Focusing Parallel Light:** All rays parallel to the optical axis converge at a single point behind the lens: the focal point (F).
- **Point-to-Point Focusing:** All light rays emanating from a single scene point (P) are focused by the lens to a corresponding single point (p) on the image plane.

The Thin Lens Model

Assumptions:

When the image plane is placed at a distance $z = f$ (the focal length) behind the lens, the thin lens projection geometry becomes identical to the pinhole model.

For simplicity, we ignore depth of field effects, assuming the entire scene is in perfect focus. This is a valid approximation when the camera is focused at infinity.



Perspective Projection

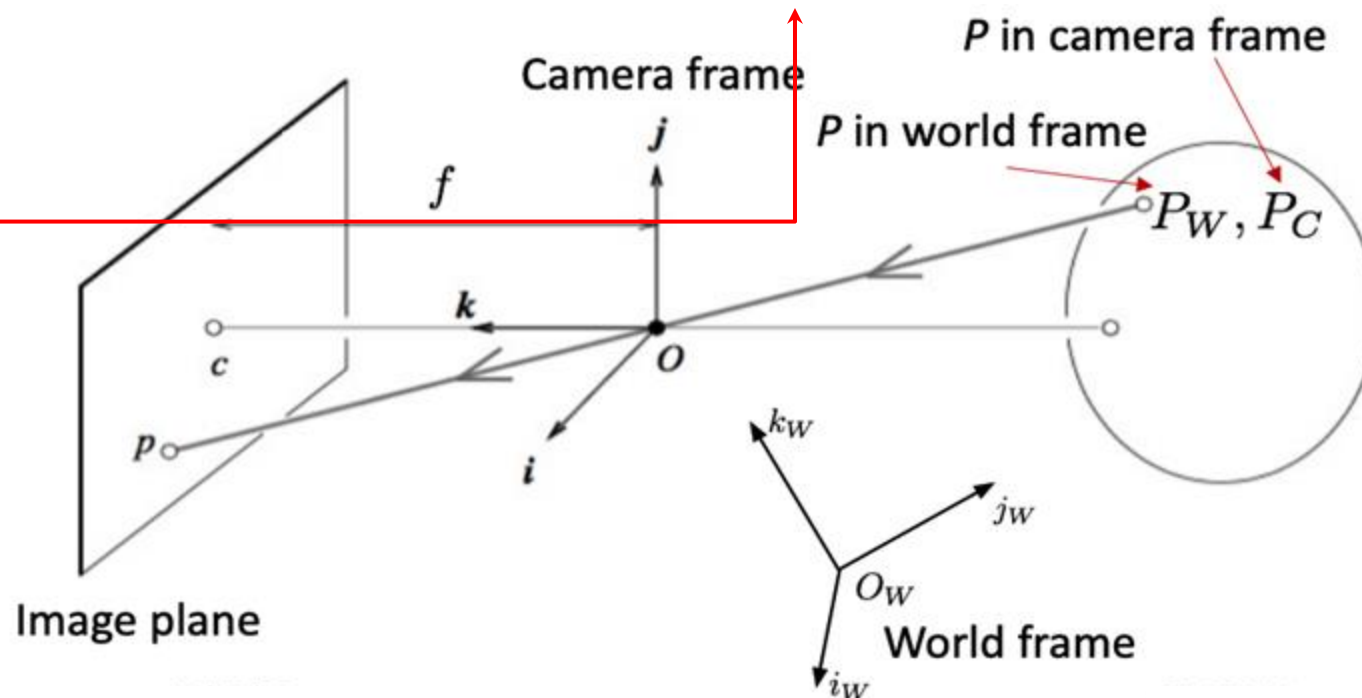
Our Goal: To project 3D points from the world onto a 2D camera image.

- We use the pinhole camera model.
- This also accurately models a thin-lens camera when focused at infinity.

Key Concept: The Camera Frame

- We define a camera coordinate frame.
- Its origin, O , is located at the lens center (the pinhole).
- The Z -axis is the optical axis, pointing out into the world.

The world frame is a fixed, global coordinate system. Its origin is chosen arbitrarily at a location separate from the camera.



Perspective Projection

1) World to Camera Transformation ($P_w \rightarrow P_c$)

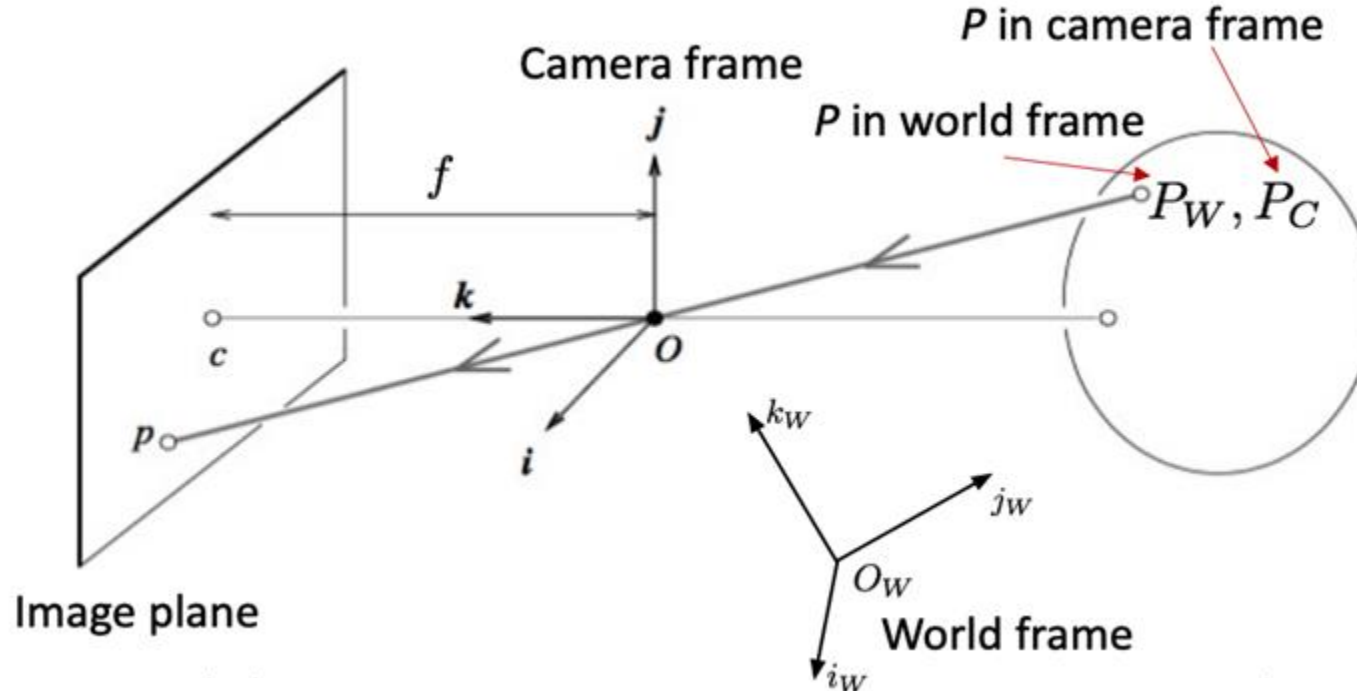
Transform the point from world coordinates to camera coordinates using the camera's pose (rotation R and translation t).

2) Perspective Projection ($P_c \rightarrow p$)

Project the 3D point in the camera frame onto the normalized image plane using the pinhole model: $(x, y) = (fX/Z, fY/Z)$.

3) Image to Pixel Transformation ($p \rightarrow (u, v)$)

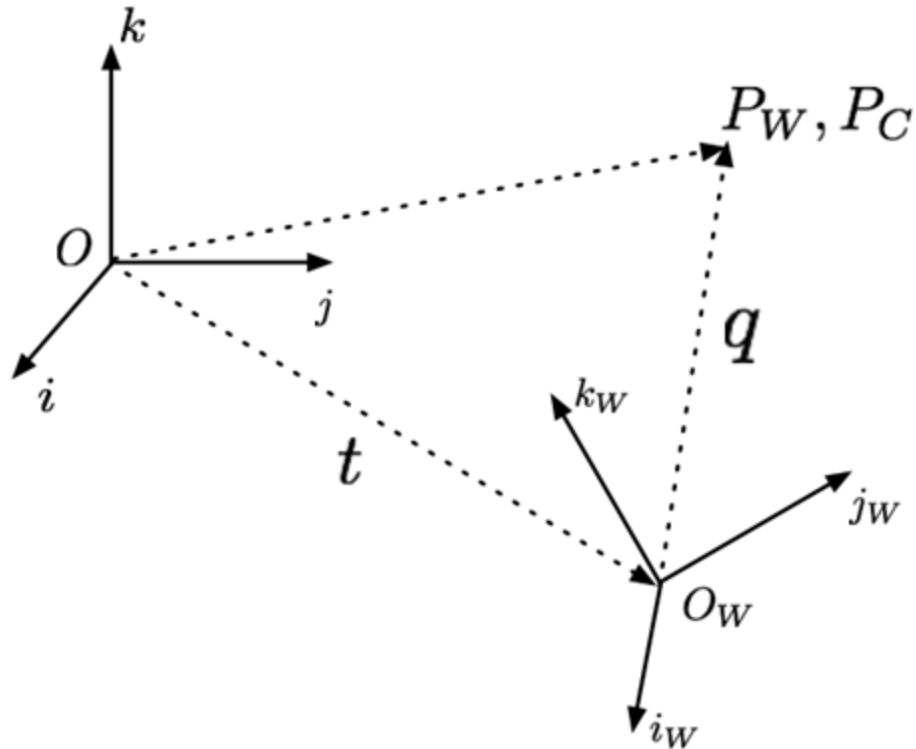
Convert the metric image coordinates into discrete pixel coordinates using the camera's intrinsic calibration matrix K .



P_w and P_c represent the same physical point in 3D space, but their coordinate values differ because they are expressed in different reference frames.

Perspective Projection

Step 1: World to Camera Transformation ($P_w \rightarrow P_c$)



$$P_C = t + q \quad \text{Rotation + Translation: Rigid transformation}$$

$$q = R P_W$$

where R is the rotation matrix relating camera and world frames

$$R = \begin{bmatrix} i_W \cdot i & j_W \cdot i & k_W \cdot i \\ i_W \cdot j & j_W \cdot j & k_W \cdot j \\ i_W \cdot k & j_W \cdot k & k_W \cdot k \end{bmatrix}$$

$$\Rightarrow P_C = t + R P_W$$

Perspective Projection

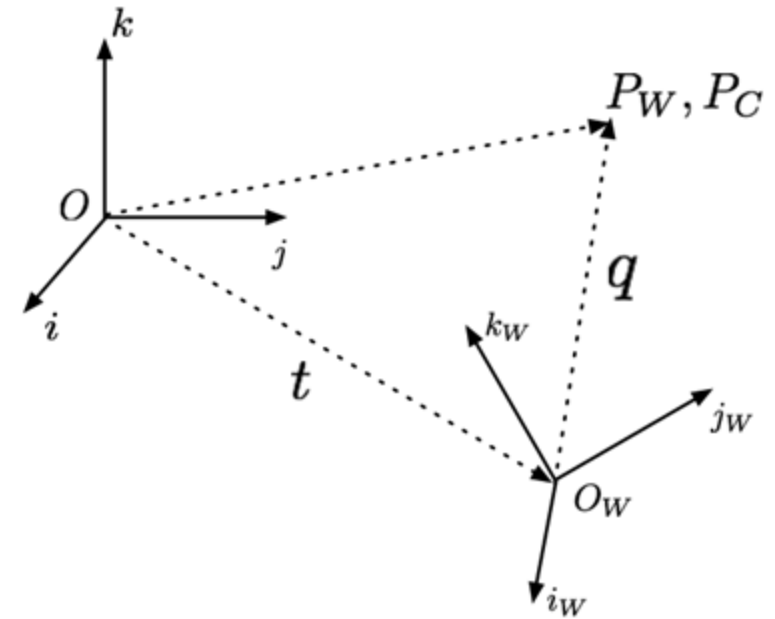
Step 1: World to Camera Transformation ($P_w \rightarrow P_c$)
In homogeneous coordinates

$$\Rightarrow P_C = t + R P_W$$

$$\begin{pmatrix} P_C \\ 1 \end{pmatrix} = \begin{bmatrix} R & t \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{pmatrix} P_W \\ 1 \end{pmatrix}$$

Point P_c in homogeneous coordinates

Point P_w in homogeneous coordinates




Perspective Projection

- Collecting all results

$$p^h = [K \quad 0_{3 \times 1}] P_C^h = K [I_{3 \times 3} \quad 0_{3 \times 1}] \begin{bmatrix} R & t \\ 0_{1 \times 3} & 1 \end{bmatrix} P_W^h$$

- Hence

Projection matrix M


$$p^h = K [R \quad t] P_W^h$$

Intrinsic parameters Extrinsic parameters

R: rotation

t: translation

Rt: extrinsic matrix

Degree of freedom:

K: 4

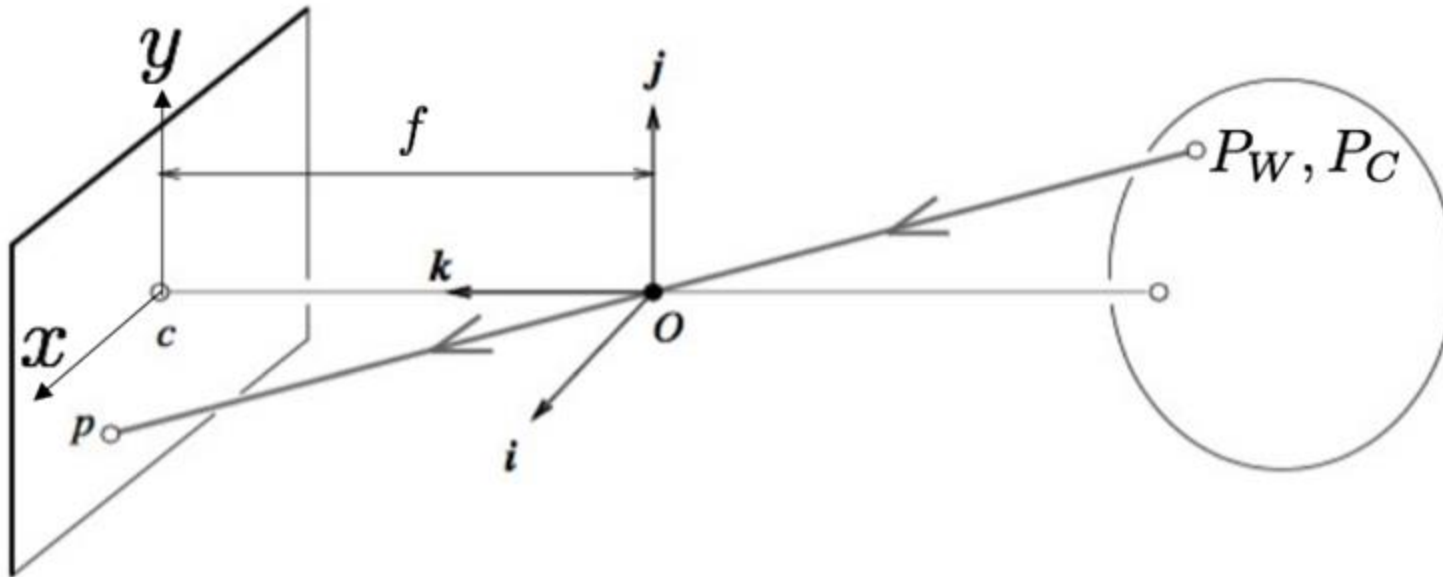
Rt: 6

Perspective Projection

Step 2: Perspective Projection ($P_c \rightarrow p$)

Map $P_c = (X_c, Y_c, Z_c)$ into $p = (x, y)$ (image plane)

We already have
$$\begin{cases} x = f \frac{X_c}{Z_c} \\ y = f \frac{Y_c}{Z_c} \end{cases}$$



$$\overline{Op} = \lambda \overline{OP}$$

$$\begin{cases} x = \lambda X \\ y = \lambda Y \\ z = \lambda Z \end{cases}$$

$$z = f$$

$$\lambda = \frac{x}{X} = \frac{y}{Y} = \frac{z}{Z}$$

$$\begin{cases} x = f \frac{X_c}{Z_c} \\ y = f \frac{Y_c}{Z_c} \end{cases}$$

Perspective Projection

Step 3: Image to Pixel Transformation ($p \rightarrow (u, v)$)

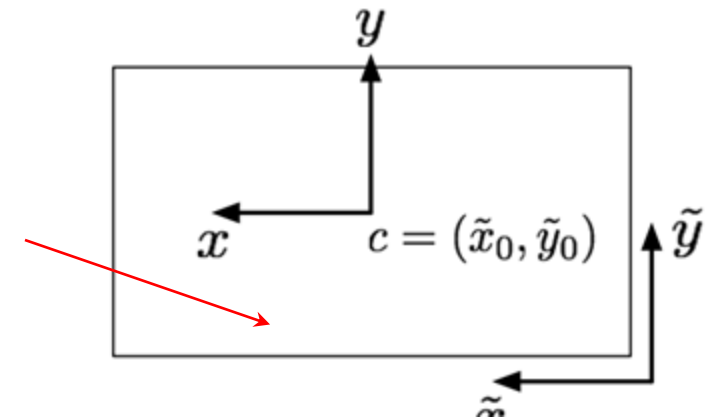
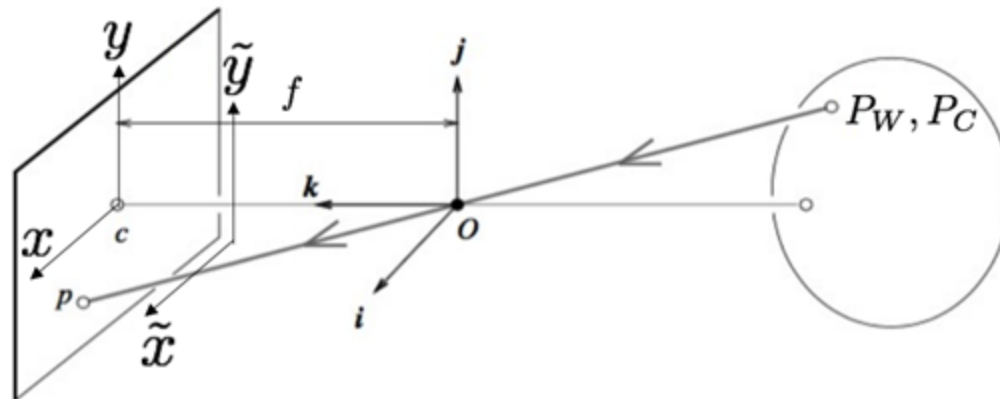
The normalized image point p exists in a metric 2D plane centered on the optical axis. We must map it to the discrete image sensor via an affine transformation:

Change of Origin: The principal point $c = (c_x, c_y)$ defines where the optical axis pierces the image plane, in pixels. We translate the point by c .

Unit Conversion: The focal length f (in meters) is converted to pixels using the sensor's pixel density, giving us f_x and f_y .

$$\tilde{x} = f \frac{X_C}{Z_C} + \tilde{x}_0, \quad \tilde{y} = f \frac{Y_C}{Z_C} + \tilde{y}_0,$$

$$\begin{cases} x = f \frac{X_C}{Z_C} \\ y = f \frac{Y_C}{Z_C} \end{cases}$$



Perspective Projection

Step 3: Image to Pixel Transformation ($p \rightarrow (u, v)$)

Convert the image coordinates to pixel coordinates (image is in centimeters, etc; pixel is different)
Simply change the scale of the coordinates

We already have

$$\tilde{x} = f \frac{X_C}{Z_C} + \tilde{x}_0, \quad \tilde{y} = f \frac{Y_C}{Z_C} + \tilde{y}_0,$$

Number of pixels
per unit distance in
image coordinates

$$u = k_x \tilde{x} = \underbrace{k_x f}_{\alpha} \frac{X_C}{Z_C} + \underbrace{k_x \tilde{x}_0}_{u_0}$$
$$v = k_y \tilde{y} = \underbrace{k_y f}_{\beta} \frac{Y_C}{Z_C} + \underbrace{k_y \tilde{y}_0}_{v_0}$$

\Rightarrow

$$\begin{aligned} u &= \alpha \frac{X_C}{Z_C} + u_0 \\ v &= \beta \frac{Y_C}{Z_C} + v_0 \end{aligned}$$

Nonlinear transformation

Homogeneous coordinates

To express the multi-stage projection (rigid transformation + perspective division + intrinsics) as a single linear mapping, we must move from Euclidean coordinates to projective geometry.

Euclidean Space: A point is (X, Y, Z) .

Homogeneous Space: The same point is represented as $(X, Y, Z, 1)$. Any point (X, Y, Z, w) where $w \neq 0$ corresponds to the Euclidean point $(X/w, Y/w, Z/w)$.

$$\begin{pmatrix} x \\ y \end{pmatrix} \Rightarrow \lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \begin{pmatrix} x \\ y \\ z \end{pmatrix} \Rightarrow \lambda \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Perspective Projection

Projection can be equivalently written in homogeneous coordinates

Focal length
+
Principal
point offset +
the pixel-
real-world
scale

Camera matrix/
Matrix of intrinsic parameters

The XYZ in camera frame

The x,y in pixels

$$\begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha X_c + u_0 Z_c \\ \beta Y_c + v_0 Z_c \\ Z_c \end{pmatrix}$$

P_c in homogeneous coordinates

Homogeneous pixel coordinates

$$p^h = [K \quad 0_{3 \times 1}] P_C^h$$

Perspective Projection

- Projection can be equivalently written in homogeneous coordinates

Point p in homogeneous pixel coordinates $\nearrow p^h = [K \quad 0_{3 \times 1}] P_C^h \nwarrow$ Point P_c in homogeneous camera coordinates

Camera Parameters

Blueprint attribute	Type	Default	Description
<code>bloom_intensity</code>	float	0.675	Intensity for the bloom post-process effect, <code>0.0</code> for disal
<code>fov</code>	float	90.0	Horizontal field of view in degrees.
<code>fstop</code>	float	1.4	Opening of the camera lens. Aperture is <code>1/fstop</code> with typical lens going down to f/1.2 (larger opening). Lar
<code>image_size_x</code>	int	800	Image width in pixels.
<code>image_size_y</code>	int	600	Image height in pixels.
<code>iso</code>	float	100.0	The camera sensor sensitivity.
<code>gamma</code>	float	2.2	Target gamma value of the camera.
<code>lens_flare_intensity</code>	float	0.1	Intensity for the lens flare post-process effect, <code>0.0</code> for d
<code>sensor_tick</code>	float	0.0	Simulation seconds between sensor captures (ticks).
<code>shutter_speed</code>	float	200.0	The camera shutter speed in seconds (1.0/s).

Projection matrix M

$$p^h = K[R \quad t]P_W^h$$

Intrinsic parameters Extrinsic parameters

R: rotation
t: translation
Rt: extrinsic
matrix

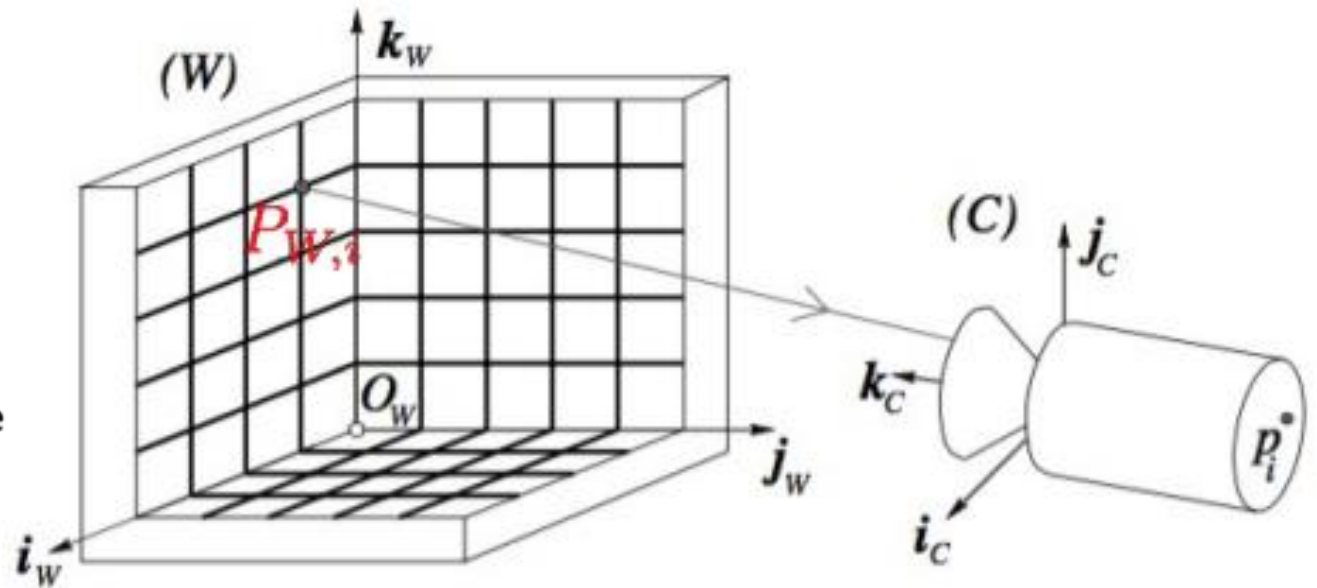
Camera Calibration

- Solve for the projection matrix $\mathbf{P} = \mathbf{K} [\mathbf{R} \mid \mathbf{t}]$.
- The model has **10 essential parameters** (4 from \mathbf{K} , 6 from $[\mathbf{R}|\mathbf{t}]$). An 11th (skew) is often negligible.
- Use a set of n known 3D points (X, Y, Z) and their corresponding 2D image points (u, v) .
- Each point correspondence provides 2 equations. Therefore, a minimum of 6 points is required to solve the system (since $6 \text{ points} * 2 = 12 \text{ equations} > 11 \text{ unknowns}$).

$$p_i \leftrightarrow P_{W,i}$$

$P_{W,1}, P_{W,2}, \dots, P_{W,n}$ with **known** positions in world frame

p_1, p_2, \dots, p_n with **known** positions in image frame



Camera Calibration

Chessboard + OpenCV

A chessboard provides a known, regular 3D structure.

Its high-contrast corners create features that are easy to detect and match across images, generating thousands of accurate correspondences automatically.

This data is used in a robust optimization to compute the camera parameters.

Sources: https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html





Thanks for your attention!

Changhao Chen
HKUST (GZ)

changhaochen@hkust-gz.edu.cn

Homepage: [changhao-chen@github.io](https://github.com/changhao-chen)