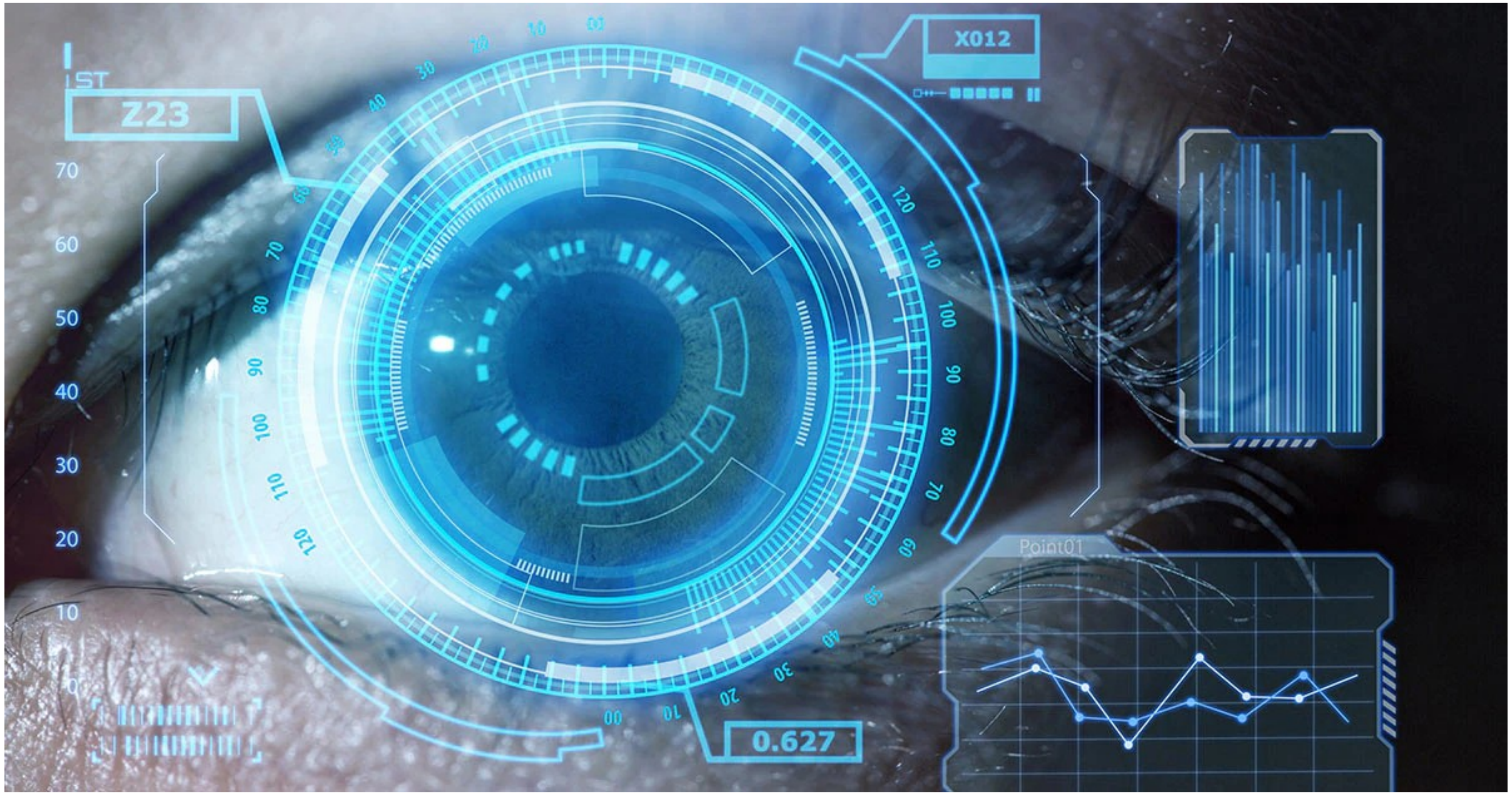
# EECE 5550: Mobile Robotics




Lecture 8: Fundamentals of Computer Vision

# Plan of the day

- Camera models
  - Image formation
  - Pinhole projection model
  - Homogeneous coordinates

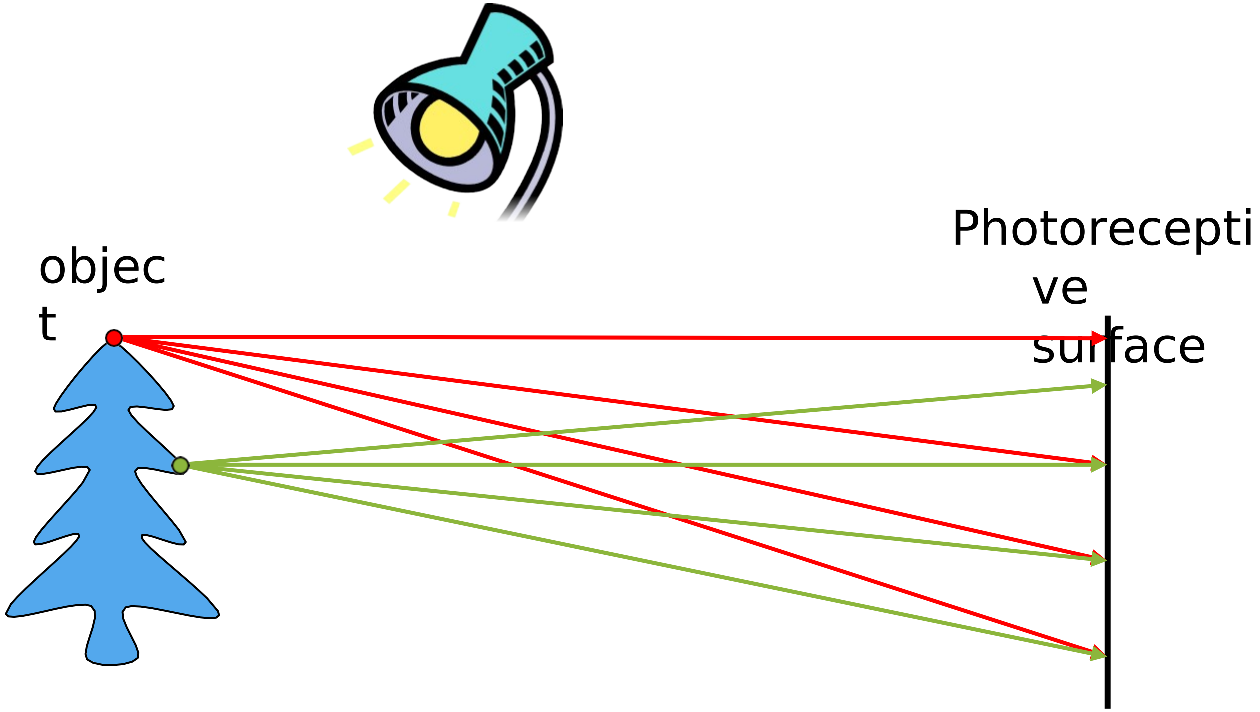
- Feature extraction
  - Corner detectors
  - Feature descriptors

# References

Today's lecture consists of selections from two lectures given by Margarita Chli in ETH Zurich's "Autonomous Mobile Robots" course (Spring 2021)

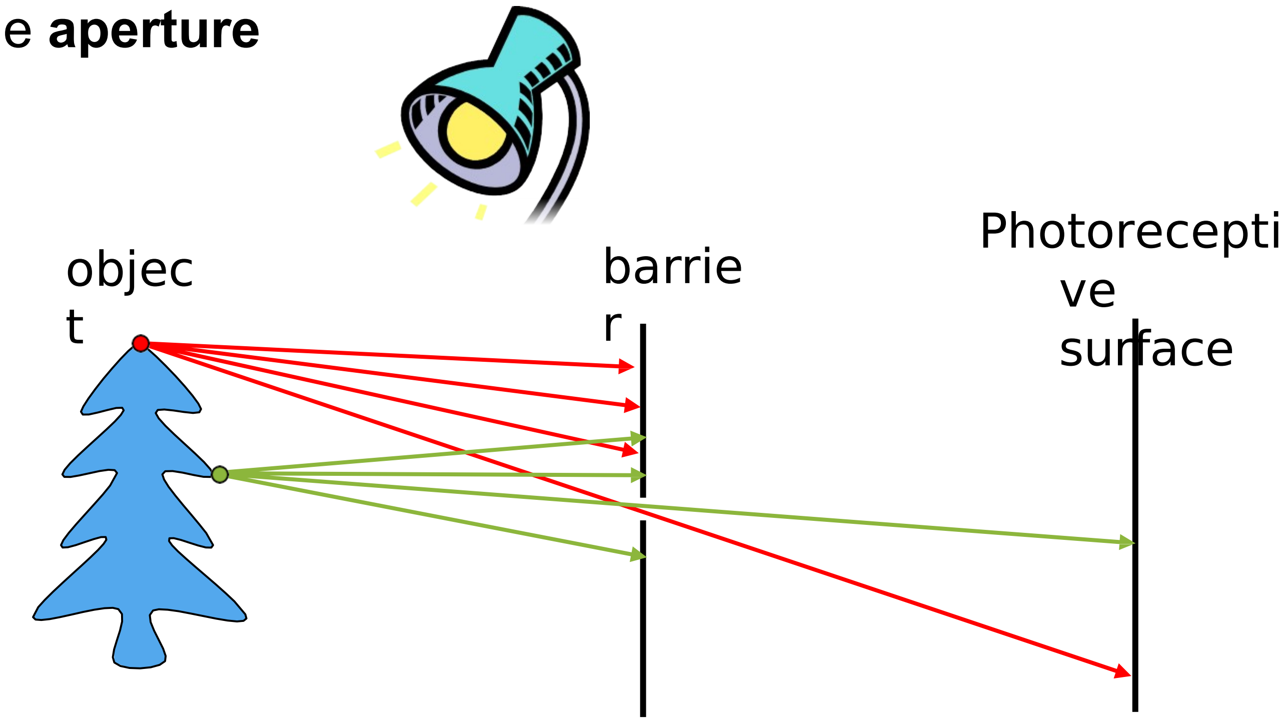

# **The camera** | image formation

- If we place a piece of film in front of an object, do we get a reasonable image?

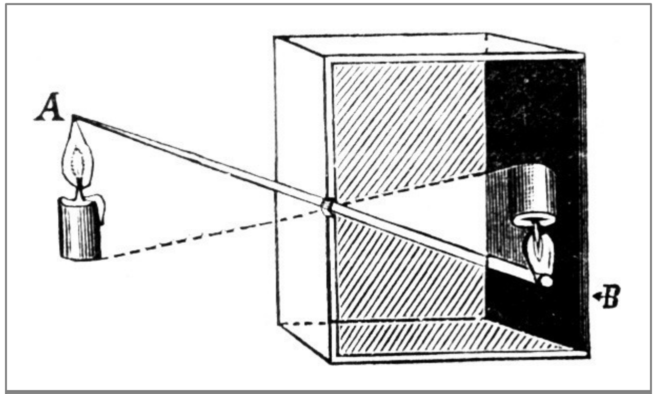


# **The camera** | image formation

- If we place a piece of film in front of an object, do we get a reasonable image?
- Add a barrier to block off most of the rays
  - This reduces blurring
  - The opening is known as the **aperture**



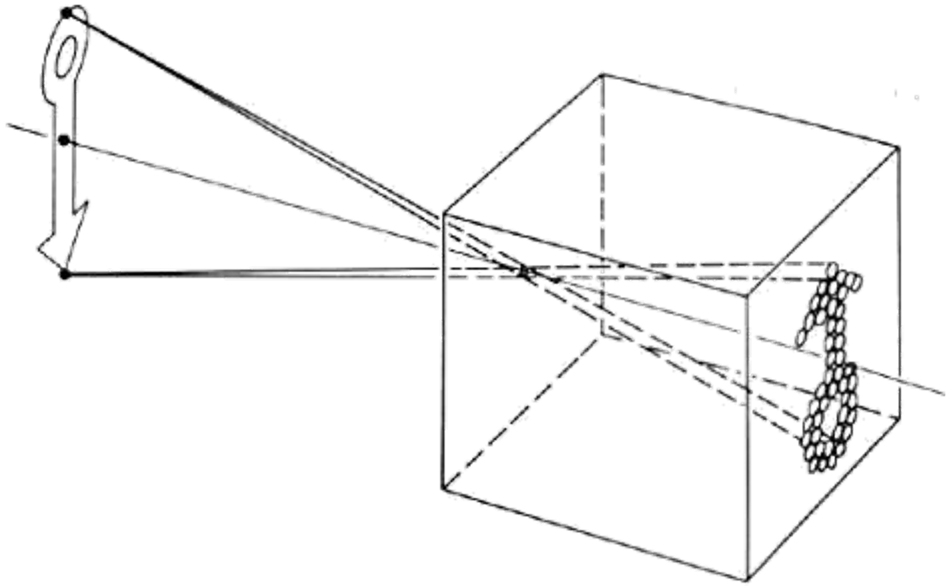
# The camera | camera obscura



commons.wikimedia.org

- Basic principle known to Mozi (470-390 BC), Aristotle (384-322 BC), Euclid (323-283 BC)
- Drawing aid for artists: described by Leonardo da Vinci (1452-1519)
- Image is inverted
- Depth of the room (box) is the effective focal length

# **The camera** | the pinhole camera model



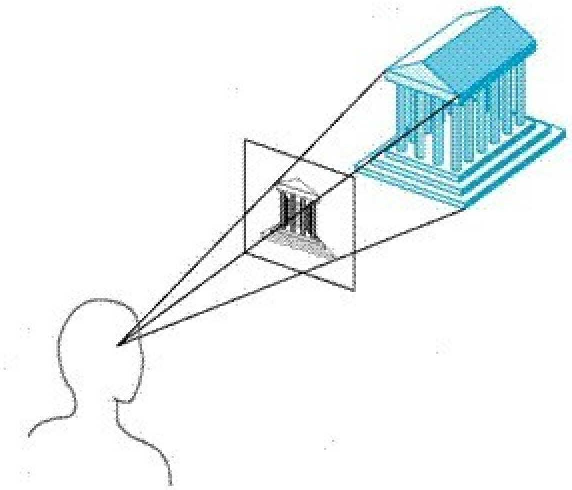
- Pinhole model:
  - Captures **beam of rays** – all rays through a single point (note: no lens!)
  - The point is called **Center of Projection** or **Optical Center**
  - The image is formed on the **Image Plane**

- We will use the pinhole camera model to describe how the image is formed

Based on slide by Steve Seitz

# Perspective projection

How world points map to pixels in the image?

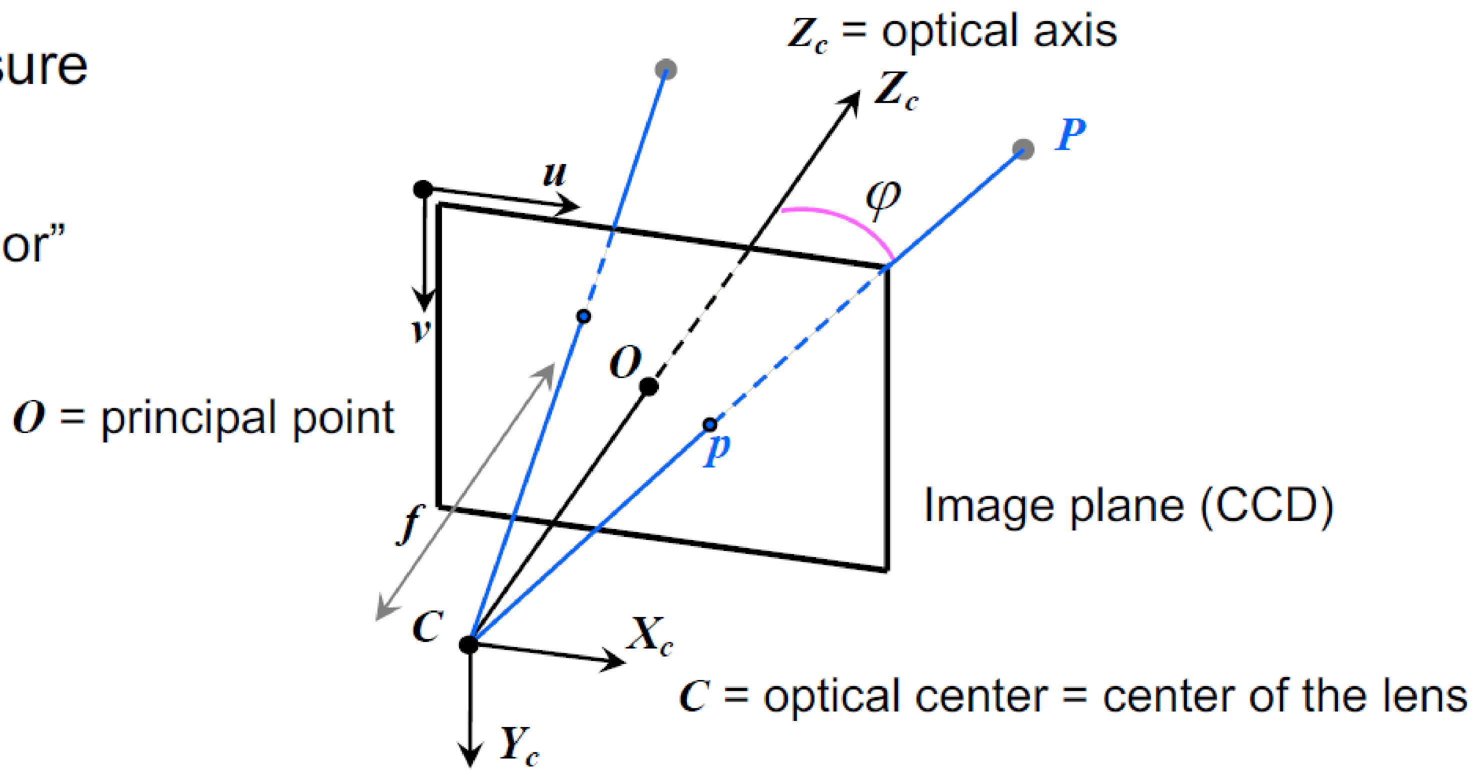


# Perspective projection

- For convenience: the image plane is usually represented in front of $C$, such that the image preserves the same orientation (i.e. not flipped)

- A camera does not measure distances but angles!

⇨ a camera is a "bearing sensor"



# **Perspective projection** | from world to pixel coordinates

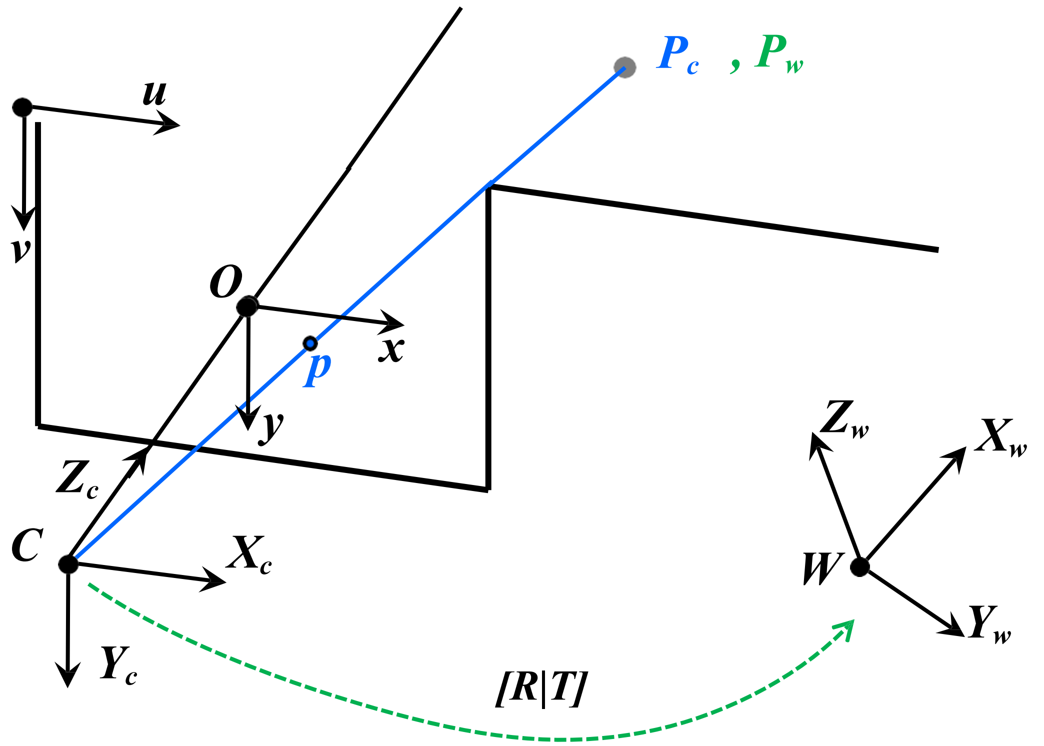
Find pixel coordinates $(u,v)$ of point $P_w$ in the world frame:

**0.** Convert world point $P_w$ to camera point $P_c$

Find pixel coordinates $(u,v)$ of point $P_c$ in the camera frame:

**1.** Convert $P_c$ to image-plane coordinates $(x,y)$

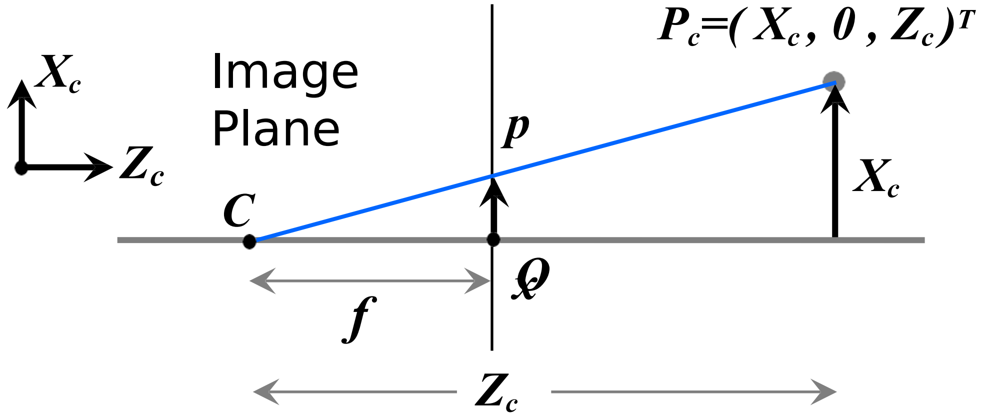
**2.** Convert $P_c$ to (discretised) pixel coordinates $(u,v)$



# **Perspective projection** | from camera frame to image plane

- The Camera point $P_c = (X_c, 0, Z_c)^T$ projects to $p = (x, y)$ onto the image plane
- From similar triangles:

$$\frac{x}{f} = \frac{X_c}{Z_c} \Rightarrow x = \frac{fX_c}{Z_c}$$




- Similarly, in the general case:

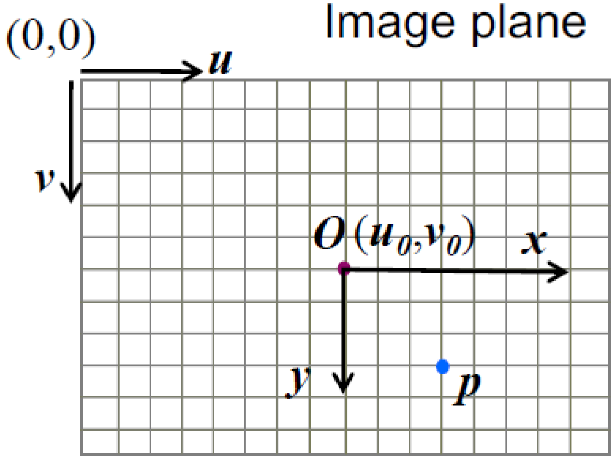
$$\frac{y}{f} = \frac{Y_c}{Z_c} \Rightarrow y = \frac{fY_c}{Z_c}$$

**1.** Convert $P_c$ to image-plane coordinates $(x,y)$

**2.** Convert $P_c$ to (discretised) pixel coordinates $(u,v)$

# **Perspective projection** | from camera frame to pixel coords.

- Convert $p$ from the local image plane coords $(x,y)$ to the pixel coords $(u,v)$, we need to account for:
  - the pixel coords of the camera optical center $O = (u_0, v_0)$
  - scale factors $k_u, k_v$ for the pixel-size in both dimensions




- So: $$u = u_0 + k_u x \Rightarrow u = u_0 + \frac{k_u f X_c}{Z_c}$$
  $$v = v_0 + k_v y \Rightarrow v = v_0 + \frac{k_v f Y_c}{Z_c}$$

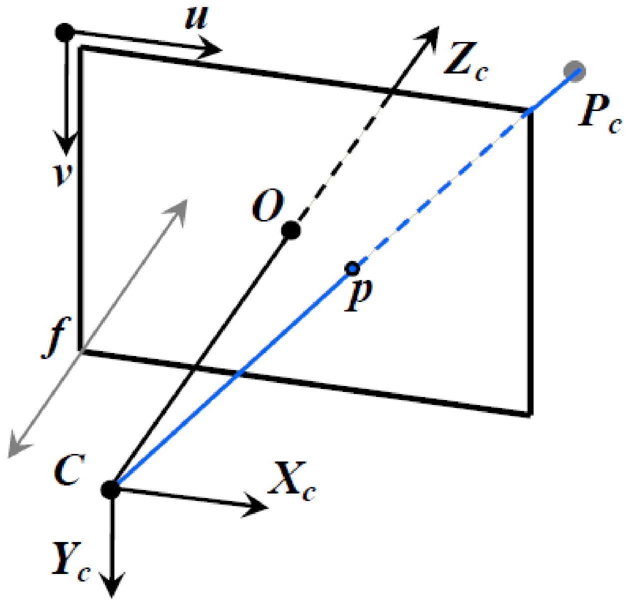
- Use **Homogeneous Coordinates** for linear mapping from 3D to 2D, by introducing an extra element (scale):

$$p = \begin{pmatrix} u \\ v \end{pmatrix} \Rightarrow \tilde{p} = \begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$ and similarly for the world coordinates. Note, usually $\lambda = 1$

# **Perspective projection** | from camera frame to pixel coords.

$$u = u_0 + \frac{k_u f X_c}{Z_c}$$

$$v = v_0 + \frac{k_v f Y_c}{Z_c}$$




- Expressed in matrix form and homogeneous coordinates:

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

- Or alternatively

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Focal length in $u$-direction

Focal length in $v$-direction

"Calibration matrix"/
"Matrix of Intrinsic Parameters"

# Perspective projection | from the world to the camera frame



$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

in homogeneous coordinates

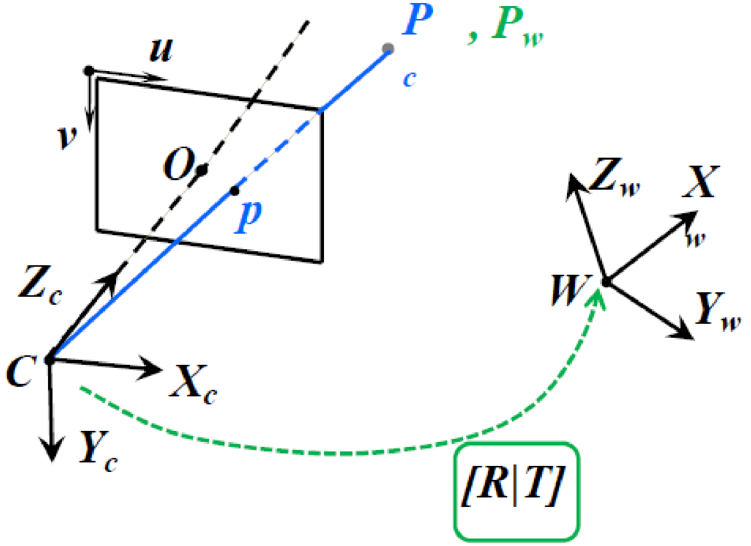
$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} & R & & T \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

From the previous slide:

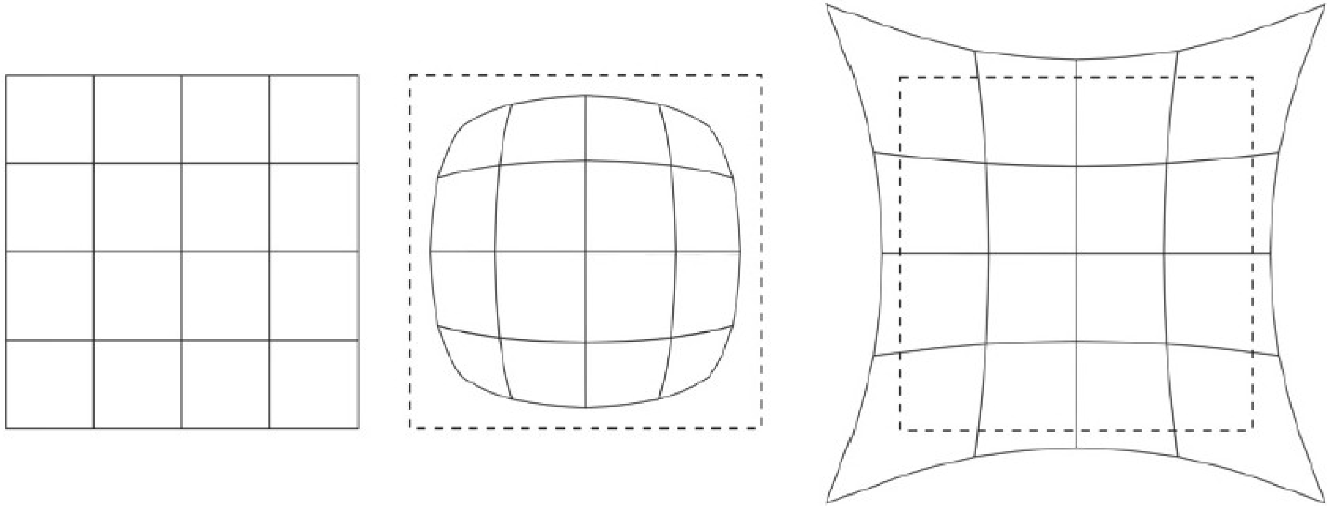
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

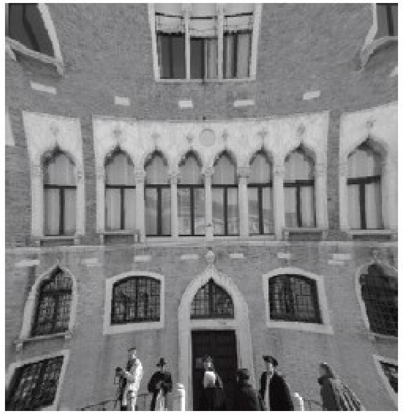
Projection Matrix

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \boxed{K\left[R|T\right]} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$



# Perspective projection | radial distortion





No distortion | Barrel distortion | Pincushion distortion

- Amount of distortion is a non-linear $f^n$ of distance from center of image

From ideal $(u,v)$ to distorted pixel coordinates $(u_d\,, v_d)$:

- Simple quadratic distortion model:

$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = (1 + k_1 r^2) \begin{bmatrix} u - u_0 \\ v - v_0 \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}$$

where Radial Distortion parameter

$$r^2 = (u - u_0)^2 + (v - v_0)^2 .$$

- Works well for most lenses

# Plan of the day

- Camera models
  - Image formation
  - Pinhole projection model
  - Homogeneous coordinates

- Feature extraction
  - Edge detection
  - Corner detection
  - Feature descriptors