Intelligent vehicles and autonomous driving

PERCEPTION SYSTEMS 2022 / 2023

Lesson 5

Alice Plebe
University of Trento

RECAP

- LIDAR
  - Rotating LIDAR vs. solid-state LIDAR
  - Scanning mechanism, 3D point clouds
  - Pulsed LIDAR
  - Amplitude modulated continuous wave (AMCW) LIDAR
  - Frequency modulated continuous wave (FMCW) LIDAR
  - Multiple return modes

CAMERA

CAMERA

Passive, light-collecting, relatively inexpensive sensor that captures rich and detailed information about a scene.

Because of its high resolution output, the camera provides orders of magnitude more information than other sensors. The camera is often considered the primary sensor in a vehicle.

It requires extensive and computationally demanding processing to exploit the information contained in the camera images.

CAMERA SENSOR

Stereo camera by Bosh

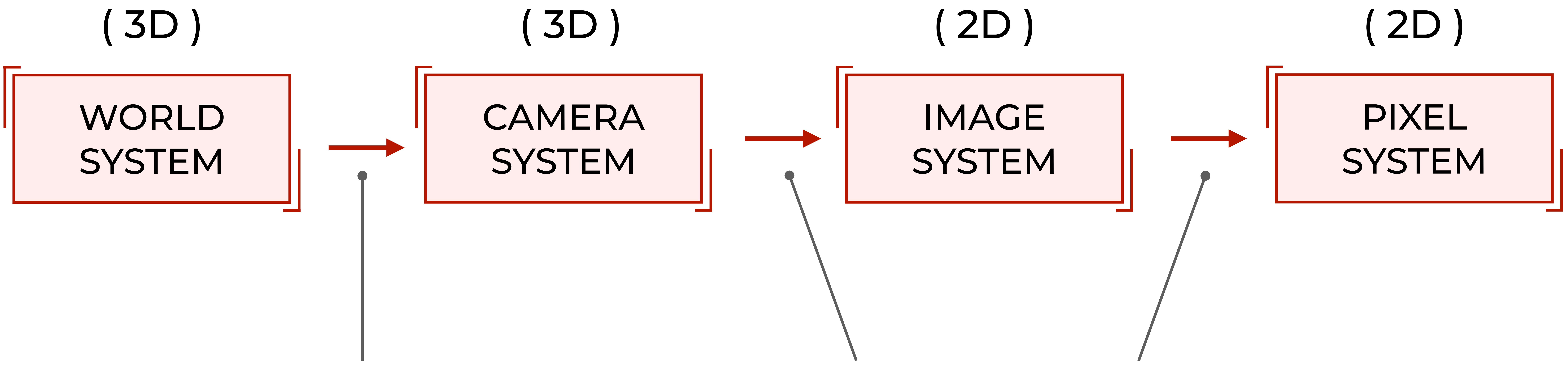
CAMERA PROJECTION

Take a point in the 3D world and project it onto a 2D plane (the image).

This transformation is determined by the camera parameters:

- **extrinsic parameters** depend on the location and orientation of the camera
- **intrinsic parameters** define how the camera captures the images, they include focal length, aperture, field-of-view, resolution, and more.

COORDINATE SYSTEMS



This transformation uses the extrinsic matrix

These transformations use the intrinsic matrix

WORLD COORDINATE SYSTEM

Basic 3D cartesian coordinate system with arbitrary origin.

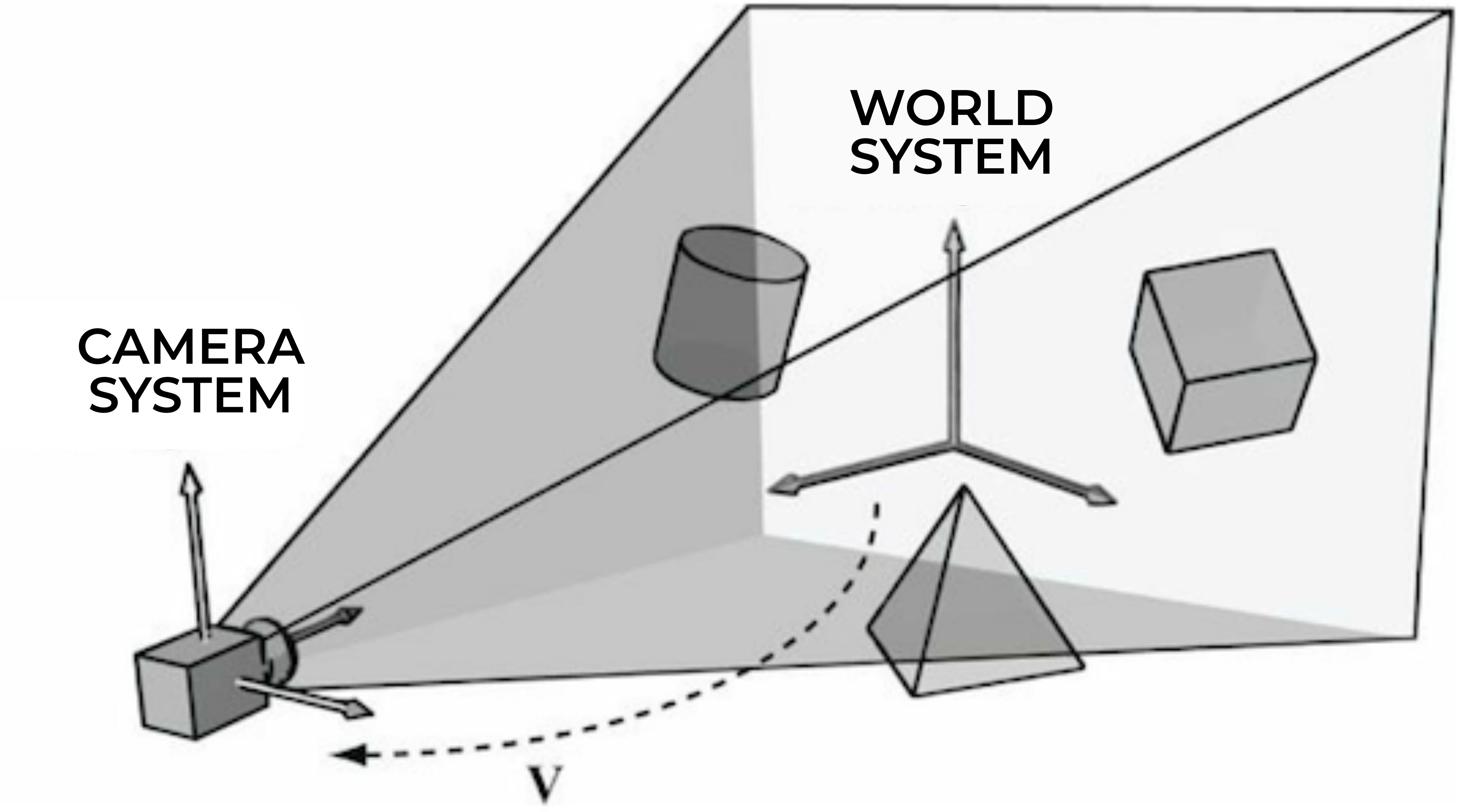
A point in this system is denoted as follows:

$$\mathbf{P}_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$$

CAMERA COORDINATE SYSTEM

The coordinate system that measures relative to the camera's origin and orientation.

$$\mathbf{P}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$$



WORLD (3D) → CAMERA (3D)

$$\mathbf{P}_c = \mathbf{Q}\,\mathbf{P}_w \quad \Longrightarrow \quad \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = [\mathbf{R} \,|\, \mathbf{T}] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

$$\mathbf{R} = \mathbf{R}_x\,\mathbf{R}_y\,\mathbf{R}_z$$

$\mathbf{Q}$ = camera extrinsic matrix (4x4)

WORLD (3D) ⟶ CAMERA (3D)

$$\mathbf{R}_X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_Y = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_Z = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T} = \begin{bmatrix} t_X \\ t_Y \\ t_Z \\ 1 \end{bmatrix}$$

IMAGE COORDINATE SYSTEM

2D coordinate system that results from the projection of 3D points in the camera coordinate system using a *pinhole camera model.*

$$\mathbf{P}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

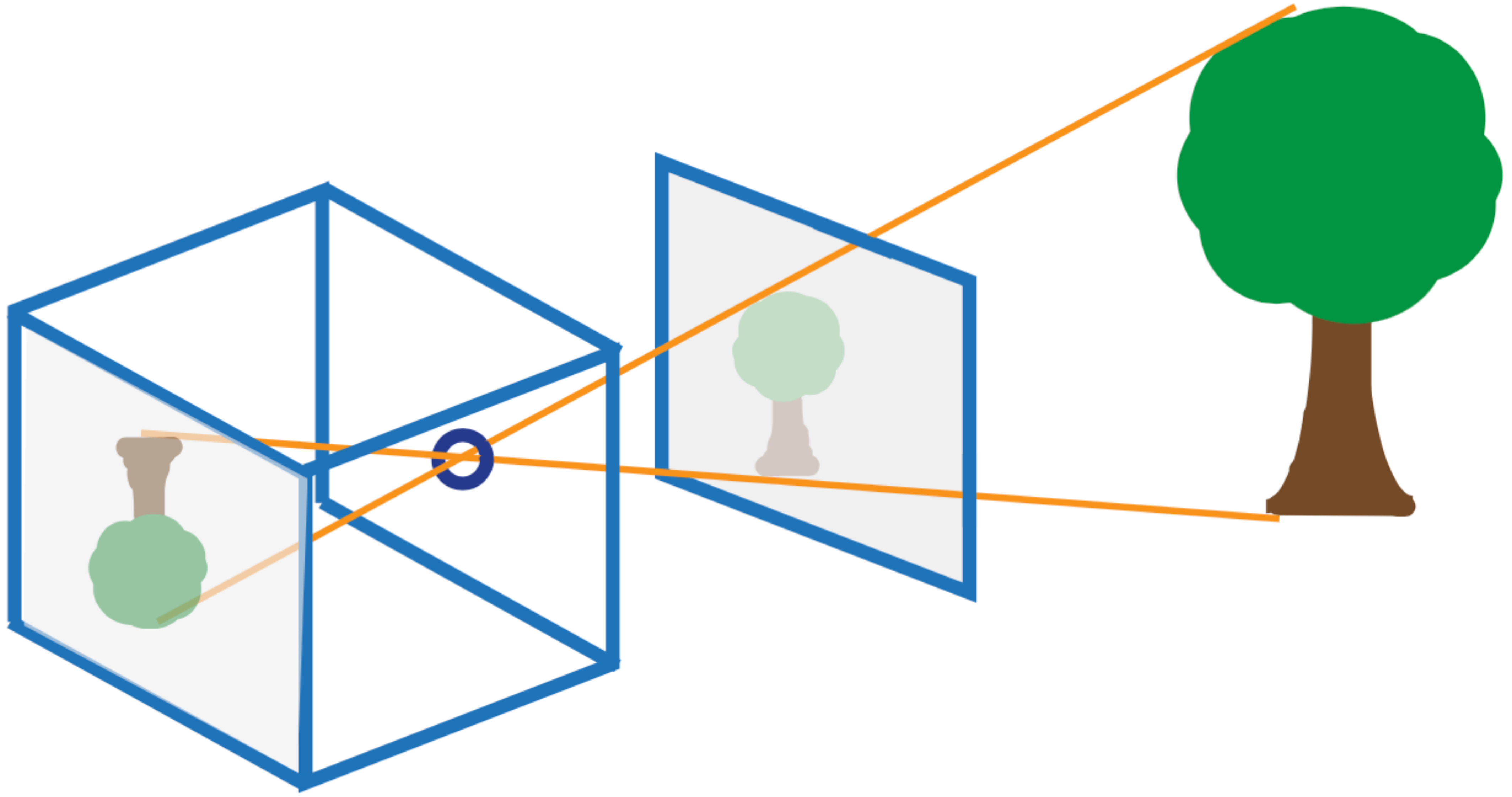
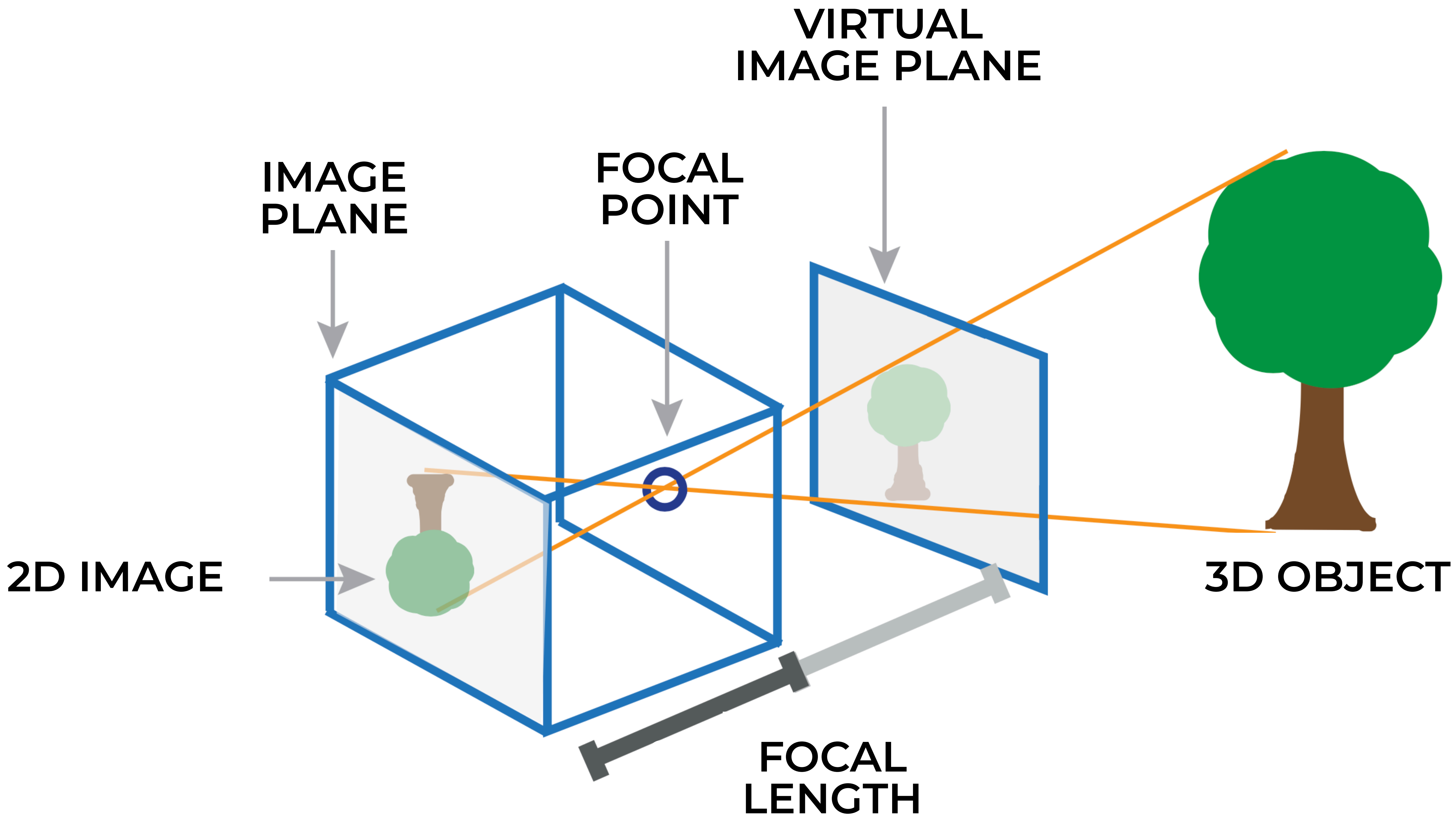
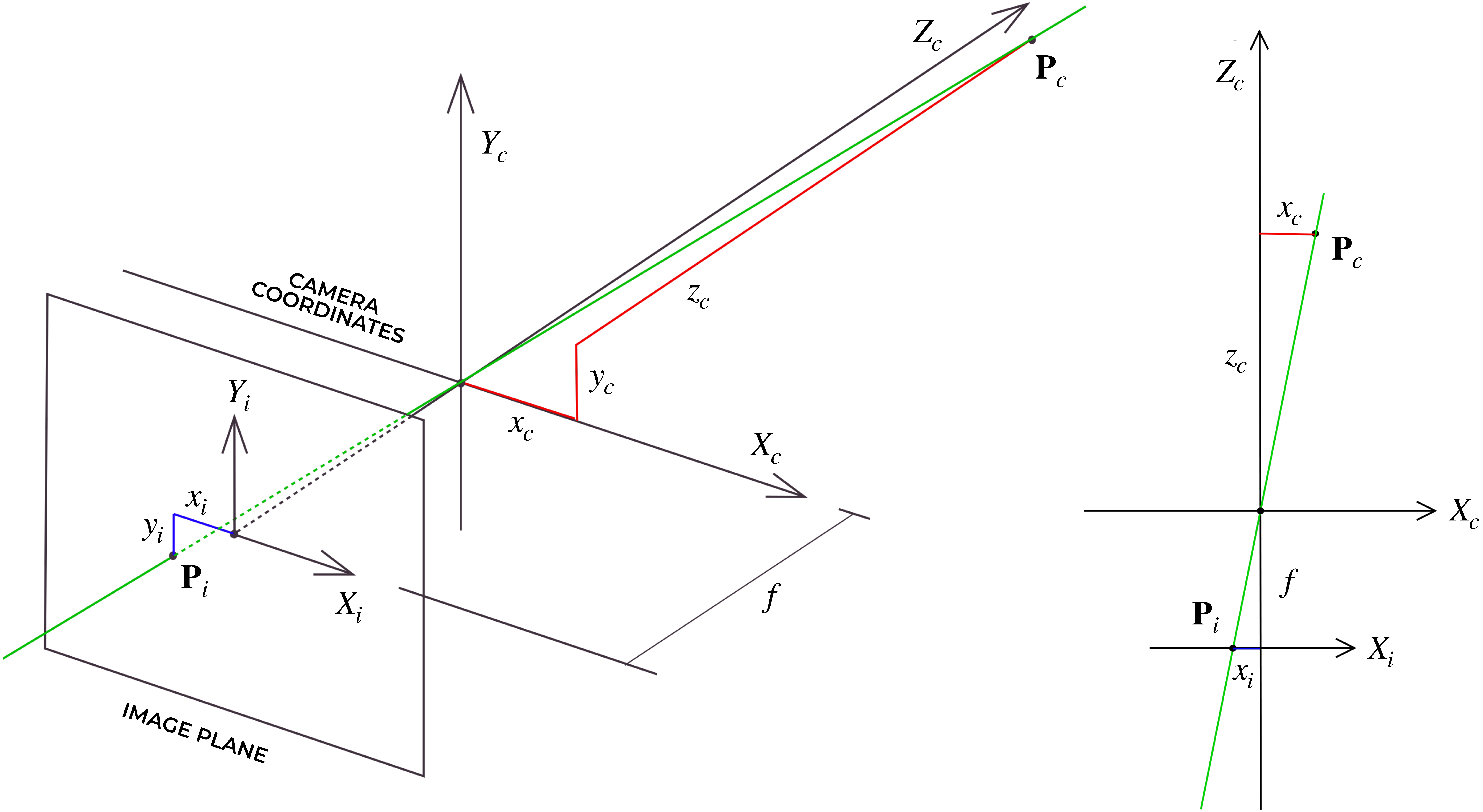


IMAGE COORDINATE SYSTEM

2D coordinate system that results from the projection of 3D points in the camera coordinate system using a *pinhole camera model.*

$$\mathbf{P}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$





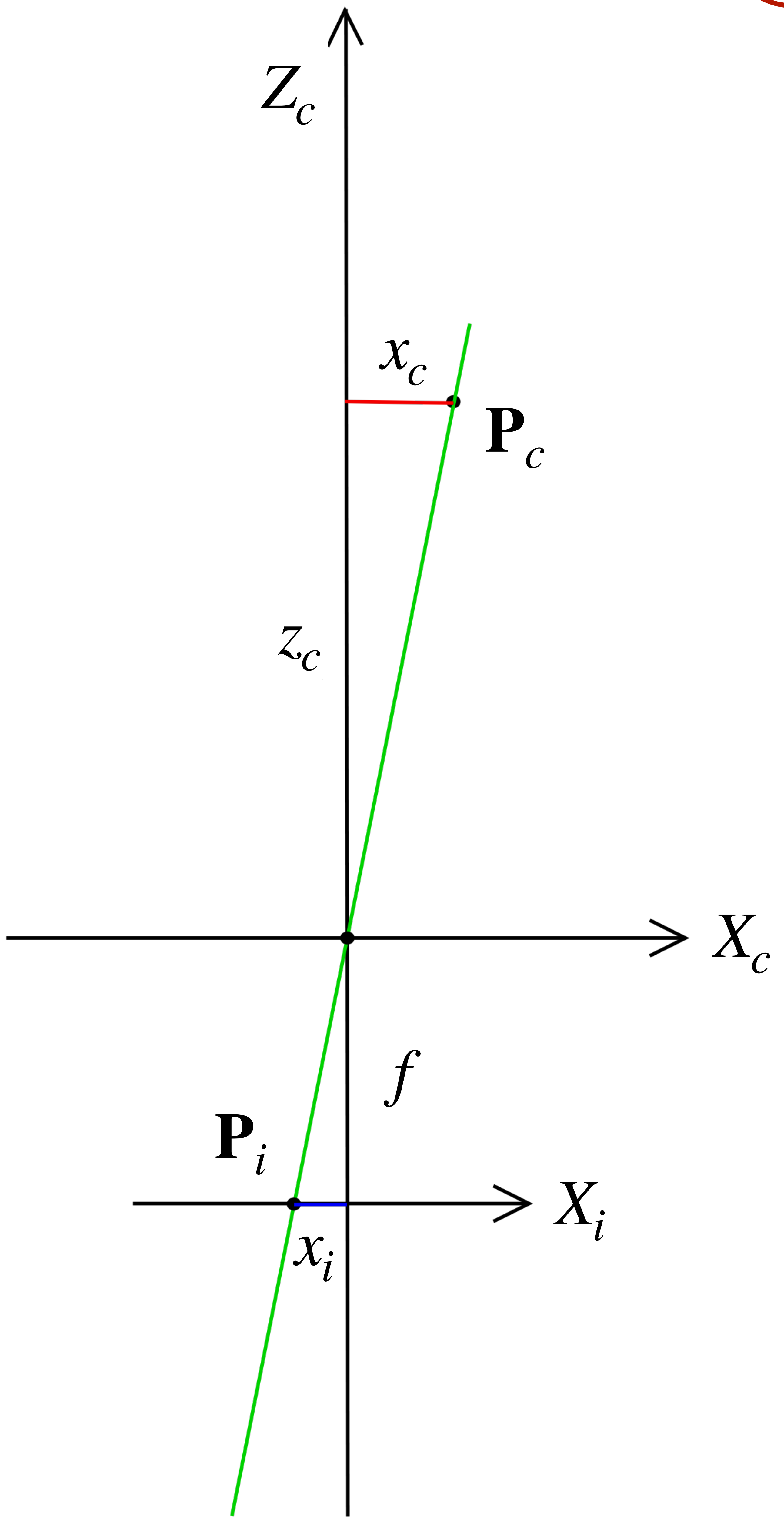
Z_c
$\mathbf{P}_c$
Y_c
CAMERA COORDINATES
z_c
y_c
x_c
X_c
Y_i
x_i
y_i
$\mathbf{P}_i$
X_i
f
IMAGE PLANE
Z_c
x_c
$\mathbf{P}_c$
z_c
X_c
f
$\mathbf{P}_i$
X_i
x_i

CAMERA (3D) → IMAGE (2D)

$$\frac{x_i}{f} = \frac{x_c}{z_c} \implies x_i = \frac{f x_c}{z_c}$$

$$\frac{y_i}{f} = \frac{y_c}{z_c} \implies y_i = \frac{f y_c}{z_c}$$

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \frac{f}{z_c} \begin{bmatrix} x_c \\ y_c \end{bmatrix}$$



CAMERA (3D) → IMAGE (2D)

In homogeneous coordinates:

$$\mathbf{P}_i = \mathbf{K}_1 \, \mathbf{P}_c \quad \Longrightarrow \quad \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

$\mathbf{K}_1$ = first part of the camera intrinsic matrix

PIXEL COORDINATE SYSTEM

The coordinate system that represents the integer values of pixels by discretizing the points in the image coordinate system.

The pixel coordinate system has origin in the left-top corner of the canvas.

$$\mathbf{P}' = \begin{bmatrix} u \\ v \end{bmatrix}$$

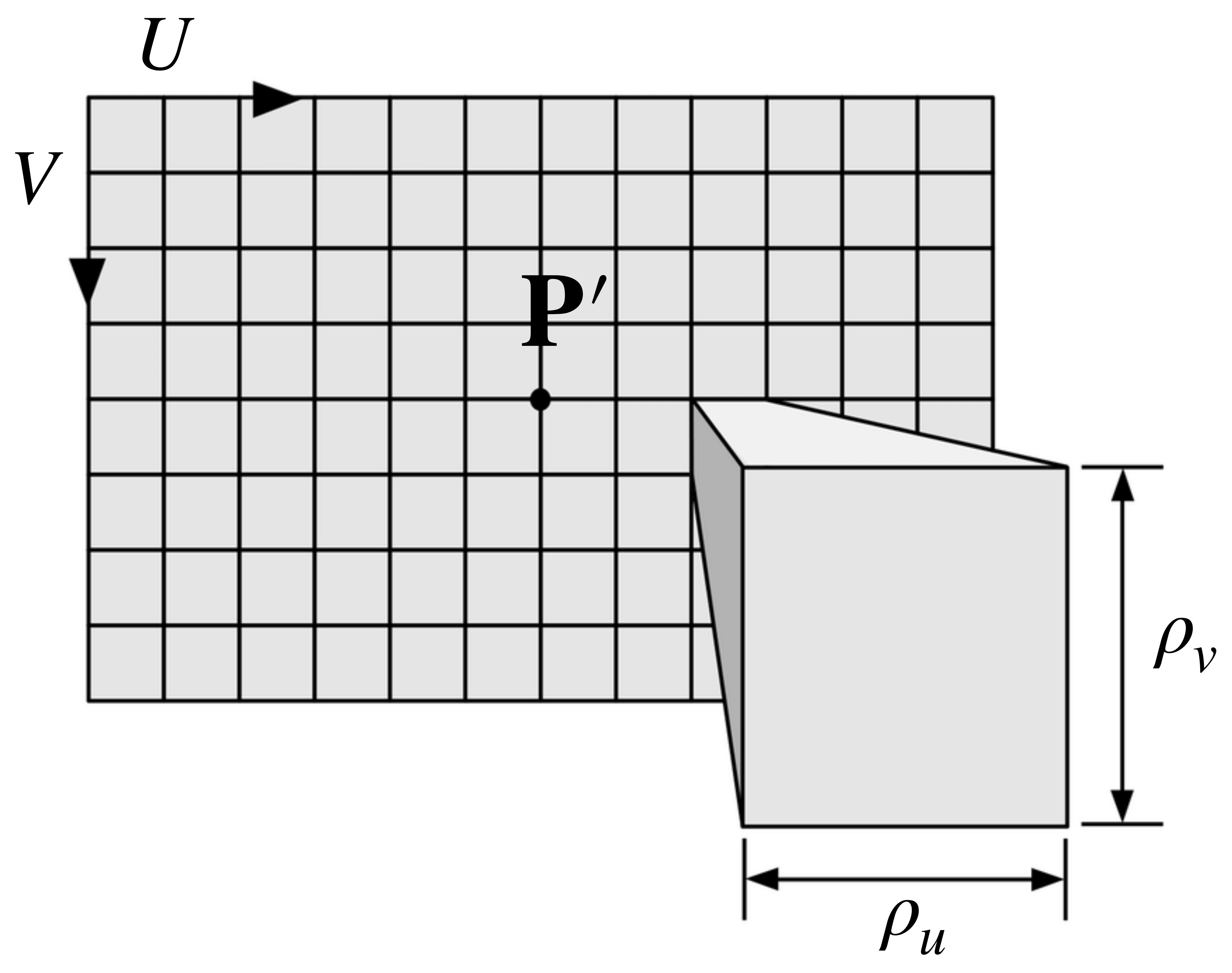


IMAGE (2D) ⟶ PIXEL (2D)

$$u = \frac{1}{\rho_u} x_i + t_u$$

$$v = t_v - \frac{1}{\rho_v} y_i$$

$$\mathbf{P}' = \mathbf{K}_2 \mathbf{P}_i \implies \begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} \frac{1}{\rho_u} & 0 & t_u \\ 0 & \frac{-1}{\rho_v} & t_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_c \end{bmatrix}$$

ρ_u , ρ_v = pixel width and height (meters)

$\mathbf{K}_2$ = second part of the camera intrinsic matrix

FULL TRANSFORMATION

$$\mathbf{P}' = \mathbf{K}\,\mathbf{Q}\,\mathbf{P}_w \quad \Longrightarrow \quad \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{f}{\rho_u} & 0 & t_u & 0 \\ 0 & \frac{f}{\rho_v} & t_v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{INTRINSIC MATRIX}} \underbrace{\left[\mathbf{R}_x \mathbf{R}_y \mathbf{R}_z \,|\, \mathbf{T}\right]}_{\text{EXTRINSIC MATRIX}} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

INTRINSIC MATRIX & EXTRINSIC MATRIX = **CAMERA MATRIX (3x4)**

CAMERA TRANSFORMATION (RECAP)

EXTRINSIC CAMERA MATRIX converts points from world coordinates to camera coordinates, and depends on the position and orientation of the camera.

- **WORLD-to-CAMERA:** 3D-3D projection. Rotation and translation.

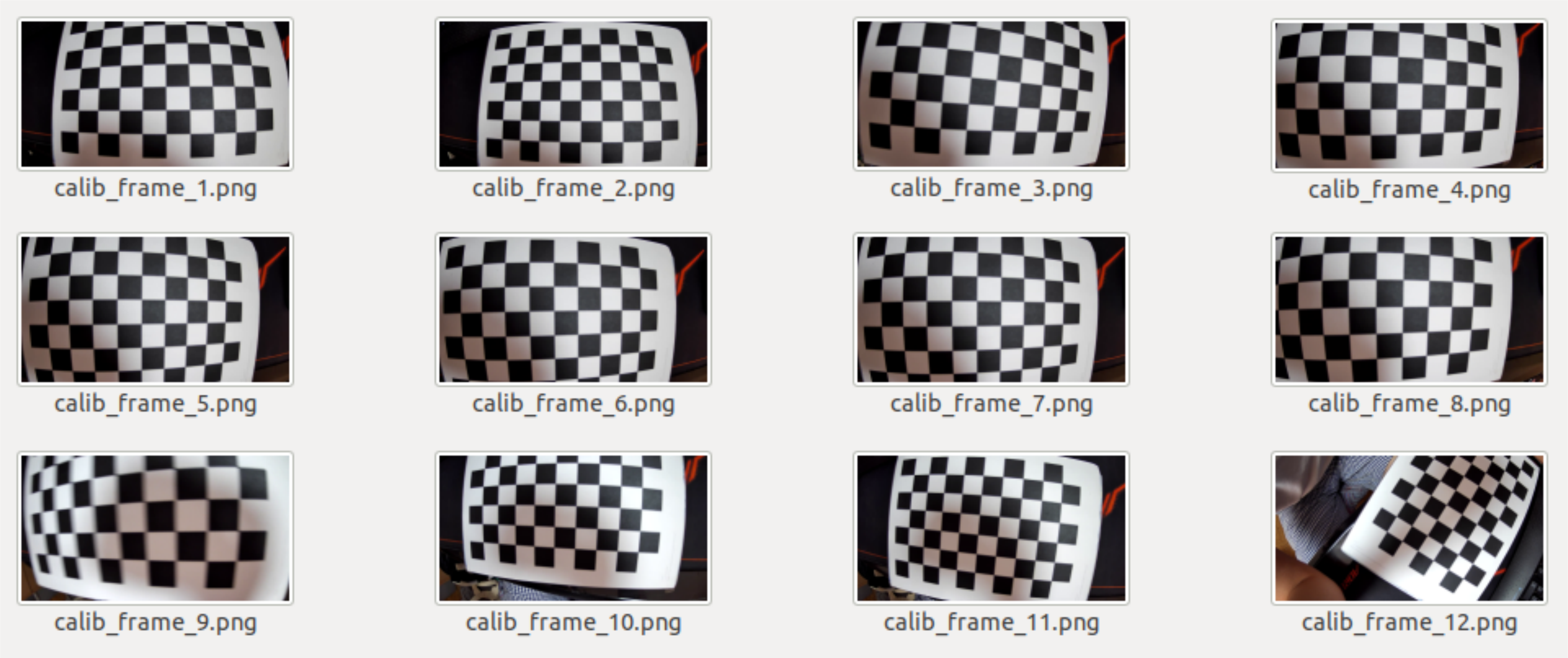
CAMERA TRANSFORMATION (RECAP)

INTRINSIC CAMERA MATRIX converts points from the camera coordinates to the pixel coordinates, and depends on camera properties (focal length, pixel dimensions, optical center, skew coefficient, lens distortion)

- **CAMERA-to-IMAGE:** 3D-2D projection. Loss of information (depth). Depends on the camera model.
- **IMAGE-to-PIXEL:** 2D-2D projection. Continuous to discrete. Quantization and origin shift.

CAMERA CALIBRATION

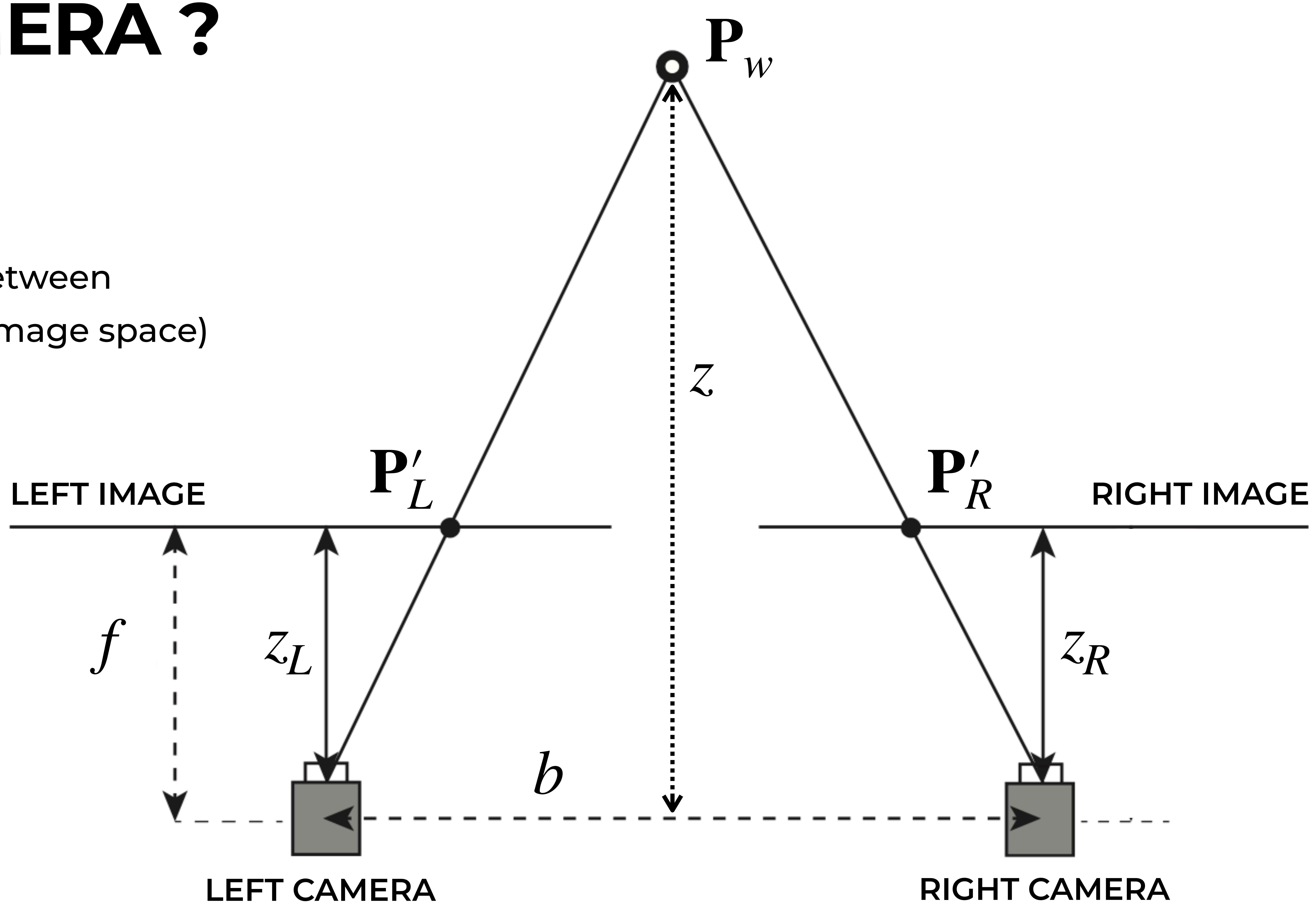
Estimates the camera matrix (12 parameters)



STEREO CAMERA ?

d = ***disparity*** (distance between $\mathbf{P}'_L$ and $\mathbf{P}'_R$ in the image space)

$$z = \frac{f\,b}{d}$$



PROS AND CONS

PROS

- rich semantic information
- cheap ($100 - $200)

CONS

- affected by light conditions (direct sunlight, darkness)
- affected by weather conditions (rain, fog, snow)

INFORMATION PROCESSING

COMPUTER VISION TASKS

- CLASSIFICATION
- CLASSIFICATION + LOCALIZATION
- OBJECT DETECTION
- OBJECT TRACKING
- SEMANTIC SEGMENTATION
- INSTANCE SEGMENTATION

COMPUTER VISION TASKS

- CLASSIFICATION
- CLASSIFICATION + LOCALIZATION
- OBJECT DETECTION
- OBJECT TRACKING
- SEMANTIC SEGMENTATION
- INSTANCE SEGMENTATION

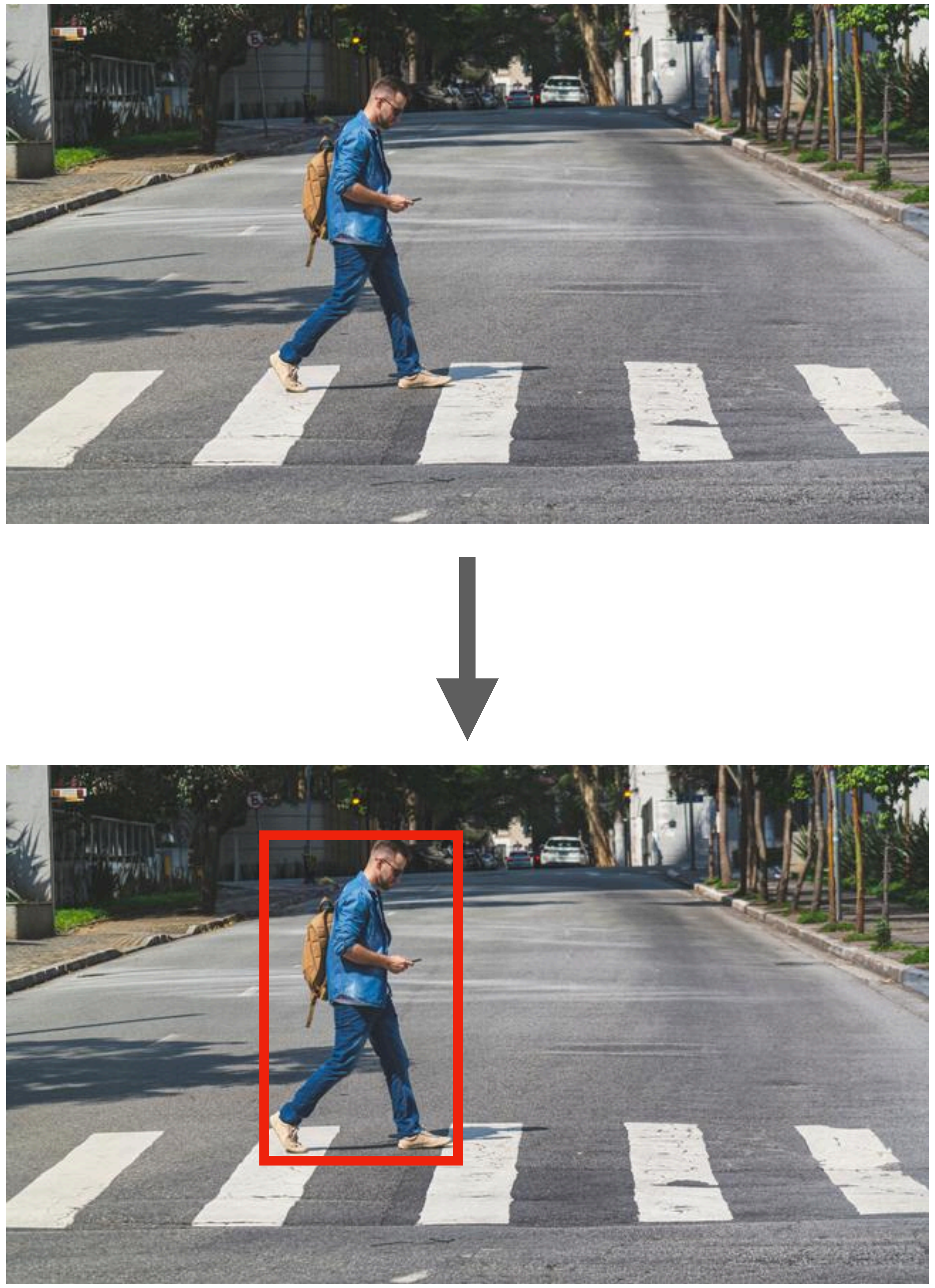
{ car, truck, bike, pedestrian }

↓

CAR

COMPUTER VISION TASKS

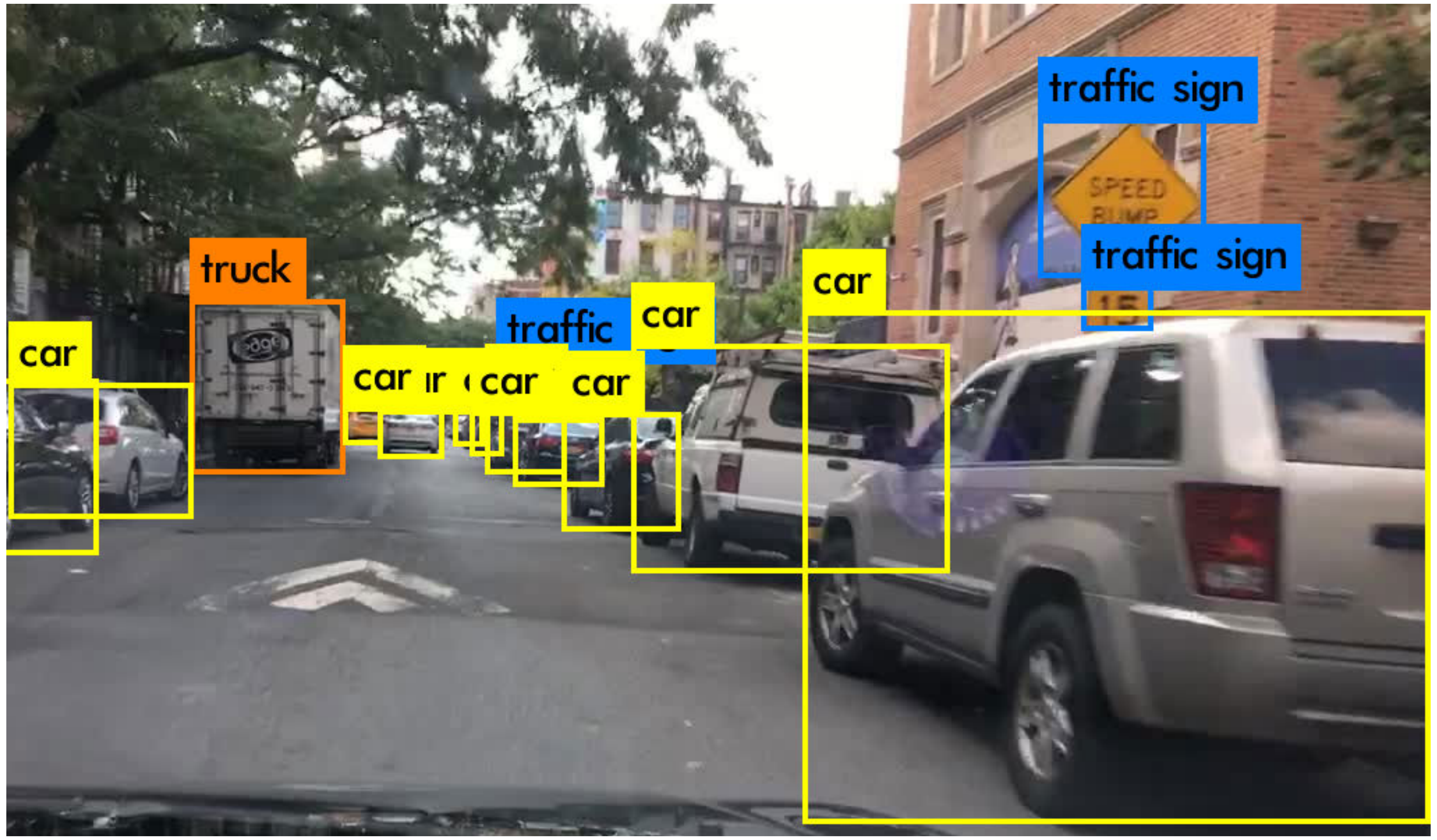
- CLASSIFICATION
- CLASSIFICATION + LOCALIZATION
- OBJECT DETECTION
- OBJECT TRACKING
- SEMANTIC SEGMENTATION
- INSTANCE SEGMENTATION



PEDESTRIAN

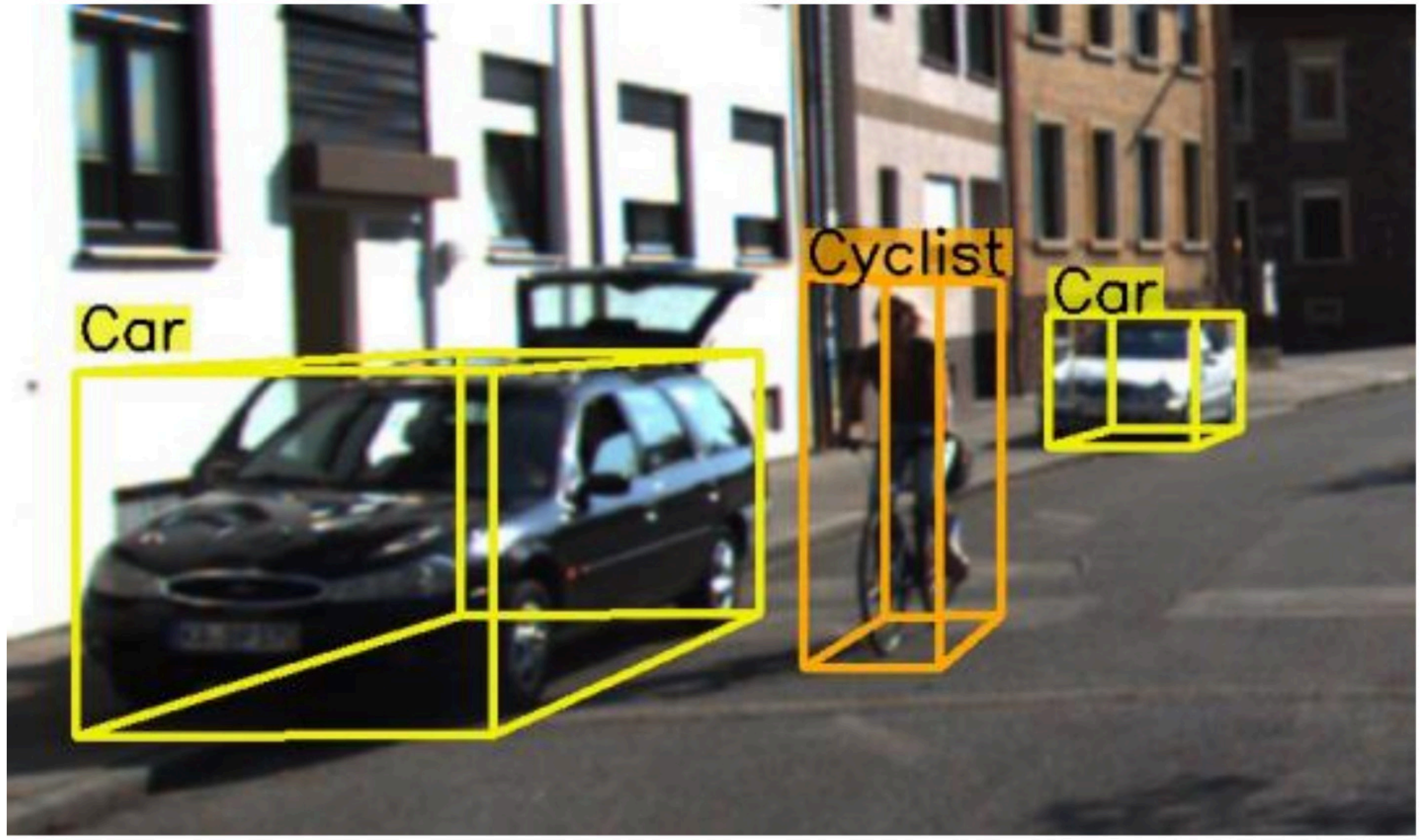
COMPUTER VISION TASKS

- CLASSIFICATION
- CLASSIFICATION + LOC
- OBJECT DETECTION
- OBJECT TRACKING
- SEMANTIC SEGMENTATION
- INSTANCE SEGMENTATION



COMPUTER VISION TASKS

- CLASSIFICATION
- CLASSIFICATION + LOC
- OBJECT DETECTION
- OBJECT TRACKING
- SEMANTIC SEGMENTATION
- INSTANCE SEGMENTATION



COMPUTER VISION TASKS

- CLASSIFICATION
- CLASSIFICATION + LOC
- OBJECT DETECTION
- OBJECT TRACKING
- SEMANTIC SEGMENTATION
- INSTANCE SEGMENTATION

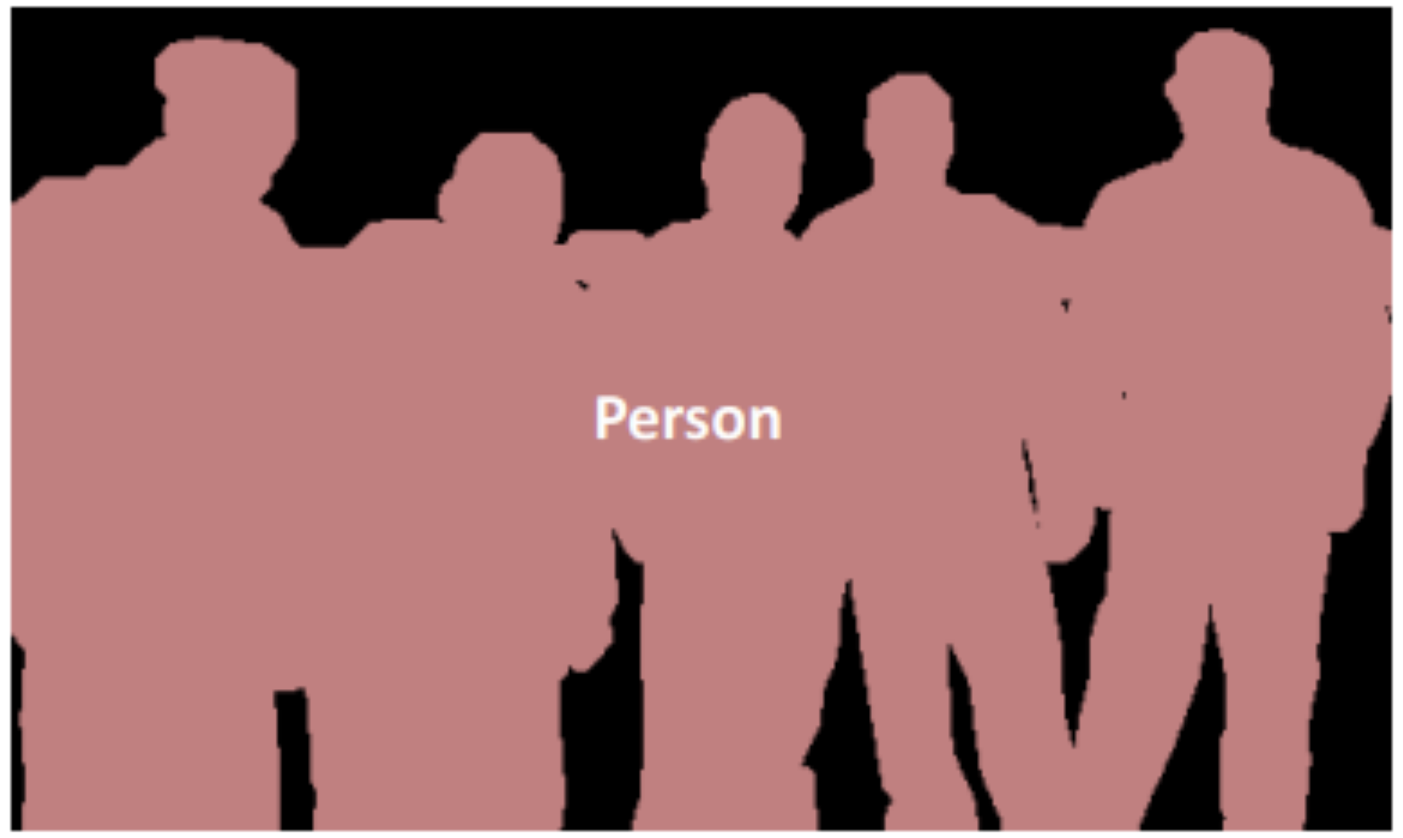
SEMANTIC SEGMENTATION

COMPUTER VISION TASKS

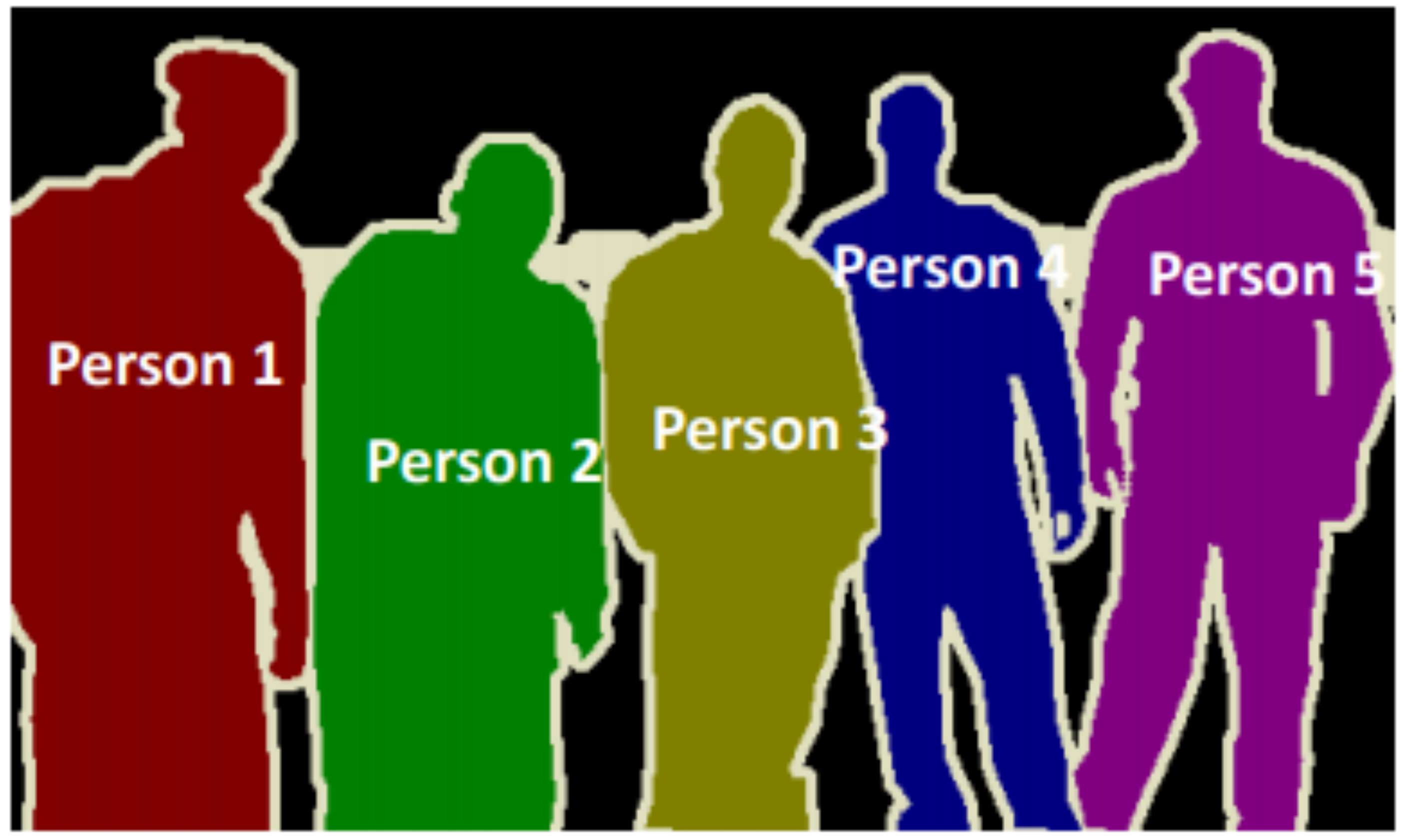
- CLASSIFICATION
- CLASSIFICATION + LOC
- OBJECT DETECTION
- OBJECT TRACKING
- SEMANTIC SEGMENTATION
- INSTANCE SEGMENTATION

COMBINING TASKS

OBJECT
DETECTION

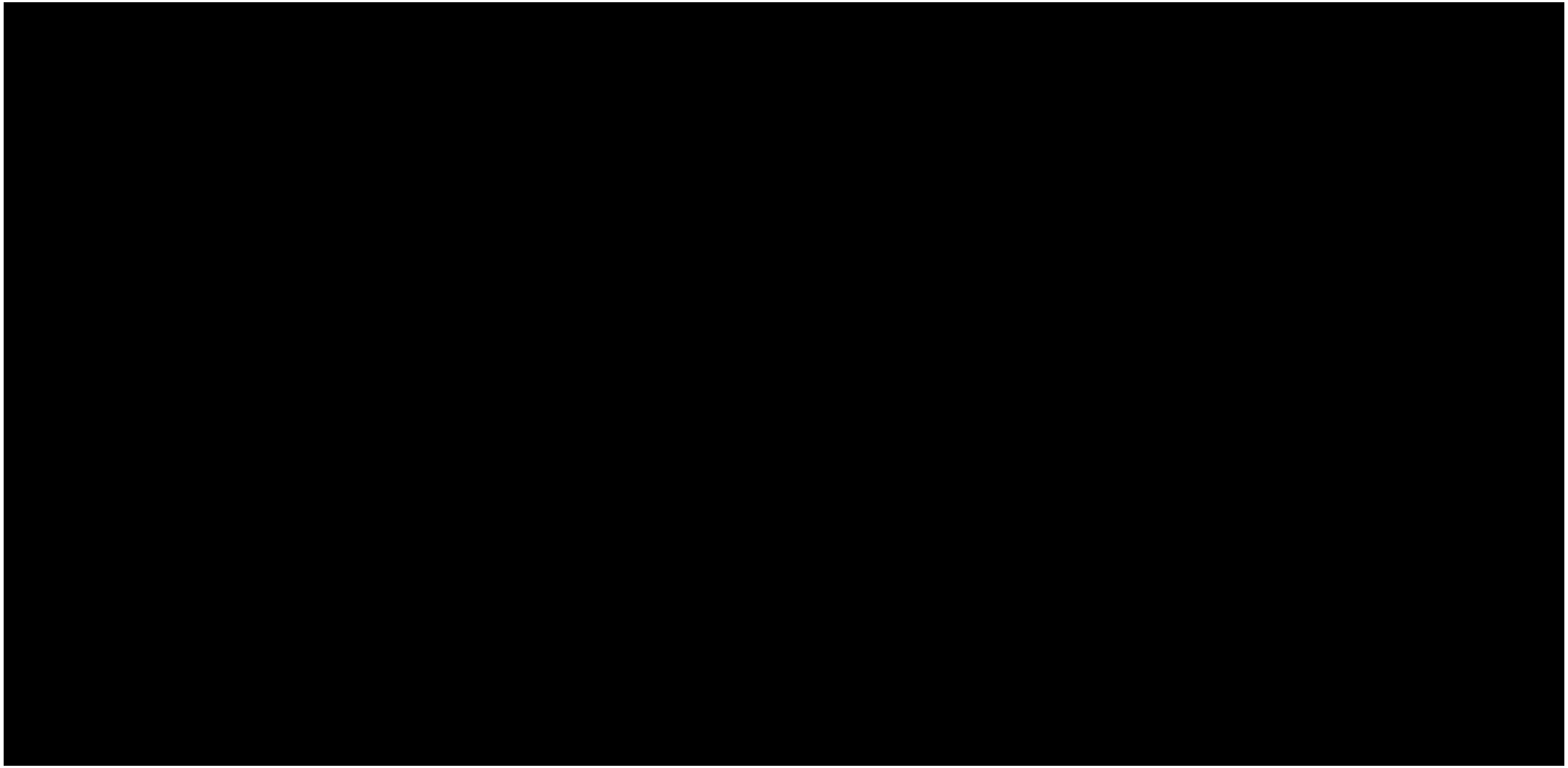


SEMANTIC
SEGMENTATION



INSTANCE
SEGMENTATION

COMBINING TASKS



SENSOR FUSION