Humanoid Robotics

Perception 1: Basics

Maren Bennewitz

Goal of This Chapter

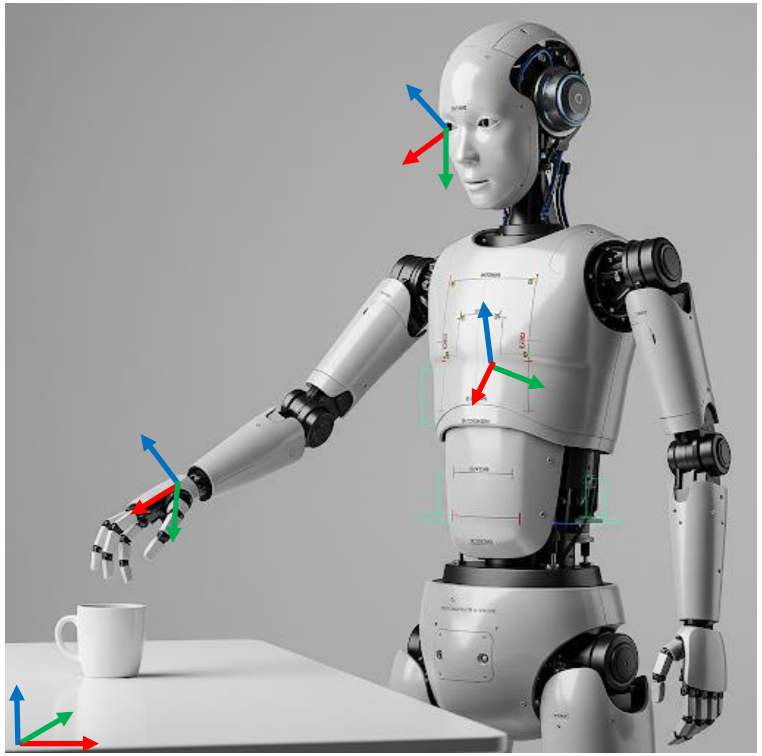
- Understand the concept of **transformations between coordinate systems** and why they are needed
- Learn a representation for transformations that allows for **easy chaining** and **inversion** of transformations
- Understand the mapping from the **world coordinate system** to the **robot's sensor coordinate system**

Why are Coordinate Transforms Needed

- **Accurate perception, motion planning, and control**
- Data from sensors (e.g., cameras, IMUs, joint encoders) are **captured in different reference frames**
- Motion planning (e.g., for object manipulation and navigation) requires **transforming between global, local, and joint-specific coordinates**

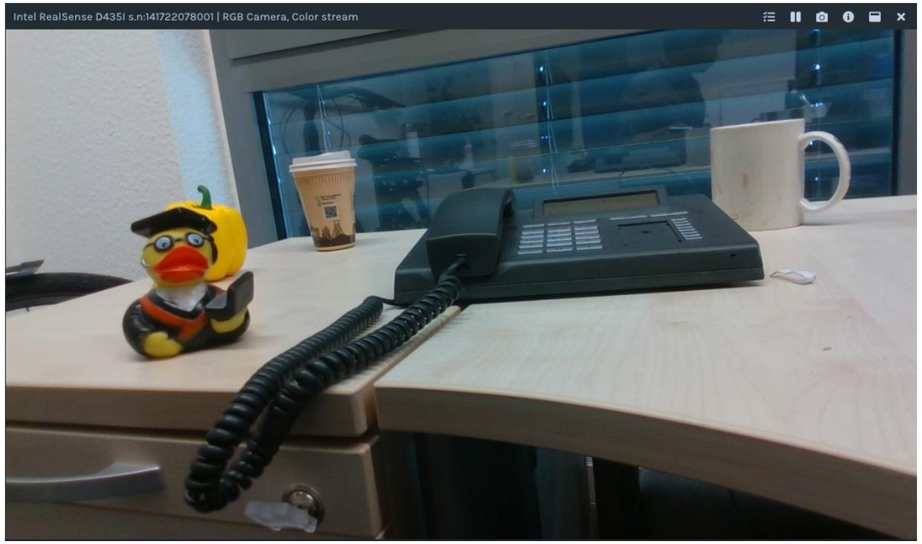
Example: Reaching for an Object

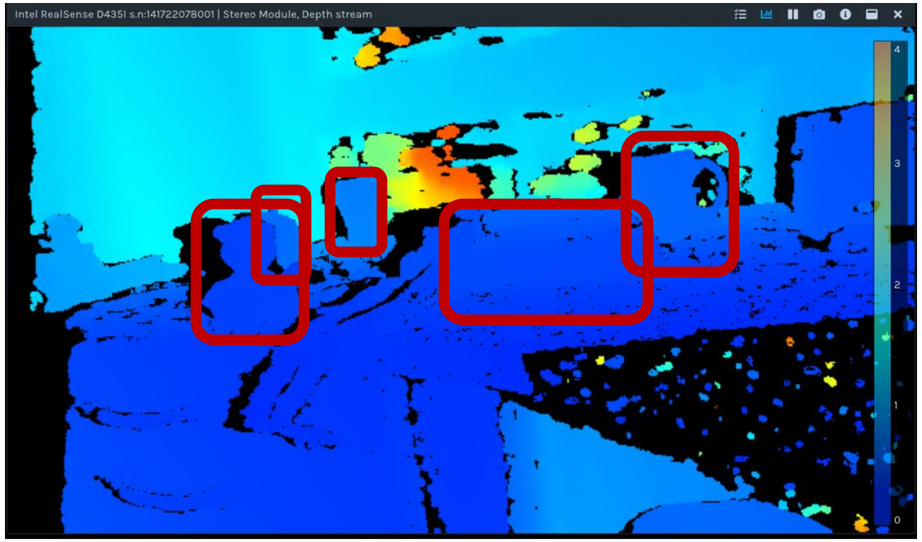
- **World Frame:** Global origin
- **Camera Frame:** Coordinates of the camera in the world, needed for accurate perception
- **Body Frame:** Coordinates of the robot's torso, needed for motion planning
- **End-Effector Frame:** Coordinates of the hand, ensures the robot's hand precisely aligns with object positions



generated by Gemini

Depth Image of RGB-D Camera

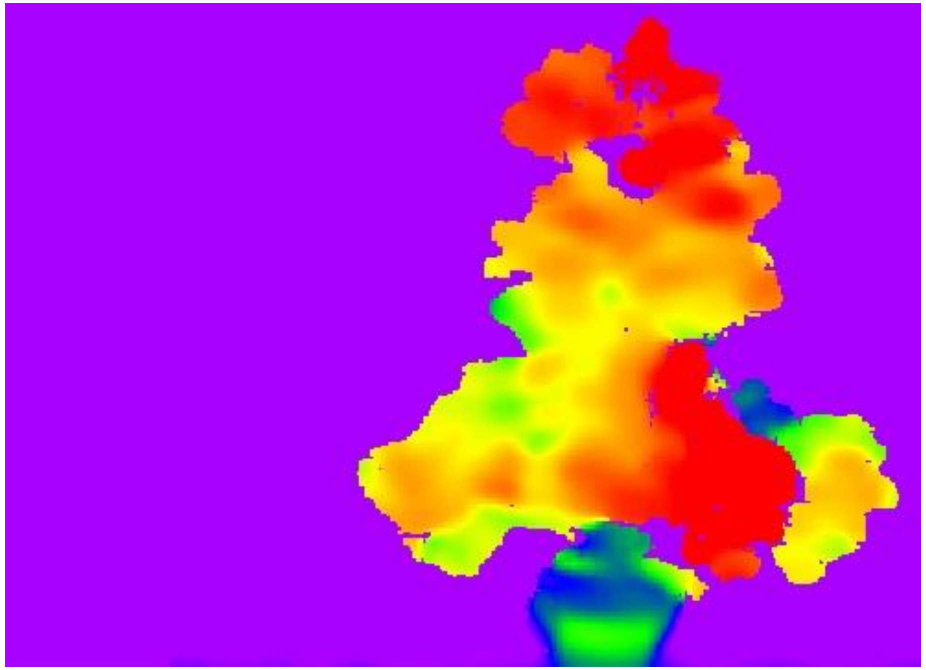




Intel Realsense D435i

Image courtesy: Intel

Example Scene – Depth Data



close, medium, more distant,
not visible

How to Measure Depth

- **Time-of-flight principle**: send laser pulses and measure their return time
- **Structured light**: projects a pattern onto the environment and uses its deformation resulting from hitting objects
- **Stereo vision**: comparing the disparity between objects in the two images

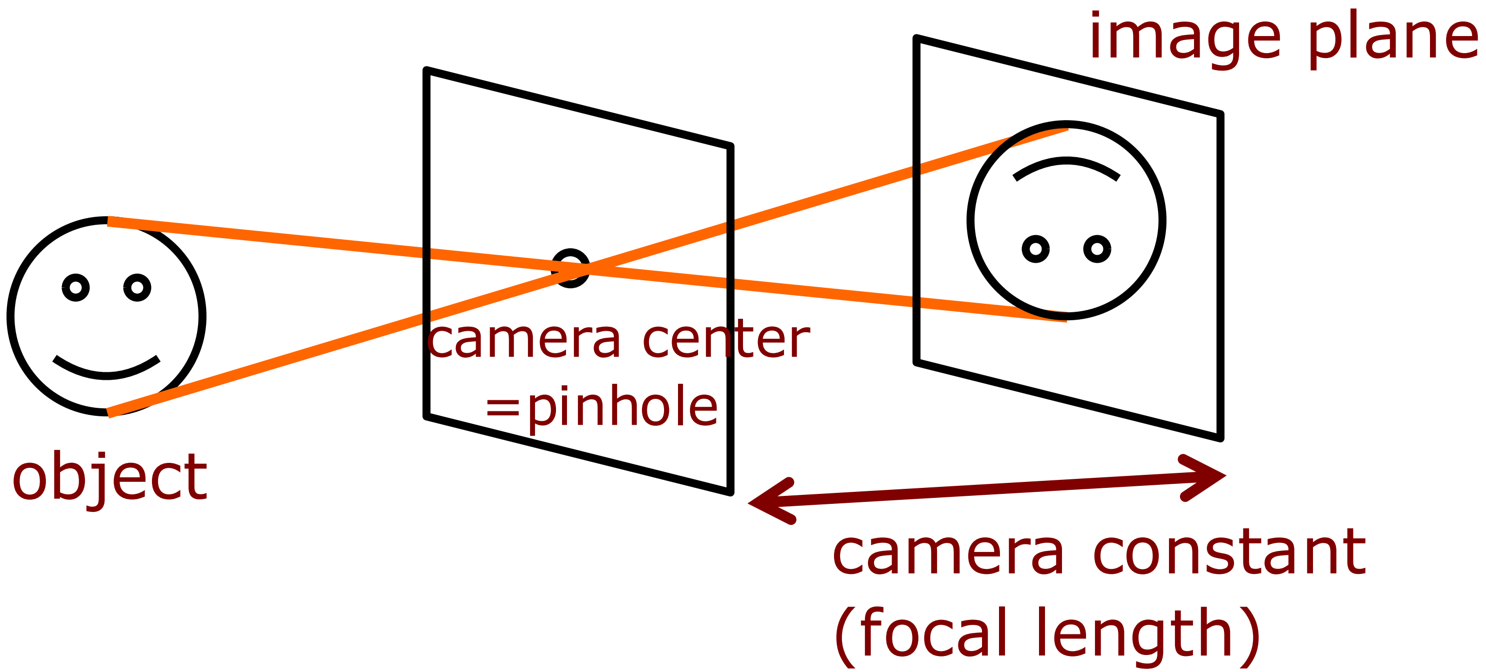
Motivation of Homogeneous Coordinates

- Cameras generate a projected image of the 3D world
- In Euclidian geometry, the math for describing this transformation gets difficult
- **Projective geometry**: alternative algebraic representation of geometric transformations
- So-called **homogeneous coordinates** are typically used in robotics
- Advantage: affine transformations and projective transformations can be expressed with **one matrix multiplication**

Pinhole Camera Model

- Describes the projection of a 3D world point into the camera image
- Assumption: box with an **infinitesimal small hole**
- The **camera center** is the intersection point of the rays, i.e., the pinhole
- The back wall is the **image plane**
- The distance between the camera center and image plane is the **camera constant**

Pinhole Camera Model



Projective Mapping

- Straight lines stay straight
- Parallel lines may intersect

Image courtesy: W. Förstner

Pinhole Camera Model: Properties

- **Line-preserving:** straight lines are mapped to straight lines
- **Not angle-preserving:** angles between lines change
- **Not length-preserving:** size of objects is inverse proportional to the distance

Homogeneous Coordinates (H.C.)

- H.C. are a system of coordinates used in projective geometry
- Formulas using H.C. are often simpler than using Euclidian coordinates
- A single matrix can represent affine and projective transformations

Notation

Point

- in homogeneous coordinates $\mathbf{x}$
- in Euclidian coordinates $\boldsymbol{x}$

2D vs. **3D** space

- lowercase = 2D
- capitalized = 3D

Homogeneous Coordinates

Definition:

The representation $\mathbf{x}$ of a geometric object is **homogeneous** if $\mathbf{x}$ and $\lambda\mathbf{x}$ represent the same object for $\lambda \neq 0$

Example:

$$\mathbf{x} = \lambda\,\mathbf{x}$$

homogeneous

$$\boldsymbol{x} \neq \lambda\,\boldsymbol{x}$$

Euclidian

Homogeneous Coordinates

- H.C. use n+1 dimensional vectors to represent n-dimensional points given in Euclidean coordinates
- Example:

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = w \begin{bmatrix} u/w \\ v/w \\ 1 \end{bmatrix} = \begin{bmatrix} u/w \\ v/w \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Definition

- **Homogeneous coordinates** of a point χ in $\mathbb{R}^2$ is a 3-dimensional vector

$$\chi : \quad \mathbf{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \text{ with } |\mathbf{x}|^2 = u^2 + v^2 + w^2 \neq 0$$

- Corresponding **Euclidian coordinates**

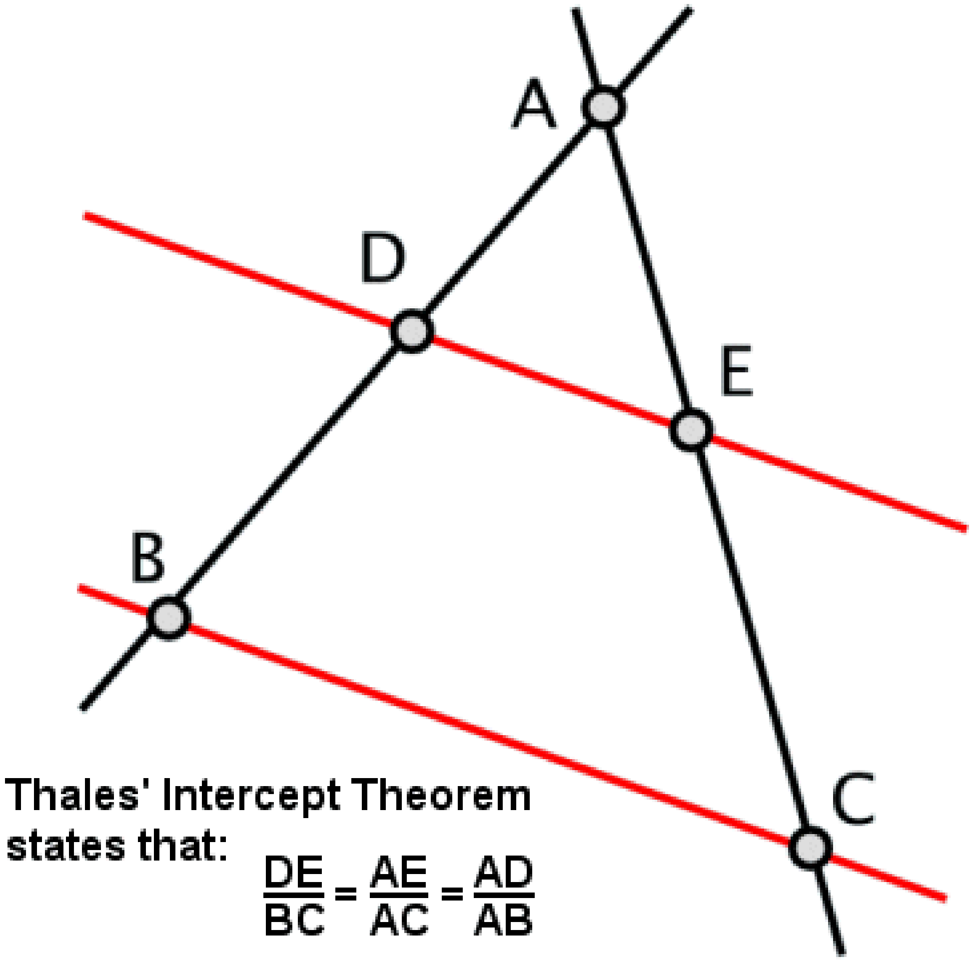
$$\chi : \quad \boldsymbol{x} = \begin{bmatrix} u/w \\ v/w \end{bmatrix} \text{ with } w \neq 0$$

From Homogeneous to Euclidian Coordinates

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} u/w \\ v/w \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} u/w \\ v/w \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

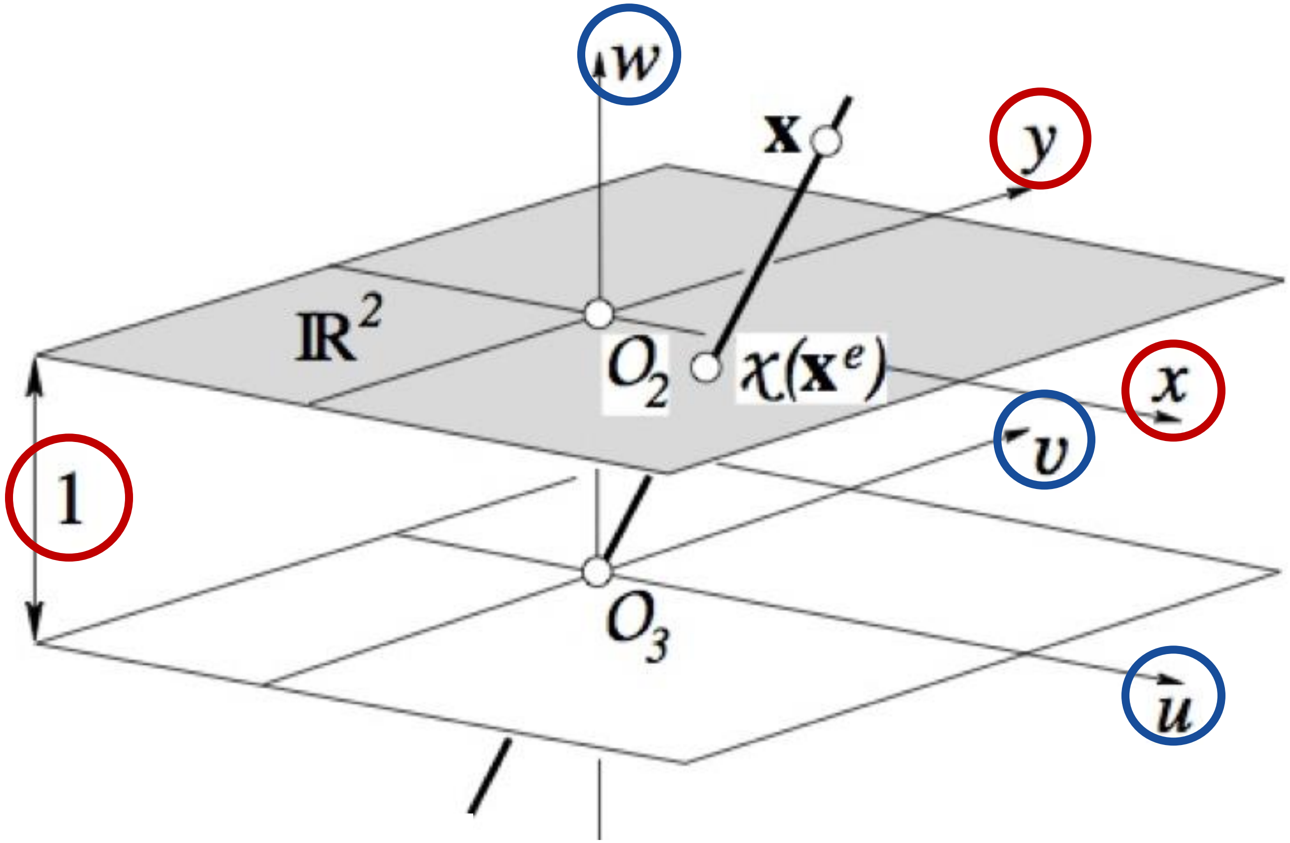
visual derivation on the next slides

Reminder: Intercept Theorem



$$\frac{BC}{AC} = \frac{DE}{AE}$$

From Homogeneous to Euclidian Coordinates



$$\frac{u}{w} = \frac{x}{1}$$

$$\frac{v}{w} = \frac{y}{1}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} u/w \\ v/w \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} u/w \\ v/w \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

Image courtesy: Förstner

3D Points

For 3D points in Euclidian space:

homogeneous **Euclidian**

$$\mathbf{X} = \begin{bmatrix} U \\ V \\ W \\ T \end{bmatrix} = \begin{bmatrix} U/T \\ V/T \\ W/T \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} U/T \\ V/T \\ W/T \end{bmatrix}$$

Origin of the Euclidian Coordinate System in Homogeneous Coordinates

$$\mathbf{O}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \mathbf{O}_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

3D Transformations

- Linear mapping

$$\mathbf{X}' = \mathsf{H}\mathbf{X}$$

- 3D translation: 3 parameters

$$\mathsf{H} = \begin{bmatrix} \mathit{I} & \boldsymbol{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$$

$$\mathit{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

$$\mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

3D Transformations

- Rotation: 3 parameters

$$H = \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

rotation
matrix

Reminder: Rotation Matrices

- 2D:

$$R^{2D}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

- 3D:

$$R_x^{3D}(\omega) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\omega) & -\sin(\omega) \\ 0 & \sin(\omega) & \cos(\omega) \end{bmatrix} \quad R_y^{3D}(\phi) = \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix}$$

$$R_z^{3D}(\kappa) = \begin{bmatrix} \cos(\kappa) & -\sin(\kappa) & 0 \\ \sin(\kappa) & \cos(\kappa) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R^{3D}(\omega, \phi, \kappa) = R_z^{3D}(\kappa) R_y^{3D}(\phi) R_x^{3D}(\omega)$$

3D Transformations

- Rigid body transformation: 6 parameters
- 3 translation + 3 rotation

$$H = \begin{bmatrix} R & \boldsymbol{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}$$

Inverting and Chaining

- Inverting a transformation

$$\begin{aligned} \mathbf{X}' &= \mathsf{H}\mathbf{X} \\ \mathbf{X} &= \mathsf{H}^{-1}\mathbf{X}' \end{aligned}$$

- Chaining transformations via matrix products (not commutative)

$$\begin{aligned} \mathbf{X}' &= \mathsf{H}_1\mathsf{H}_2\mathbf{X} \\ &\neq H_2\mathsf{H}_1\mathbf{X} \end{aligned}$$

Coordinate Frames

1. World coordinate frame
2. Camera coordinate frame
3. Image coordinate frame
4. Sensor coordinate frame

Coordinate Frames

1. World coordinate frame S_o
   written as: $[X, Y, Z]^\mathsf{T}$
2. Camera coordinate frame S_k
   written as: $[\,{}^kX,\ {}^kY,\ {}^kZ]^\mathsf{T}$
3. Image coordinate frame S_c
   written as: $[\,{}^cx,\ {}^cy]^\mathsf{T}$
4. Sensor coordinate frame S_s
   written as: $[\,{}^sx,\ {}^sy]^\mathsf{T}$

Transformation from World to Sensor

Goal: Compute the mapping

$$\begin{bmatrix} {}^s x \\ {}^s y \\ 1 \end{bmatrix} = {}^s\mathsf{H}_c \, {}^c\mathsf{H}_k \, {}^k\mathsf{H}_o \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

in the sensor frame	image to sensor	camera to image	world to camera	in the world frame

Visualization of the Transformation

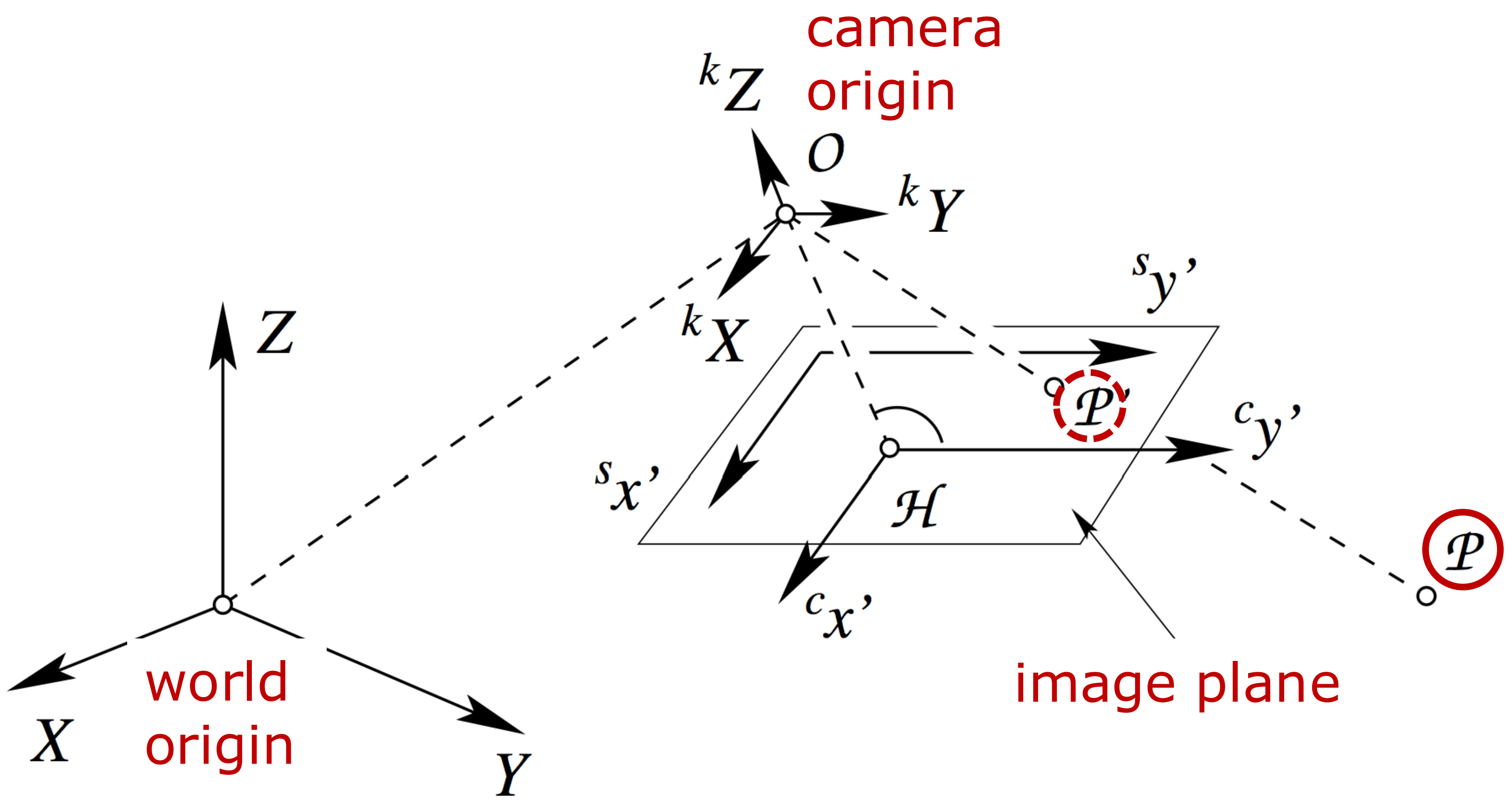
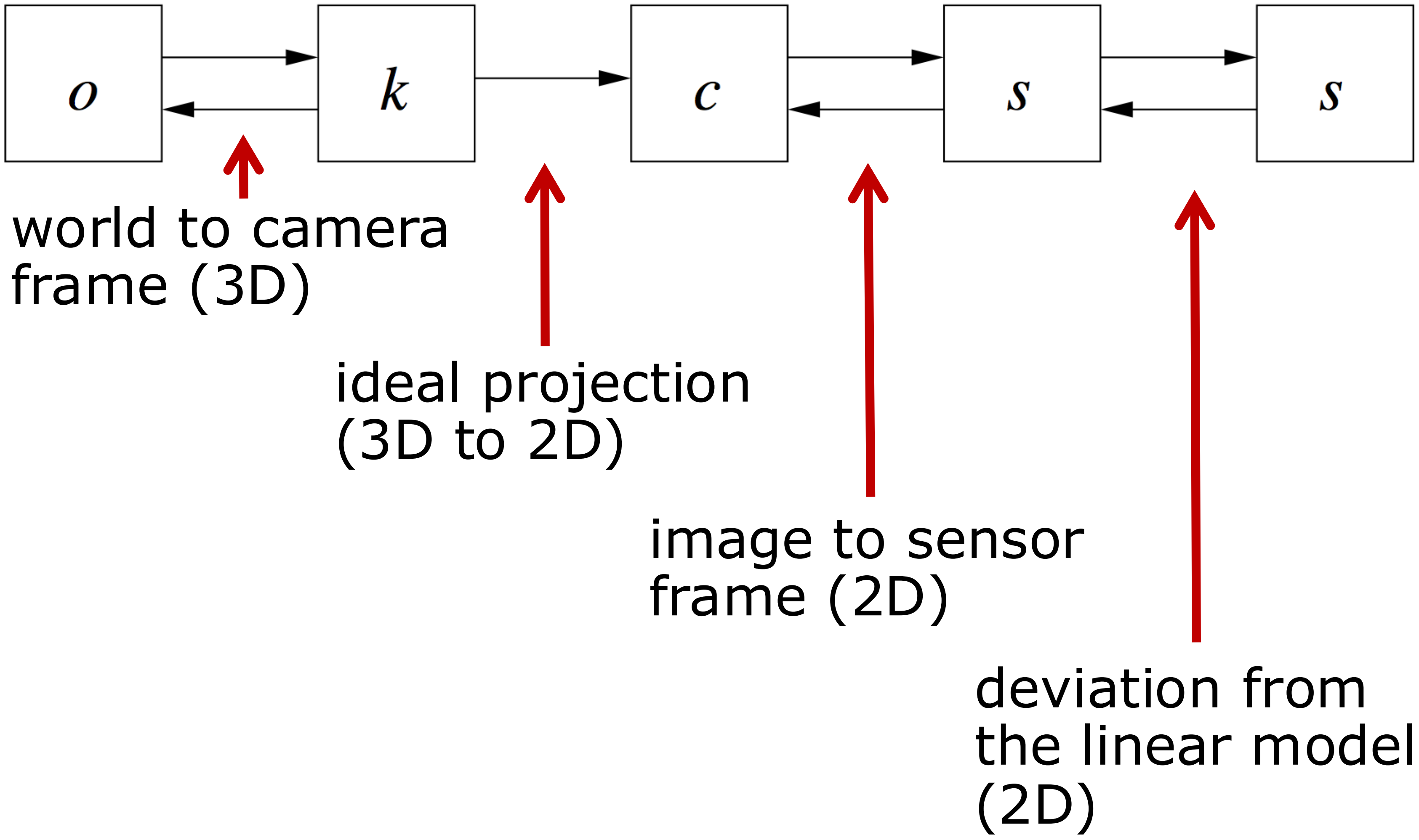
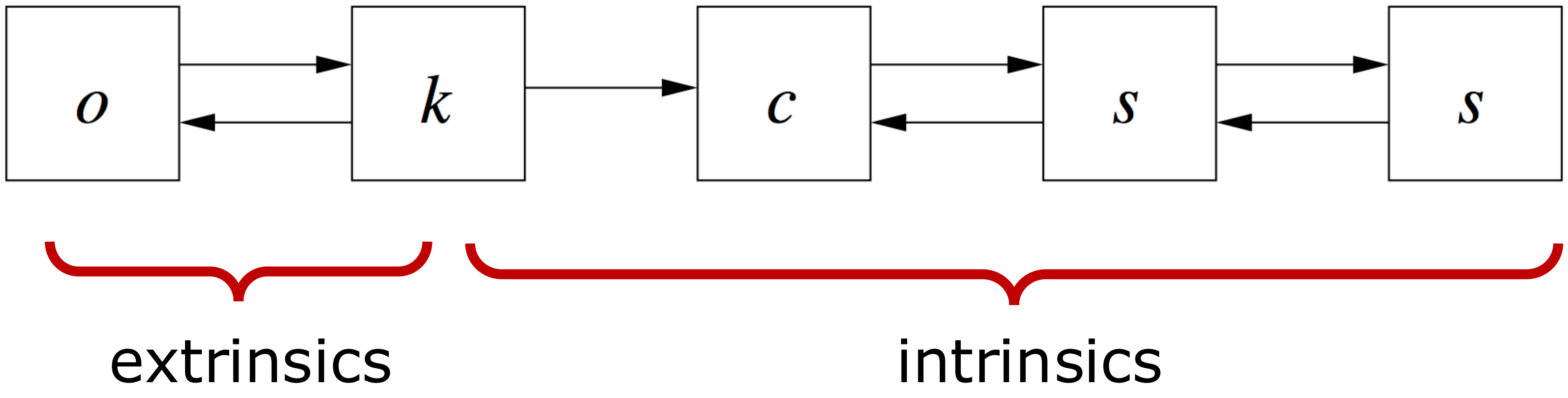


Image courtesy: Förstner

From the World to the Sensor



Extrinsic and Intrinsic Parameters



- **Extrinsic parameters** describe the pose of the camera in the world
- **Intrinsic parameters** describe the mapping of the scene in front of the camera to the pixels in the final image (sensor)

Extrinsic Parameters

- Pose of the camera with respect to the world
- Invertible transformation

How many parameters are needed?

6 parameters: 3 for the position and
3 for the orientation

Extrinsic Parameters

- Point $\mathcal{P}$ with coordinates in world coordinates

$$\boldsymbol{X}_P = [X_P, Y_P, Z_P]^\mathsf{T}$$

- Origin of the camera frame

$$\boldsymbol{X}_O = [X_O, Y_O, Z_O]^\mathsf{T}$$

Transformation

- **Translation** between the origin of the world frame and the camera frame

$$\boldsymbol{X}_O = [X_O, Y_O, Z_O]^\mathsf{T}$$

- **Rotation** R from the frame S_o to S_k

- **Complete transform:**

$${}^k\mathbf{X}_P = {}^k\mathsf{H}\,\mathbf{X}_P \quad \text{with} \quad {}^k\mathsf{H} = \begin{bmatrix} R & -R\boldsymbol{X}_O \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$$

homogeneous coordinates

Transformation: Derivation

- In **Euclidian coordinates** ${}^k\boldsymbol{X}_P = R(\boldsymbol{X}_P - \boldsymbol{X}_O)$
- Expressed in **homogeneous coordinates**

$$\begin{bmatrix} {}^k\boldsymbol{X}_P \\ 1 \end{bmatrix} = \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} I_3 & -\boldsymbol{X}_O \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_P \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} R & -R\boldsymbol{X}_O \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_P \\ 1 \end{bmatrix}$$

Euclidian coordinates

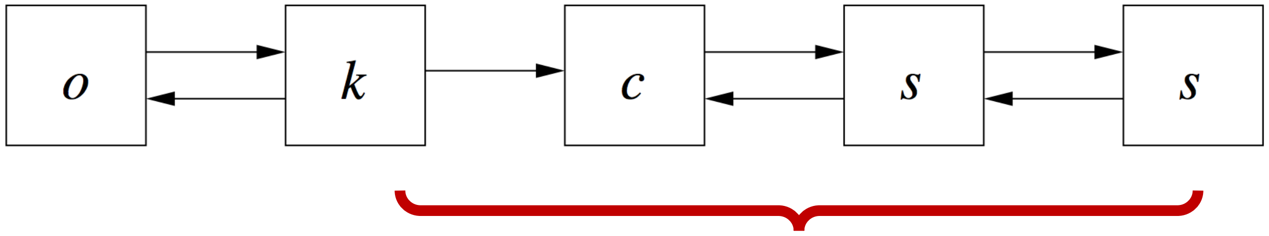
- or written as

$${}^k\mathbf{X}_P = {}^k\mathsf{H}\,\mathbf{X}_P \quad \text{with} \quad {}^k\mathsf{H} = \begin{bmatrix} R & -R\boldsymbol{X}_O \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$$

homogeneous coordinates

Intrinsic Parameters

- For the process of projecting points from the **camera frame** to the **sensor frame**
- Invertible transformations:
  - Image plane to sensor frame
  - Model deviations
- Not directly invertible: projection



Mapping from Camera to Sensor Frame (1)

$$\mathsf{K} = \begin{bmatrix} c & 0 & x_H \\ 0 & c(1+m) & y_H \\ 0 & 0 & 1 \end{bmatrix}$$

- **Calibration matrix** contains 4 parameters:
  - Camera constant: c
  - Offset from image frame origin: x_H, y_H
  - Scale difference: m

Mapping from Camera to Sensor Frame (2)

- Final transformation to cover the non-linear effects
- **Location-dependent** shift in the sensor coordinate system
- Individual shift for each pixel according to the distance from the image center

$$
{}^{a}\mathsf{K}(\boldsymbol{x},\boldsymbol{q}) = \begin{bmatrix} c & 0 & x_H + \Delta x(\boldsymbol{x},\boldsymbol{q}) \\ 0 & c(1+m) & y_H + \Delta y(\boldsymbol{x},\boldsymbol{q}) \\ 0 & 0 & 1 \end{bmatrix}
$$

Calibrated Camera

- If the intrinsics are **unknown**, we call the camera **uncalibrated**
- If the intrinsics are **known**, we call the camera **calibrated**
- The process of obtaining the intrinsics is called **camera calibration**

Summary (1)

- Homogeneous coordinates are an alternative representation for transformations
- Simplify mathematical expressions
- Allow for easy chaining and inversion of transformations
- Modeled through an extra dimension

Summary (2)

- Mapping from the world frame to the sensor frame
- **Extrinsics** = world to camera frame
- **Intrinsics** = camera to sensor frame
- Assumption: Pinhole camera model
- Non-linear model for lens distortion
- We need to know the camera parameters to
  - Map from the world coordinate system to the sensor coordinate system
  - Realize robot interaction with the real world

Literature

- Multiple View Geometry in Computer Vision, R. Hartley and A. Zisserman, Ch. 2, Ch. 3, and Ch. 6