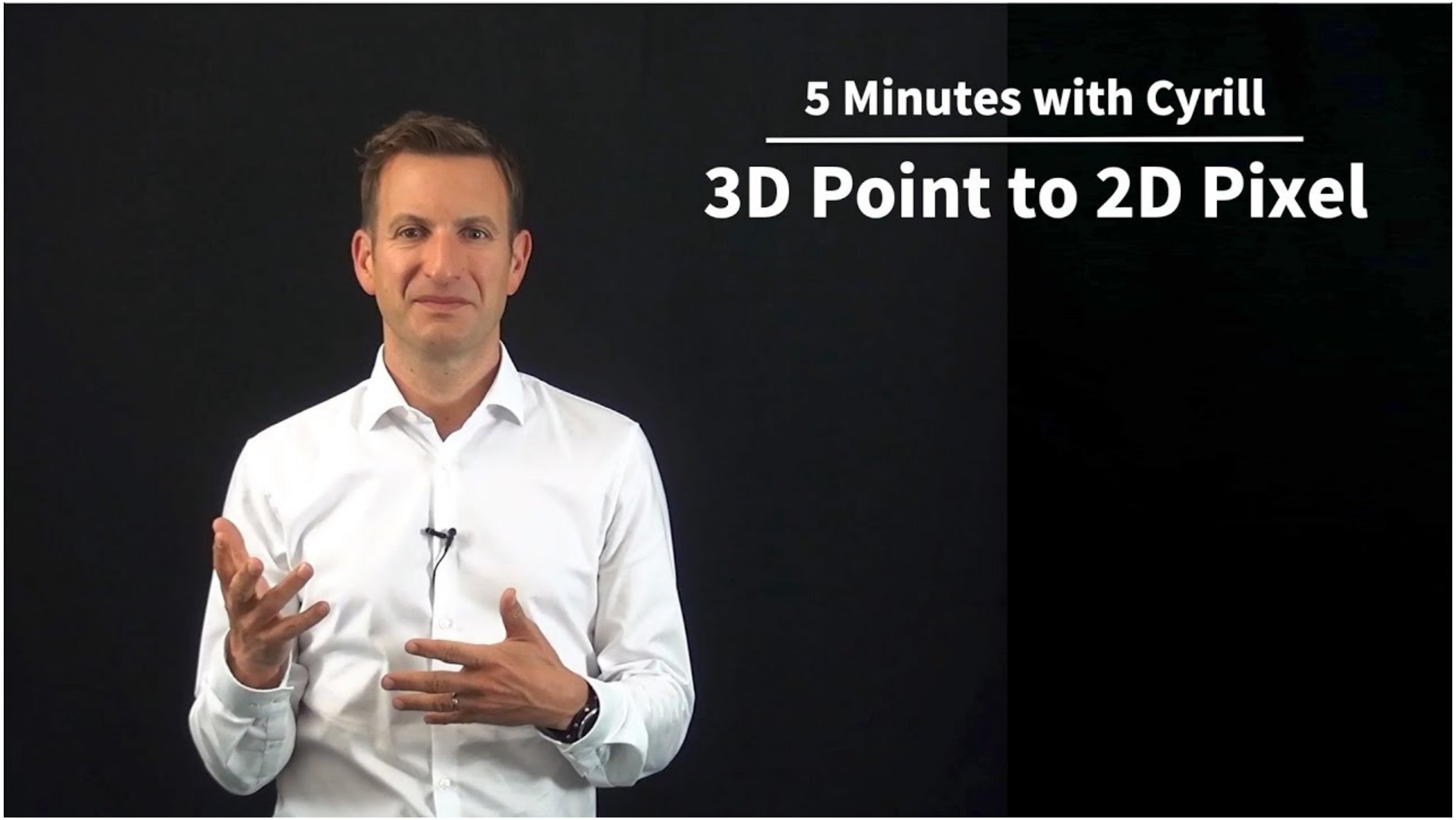
Photogrammetry & Robotics Lab

Camera Parameters: Extrinsics and Intrinsics

Cyrill Stachniss

The slides have been created by Cyrill Stachniss.

5 Minute Preparation for Today



https://www.ipb.uni-bonn.de/5min/

5 Minute Preparation for Today

https://www.ipb.uni-bonn.de/5min/

Goal: Describe How a Point is Mapped to a Pixel Coordinate

$$\mathbf{x} = \mathrm{P}\mathbf{X}$$

pixel coordinate | trans-formation | world coordinate

Goal: Describe How a 3D Point is Mapped to a 2D Pixel Coord.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathsf{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

2D pixel coordinate | trans-formation | 3D world coordinate

Coordinate Systems

1. World/object coordinate system

2. Camera coordinate system

3. Image plane coordinate system

4. Sensor coordinate system

Coordinate Systems

1. World/object coordinate system S_o
   written as: $[X, Y, Z]^\mathsf{T}$

   no index means object system

2. Camera coordinate system S_k
   written as: $[\,{}^kX,\ {}^kY,\ {}^kZ]^\mathsf{T}$
3. Image plane coordinate system S_c
   written as: $[\,{}^cx,\ {}^cy]^\mathsf{T}$
4. Sensor coordinate system S_s
   written as: $[\,{}^sx,\ {}^sy]^\mathsf{T}$

Transformation

We want to compute the mapping

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

in the sensor system	image plane to sensor	camera to image	object to camera	in the object system

Example

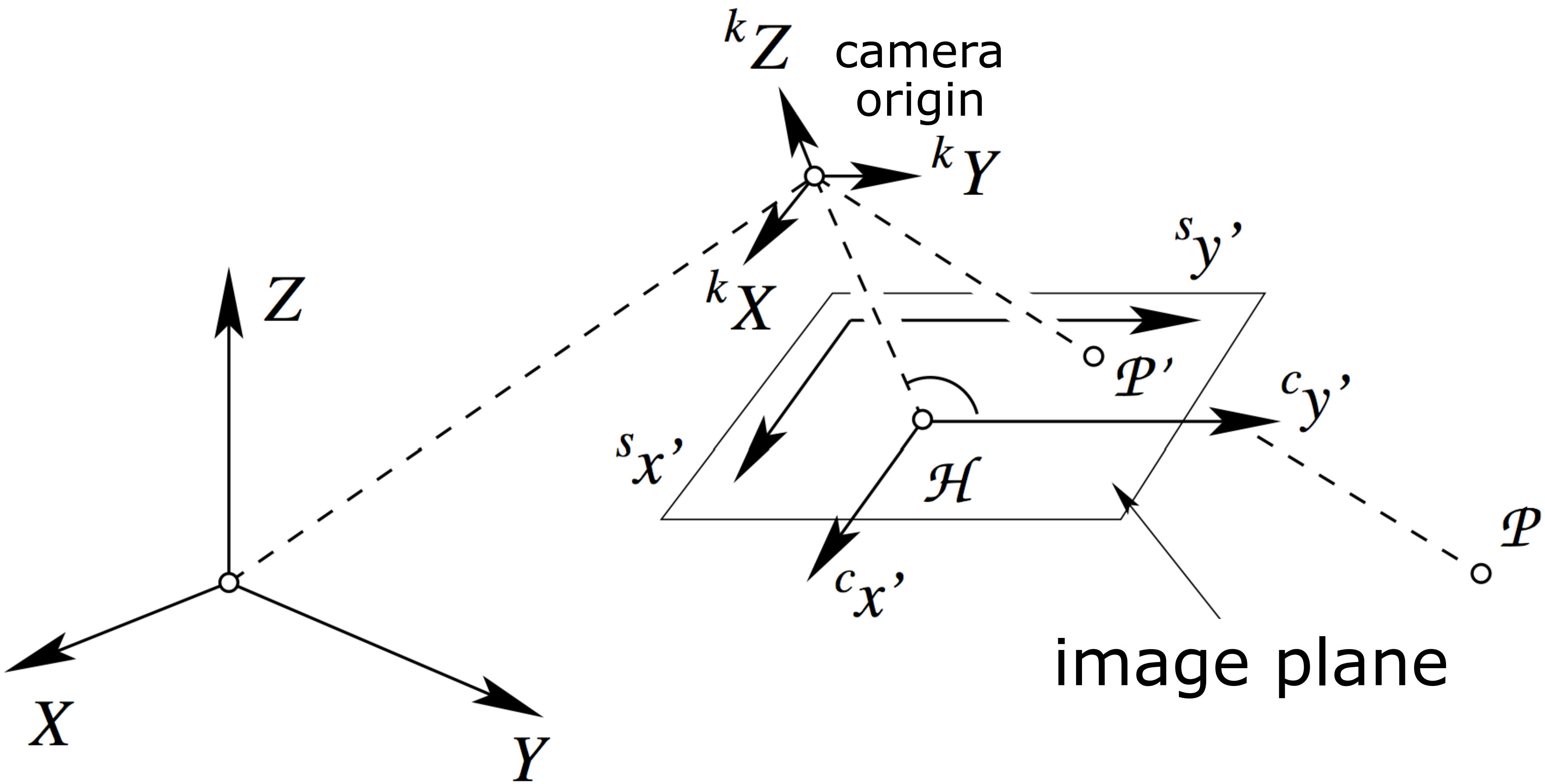
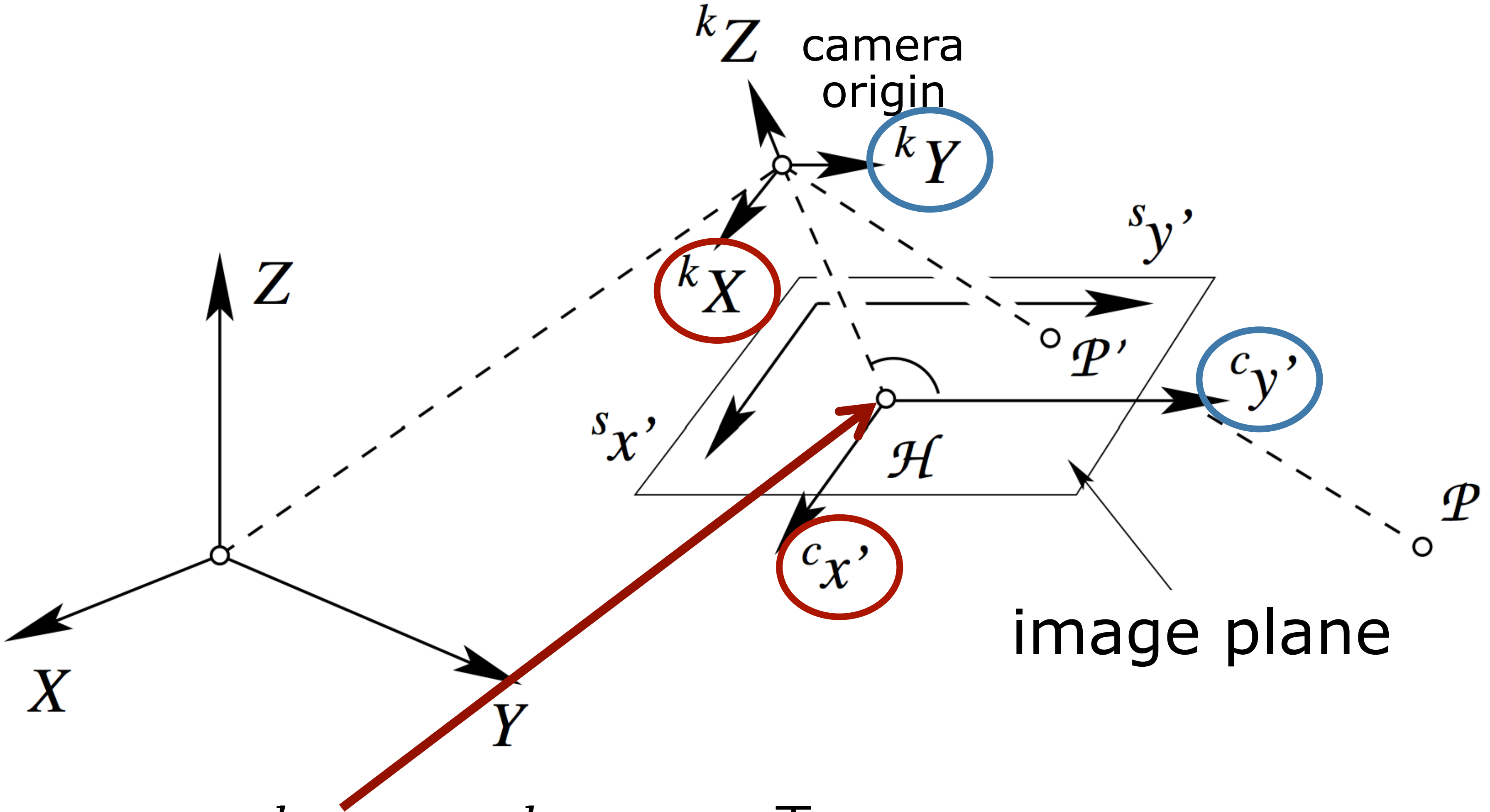


Image courtesy: Förstner

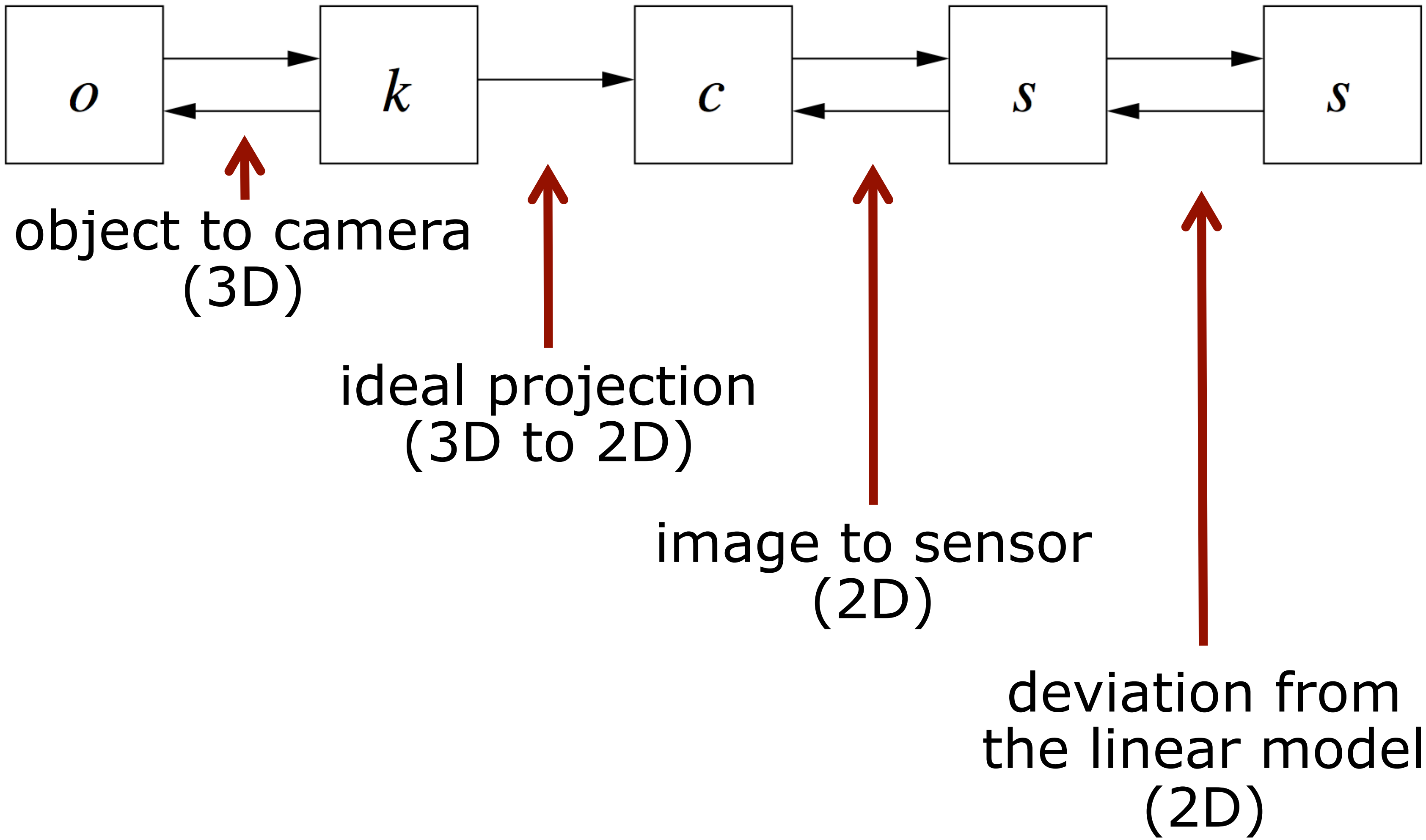
Example

The directions of the x-and y-axes in the c.s. k and c are identical. The origin of the c.s. c expressed in k is (0, 0, c)

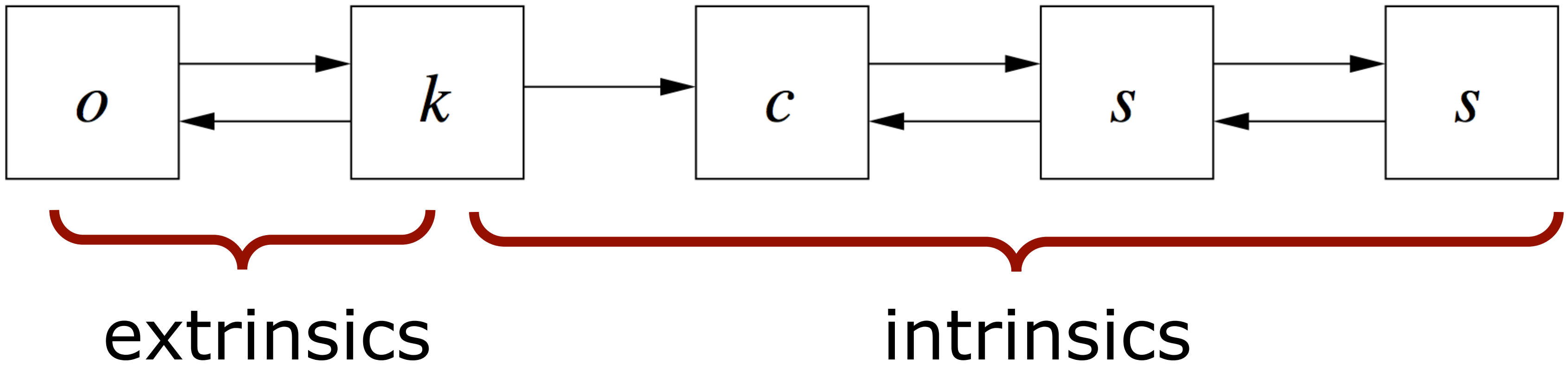


$$ {}^{k}O_c = {}^{k}[0, 0, c]^{\mathsf{T}} \text{ (with } c < 0\text{)} $$

From the World to the Sensor



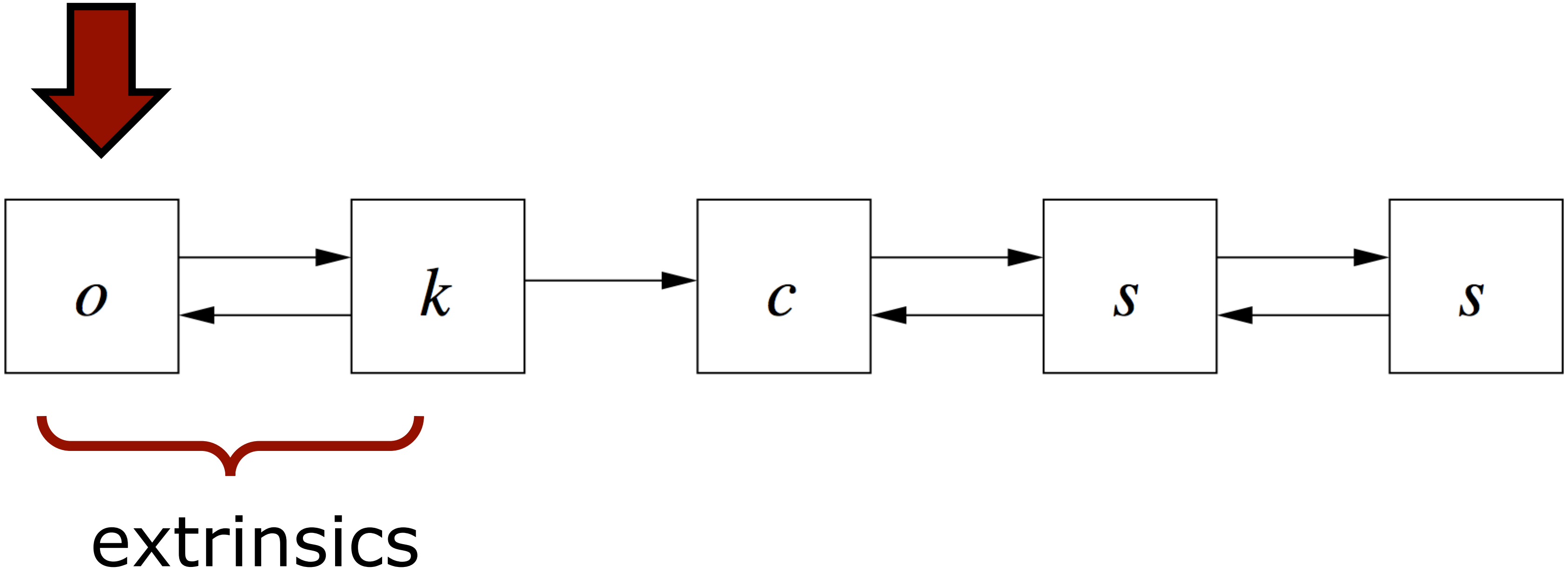
Extrinsic & Intrinsic Parameters



- 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

Where Are We in the Process?



Extrinsic Parameters

- Describe the pose (pose = position and heading) of the camera with respect to the world
- Invertible transformation

How many parameters are needed?

6 parameters: 3 for the position +
3 for the heading

Extrinsic Parameters

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

$$\boldsymbol{X}_{\mathcal{P}} = [X_{\mathcal{P}}, Y_{\mathcal{P}}, Z_{\mathcal{P}}]^{\mathsf{T}}$$

- Center O of the projection (origin of the camera coordinate system)

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

- $\boldsymbol{X}_O$ is sometimes also called $\boldsymbol{Z}$ or $\boldsymbol{Z}_O$

Transformation

- **Translation** between the origin of the world c.s. and the camera c.s.

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

- **Rotation** R from S_o to S_k.
- In Euclidian coordinates this yields

$${}^k\boldsymbol{X}_\mathcal{P} = R(\boldsymbol{X}_\mathcal{P} - \boldsymbol{X}_O)$$

Transformation in H.C.

- In Euclidian coordinates ${}^k\boldsymbol{X}_{\mathcal{P}} = R(\boldsymbol{X}_{\mathcal{P}} - \boldsymbol{X}_O)$
- Expressed in Homogeneous Coord.

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

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

Euclidian

H.C.

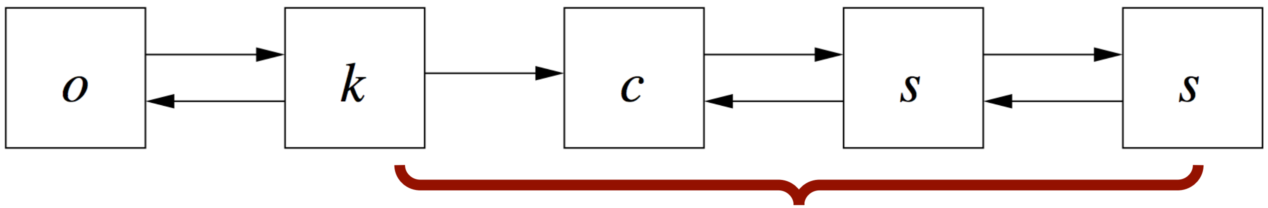
- or written in short as

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

Intrinsic Parameters

Intrinsic Parameters

- The process of projecting points from the camera c.s. to the sensor c.s.
- Invertible transformations:
  - image plane to sensor
  - model deviations
- Not invertible: central projection



Mapping as a 3 Step Process

We split up the mapping into 3 steps

1. Ideal perspective projection to the image plane
2. Mapping to the sensor coordinate system ("where the pixels are")
3. Compensation for the fact that the two previous mappings are idealized

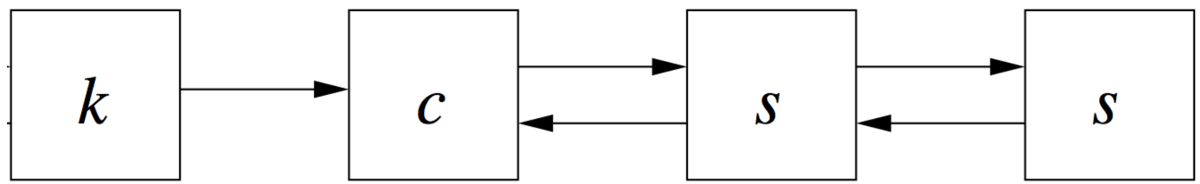
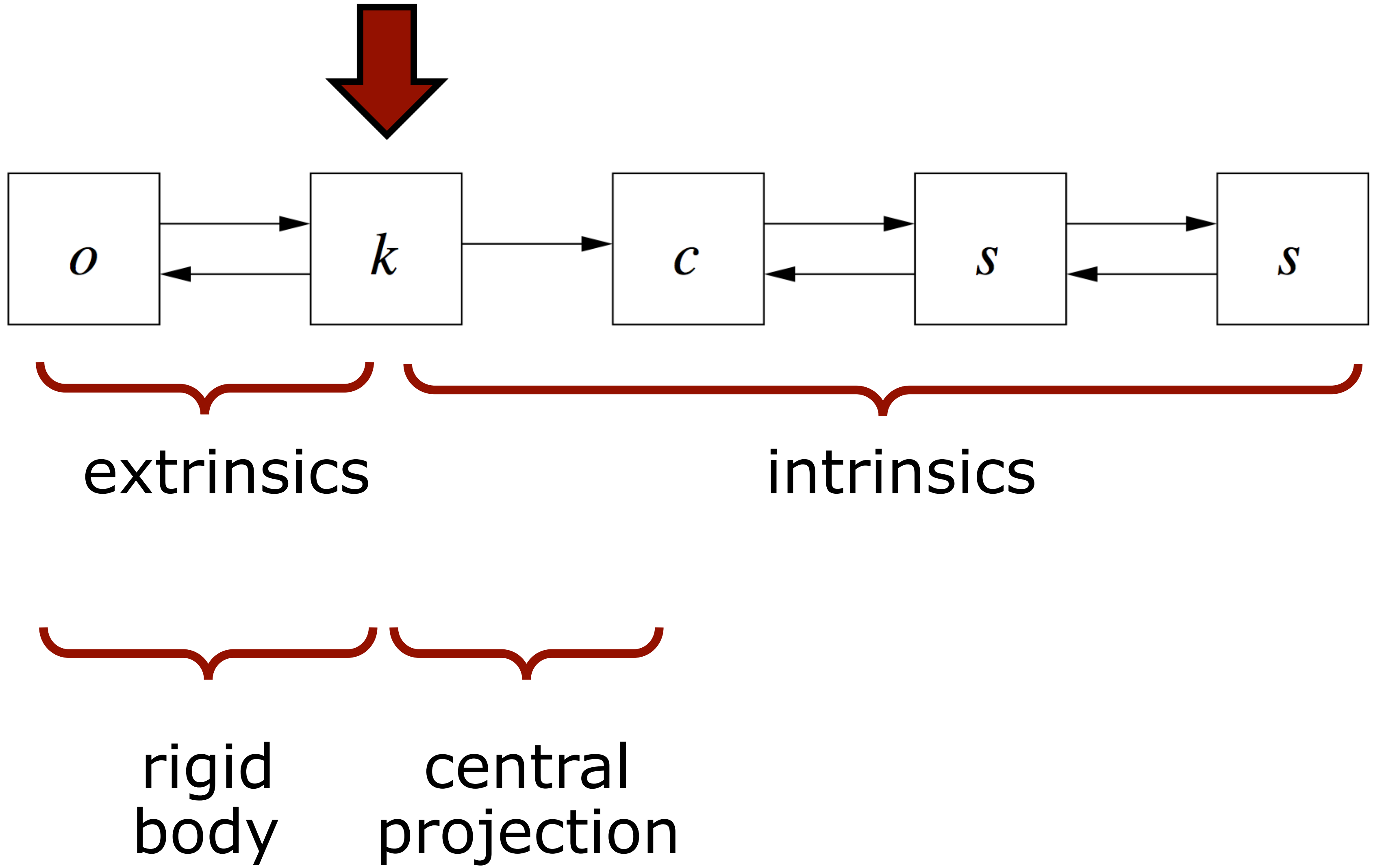


Image courtesy: Förstner

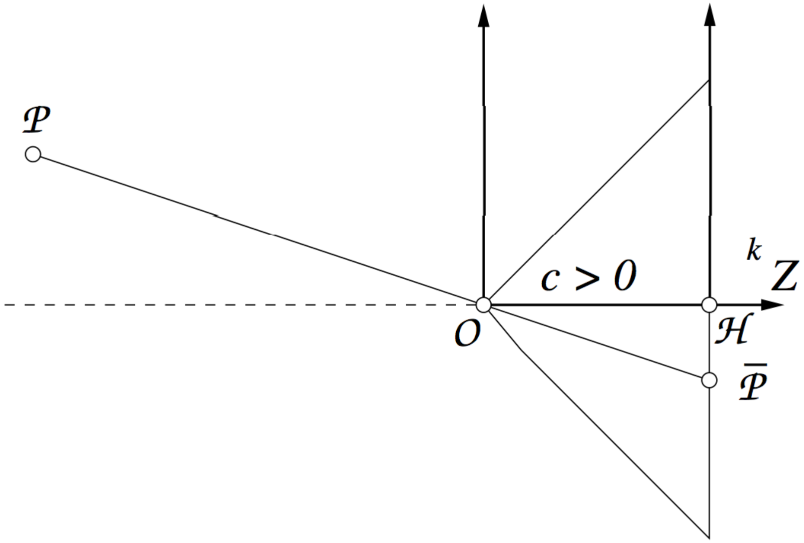
Where Are We in the Process?

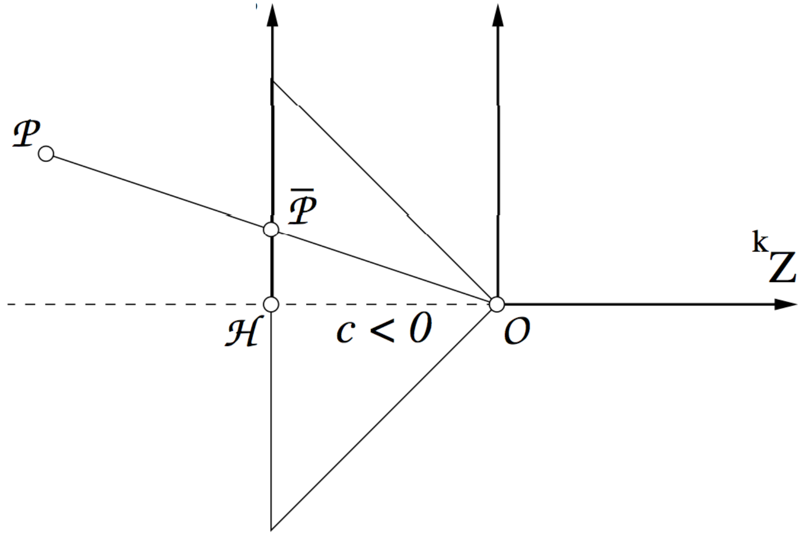


Ideal Perspective Projection

- Distortion-free lens
- All rays are straight lines and pass through the projection center. This point is the origin of the camera coordinate system S_k
- Focal point and principal point lie on the optical axis
- The distance from the camera origin to the image plane is the constant c

Image Coordinate System





Physically motivated coordinate system: c>0

Most popular image coordinate system: c<0

rotation by 180 deg

Image courtesy: Förstner

Camera Constant

- Distance between the center of projection $\mathcal{O}$ and the principal point $\mathcal{H}$
- Value is computed as part of the camera calibration
- **Here: coordinate system with** $c < 0$

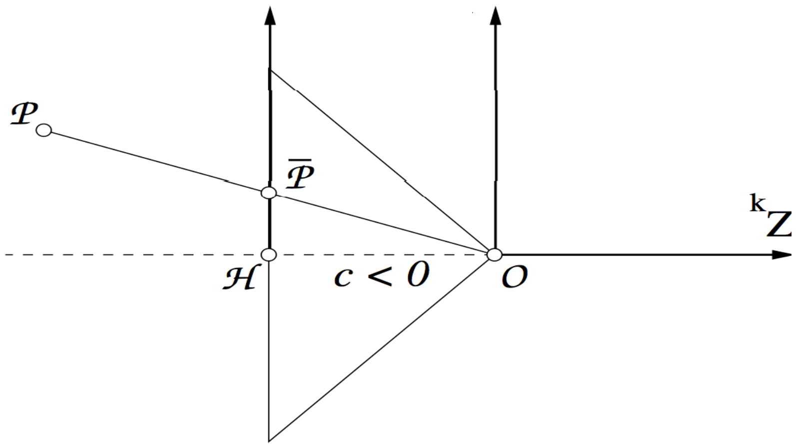
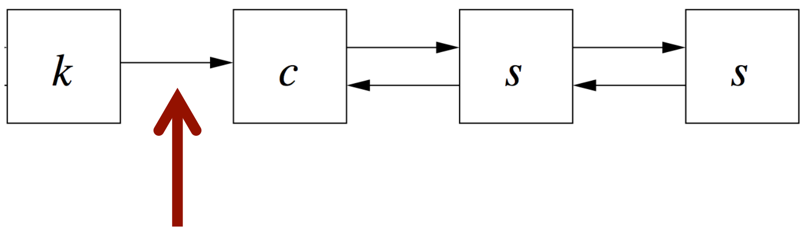
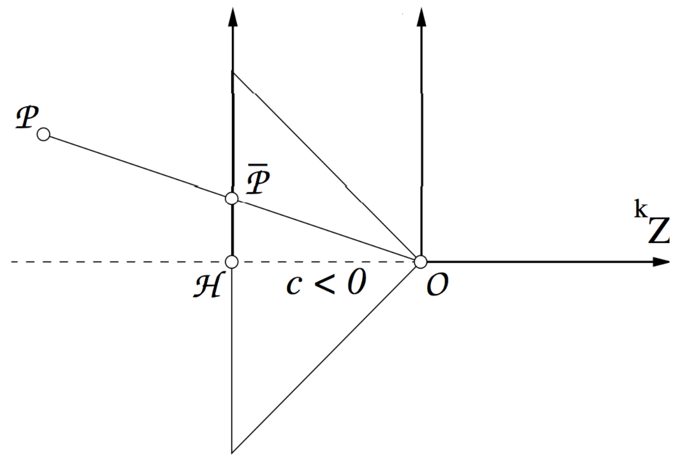


Image courtesy: Förstner

Ideal Perspective Projection

Through the intercept theorem, we obtain for the point $\overline{\mathcal{P}}$ projected onto the image plane the coordinates $[\,{}^{c}x_{\overline{\mathcal{P}}},\ {}^{c}y_{\overline{\mathcal{P}}}]$

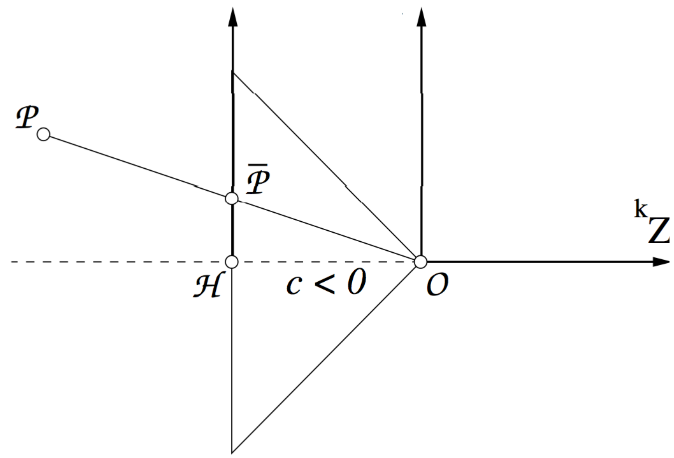




Ideal Perspective Projection

Through the intercept theorem, we obtain for the point $\overline{\mathcal{P}}$ projected onto the image plane the coordinates $[\,{}^{c}x_{\overline{\mathcal{P}}},\ {}^{c}y_{\overline{\mathcal{P}}}]$

$$
\begin{aligned}
{}^{c}x_{\overline{\mathcal{P}}} := {}^{k}X_{\overline{\mathcal{P}}} &= c\,\frac{{}^{k}X_{\mathcal{P}}}{{}^{k}Z_{\mathcal{P}}} \\
{}^{c}y_{\overline{\mathcal{P}}} := {}^{k}Y_{\overline{\mathcal{P}}} &= c\,\frac{{}^{k}Y_{\mathcal{P}}}{{}^{k}Z_{\mathcal{P}}} \\
\left(c = {}^{k}Z_{\overline{\mathcal{P}}} \right. &= \left. c\,\frac{{}^{k}Z_{\mathcal{P}}}{{}^{k}Z_{\mathcal{P}}}\right)
\end{aligned}
$$



In Homogenous Coordinates

- We can express that in H.C.

$$\begin{bmatrix} {}^kU_{\overline{\mathcal{P}}} \\ {}^kV_{\overline{\mathcal{P}}} \\ {}^kW_{\overline{\mathcal{P}}} \\ {}^kT_{\overline{\mathcal{P}}} \end{bmatrix} = \begin{bmatrix} c & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^kX_{\mathcal{P}} \\ {}^kY_{\mathcal{P}} \\ {}^kZ_{\mathcal{P}} \\ 1 \end{bmatrix}$$

- and drop the 3rd coordinate (row)

$${}^c\mathbf{x}_{\overline{\mathcal{P}}} = \begin{bmatrix} {}^cu_{\overline{\mathcal{P}}} \\ {}^cv_{\overline{\mathcal{P}}} \\ {}^cw_{\overline{\mathcal{P}}} \end{bmatrix} = \begin{bmatrix} c & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^kX_{\mathcal{P}} \\ {}^kY_{\mathcal{P}} \\ {}^kZ_{\mathcal{P}} \\ 1 \end{bmatrix}$$

Verify the Result

- Ideal perspective projection is

$$ {}^{c}x_{\overline{\mathcal{P}}} = c\frac{{}^{k}X_{\mathcal{P}}}{{}^{k}Z_{\mathcal{P}}} \qquad {}^{c}y_{\overline{\mathcal{P}}} = c\frac{{}^{k}Y_{\mathcal{P}}}{{}^{k}Z_{\mathcal{P}}} $$

- Our results is

$$ \begin{bmatrix} {}^{c}x_{\overline{\mathcal{P}}} \\ {}^{c}y_{\overline{\mathcal{P}}} \\ 1 \end{bmatrix} = \begin{bmatrix} c & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^{k}X_{\mathcal{P}} \\ {}^{k}Y_{\mathcal{P}} \\ {}^{k}Z_{\mathcal{P}} \\ 1 \end{bmatrix} $$

$$ = \begin{bmatrix} c\;{}^{k}X_{\mathcal{P}} \\ c\;{}^{k}Y_{\mathcal{P}} \\ {}^{k}Z_{\mathcal{P}} \end{bmatrix} = \begin{bmatrix} c\,\frac{{}^{k}X_{\mathcal{P}}}{{}^{k}Z_{\mathcal{P}}} \\ c\,\frac{{}^{k}Y_{\mathcal{P}}}{{}^{k}Z_{\mathcal{P}}} \\ 1 \end{bmatrix} $$

In Homogenous Coordinates

- Thus, we can write for any point

$$ {}^c\mathbf{x}_{\overline{\mathcal{P}}} = {}^c\mathsf{P}_k \, {}^k\mathbf{X}_{\mathcal{P}} $$

- with

$$ {}^c\mathsf{P}_k = \begin{bmatrix} c & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} $$

Assuming an Ideal Camera…

…leads us to the mapping using the intrinsic and extrinsic parameters

$$ {}^{c}\mathbf{x} = {}^{c}\mathsf{P}\,\mathbf{X} $$

with

$$ {}^{c}\mathsf{P} = {}^{c}\mathsf{P}_{k}\,{}^{k}\mathsf{H} = \begin{bmatrix} c & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & -R\boldsymbol{X}_{O} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} $$

Calibration Matrix

- We can now define the **calibration matrix for the ideal camera**

$$ {}^{c}\mathsf{K} = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

- We can write the overall mapping as

$$ {}^{c}\mathsf{P} = {}^{c}\mathsf{K}[R| - R\boldsymbol{X}_O] = {}^{c}\mathsf{K}\, R\, [I_3| - \boldsymbol{X}_O] $$

3x4 matrices

Notation

We can write the overall mapping as

$$ {}^c\mathsf{P} = {}^c\mathsf{K}[R| - R\boldsymbol{X}_O] = {}^c\mathsf{K}\, R\, [I_3| - \boldsymbol{X}_O] $$

short for

$$ [I_3| - \boldsymbol{X}_O] = \begin{bmatrix} 1 & 0 & 0 & -X_O \\ 0 & 1 & 0 & -Y_O \\ 0 & 0 & 1 & -Z_O \end{bmatrix} $$

Calibration Matrix

$$ {}^c\mathsf{K} = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

- We have the projection

$$ {}^c\mathsf{P} = {}^c\mathsf{K}\ R\ [I_3 | -\boldsymbol{X}_O] $$

- that maps a point to the image plane

$$ {}^c\mathbf{x} = {}^c\mathsf{K} R [I_3 | -\boldsymbol{X}_O] \mathbf{X} $$

- and yields for the coordinates of ${}^c\mathbf{x}$

$$ \begin{bmatrix} {}^c u' \\ {}^c v' \\ {}^c w' \end{bmatrix} = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X - X_O \\ Y - Y_O \\ Z - Z_O \end{bmatrix} $$

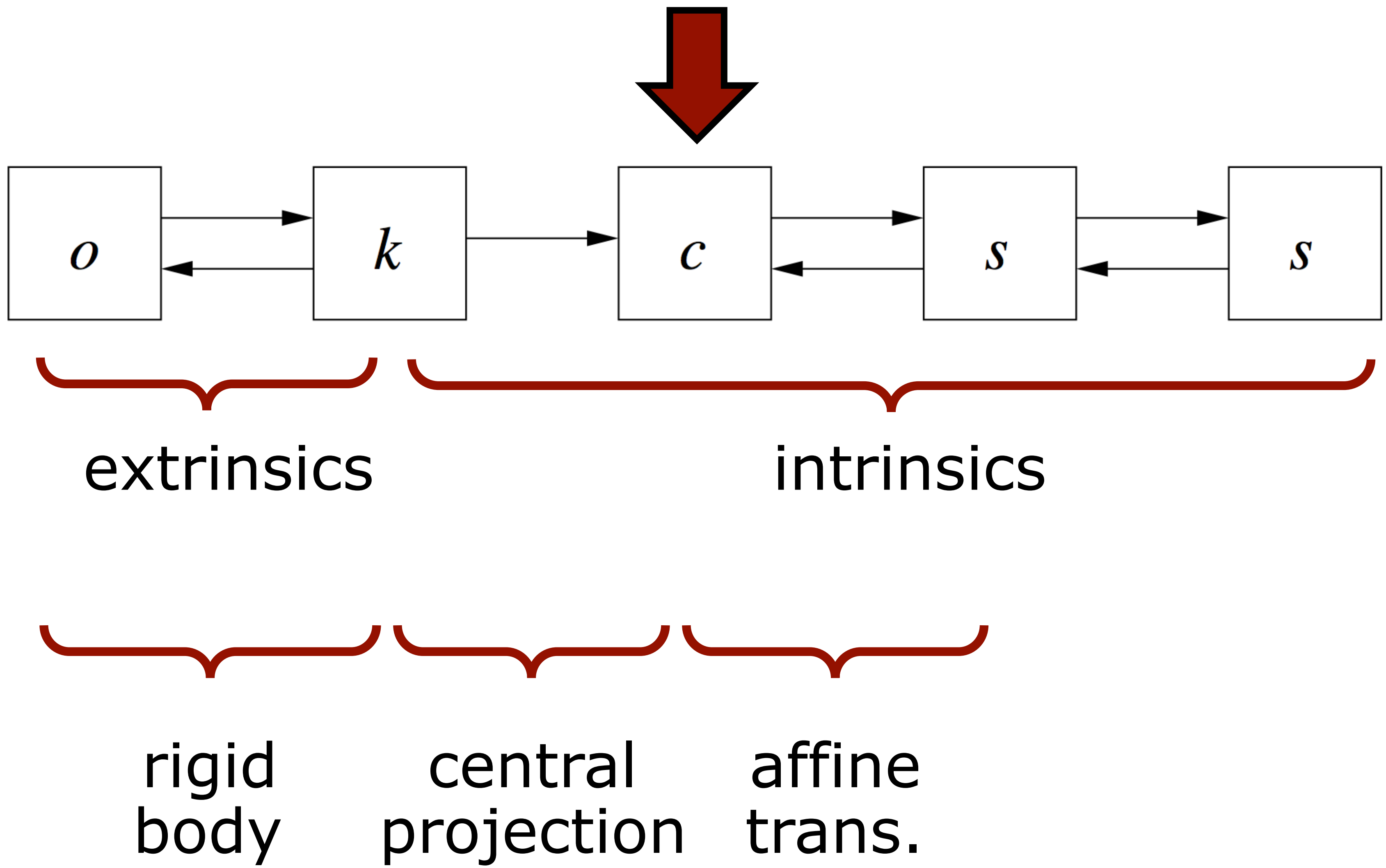
In Euclidian Coordinates

- This leads to the so-called collinearity equation for the image coordinates

$$
\begin{aligned}
{}^c x &= c \frac{r_{11}(X - X_O) + r_{12}(Y - Y_O) + r_{13}(Z - Z_O)}{r_{31}(X - X_O) + r_{32}(Y - Y_O) + r_{33}(Z - Z_O)} \\
{}^c y &= c \frac{r_{21}(X - X_O) + r_{22}(Y - Y_O) + r_{23}(Z - Z_O)}{r_{31}(X - X_O) + r_{32}(Y - Y_O) + r_{33}(Z - Z_O)}
\end{aligned}
$$

Mapping to the Sensor (wihtout non-linear errors)

Where Are We in the Process?



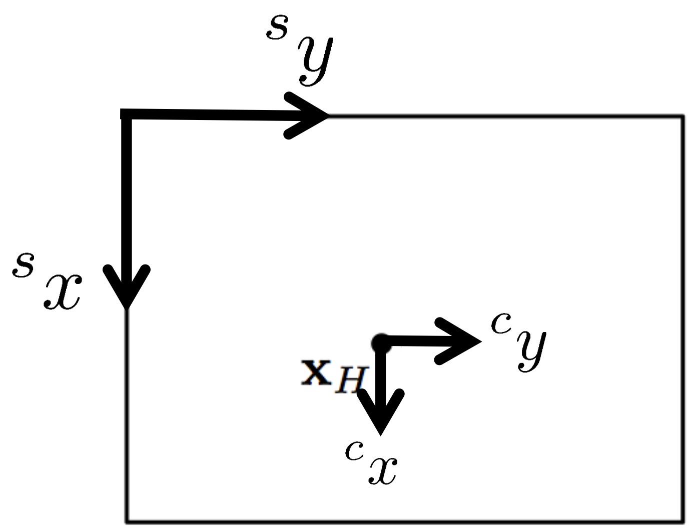
Linear Errors

- The next step is the mapping from the image to the sensor
- Location of the principal point in the image
- Scale difference in x and y based on the chip design
- Shear compensation

Location of the Principal Point

- The origin of the sensor system is not at the principal point
- Compensation through a shift

$$ {}^s\mathsf{H}_c = \begin{bmatrix} 1 & 0 & x_H \\ 0 & 1 & y_H \\ 0 & 0 & 1 \end{bmatrix} $$



Shear and Scale Difference

- Scale difference m in x and y
- Shear compensation s (for digital cameras, we typically have $s \approx 0$)

$$ {}^{s}\mathsf{H}_c = \begin{bmatrix} 1 & s & x_H \\ 0 & 1+m & y_H \\ 0 & 0 & 1 \end{bmatrix} $$

- Finally, we obtain

$$ {}^{s}\mathbf{x} = {}^{s}\mathsf{H}_c \, {}^{c}\mathsf{K}R[I_3 | -\boldsymbol{X}_O]\mathbf{X} $$

Calibration Matrix

Often, the transformation ${}^s\mathsf{H}_c$ is combined with the calibration matrix ${}^c\mathsf{K}$, i.e.

$$
\begin{aligned}
\mathsf{K} &\doteq {}^s\mathsf{H}_c\,{}^c\mathsf{K} \\
&= \begin{bmatrix} 1 & s & x_H \\ 0 & 1+m & y_H \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c & cs & x_H \\ 0 & c(1+m) & y_H \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

Calibration Matrix

- This calibration matrix is an **affine** transformation

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

- contains 5 parameters:
  - camera constant: c
  - principal point: x_H, y_H
  - scale difference: m
  - shear: s

DLT: Direct Linear Transform

- The mapping $\mathcal{x} = \mathcal{P}(\mathcal{X}):\quad \mathbf{x} = \mathsf{P}\mathbf{X}$
- with $\mathsf{P} = \mathsf{K}R[I_3 | -\boldsymbol{X}_O]$

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

- is called the **direct linear transform**
- It is the model of the **affine camera**
- **Affine camera** = camera with an affine mapping to the sensor c.s. (after the central projection is applied)

DLT: Direct Linear Transform

- The homogeneous projection matrix

$$\mathsf{P} = \mathsf{K}R[\boldsymbol{I}_3 | -\boldsymbol{X}_O]$$

- contains **11 parameters**
  - 6 extrinsic parameters: $R, \boldsymbol{X}_O$
  - 5 intrinsic parameters: c, x_H, y_H, m, s

DLT: Direct Linear Transform

- The homogeneous projection matrix

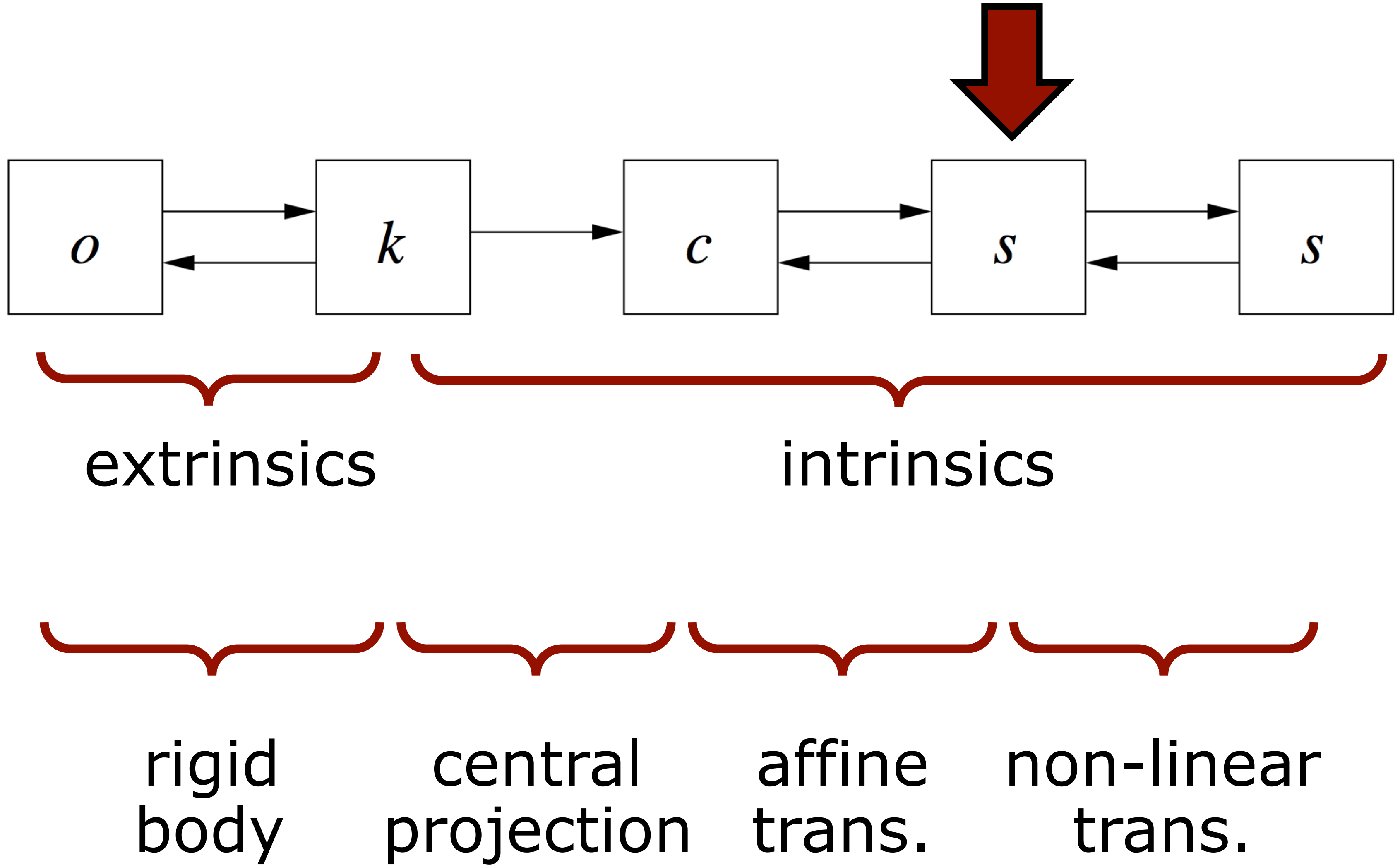
$$\mathsf{P} = \mathsf{K}R[\boldsymbol{I}_3 | -\boldsymbol{X}_O]$$

- contains **11 parameters**
  - 6 extrinsic parameters: $R, \boldsymbol{X}_O$
  - 5 intrinsic parameters: c, x_H, y_H, m, s
- Euclidian world:

$$\begin{aligned} {}^s x &= \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}} \\ {}^s y &= \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}} \end{aligned}$$

Non-Linear Errors

Where Are We in the Process?

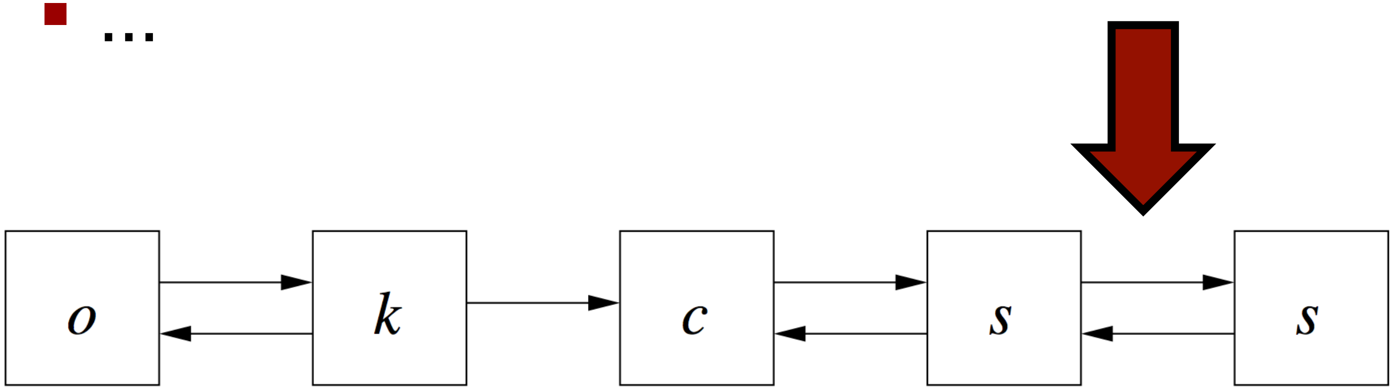


Non-Linear Errors

- So far, we considered only **linear** errors (DLT)
- The real world is **non-linear**
- Reasons for non-linear errors

Non-Linear Errors

- So far, we considered only **linear** errors (DLT)
- The real world is **non-linear**
- Reasons for non-linear errors
  - Imperfect lens
  - Planarity of the sensor
  - …



General Mapping

- Idea: add a last step that covers the non-linear effects
- **Location-dependent** shift in the sensor coordinate system
- Individual shift for each pixel
- General mapping

$$
\begin{aligned}
{}^{a}x &= {}^{s}x + \Delta x(\boldsymbol{x}, \boldsymbol{q}) \\
{}^{a}y &= {}^{s}y + \Delta y(\boldsymbol{x}, \boldsymbol{q})
\end{aligned}
$$

in the image plane

Example

Left:not straight line preserving

Right: rectified image

Image courtesy: Abraham

General Mapping in H.C.

- General mapping yields

$$ {}^{a}\mathbf{x} = {}^{a}\mathsf{H}_{s}(\boldsymbol{x}) \; {}^{s}\mathbf{x} $$

- with

$$ {}^{a}\mathsf{H}_{s}(\boldsymbol{x}) = \begin{bmatrix} 1 & 0 & \Delta x(\boldsymbol{x}, \boldsymbol{q}) \\ 0 & 1 & \Delta y(\boldsymbol{x}, \boldsymbol{q}) \\ 0 & 0 & 1 \end{bmatrix} $$

- so that the overall mapping becomes

$$ {}^{a}\mathbf{x} = {}^{a}\mathsf{H}_{s}(\boldsymbol{x}) \, \mathsf{K} R [I_3 | - \boldsymbol{X}_O] \mathbf{X} $$

General Calibration Matrix

- General calibration matrix is obtained by combining the one of the affine camera with the general mapping

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

- resulting in the general camera model

$$ {}^a\mathbf{x} = {}^a\mathsf{P}(\boldsymbol{x},\boldsymbol{q})\,\mathbf{X} $$

$$ {}^a\mathsf{P}(\boldsymbol{x},\boldsymbol{q}) = {}^a\mathsf{K}(\boldsymbol{x},\boldsymbol{q})\,R[I| - \boldsymbol{X}_O] $$

Approaches for Modeling ${}^{a}\mathrm{H}_s(\boldsymbol{x})$

Large number of different approaches to model the non-linear errors

Physics approach

- Well motivated
- There are large number of reasons for non-linear errors …

Phenomenological approaches

- Just model the effects
- Easier but do not identify the problem

Example: Barrel Distortion

- A standard approach for wide angle lenses is to model the barrel distortion

$$
\begin{aligned}
{}^{a}x &= x(1 + q_1 r^2 + q_2 r^4) \\
{}^{a}y &= y(1 + q_1 r^2 + q_2 r^4)
\end{aligned}
$$

- with $[x, y]^{\mathsf{T}}$ being point as projected by an ideal pin-hole camera
- with r being the distance **of the pixel** in the image to the principal point
- The terms q_1, q_2 are the additional parameters of the general mapping

Radial Distortion Example

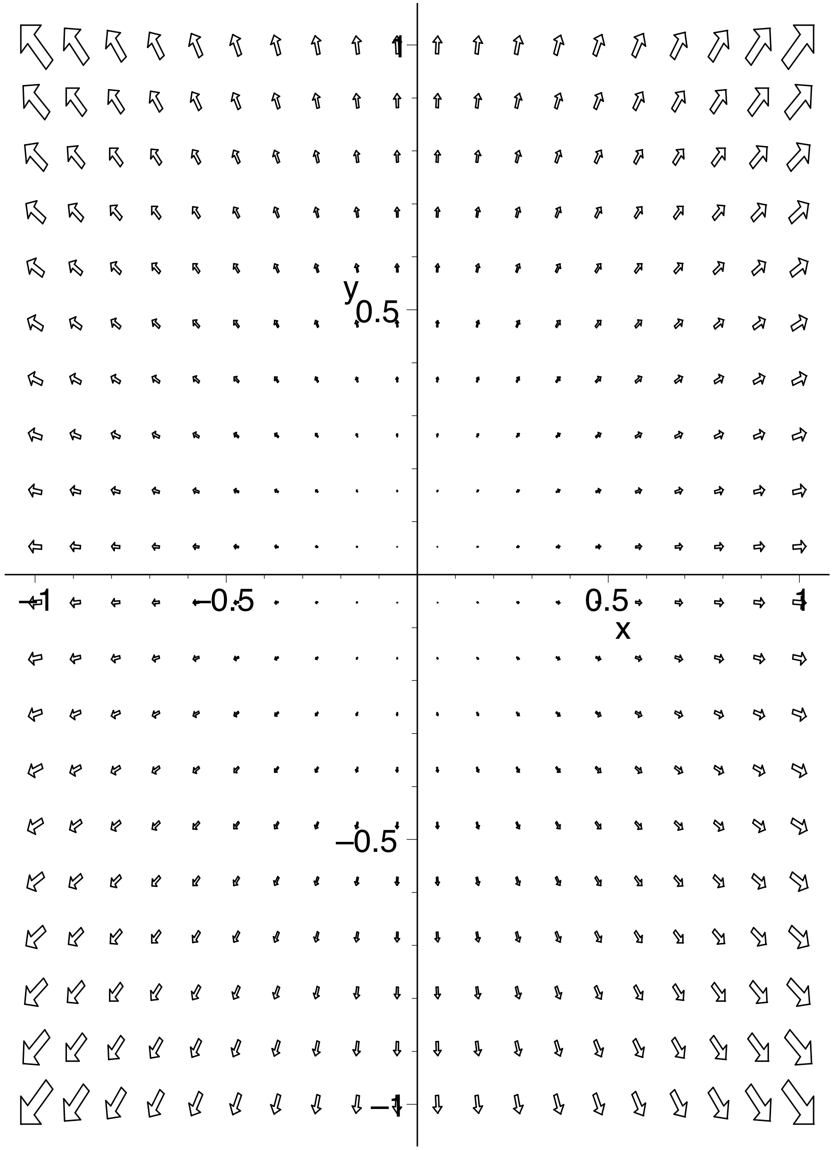


Image courtesy: Förstner

Mapping as a Two Step Process

1. Projection of the affine camera

$$ {}^{s}\mathbf{x} = \mathsf{P}\mathbf{X} $$

2. Consideration of non-linear effects

$$ {}^{a}\mathbf{x} = {}^{a}\mathsf{H}_{s}(\boldsymbol{x})\ {}^{s}\mathbf{x} $$

Individual mapping for each point!

What to Do If We Want to Get Information About the Scene?

Inversion of the Mapping

- **Goal:** map from ${}^{a}\mathbf{x}$ back to $\mathbf{X}$
- 1st step: ${}^{a}\mathbf{x} \rightarrow {}^{s}\mathbf{x}$
- 2nd step: ${}^{s}\mathbf{x} \rightarrow \mathbf{X}$

Inversion of the Mapping

- Goal: map from ${}^{a}\mathbf{x}$ back to $\mathbf{X}$
- **1st step:** ${}^{a}\mathbf{x} \to {}^{s}\mathbf{x}$
- 2nd step: ${}^{s}\mathbf{x} \to \mathbf{X}$

${}^{a}\mathbf{x} \to {}^{s}\mathbf{x}$

- The general nature of ${}^{a}\mathsf{H}_{s}(\boldsymbol{x})$ in ${}^{a}\mathbf{x} = {}^{a}\mathsf{H}_{s}(\boldsymbol{x})\, {}^{s}\mathbf{x}$ requires an iterative solution

depends on the coordinate of the point to transform

Inversion Step 1: ${}^{a}\mathbf{x} \rightarrow {}^{s}\mathbf{x}$

- Iteration due to unknown $\boldsymbol{x}$ in ${}^{a}\mathsf{H}_s(\boldsymbol{x})$
- Start with ${}^{a}\mathbf{x}$ as the initial guess

$$\mathbf{x}^{(1)} = [\,{}^{a}\mathsf{H}_s(\,{}^{a}\mathbf{x})]^{-1}\ {}^{a}\mathbf{x}$$

often w.r.t. the principal point

- and iterate

$$\mathbf{x}^{(\nu+1)} = \left[\,{}^{a}\mathsf{H}_s(\mathbf{x}^{(\nu)})\right]^{-1}\ {}^{a}\mathbf{x}$$

- As ${}^{a}\mathbf{x}$ is often a good initial guess, this procedure converges quickly

Inversion Step 2: ${}^s\mathbf{x} \to \mathbf{X}$

- The next step is the **inversion of the projective mapping**
- We cannot reconstruct the 3D point but the ray though the 3D point
- With the known matrix P, we can write

$$
\begin{aligned}
\lambda \mathbf{x} &= \mathsf{P}\mathbf{X} = \mathsf{K}R[I_3| - \boldsymbol{X}_O]\mathbf{X} \\
&= [\mathsf{K}R| - \mathsf{K}R\boldsymbol{X}_O] \begin{bmatrix} \boldsymbol{X} \\ 1 \end{bmatrix} \\
&= \mathsf{K}R\boldsymbol{X} - \mathsf{K}R\boldsymbol{X}_O
\end{aligned}
$$

factor resulting from the H.C.

Inversion Step 2: ${}^s\mathbf{x} \rightarrow \mathbf{X}$

- Starting from $\lambda \mathbf{x} = \mathsf{K}R\boldsymbol{X} - \mathsf{K}R\boldsymbol{X}_O$
- we obtain

$$
\begin{aligned}
\boldsymbol{X} &= (\mathsf{K}R)^{-1}\mathsf{K}R\boldsymbol{X}_O + \lambda(\mathsf{K}R)^{-1}\mathbf{x} \\
&= \boldsymbol{X}_O + \lambda(\mathsf{K}R)^{-1}\mathbf{x}
\end{aligned}
$$

- The term $\lambda(\mathsf{K}R)^{-1}\mathbf{x}$ describes the direction of the ray from the camera origin $\boldsymbol{X}_O$ to the 3D point $\boldsymbol{X}$

Classification of Cameras

extrinsic parameters		intrinsic parameters			
$\boldsymbol{X}_0$ (X, Y, Z)	R (ω, ϕ, κ)	c	x_H, y_H	m, s	$q_1, q_2, \ldots$

Classification of Cameras

extrinsic parameters	
$\boldsymbol{X}_0$ (X, Y, Z)	
normalized	

Example: pinhole camera for which the principal point is the origin of the image coordinate system, the x- and y-axis of the image coordinate system is aligned with the x-/y-axis of the world c.s. and the distance between the origin and the image plane is 1

Classification of Cameras

extrinsic parameters

$\boldsymbol{X}_0$ (X, Y, Z)	R (ω, ϕ, κ)	
normalized		
unit camera		

Example: pinhole camera for which the principal point (x, y) is the origin of the image coordinate system and the distance between the origin and the image plane is 1

Classification of Cameras

extrinsic parameters		intrinsic parameters	
$\boldsymbol{X}_0$ (X, Y, Z)	R (ω, ϕ, κ)	c	
normalized			
unit camera			
ideal camera			

Example: pinhole camera for which the x/y coordinate of the principal point is the origin of the image coordinate system

Classification of Cameras

extrinsic parameters		intrinsic parameters		
$\boldsymbol{X}_0$ (X, Y, Z)	R (ω, ϕ, κ)	c	x_H, y_H	
normalized				
unit camera				
ideal camera				
Euclidian camera				

Example: pinhole camera using a Euclidian sensor in the image plane

Classification of Cameras

extrinsic parameters		intrinsic parameters			
$\boldsymbol{X}_0$ (X, Y, Z)	R (ω, ϕ, κ)	c	x_H, y_H	m, s	
normalized					
unit camera					
ideal camera					
Euclidian camera					
affine camera					

Example: camera that preserves straight lines

Classification of Cameras

extrinsic parameters		intrinsic parameters			
$\boldsymbol{X}_0$ (X, Y, Z)	R (ω, ϕ, κ)	c	x_H, y_H	m, s	$q_1, q_2, \ldots$
normalized					
unit camera					
ideal camera					
Euclidian camera					
affine camera					
general camera					

Example: camera with non-linear distortions

Calibration Matrices

camera	calibration matrix	#parameters
unit	$^0\mathsf{K} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	6 (6+0)
ideal	$^k\mathsf{K} = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 1 \end{bmatrix}$	7 (6+1)
Euclidian	$^p\mathsf{K} = \begin{bmatrix} c & 0 & x_H \\ 0 & c & y_H \\ 0 & 0 & 1 \end{bmatrix}$	9 (6+3)
affine	$\mathsf{K} = \begin{bmatrix} c & cs & x_H \\ 0 & c(1+m) & y_H \\ 0 & 0 & 1 \end{bmatrix}$	11 (6+5)
general	$^a\mathsf{K} = \begin{bmatrix} c & cs & x_H + \Delta x \\ 0 & c(1+m) & y_H + \Delta y \\ 0 & 0 & 1 \end{bmatrix}$	11+N

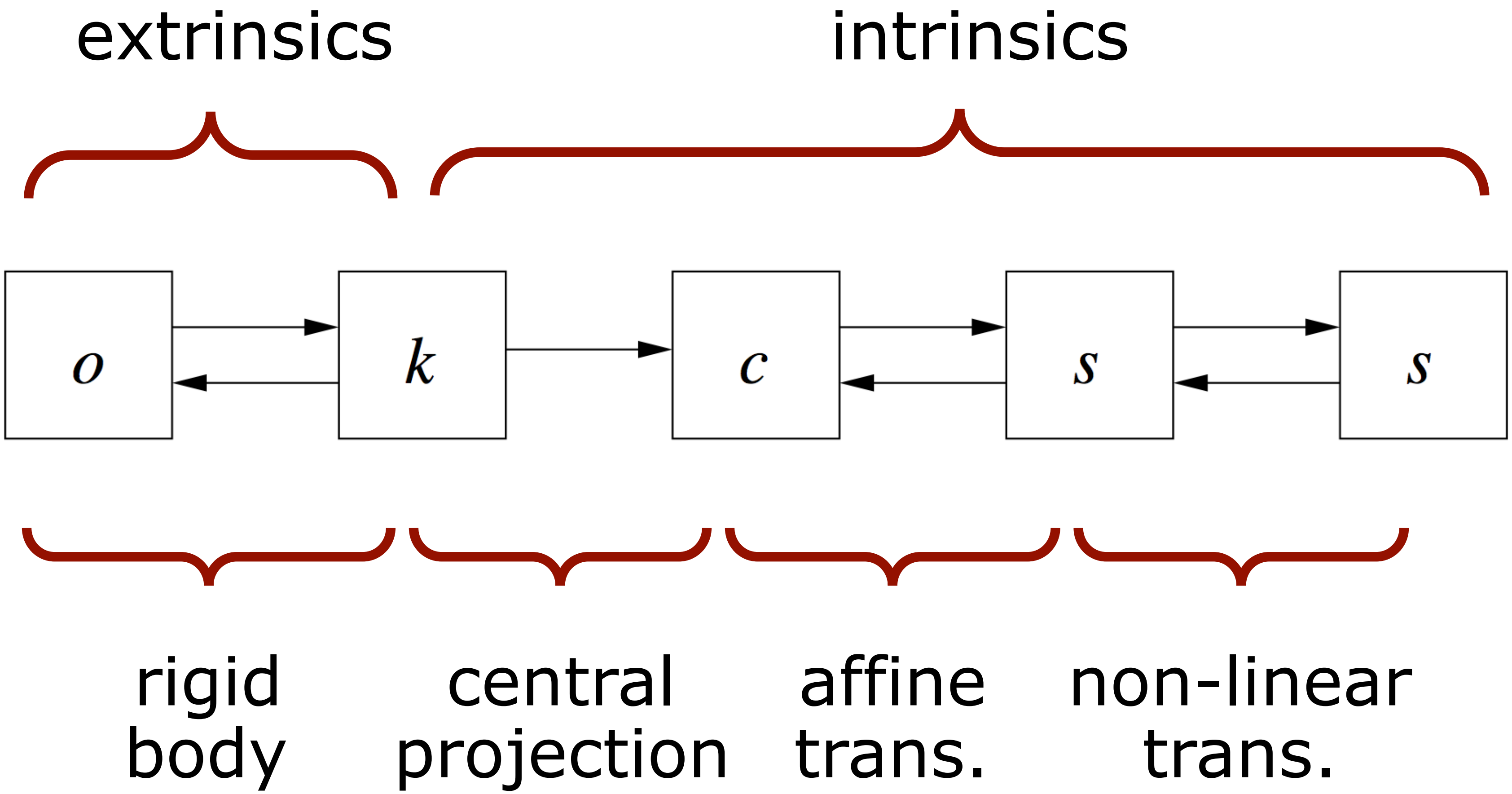
Calibrated Camera

- 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**
- If the intrinsics are known and do not change, the camera is called **metric camera**

Summary

- We described the mapping from the world c.s. to individual pixels (sensor)
- **Extrinsics** = world to camera c.s.
- **Intrinsics** = camera to sensor c.s.
- **DLT** = Direct linear transform
- Non-linear errors
- Inversion of the mapping process

Summary of the Mapping



Literature

- Förstner & Wrobel, Photogrammetric Computer Vision, Chapter "Geometry of the Single Image", 11.1.1 – 11.1.6
- Förstner, Scriptum Photogrammetrie I, Chapter "Einbild-Photogrammetrie", subsections 1 & 2

Slide Information

- The slides have been created by Cyrill Stachniss as part of the photogrammetry and robotics courses.
- **I tried to acknowledge all people from whom I used images or videos. In case I made a mistake or missed someone, please let me know.**
- The photogrammetry material heavily relies on the very well written lecture notes by Wolfgang Förstner and the Photogrammetric Computer Vision book by Förstner & Wrobel.
- Parts of the robotics material stems from the great Probabilistic Robotics book by Thrun, Burgard and Fox.
- If you are a university lecturer, feel free to use the course material. If you adapt the course material, please make sure that you keep the acknowledgements to others and please acknowledge me as well. To satisfy my own curiosity, please send me email notice if you use my slides.

Cyrill Stachniss, cyrill.stachniss@igg.uni-bonn.de