Computer Vision (CS763)

Teaching cameras to “see”

Camera Extrinsics and Intrinsics

Arjun Jain

Motivation

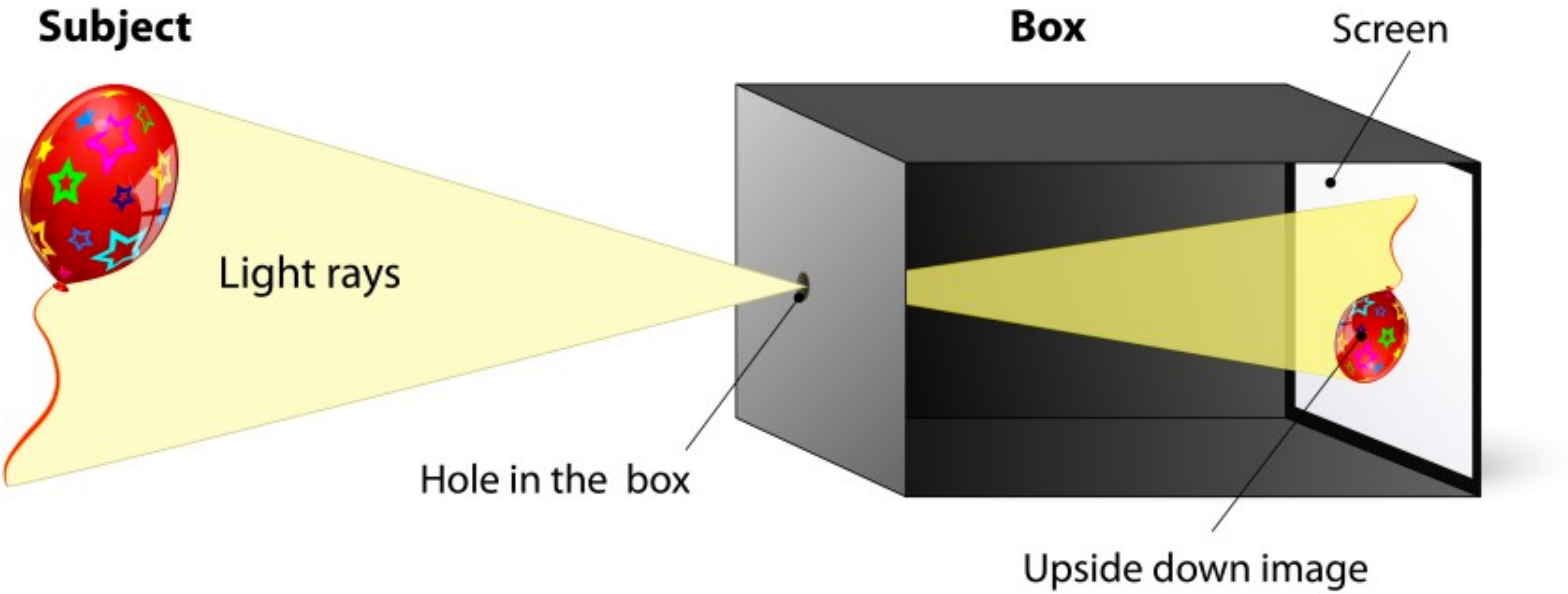
For estimating the geometry of the scene based on images, we need to understand the image acquisition

Image courtesy: Förstner

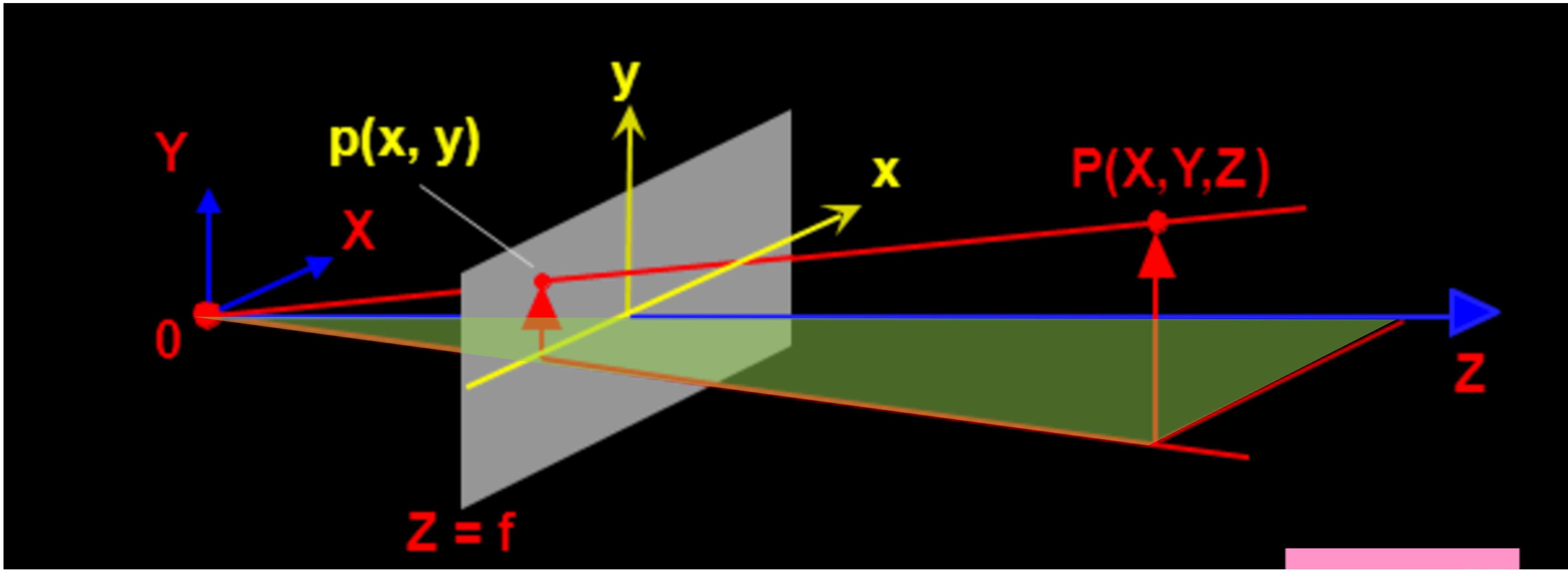
Pin Hole Camera Model

1. A box with an infinitesimal hole
2. Camera center is the intersection point of all incoming rays
3. Back wall is the image plane
4. Distance between the hole and the back wall is called **camera constant**

Pin Hole Camera Model



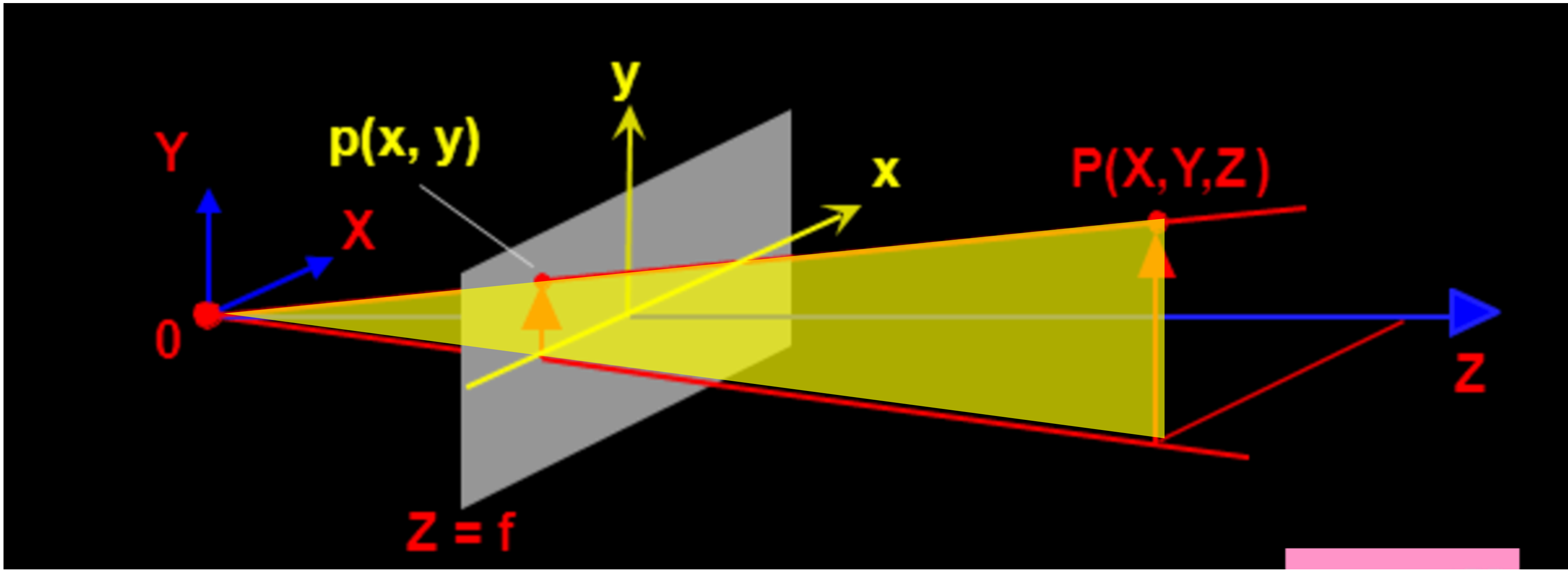
Pin Hole Camera Model



- From similar triangles: $x/X = c/Z$
- $(X, Y, Z)^T \longmapsto (cX/Z,\ cY/Z)$

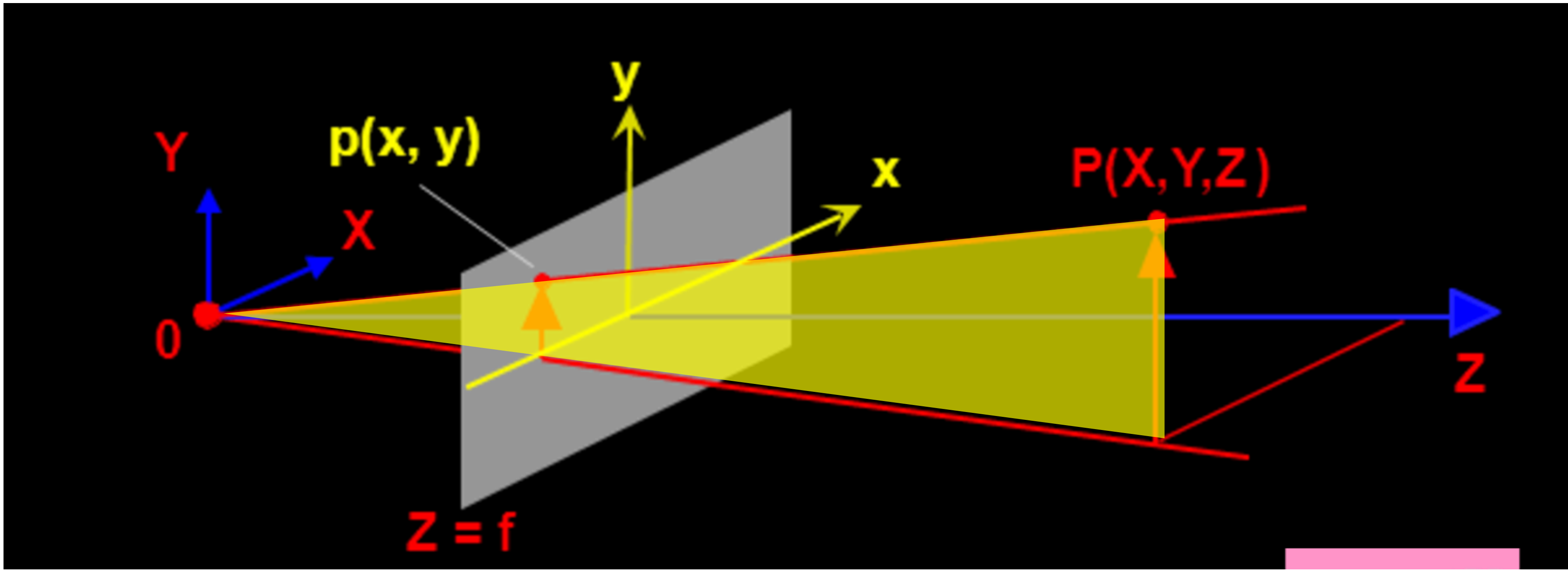
non-linear ☹

Pin Hole Camera Model



- From similar triangles: $y/Y = c/Z$

Pin Hole Camera Model



$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \longmapsto \begin{bmatrix} cX \\ cY \\ Z \end{bmatrix} = \begin{bmatrix} c & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Linear Projection in H.C.!

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

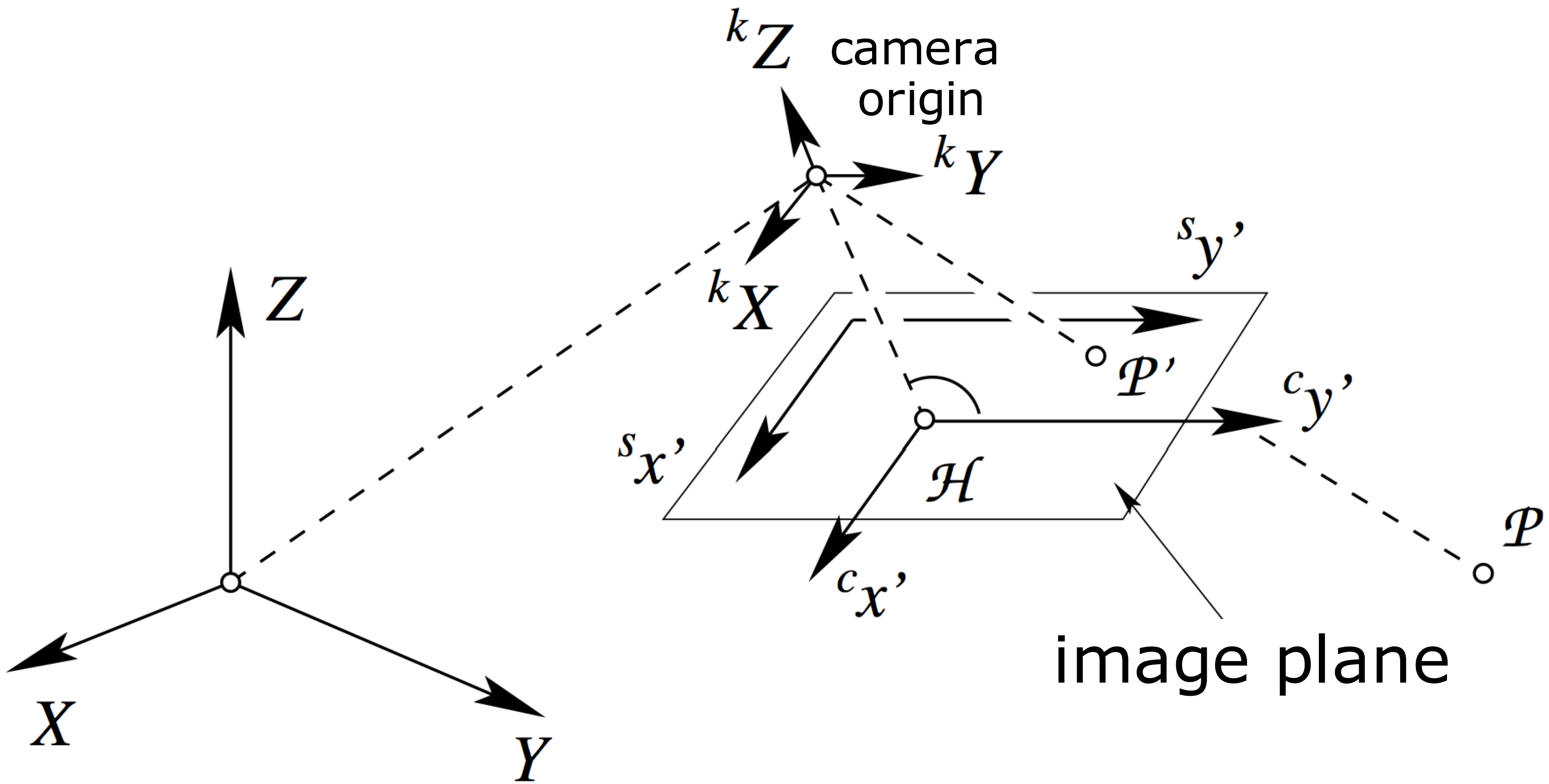
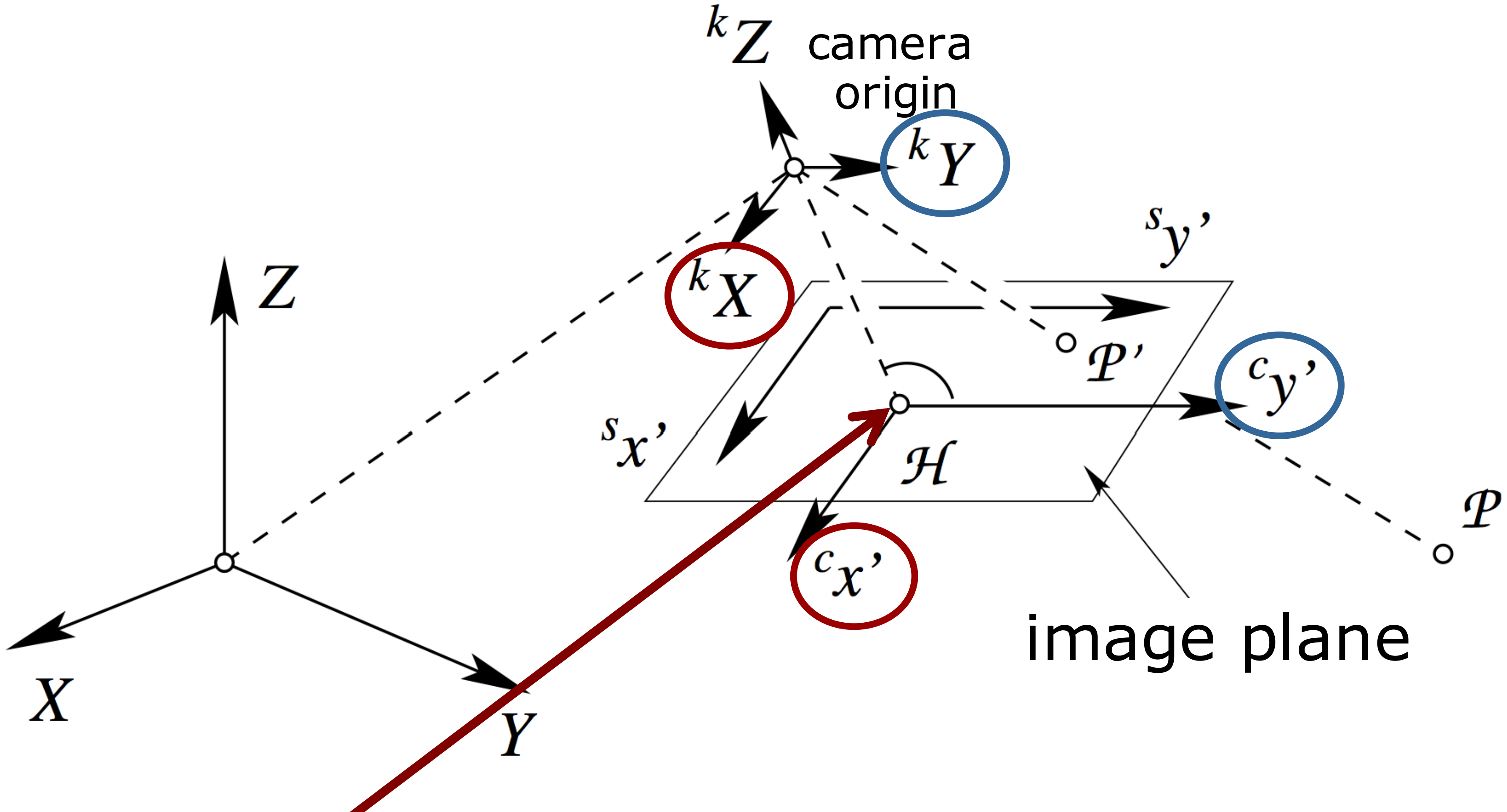


Image courtesy: Förstner

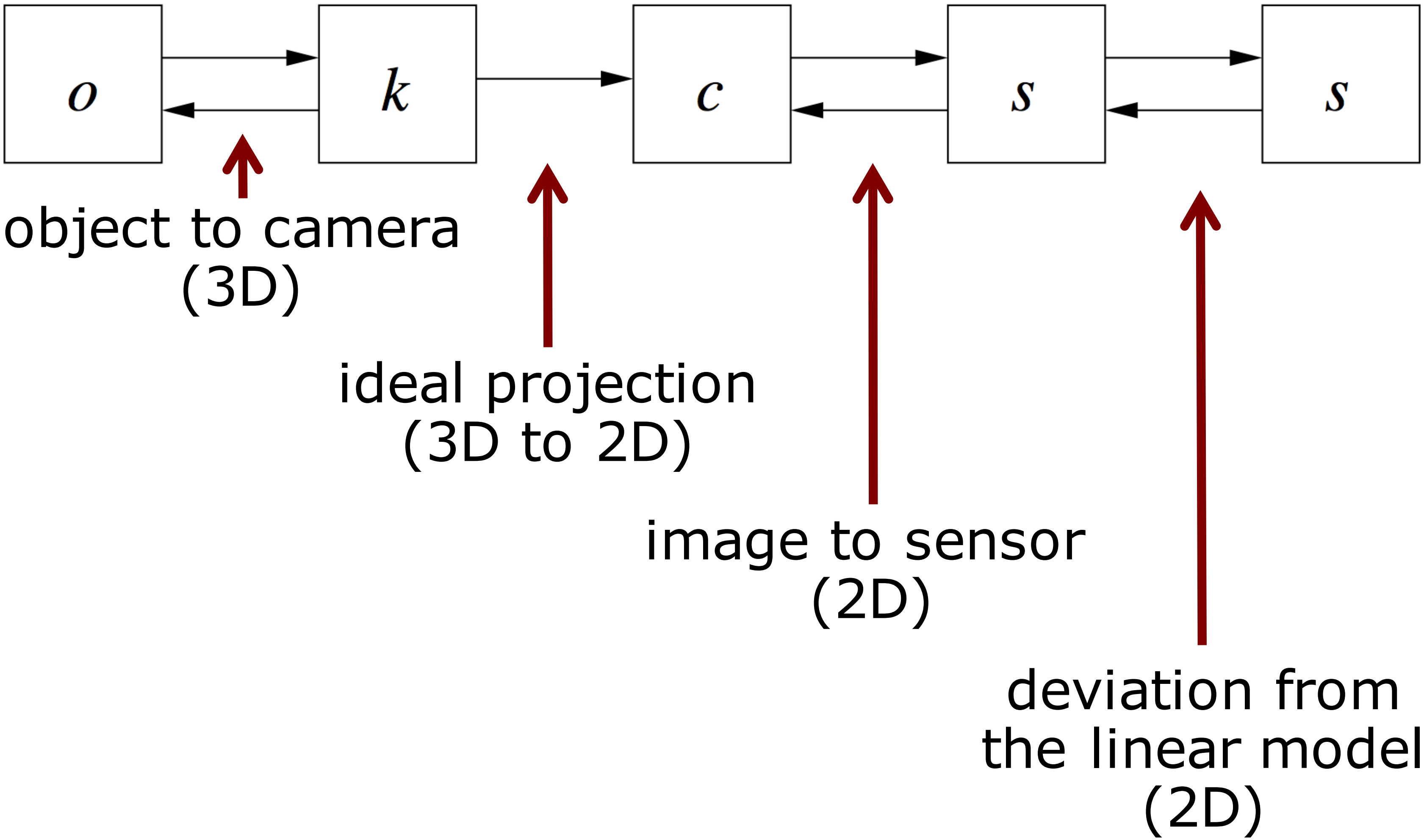
Example

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)

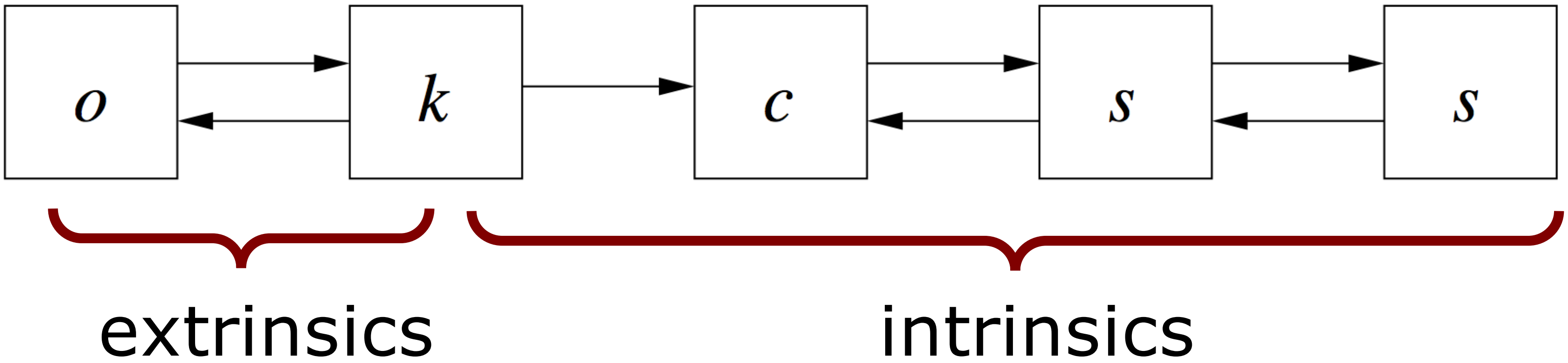


$$^kO_c = {}^k[0, 0, c]^\mathsf{T} \text{ (with } c < 0\text{)}$$

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

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)

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

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.

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

Euclidian

H.C.

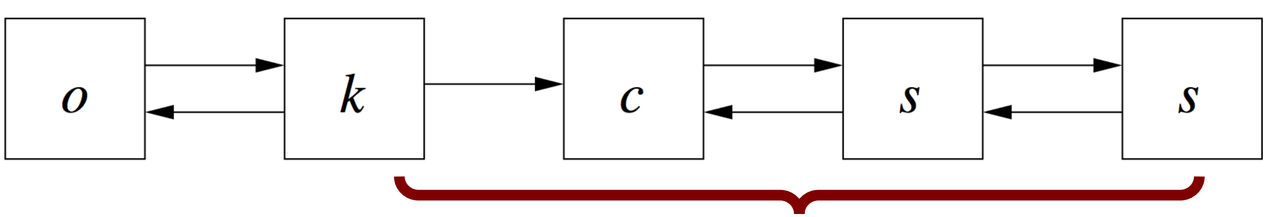
- or written in short as

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

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



Projection: From 3D to 2D

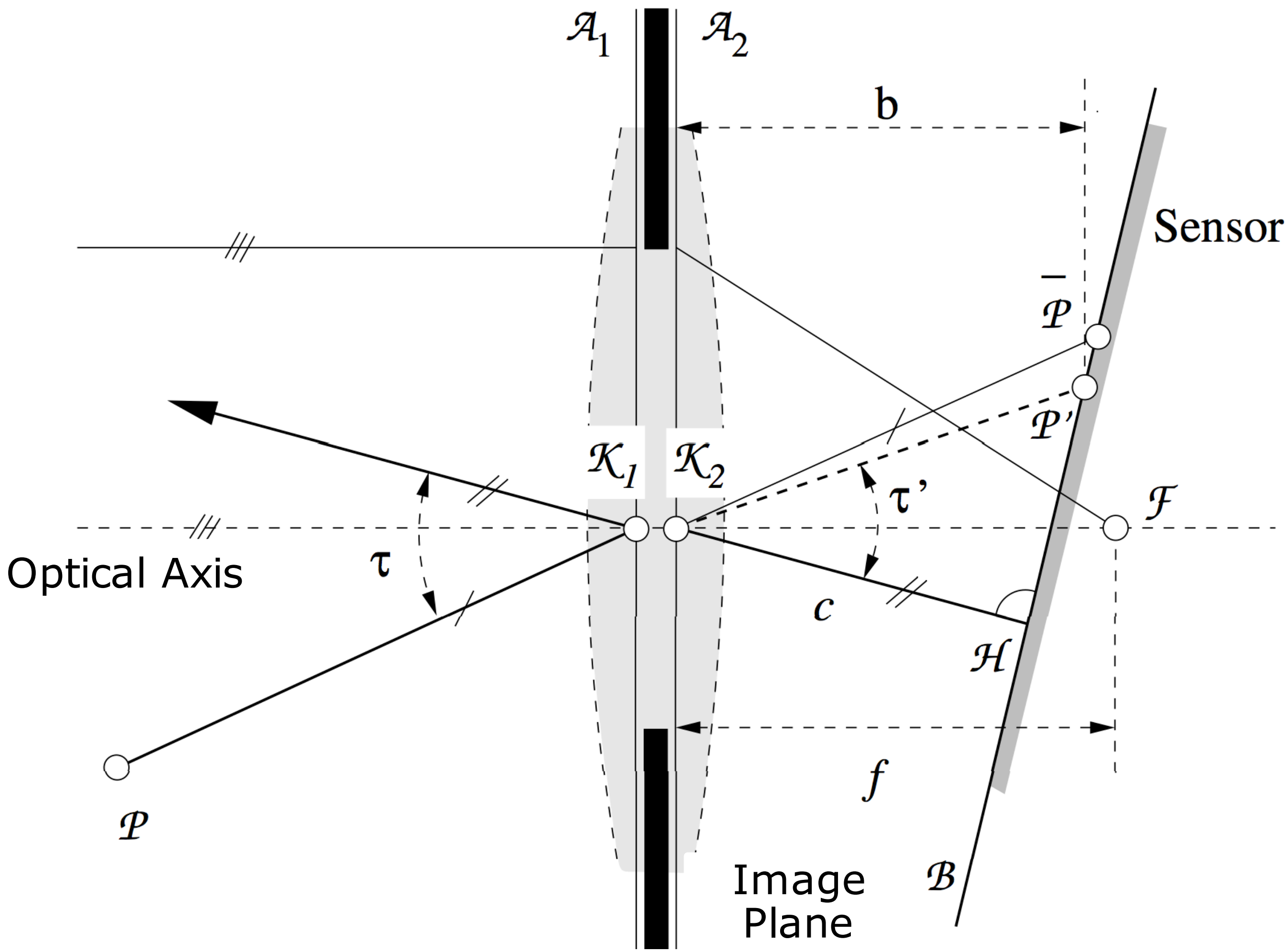


Image courtesy: Förstner

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 mapping are idealized

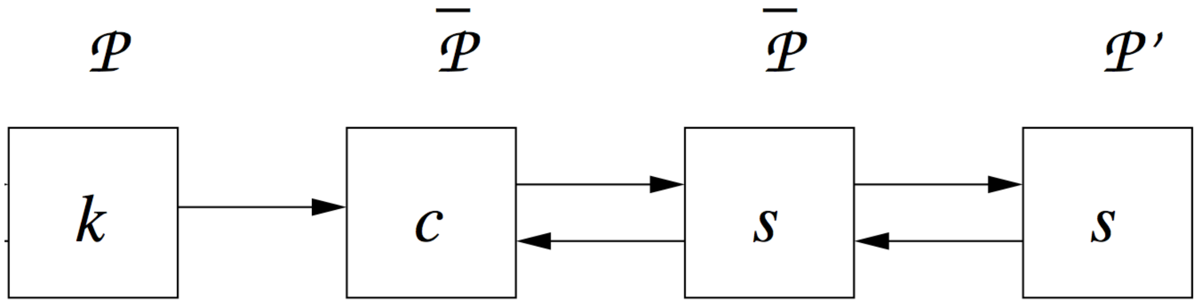
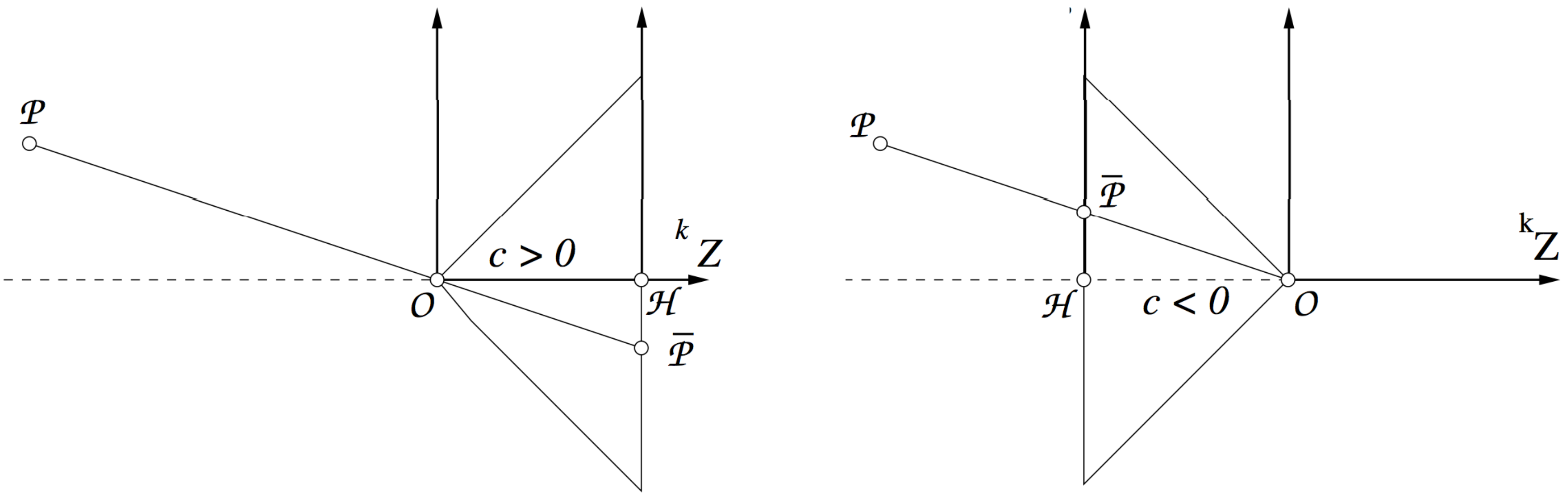


Image courtesy: Förstner

Ideal Perspective Projection

- Distortion-free lens
- Focal point $\mathcal{F}$ and principal point $\mathcal{H}$ lie on the optical axis
- All rays are straight lines and pass through $\mathcal{K}_1 = \mathcal{K}_2$. This point is the origin of the camera coordinate system S_k
- The distance from the camera origin to the image plane is the constant c

Image Coordinate System



Physically motivated coordinate system: c>0

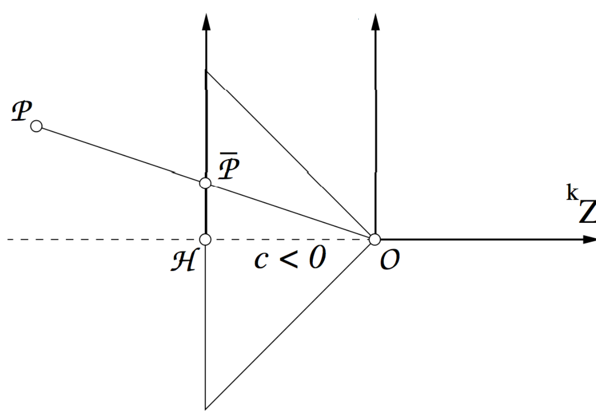
Most popular image coordinate system: c<0

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 $c < 0$

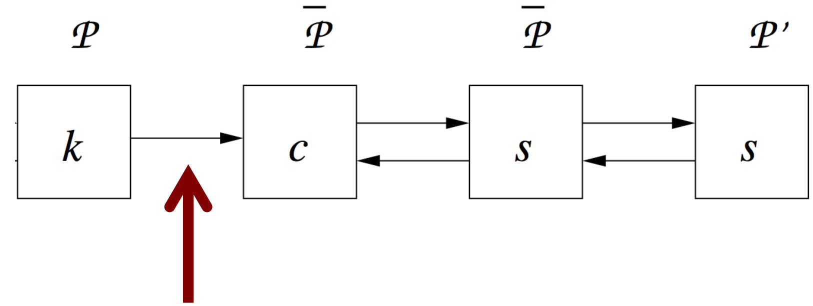


mage courtesy: Förstner

Ideal Perspective Projection

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

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



(Similar Triangles)

In Homogenous Coordinates

- We can express that in H.C.

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

- and drop the 3rd coordinate

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

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**

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

- By re-writing 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 ${}^c\mathbf{x}$

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

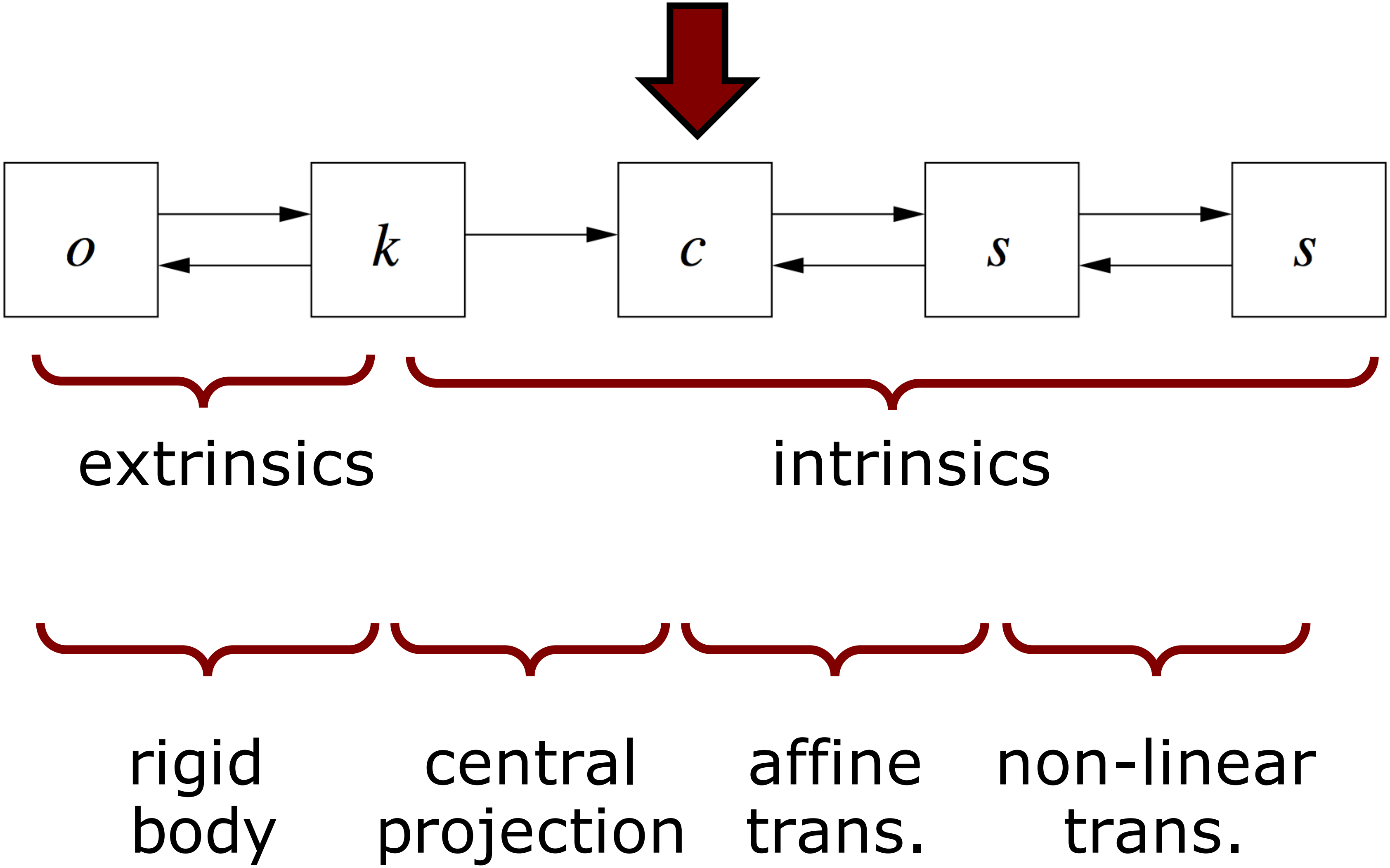
In Euclidian Coordinates

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

$$
{}^{c}x = c\frac{r_{11}(X - X_O) + r_{12}(Y - Y_O) + r_{13}(Z - Z_O)}{r_{31}(X - X_O) + r_{32}(Y - Y_O) + r_{33}(Z - Z_O)}
$$

$$
{}^{c}y = c\frac{r_{21}(X - X_O) + r_{22}(Y - Y_O) + r_{23}(Z - Z_O)}{r_{31}(X - X_O) + r_{32}(Y - Y_O) + r_{33}(Z - Z_O)}
$$

Where Are We in the Process?



Mapping to the Sensor (assuming linear errors)

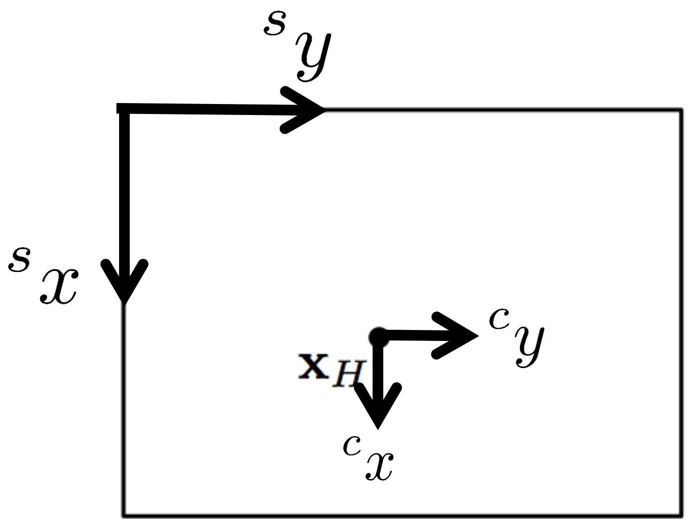
Linear Errors

- 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
- Sheer compensation

Location of the Principal Point

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

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



Sheer and Scale Difference

- Scale difference m in x and y
- Sheer 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
  - sheer: s

DLT: Direct Linear Transform

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

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

- 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

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

- 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

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

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

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

2D Homography

Fundamental Theorem of Projective Geometry

- Every one-to-one, straight-line preserving mapping of a projective space $\mathbb{P}^n$ onto itself is a homography (projectivity) for $2 \leq n < \infty$

- Implies that all one-to-one, straight-line preserving transformations are linear if we use projective(homogeneous) coordinates

2D Homography/Projectivity

- Homography

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

or

$$\mathbf{x}' = \mathbf{H}\,\mathbf{x}$$

8DOF

- Most general 2D transform – projectivity – parallel lines are not parallel – but lines maps lines

Homography

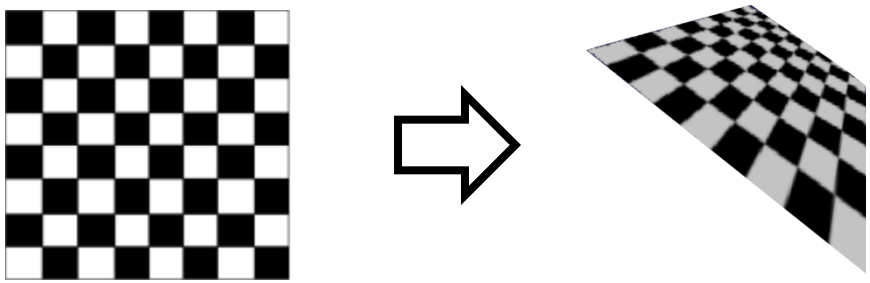
- Note: homographies are not restricted to 2D

- General definition: A homography is a non-singular, line preserving, projective mapping $h = \mathbb{P}^n \rightarrow \mathbb{P}^n$

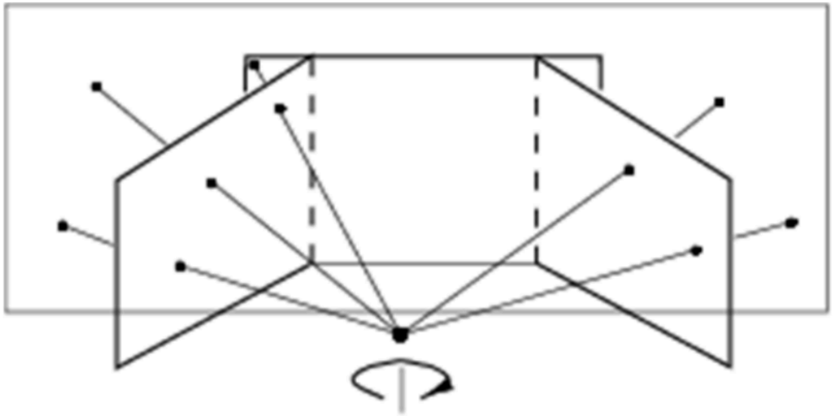
- It is represented by a square $(n+1)$ dim matrix with $(n+1)^2{-}1$ DOF

... back to 2D Homography

- Models perspective effects for a planar scene

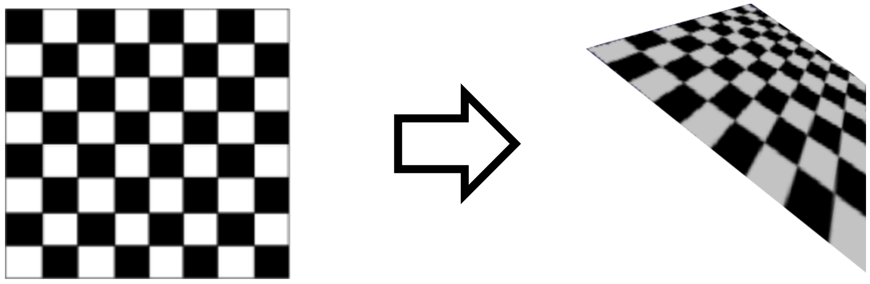


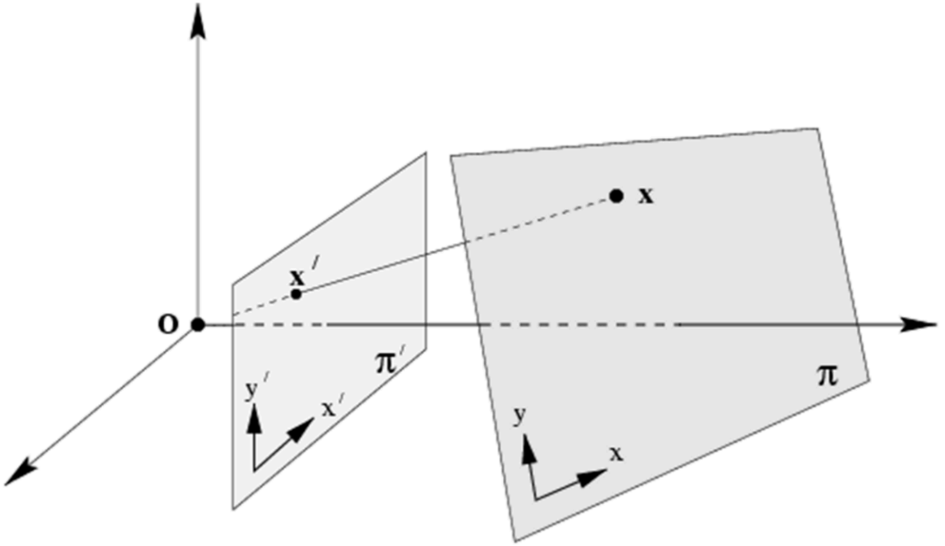
- Models perspective effects from camera rotations



2D Homography

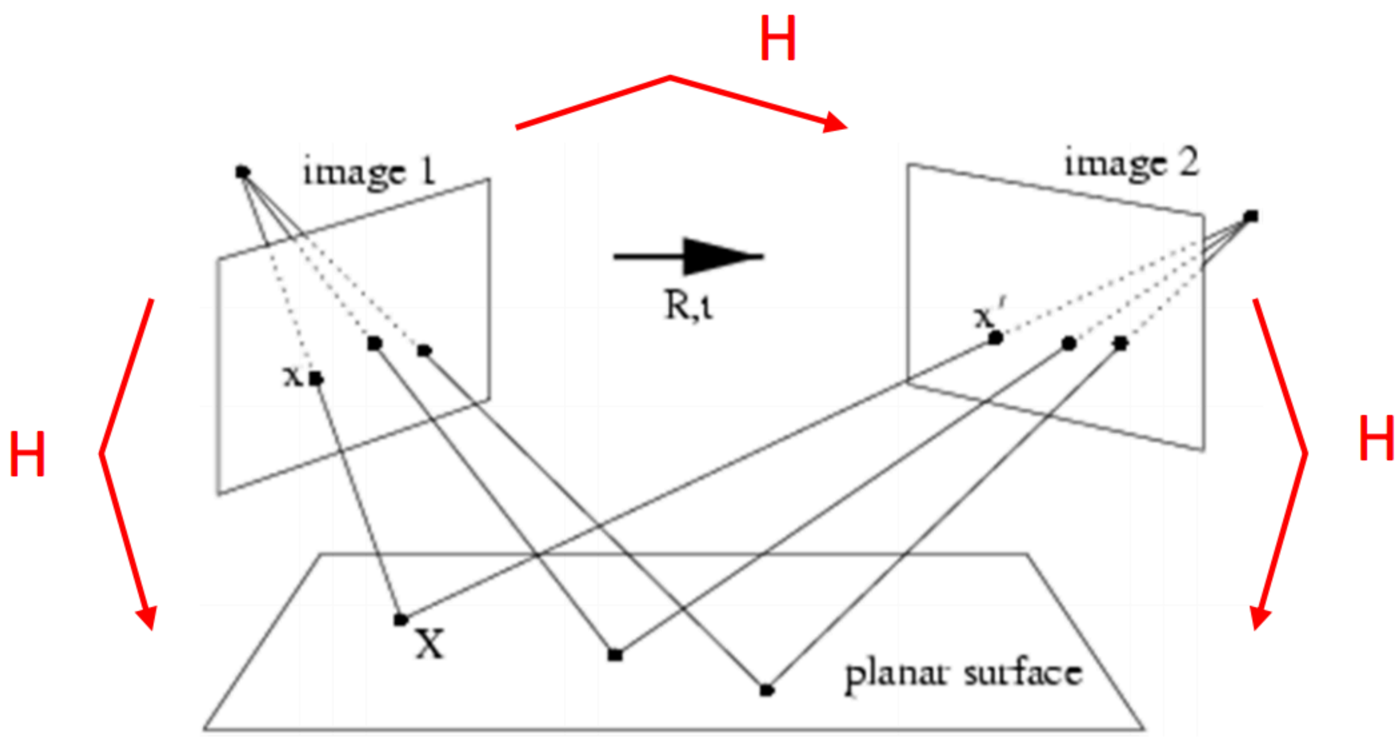
- Mapping between planes





2D Homography

- Rotating / translating camera, planar world



Projection of a 3D Plane

- Place world coordinate frame on object plane

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

Projection of a 3D Plane

- Place world coordinate frame on object plane

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Projection of a 3D Plane

- Place world coordinate frame on object plane

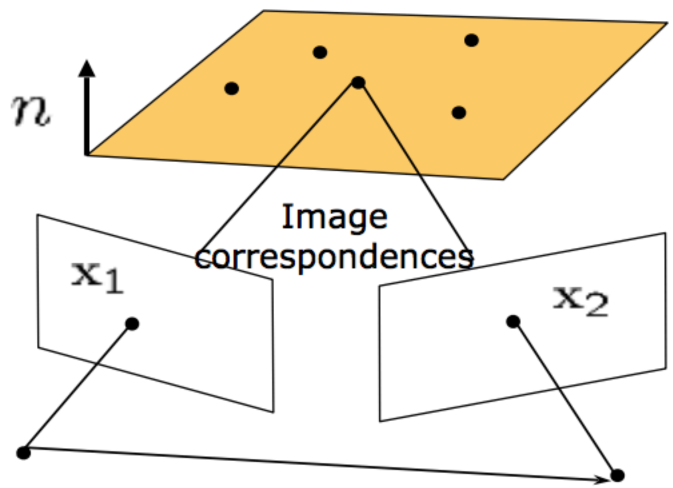
$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} fr_{11} & fr_{12} & ft_x \\ fr_{21} & fr_{22} & ft_y \\ r_{31} & r_{32} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Convert between a location on object plane(world) and image coordinate with a 3X3 matrix H

2 Views of a 3D Plane



$$\lambda_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = H_1 \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

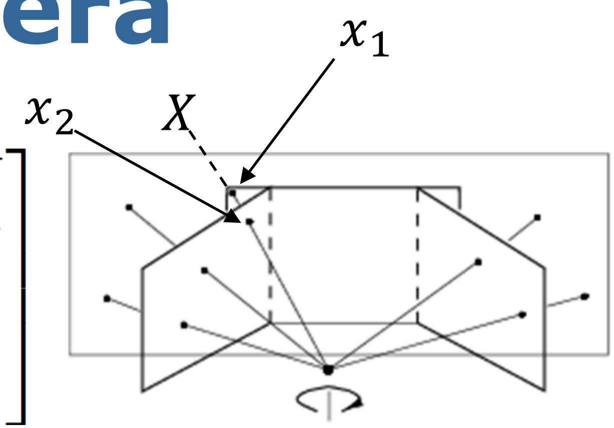
$$\lambda_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = H_2 \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

$$\Rightarrow \quad \lambda \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = H_2 H_1^{-1} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Convert between projections of a 3D plane in 2 views with a 3X3 matrix H

Pure Rotation of Camera

$$\lambda_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \overbrace{\begin{bmatrix} f_1 & 0 & 0 \\ 0 & f_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{K_1} \begin{bmatrix} \overbrace{\begin{matrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{matrix}}^{R_1} & \begin{matrix} \cancel{t_x} \\ \cancel{t_y} \\ \cancel{t_z} \end{matrix} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ \cancel{1} \end{bmatrix}$$



$$\lambda_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \overbrace{\begin{bmatrix} f_2 & 0 & 0 \\ 0 & f_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{K_2} \begin{bmatrix} \overbrace{\begin{matrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{matrix}}^{R_2} & \begin{matrix} \cancel{t_x} \\ \cancel{t_y} \\ \cancel{t_z} \end{matrix} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ \cancel{1} \end{bmatrix}$$

$$\lambda \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \underbrace{K_2 R_2 {R_1}^{-1} {K_1}^{-1}}_{H} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

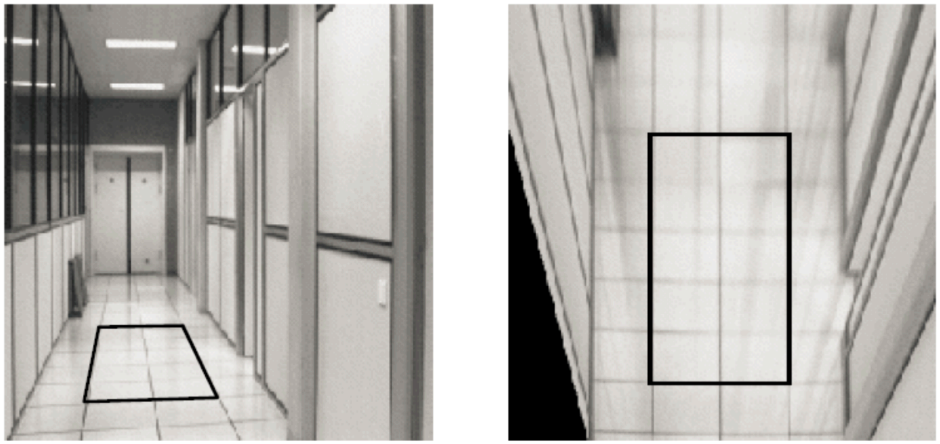
Convert between 2D locations on 2 images with pure camera rotation with a 3X3 matrix H

Take Home Points for 2D Homography

$$\lambda \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

- Most general 2D transform (projective)
- If camera rotates about its center, then the images are related by a homography irrespective of scene depth.
- If the scene is planar, then images from any two cameras are related by a homography.
- Homography mapping is a 3x3 matrix with 8 degrees of freedom

Applications of H (Homework)



from Hartley & Zisserman

Rectification

Image Stitching

How to Estimate 2D Homography

- Analogous to Zhang's camera calibration (next lecture)

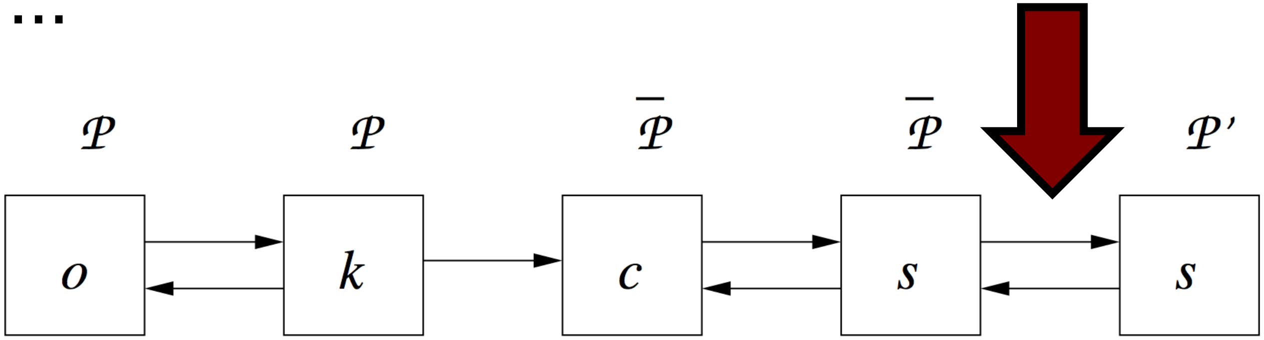
Non-Linear Errors

Non-Linear Errors

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

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

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

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 approach

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

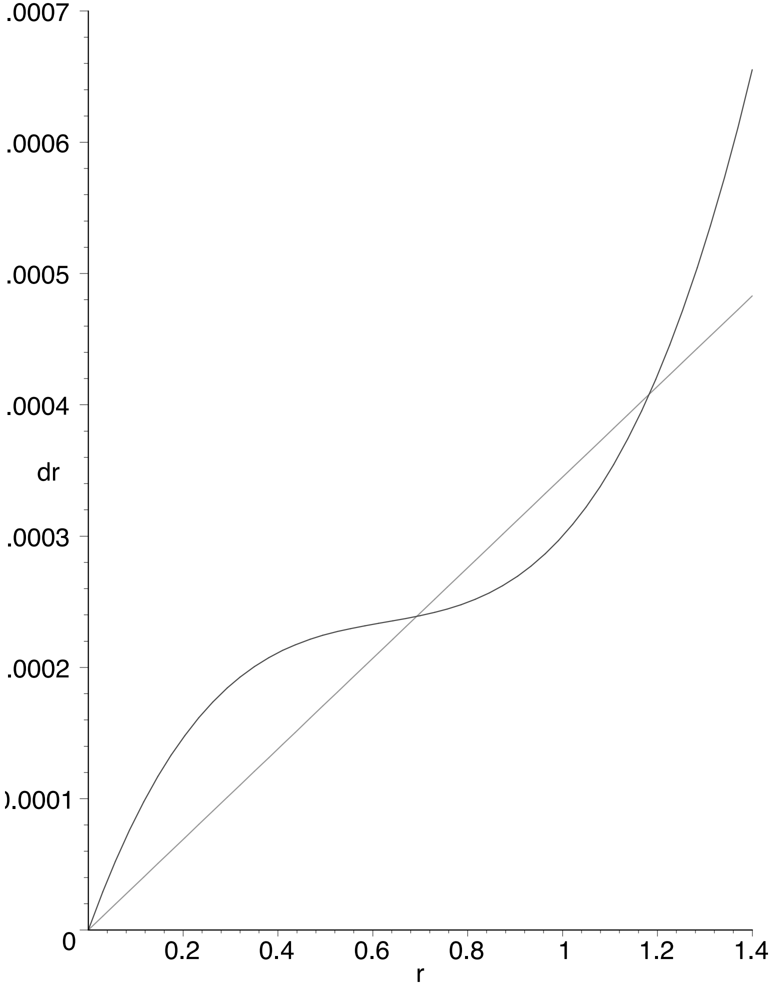
Example: Barrel Distortion

- 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



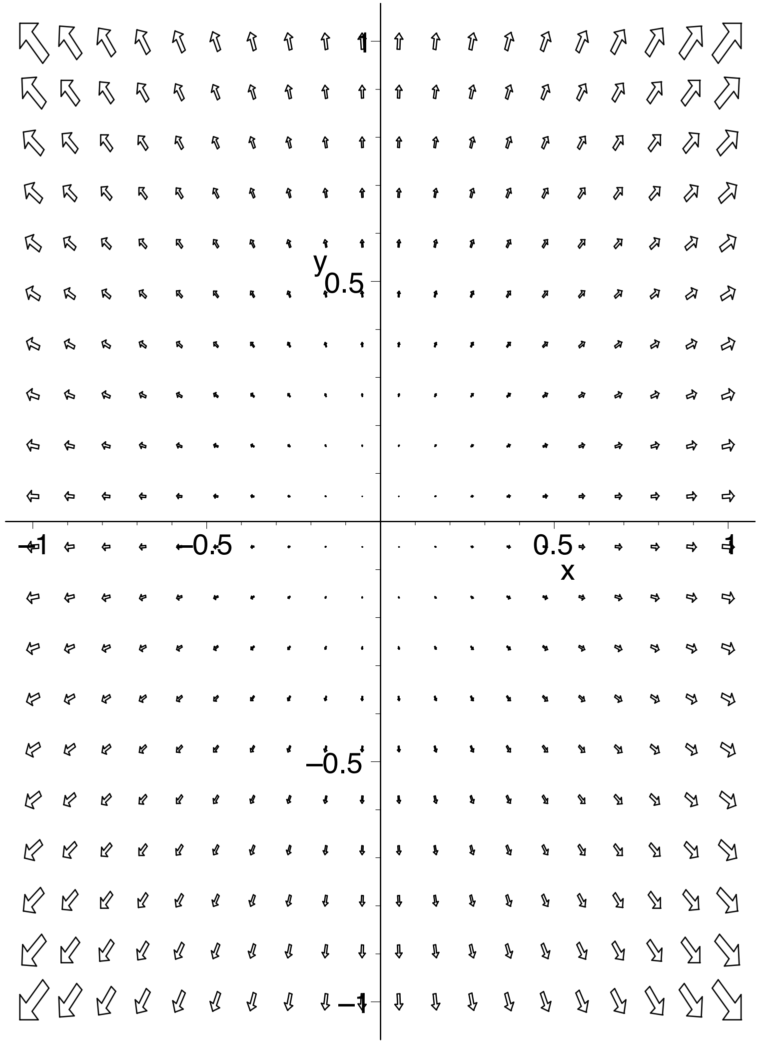
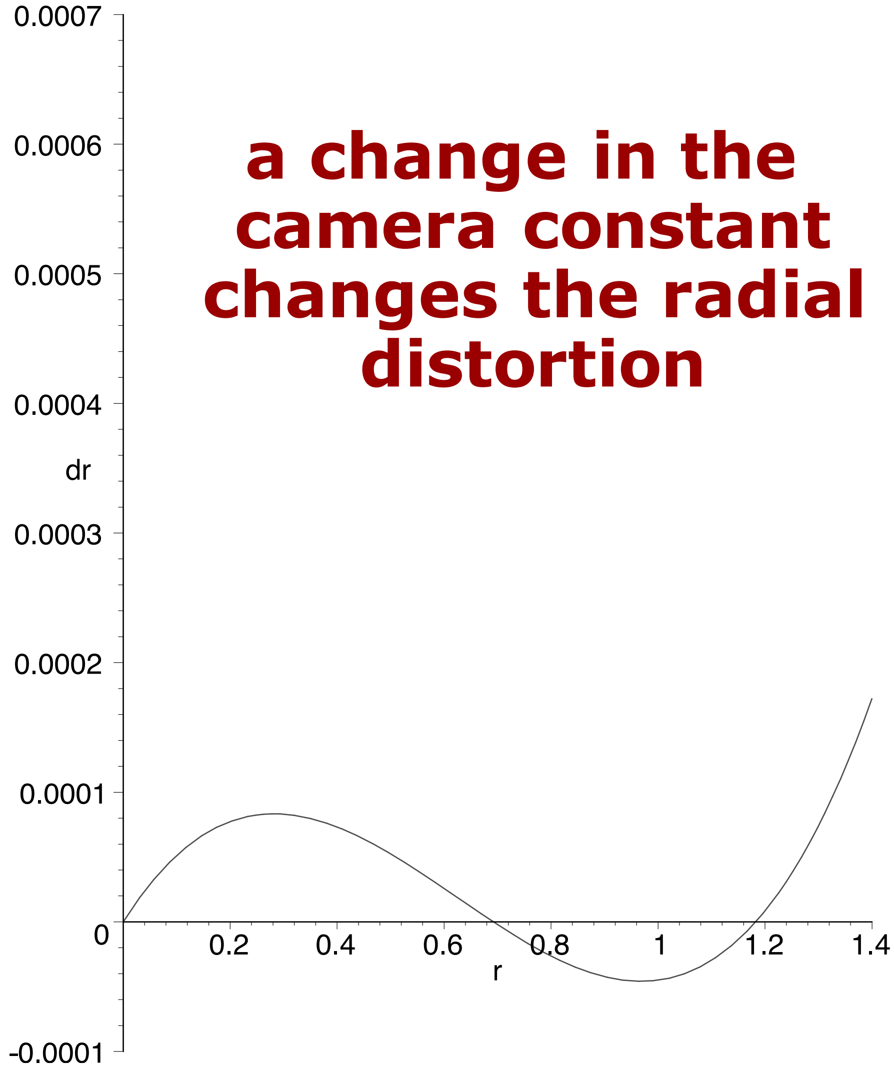


Image courtesy: Förstner

Radial Distortion and Camera Constant Are Dependent



a change in the camera constant changes the radial distortion

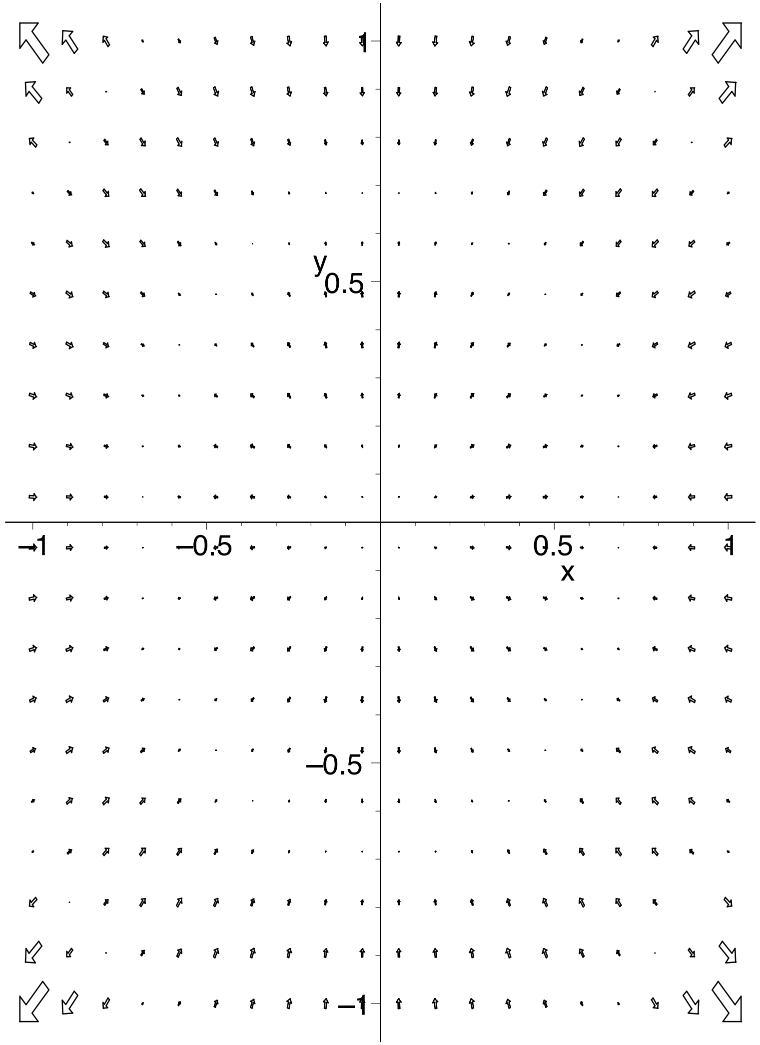


Image courtesy: Förstner

Outlook: What Is Next?

- Estimating the intrinsics
- Estimating the pose of a camera given knowledge about the 3D scene
- Estimating the object location given the known pose of a camera

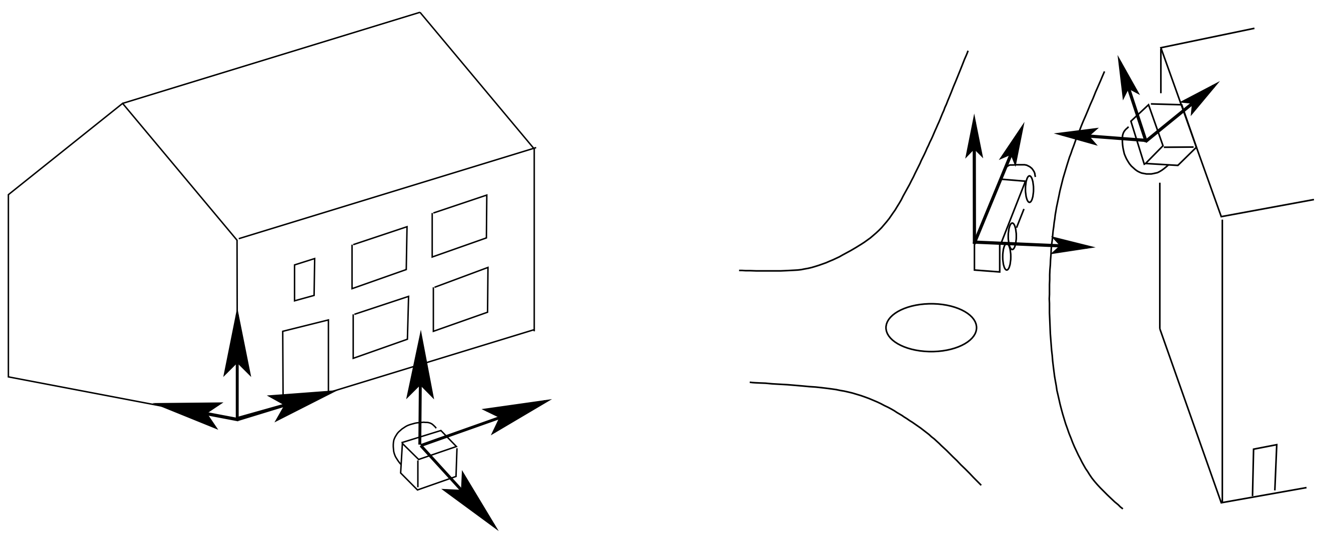


Image courtesy: Förstner

Slide Information

- The slides have been created by Cyrill Stachniss (cyrill.stachniss@igg.uni-bonn.de) as part of the photogrammetry and robotics courses.
- A lot of material from Ajit Rajwade’s CS763 course
- Thanks to Parag for some slides
- **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.

Arjun Jain, ajain@cse.iitb.ac.in