COMPUTER VISION AND PHOTOGRAMMETRY

CAMERA GEOMETRY AND CALIBRATION

CONTENT

- Review Image Formation
- Homogeneous Coordinates
- Lens Distortion
- Camera Calibration
- Calibration Methods
- Lens Distortion CALIBRATION
- LAB 1

LAB 0

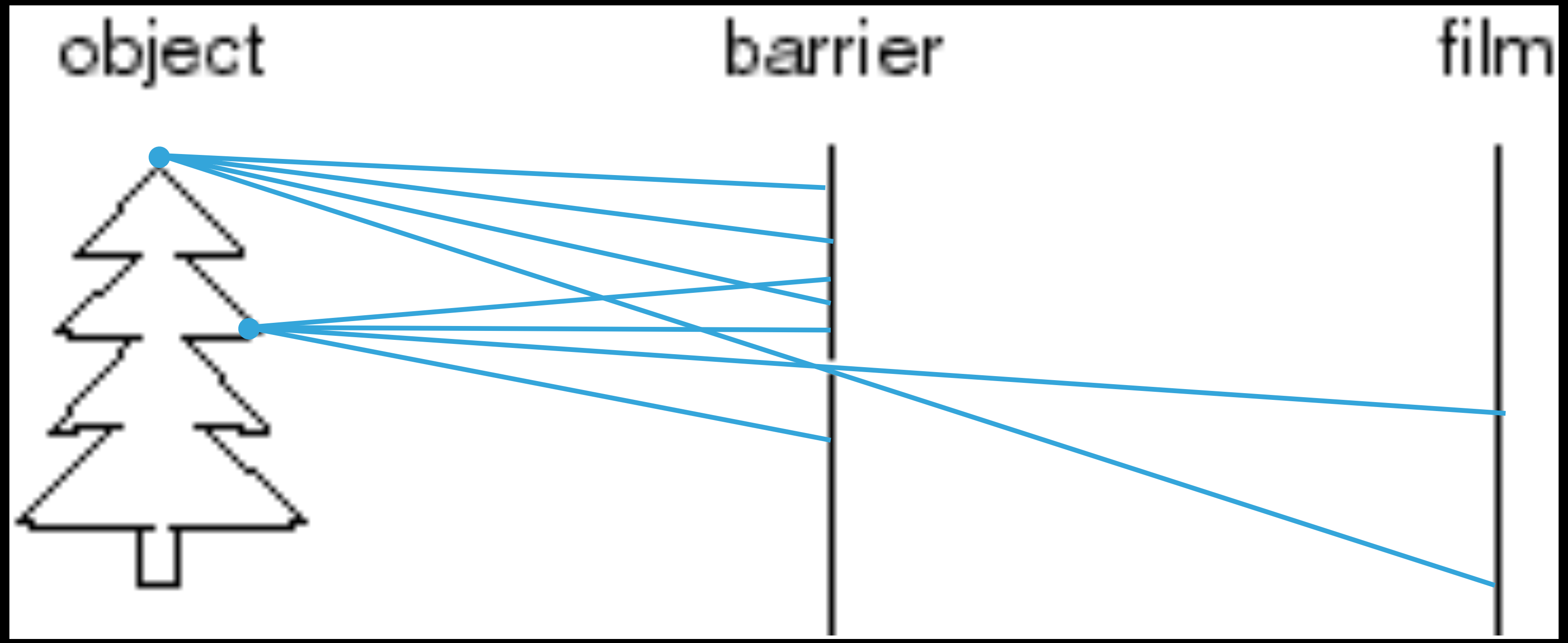
PYTHON AND IMAGES

- Install python. I recommend Anaconda.
- Try Jupyter! Write your report over your code!
- You most likely will use:
  - numpy
  - imageio, Scikit-image, scipy.ndimage
  - Read an image, save an image, do some tutorials, try some filters, so some manipulation on the data.

IMAGE FORMATION AND PROJECTION MATRIX

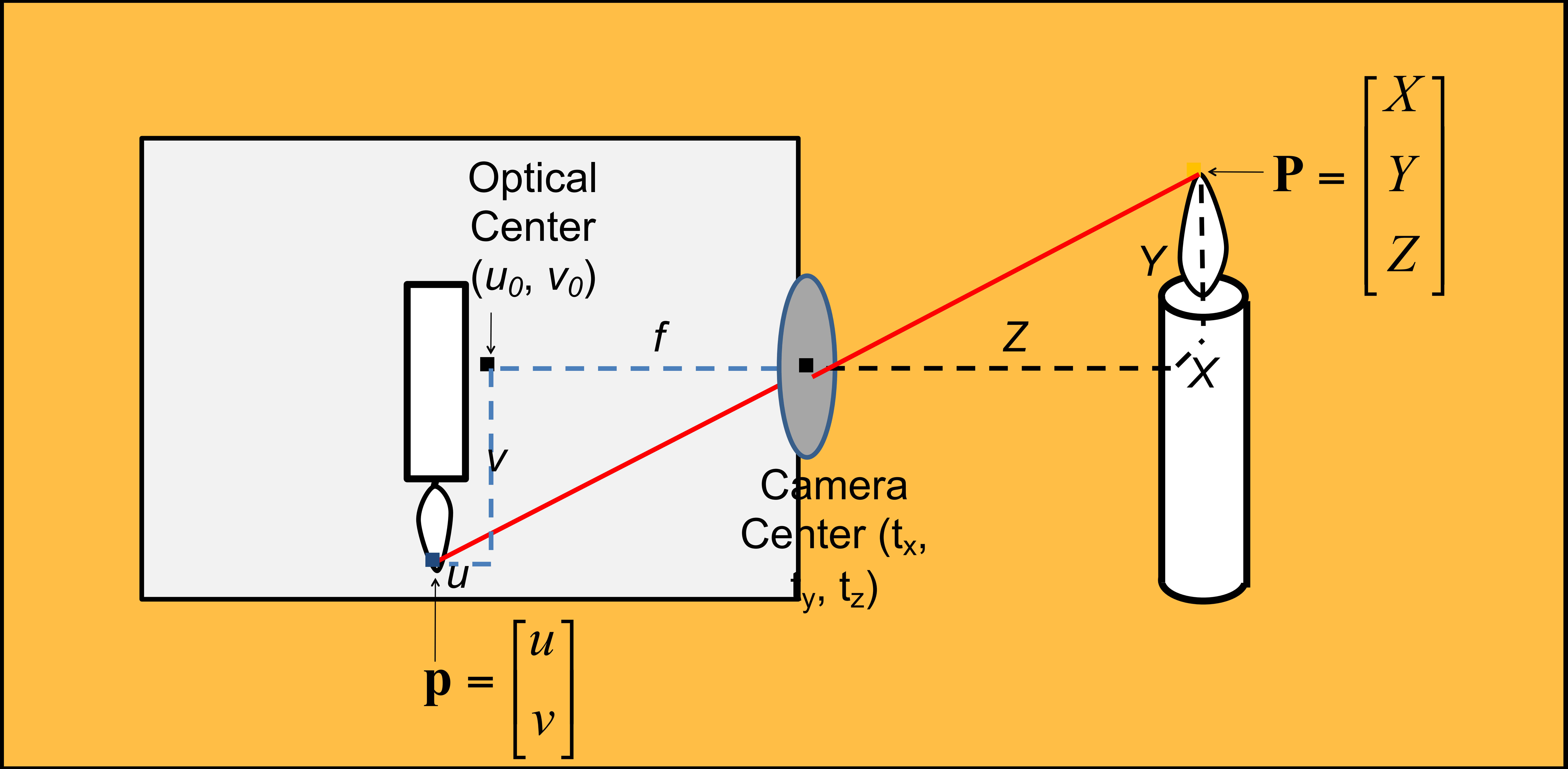
PINHOLE CAMERA

- Idea 2: add a barrier to block off most of the rays
  - This reduces blurring
  - The opening known as the aperture



Slide source: Seitz

PROJECTION: WORLD COORDINATES → IMAGE COORDINATES



PROJECTION: WORLD COORDINATES → IMAGE COORDINATES

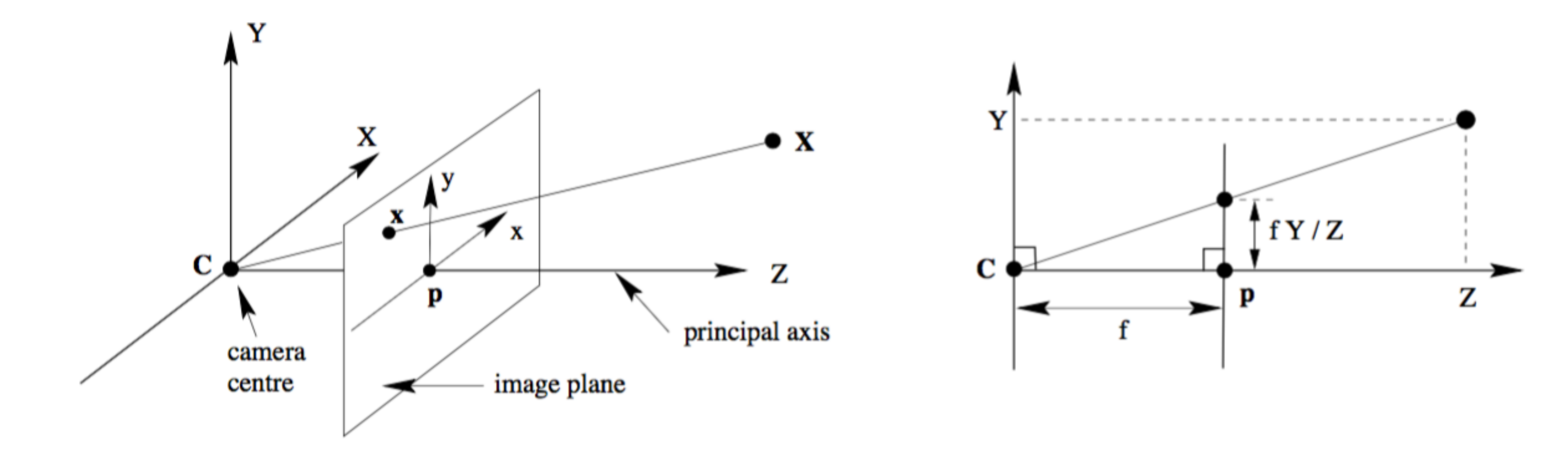
- A camera is a mapping between the 3D world (object space) and a 2D image

PROJECTION: WORLD COORDINATES → IMAGE COORDINATES

PROJECTION

- what is lost?
  - length?
  - angles?
  - straight lines?

PROJECTION: WORLD COORDINATES→IMAGE COORDINATES



$$(X, Y, Z)^{\mathsf{T}} \mapsto (fX/Z, fY/Z)^{\mathsf{T}}$$

HOMOGENEOUS COORDINATES

$$(X, Y, Z)^{\top} \mapsto (fX/Z, fY/Z)^{\top}$$

- Express projection as a linear mapping

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

HOMOGENEOUS COORDINATES

Converting to homogeneous coordinates

- Conversion

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

homogeneous scene coordinates

Converting from homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

HOMOGENEOUS LINES (2D)

- A line in 2D.

$$ax + by + c = 0.$$

- this is the same line (if K is not 0)

$$(ka)x + (kb)y + (kc) = 0$$

- Homogeneous Vector

$$(a, b, c)^{\mathsf{T}} \text{ and } k(a, b, c)^{\mathsf{T}}$$

HOMOGENEOUS POINTS (2D)

- All points in a line

A point $\mathbf{x} = (x, y)^\top$ $\quad$ $ax + by + c = 0.$

- in matrix form

$$(x, y, 1)(a, b, c)^\top = (x, y, 1)\mathbf{l} = 0$$

- Homogeneous Vector

$$(kx, ky, k)\mathbf{l} = 0$$

HOMOGENEOUS COORDINATES

Converting to homogeneous coordinates

- Conversion

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

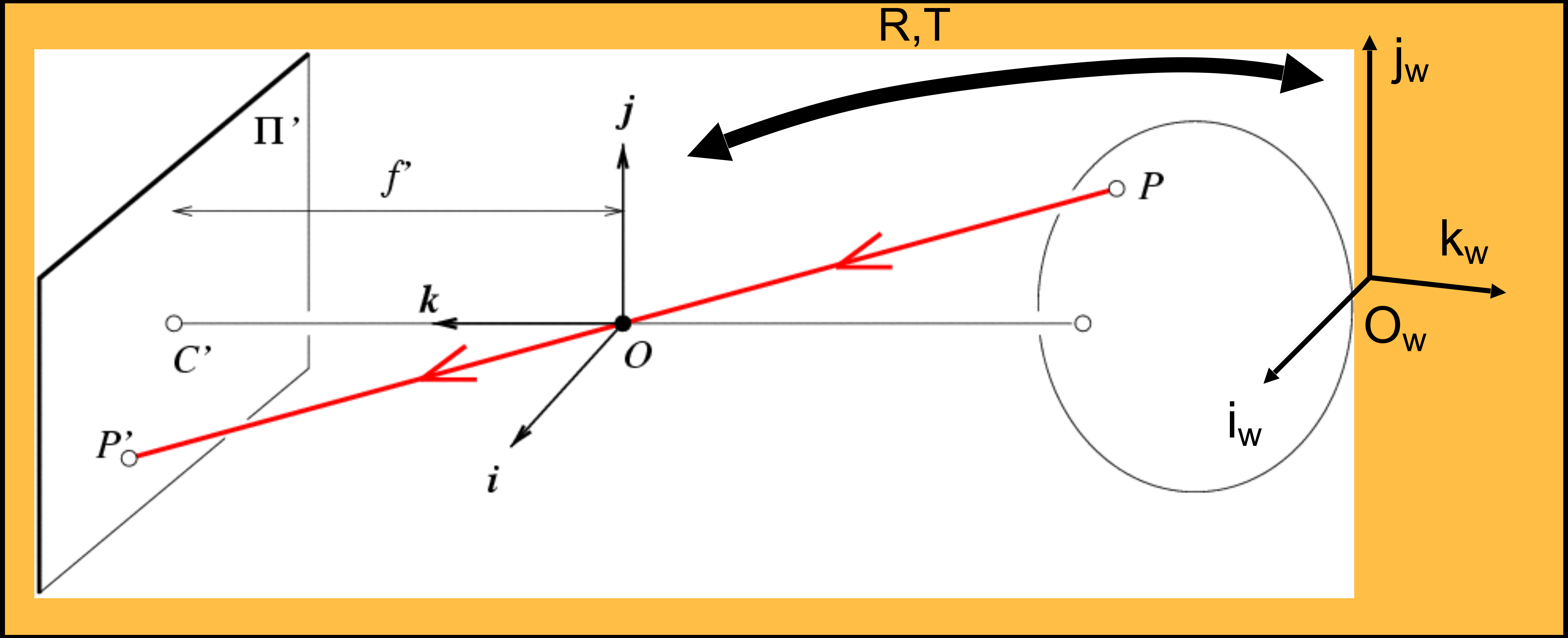
homogeneous scene coordinates

Converting from homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

PROJECTION MATRIX



$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

x: Image Coordinates: (u,v,1)

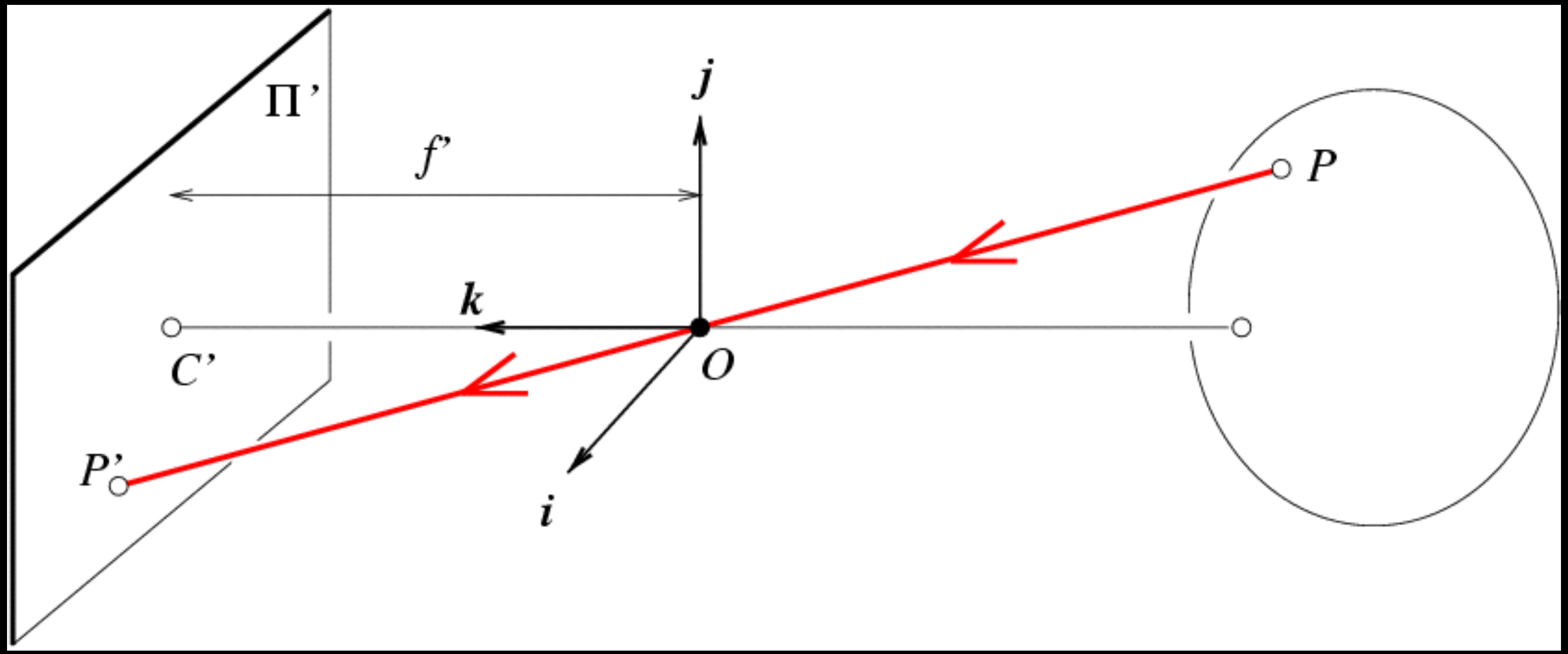
K: Intrinsic Matrix (3x3)

R: Rotation (3x3)

t: Translation (3x1)

X: World Coordinates: (X,Y,Z,1)

PROJECTION MATRIX



Intrinsic Assumptions

- Unit aspect ratio
- Optical center at (0,0)
- No skew

Extrinsic Assumptions

- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}f & 0 & 0 & 0\\0 & f & 0 & 0\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

K

Slide Credit: Saverese

REMOVE ASSUMPTION: KNOWN OPTICAL CENTER

Intrinsic Assumptions

Unit aspect ratio

No skew

Extrinsic Assumptions

No rotation

Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u \\ v \\ 1\end{bmatrix} = \begin{bmatrix}\boxed{\begin{matrix}f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1\end{matrix}} & \begin{matrix}0 \\ 0 \\ 0\end{matrix}\end{bmatrix}\begin{bmatrix}x \\ y \\ z \\ 1\end{bmatrix}$$

REMOVE ASSUMPTION: KNOWN OPTICAL CENTER

Intrinsic Assumptions

No skew

Extrinsic Assumptions

No rotation

Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

REMOVE ASSUMPTION: KNOWN OPTICAL CENTER

Intrinsic Assumptions

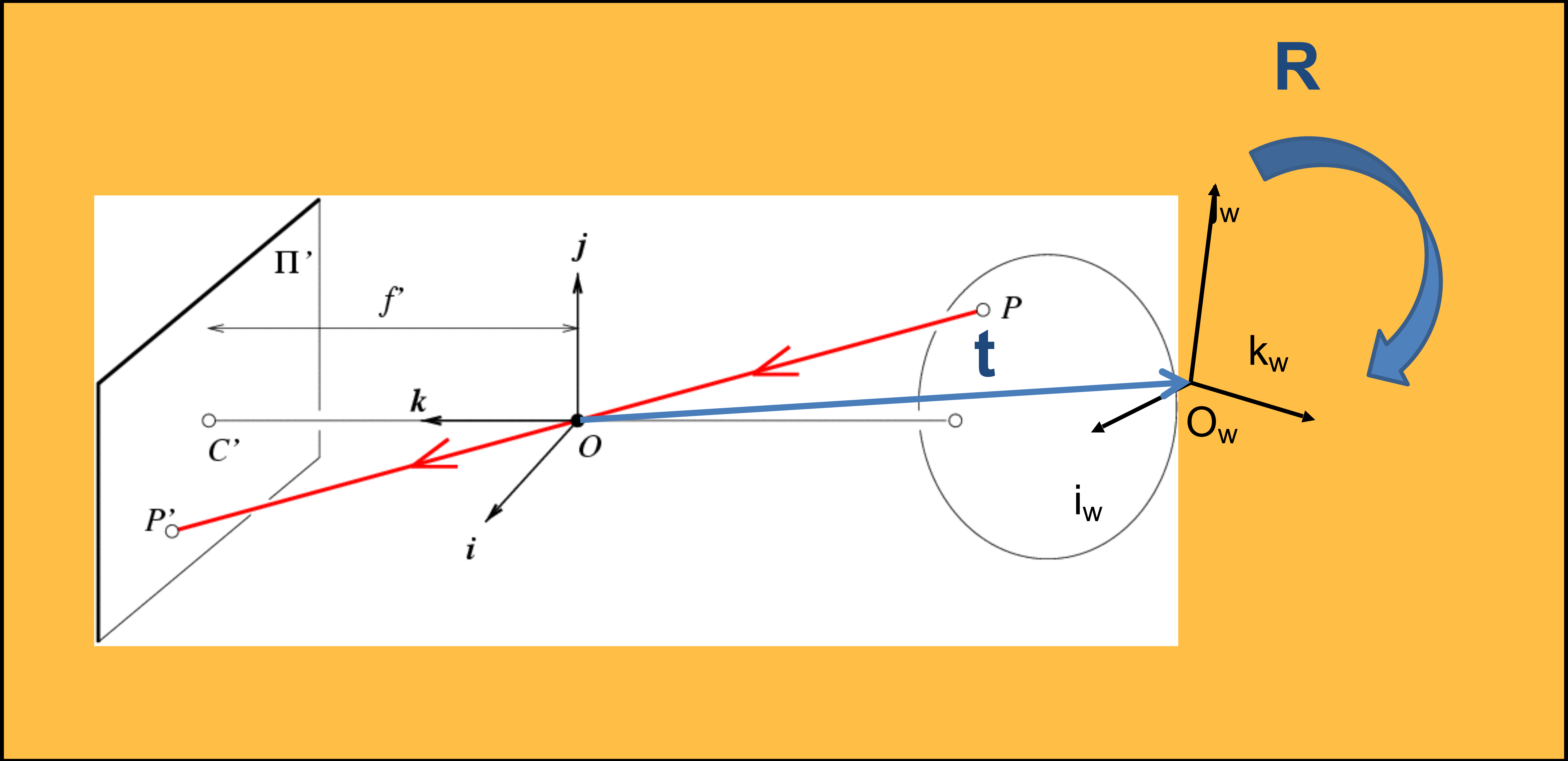
Extrinsic Assumptions

No rotation

Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}\alpha & s & u_0 & 0\\0 & \beta & v_0 & 0\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

ORIENTED AND TRANSLATED CAMERA



REMOVE ASSUMPTION: KNOWN OPTICAL CENTER

Intrinsic Assumptions

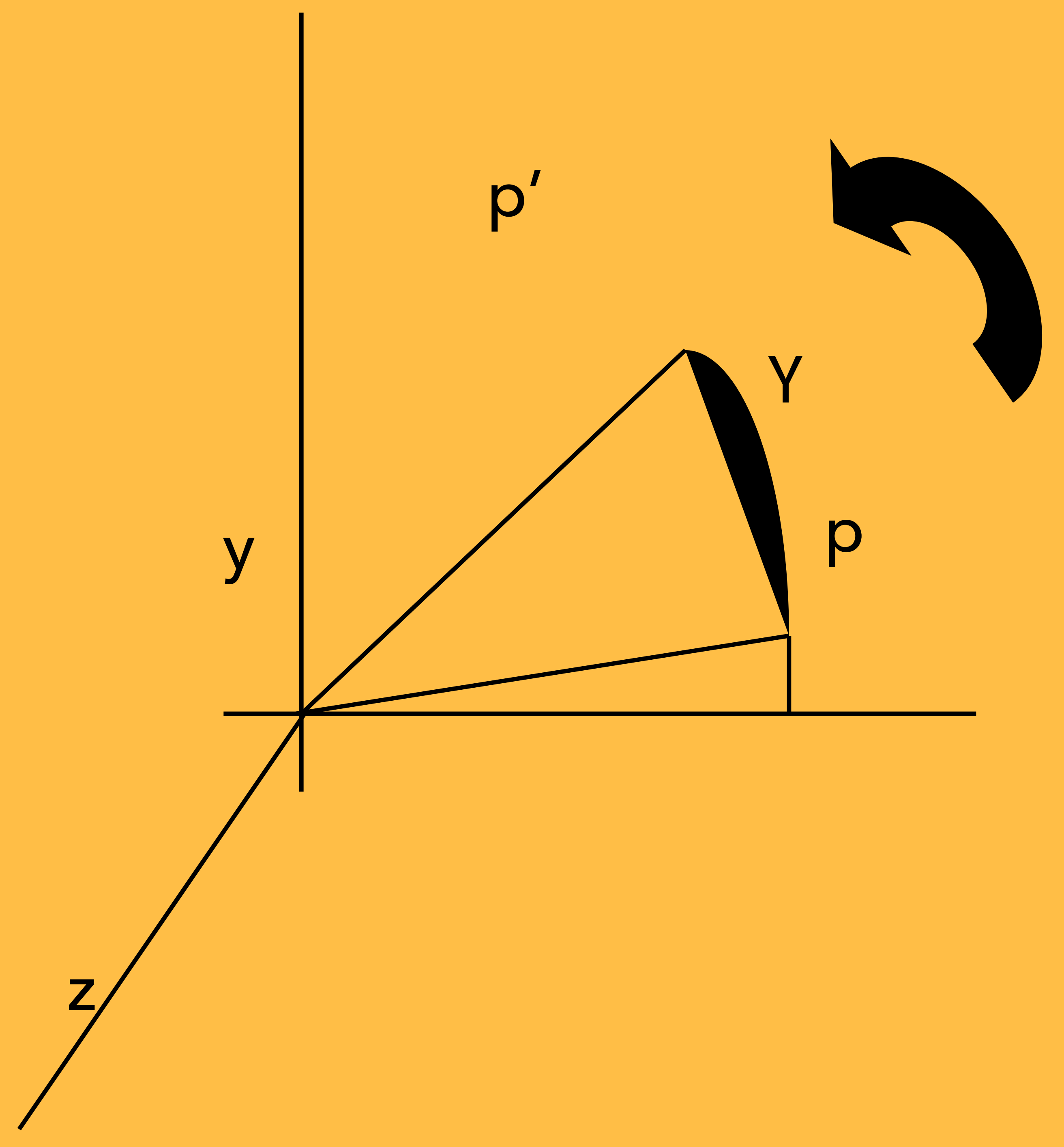
Extrinsic Assumptions

No rotation

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{t}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}\alpha & 0 & u_0\\0 & \beta & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & t_x\\0 & 1 & 0 & t_y\\0 & 0 & 1 & t_z\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

3D ROTATION OF POINTS

Rotation around the coordinate axes, counter-clockwise:



$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ALLOW CAMERA ROTATION

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$w\begin{bmatrix}u \\ v \\ 1\end{bmatrix} = \begin{bmatrix}\alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z\end{bmatrix}\begin{bmatrix}x \\ y \\ z \\ 1\end{bmatrix}$$

HOW TO CALIBRATE THE CAMERA?

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

DEGREES OF FREEDOM

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$w\begin{bmatrix}u\\ v\\ 1\end{bmatrix} = \begin{array}{c} 5 \\ \begin{bmatrix}\alpha & s & u_0\\ 0 & \beta & v_0\\ 0 & 0 & 1\end{bmatrix}\end{array} \begin{array}{c} 6 \\ \begin{bmatrix}r_{11} & r_{12} & r_{13} & t_x\\ r_{21} & r_{22} & r_{23} & t_y\\ r_{31} & r_{32} & r_{33} & t_z\end{bmatrix}\end{array} \begin{bmatrix}x\\ y\\ z\\ 1\end{bmatrix}$$

RADIAL DISTORTION

BEYOND PINHOLES: RADIAL DISTORTION

- Common in wide-angle lenses or for special applications (e.g., security)
- Creates non-linear terms in projection
- Usually handled by through solving for non-linear terms and then correcting image

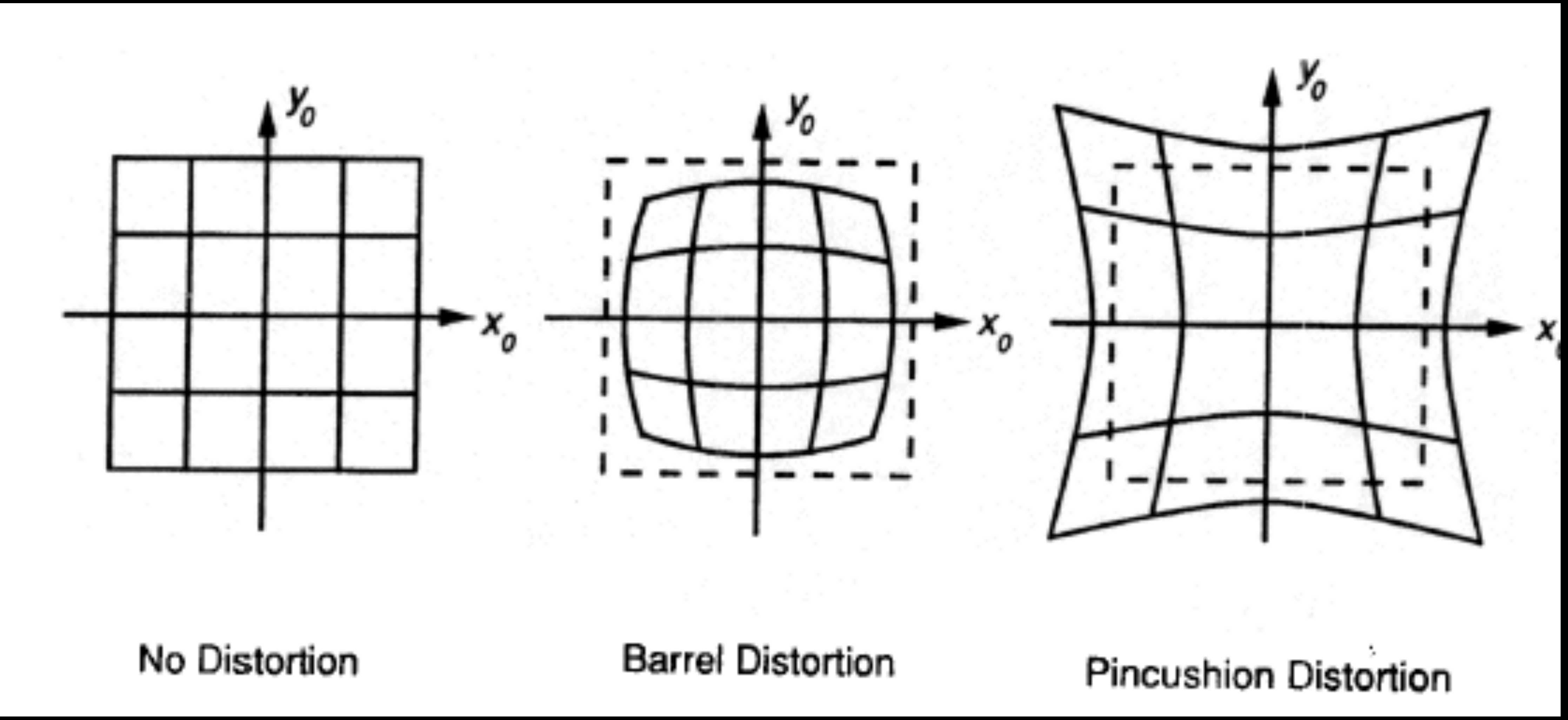
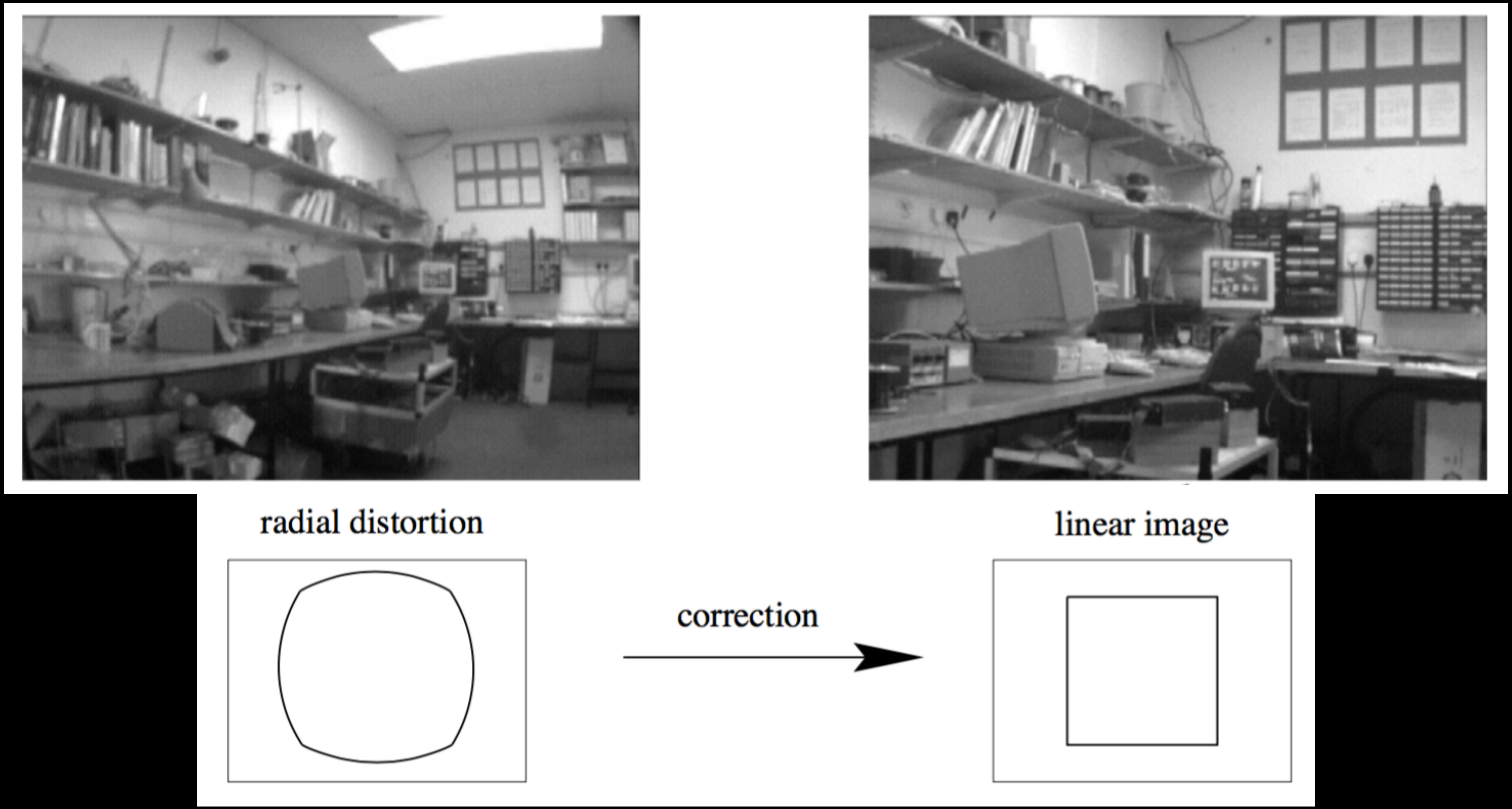


Image from Martin Habbecke

RADIAL DISTORTION



RADIAL DISTORTION

- new image coordinates are:

$$
\begin{aligned}
\hat{x} &= x(1 + \kappa_1 r^2 + \kappa_2 r^4) \\
\hat{y} &= y(1 + \kappa_1 r^2 + \kappa_2 r^4),
\end{aligned}
$$

- where: k1 and k2 are the distortion parameters.

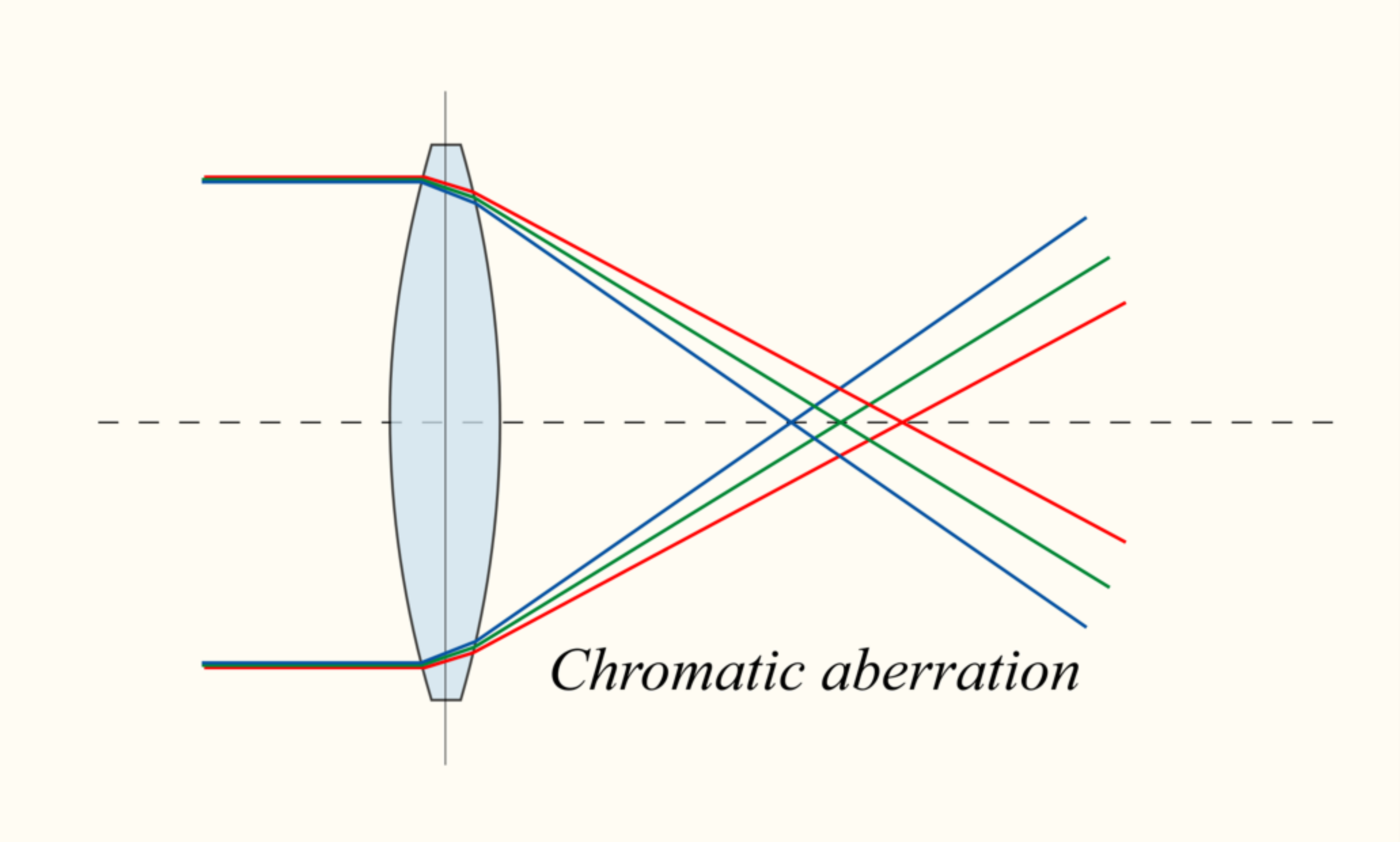
$$r^2 = x^2 + y^2$$

ESTIMATION OF RADIAL DISTORTION

- Plumb-line method
- Another approach is to use several overlapping images and to combine the estimation of the radial distortion parameters with the image alignment process
- Estimation of the other intrinsic and extrinsic parameters

CHROMATIC ABERRATION

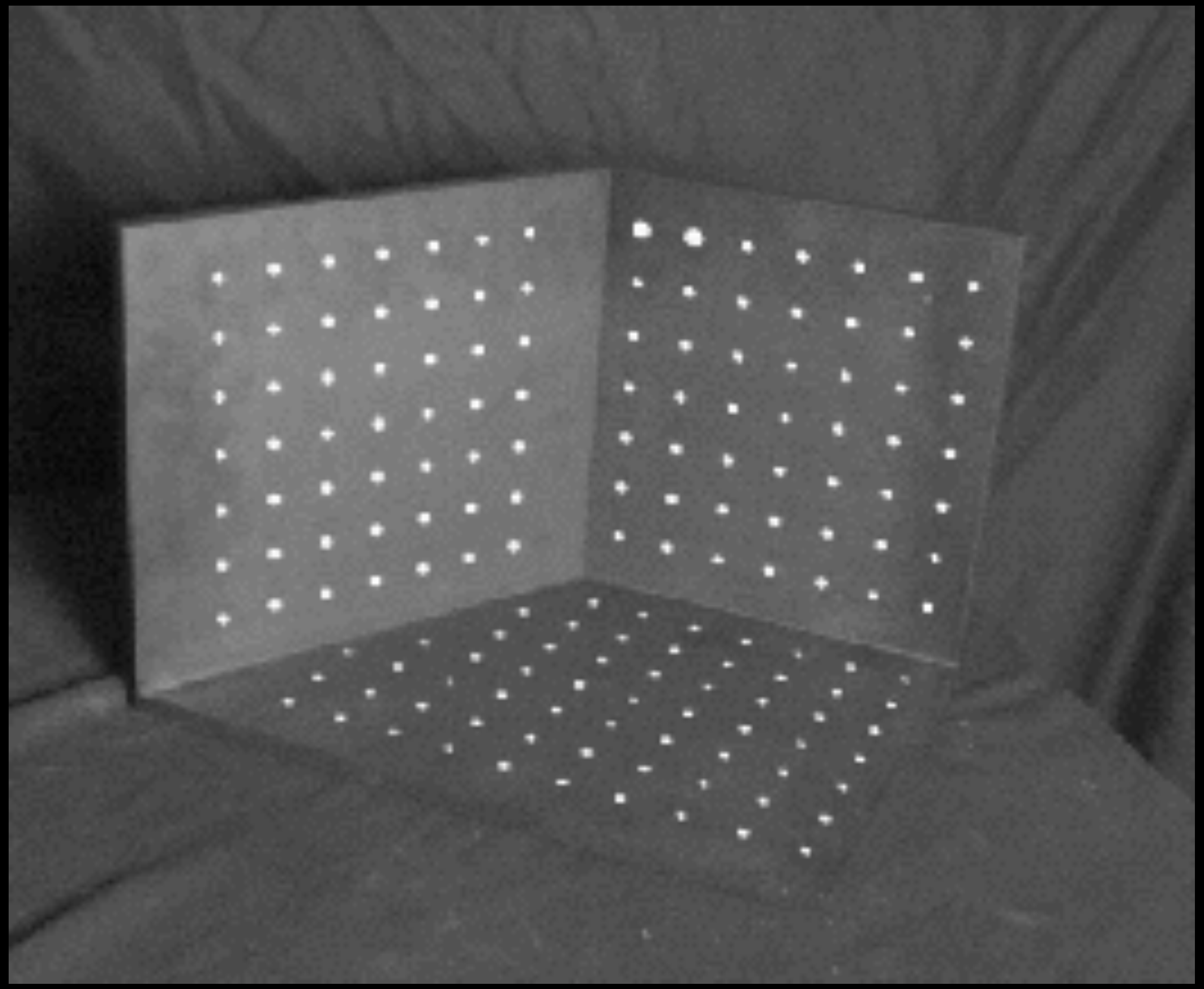
CHROMATIC ABERRATION



CAMERA CALIBRATON AND POSE ESTIMATION

CALIBRATING THE CAMERA

- Use an scene with known geometry
  - Correspond image points to 3d points
  - Get least squares solution (or non-linear solution)



$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

TEXT

Known
$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$
Known

$$su = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$$

$$sv = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$$

$$s = m_{31}X + m_{32}Y + m_{33}Z + m_{34}$$

$$u = \frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}}$$

$$v = \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}}$$

TEXT

$$\text{Known 2d} \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \text{Known}$$

$$u = \frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}}$$

$$v = \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}}$$

$$(m_{31}X + m_{32}Y + m_{33}Z + m_{34})u = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$$
$$(m_{31}X + m_{32}Y + m_{33}Z + m_{34})v = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$$

$$m_{31}uX + m_{32}uY + m_{33}uZ + m_{34}u = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$$
$$m_{31}vX + m_{32}vY + m_{33}vZ + m_{34}v = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$$

TEXT

$$\text{Known 2d} \quad \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \text{Known 3d}$$

$$m_{31}uX + m_{32}uY + m_{33}uZ + m_{34}u = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$$
$$m_{31}vX + m_{32}vY + m_{33}vZ + m_{34}v = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$$

$$0 = m_{11}X + m_{12}Y + m_{13}Z + m_{14} - m_{31}uX - m_{32}uY - m_{33}uZ - m_{34}u$$
$$0 = m_{21}X + m_{22}Y + m_{23}Z + m_{24} - m_{31}vX - m_{32}vY - m_{33}vZ - m_{34}v$$

METHOD 2 – NONHOMOGENEOUS LINEAR SYSTEM

- Solve for m's entries using linear least squares

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Ax=b form

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 \\ & & & & & & \boxed{?} & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ \boxed{?} \\ u_n \\ v_n \end{bmatrix}$$

```
M = A\Y;
M = [M;1];
M = reshape(M,[],3)';
```

METHOD 1 – HOMOGENEOUS LINEAR SYSTEM

- Solve for m's entries using linear least squares

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Ax=0 form

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & & & \boxed{?} & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \boxed{?} \\ 0 \\ 0 \end{bmatrix}$$

```
[U, S, V] = svd(A);

M = V(:,end);

M = reshape(M,[],3)';
```

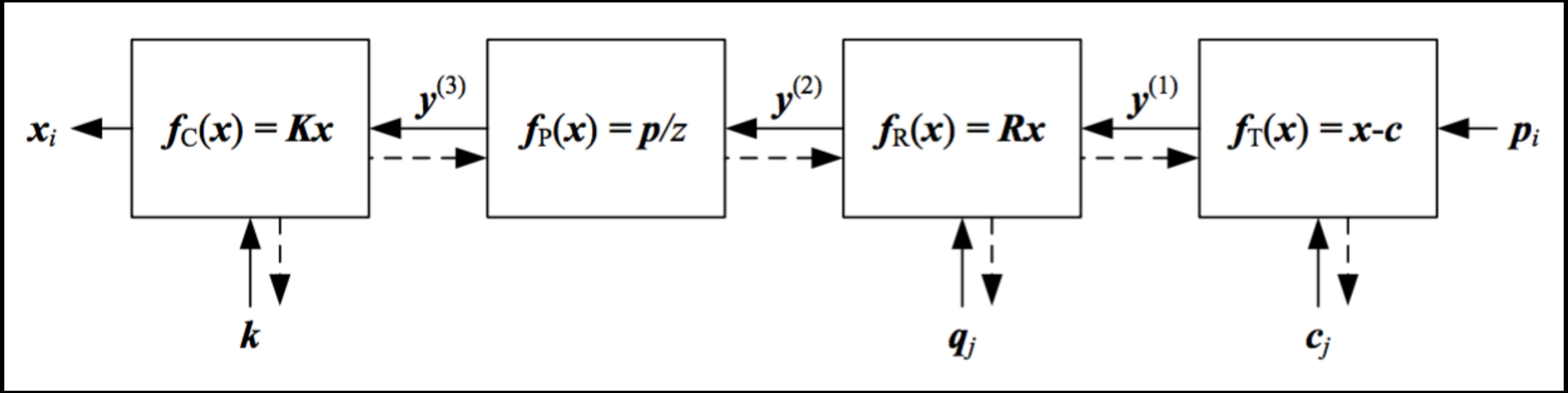
CAN WE FACTORIZE M BACK TO K [R | T]?

- Yes!
- You can use RQ factorization (note – not the more familiar QR factorization). R (right diagonal) is K, and Q (orthogonal basis) is R. T, the last column of [R | T], is inv(K) * last column of M.
    - But you need to do a bit of post-processing to make sure that the matrices are valid. See http://ksimek.github.io/2012/08/14/decompose/

CALIBRATION WITH LINEAR METHOD

- Advantages
  - Easy to formulate and solve
  - Provides initialization for non-linear methods
- Disadvantages
  - Doesn’t directly give you camera parameters
  - Doesn’t model radial distortion
  - Can’t impose constraints, such as known focal length

NON-LINEAR METHODS



NON-LINEAR METHODS

- Non-linear methods are preferred
  - Define error as difference between projected points and measured points
  - Estimate an initital guess with the linear method.
  - Minimize reprojection error using iterative methods such Levenberg-Marquardt

CAN WE FACTORIZE M BACK TO K [R | T]?

- Yes!
- You can use RQ factorization (note – not the more familiar QR factorization). R (right diagonal) is K, and Q (orthogonal basis) is R. T, the last column of [R | T], is inv(K) * last column of M.

CAN WE FACTORIZE M BACK TO K [R | T]?

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}\alpha & s & u_0\\0 & \beta & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & \boxed{t_x}\\r_{21} & r_{22} & r_{23} & t_y\\r_{31} & r_{32} & r_{33} & t_z\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

RQ Factorization
R is upper triangular
Q is orthogonal basis - rotation

$$\begin{bmatrix}su\\sv\\s\end{bmatrix} = \left[\underbrace{\boxed{\begin{matrix}* & * & *\\ * & * & *\\ * & * & *\end{matrix}}}_{Q}\ \boxed{\begin{matrix}*\\ *\\ *\end{matrix}}\right]\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

This is **t** * K

So K^{-1} m_4 is **t**

Q is K * R. So we just need $-Q^{-1}$ m_4

CAN WE FACTORIZE M BACK TO K [R | T]?

- RQ Factorization gives
- Force the diagonal elements of K to be positive, which is the correct approach if two conditions are true:
  - your image's X/Y axes point in the same direction as your camera's X/Y axes.
  - your camera looks in the positive-z direction.

```
# make diagonal of K positive
T = diag(sign(diag(K)));

K = K * T;
R = T * R; # (T is its own inverse)
```

CAN WE FACTORIZE M BACK TO K [R | T]?

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$w\begin{bmatrix}u\\ v\\ 1\end{bmatrix} = \begin{bmatrix}\alpha & s & u_0\\ 0 & \beta & v_0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_x\\ r_{21} & r_{22} & r_{23} & t_y\\ r_{31} & r_{32} & r_{33} & t_z\end{bmatrix}\begin{bmatrix}x\\ y\\ z\\ 1\end{bmatrix}$$

This is not the camera center C. It is –RC (because a point will be rotated before t_x, t_y, and t_z are added)

So we need $-R^{-1} K^{-1} m_4$ to get C

$$\begin{bmatrix}su\\ sv\\ s\end{bmatrix} = \begin{bmatrix}* & * & * & *\\ * & * & * & *\\ * & * & * & *\end{bmatrix}\begin{bmatrix}X\\ Y\\ Z\\ 1\end{bmatrix}$$

Q

This is **t** * K

So K^{-1} m_4 is **t**

Q is K * R. So we just need $-Q^{-1}$ m_4

SUMMARY

- Projection matrix
- What is calibration.
- How to get P from 2D - 3D Correspondences
- How to get camera position and orientation from P