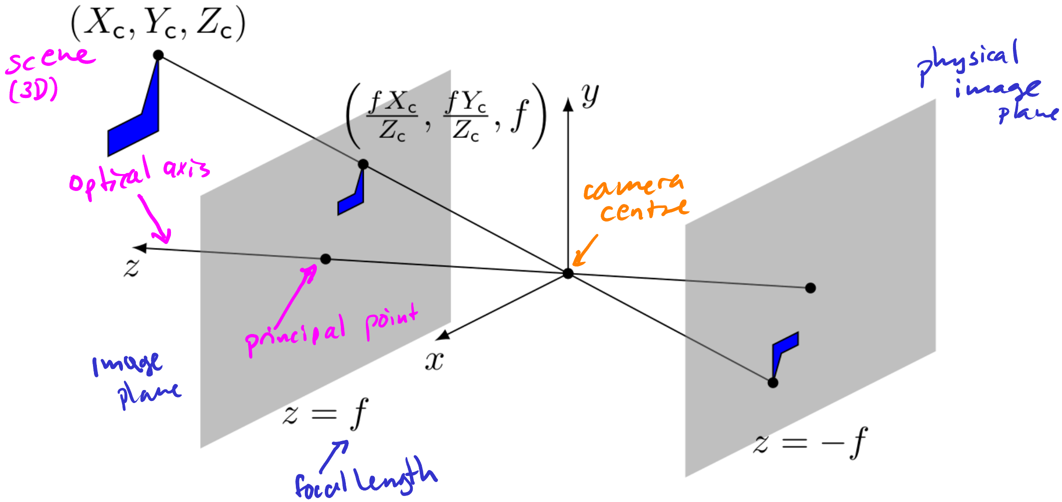
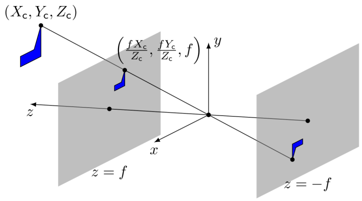
The pinhole perspective camera



Linear mapping of hom. coords.



- Idealised/simplified description of image formation
- Maps (X_c, Y_c, Z_c) to $\left(\frac{fX_c}{Z_c}, \frac{fY_c}{Z_c}, \cancel{f}\right)$ superfluous!
- In homogeneous coordinates:

$$\begin{pmatrix} \frac{fX_c}{Z_c} \\ \frac{fY_c}{Z_c} \\ 1 \end{pmatrix} \sim \begin{pmatrix} fX_c \\ fY_c \\ Z_c \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}$$

Pixel coords. & intrinsic calibration

- Pixel coordinates:

$$\begin{cases} x = \dfrac{f_x X_c}{Z_c} + c_x \\ y = \dfrac{f_y Y_c}{Z_c} + c_y \end{cases} \quad \text{with} \quad \begin{cases} f_x = \sigma_x f \\ f_y = \sigma_y f \end{cases} \quad \text{usually } \sigma_x \approx \sigma_y$$

coordinates of the principal point

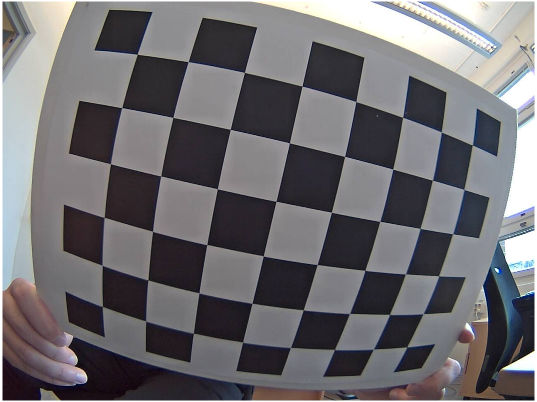
individual scaling if pixels are not square

- Now the projection becomes

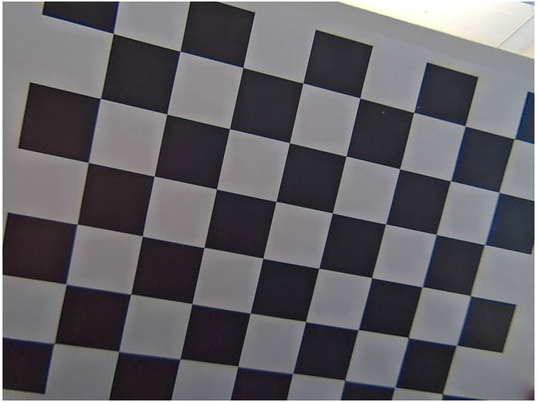
"skew" sometimes nonzero

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} f_x X_c + c_x Z_c \\ f_y Y_c + c_y Z_c \\ Z_c \end{pmatrix} = \underbrace{\begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}}_{\substack{\text{intrinsic calibration matrix} \\ K}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = K \left(\overset{3\times 3}{I} \;\; \overset{3\times 1}{0} \right) \mathbf{X}_c$$

Lens distortion



- Lens distortion
- Not pinhole perspective



- No lens distortion
- Pinhole perspective

"Camera calibration"

- Usually means determining intrinsic parameters together with a lens distortion model (e.g. the Brown-Conrady model)

- Good facilities in OpenCV (also exists in Matlab), based on calibration images (chequerboard)

- If the camera settings are left unchanged, the camera calibration is typically also unchanged

World coords. & extrinsic calibration

- World coordinates:

$$\mathbf{X}_c \sim \underset{\substack{\uparrow \\ \text{rigid} \\ \text{transform}}}{T}\mathbf{X} = \begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

- Now the projection becomes

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim K(I\ \ 0)\mathbf{X}_c \sim K\underline{(I\ \ 0)\begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \underset{\text{intrinsic parameters}}{K}\ \overset{\text{extrinsic parameters}}{\underset{3\times 4}{(R\ \ t)}}\mathbf{X}$$

Camera centre & DoF

- The only point which does not produce a well defined image is the camera centre. The only invalid 3-vector is (0,0,0). Conclusion:

The camera centre n is given by the null space of C, i.e. $Cn = 0$.

- In the most general case, K has five DoF. Together with three in R and three in t, this adds up to eleven DoF for a general C.

$$C = \underbrace{\begin{pmatrix} f_x & s & c_x \\ & f_y & c_y \\ & & 1 \end{pmatrix}}_{K} \begin{pmatrix} R & t \end{pmatrix}$$

- Any 3x4 matrix with rank three represents a camera!

QR and RQ decompositions

$Ax=b \Leftrightarrow QRx=b \Leftrightarrow$
$Rx=Q^Tb$

Any m by n matrix A may be decomposed as A = QR, where Q is orthogonal and R is upper triangular.

- In Matlab: [Q,R] = qr(A)

$A=$ 

Lesser known: RQ decomposition

- suppose $\tilde{A}=\tilde{Q}\tilde{R}$, then $\tilde{A}^T=\tilde{R}^T\tilde{Q}^T$

not upper triangular!

$\tilde{A}^T=$ 

- Let $A=\mathrm{flip}(\tilde{A}^T)$, then $A=\mathrm{flip}(\tilde{R}^T)\,\mathrm{flip}(\tilde{Q}^T) = RQ$

reverse order of all rows and all columns

Camera resectioning

The camera resectioning problem:

Given a camera projection matrix C, decompose it into its extrisic and and extrinsic parameters,

$$C = K(R \;\; t)$$

One solution: Use RQ decomposition!

If $C = (A \;\; b)$, RQ-decompose $A = KR$

$$\Rightarrow C = (A \;\; b) = (KR \;\; b) = K(R \;\; \underbrace{K^{-1}b}_{=t}) = K(R \;\; t).$$

(No guarantees, e.g. the focal lengths may become negative, etc.)

Kronecker & vec

See the TeX-ed version instead!