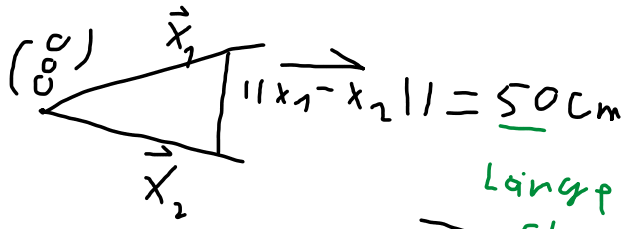
$$\vec{x}_1 = z \cdot K^{-1} \begin{pmatrix} z_1 \\ z_2 \\ 1 \end{pmatrix}$$

$$\vec{x}_2 = z \cdot K^{-1} \begin{pmatrix} z_1 \\ z_2 \\ 1 \end{pmatrix}$$



Langer Schachbrett

Pixel links

Pixel rechts

$$50 = \| z \cdot \underbrace{K^{-1}} \begin{pmatrix} z_{11} \\ z_{12} \\ 1 \end{pmatrix} - z \cdot \underbrace{K^{-1}} \begin{pmatrix} z_{21} \\ z_{22} \\ 1 \end{pmatrix} \|$$

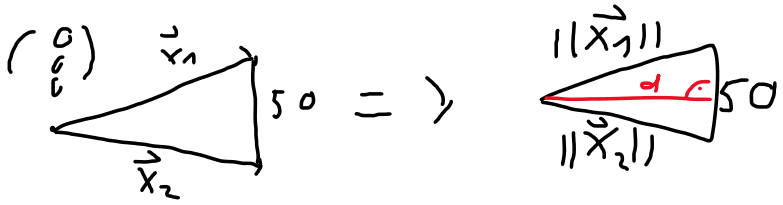
Kamera Matrix

$$\Rightarrow 50 = \| z \cdot \vec{v}_1 - z \cdot \vec{v}_2 \|$$

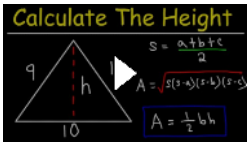
$$50 = \| z \cdot (\vec{v}_1 - \vec{v}_2) \|$$

⇒ bekommen wir z

⇒ $\vec{x}_1$ und $\vec{x}_2$ voll definiert



How To Calculate The Height of a Triangle Using Heron's Formula



$$A = \frac{1}{2} \cdot 50 \cdot d$$

$$\frac{2A}{50} = d$$

$$s = \frac{\|\vec{x}_1\| + 50 + \|\vec{x}_2\|}{2}$$

$$A = \sqrt{s(s - \|\vec{x}_1\|)(s - 50)(s - \|\vec{x}_2\|)}$$

⇒ We have p

=> We have everything for d

How to get CameraMatrix = K

https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

Für Part1 hat Felix Berens jeden 3. Frame verwendet