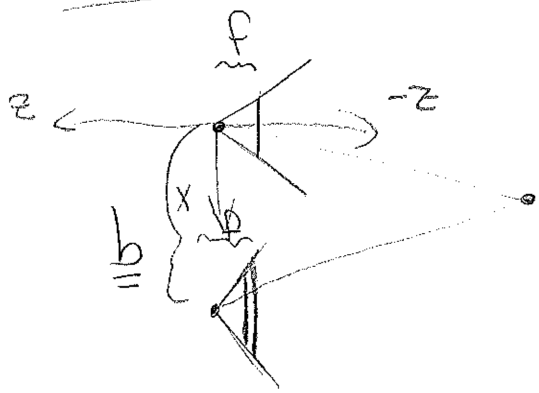
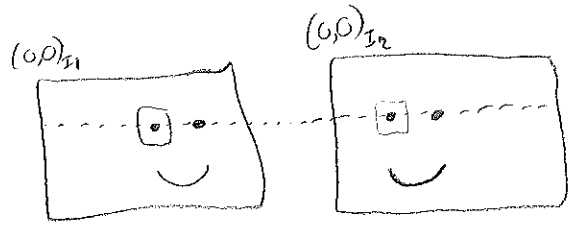
Rectified Stereo Cameras

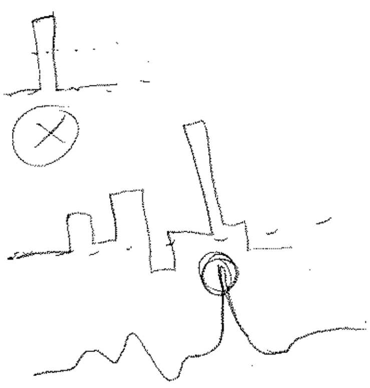




If we can find correspondence from I_1 to I_2, we can calculate depth.
This is the fundamental problem in stereo vision; most other problems (rectification, depth from disparity, scaling, choosing baselines) have ~~relatively~~ reasonably good solutions.

Matching - Local methods

- Consider a range of possible disparities, compare windows "photoconsistency".
- Many metrics:



  - SSD
  - (N)CC → convolve R img w/ left patch - large product when equal.
    - ↳ NCC → normalize patches (subtract μ, divide by σ) before multiplication to handle photometric change
  - SAD

Cost Volume

For each row,
  for each column
    for each disparity:
      compute C(patch1, patch2)

C = np.array(h, w, d) ← # possible disparities

disp = np.max(C, axis=2)

Intrinsics

1A

$$\begin{bmatrix} x \\ y \\ -z/f \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

intrinsics ↙ ↘ projection

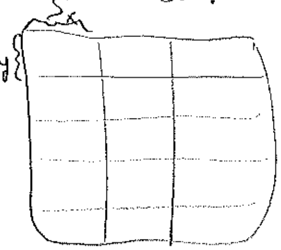
K

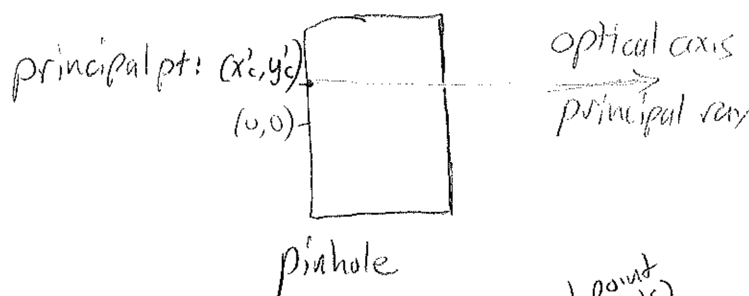
$$\begin{bmatrix} f & 0 & c_x \\ 0 & -\alpha f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

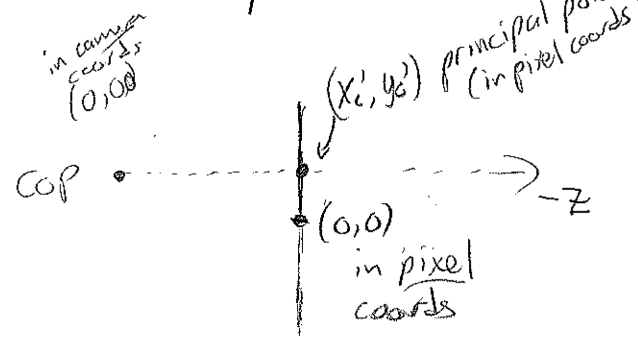
- f is focal length
- α = aspect ratio
- C_x, C_y = principal point

Because K is scale invariant, important # is $\alpha = \frac{s_x}{s_y}$ = aspect ratio.

Sensor:



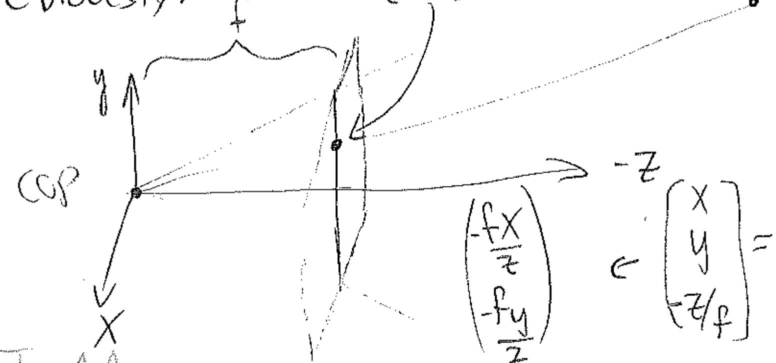




Cameras: Intrinsics, Extrinsics

Previously:

$$\begin{bmatrix} x \\ y \\ -z/f \end{bmatrix} \qquad \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$ <u>scene</u> coordinates or <u>camera</u> coordinates



$$\begin{pmatrix} -\frac{fx}{z} \\ -\frac{fy}{z} \end{pmatrix} \Leftarrow \begin{bmatrix} x \\ y \\ -z/f \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

To 1A

Extrinsics

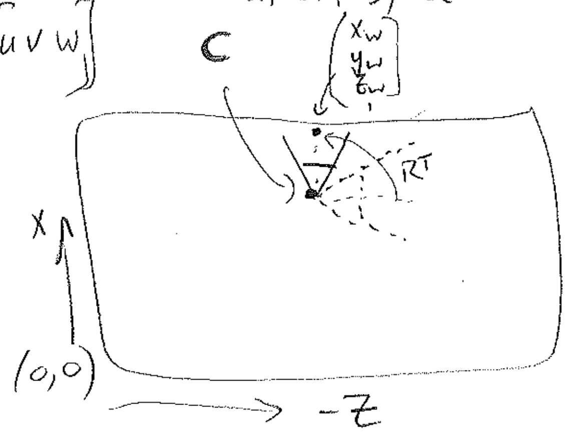
What if we have multiple cameras, or the world lives in its own coordinate system?

<u>world</u> coordinates

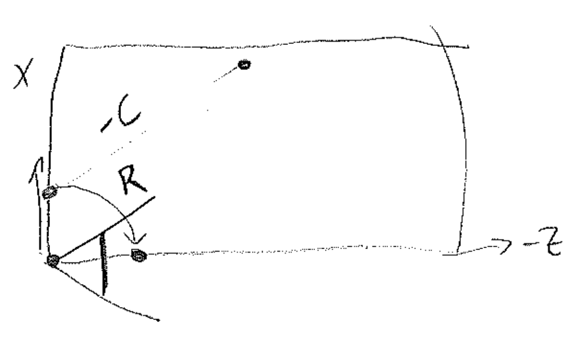
$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} & & \\ & ? & \\ & & \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

what goes here? Transform <u>world</u> points to camera points

Suppose camera is centered at world coordinates C. $\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$
and can be rotated <u>to</u> its world orientation by a 3x3 matrix $R^T = \begin{bmatrix} u & v & w \end{bmatrix}$



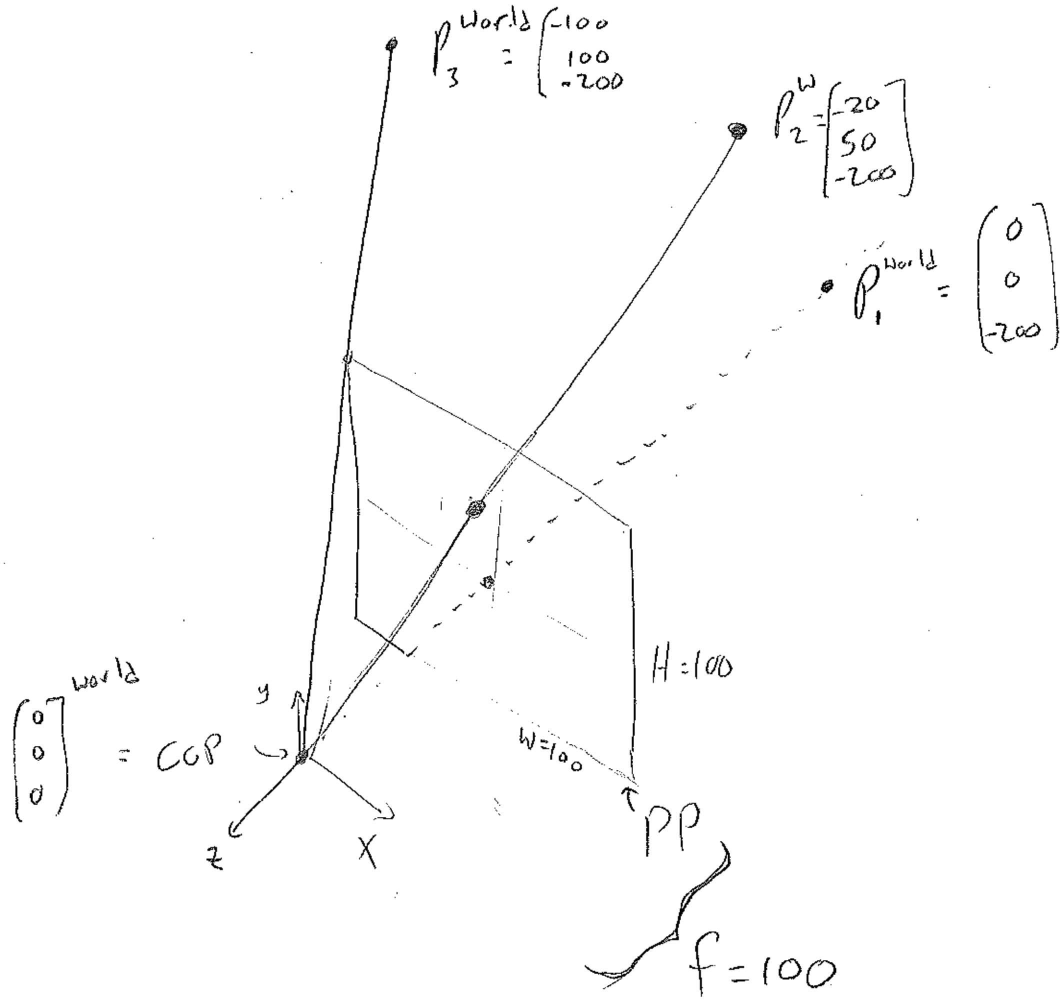
$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_{3\times3} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ 0\ 0\ 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} I_{3\times3} & -C \\ 0\ 0\ 0 & 1 \end{bmatrix}}_{\text{translate}} \underbrace{\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}}_{\text{world}}$$



2

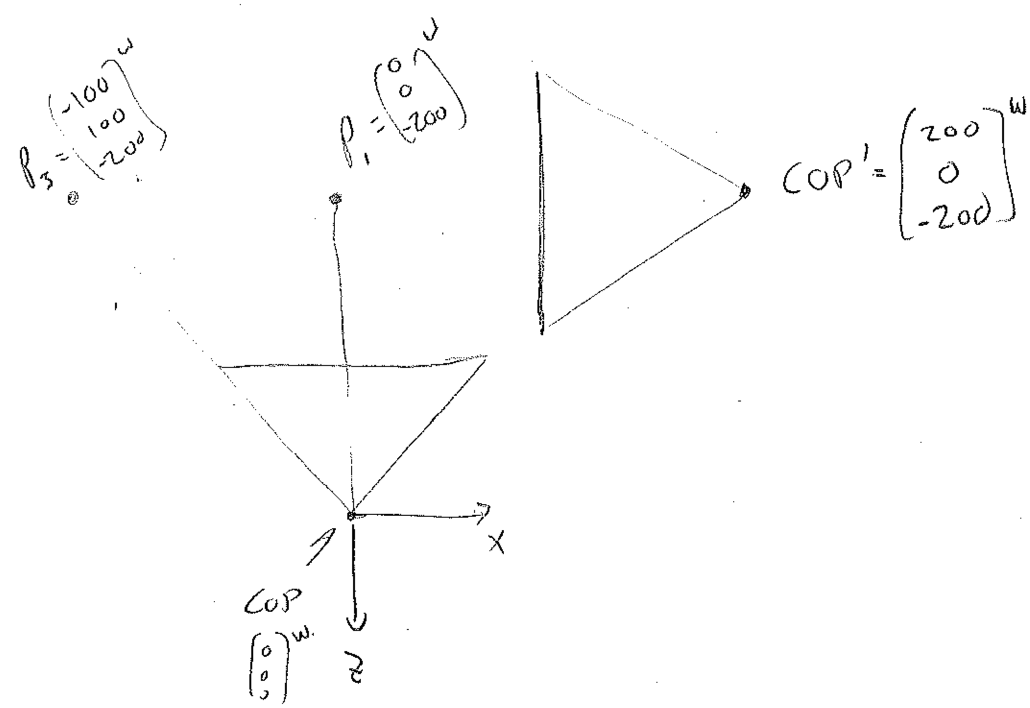
$$\underbrace{\begin{bmatrix} x \\ y \\ -z/f \end{bmatrix}}_{\text{pixel coords}} = \underbrace{\underbrace{\begin{bmatrix} -fs_x & 0 & x_c' \\ 0 & -fs_y & y_c' \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Intrinsics}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{project}} \underbrace{\underbrace{\begin{bmatrix} & R & & 0 \\ & & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{rotate}} \underbrace{\begin{bmatrix} & I_{3\times 3} & & -c \\ & & & \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{translate}}}_{\text{extrinsics}}}_{\Pi} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

camera coords



$p_3^{world} = \begin{bmatrix} -100 \\ 100 \\ -200 \end{bmatrix}$
$p_2^{w} = \begin{bmatrix} -20 \\ 50 \\ -200 \end{bmatrix}$
$p_1^{world} = \begin{bmatrix} 0 \\ 0 \\ -200 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^{world} = COP$
y
z
X
H = 100
W = 100
PP
f = 100

Top-Down



$p_3 = \begin{pmatrix} -100 \\ 100 \\ -200 \end{pmatrix}^{w}$
$p_1 = \begin{pmatrix} 0 \\ 0 \\ -200 \end{pmatrix}^{w}$
$COP' = \begin{bmatrix} 200 \\ 0 \\ -200 \end{bmatrix}^{w}$
X
COP
$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^{w}$
z