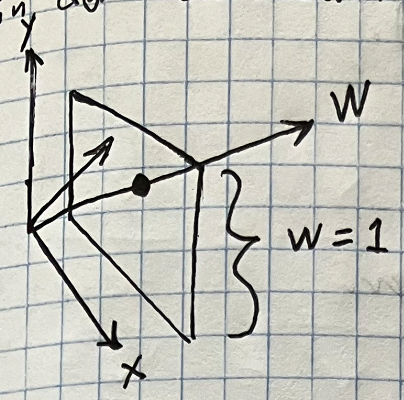
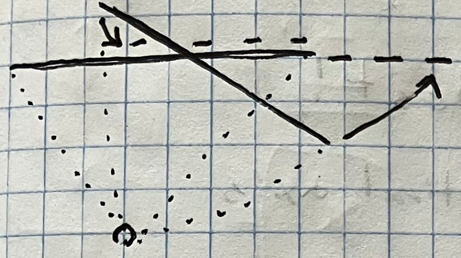
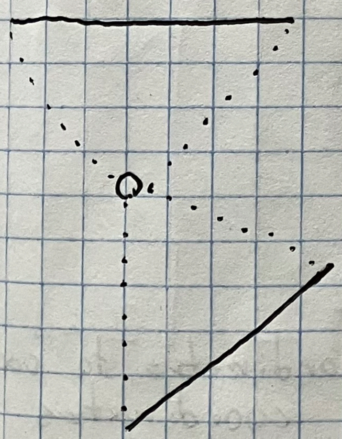
Computer Vision!

Homography intuition: transformation of a 2D plane in an abstract 3D space



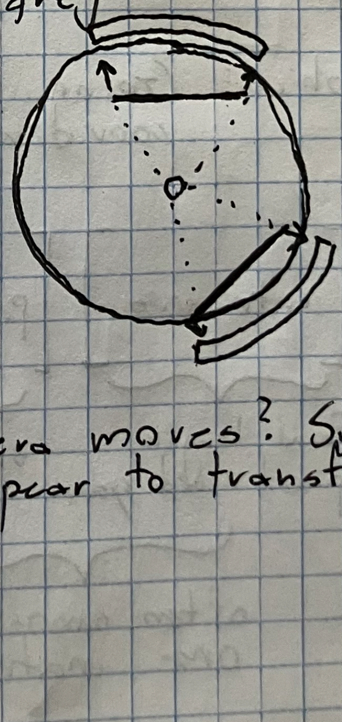


Current method for panorama stitching: project the plane of one image onto another.



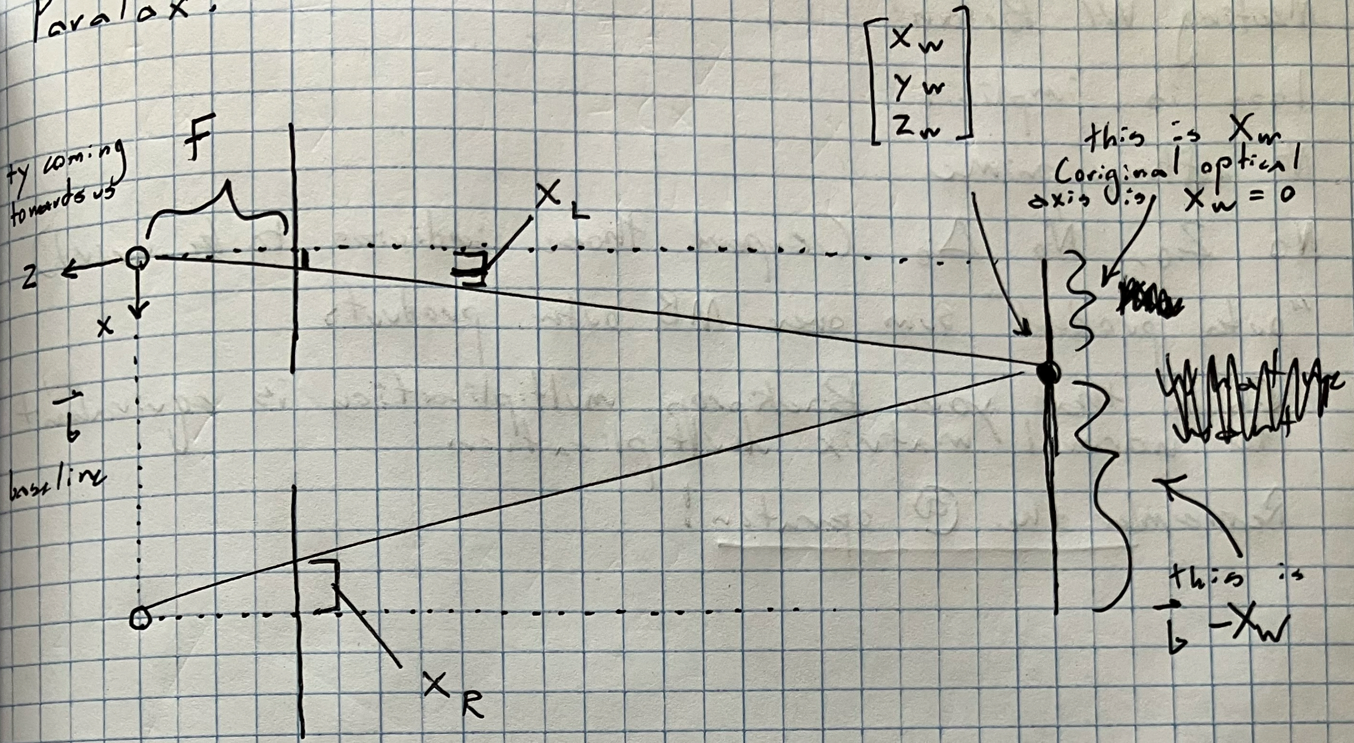
But, what if the angle is unreasonable?

L Instead of projecting to a plane, we project to the inside of a sphere.



Additionally, what if our camera moves? Suddenly, distance matters (things far away appear to transform less than things nearby).

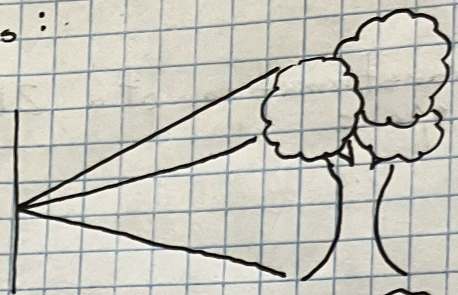
Paralax!



Solving for Z_y:

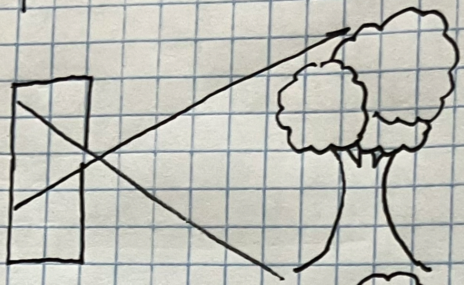
Cameras:

Mk I



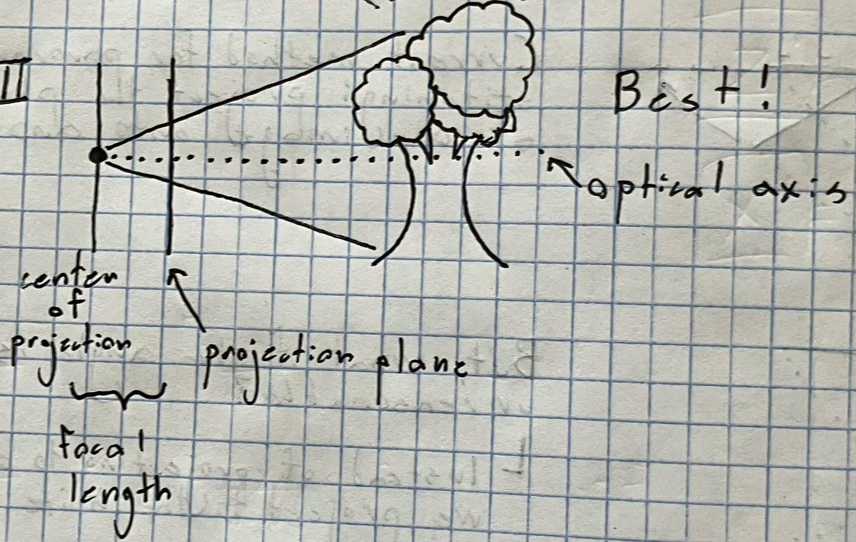
Bad.

Mk II



Better.

Mk III



Best!

Talkin' Graphics: Going from world coordinates to camera coordinates to image coordinates

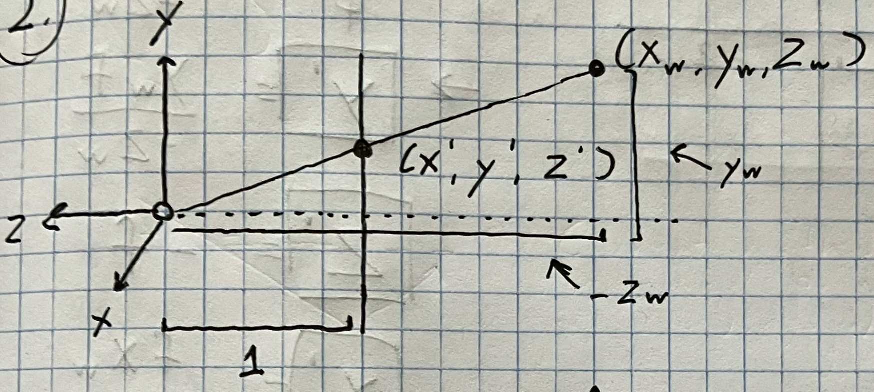
$$X_{img} = \underbrace{\underbrace{\left[\ \text{Intrinsics}\ \right]}_{\substack{\text{Focal length,}\\ \text{pixel coord system}}} \underbrace{\left[\ \text{Projection}\ \right]}_{\text{3D to 2D}}}_{\substack{\text{often smooshed into}\\ \text{one matrix}}} \overset{\uparrow}{\underset{X_{img}}{}} \underbrace{\left[\ \text{Extrinsics}\ \right]}_{\text{camera pos}} X_{world}$$

1. The MkIII will no longer be upsidedown, for one.
   ↳ Does this mean $y^{MkII} = -y^{MkIII}$?

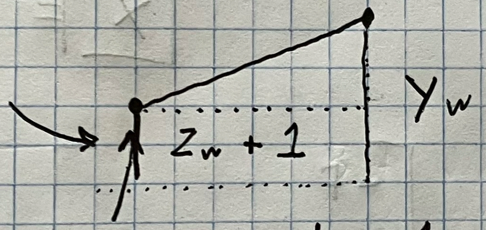
$$(x^{MkII}, y^{MkIII}) = -(x^{MkIII}, y^{MkIII})$$

It's flipped for both (think 3D)

2.



$$y' = \frac{y_w}{-z_w}$$



So this is $z' = -1$

Finally, $x' = \dfrac{x_w}{-z_w}$

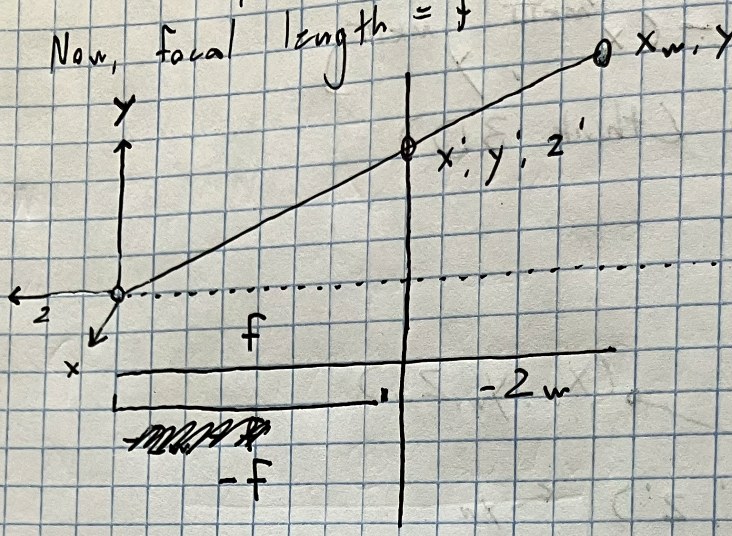
3.

Now, for matrices

$$\begin{bmatrix} \frac{x_w}{z_w} \\ \frac{y_w}{z_w} \\ 1 \end{bmatrix} \sim \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$$

This identity matrix thing only works for focal len 1

Now, focal length = f



$$\frac{y'}{f} = \frac{Y_w}{-Z_w}$$

$$\rightarrow \boxed{y'} = \frac{Y_w f}{-Z_w}$$

$$\boxed{Z'} = -f$$

$$\boxed{x'} = \frac{f X_w}{-Z_w}$$

$$\begin{bmatrix} \frac{-fX_w}{Z_w} \\ \frac{-fY_w}{Z_w} \\ 1 \end{bmatrix} \sim \begin{bmatrix} -fX_w \\ -fY_w \\ Z_w \end{bmatrix} = \begin{bmatrix} -f & & \\ & -f & \\ & & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$$

(I mixed these up originally)

Implied: transformation to $w = 1$, which sort of implicitly resets our f to 1 via the $\sim$ normalization step