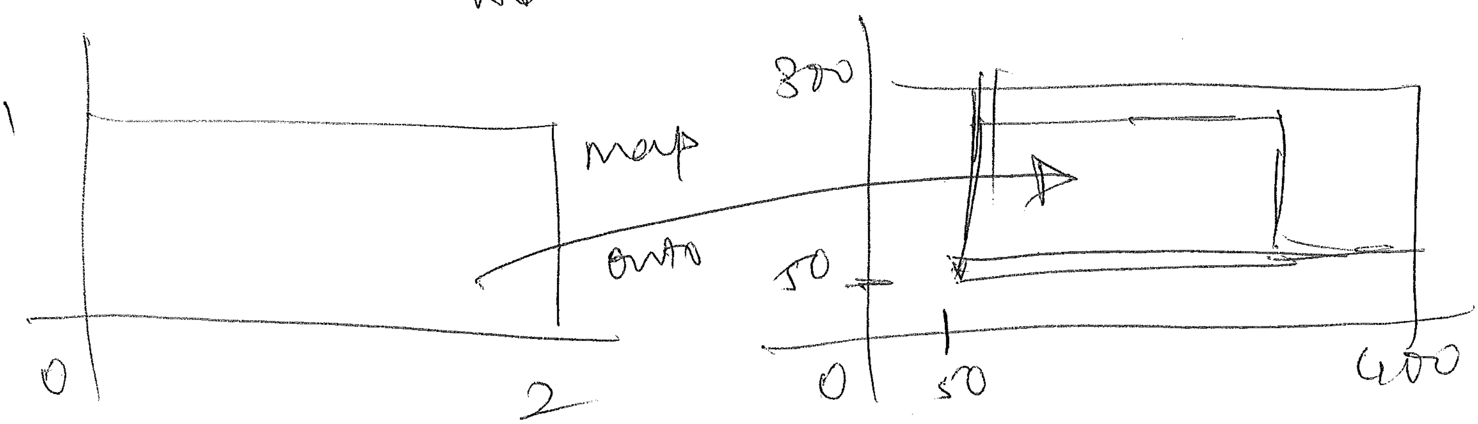
$(x, y) = (X, Y) +$ mapping + "shift in origin" ①

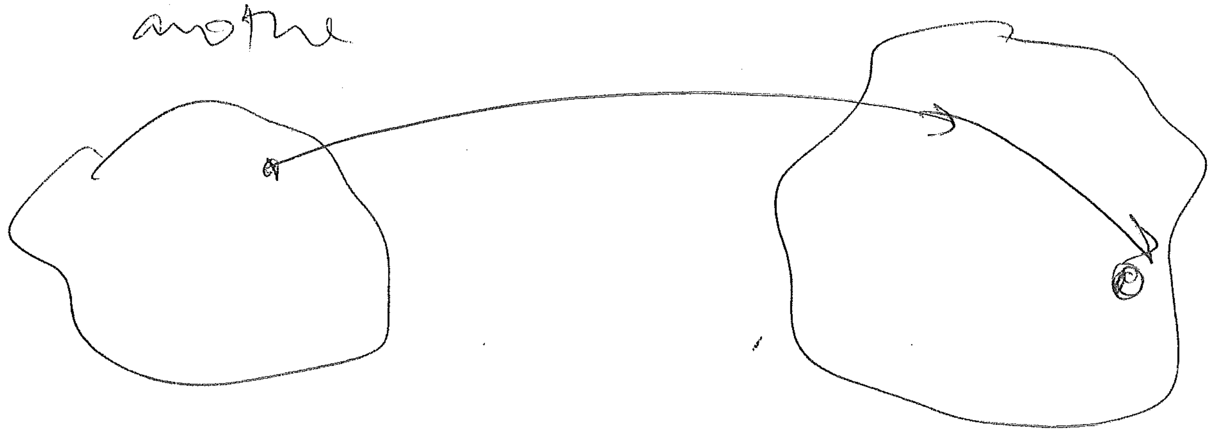
→ In mouse clicks, we get the "inverse problem"



↳ viewports can now be used for zooming, clipping too?!

~~~~ What is linearity (atleast for vectors)

1:1 mapping from one space to another



[maintains (one?) vector addition & scalar multiplication]
~~~~

→ non linear maps $\left(\begin{array}{l} x \rightarrow x^2 \text{ or} \\ x \rightarrow x+1 \text{ (Real } H\text{'s)} \end{array} \right)$

Affine transformation

$$x \longrightarrow Ax + b$$

(linear transform + translation)

1. colinearity between points maintained
2. Ratio's of distance is preserved

(linear transforms :- (Rotations, scaling, shear + translation)

— Can be combined.

Homogenous coords

2

- Help calculations in projective space.

i.e. plane at $\infty = (x, y, z, \cancel{w}) \to 0$

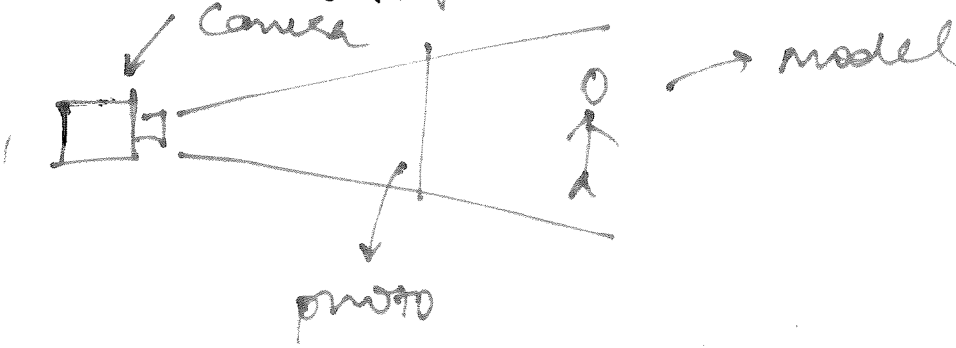
~~w for any~~ $w = 1$, gives cartesian co-ordinates.

- represent ~~matrix~~ projection & translation as matrix operation.
- How does a projective operation remain an affine transform.
  - Incidence & cross ratio's maintained!

$$\begin{bmatrix} R & T \\ 0\;0\;0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}$$

The Camera analogy



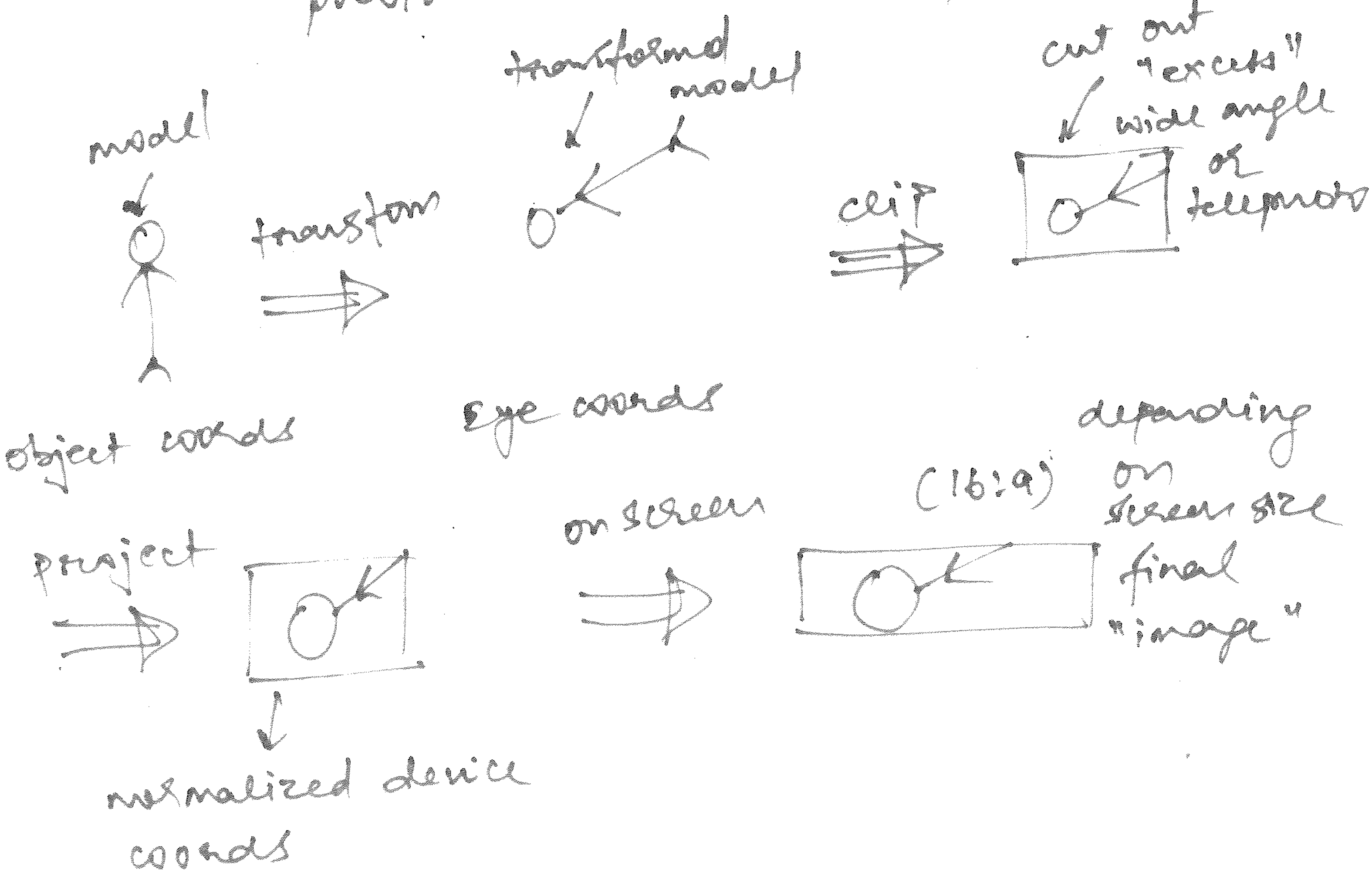
Analogy

camera ⟶ viewing

world ⟶ model, geometric objects

lens ⟶ projection

photo ⟶ viewport

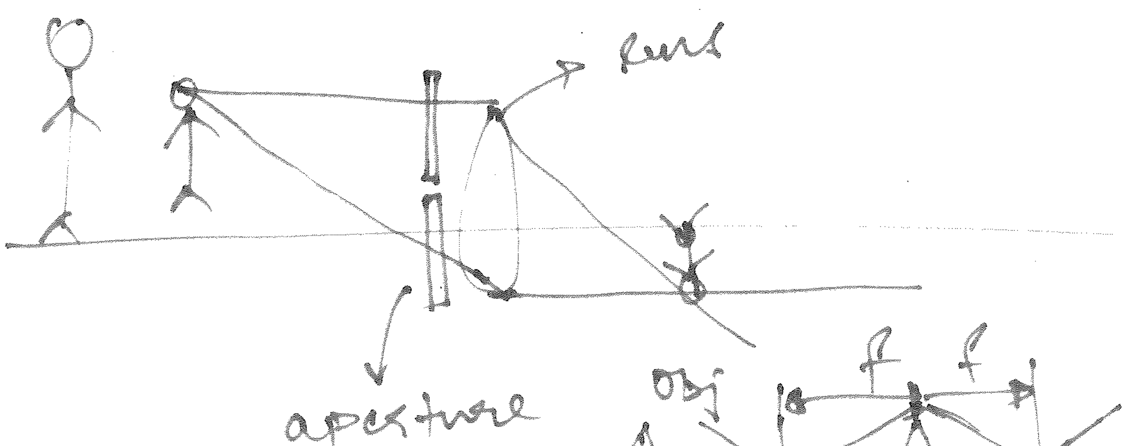


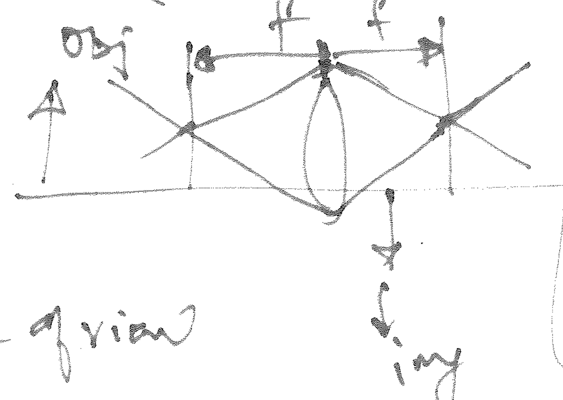
~~What are the properties~~ of a camera?
(relevant)

OGL does 2 things using the "projection matrix"

- clip out "unnecessary" stuff from scene
- Perform projection operations

☆ Equivalent to a camera?





point till which "object" converges ~~before~~ diverging again!

- focal length?
- field of view | angle of view

binocular
(depth perception!)
140°

peripheral

$$\alpha = 2 \tan^{-1}\left(\frac{d}{2f}\right)$$

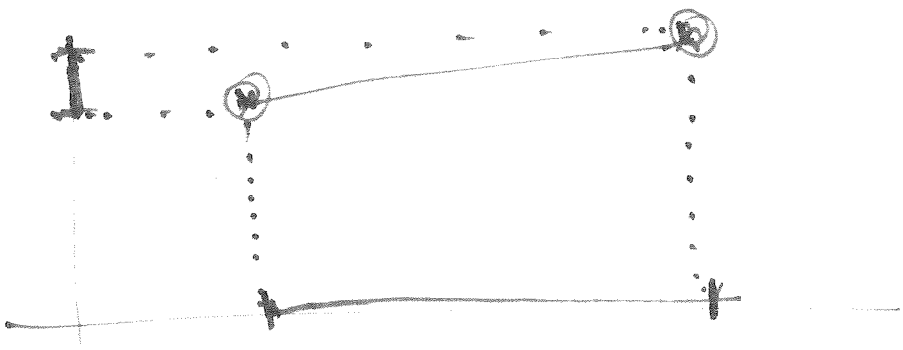
d is max image size

2 options for camera placement

→ fix camera & move objects

→ or move objects & leave camera in place { pitfalls? }

~ projections



– projection can now be represented as

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \text{ or } \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

in 3d matrix would look like.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Can also be done at an "angle" { oblique }

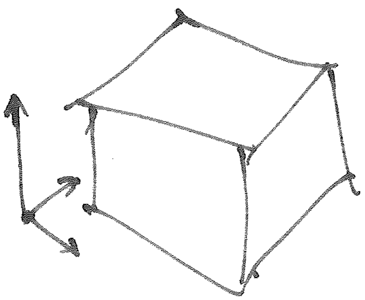
$$\begin{bmatrix} 0 & 0 \\ \alpha & 1 \end{bmatrix} \Rightarrow \text{try the math}$$

- Isometric view

8 diff views, rotation abt

2 axis

→ apparent angle between axis = 120°



$$\left[R(\theta_1, x) \right] \left[R(\theta_2, y) \right]$$

$$\theta_1 \simeq 35^\circ$$

$$\theta_2 \simeq 45^\circ$$

- popular in 2D games representing 3D

☆ As objects have a "square" footprint can accomodate images representing 3D easily

- perspective

→ projected lines are not "||el" but converge at a point

→ Special case of orthographic projection (convergence at ∞)

matrix itself looks like

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \quad 1/d \rightarrow w!$$

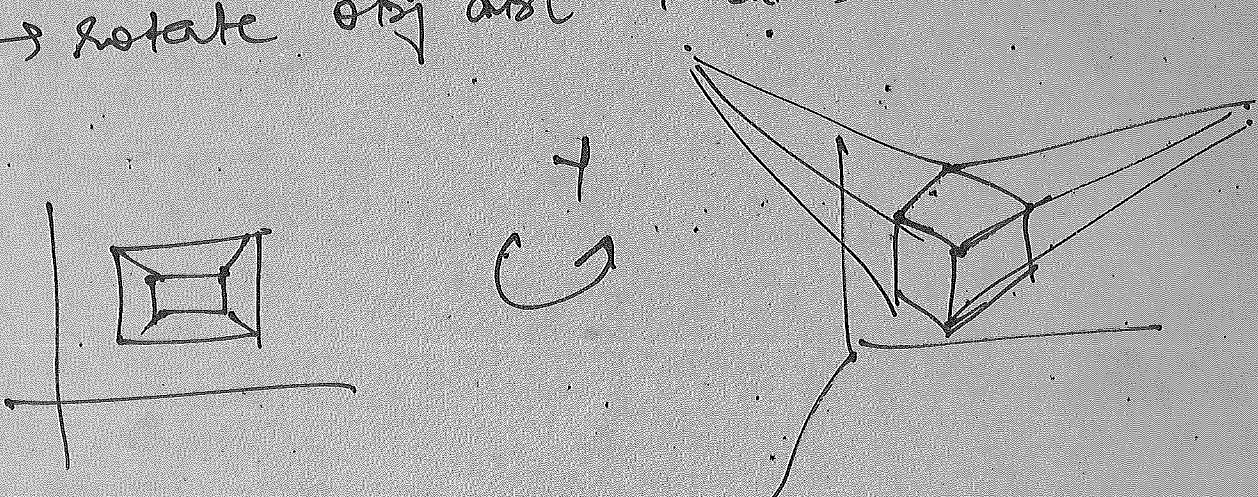
$$\begin{bmatrix} x \\ y \\ z \\ z/d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$z/d \rightarrow w$

vanishing point

\# principal axis intersecting projection plane

→ rotate obj abt y axis



$$P_2 = [R_{y,\theta}] * P * [R_{y,-\theta}]$$

$$P_2 = \left[\begin{matrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1/d \end{matrix}\;\begin{matrix} s(\theta)/d \\ 0 \\ c\theta/d \\ 0 \end{matrix}\right]$$

whats the net effect

think of it as a line with slope

→ 1/d, what happens when $d \to 0$?

→ All points emanate from vertex & go to a "common viewpoint"