

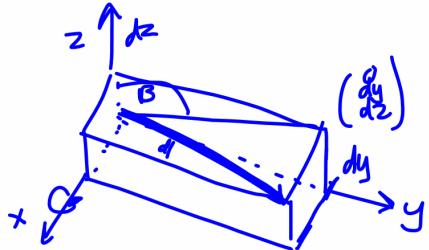
② is 'difficult'

||  
|| Reduce to 2 rotations ||

around coord.-system axis

First: Rotate  $dl$  into  $xz$  plane, by rotating

around  $x$ -axis using angle  $\beta$



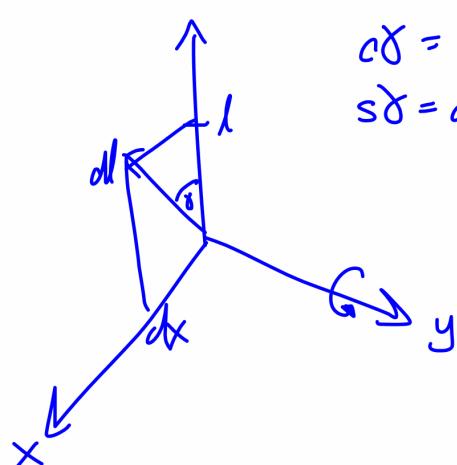
$$c\beta = \frac{dz}{dl}$$

$$s\beta = \frac{dy}{dl} \Rightarrow \text{Rot } X(\beta)$$

$$l = \sqrt{dy^2 + dz^2}$$

$$\Rightarrow \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{dy}{dl} & \frac{dz}{dl} & 0 \\ 0 & \frac{dy}{dl} & \frac{dz}{dl} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Second: Rotate intermediate vector  
into  $z$ -axis /  $lb_z$  vector



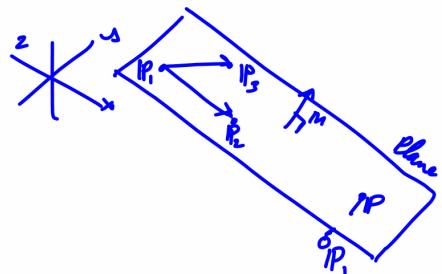
$$c\gamma = l \Rightarrow c(-\gamma) = l$$

$$s\gamma = dx \Rightarrow s(-\gamma) = -dx$$

$$= \text{Rot } Y(\gamma) = \begin{pmatrix} l & 0 & -dx & 0 \\ 0 & 1 & 0 & 0 \\ dx & 0 & l & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \boxed{M_2 = \text{Rot } Y(\gamma) \cdot \text{Rot } X(\beta)}$$

Ex: Refl. wrt. General Plane

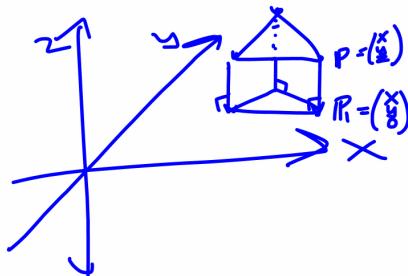


$$\begin{aligned} \mathbf{n} &= (\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_3 - \mathbf{P}_1) \\ &\uparrow \text{cross product} \\ &\text{length } 1 \end{aligned}$$

$\Rightarrow$  Reduce to Refl XY:

- ① Transl.  $(-\mathbf{P}_1)$
- ② Rotate M into  $\mathbb{R}^3$
- ③ refl XY
- ④ "Inverse" of ②
- ⑤ "Inverse" of ①

Example: Axonometric Proj.

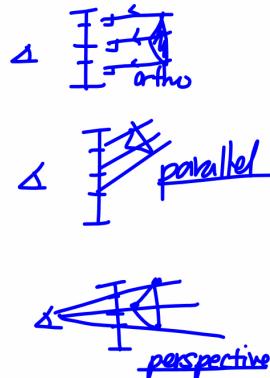


$$\text{Proj } YZ = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

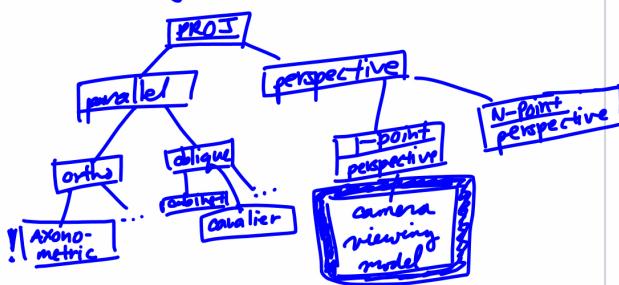
$$\text{Proj } XZ = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\text{Proj } XY = \begin{pmatrix} 1 & 0 & 0 & | & 0 \\ 0 & 1 & 0 & | & 0 \\ 0 & 0 & 1 & | & 0 \\ \hline 0 & 0 & 0 & | & 1 \end{pmatrix}$$

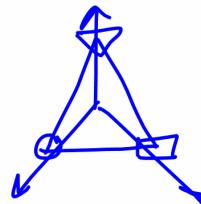
### Simple Projections



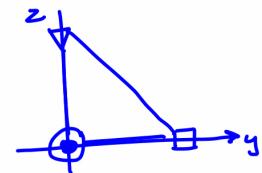
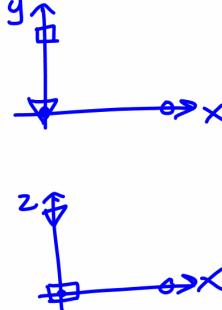
### Projections



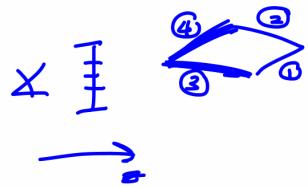
### 3 Axonometric Projection of a tetrahedron



### 3 projections/views



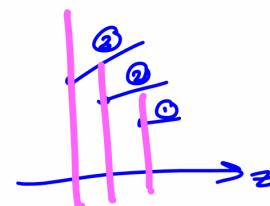
### Hidden Surface Algorithm: "Painters Alg."



- i) sort all polygons (= edges in picture)
- ii) back-to-front drawing

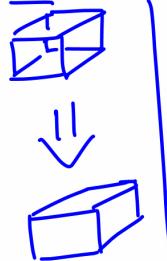
### Needed: Definition of Depth, of Edge/Polygon:

⇒ Depth of an edge/poly is minimal value of the z-coordinate of all its vertices



Algo. works for  
|| Planar, non-intersecting  
|| Polygons only

EX:



Rendering according to "Distance"

