More transformations

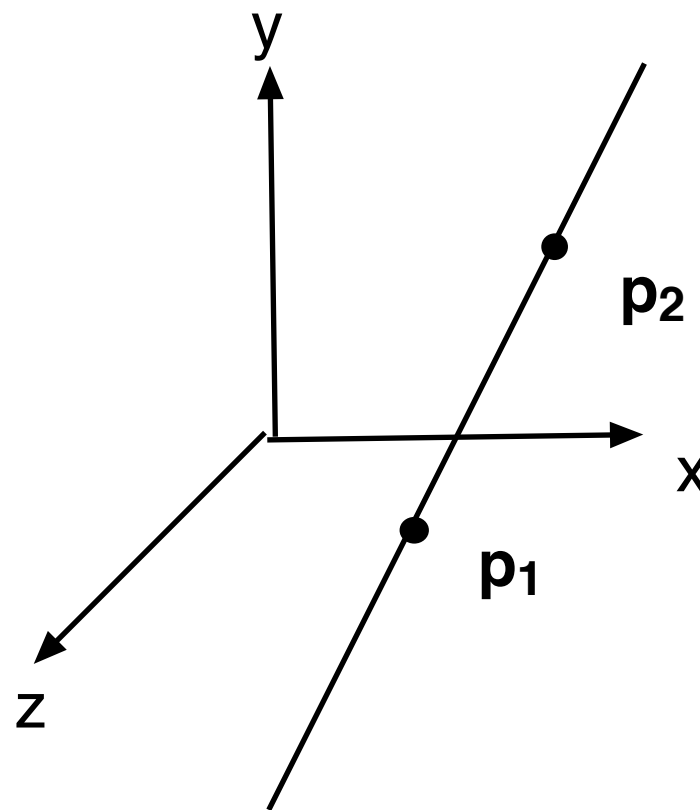# Rotation around arbitrary axis

Rotation around arbitrary axis
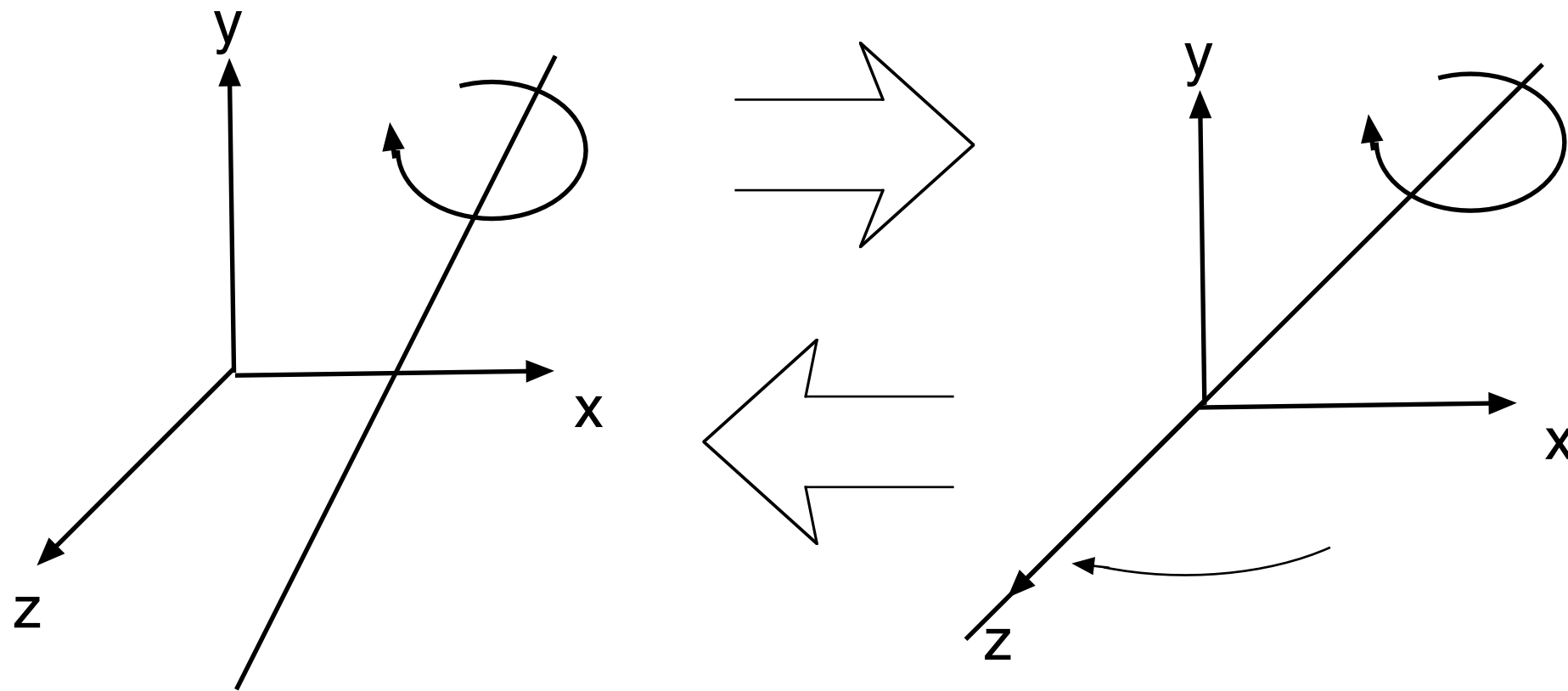
Transform to align axis with the Z axis, rotate, and transform back.
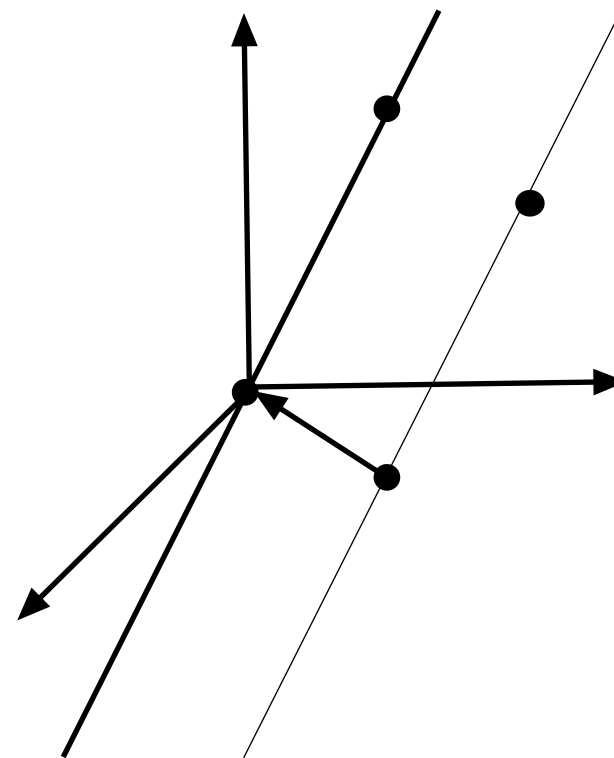
Definition of the rotation axis
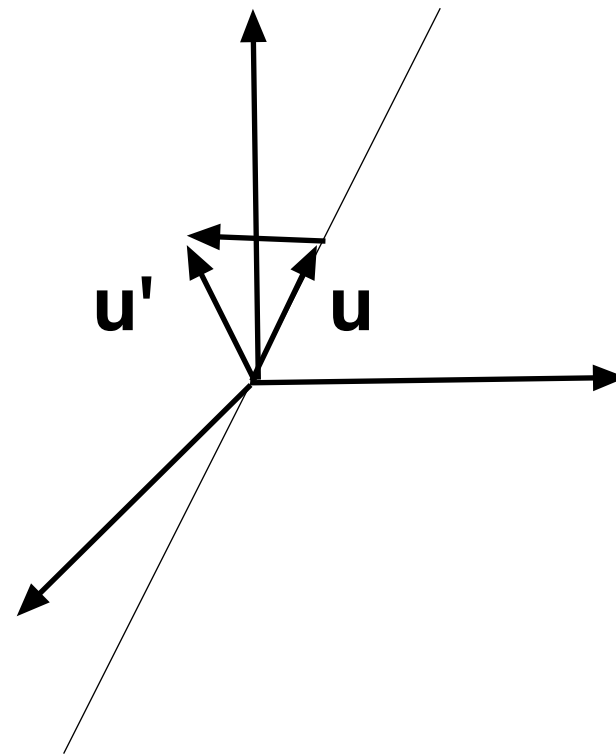

p_1 and p_2 define the rotation axis

Translate to origin
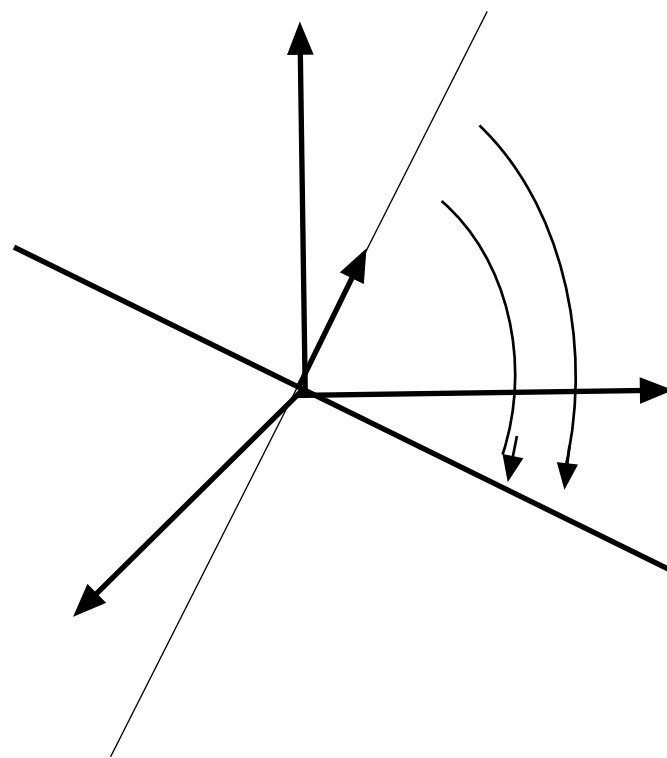

$T(-p_1)$

Geometrical method
Finding an angle to rotate around X

Project u on the yz plane = (0, u_y, u_z)
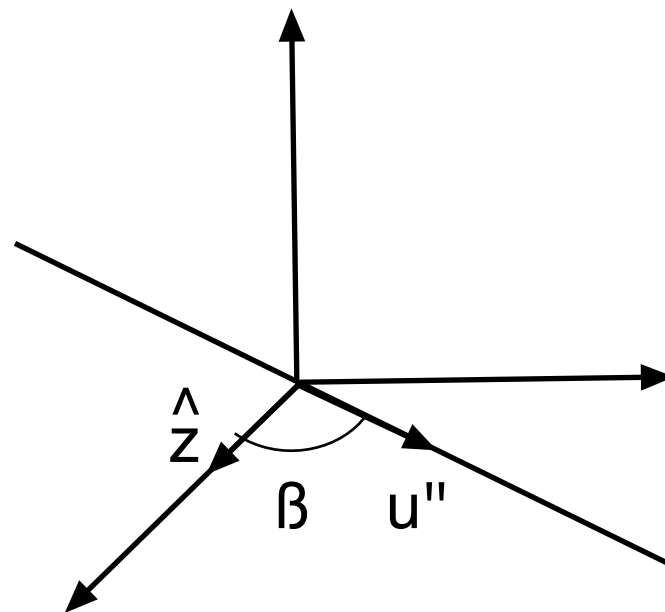
Rotate around X
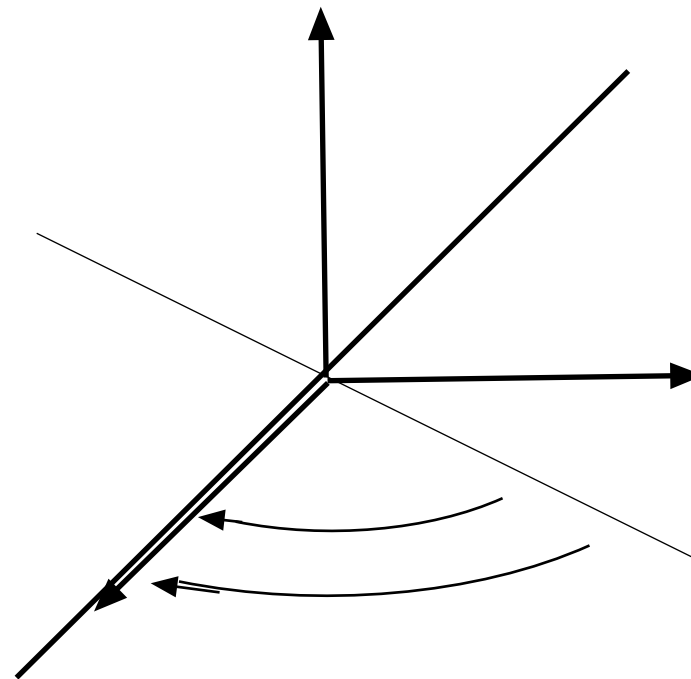

$R_X(\alpha)$

Finding an angle to rotate around Y

u'' and $\hat{z}$ gives the angle ß in the xz plane

Rotate around Y

$$R_y(\beta)$$

Rotation around arbitrary axis, summary:

The axis to rotate around is given as two points, **p**$_1$ and **p**$_2$.

v = **p**$_2$ - **p**$_1$

u = **v** / |**v**| = (u_x, u_y, u_z) Normalized!

d = $\sqrt{u_y^2 + u_z^2}$)

$$\mathbf{R}_X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & u_z/d & -u_y/d & 0 \\ 0 & u_y/d & u_z/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
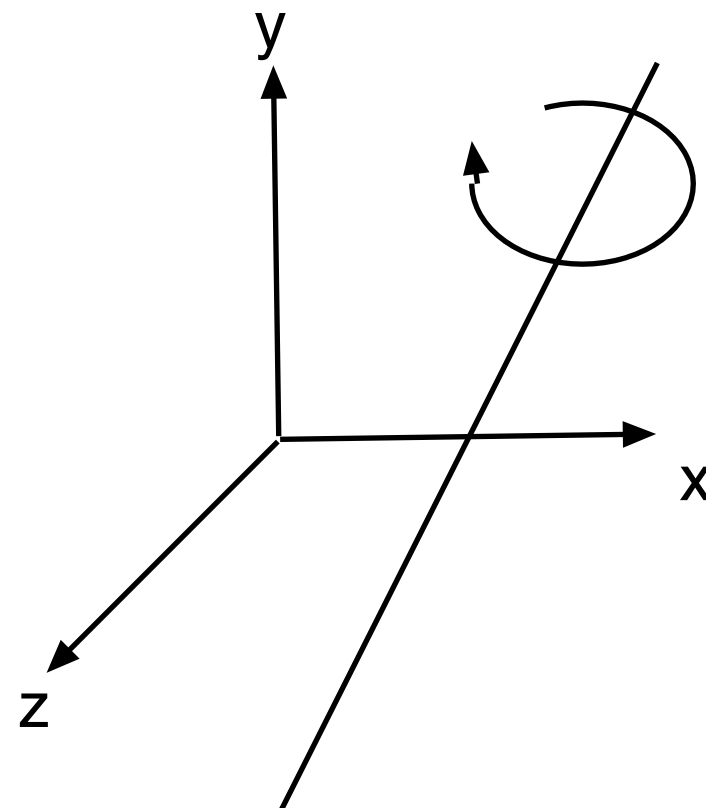
$$\mathbf{R}_y = \begin{bmatrix} d & 0 & -u_x & 0 \\ 0 & 1 & 0 & 0 \\ u_x & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Total transformation:

$$R(\theta) = T(p_1) * R_x{}^T * R_y{}^T * R_z(\theta) * R_y * R_x * T(-p_1)$$

Rotation around arbitrary axis in OpenGL

Create matrices, multiply on CPU, upload to uniform matrices.

Rotation around arbitrary axis, using change of basis:

$$\mathbf{v} = \mathbf{p}_2 - \mathbf{p}_1$$

$$\mathbf{u_z} = \mathbf{u} = \mathbf{v} / |\mathbf{v}| = (u_x, u_y, u_z) \text{ Normalized!}$$

$$\mathbf{u_y} = \mathbf{u} \times (u_x,0,0) / |\mathbf{u} \times (u_x,0,0)|$$

$$\mathbf{u_x} = \mathbf{u_y} \times \mathbf{u_z}$$

$$\mathbf{R} = \begin{bmatrix} u_{x1} & u_{x2} & u_{x3} & 0 \\ u_{y1} & u_{y2} & u_{y3} & 0 \\ u_{z1} & u_{z2} & u_{z3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Total transformation:

$$R(\theta) = T(\mathbf{p}_1) * R^T * R_z(\theta) * R * T(-\mathbf{p}_1)$$

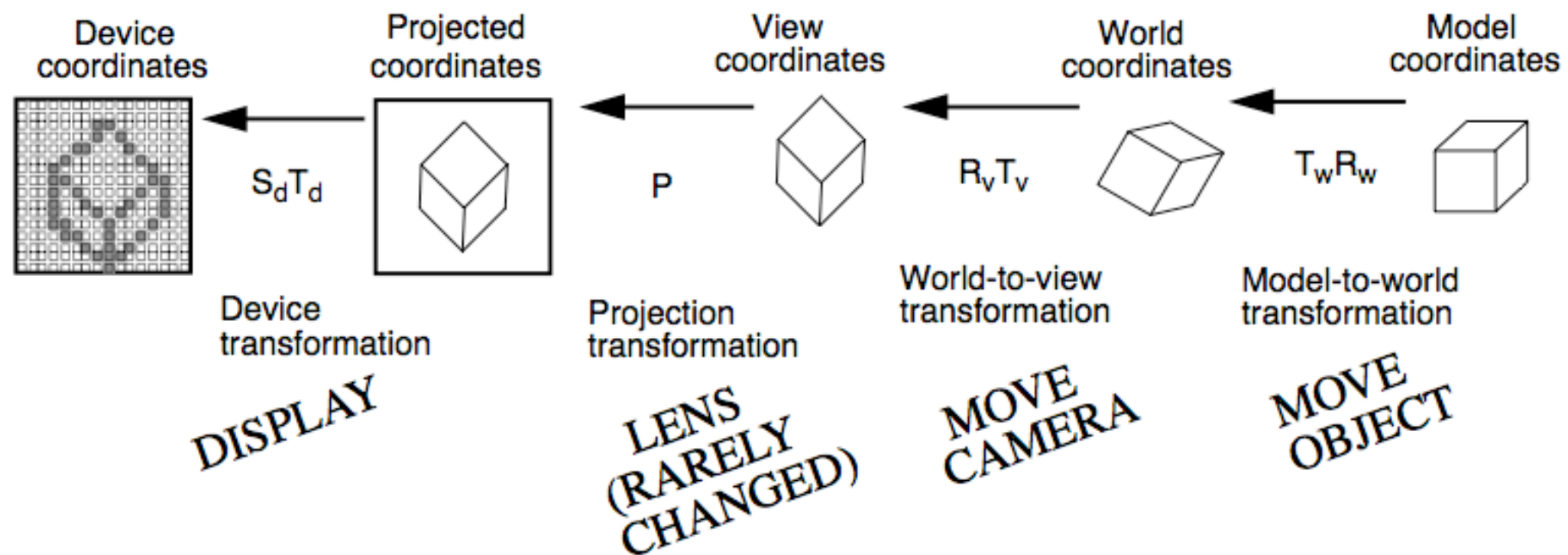# Application of rotation around arbitrary axis: Trackball control

Create rotation from user's mouse input

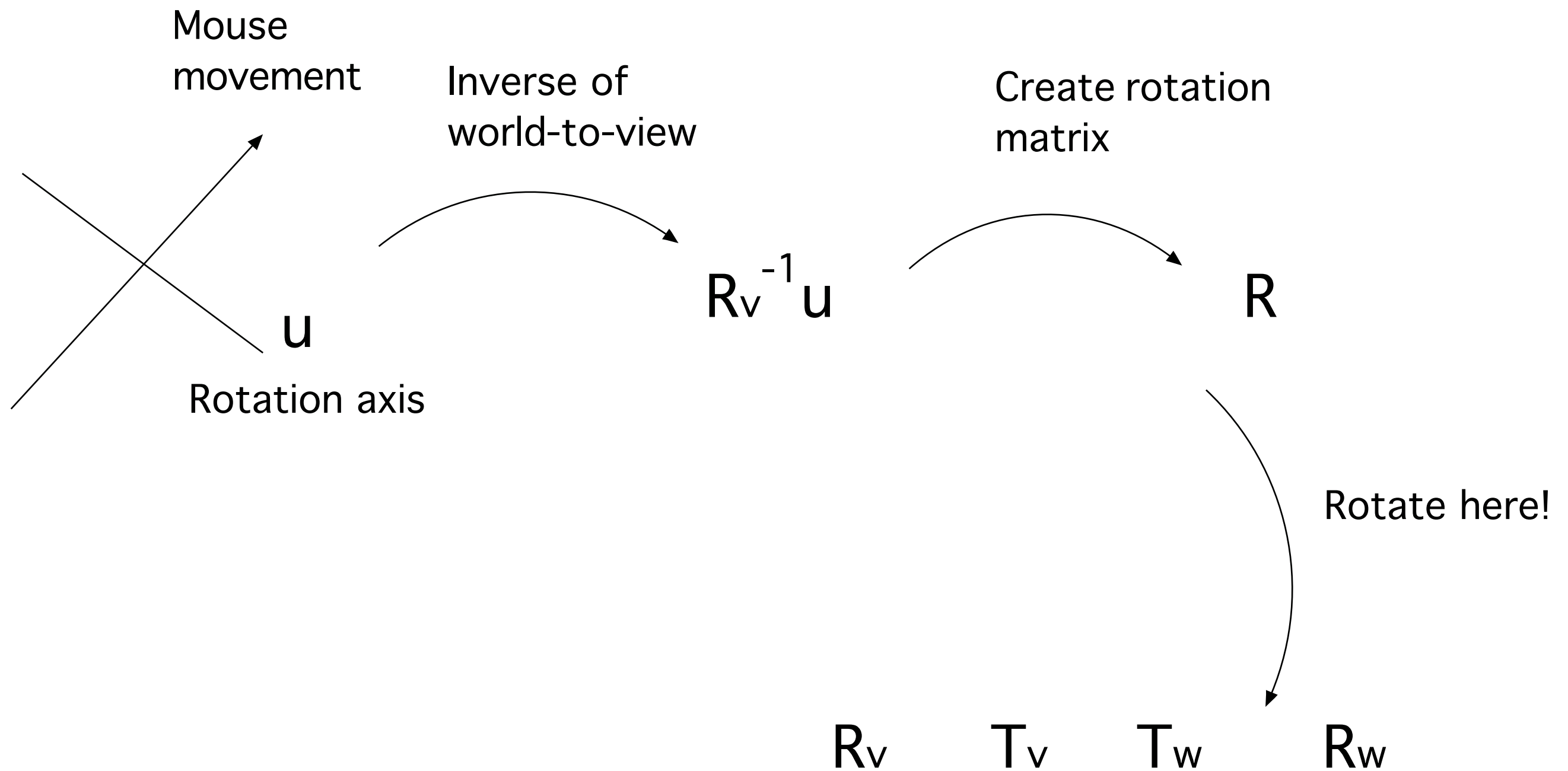## Picking for object selection

Trackball: What coordinate system?

Input on screen -> view coordinates

Rotation of model -> rotate near model

Solution: Transform to model coordinates, rotations only (avoid rotation of translation)

Mouse
movement

Inverse of
world-to-view

Create rotation
matrix

u

Rotation axis

$R_v^{-1}u$

R

Rotate here!

$R_v \quad T_v \quad T_w \quad R_w$

Pretty easy, just rotation around arbitrary axis and knowing the transformation sequence!