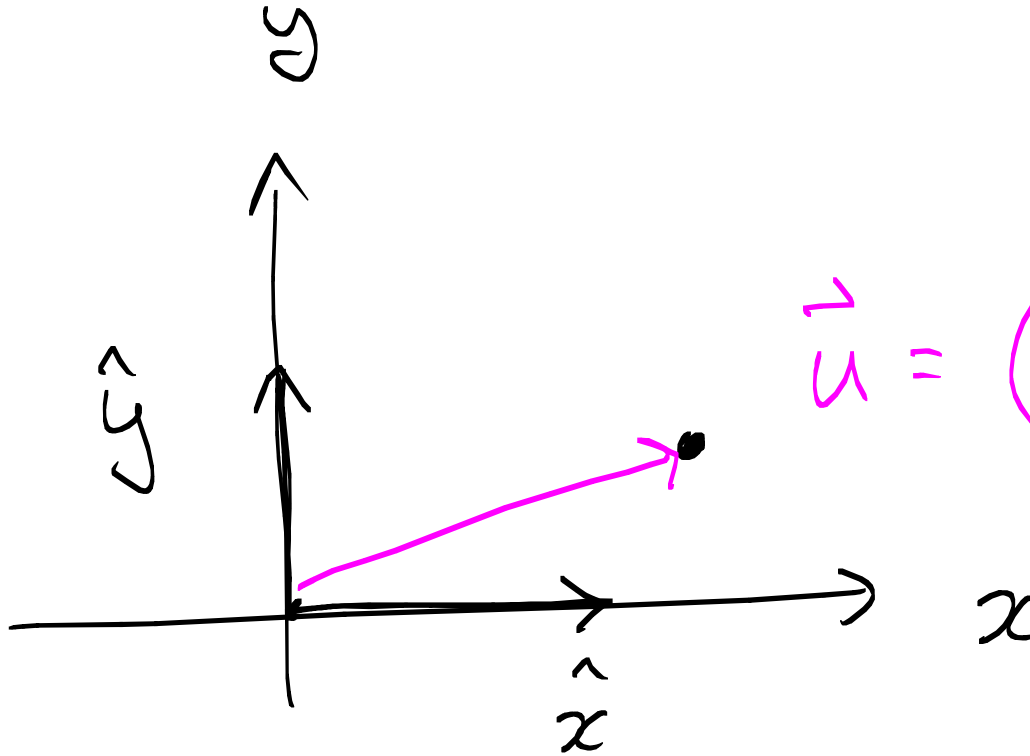
# lecture 2

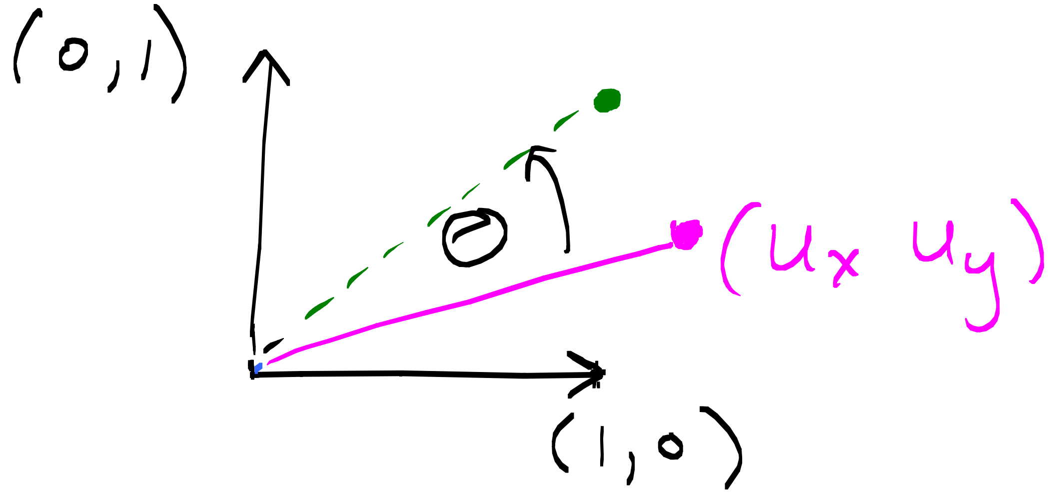
- model transformations
  (rotations, scaling, translation)

- intro to homogeneous coordinates




$$\vec{u} = (u_x \; u_y)$$

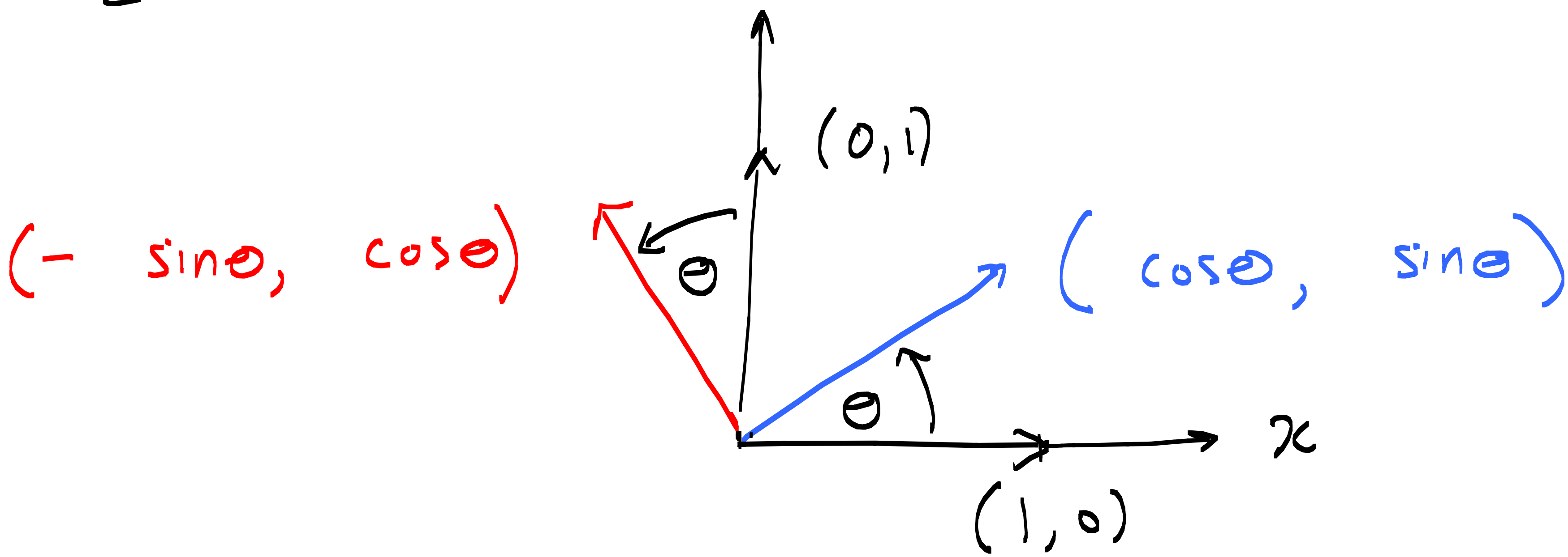
# 2D Rotation




$$\begin{bmatrix} u_x' \\ u_y' \end{bmatrix} = R_\theta \begin{bmatrix} u_x \\ u_y \end{bmatrix}$$

2x2 matrix

$$\begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} = R_\theta \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$



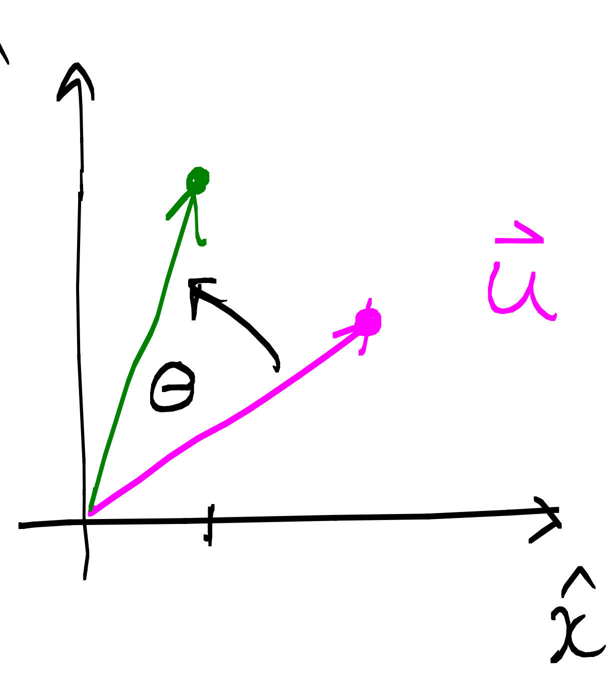

$$\begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} = R_\theta \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}_{2\times 2}$$

Two ways to think about R.

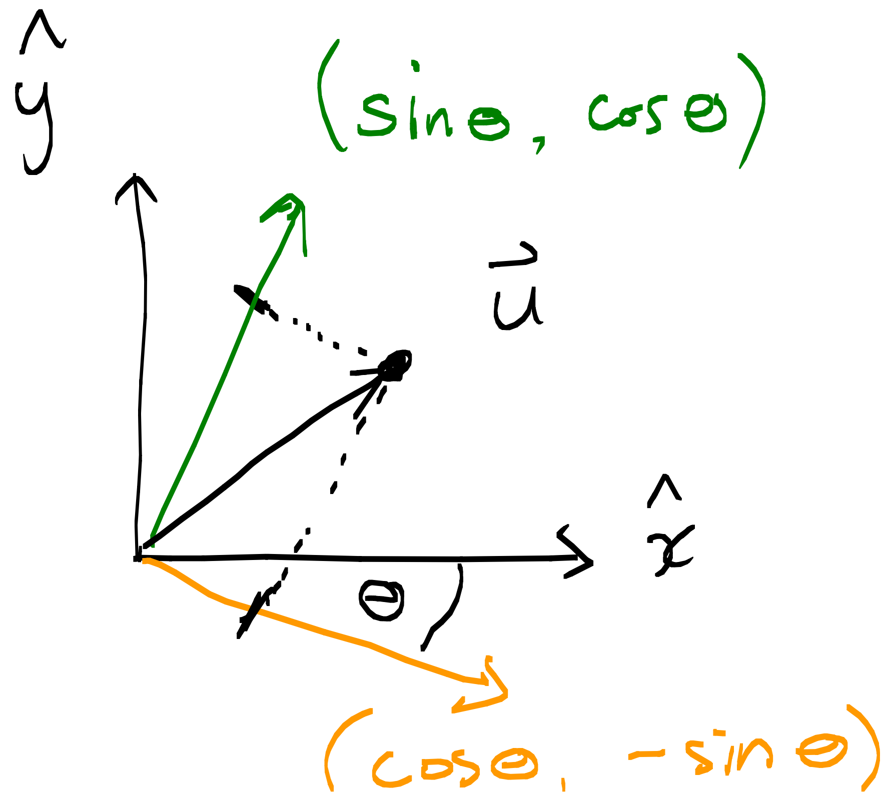
1) R rotates points within a fixed coordinate frame ("world coordinates")

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$



2) R maps to a new coordinate system by projecting onto new axes.

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$



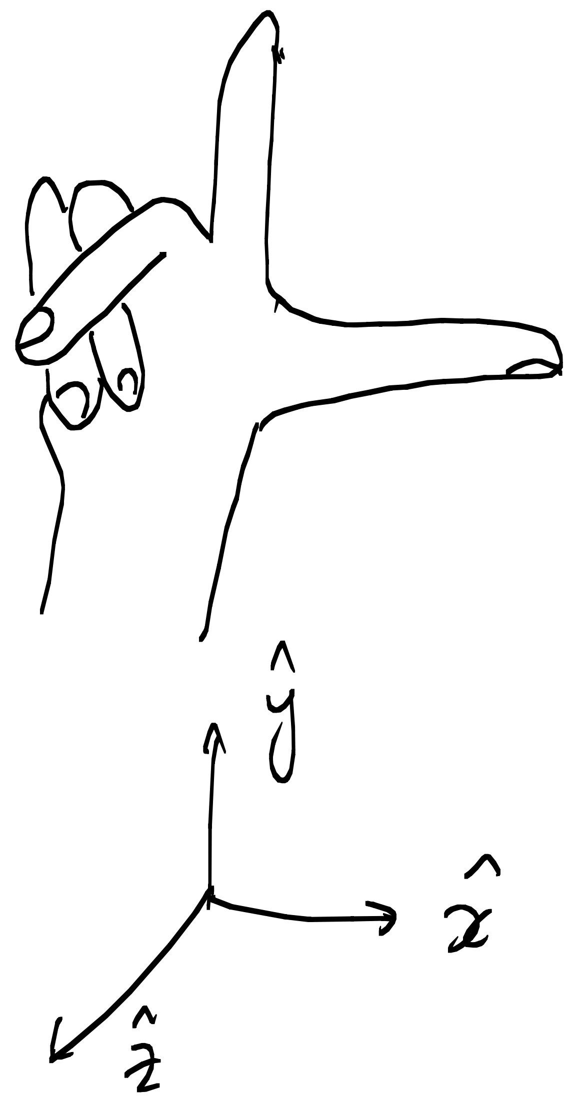
# How will rotations be used?

1) re-orient an object ("model")

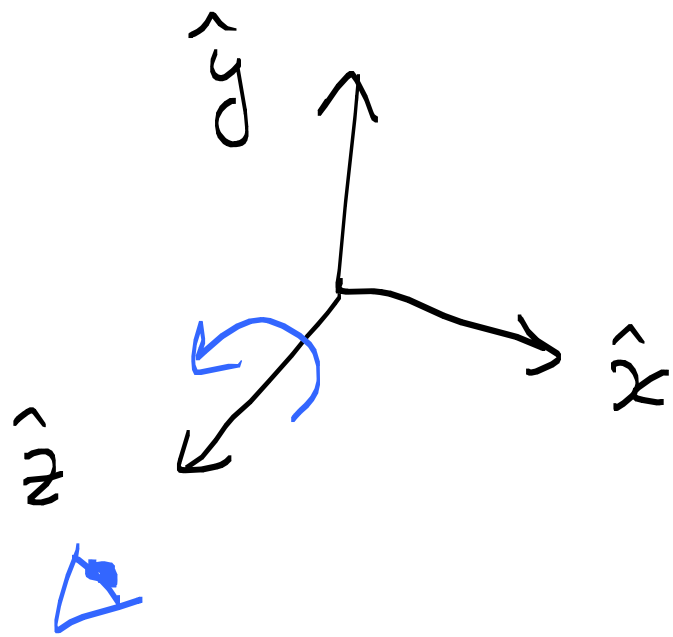
2) map from world coordinates to camera coordinates ("view")

# 3D Rotations

Left vs. Right Hand Coordinates

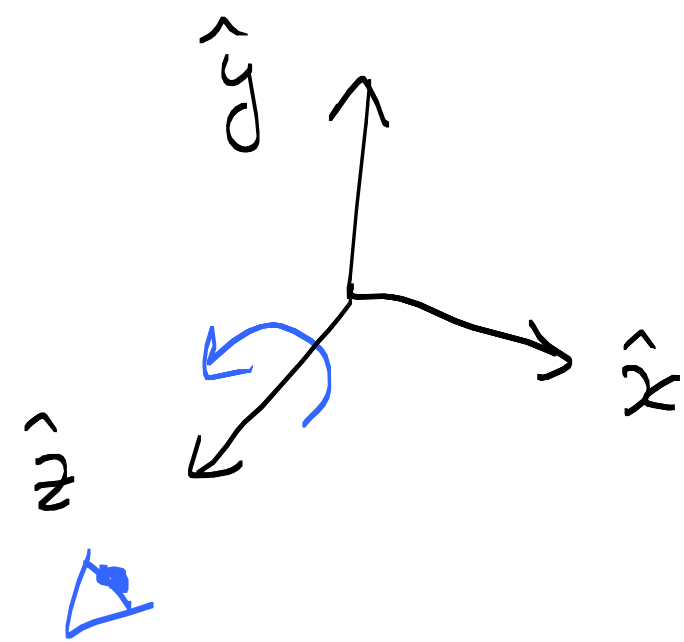
# Example:  rotate about z axis



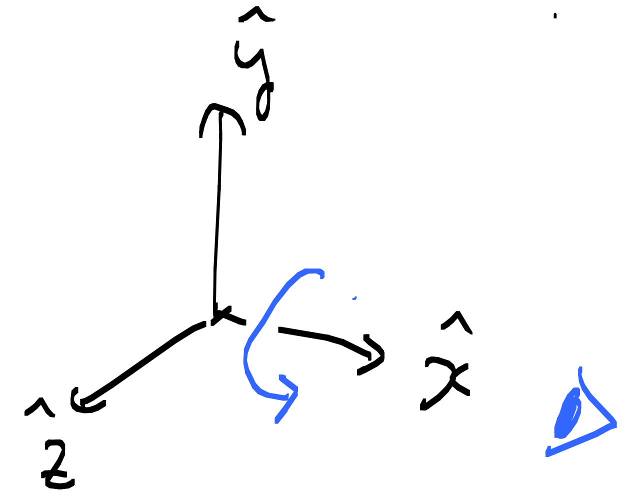

counter-clockwise

(assuming eye is looking in the -z direction and the coordinates are righthanded)

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



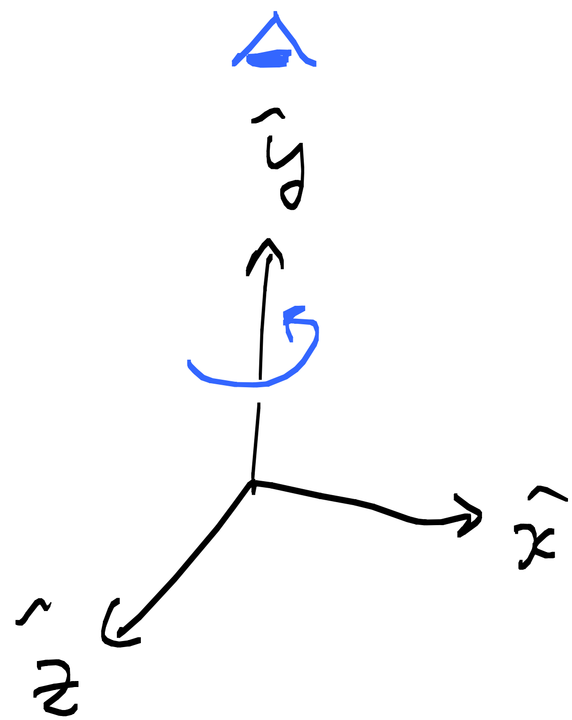
$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$




Counter-clockwise

note signs

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$



# General 3D rotation

$R_{3\times 3}$ such that

- $R^T R = I$ ⟵ identity matrix

  that is, $R^{-1} = R^T$

- determinant of $R$ is $1$.

**Claim:** Rotation matrices preserve dot product.

i.e. For any vectors u, v

$$\vec{u}\cdot\vec{v} = (R\vec{u})\cdot(R\vec{v})$$

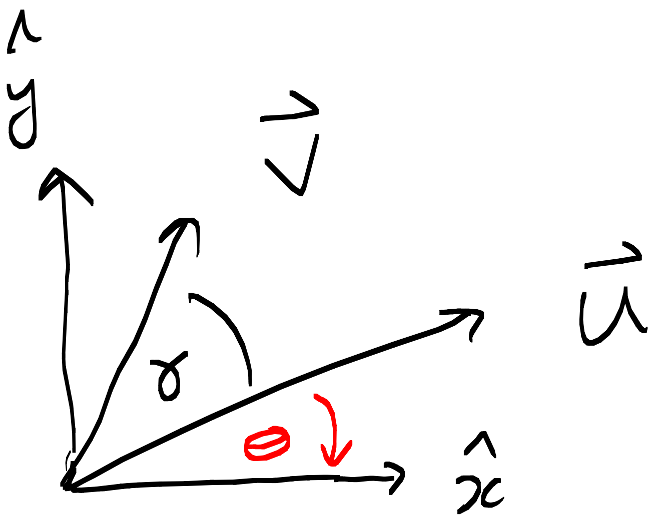
**Proof**

$$(R\vec{u})^T R\vec{v} = \vec{u}^T R^T R \vec{v}$$

$$= \vec{u}^T \vec{v}$$

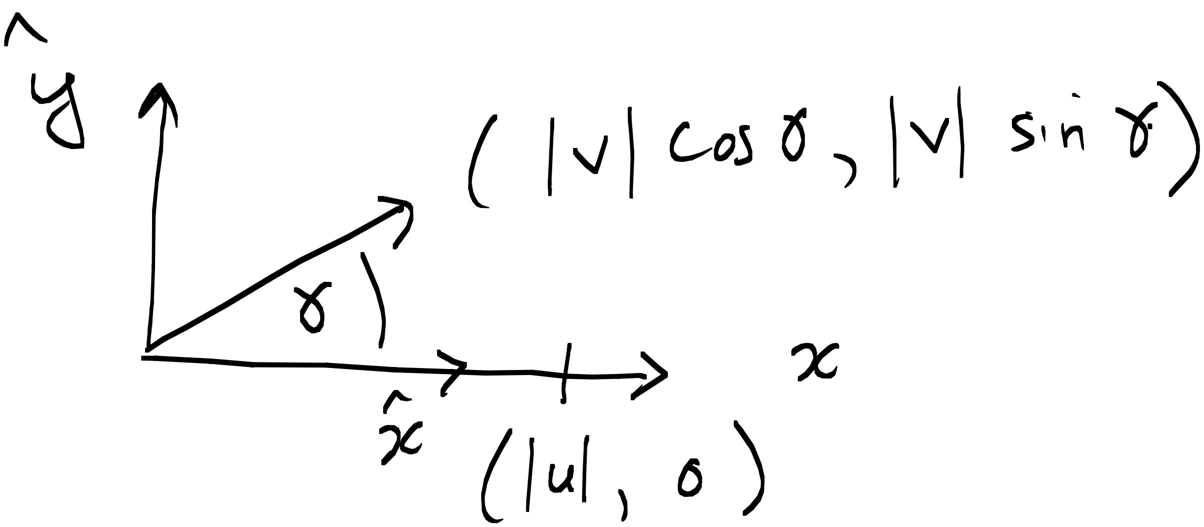
$$= \vec{u}\cdot\vec{v}$$

# Example (2D)




$$\vec{u} \cdot \vec{v} = |\vec{u}||\vec{v}| \cos \gamma$$

Rotate $\vec{u}$ to $x$ gives:



# Rotation versus Reflection

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

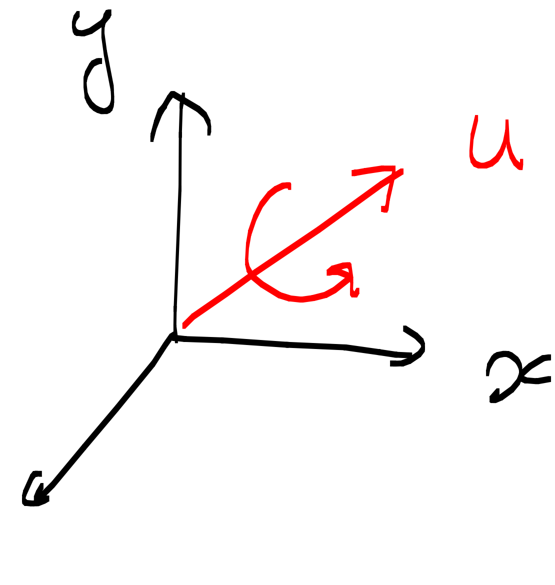
reflection about $x=0$ plane

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

reflection about $x=y$ plane

For these examples, determinant is $-1$ (not a rotation).

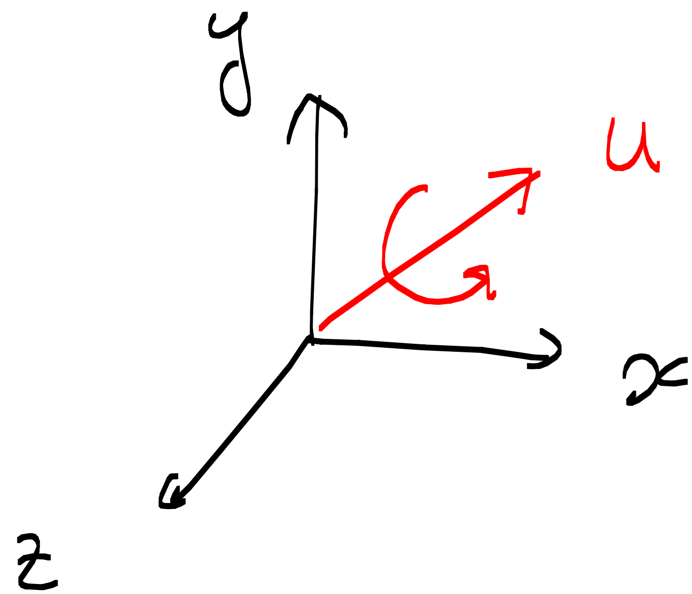
A rotation matrix R always defines an <u>axis</u> of rotation and an <u>angle</u> of rotation.




Given R, what is axis and angle?

$$R\vec{u} = \vec{u}$$

$\vec{u}$ is axis of rotation (eigenvector with eigenvalue 1)




<u>Exercise:</u>

What is angle?

# Example Problem 1

Given a unit vector $\vec{p}$, find a 3D rotation matrix that maps $\hat{z}$ to $\vec{p}$.

$$R\hat{z} = \vec{p}.$$

Assume $\vec{p} \neq \hat{z}$ since in that case the problem is trivial.

## Step 1

Observe the $3^{RD}$ column of $R$ must be the vector $p$.

Why?

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \underbrace{\begin{bmatrix} & & \\ & R & \\ & & \end{bmatrix}}_{3 \times 3} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Step 2: The first two columns of $R$ must be orthornormal to $p$.

Since $\vec{p} \neq \hat{z}$, we can use:

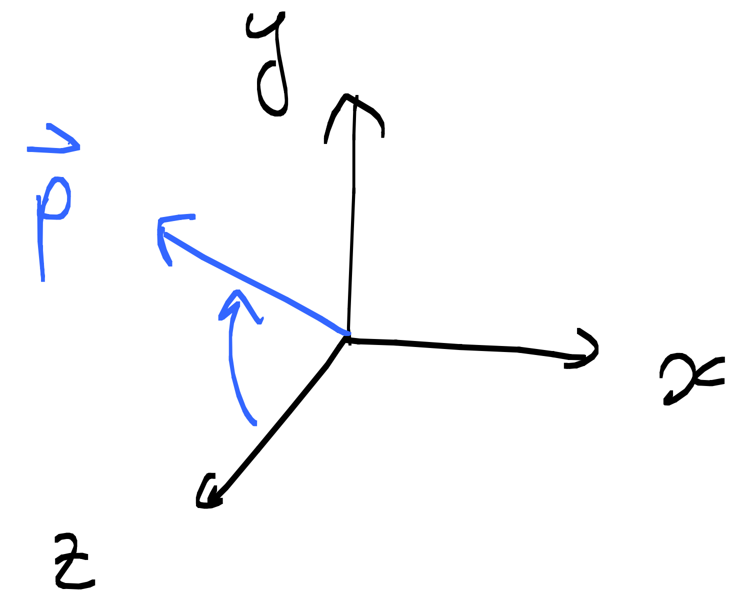
$$R = \begin{bmatrix} p' \times p & \underbrace{\dfrac{p \times z}{|p \times z|}}_{p'} & \vec{p} \end{bmatrix}$$

We just need to check that the determinant is $1$ (not $-1$).

## Recall: Example Problem 1

We have found $R$ such that

$$\vec{p} = R\,\hat{z}.$$

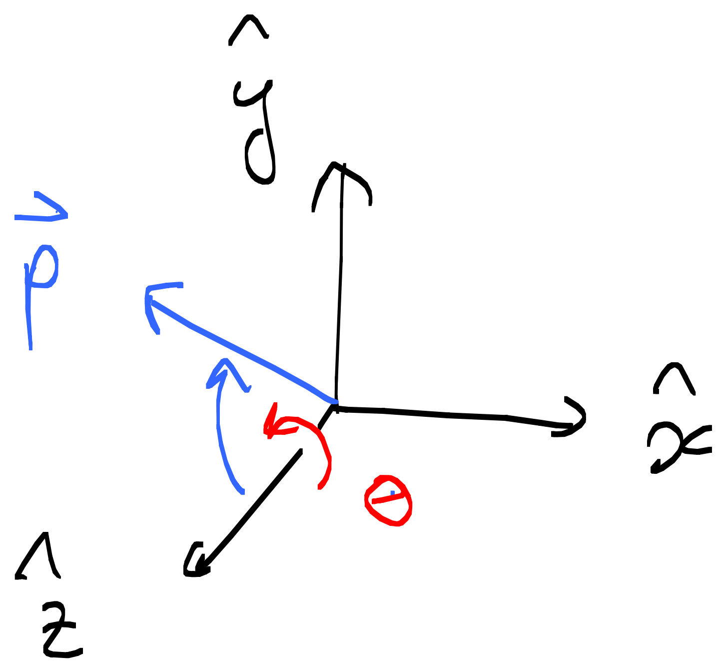



Is $R$ unique?

No, because

$$\vec{p} = \underbrace{R\,R_z(\theta)}\,\hat{z}$$

is another solution for any $\theta$.



# Problem Example 2

Find a rotation that maps a unit vector $\vec{p}$ to $\hat{z}$.

(Easy.)

$$\hat{z} = R^T \vec{p}$$

# Problem Example 3

Find a rotation matrix that rotates by $\theta$ around an axis $\vec{p}$.

**Step 1**: rotate $p$ to $z$ axis.

**Step 2**: rotate by $\theta$ around $\hat{z}$.

**Step 3**: rotate $z$ axis to $p$.

$$R \quad R_z(\theta) \, R^T$$

# Problem Example 4

Find a rotation matrix that rotates by $\theta$ around an axis ***p*** and that is composed of a sequence of rotations *only* around axes ***x, y, z.***

Example solution: (think this through for yourself )

1. Rotate around x axis to bring p to the xy plane.
2. Rotate around z axis to bring p to the y axis.
3. Rotate by $\theta$ around y axis.
4. Apply inverse rotation of 2.
5. Apply inverse rotation of 1.

# ASIDE: Representations of rotations

(very important for Computer Animation)

1) Axis-Angle -> OpenGL's glRotate()

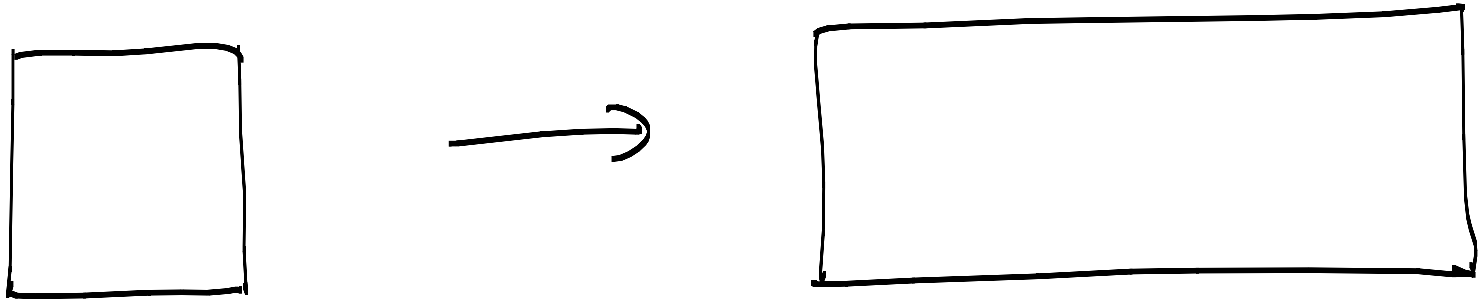
2) Euler angles (Rz Rx Ry)

3) Quaternions

https://www.youtube.com/watch?v=syQnn_xuB8U&list=PL2y2aRaUaygU2zXme_Z11GyJUslwgaeUD

https://www.youtube.com/watch?v=zc8b2Jo7mno

# Scaling



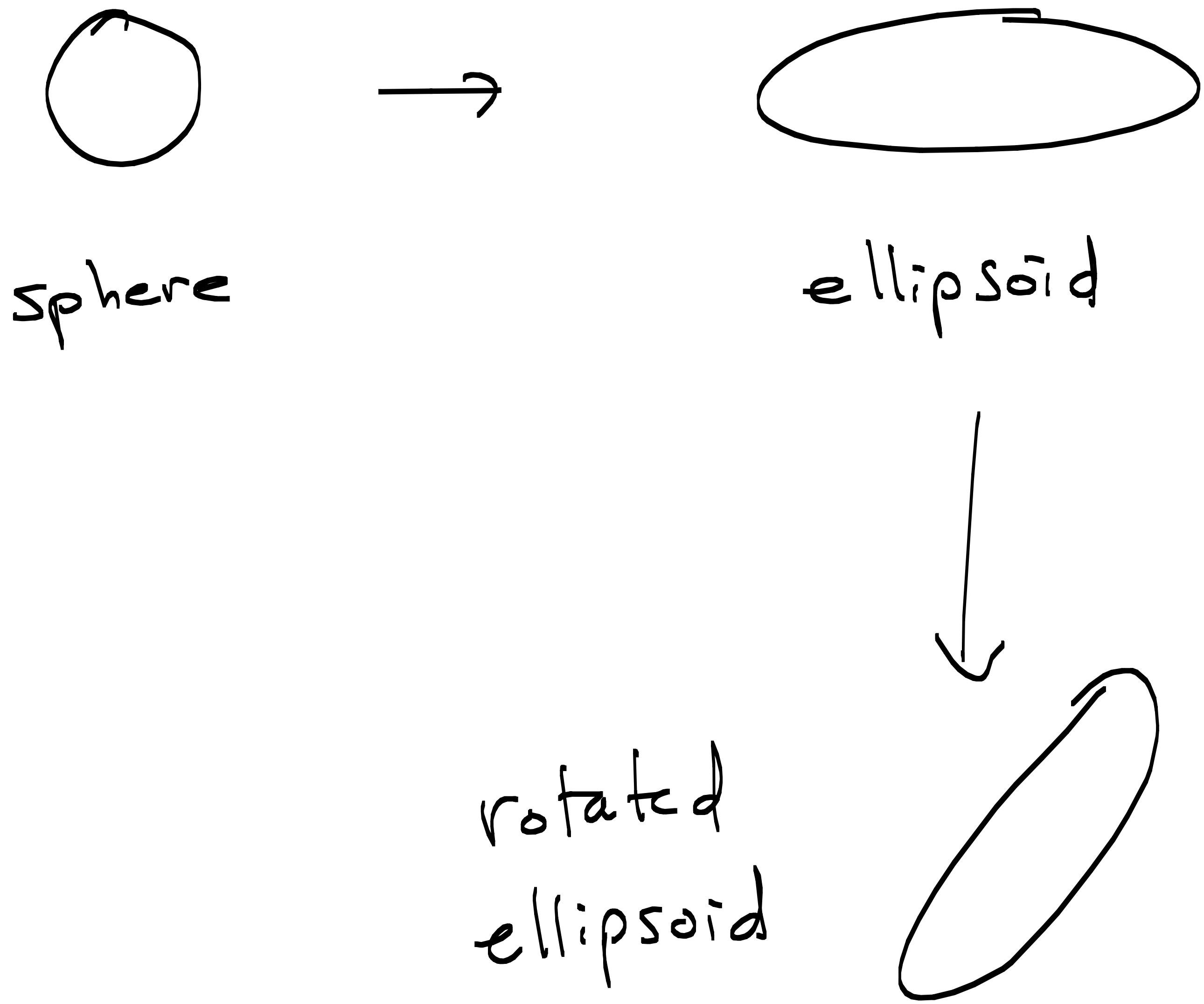
e.g. stretch in $x$ direction

$$(x, y, z) \longrightarrow (sx, y, z)$$

# Scaling

$$\begin{bmatrix} S_x x \\ S_y y \\ S_z z \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

This transformation stretches $(s > 1)$ or compresses $(s < 1)$ the scene in each of the canonical directions.


sphere
ellipsoid
rotated
ellipsoid

# Translation by $\vec{t}$

$$T : (x, y, z) \longrightarrow (x + t_x, y + t_y, z + t_z)$$

But this is <u>not</u> a linear transformation.

Why not?

$$\underbrace{T(\vec{u}+\vec{v})}_{\vec{u}+\vec{v}+\vec{t}} \neq \underbrace{T\vec{u}+T\vec{v}}_{\vec{u}+\vec{v}+2\vec{t}}$$

So we cannot represent $T$ by a $3 \times 3$ matrix.

Trick: use a 4th coordinate.

$$\begin{bmatrix} x + t_x \\ y + t_y \\ z + t_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

This is called a "homogeneous coordinates" representation.

In computer graphics, we always use a 4D representation to transform points.

$$\left[\begin{array}{c|c} R & \begin{matrix}0\\0\\0\end{matrix} \\ \hline \begin{matrix}0 & 0 & 0\end{matrix} & 1 \end{array}\right] \qquad \left[\begin{array}{c|c} S & \begin{matrix}0\\0\\0\end{matrix} \\ \hline \begin{matrix}0 & 0 & 0\end{matrix} & 1 \end{array}\right]$$

rotation

scaling

# Homogeneous Coordinates

We represent $(x,y, z)$ by $(x, y, z, 1)$.

Now define an equivalence:

$(x, y, z, 1) \equiv (w x, wy, wz, w)$ for any $w \neq 0$.

This takes each line $\{ (wx, wy, wz, w) \}$ in R^4
and associates it with the 3D point $(x, y, z)$.

# Careful:

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} + \begin{bmatrix} a' \\ b' \\ c' \\ d' \end{bmatrix} \not\equiv \begin{bmatrix} a + a' \\ b + b' \\ c + c' \\ d + d' \end{bmatrix}$$

The above is an abuse of notation.
It is meant to express that:

$$\begin{bmatrix} a/d \\ b/d \\ c/d \end{bmatrix} + \begin{bmatrix} a'/d' \\ b'/d' \\ c'/d' \end{bmatrix} \neq \begin{bmatrix} (a+a')/(d+d') \\ (b+b')/(d+d') \\ (c+c')/(d+d') \end{bmatrix}$$

# Points at infinity

Take $(x,y,z)$ and consider

$$\lim_{s \to \infty} (sx, sy, sz).$$

This can be expressed using homogeneous coordinates:

$$(sx, sy, sz, 1) \equiv \left(x, y, z, \frac{1}{s}\right)$$

Letting $s \to \infty$ gives $(x, y, z, 0)$.

Called a "point at infinity" (or "direction vector")

How do points at infinity behave under:

- rotation
- translation
- scaling

?

$$\begin{bmatrix} R\begin{bmatrix} x \\ y \\ z \end{bmatrix} \\ 0 \end{bmatrix} = \left[\begin{array}{ccc|c} & R & & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} S_x x \\ S_y y \\ S_z z \\ 0 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$$

Scaling stretches / compresses the axes.

What does $(x, y, z, \varepsilon)$ represent

as $\varepsilon \to 0$ from positive side

versus $\varepsilon \to 0$ from negative side?