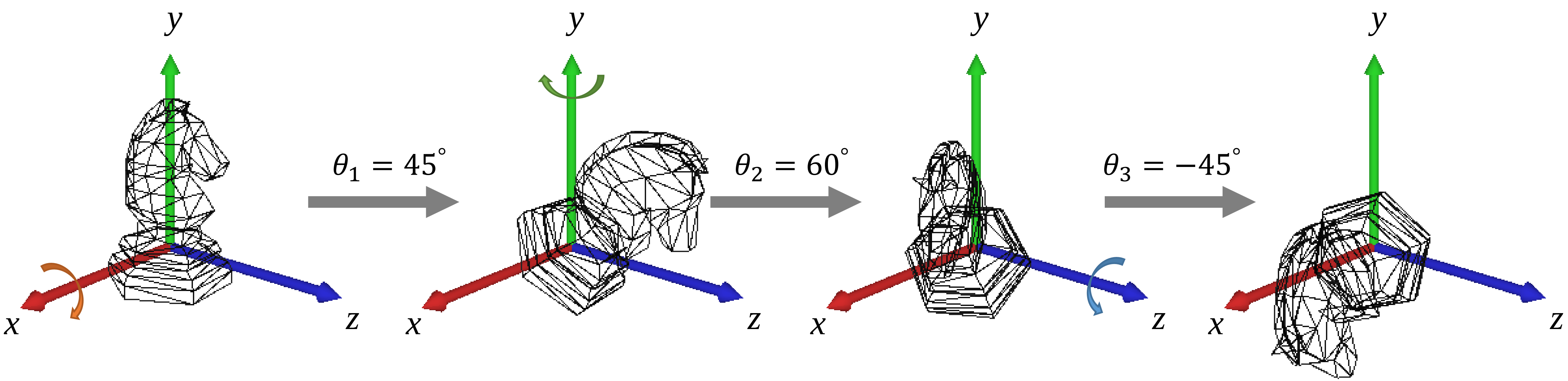
10. Euler transforms and quaternions

Prof. HyeongYeop Kang
siamiz@khu.ac.kr
YouTube: HKang IIIXR LAB
IIIXR LAB

Euler Transforms

Euler transform and Euler angles.

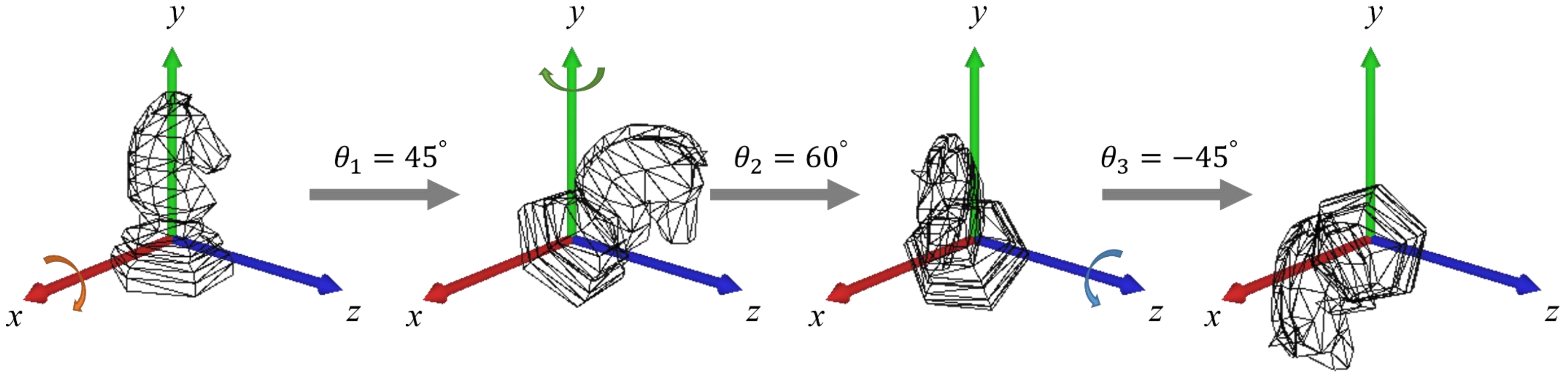
- When we successively rotate an object about the principal axes, the object acquires an arbitrary orientation.
- This method of determining an object's orientation is called the *Euler transform*, and the rotation angles, $(\theta_1, \theta_2, \theta_3)$, are called the *Euler angles*.



Euler Transforms

Euler transform and Euler angles.

- Concatenating three matrices produces a single matrix defining an arbitrary orientation.

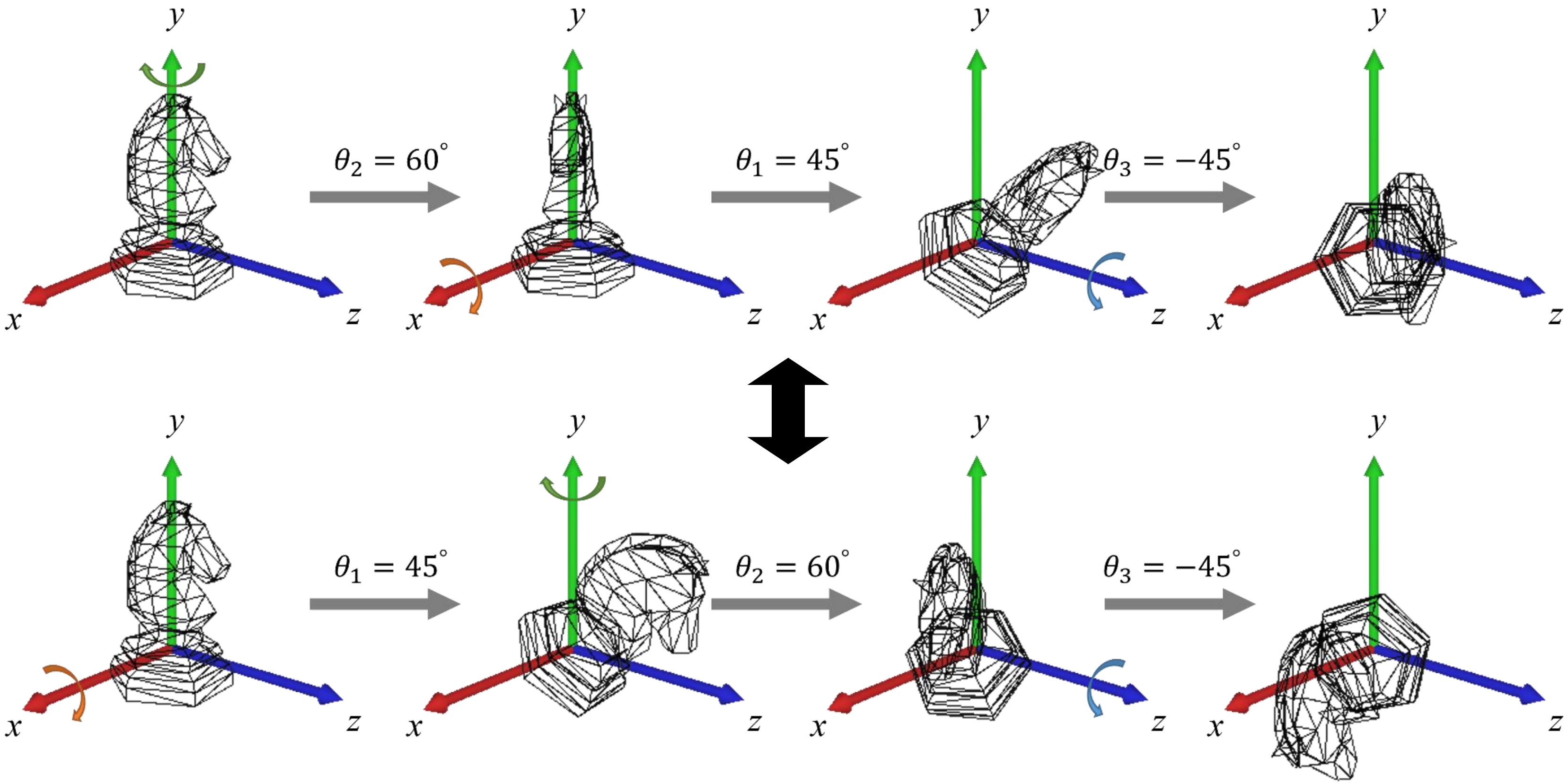


$$R_x(45^\circ)R_y(60^\circ)R_z(\text{-}45^\circ) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{2} & 0 & -\frac{\sqrt{3}}{2} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\sqrt{3}}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Euler Transforms

The order of rotation

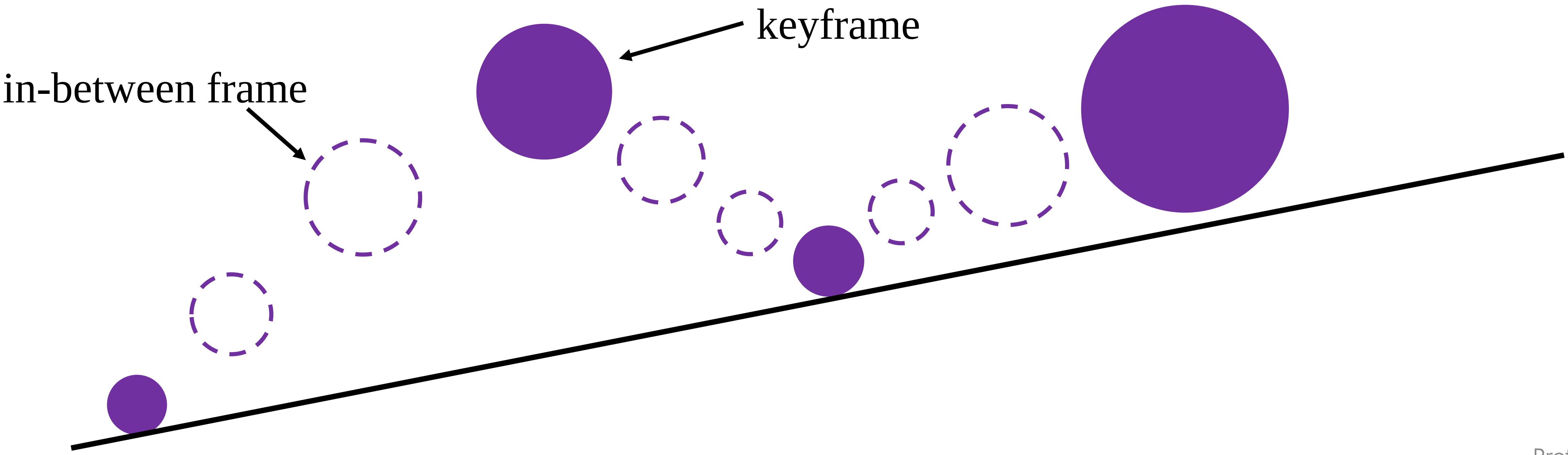
- The rotation axes are not necessarily taken in the order of x, y, and z.
- Shown below is the order of y, x, and z. Observe that the teapot has a different orientation from the previous one.



Keyframe Animation in 2D

Keyframe

- In the traditional hand-drawn cartoon animation, the senior key artist would draw the keyframes, and the junior artist would fill the in-between frames.
- For a 30-fps computer animation, for example, much fewer than 30 frames are defined per second. They are the keyframes.

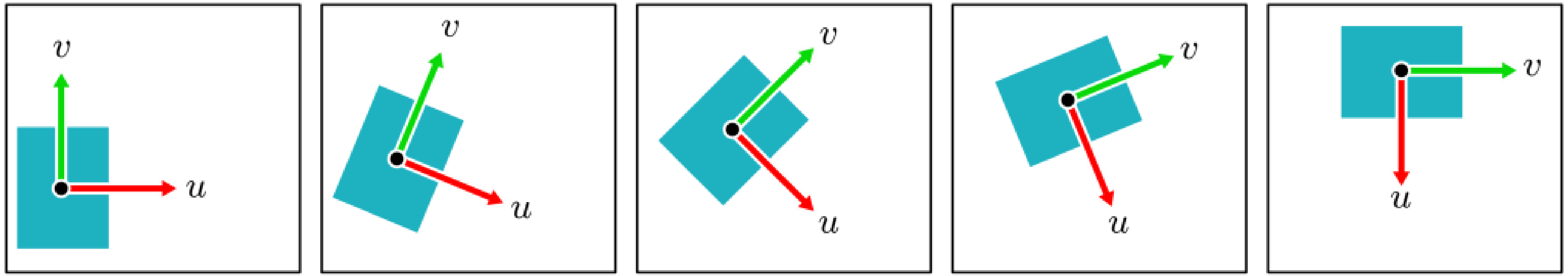


Keyframe Animation in 2D

Keyframe

- In real-time computer animation, the in-between frames are automatically filled at run time.
- The key data are assigned to the keyframes, and they are interpolated to generate the in-between frames.
- In the example, the center position p and orientation angle θ are interpolated.
  - $p(t) = (1 - t)p_0 + tp_1$
  - $\theta(t) = (1 - t)\theta_0 + t\theta_1$

$t = 0$ $t = 1$

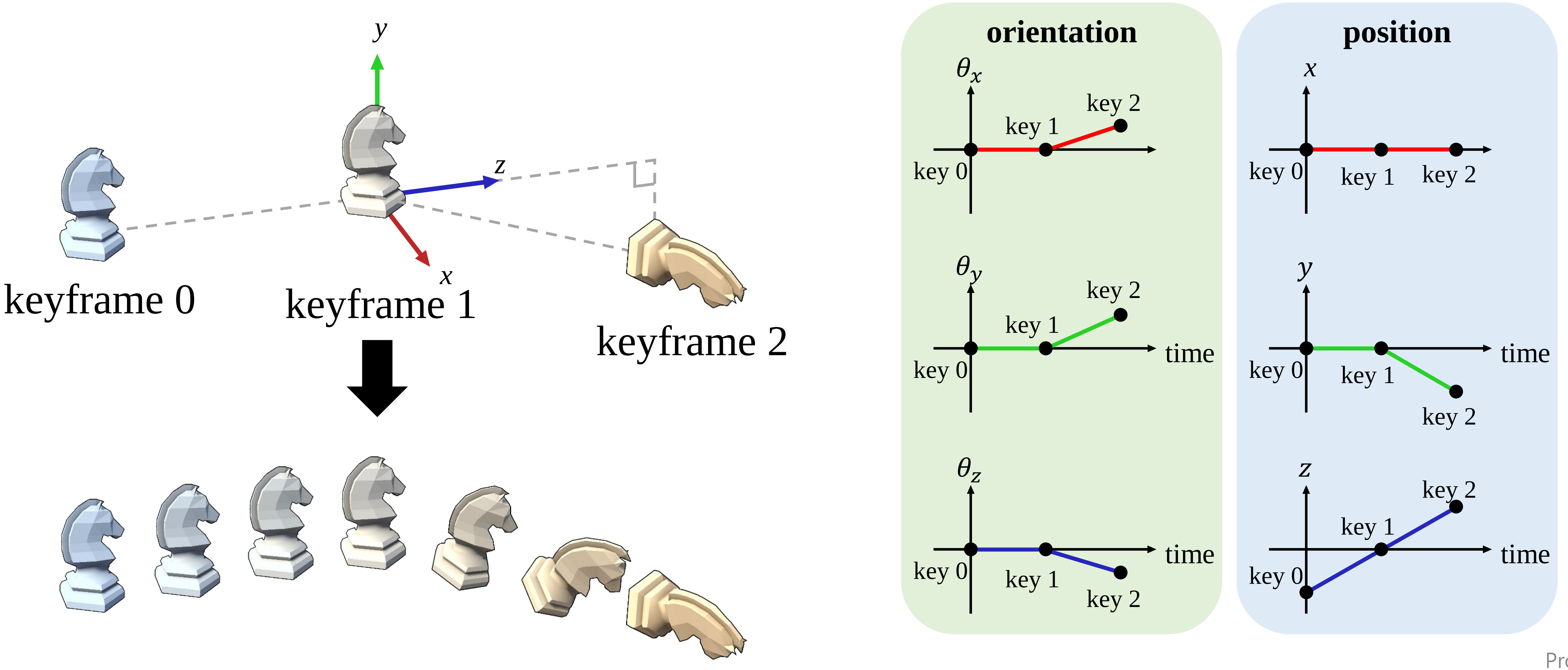


keyframe 0 keyframe 1

Keyframe Animation in 3D

Sampling between keyframes.

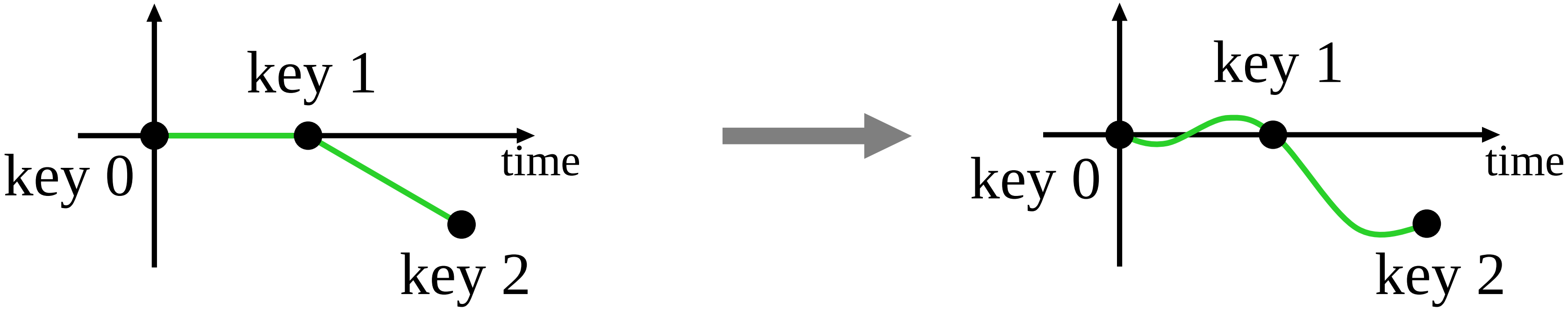
- Seven chess piece instances are defined by sampling the graphs seven times.



Keyframe Animation in 3D

Animation smoothing

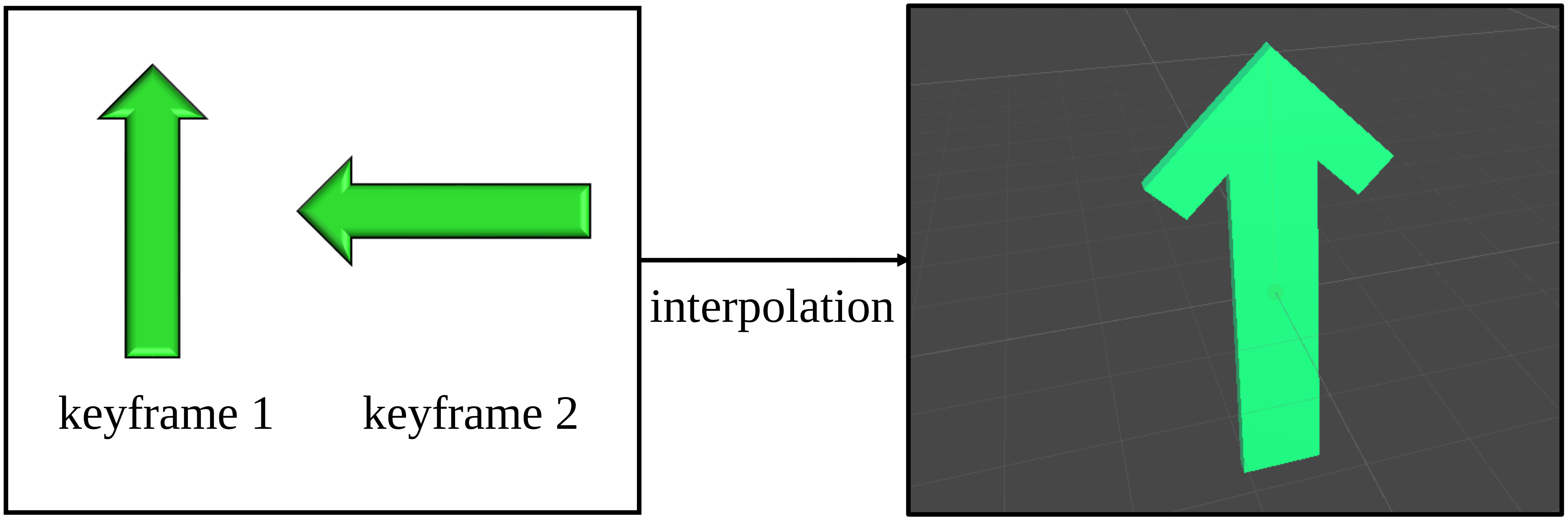
- Smoother animation may often be obtained using a higher-order interpolation.



A Problem of Euler Angles

Euler angles is intuitive and easy to use!

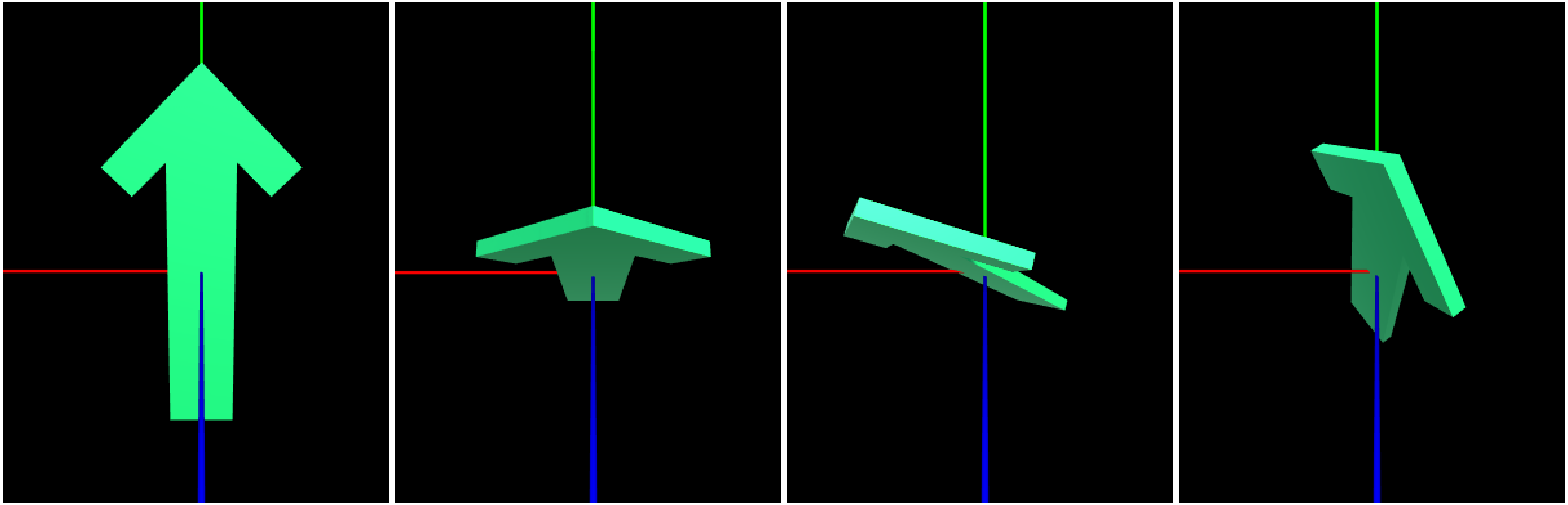
- However, every ease of use will come up by a cost.
- Euler method can’t provide you a nice interpolation between two different Euler rotations.



A Problem of Euler Angles

Why?

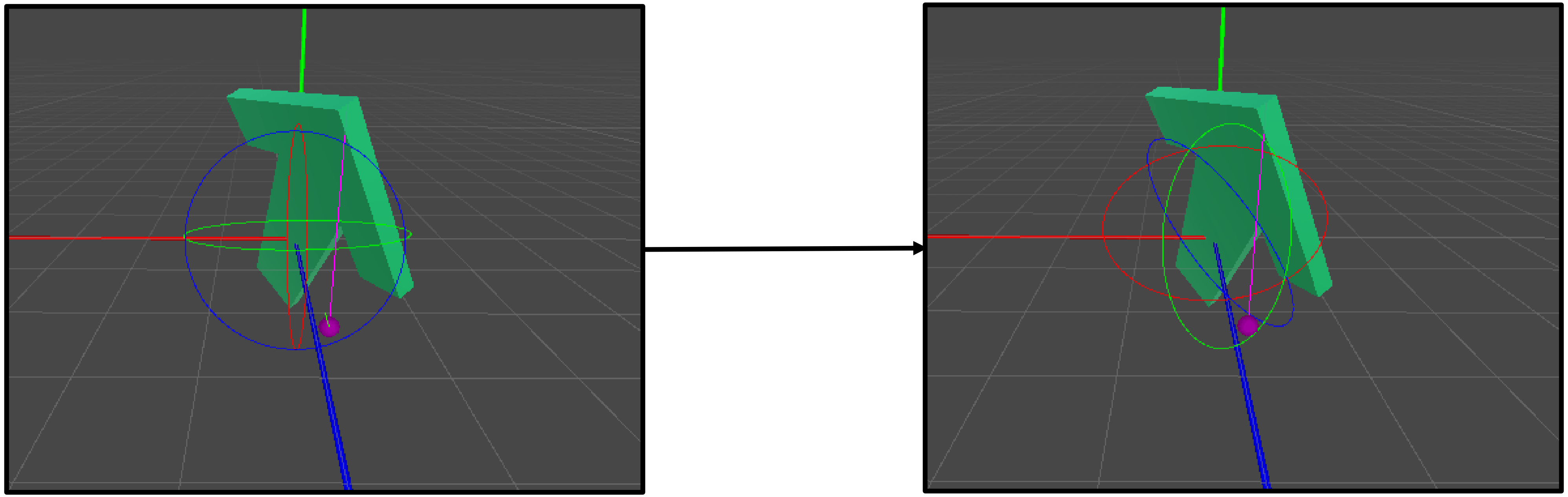
- Euler angle rotates object only along the axis.
- Imagine the rotation of (60, 30, 45).



A Problem of Euler Angles

Why?

- Euler angle rotates object only along the axis.
- Imagine the rotation of (60, 30, 45) and rotates the object along the Y axis again.
- Since the previous rotation distorted the *rotation axis (gimbal axis)*, Y rotation cannot be achieved by rotating one *axis*. two or three *axis* will be rotating to reach the specified goal.

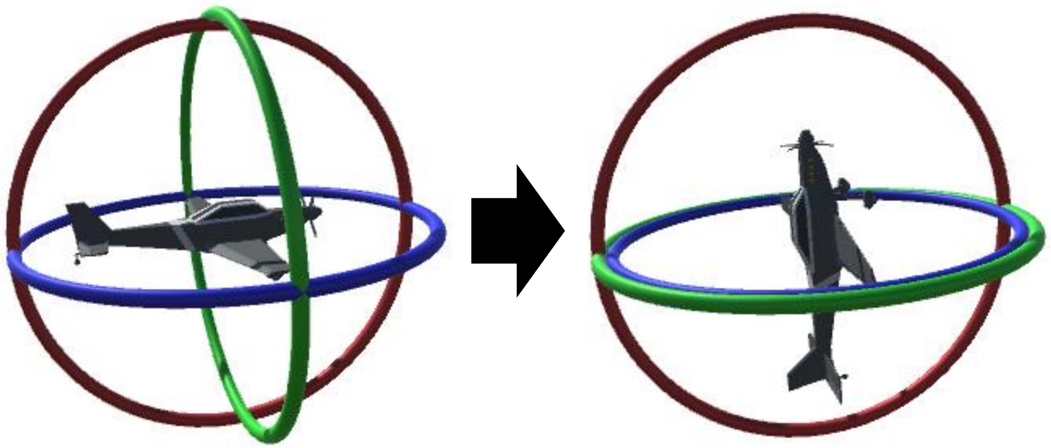


rotation along the Y axis cannot be replayed correctly

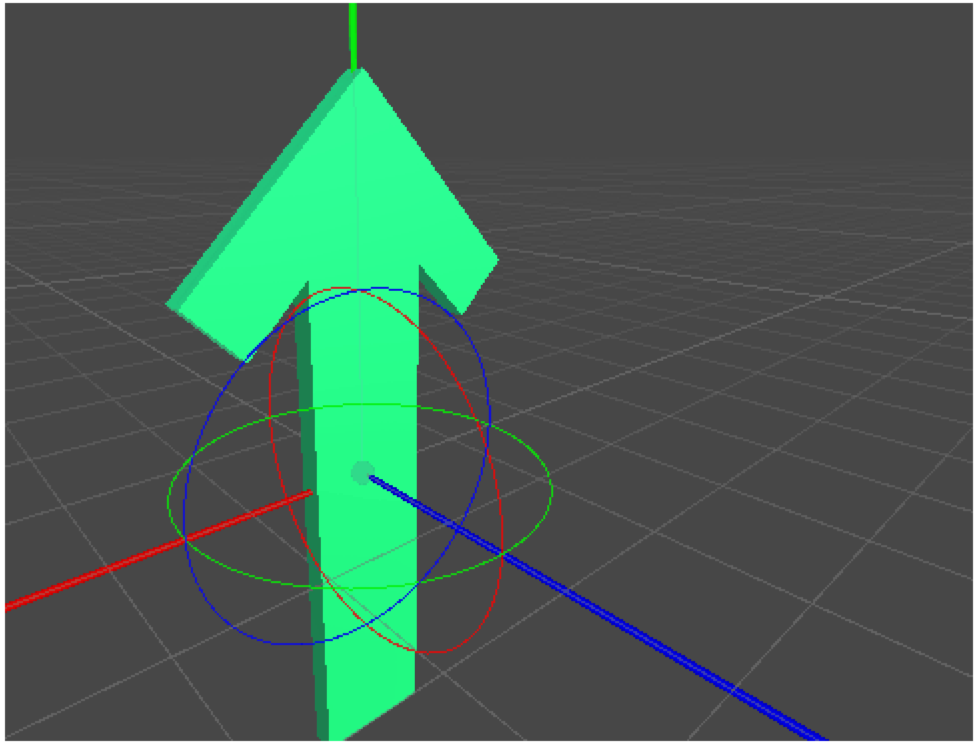
A Problem of Euler Angles

Gimbal lock

- Gimbal coordinate system (Euler rotation) suffers from an issue called gimbal lock.
- When it occurs, you lose one degree of freedom.



gimbal lock



Quaternion

Quaternion?

- To overcome the limitation of the Euler angle, the quaternion number system has been used in graphics.
- Conceptually, quaternions are used to represent an axis-angle rotation about an arbitrary axis.
- A spatial rotation around a fixed point of θ radians about a unit axis $\hat{e} = (\hat{x}, \hat{y}, \hat{z})$ is given by the unit quaternion
  - $\mathbf{q} = (q_x, q_y, q_z, q_w) = q_x i + q_y j + q_z k + q_w = e^{\frac{\theta}{2}(\hat{x}i+\hat{y}j+\hat{z}k)} = \cos\frac{\theta}{2} + (\hat{x}i + \hat{y}j + \hat{z}k)\sin\frac{\theta}{2}$

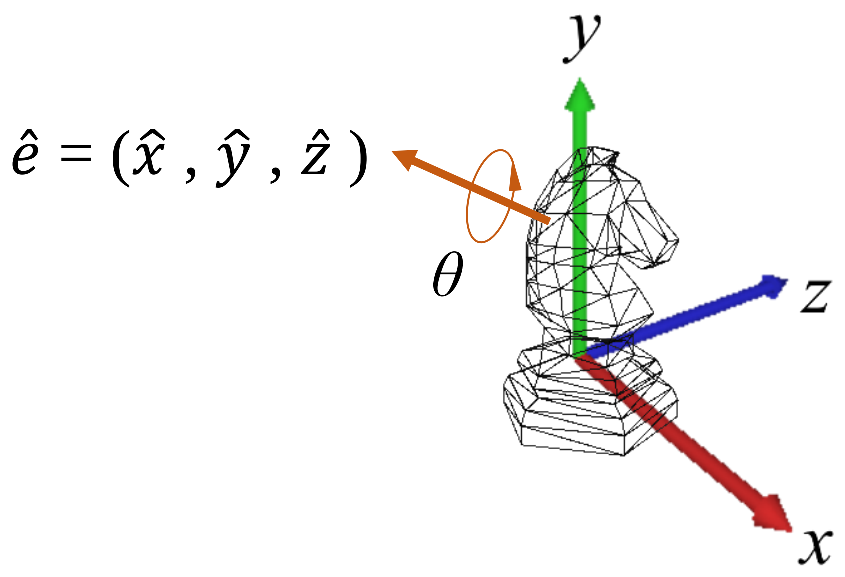
    $= (\hat{x}\sin\frac{\theta}{2}, \hat{y}\sin\frac{\theta}{2}, \hat{z}\sin\frac{\theta}{2}, \cos\frac{\theta}{2})$

$e^{ix} = \cos x + i\sin x$ (Euler's formula)

Quaternion

Quaternion?

- Quaternions are more compact, efficient, and numerically stable compared to Euler Rotation.
- However, quaternions look more complex and difficult to understand.
- Furthermore, due to the periodic nature of sine and cosine, rotation angles differing precisely by 2π will be encoded into identical quaternions and recovered angles in radians will be limited to $[0, 2\pi]$.



Quaternion

A quaternion is an extended complex number.

- Quaternions encode the axis-angle representation in four numbers.
  - $\mathbf{q} = (q_x, q_y, q_z, q_w) = q_x i + q_y j + q_z k + q_w$, where (q_x, q_y, q_z, q_w) are real numbers and (i, j, k) are symbols that represent unit-vectors pointing along the three spatial axes.
  -

- A quaternion consists of a vector part and a scalar part, $\mathbf{q} = (q_v, q_w)$
  - The $q_x i + q_y j + q_z k$ is called the vector part (imaginary part) of $\mathbf{q}$.
  - The q_w is the scalar part (real part) of $\mathbf{q}$.

- The multiplication of basis is as follows:
  - $i1 = 1i = i$, $j1 = 1j = j$, $k1 = 1k = k$.
  - $i^2 = j^2 = k^2 = -1$.
  - $ij = k$, $ji = -k$, $jk = i$, $kj = -i$, $ki = j$, $ik = -j$.

$\times$	1	i	j	k
1	1	i	j	k
i	i	-1	k	$-j$
j	j	$-k$	-1	i
k	k	j	$-i$	-1

Quaternion

Hamilton Product

- For two quaternions of **p** and **q**, their product (called the Hamilton product) is determined by the products of the basis elements and the distributive law.
  - $\mathbf{p} = (p_x, p_y, p_z, p_w)$
  - $\mathbf{q} = (q_x, q_y, q_z, q_w)$
  - $\mathbf{pq} = (p_x i + p_y j + p_z k + p_w)(q_x i + q_y j + q_z k + q_w)$

$$= (p_x q_w + p_y q_z - p_z q_y + p_w q_x)\color{red}{i} + (-p_x q_z + p_y q_w + p_z q_x + p_w q_y)\color{red}{j}$$

$$+ (p_x q_y - p_y q_x + p_z q_w + p_w q_z)\color{red}{k} + (-p_x q_x - p_y q_y - p_z q_z + p_w q_w)$$

Quaternion

Conjugation of the quaternion

- The conjugate of the quaternion **q** is the quaternion **q*** = $(-q_v, q_w) = -q_x i - q_y j - q_z k + q_w$.
- Conjugating an element twice equals the original element: **(q*)* = q**
- The conjugate of a product of two quaternions is the product of the conjugates in the reverse order: **(pq)* = q*p***
- The conjugation can be used to extract the scalar and vector parts of **p**.
  - Scalar part: $\frac{1}{2}(\mathbf{p} + \mathbf{p}^*)$
  - Vector part: $\frac{1}{2}(\mathbf{q} + \mathbf{q}^*)$

Quaternion

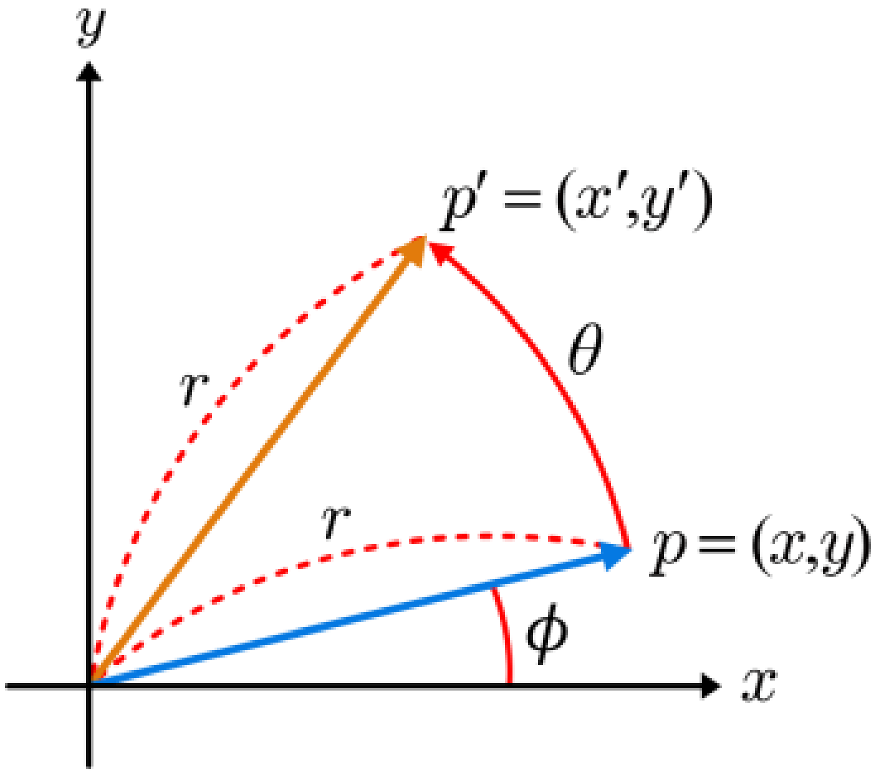
Unit Quaternion

- The magnitude of a quaternion **p** can be obtained by the square root of the product of a quaternion with its conjugate.
  - This is called its norm and is denoted $||\mathbf{p}||$.
  - Therefore, $||\mathbf{p}|| = \sqrt{\mathbf{pp}*} = \sqrt{(p_x)^2 + (p_y)^2 + (p_z)^2 + (p_w)^2}$

- A quaternion whose norm is 1 is called a unit quaternion.
  - $\hat{\mathbf{p}} = \dfrac{\mathbf{p}}{||\mathbf{p}||}$

2D rotation through complex numbers

Recall 2D rotation



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} xcos\theta - ysin\theta \\ xsin\theta + ycos\theta \end{pmatrix}$$

- Let us represent (x, y) by a complex number $x + yi$, and denote it by **p**.
- Given the rotation angle θ, let us define a unit-length complex number, $\mathbf{q} = cos\theta + sin\theta i$.
- Surprisingly, the real and imaginary parts of **pq** represent the rotated coordinates:
  - $pq = (x + yi)(cos\theta + sin\theta i) = (xcos\theta - ysin\theta) + (xsin\theta + ycos\theta)i$

3D rotation through complex numbers

Quaternions can also be used to describe 3D rotation.

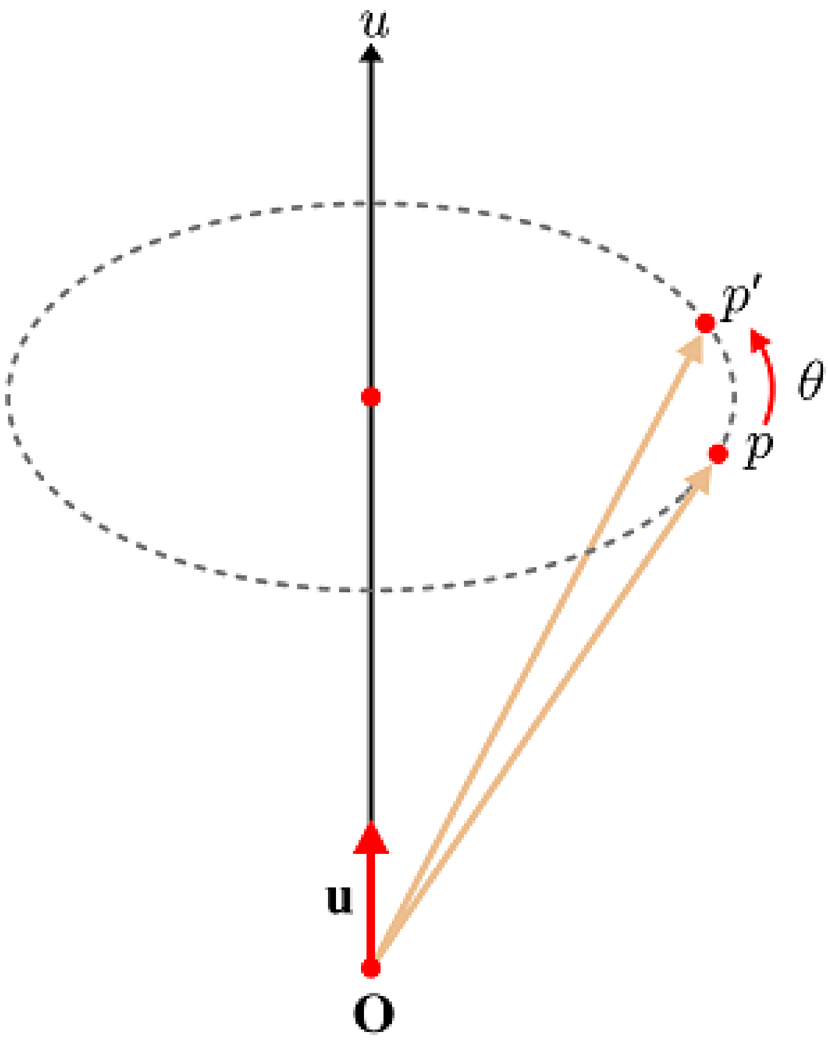
- $(q_x, q_y, q_z, q_w) = q_x i + q_y j + q_z k + q_w$
- Consider rotating a 3D vector p about an axis u by an angle θ.
  - Let us represent $p = (x, y, z)$ by $xi + yj + zk$, and denote it by a quaternion **p** whose real part is 0.

  $\rightarrow \mathbf{p} = (p_v, p_w) = (p, 0)$

  - Define a *unit quaternion* **q** using a unit vector u and θ.

  $\rightarrow \mathbf{q} = (q_v, q_w) = (sin\frac{\theta}{2}u, cos\frac{\theta}{2})$

  - Then the imaginary part of the calculation result of **qpq*** represents the rotated vector.



3D rotation through complex numbers

Proof

- When we denote **p** and **q** by (p_v, q_w) and (q_v, q_w), respectively, **qp** can be written as follows:

$$\begin{aligned}\mathbf{qp} = &(p_x q_w + p_y q_z - p_z q_y + p_w q_x)i \\ &+ (-p_x q_z + p_y q_w + p_z q_x + p_w q_y)j \\ &+ (p_x q_y - p_y q_x + p_z q_w + p_w q_z)k \\ &+ (-p_x q_x - p_y q_y - p_z q_z + p_w q_w)\end{aligned}$$

$$\Rightarrow \quad (p_v \times q_v + q_w p_v + p_w q_v \, , \, p_w q_w - p_v \cdot q_v) \text{ (Eq. 1)}$$

$$p_v \times q_v = \begin{vmatrix} i & j & k \\ p_x & p_y & p_z \\ q_x & q_y & q_z \end{vmatrix}$$

$$= i(p_y q_z - p_z q_y) + j(p_z q_x - p_x q_z) + k(p_x q_y - p_y q_x)$$

$$q_w p_v = p_x q_w i + p_y q_w j + p_z q_w k$$

$$p_w q_v = p_w q_x i + p_w q_y j + p_w q_z k$$

3D rotation through complex numbers

- Using $\mathbf{qp} = (p_v \times q_v + q_w p_v + p_w q_v\,,\ p_w q_w - p_v \cdot q_v)$, $\mathbf{qpq}^*$ is expanded as follows:

$\mathbf{qpq}^* = (q_v \times p_v + q_w p_v,\ - q_v \cdot p_v)\ \mathbf{q}^*$ (since $p_w = 0$)

$= (q_v \times p_v + q_w p_v,\ - q_v \cdot p_v)\ (-q_v,\ q_w)$

$= ((q_v \times p_v + q_w p_v) \times (-q_v) + q_w(q_v \times p_v + q_w p_v) + (- q_v \cdot p_v)(-q_v),\ (- q_v \cdot p_v)q_w - (q_v \times p_v + q_w p_v) \cdot (-q_v))$

[Using Eq. 1]

$= ((q_v \cdot p_v)q_v - (q_v \cdot p_v)\ p_v + 2q_w(q_v \times p_v) + q_w^2 p_v + (q_v \cdot p_v)\ q_v\ ,0)$

[Using $a \times (b \times c) = (a \cdot c)b - (a \cdot b)c$, the real part becomes 0 since $(q_v \times p_v) \cdot q_v = 0$]

$= (2(q_v \cdot p_v)q_v + (q_w^2 - ||q_v||^2)p_v + 2q_w(q_v \times p_v)\ ,0)$

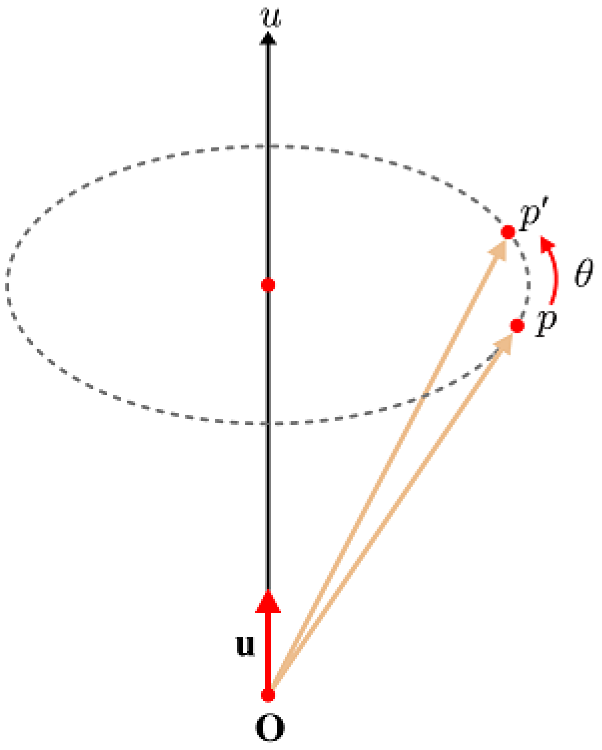
$= (2sin^2\frac{\theta}{2}(u \cdot p_v)u + (cos^2\frac{\theta}{2} - sin^2\frac{\theta}{2})p_v + 2cos\frac{\theta}{2}\,sin\frac{\theta}{2}(u \times p_v),0)$ [Using $q_v = sin\frac{\theta}{2}u$, $q_w = cos\frac{\theta}{2}u$]

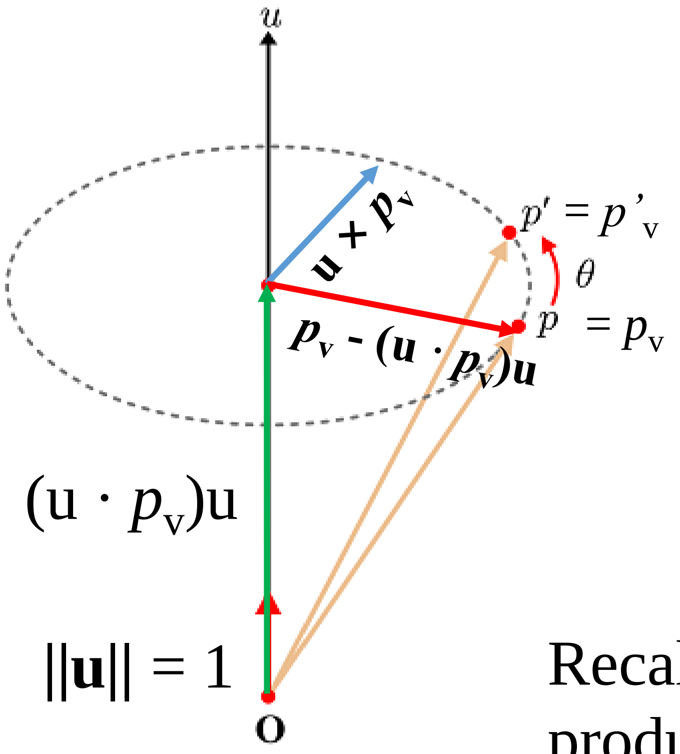
$= ((1-cos\theta)(u \cdot p_v)u + cos\theta p_v + sin\theta(u \times p_v),0)$

[Using $sin^2\frac{\theta}{2} = \frac{1-cos\theta}{2}$, $cos^2\frac{\theta}{2} = \frac{1+cos\theta}{2}$, $2sin\theta cos\theta = sin2\theta$]

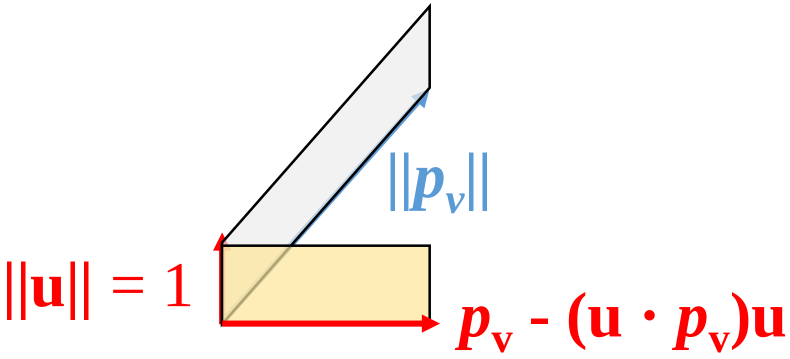
$= ((u \cdot p_v)u + cos\theta(p_v - (u \cdot p_v)u\ ) + sin\theta(u \times p_v),0)$

3D rotation through complex numbers





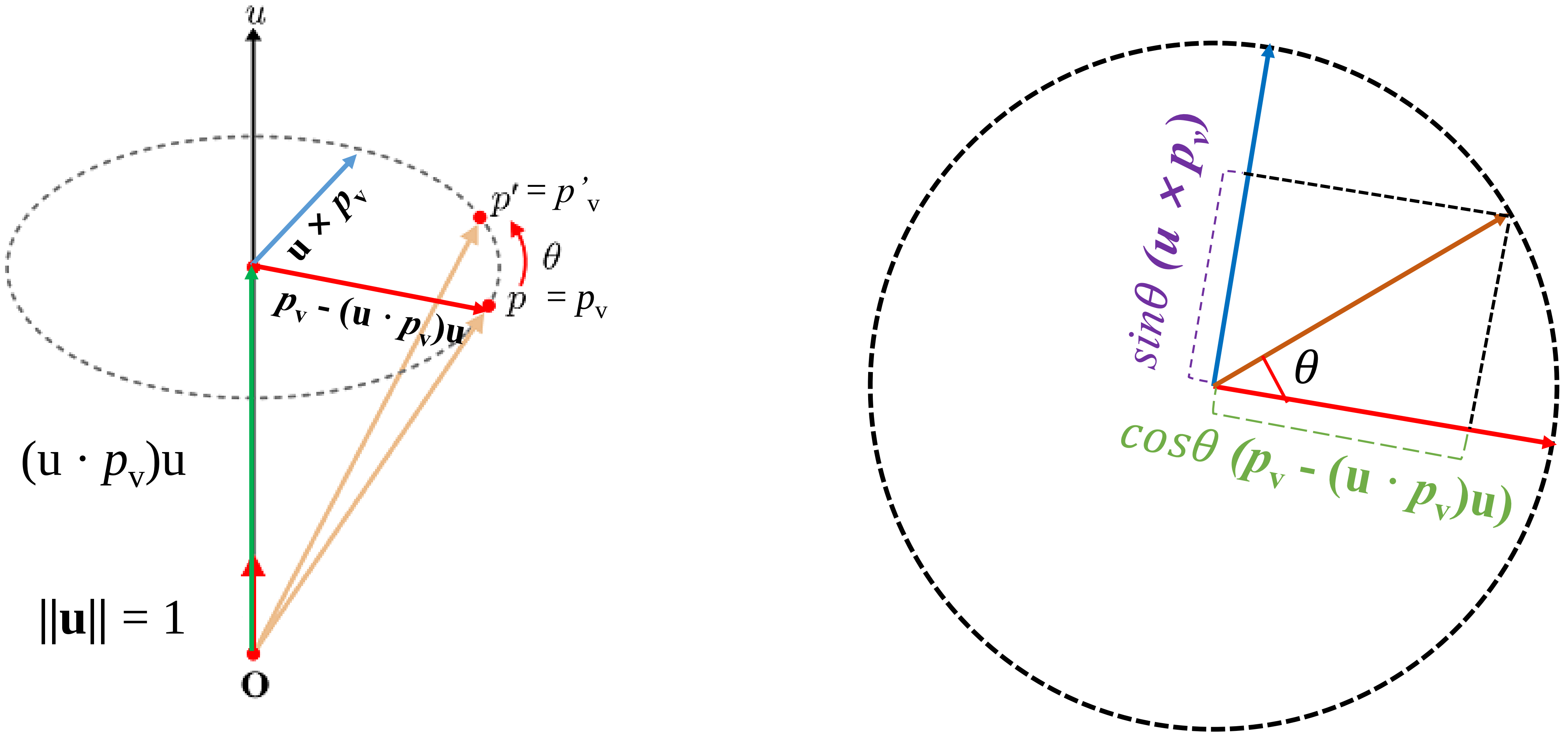
$\mathbf{u} \times p_v = \mathbf{u} \times \{p_v - (\mathbf{u} \cdot p_v)\mathbf{u}\}$



Recall that the magnitude of the cross product equals the area of the parallelogram spanned by **u** and p_v.

$$((\mathrm{u} \cdot p_v)\mathrm{u} + \cos\theta(p_v - (\mathrm{u} \cdot p_v)\mathrm{u}) + \sin\theta(\mathrm{u} \times p_v),0)$$

3D rotation through complex numbers



$$p` = ((u \cdot p_v)u + \cos\theta(p_v - (u \cdot p_v)u) + \sin\theta(u \times p_v),0)$$

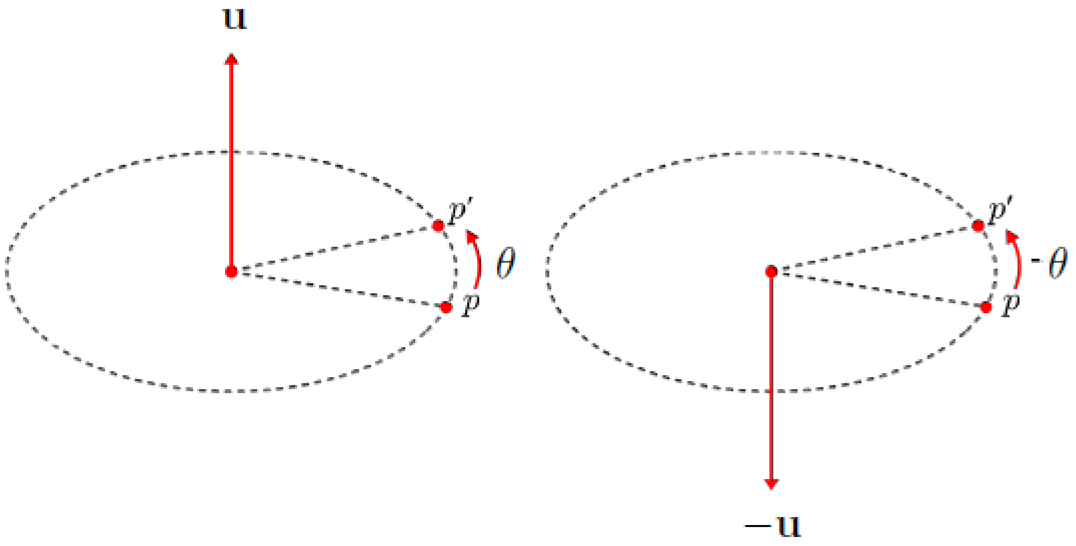
3D rotation through complex numbers

Multiple Quaternions

- Let $\mathbf{p}'$ denote $\mathbf{qpq^*}$. It represents the rotated vector p'. Consider rotating p' by another quaternion $\mathbf{r}$.
- The combined rotation is represented in $\mathbf{rq}$.
  - $\mathbf{rp'r^* = r(qpq^*)r^* = (rq)p(q^*r^*) = (rq)p(rq)^*}$

Quaternion and Negation

- "Rotation about $\mathbf{u}$ by θ" is identical to "rotation about $-\mathbf{u}$ by $-\theta$."



Quaternion and Matrix

A quaternion **q** representing a rotation can be converted into a matrix form. If **q** = (q_x,q_y,q_z,q_w), the rotation matrix is defined as follows:

$$\begin{pmatrix} 1-2(q_y^2+q_z^2) & 2(q_xq_y-q_wq_z) & 2(q_xq_z+q_wq_y) & 0 \\ 2(q_xq_y+q_wq_z) & 1-2(q_x^2+q_z^2) & 2(q_yq_z-q_wq_x) & 0 \\ 2(q_xq_z-q_wq_y) & 2(q_yq_z+q_wq_x) & 1-2(q_x^2+q_y^2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Conversely, given a rotation matrix, we can compute its quaternion. It requires us to extract $\{q_x,q_y,q_z,q_w\}$ given the above matrix.

- Compute the sum of all diagonal elements.

$$4-4(q_x^2+q_y^2+q_z^2) = 4-4(1-q_w^2) = 4q_w^2$$

- So, we obtain q_w.
- Subtract m_{12} from m_{21} of the above matrix.

$$m_{21}-m_{12} = 2(q_xq_y+q_wq_z)-2(q_xq_y-q_wq_z) = 4q_wq_z$$

- As we know q_w, we can compute q_z. Similarly, we can compute q_x and q_y.