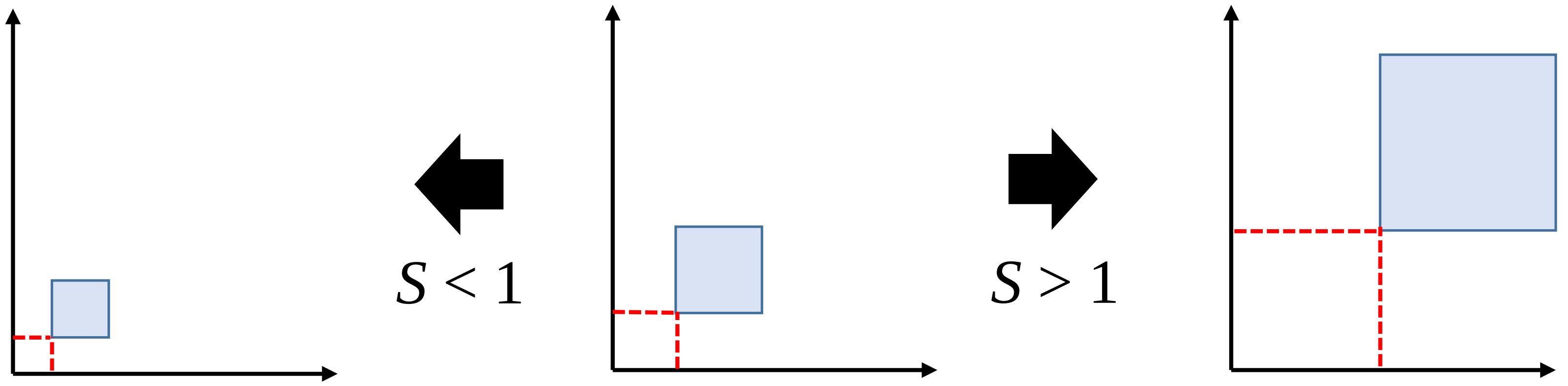
3. Spaces and Transforms

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

Scaling

What is a scaling?

- The scaling is a process of altering the size of objects.
- Scaling operation can be achieved by multiplying each vertex coordinate by scaling factor, S.
    - 2D scaling example: $(x, y) \times (s_x, s_y)$
    - 3D scaling example: $(x, y, z) \times (s_x, s_y, s_z)$
- If the scaling factor is less than 1, the size of the object will be reduced.
- If the scaling factor is greater than 1, the size of the object will be increased.



Scaling

2D scaling

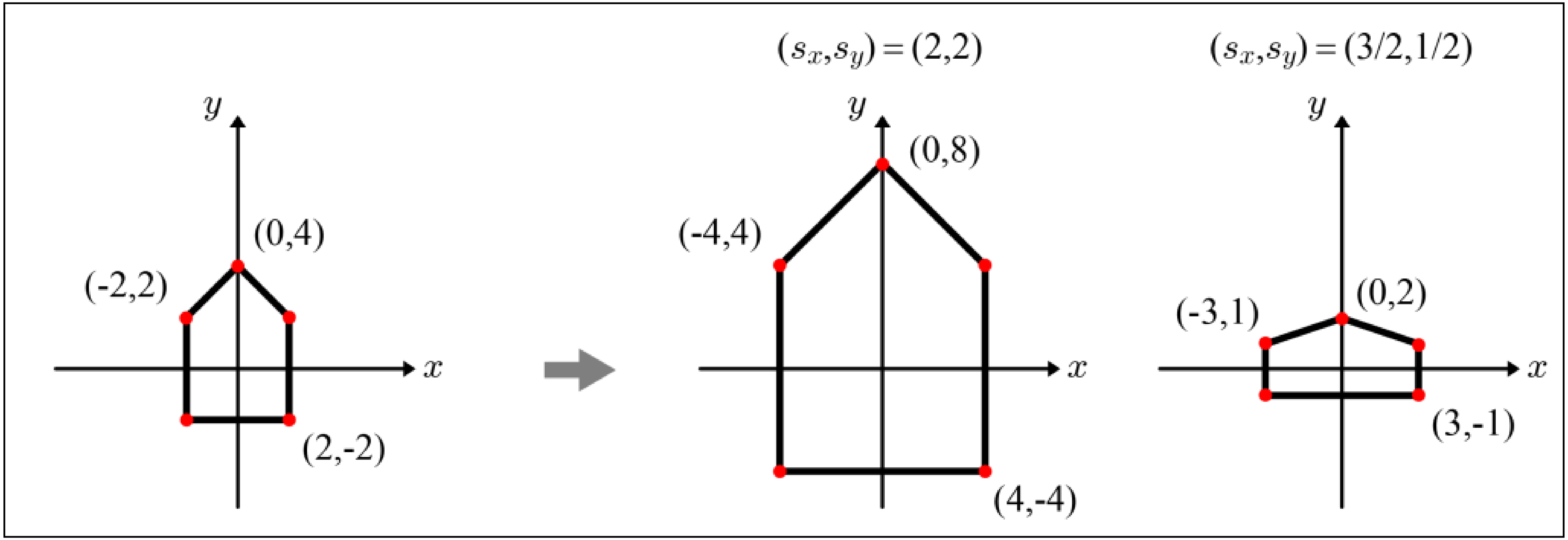
- 2D scaling, denoted as *S*, is represented by a 2 × 2 matrix.

$$S = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix}$$

- S_x and S_y are the scaling factors along the *x*-axis and *y*-axis, respectively.
- A 2D vector, (*x*, *y*) is scaled through matrix-vector multiplication.

$$\begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} s_x x \\ s_y y \end{pmatrix}$$

Scaling



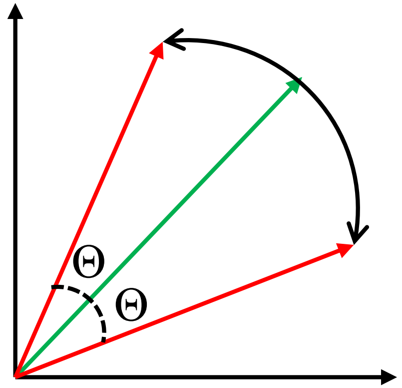
2D scaling examples: every vertex of a polygon is multiplied by the same scaling matrix

Rotation

What is a rotation?

- Rotation is the process of rotating an object around an axis of rotation.
- There are two types of rotations depending on the direction of the movement:
    - Clockwise rotation
    - Counterclockwise (anti-clockwise) rotation
- Mostly, the positive value of the rotation angle rotates an object in the counterclockwise direction.

counterclockwise rotation



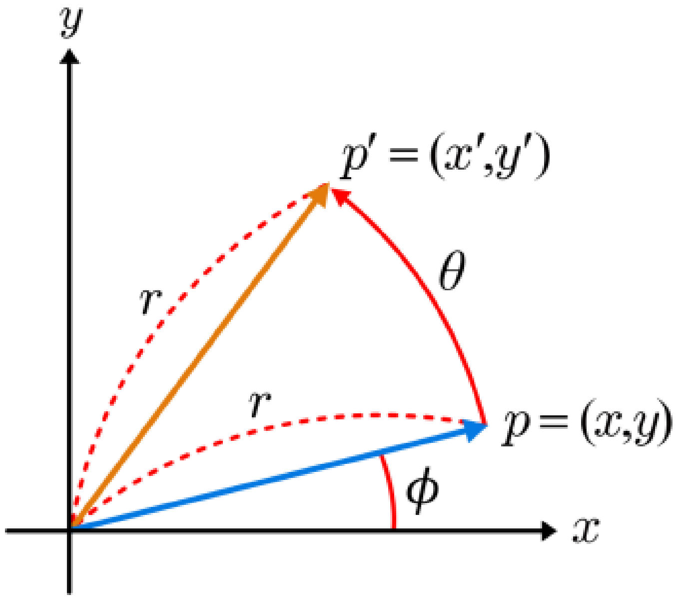
clockwise rotation

Rotation

Let's consider the 2D rotation of a vector.

- Vector p is rotated about the origin by θ to define p'.
- When the length of p is r, its coordinates are defined as follows:

$$x = r cos\phi$$
$$y = r sin\phi$$



- Then, the coordinates of p' can be computed as follows:

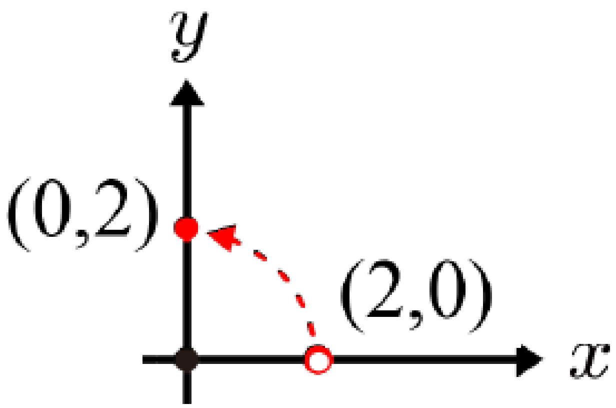
$$\begin{aligned} x' &= r cos(\phi + \theta) \\ &= r cos\phi cos\theta - r sin\phi sin\theta \\ &= x cos\theta - y sin\theta \end{aligned} \qquad \begin{aligned} y' &= r sin(\phi + \theta) \\ &= r cos\phi sin\theta + r sin\phi cos\theta \\ &= x sin\theta + y cos\theta \end{aligned}$$

- These can be combined into a matrix-vector multiplication form:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \underbrace{\begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix}}_{R(\theta)} \begin{pmatrix} x \\ y \end{pmatrix}$$

Rotation

An example



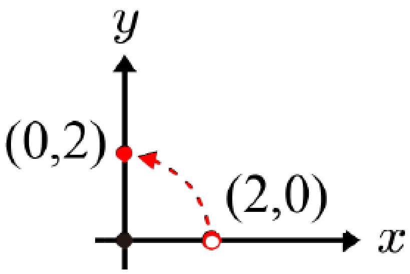
$$R(90^\circ) = \begin{pmatrix} cos(90^\circ) & -sin(90^\circ) \\ sin(90^\circ) & cos(90^\circ) \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

Rotation

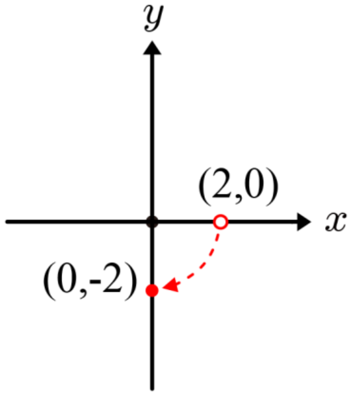
Counter-clockwise rotation and *clockwise rotation*

- By default, the rotation is *counter-clockwise*.



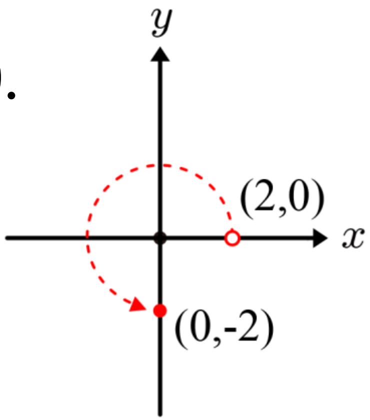
- The matrix for the *clockwise rotation* by θ is obtained by inserting -θ into the 2D rotation matrix.

$$R(-90°) = \begin{pmatrix} cos(-90°) & -sin(-90°) \\ sin(-90°) & cos(-90°) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$



- Note that rotation by -θ is equivalent to rotation by 2π-θ: 360 – 90 = 270.

$$R(270°) = \begin{pmatrix} cos270° & -sin270° \\ sin270° & cos270° \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$



Translation

Unlike *scaling* and *rotation*, *translation* is represented as *vector addition*.

- Translation displaces a point at (x, y) to $(x + d_x, y + d_y)$.
- We call (d_x, d_y) the translation vector.

$$\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \end{pmatrix} = \begin{pmatrix} x + dx \\ y + dy \end{pmatrix}$$

- Given the 2D Cartesian coordinates (x, y) of a point, we can take the 3D vector $(x, y, 1)$ as its homogeneous coordinates.
- Then, we can describe *translation* as *matrix multiplication*.

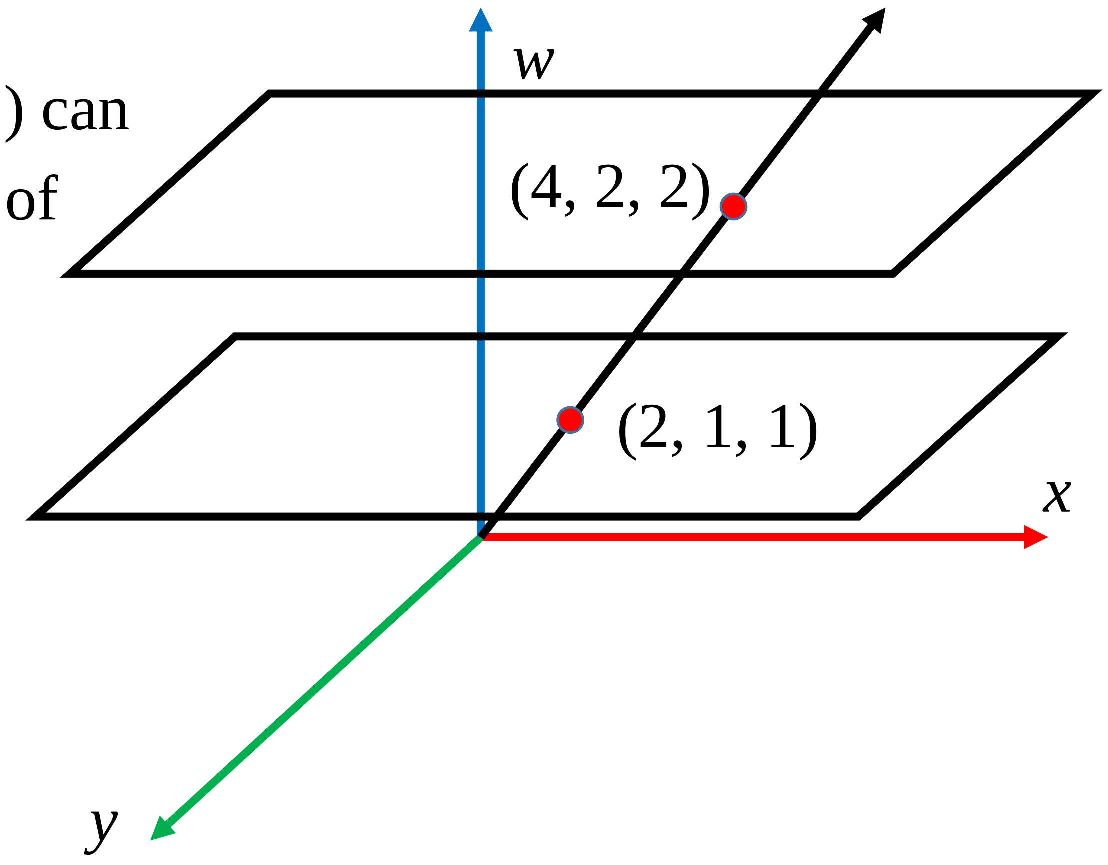
$$\begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + dx \\ y + dy \\ 1 \end{pmatrix}$$

Homogeneous coordinates

Homogeneous coordinates

- Given a 2D point, (x, y), its homogeneous coordinates are not necessarily $(x, y, 1)$ but (wx, wy, w) with non-zero w.

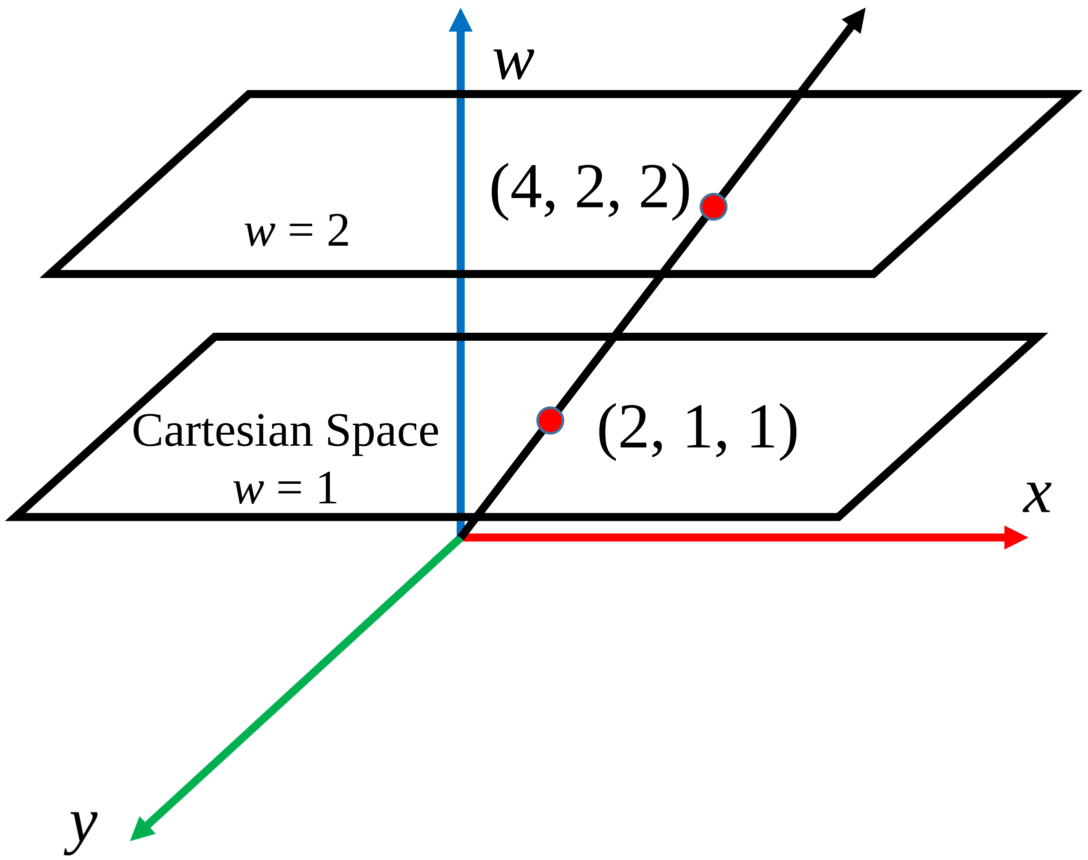
- For example, the Cartesian coordinates (2, 1) can be converted into homogeneous coordinates of (2, 1, 1), (4, 2, 2), (6, 3, 3), etc.



Homogeneous coordinates

Homogeneous coordinates

- Suppose that we are given the homogeneous coordinates, (X, Y, w).
- By dividing every coordinate by w, we obtain $(X/w, Y/w, 1)$.
- This corresponds to projecting a point on the line onto the plane, $w = 1$.
- We then take the first two components, $(X/w, Y/w)$, as the Cartesian coordinates.



Homogeneous coordinates

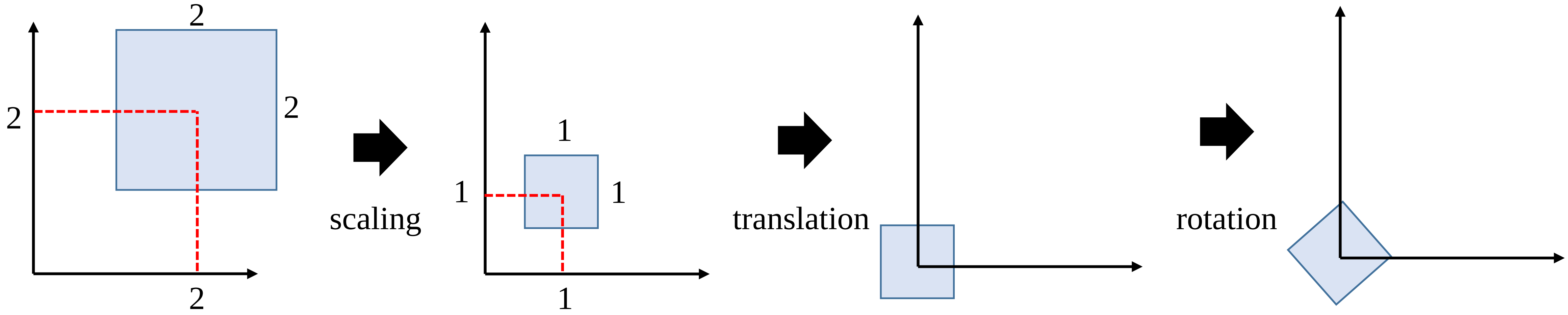
Homogeneous coordinates

- For handling the homogeneous coordinates, the 3×3 matrices for scaling and rotation need to be altered.

$$\begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \Rightarrow \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix} \Rightarrow \begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

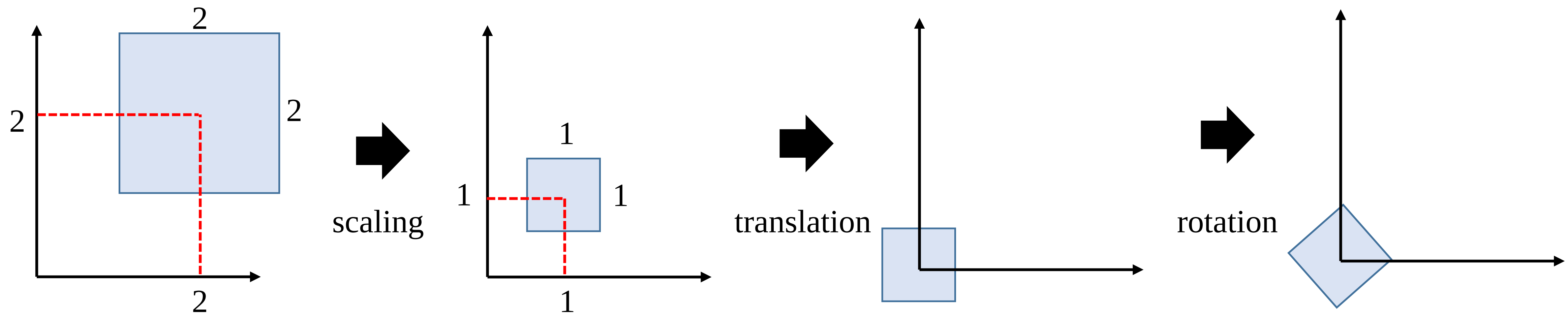
Practice

1. Write the 3×3 matrices for scaling, translation and rotation.



Practice - Solution

1. Write the 3×3 matrices for scaling, translation and rotation.

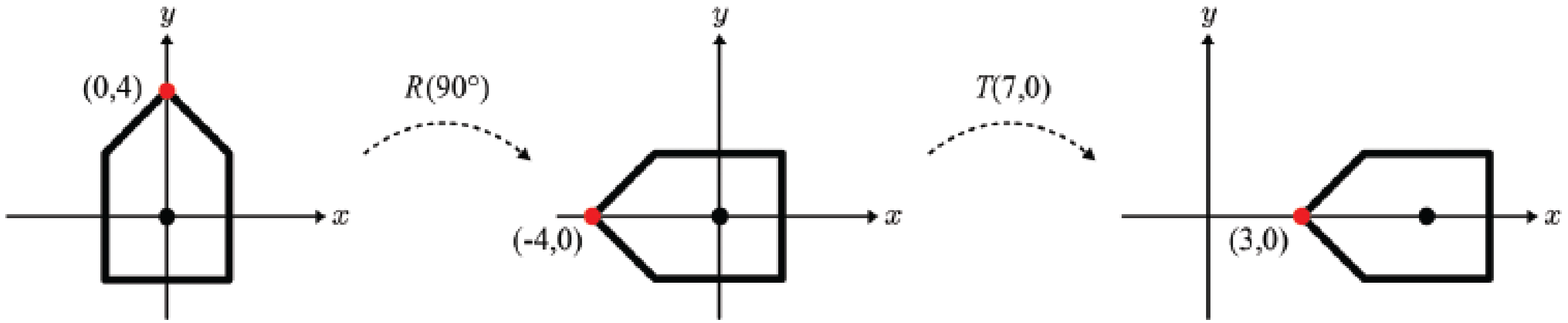


$$\begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} \cos 45^\circ & -\sin 45^\circ & -1 \\ \sin 45^\circ & \cos 45^\circ & -1 \\ 0 & 0 & 1 \end{pmatrix}$$

Composition of 2D Transform

An object may go through multiple transforms.

- Let’s rotate and translate the polygon.

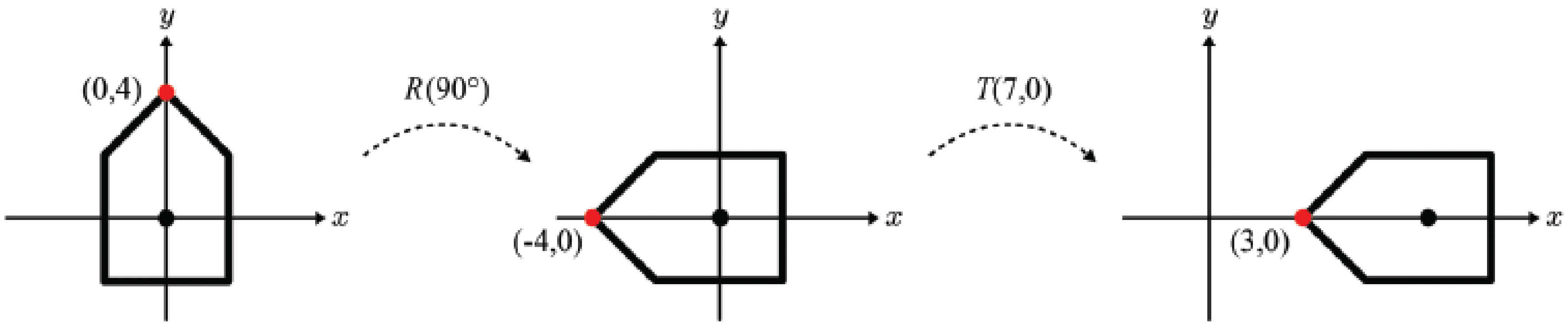


- We denote the rotation by $R(90^\circ)$ and the translation by $T(7, 0)$:

$$R(90^\circ) = \begin{pmatrix} cos90^\circ & -sin90^\circ & 0 \\ sin90^\circ & cos90^\circ & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad T(7,0) = \begin{pmatrix} 1 & 0 & 7 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Composition of 2D Transform

An object may go through multiple transforms.



- The vertex located at (0, 4) is rotated to (-4, 0) by *R*(90˚). Then, the vertex at (-4, 0) is translated to (3, 0) by *T*(7, 0).

$$\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 4 \\ 1 \end{pmatrix} = \begin{pmatrix} -4 \\ 0 \\ 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 7 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -4 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix}$$

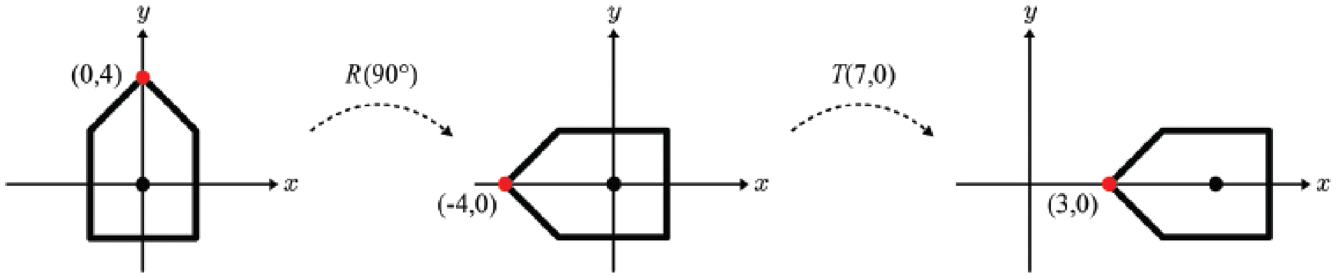
Composition of 2D Transform

Combine matrices

- As both $R(90^\circ)$ and $T(7, 0)$ are represented in 3×3 matrices, they can be concatenated to make a 3×3 matrix:

$$T(7,0)R(90^\circ) = \begin{pmatrix} 1 & 0 & 7 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 7 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- The vertex originally located at $(0, 4)$ is instantly transformed to $(3, 0)$ by the combined matrix.



$$\begin{pmatrix} 0 & -1 & 7 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 4 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix}$$

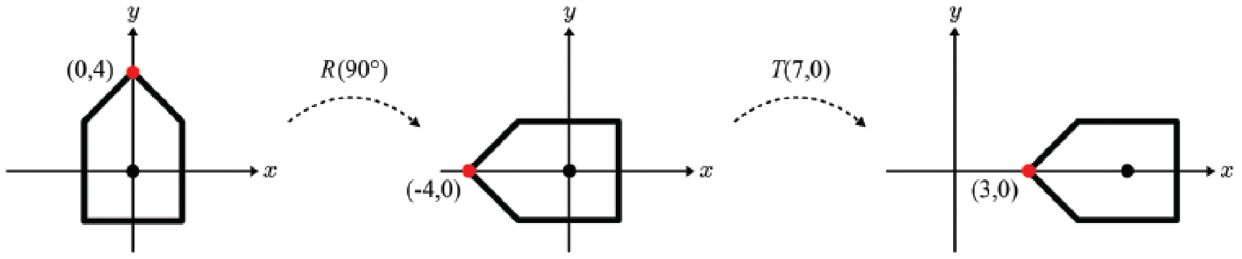
Composition of 2D Transform

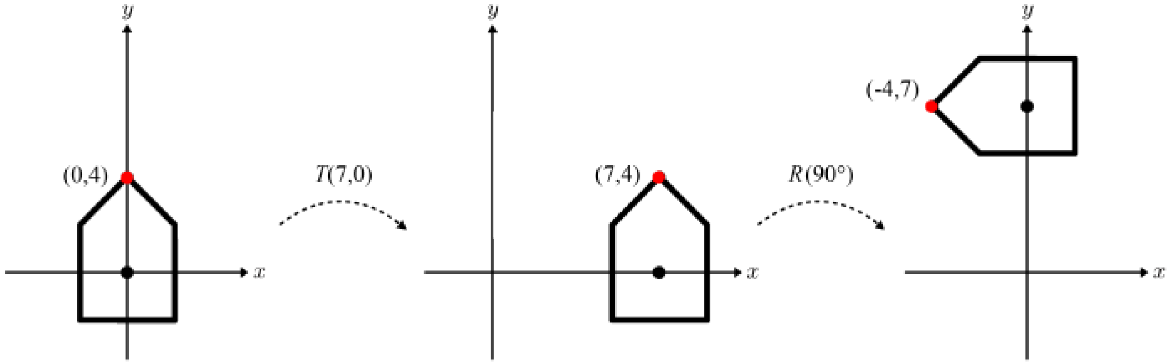
Noncommutative

- Matrix multiplication is noncommutative, which means “order matters”.
- Rotation (*R*) followed by translation (*T*) vs. *T* followed by *R*

$$TR = \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta & d_x \\ sin\theta & cos\theta & d_y \\ 0 & 0 & 1 \end{pmatrix}$$

$$RT = \begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta & d_x cos\theta - d_y sin\theta \\ sin\theta & cos\theta & d_x sin\theta + d_y cos\theta \\ 0 & 0 & 1 \end{pmatrix}$$





Composition of 2D Transform

Rotation about an arbitrary point.

- The rotation we learned so far is “about the origin”.
- Now consider rotation about arbitrary point, which is not the origin.

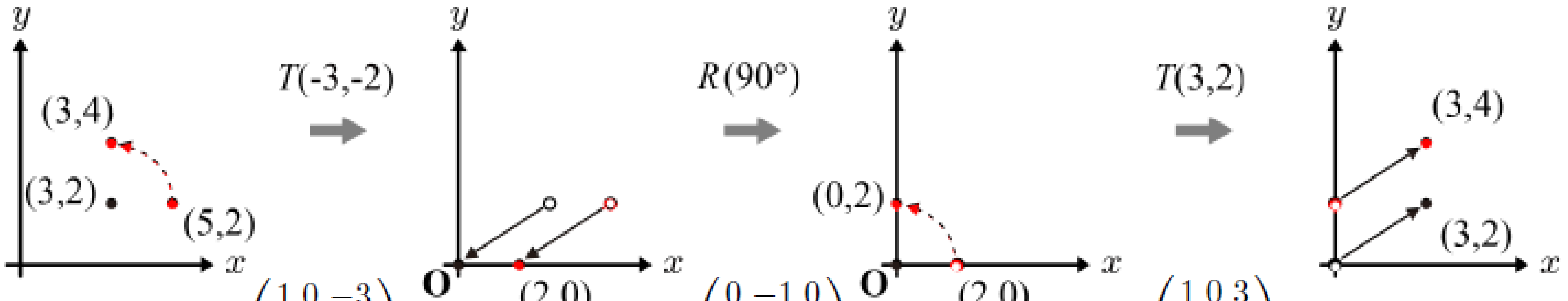
Rotating a point at (x, y) about an arbitrary point, (a, b)

- Translating (x, y) by $(-a, -b)$
- Rotating the translated point about the origin
- Back-translating the rotated point by (a, b)

Composition of 2D Transform

Example: rotation (5, 2) about (3, 2)

- Translating (5, 2) by (-3, -2)
- Rotating the translated point about the origin
- Back-translating the rotated point by (3, 2)



$$\begin{pmatrix}1&0&-3\\0&1&-2\\0&0&1\end{pmatrix} \quad \begin{pmatrix}0&-1&0\\1&0&0\\0&0&1\end{pmatrix} \quad \begin{pmatrix}1&0&3\\0&1&2\\0&0&1\end{pmatrix}$$

$$\begin{pmatrix}1&0&3\\0&1&2\\0&0&1\end{pmatrix}\begin{pmatrix}0&-1&0\\1&0&0\\0&0&1\end{pmatrix}\begin{pmatrix}1&0&-3\\0&1&-2\\0&0&1\end{pmatrix} = \begin{pmatrix}0&-1&5\\1&0&-1\\0&0&1\end{pmatrix} \Rightarrow \begin{pmatrix}0&-1&5\\1&0&-1\\0&0&1\end{pmatrix}\begin{pmatrix}5\\2\\1\end{pmatrix} = \begin{pmatrix}3\\4\\1\end{pmatrix}$$

Affine transform

Affine transform

- Linear transform (Scaling, Rotation, etc.)
- Translation (Translation does not fall into the linear transform class)

No matter how many affine matrices are given, they can be combined into a matrix.

$$RT = \begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta & d_x cos\theta - d_y sin\theta \\ sin\theta & cos\theta & d_x sin\theta + d_y cos\theta \\ 0 & 0 & 1 \end{pmatrix}$$

$$SRT = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} cos\theta & -sin\theta & d_x cos\theta - d_y sin\theta \\ sin\theta & cos\theta & d_x sin\theta + d_y cos\theta \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s_x cos\theta & -s_x sin\theta & s_x d_x cos\theta - s_x d_y sin\theta \\ s_y sin\theta & s_y cos\theta & s_y d_x sin\theta + s_y d_y cos\theta \\ 0 & 0 & 1 \end{pmatrix}$$

Affine transform

Ignoring the third row, we often denote the remaining 2×3 elements $[L|t]$, where L is a 2×2 matrix and t is a 2D column vector.

- L represents the 'combined' linear transform.
- In contrast, t represents a 'combined' translation, which may contain the input linear-transform terms.

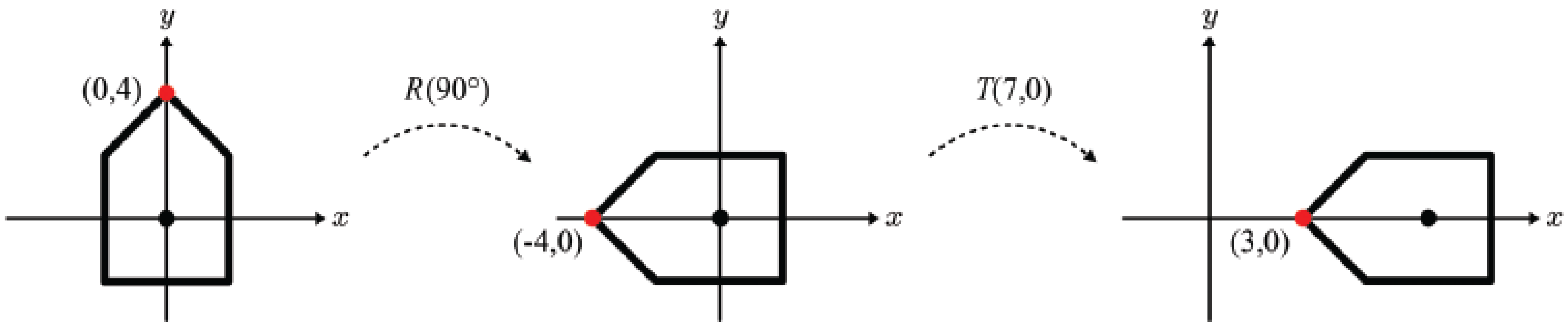
$$TR = \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta & d_x \\ sin\theta & cos\theta & d_y \\ 0 & 0 & 1 \end{pmatrix}$$

$$RT = \begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta & d_x cos\theta - d_y sin\theta \\ sin\theta & cos\theta & d_x sin\theta + d_y cos\theta \\ 0 & 0 & 1 \end{pmatrix}$$

$$SRT = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} cos\theta & -sin\theta & d_x cos\theta - d_y sin\theta \\ sin\theta & cos\theta & d_x sin\theta + d_y cos\theta \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s_x cos\theta & -s_x sin\theta & s_x d_x cos\theta - s_x d_y sin\theta \\ s_y sin\theta & s_y cos\theta & s_y d_x sin\theta + s_y d_y cos\theta \\ 0 & 0 & 1 \end{pmatrix}$$

Affine transform

Revisit $T(7,0)R(90^o)$.



$$TR = \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

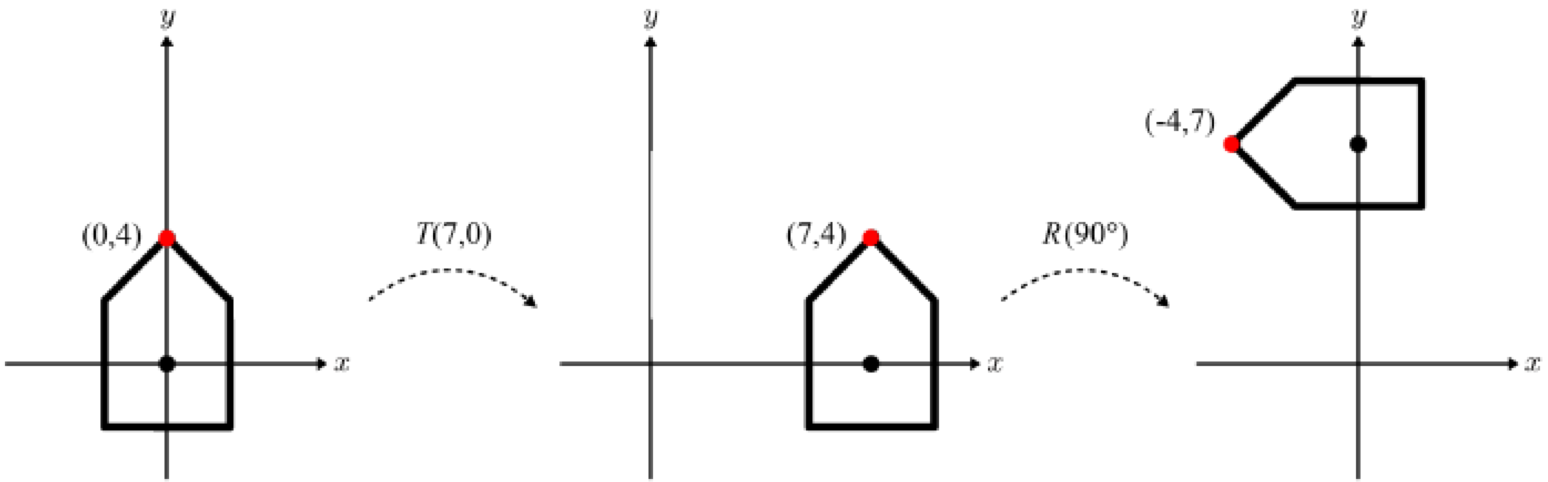
$$= \begin{pmatrix} cos\theta & -sin\theta & d_x \\ sin\theta & cos\theta & d_y \\ 0 & 0 & 1 \end{pmatrix}$$

$$T(7,0)R(90^\circ) = \begin{pmatrix} 1 & 0 & 7 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & -1 & 7 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \leftarrow [\boldsymbol{L}|\boldsymbol{t}]$$

Affine transform

Revisit $R(90^o)T(7,0)$.



$$RT = \begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} cos\theta & -sin\theta & d_x cos\theta - d_y sin\theta \\ sin\theta & cos\theta & d_x sin\theta + d_y cos\theta \\ 0 & 0 & 1 \end{pmatrix}$$

$$R(90^\circ)T(7,0) = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 7 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 7 \\ 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 7 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Conceptual decomposition of [***L***|***t***]

- [***L***|***t***] transforms a point, *p*, is ***L****p*+***t***.
- *L* is applied first.
- The linear-transformed object is translated by *t*.

Rigid Motion

Consider a combination of rotations and translations only, e.g., no scaling is involved.

- When the combined affine matrix applies to an object, the pose (position + orientation) of the object is changed but its shape is not.
- In this sense, the transform is named *rigid-body motion* or simply *rigid motion*.

No matter how many rotations and translations are combined, the resulting matrix is of the structure, $[\boldsymbol{R}|\boldsymbol{t}]$.

- $\boldsymbol{R}$ represents the 'combined' rotation, which does not include any translation terms.
- $\boldsymbol{t}$ represents the 'combined' translation, which usually includes the rotation terms.

Transforming an object by $[\boldsymbol{R}|\boldsymbol{t}]$ is conceptually decomposed into two steps: $\boldsymbol{R}$ is applied first and then the rotated object is translated by $\boldsymbol{t}$, i.e., the way $[\boldsymbol{R}|\boldsymbol{t}]$ transforms a point, p, is $\boldsymbol{R}p+\boldsymbol{t}$.

Practice

1. For matrix-vector multiplication, let us use row vectors ($w = 1$).

   - Write the translation matrix that translates (x, y) by (dx, dy).
   - Write the rotation matrix that rotates (x, y) by θ.
   - Write the scaling matrix with scaling factors s_x and s_y.

Practice - Solution

1. For matrix-vector multiplication, let us use row vectors (w = 1).
   - Write the translation matrix that translates (x, y) by (dx, dy).

$$\begin{pmatrix} x & y & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ dx & dy & 1 \end{pmatrix}$$

   - Write the rotation matrix that rotates (x, y) by θ.

$$\begin{pmatrix} x & y & 1 \end{pmatrix} \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

   - Write the scaling matrix with scaling factors s_x and s_y.

$$\begin{pmatrix} x & y & 1 \end{pmatrix} \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Practice

1. Write a 3×3 transform matrix that rotates $(x, y)^T$ by 60°, and translates by $(2, -1)$.

2. Write a 3×3 transform matrix that rotates (x, y) by 60°, and translates by $(2, -1)$.

Practice - Solution

1. Write a 3×3 transform matrix that rotates $(x, y)^T$ by 60°, and translates by (2, -1).

$$\begin{pmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 2 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

2. Write a 3×3 transform matrix that rotates (x, y) by 60°, and translates by (2, -1).

$$\begin{pmatrix} x & y & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ 2 & -1 & 1 \end{pmatrix}$$

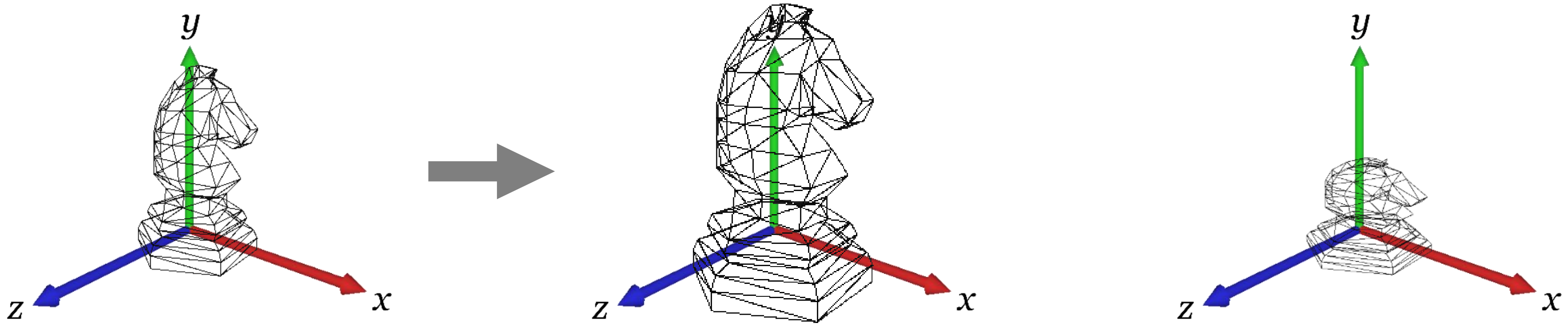
3D Scaling

3D scaling with the scaling factors, s_x ,s_y and s_z.

$$\begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} s_x x \\ s_y y \\ s_z z \end{pmatrix}$$

If all of the scaling factors are identical, the scaling is called uniform. Otherwise, it is a non-uniform scaling.

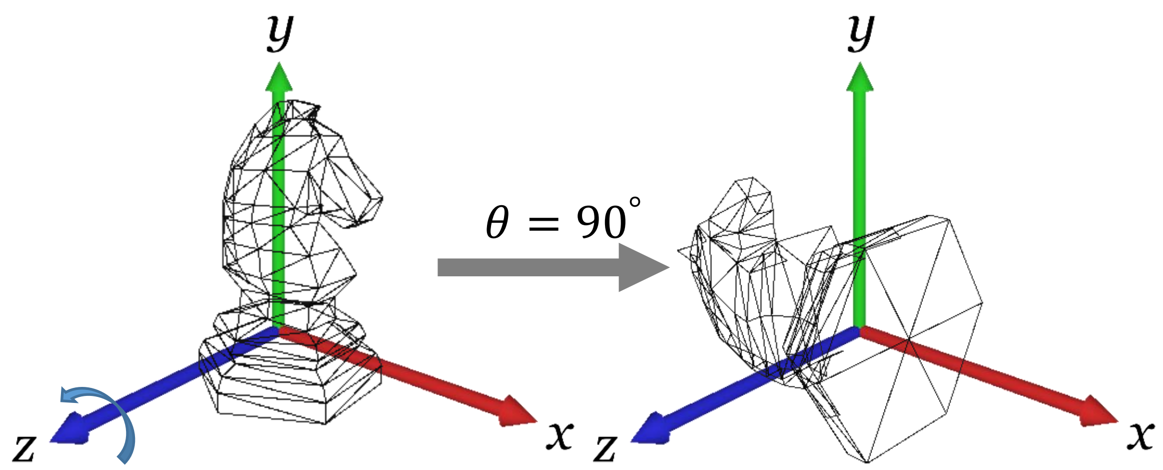
$(S_x ,S_y, S_z) = (1.5, 1.5, 1.5)$ $(S_x ,S_y, S_z) = (1, 0.5, 1)$

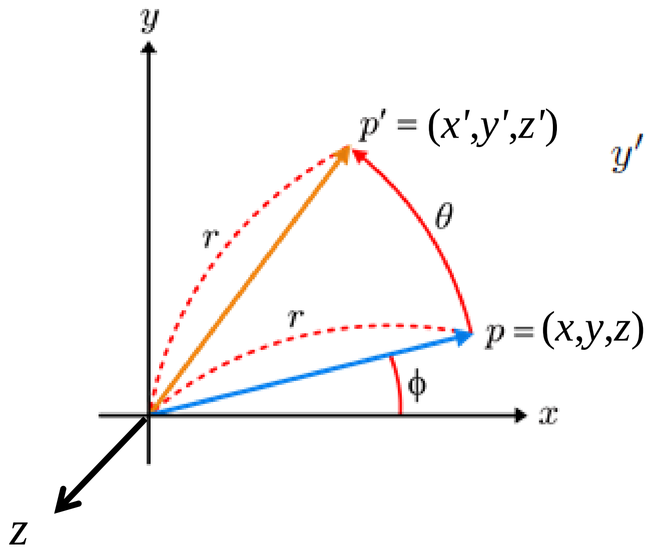


3D Rotation

Unlike 2D rotation which requires a center of rotation, 3D rotation requires an axis of rotation.

- Let's consider 3D rotations about x-axis (R_x), y-axis (R_y), and z-axis (R_z)
- Following example shows an rotation about R_z.





$$\begin{aligned} x' &= r\cos(\phi+\theta) \\ &= r\cos\phi\cos\theta - r\sin\phi\sin\theta \\ &= x\cos\theta - y\sin\theta \end{aligned}$$

$$\begin{aligned} y' &= r\sin(\phi+\theta) \\ &= r\cos\phi\sin\theta + r\sin\phi\cos\theta \\ &= x\sin\theta + y\cos\theta \end{aligned}$$

$$x' = x\cos\theta - y\sin\theta$$

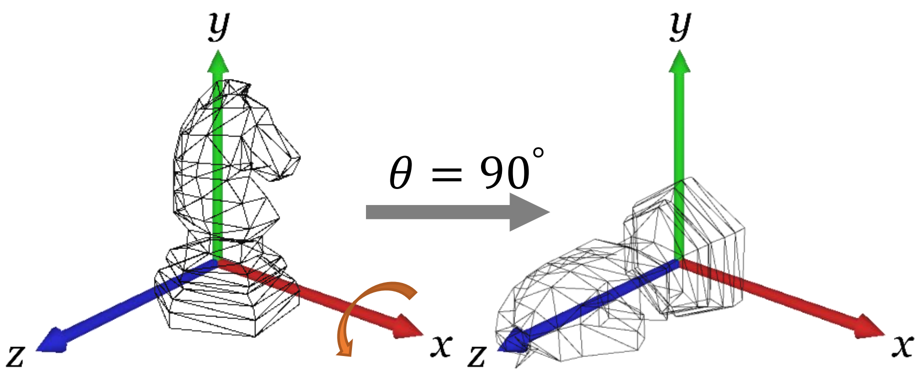
$$y' = x\sin\theta + y\cos\theta$$

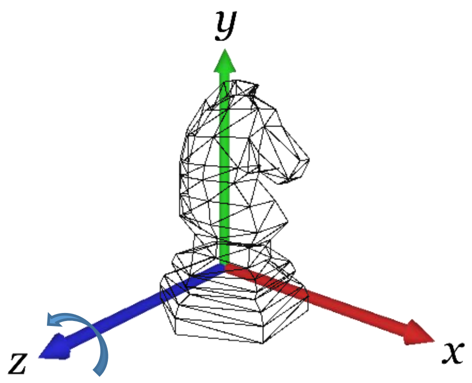
$$z' = z$$

$$\boxed{\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}}$$

3D Rotation

The following example shows a rotation about R_x.





$$\overset{y'}{x'} = \overset{y}{x}\cos\theta - \overset{z}{y}\sin\theta$$

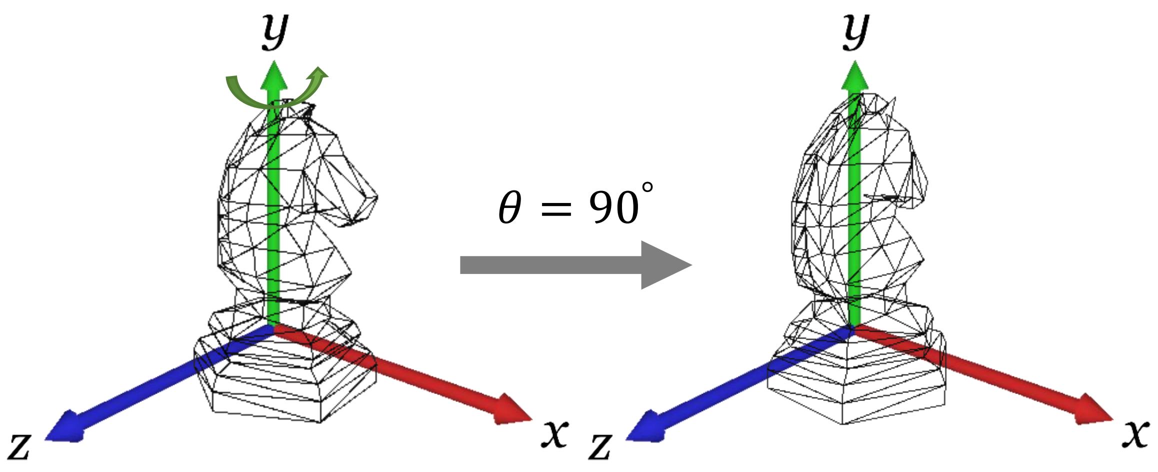
$$\underset{z'}{y'} = \underset{y}{x}\sin\theta + \underset{z}{y}\cos\theta$$

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \boxed{\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

- Obviously, $x' = x$.
- When the thumb of the right hand is aligned with the rotation axis, the other fingers curl from the *y*-axis to the *z*-axis.
- Returning to R_z, observe the fingers curl from the *x*-axis to the *y*-axis.
- Shifting from R_z to R_x, the *x*-axis is replaced with the *y*-axis, and the *y*-axis is replaced with the *z*-axis.
- By making such replacements, the matrix for R_x is obtained.

3D Rotation

The rotation matrix for R_y is as follows:

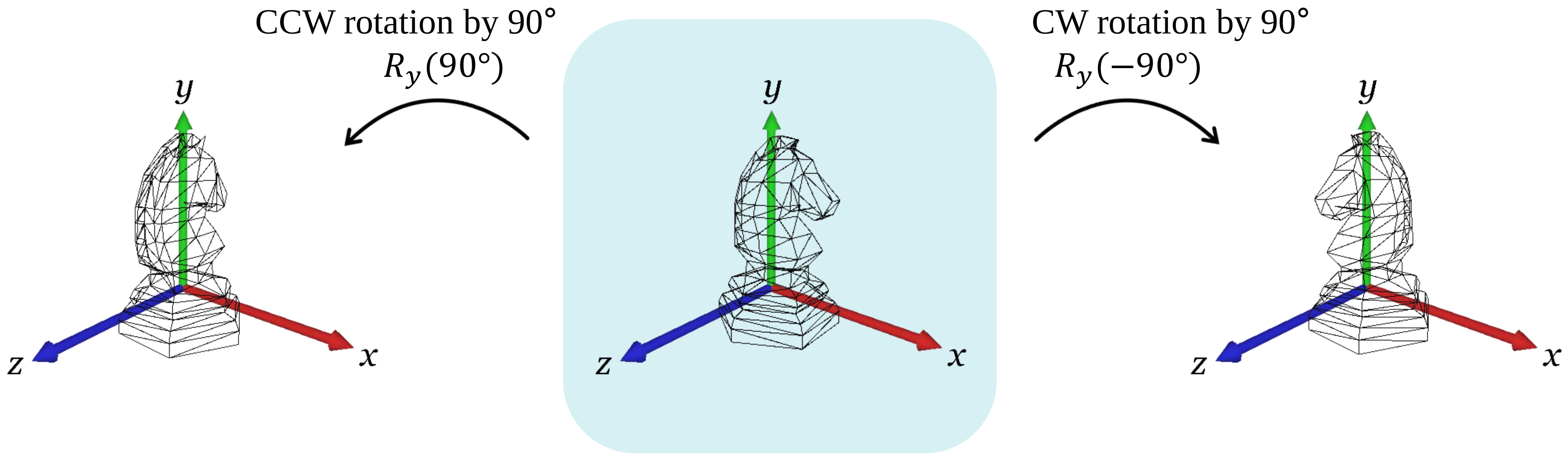


$$R_y = \begin{pmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{pmatrix}$$

3D Rotation

CCW vs. CW rotations

- If the rotation is CCW with respect to the axis pointing toward you, the rotation angle is positive.
- If the rotation is CW, its matrix is defined with the negated rotation angle.
- Note that rotation by -θ is equivalent to rotation by 2π-θ.



3D Translation

Recall that, translation is represented as vector addition.

- To represent translation as matrix multiplication, homogeneous coordinates is exploited.

$$\begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x + d_x \\ y + d_y \\ z + d_z \\ 1 \end{pmatrix}$$

The 3×3 matrices developed for 3D scaling and rotation are extended to 4×4 matrices. See the scaling example.

$$\begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

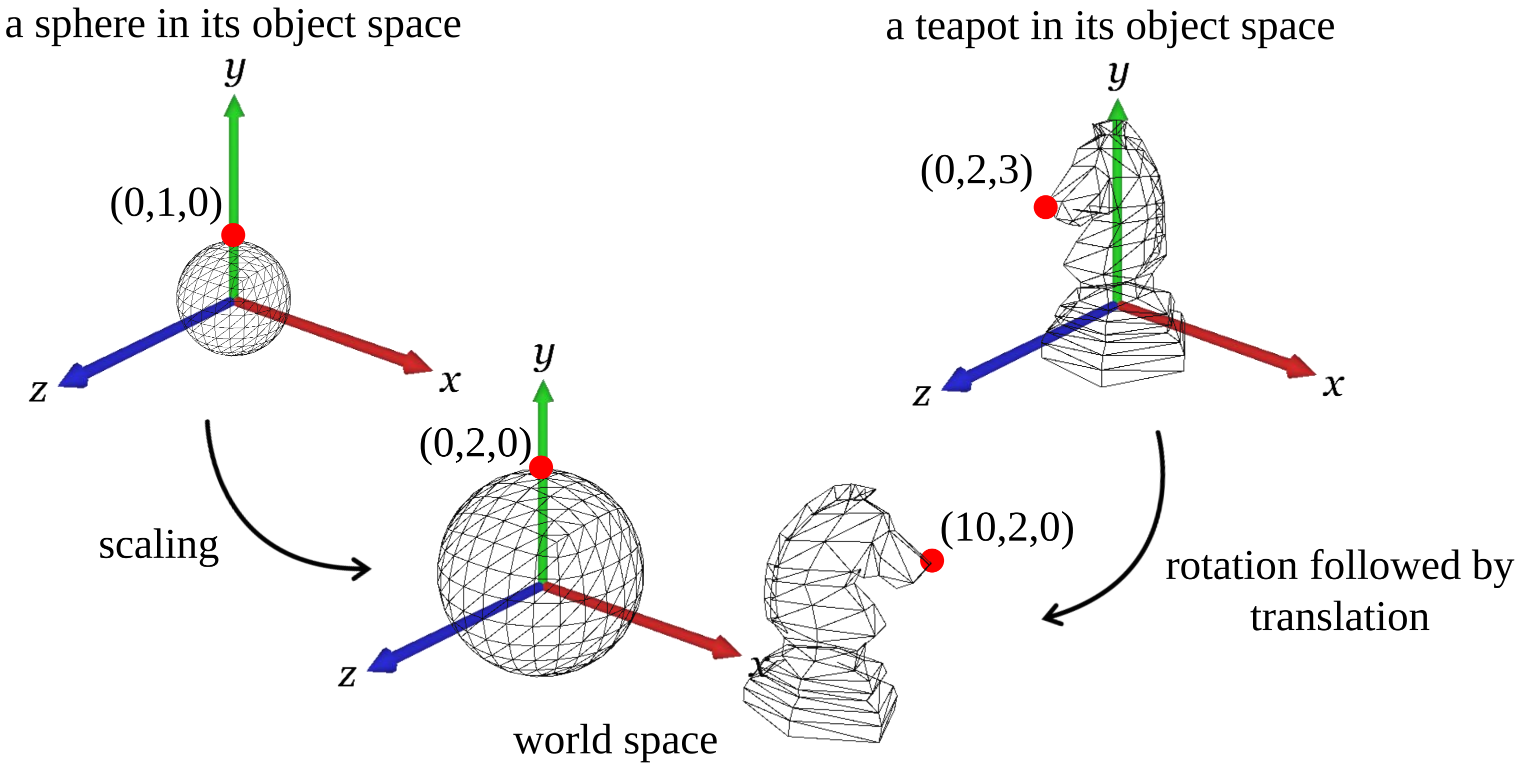
Application: World Transform

Object space vs. world space

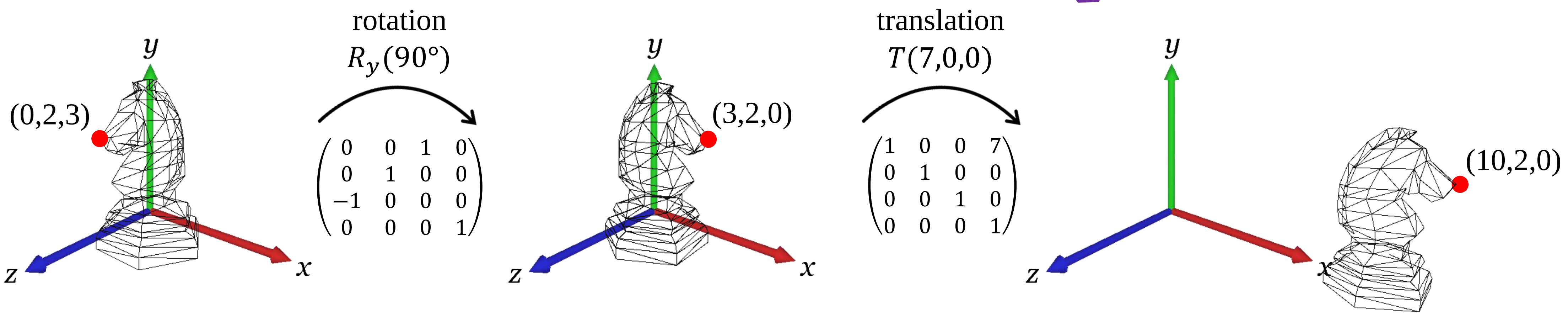
- The coordinate system used for creating an object is named object space.
- The object space for a model typically has no relationship to that of another model.
- The world transform 'assembles' all models into a single coordinate system called world space.

$$\begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 0 \\ 1 \end{pmatrix}$$



Application: World Transform



$$R_y(\theta) = \begin{pmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{pmatrix} \qquad R_y(90°) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 0 \\ 1 \end{pmatrix}$$

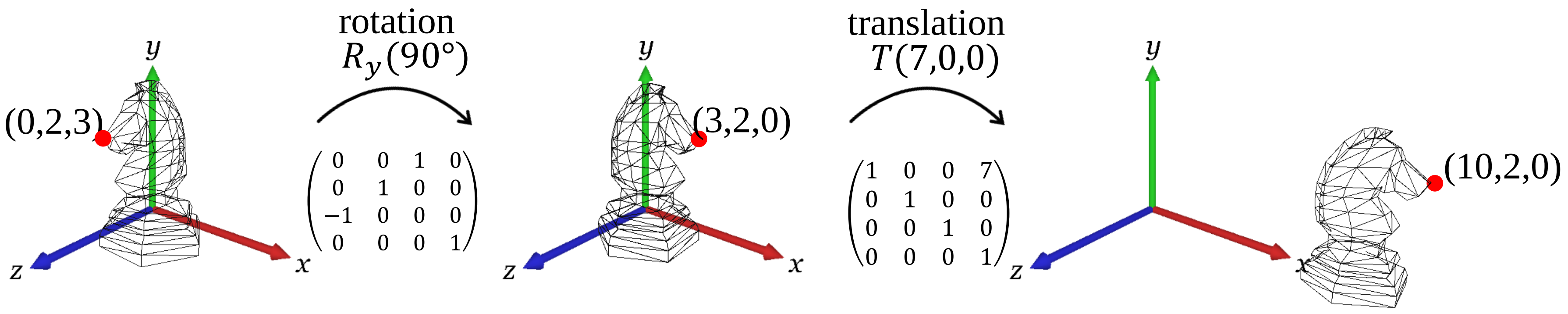
$$T(7,0,0) = \begin{pmatrix} 1 & 0 & 0 & 7 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 & 0 & 7 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 10 \\ 2 \\ 0 \\ 1 \end{pmatrix}$$

$$T(7,0,0)R_y(90°) = \begin{pmatrix} 1 & 0 & 0 & 7 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 7 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 0 & 1 & 7 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 10 \\ 2 \\ 0 \\ 1 \end{pmatrix}$$

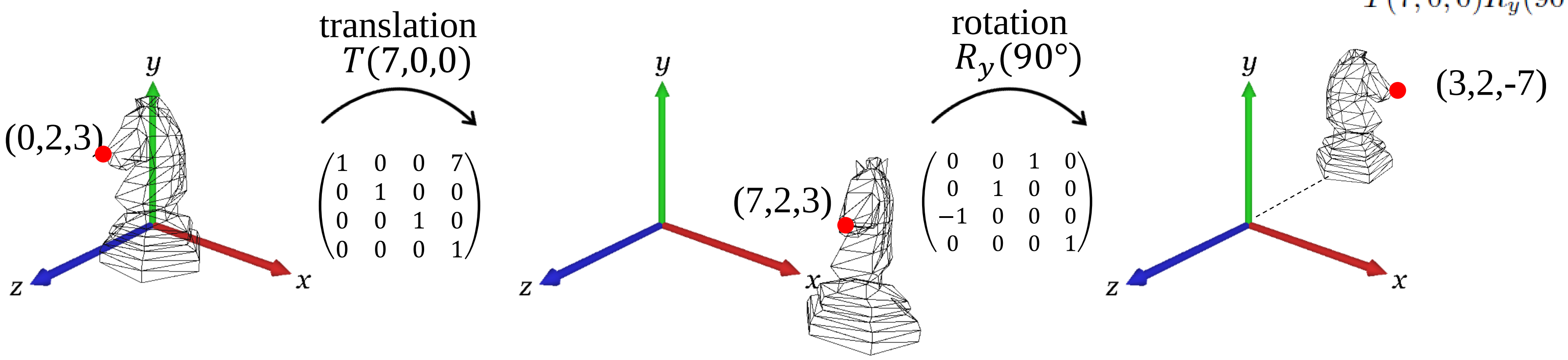
3D Affine Transforms

The discussions we had on 2D affine transforms apply to 3D affine transforms.

- Recall that the matrix multiplication is not commutative.



$$T(7,0,0)R_y(90°) = \begin{pmatrix} 1 & 0 & 0 & 7 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 7 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

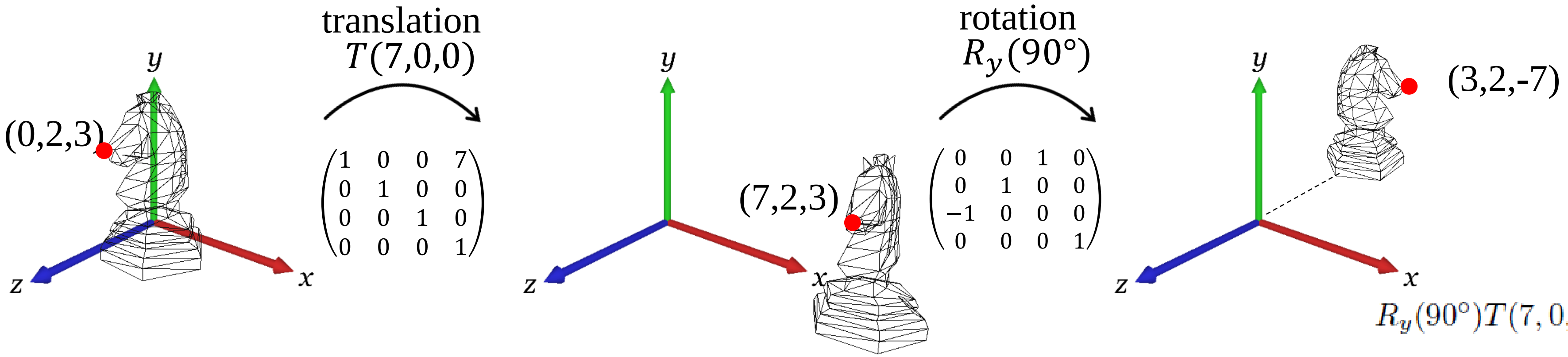


$$R_y(90°)T(7,0,0) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 7 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -7 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

3D Affine Transforms

Concatenating matrices

- When 3D scaling, rotation, and translation matrices are concatenated to make a 4×4 matrix, the fourth row is always (0 0 0 1).
- Ignoring the fourth row of an affine matrix, we often denote the remaining 3×4 elements by [$\boldsymbol{L}$|$\boldsymbol{t}$], where $\boldsymbol{L}$ is a 3×3 matrix that represents a 'combined' linear transform, and $\boldsymbol{t}$ is a 3D column vector that represents a 'combined' translation.
- [$\boldsymbol{L}$|$\boldsymbol{t}$] is conceptually decomposed into two steps: $\boldsymbol{L}$ is applied first and then the linear-transformed object is translated by $\boldsymbol{t}$.



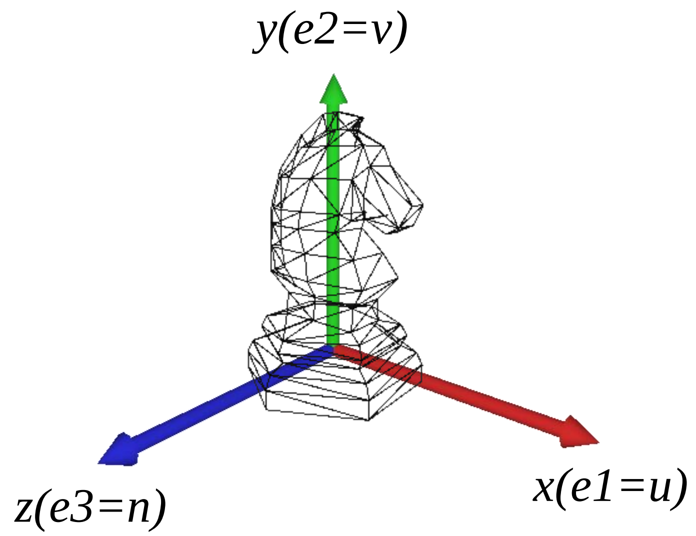
$$R_y(90°)T(7,0,0) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 7 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -7 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

[$\boldsymbol{L}$|$\boldsymbol{t}$]

Rotation and Object-space Basis

Basis of the world space and object space

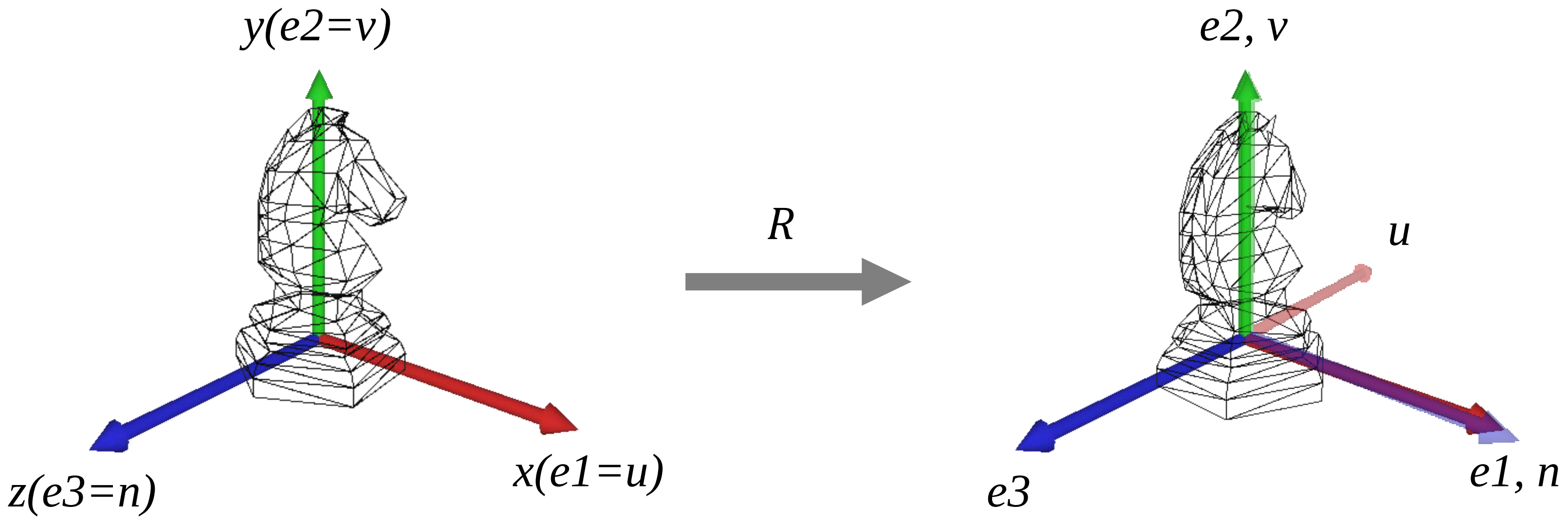
- $\{e_1, e_2, e_3\}$ represents the standard basis of the world space.
- $\{u, v, n\}$ represents the orthonormal basis of the object space.
- In the following figure, world space basis and object space basis are identical.



Rotation and Object-space Basis

Rotating Object-space basis

- A rotation applied to an object changes its orientation, which can be described by the '*rotated*' basis of the object space.
- Let R denote the rotation.



Rotation and Object-space Basis

R relates e_1 and u, which was initially identical to e_1.

$$Re_1 = R\begin{pmatrix}1\\0\\0\end{pmatrix} = \begin{pmatrix}u_x\\u_y\\u_z\end{pmatrix}$$

Similarly, R transforms e_2 and e_3 into v and n, respectively:

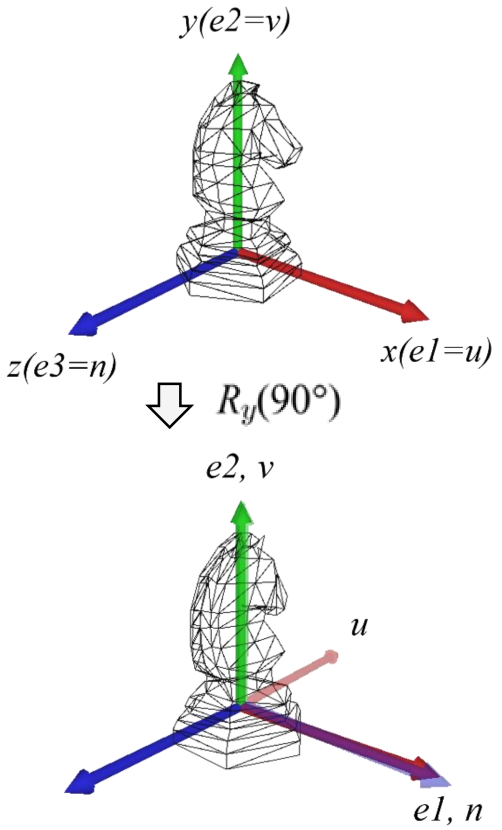
$$Re_2 = R\begin{pmatrix}0\\1\\0\end{pmatrix} = \begin{pmatrix}v_x\\v_y\\v_z\end{pmatrix} \qquad Re_3 = R\begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix}n_x\\n_y\\n_z\end{pmatrix}$$

The above three are combined:

$$R\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix} = \begin{array}{c}\begin{pmatrix}\boxed{\begin{matrix}u_x\\u_y\\u_z\end{matrix}} & \boxed{\begin{matrix}v_x\\v_y\\v_z\end{matrix}} & \boxed{\begin{matrix}n_x\\n_y\\n_z\end{matrix}}\end{pmatrix}\\ \begin{matrix}\;u\quad & \;v\quad & \;n\end{matrix}\end{array}$$

$$R_y(90^\circ) = \begin{pmatrix}cos90^\circ & 0 & sin90^\circ\\ 0 & 1 & 0\\ -sin90^\circ & 0 & cos90^\circ\end{pmatrix}$$

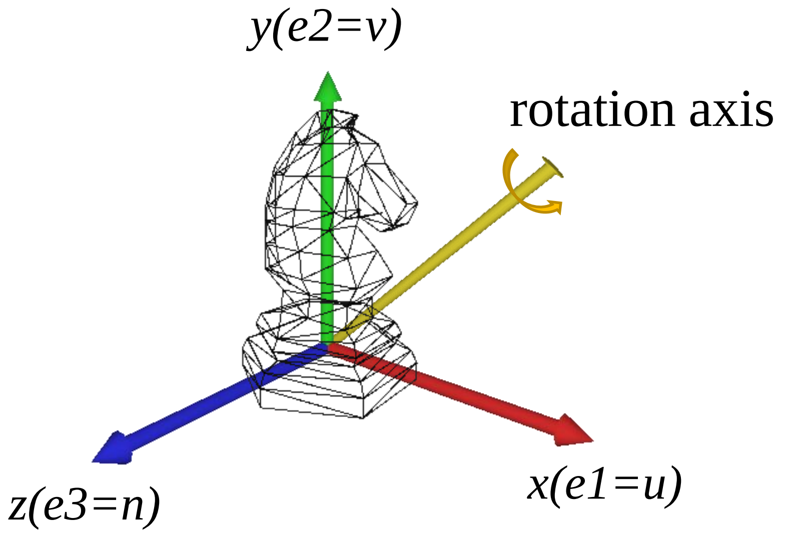
$$= \begin{pmatrix}0&0&1\\0&1&0\\-1&0&0\end{pmatrix}$$



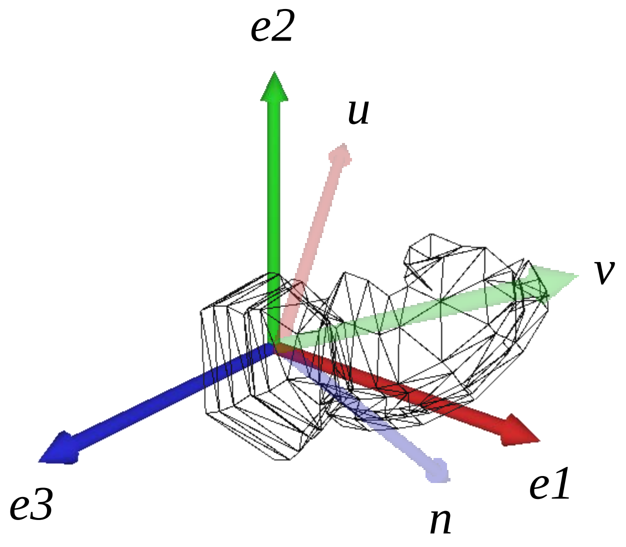
R's columns are u, v, and n. Given the '*rotated*' object-space basis, $\{u, v, n\}$, the rotation matrix is immediately determined, and vice versa.

Rotation and Object-space Basis

The observation we have made holds in general.



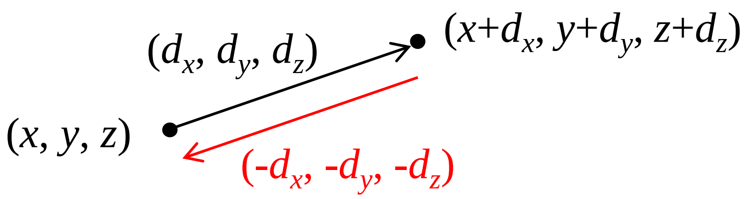
$$R = \begin{pmatrix} u_x & v_x & n_x \\ u_y & v_y & n_y \\ u_z & v_z & n_z \end{pmatrix}$$



Inverses of Translation and Scaling

Inverse translation

$$\begin{pmatrix} 1 & 0 & 0 & -d_x \\ 0 & 1 & 0 & -d_y \\ 0 & 0 & 1 & -d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \Leftrightarrow \begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



Inverse transform in inverse matrix

$$T^{-1}T = \begin{pmatrix} 1 & 0 & 0 & -d_x \\ 0 & 1 & 0 & -d_y \\ 0 & 0 & 1 & -d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = I$$

Inverse scaling

$$\begin{pmatrix} \frac{1}{s_x} & 0 & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 & 0 \\ 0 & 0 & \frac{1}{s_z} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \Leftrightarrow \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Inverse Rotation

Given a rotation matrix *R*, its columns (*u*, *v*, and *n*) make up an orthonormal basis, i.e., $u \cdot u = v \cdot v = n \cdot n = 1$ and $u \cdot v = v \cdot n = n \cdot u = 0$.

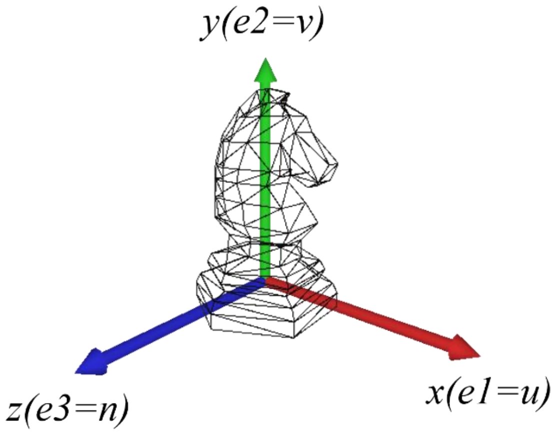
- Let's multiply *R*'s transpose (R^T) with *R*:

$$
\begin{aligned}
R^T R &= \begin{pmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ n_x & n_y & n_z \end{pmatrix} \begin{pmatrix} u_x & v_x & n_x \\ u_y & v_y & n_y \\ u_z & v_z & n_z \end{pmatrix} \\
&= \begin{pmatrix} u \cdot u & u \cdot v & u \cdot n \\ v \cdot u & v \cdot v & v \cdot n \\ n \cdot u & n \cdot v & n \cdot n \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
&= I
\end{aligned}
$$

- This says that $R^{-1} = R^T$, i.e., the inverse of a rotation matrix is its transpose.
- Because *u*, *v*, and *n* form the columns of *R*, they form the rows of R^{-1}.

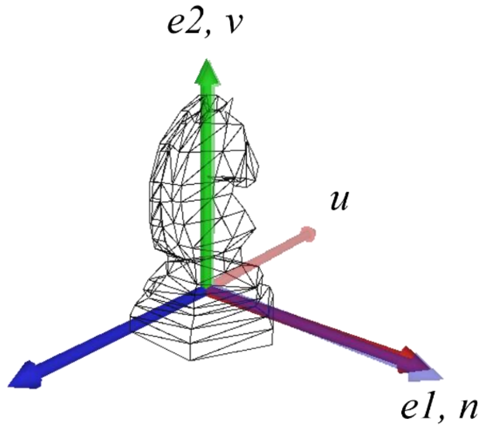
Inverse Rotation

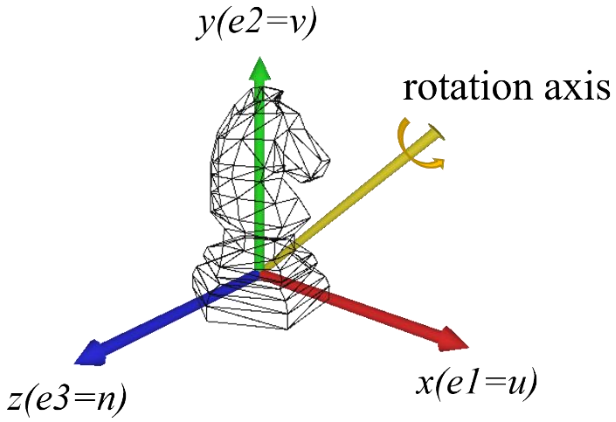
Recall that u, v, and n form the columns of R. As $R^{-1}=R^T$, u, v, and n form the rows of R^{-1}.



$$R_y(90°) = \begin{pmatrix} u & v & n \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

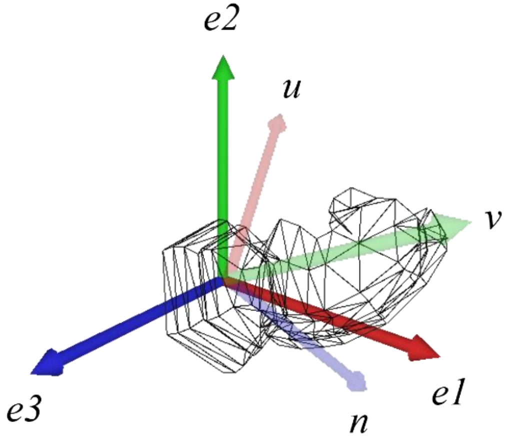
$$R_y^{-1}(90°) = \begin{pmatrix} u \\ v \\ n \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$





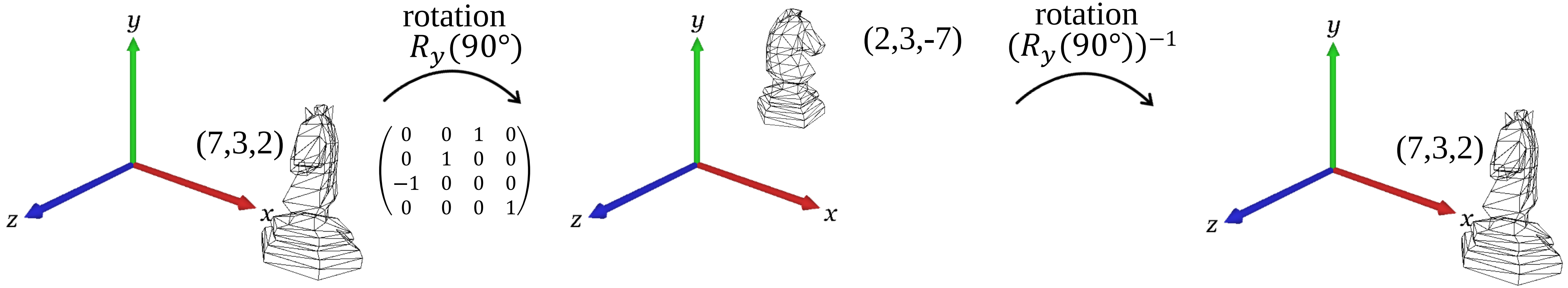
$$R = \begin{pmatrix} u_x & v_x & n_x \\ u_y & v_y & n_y \\ u_z & v_z & n_z \end{pmatrix}$$

$$R^{-1} = \begin{pmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ n_x & n_y & n_z \end{pmatrix}$$



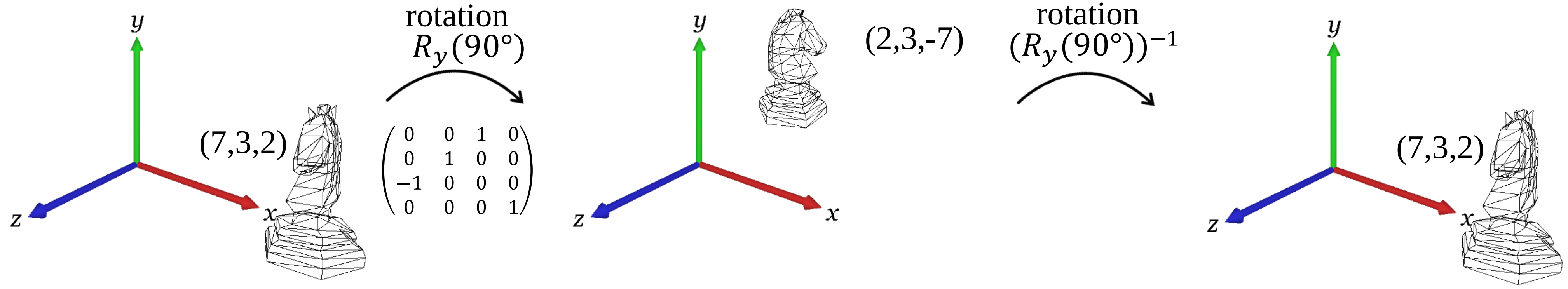
Practice

1. Write a 4×4 rotation matrix of $(R_y(90°))^{-1}$.



Practice - Solution

1. Write a 4×4 rotation matrix of $(R_y(90°))^{-1}$.



$$R_y(-90°) = \begin{pmatrix} \cos -90^\circ & 0 & \sin -90^\circ & 0 \\ 0 & 1 & 0 & 0 \\ -\sin -90^\circ & 0 & \cos -90^\circ & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{transpose}(R_y(90°)) = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{pmatrix}$$