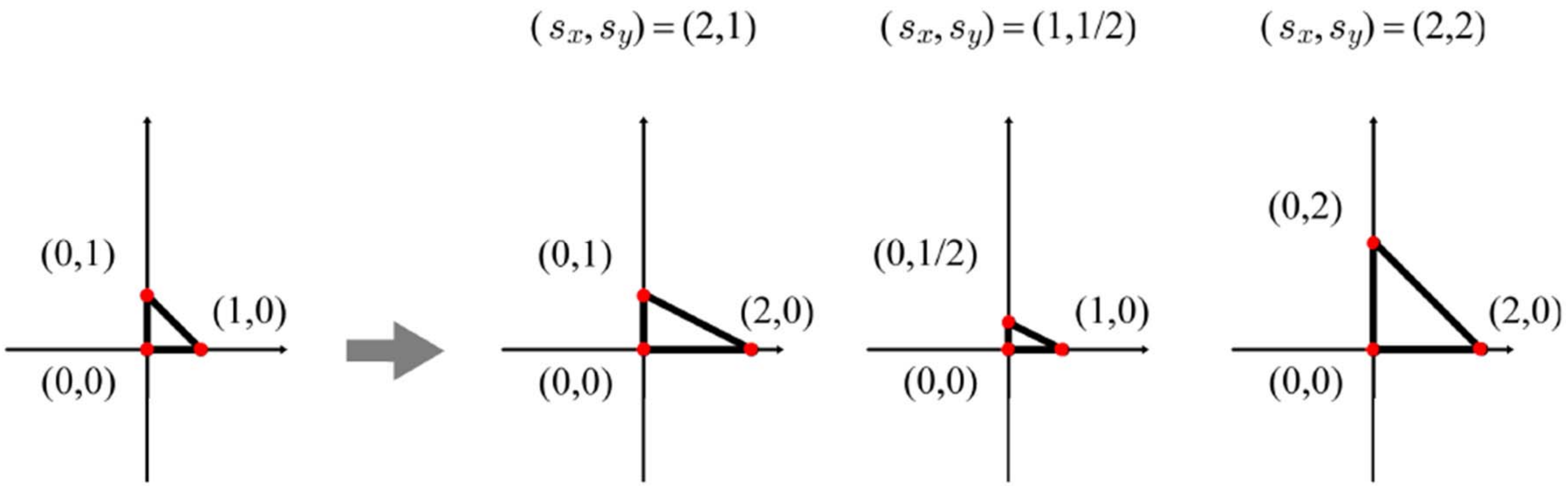
Chapter IV
Spaces and Transforms

Scaling

- 2D scaling with the scaling factors, s_x and s_y, which are independent.

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

- Examples

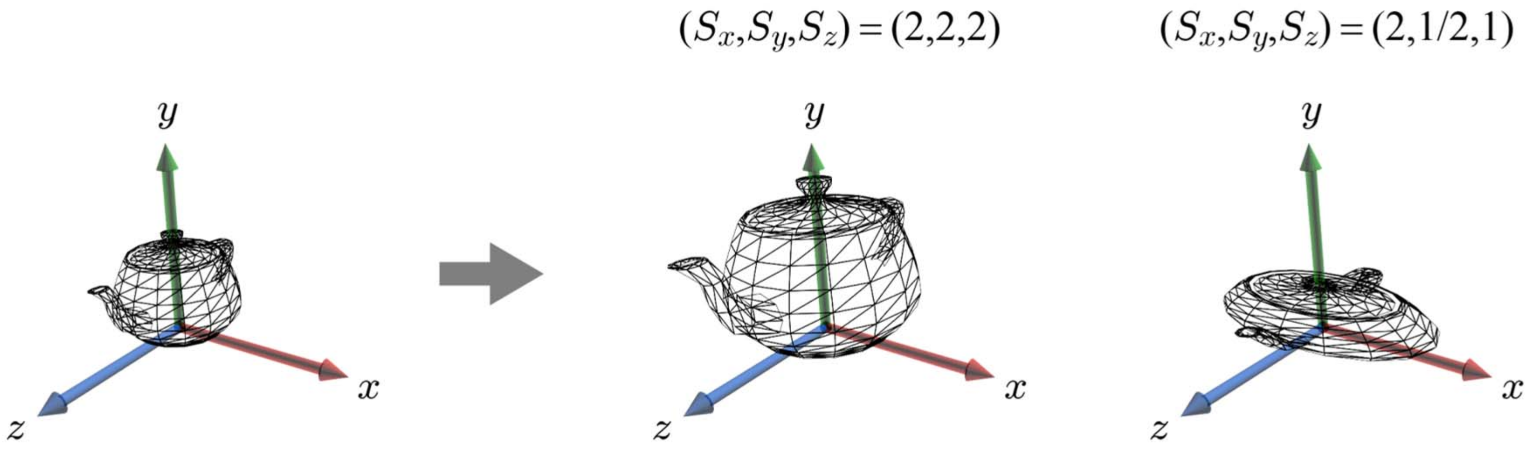


- When a polygon is scaled, all of its vertices are processed by the same scaling matrix.

Scaling (cont'd)

- 3D scaling

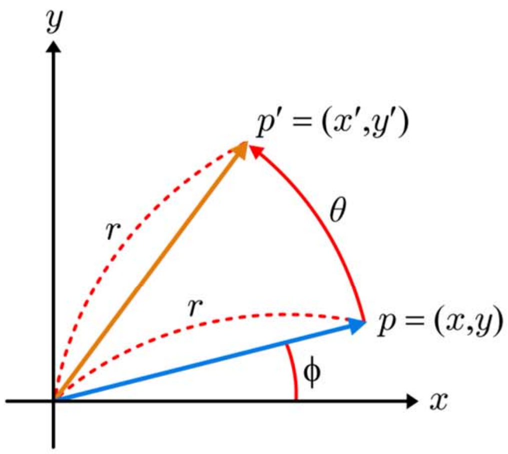
$$\begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} s_x x \\ s_y y \\ s_z z \end{pmatrix}$$



- In the same manner, the transform for a polygon mesh applies to all of its vertices.
- If all of the scaling factors are identical, the scaling is called *uniform*. Otherwise, it is a *non-uniform scaling*.

Rotation

- 2D rotation



$$x = rcos\phi$$
$$y = rsin\phi$$

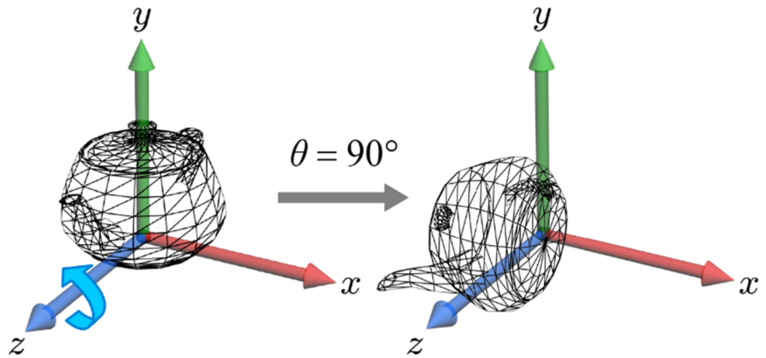
$$\begin{aligned} x' &= rcos(\phi + \theta) \\ &= rcos\phi cos\theta - rsin\phi sin\theta \\ &= xcos\theta - ysin\theta \end{aligned}$$

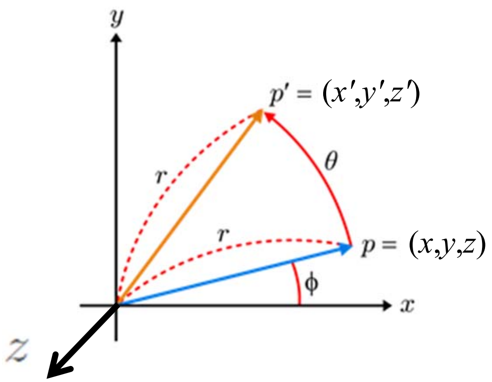
$$\begin{aligned} y' &= rsin(\phi + \theta) \\ &= rcos\phi sin\theta + rsin\phi cos\theta \\ &= xsin\theta + ycos\theta \end{aligned}$$

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

Rotation (cont'd)

- 2D rotation is defined "about the origin." In contrast, 3D rotation requires the *rotation axis*.
- Let's consider 3D rotations about x-axis (R_x), y-axis (R_y), and z-axis (R_z)
- First of all, R_z.





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

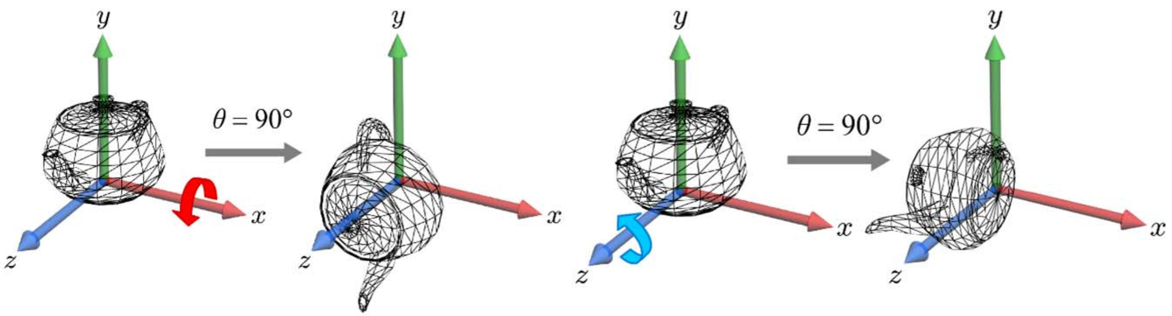
$$y' = x\sin\theta + y\cos\theta$$

$$z' = z$$

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

Rotation (cont'd)

- Observation for R_x.
  - Obviously, $x'=x$.
  - In R_x, the z-axis is turned counter-clockwise by 90° from the y-axis "when seen from the rotation axis."
  - In R_z, the y-axis is turned counter-clockwise by 90° from the x-axis "when seen from the rotation axis."
  - With respect to the rotation axis, the role of x-axis in R_z is taken by the y-axis in R_x. Similarly, the role of y-axis in R_z is taken by the z-axis in R_x.
  - Then, R_x is obtained by making such replacements in R_z.



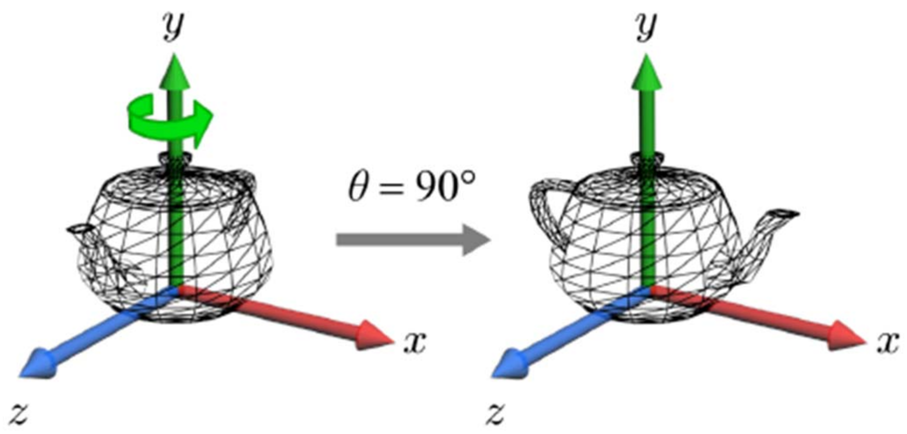
R_x R_z

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

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

Rotation (cont'd)

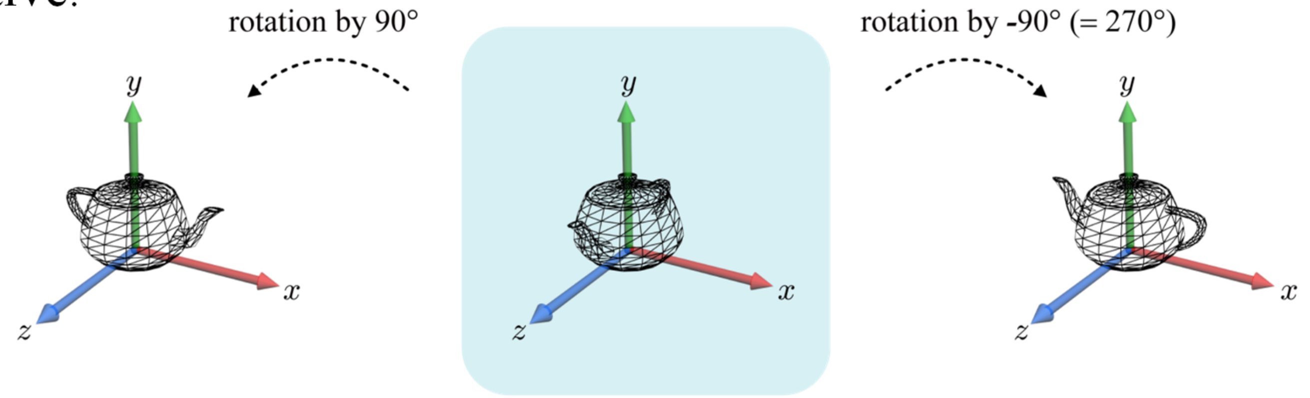
- In the same manner, we can define the matrix for R_y.



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

Rotation (cont'd)

- The *sign* of the rotation angle is determined as follows: Look at the origin of the coordinate system such that the axis of rotation points toward you. If the rotation is counter-clockwise, the angle is positive. If the rotation is clockwise, it is negative.

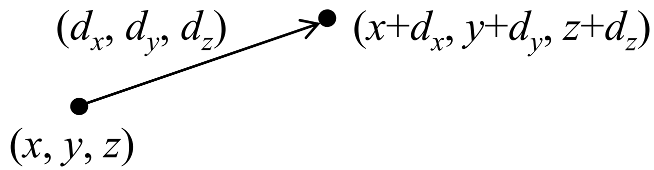


- Note that rotation by -θ is equivalent to rotation by (2π-θ).

Translation

- Translation is represented as *vector addition*.

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



- Affine transform
  - Linear transform – represented by *matrix multiplication*
    - Scaling
    - Rotation
    - etc.
  - Translation

Translation and Homogeneous Coordinates

- Fortunately, we can describe translation as matrix multiplication if we use the *homogeneous coordinates*.
- Given the 3D Cartesian coordinates (x, y, z) of a point, we can simply take (x, y, z, 1) as its homogeneous coordinates.
- We can then describe translation as *matrix multiplication*.

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

Homogeneous Coordinates

- For a point, the fourth component of the homogeneous coordinates is not necessarily 1 and is denoted by *w*.
- Cartesian coordinates → homogeneous coordinates
  - Cartesian coordinates (*x*, *y*, *z*) are converted into homogeneous coordinates (*wx*, *wy*, *wz*, *w*) with non-zero *w*.
  - For example, the Cartesian coordinates (1,2,3) can be converted into multiple homogeneous coordinates, (1,2,3,1), (2,4,6,2), (3,6,9,3), etc.
- Homogeneous coordinates → Cartesian coordinates
  - Given the homogeneous coordinates (*x*, *y*, *z*, *w*), the corresponding Cartesian coordinates are (*x*/*w*, *y*/*w*, *z*/*w*).

$$\begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x + d_x \\ y + d_y \\ z + d_z \\ 1 \end{pmatrix} \Leftrightarrow \begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2x \\ 2y \\ 2z \\ 2 \end{pmatrix} = \begin{pmatrix} 2x + 2d_x \\ 2y + 2d_y \\ 2z + 2d_z \\ 2 \end{pmatrix} = \begin{pmatrix} x + d_x \\ y + d_y \\ z + d_z \\ 1 \end{pmatrix}$$

Homogeneous Coordinates (cont'd)

- For handling the homogeneous coordinates, the 3x3 matrices for scaling and rotation need to be altered.

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

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

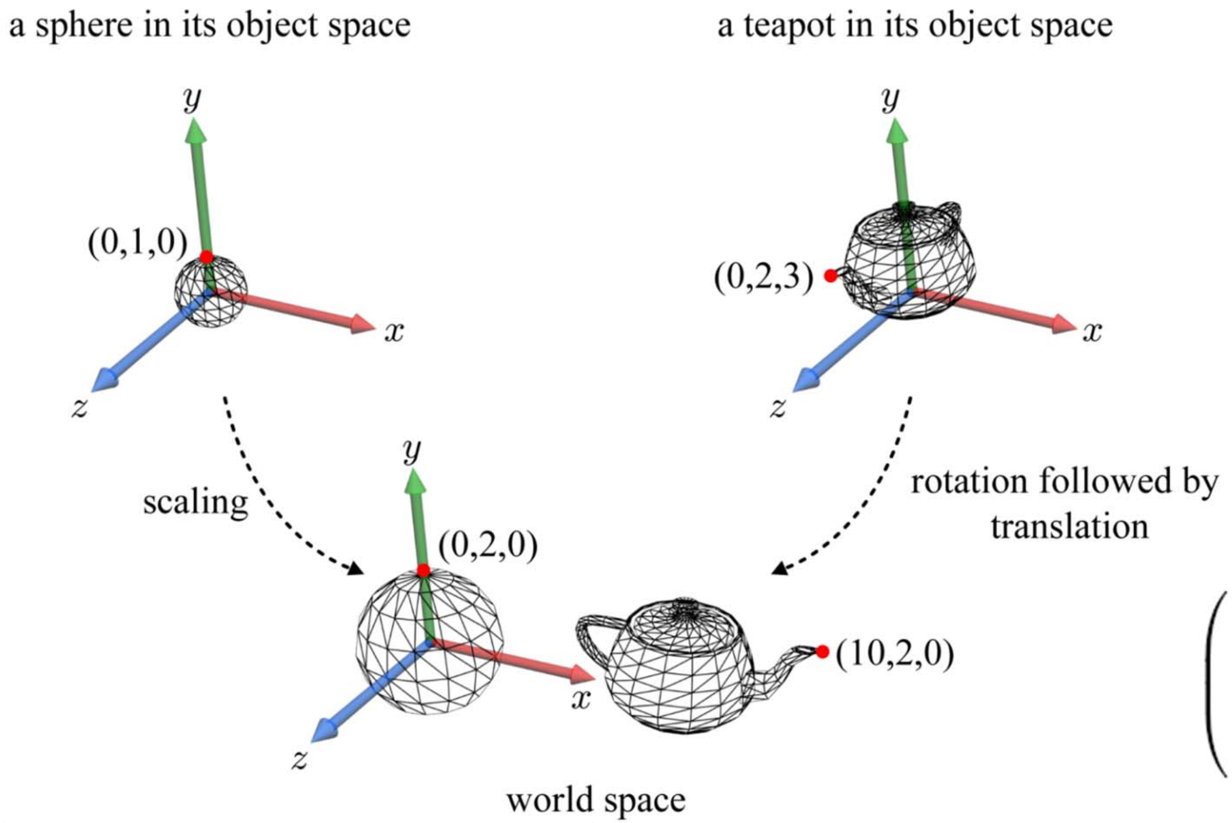
- Now that both linear transform and the translation are represented in 4x4 matrices, the linear transform and the translation can be combined into a single 4x4 matrix.

$$\begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 & d_x \\ 0 & s_y & 0 & d_y \\ 0 & 0 & s_z & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- No matter how many linear transforms and translations are given, they can be combined into a single 4x4 matrix.

Application: *World Transform*

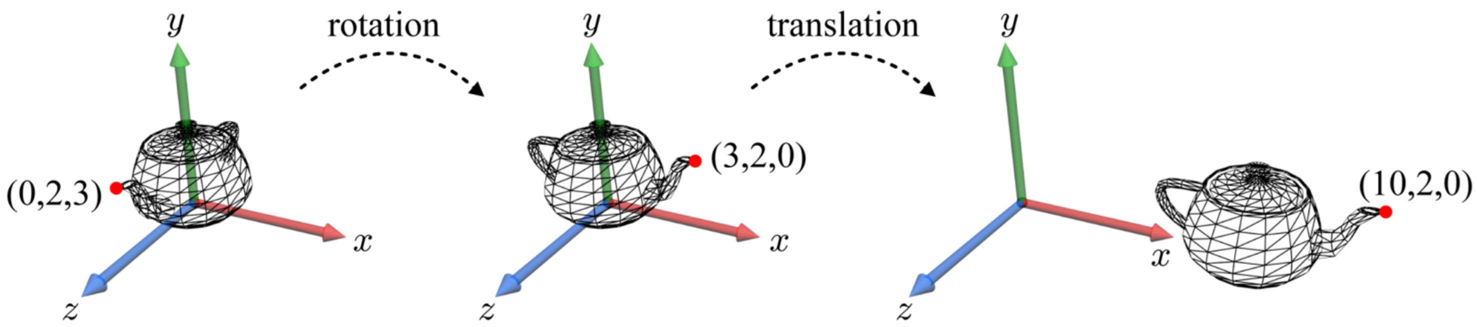
- The coordinate system used for creating an object is named *object space*.
- The object space for a model typically has no relationship to that of another model. The *world transform* ‘assembles’ all models into a single coordinate system called *world space*.



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

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

Application: World Transform (cont'd)



$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta_x & -sin\theta_x & 0 \\ 0 & sin\theta_x & cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} cos\theta_y & 0 & sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -sin\theta_y & 0 & cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} cos\theta_z & -sin\theta_z & 0 & 0 \\ sin\theta_z & cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

$$\begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad T = \begin{pmatrix} 1 & 0 & 0 & 7 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 & 0 & 7 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 10 \\ 2 \\ 0 \\ 1 \end{pmatrix}$$

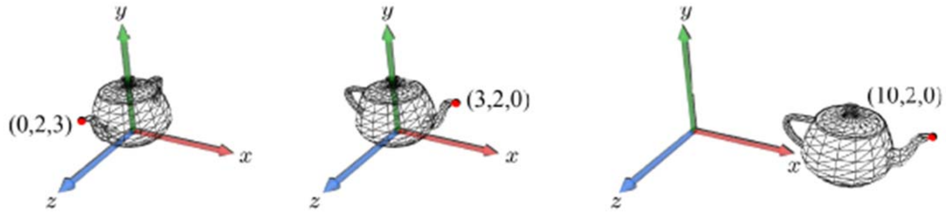
$$TR = \begin{pmatrix} 1 & 0 & 0 & 7 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 7 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 0 & 0 & 1 & 7 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 10 \\ 2 \\ 0 \\ 1 \end{pmatrix}$$

Affine Transform

- A rotation, R_y, followed by a translation, T

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

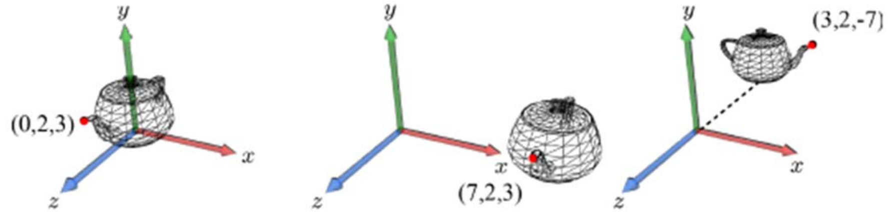
$$= \begin{pmatrix} cos\theta & 0 & sin\theta & d_x \\ 0 & 1 & 0 & d_y \\ -sin\theta & 0 & cos\theta & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



- Observe that, in the combined matrix, the upper-left 3x3 sub-matrix is filled with the input rotation, and the fourth column is with the input translation.
- Now reverse the order and observe that matrix multiplication is not commutative.

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

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



Affine Transform (cont'd)

- Suppose that a series of linear transforms and translations is concatenated to make a single 4x4 affine matrix.
  - Its fourth row is always (0 0 0 1)
  - The 3x4 elements are denoted by [***L***|***t***], i.e., by a 3x3 matrix ***L*** augmented with a 3D vector ***t***. ***L*** represents a 'combined' linear transform, which does not include any terms from the input translations, whereas ***t*** represents a 'combined' translation, which may contain the input linear-transform terms.

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

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

Affine Transform (cont'd)

- Let us take R_yT from the previous slide and combine with a scaling.

$$S(R_yT) = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} cos\theta & 0 & sin\theta & d_x cos\theta + d_z sin\theta \\ 0 & 1 & 0 & d_y \\ -sin\theta & 0 & cos\theta & -d_x sin\theta + d_z cos\theta \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

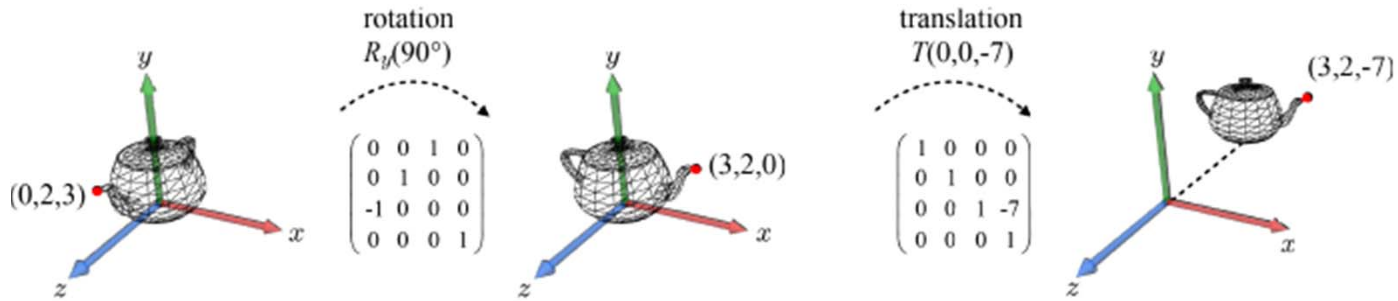
- Again, this is denoted by [***L***|***t***], where ***L*** represents a 'combined' linear transform and ***t*** represents a 'combined' translation.

Affine Transform (cont'd)

- Given a 3x4 matrix for an affine transform, [***L***|***t***], its application to an object is described as follows: ***L*** is applied first and then the linearly-transformed object is translated by ***t***.

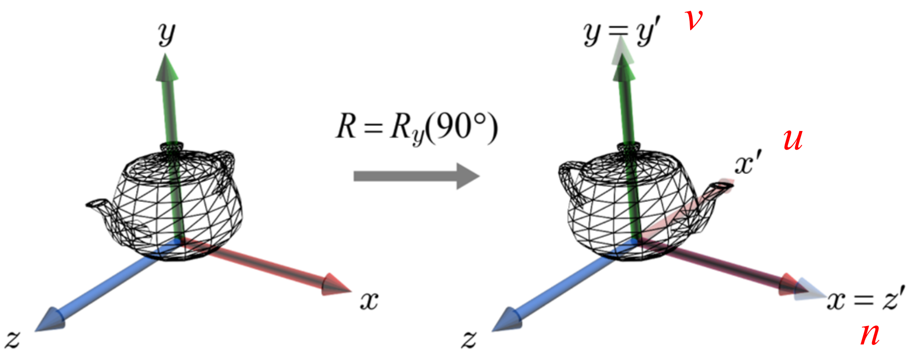
$$R_yT = \begin{pmatrix} cos\theta & 0 & sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -sin\theta & 0 & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} cos\theta & 0 & sin\theta & d_xcos\theta + d_zsin\theta \\ 0 & 1 & 0 & d_y \\ -sin\theta & 0 & cos\theta & -d_xsin\theta + d_zcos\theta \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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



Rotation and Object-space Basis

- An object can be thought of as being stuck to its object space, i.e., each vertex of the object is fixed and immovable within the object space.
- Initially the object space can be considered identical to the world space.
- A rotation applied to an object defines its orientation, and obviously the orientation is described by the axes of the 'rotated' object space.



- In the above example, x', y', and z' are the object-space axes and x, y, and z are the world-space axes.
- Let us denote the unit vectors along x', y', and z' by u, v, and n, respectively: $\{u, v, n\}$ is the *basis* of the object space, describing the object's orientation.

Rotation and Object-space Basis (cont'd)

- In general, the world space is associated with the standard basis, $\{e_1, e_2, e_3\}$.
- Initially the object space is identical to the world space, but it is *rotated* (by R) to have the orientation $\{u, v, n\}$. Specifically, e_1 is rotated into u, and it is described as follows:

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

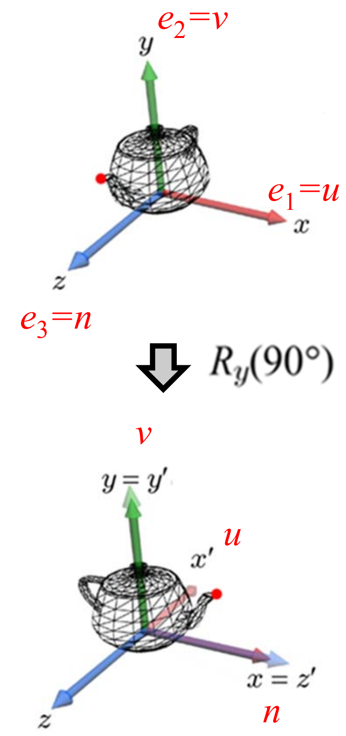
- Similarly, R transforms e_2 and e_3 into v and n, respectively:

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

- The above three are combined:

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

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

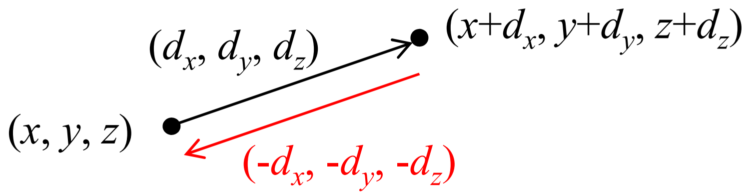


- R's columns are u, v, and n: Given a rotation matrix, the object-space basis with respect to the world space is immediately determined, and vice versa.

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

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

- 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

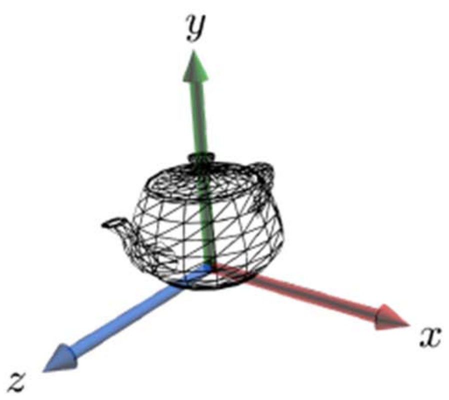
- Note that $\{u, v, n\}$ is an orthonormal basis, i.e., $u \cdot u = v \cdot v = n \cdot n = 1$ and $u \cdot v = v \cdot n = n \cdot u = 0$.
- Let's multiply R's transpose (R^T) with R:

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

- 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 (cont'd)

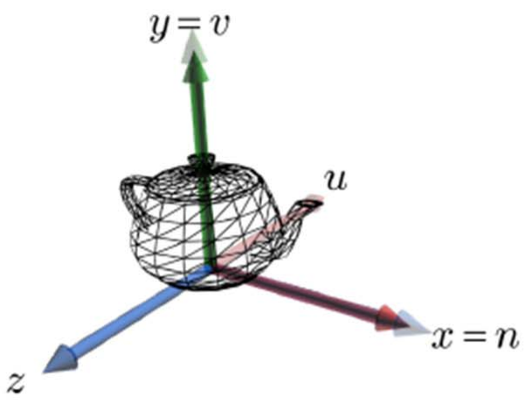
$$R_y^{-1}(90°) = R_y(-90°) = \begin{pmatrix} cos(-90°) & 0 & sin(-90°) \\ 0 & 1 & 0 \\ -sin(-90°) & 0 & cos(-90°) \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$



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

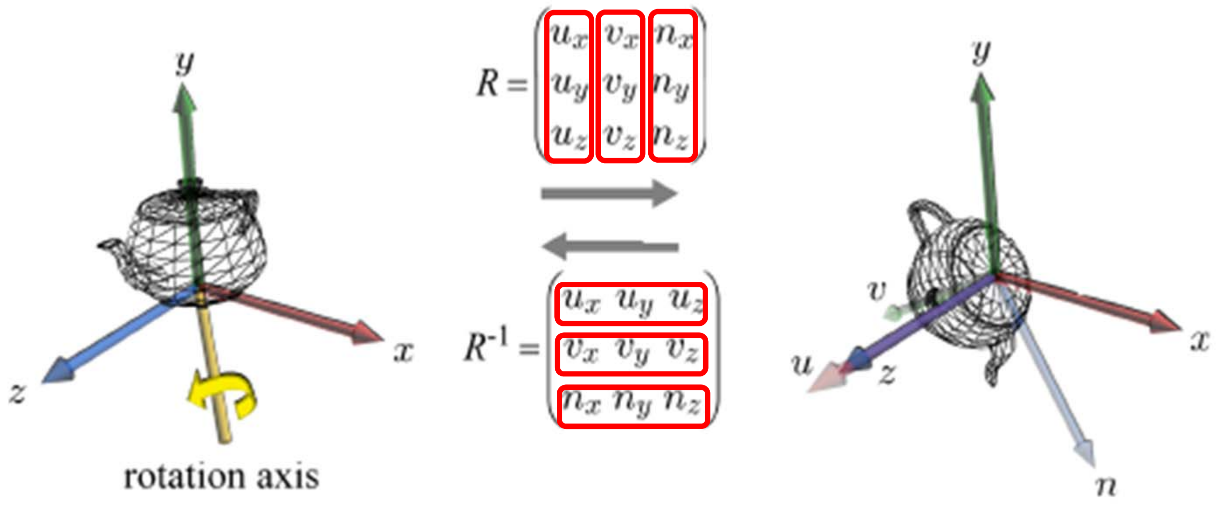
$$\rightleftarrows$$

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



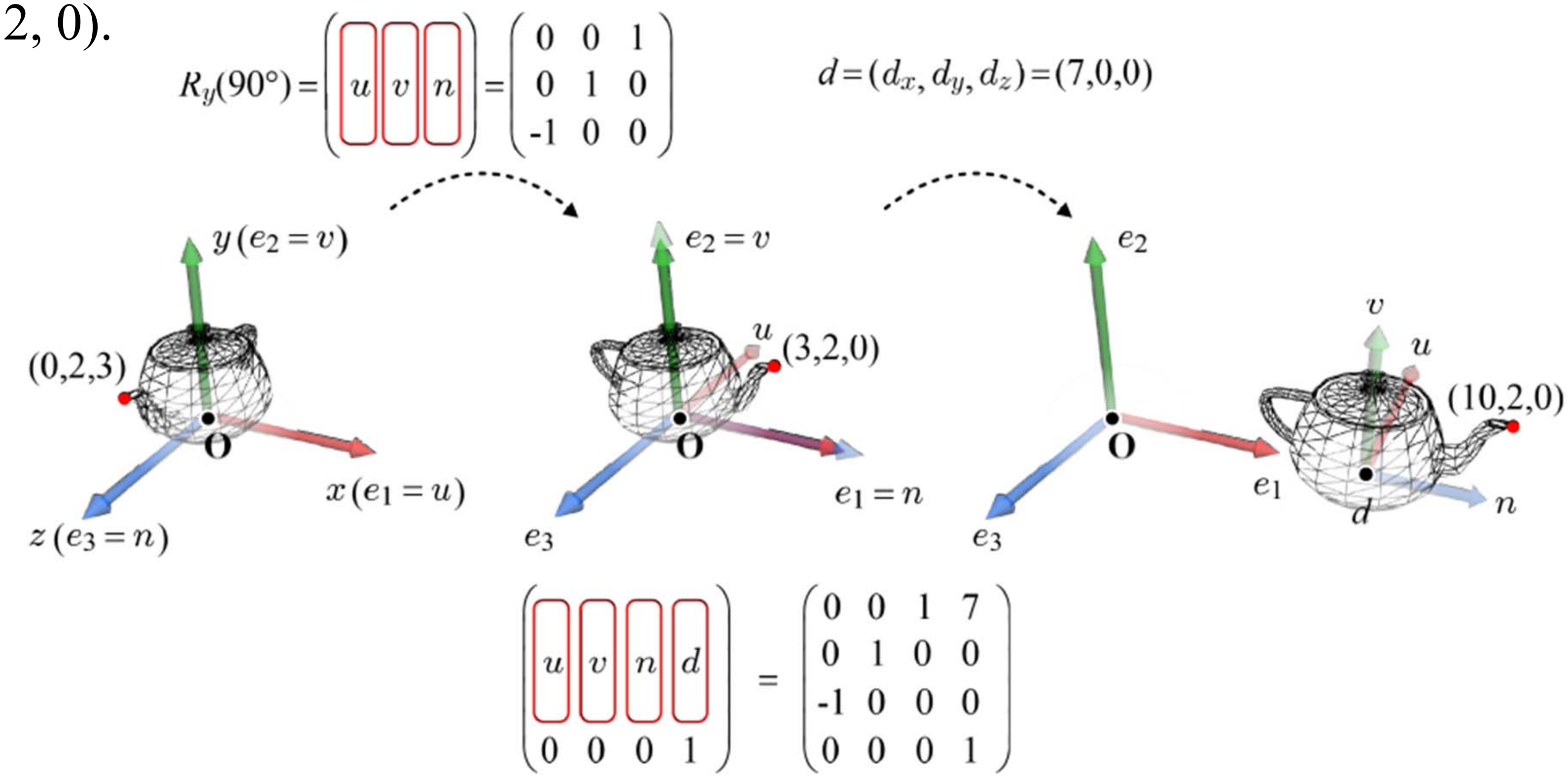
Rotation - Summary

- What has been presented so far applies in general.
- Consider a rotation “about an arbitrary axis.”
  - Suppose that its matrix R is obtained somehow. In fact, it is computable.
  - Then, the rotated object-space basis $\{u, v, n\}$ is immediately determined by taking the columns of R.
  - Inversely, if $\{u, v, n\}$ is known a priori, R is also immediately determined. Fig.
  - Of course, $R^{-1}=R^T$.



Basis Change and Space Change

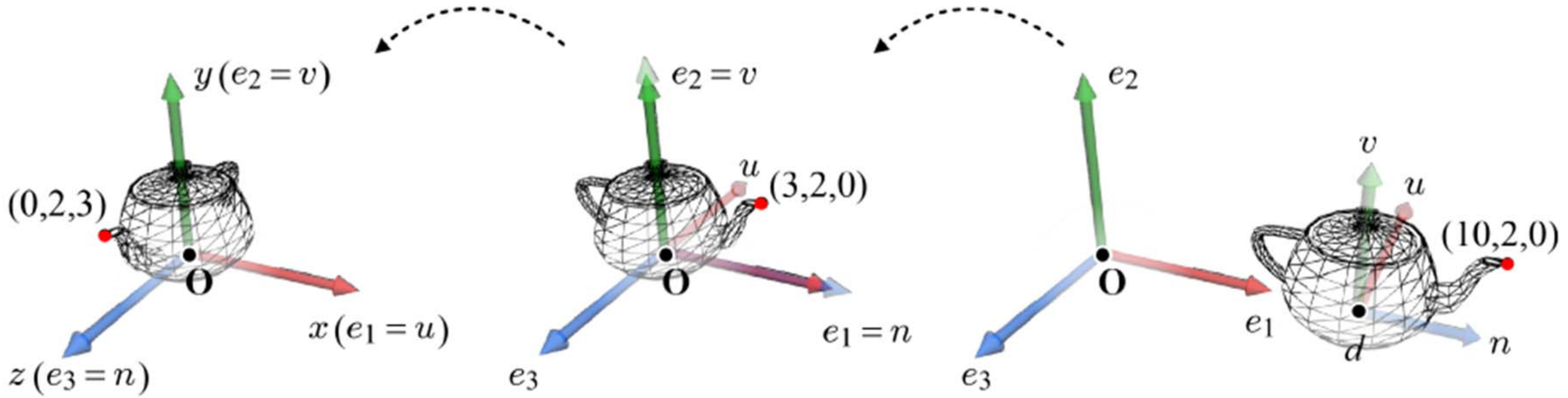
- In the example below, we have two distinct spaces: the world space, $\{e_1, e_2, e_3, \mathbf{O}\}$, and the object space, $\{u, v, n, d\}$. A point is given distinct coordinates in different spaces: the object-space coordinates of the teapot's mouth end remain fixed, (0, 2, 3), whereas its world-space coordinates are (10, 2, 0).
- $[R|d]$ transforms the object-space coordinates, (0, 2, 3), to the world-space ones, (10, 2, 0).



Basis Change and Space Change (cont'd)

- Consider the inverse transform. It is conceptually equivalent to transforming the object space such that it becomes identical to the world space.
- As the object space is now identical to the world space, we can take the transformed world-space coordinates, (0, 2, 3), as the object-space coordinates.

$$R_y^{-1}(90°) = \begin{pmatrix} \boxed{\quad u \quad} \\ \boxed{\quad v \quad} \\ \boxed{\quad n \quad} \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \qquad -d = (-7,0,0)$$



$$\begin{pmatrix} \boxed{\quad u \quad} & 0 \\ \boxed{\quad v \quad} & 0 \\ \boxed{\quad n \quad} & 0 \\ 0\ \ 0\ \ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & \\ 0 & 1 & 0 & -d \\ 0 & 0 & 1 & \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \boxed{\quad u \quad} & -d\cdot u \\ \boxed{\quad v \quad} & -d\cdot v \\ \boxed{\quad n \quad} & -d\cdot n \\ 0\ \ 0\ \ 0 & 1 \end{pmatrix}$$

Basis Change and Space Change (cont'd)

- The previous two pages showed *space-change* matrices. One is form the object space to the world space, and the other is from the world space to the object space.
- In contrast, the rotations change the bases, not the origin. They are *basis-change* matrices.