Computer Graphics Lecture 04: Transformation

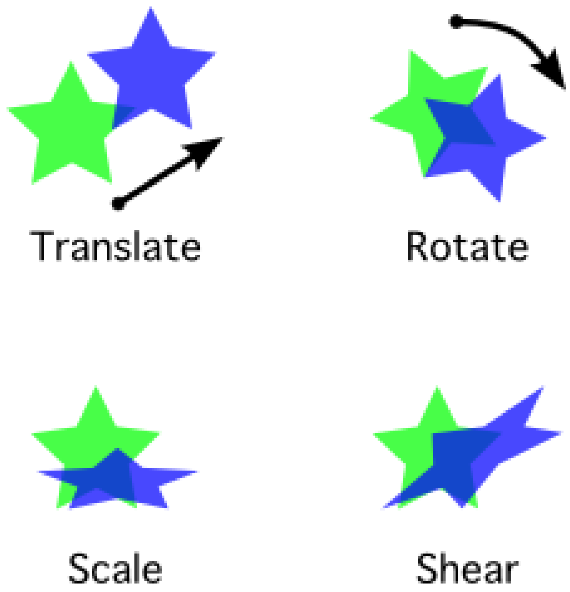
DR. ELVIS S. LIU
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY
SPRING 2018

Basic Transformation

- Transform objects are a mathematical mapping from one coordinate system to another
- The coordinate systems can differ in terms of
  - Position (i.e., location of the origin)
  - Scale
  - Axis direction
  - Relative axis orientation
- Transform objects can be applied to other transforms, vectors, or shape objects to map them between coordinate systems.

Types of Transformation

- The most basic ones are
  - Translation
  - Scaling
  - Rotation
  - Shear
  - And others, e.g., perspective transformation, projection, etc.
- Basic types of transformations
  - Rigid body: preserves length and angle
  - Affine: preserves parallel lines, not angles and lengths
  - Free-form: anything goes

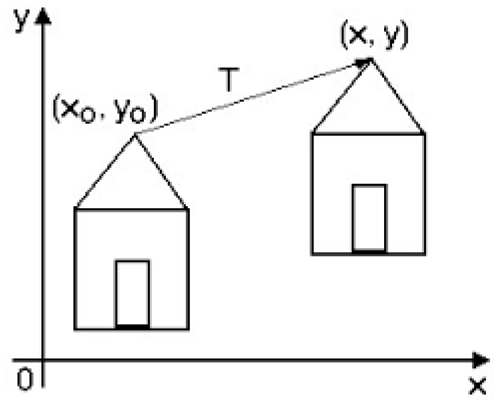


2D Transformation

LECTURE 04: TRANSFORMATION

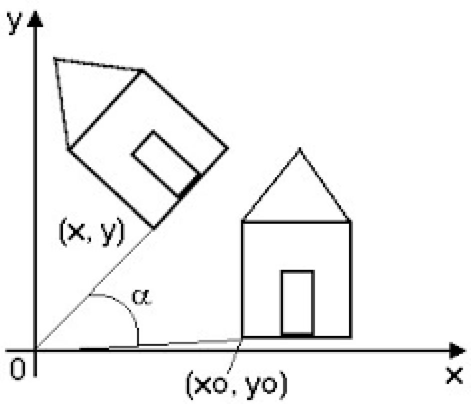
Translation in 2D

- Similar vector addition
- Vector form
  - $P = P_0 + T$
  - $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix}$
- Scalar form
  - $x = x_0 + dx$
  - $y = y_0 + dy$



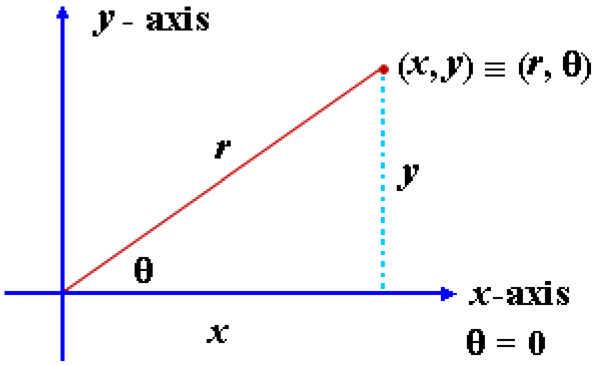
Rotation in 2D

- Positive angles: counterclockwise
- For negative angles
  - $\cos(-\alpha) = \cos(\alpha)$
  - $\sin(-\alpha) = -\sin(\alpha)$
- Vector form
  - $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$
- Scalar form
  - $x = x_0 \cos(\alpha) - y_0 \sin(\alpha)$
  - $y = x_0 \sin(\alpha) + y_0 \cos(\alpha)$



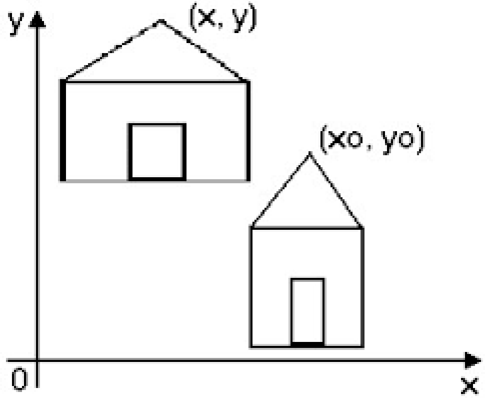
Derivation of Rotation Matrix

- Make use of polar coordinates:
  - $(r, \theta) \rightarrow (x,y)$
- $x_0 = r\cos(\theta)$, $y_0 = r\sin(\theta)$
- Rotating further an angle α about origin
  - $x = r\cos(\theta+\alpha)$, $y = r\sin(\theta+\alpha)$
- Hence, new position of x
  - $x = r\cos(\theta+\alpha)$
  - $x = r\cos(\theta)\cos(\alpha) - r\sin(\theta)\sin(\alpha)$
  - $x = x_0\cos(\alpha) - y_0\sin(\alpha)$
- New position of y
  - $y = r\sin(\theta+\alpha)$
  - $y = r\sin(\theta)\cos(\alpha) + r\cos(\theta)\sin(\alpha)$
  - $y = y_0\cos(\alpha) + x_0\sin(\alpha)$



Scaling in 2D

- $P = S \times P_0$
- Vector form
  - $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$
- Scalar form
  - $x = s_x \, x_0$
  - $y = s_y \, y_0$
- Uniform $s_x = s_y$
- Non uniform $s_x \neq s_y$



Shearing in 2D – X Shear

- Object shears in the x direction
- $P = SH_x \times P_0$
- Vector form
  - $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$
- Scalar form
  - $x = x_0 + a\, y_0$
  - $y = y_0$

Shearing in 2D – Y Shear

- Object shears in the y direction
- $P = SH_x \times P_0$
- Vector form
  - $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ b & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$
- Scalar form
  - $x = x_0$
  - $y = b\, x_0 + y_0$

Alternative Form of Translation

- What is the difference between translation and other types of transformation?

Alternative Form of Translation (cont.)

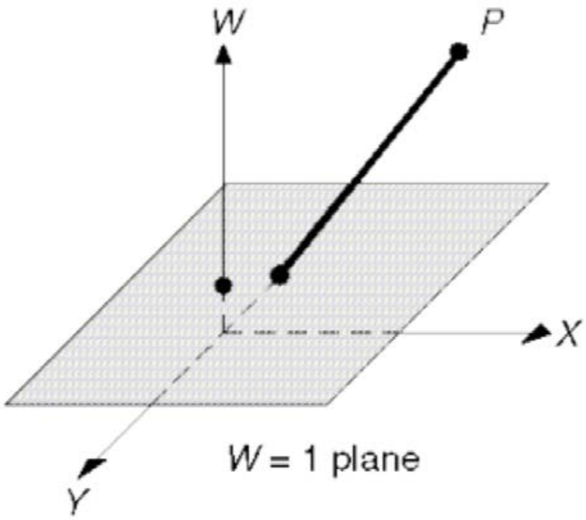
- Recall that we use vector addition for translation in 2D
  - $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix}$
- Using vector addition is not consistent with our method of treating transformations as matrices
- Translation cannot be expressed as matrix-vector multiplications
- If we could treat all transformations in a consistent manner, i.e., with matrix representation, then could combine transformations by composing their matrices

Homogeneous Coordinates

- How?
  - Add an additional dimension, the w-axis, and an extra coordinate, the w-component
- Homogeneous coordinates for 2D points
  - (x,y) turns into (x,y,w)
  - If (x,y,w) and (x',y',w') are multiples of one another, they represent the same point
  - Typically, $w \neq 0$
  - Points with $w = 0$ are points at infinity

2D Homogeneous Coordinates

- Cartesian coordinates of the homogenous point (x,y,w):
  - x/w, y/w   (divide through by w)
- Out typical homogenised points:
  - (x,y,1)
- Connection to 3D?
  - (x,y,1) represents a 3D point on the plane w = 1
  - A homogeneous point is a line in 3D, through the origin



Transformation in Homogeneous Coordinates

- Allow expression of all three 2D transformations as 3x3 matrices
- General form of the affine (linear) transformation
  - $$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & p \\ c & d & q \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
  - x' = ax + by + p
  - y' = cx + dy + q
- For example, 2D Translation in homogeneous coordinates
  - $$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Back to Translation

- Our translation matrix (***T***) can now be represented by embedding the translation vector in the right column:

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix}$$

- To verify that this is the right matrix, multiply it by a homogenised point ***v*** :

$$\boldsymbol{Tv} = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+dx \\ y+dy \\ 1 \end{bmatrix} = \boldsymbol{v}'$$

- Coordinates have been translated, and ***v'*** is still homogeneous

Transformations Homogenised

- Let's homogenize our all matrices! Doesn't affect linearity of scaling and rotation

- 2D Rotation Matrix
  - $$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- 2D Scaling Matrix
  - $$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Transformations Homogenised (cont.)

- 2D Shearing Matrix
  - $\begin{bmatrix} 1 & a & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- These transformations are called **affine** transformations, which means they preserve ratios of distances between points on a straight line (but not necessarily (0, 0))

Transformations Homogenised (Example)

- Scaling: Scale by 15 in the *x* direction, 17 in the *y*

$$\begin{bmatrix} 15 & 0 & 0 \\ 0 & 17 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Rotation: Rotate by 123°

$$\begin{bmatrix} \cos(123) & -\sin(123) & 0 \\ \sin(123) & \cos(123) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Translation: Translate by -16 in the *x* direction, +18 in the *y*

$$\begin{bmatrix} 1 & 0 & -16 \\ 0 & 1 & 18 \\ 0 & 0 & 1 \end{bmatrix}$$

Vectors vs. Homogeneous Coordinates

- There is a distinction between a point in homogeneous coordinates and vectors
- We use homogeneous coordinates to more conveniently represent translation; hence points are represented as:
  - $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$
- A vector can be rotated/scaled, but not translated (can think of it as always starting at origin), so don't use the homogeneous coordinate:
  - $\begin{bmatrix} x \\ y \\ 0 \end{bmatrix}$
  - That way, the translation matrix won't have any affect on vectors

Inverse of Transformation

- When we want to undo a transformation, we'll need to find the matrix's inverse.
- Thanks to homogenisation, they are all invertible
- Inverse of Translation T(dx,dy) = T(-dx,-dy)
  - $$\begin{bmatrix}1 & 0 & dx\\0 & 1 & dy\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & -dx\\0 & 1 & -dy\\0 & 0 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$$
  - Hence, $T(-dx,-dy) = T(dx,dy)^{-1}$

Inverse of Transformations (cont.)

- Inverse of Rotation
  - $R(\theta)^{-1} = R(-\theta)$
  - $= \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- Inverse of Scaling
- $S(s_x,s_y)^{-1} = S(1/s_x,1/s_y)$
- $= \begin{bmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Inverse of Transformations (cont.)

- Inverse of Shearing
  - $Sh(a,b)^{-1} = Sh(-a, -b)$
  - $\begin{bmatrix} 1 & -a & 0 \\ -b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Inverse and Transpose of Rotation Matrix

- Prove that the inverse of a rotation matrix M is just its transpose M^T
- Take a rotation matrix $M = [v_1 \quad v_2 \quad v_3]$ (where each v_i is a vector)
- First note some properties of M
  - The columns are orthogonal to each other: $v_i \bullet v_j = 0 \quad (i \neq j)$
  - Columns have unit length: $||v_i|| = 1$
- Let's see what multiplying M^T and M produces:

$$\begin{bmatrix} v_{1x} & v_{1y} & v_{1z} \\ v_{2x} & v_{2y} & v_{2z} \\ v_{3x} & v_{3y} & v_{3z} \end{bmatrix} \begin{bmatrix} v_{1x} & v_{2x} & v_{3x} \\ v_{1y} & v_{2y} & v_{3y} \\ v_{1z} & v_{2z} & v_{3z} \end{bmatrix} = \begin{bmatrix} v_1 \bullet v_1 & v_1 \bullet v_2 & v_1 \bullet v_3 \\ v_2 \bullet v_1 & v_2 \bullet v_2 & v_2 \bullet v_3 \\ v_3 \bullet v_1 & v_3 \bullet v_2 & v_3 \bullet v_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Using the properties above, we see that this is the identity matrix, so $M^T = M^{-1}$

Composition of Transformations in 2D

- An object in a scene uses many transformations in sequence. How do we represent this in terms of functions?
  - Concatenate basic transforms sequentially
  - This corresponds to multiplication of the transform matrices, thanks to homogeneous coordinates
- Transformation is a function; by associativity, we can compose functions:
  - *(f o g)(i)*
- This is the same as first applying g, then applying *f*:
  - *f(g(i))*
- Consider our functions *f* and *g* as matrices (M_1 and M_2) and our input as a vector *v*
- Our composition is equivalent to $M_1 M_2 v$

Composition of Translation

- What happens when a point goes through $T(dx_1, dy_1)$ and then $T(dx_2, dy_2)$?
- Combined translation $T(dx_1+dx_2,dy_1+dy_2)$
  - $\begin{bmatrix}1 & 0 & dx_2\\0 & 1 & dy_2\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & dx_1\\0 & 1 & dy_1\\0 & 0 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & dx_1 + dx_2\\0 & 1 & dy_1 + dy_2\\0 & 0 & 1\end{bmatrix}$
- Concatenation of transformations: matrix multiplication

Composition of Different Transformations

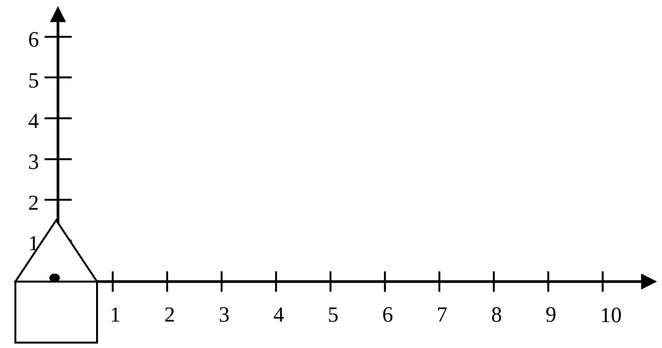
- We can form compositions of transformation matrices to form a more complex transformation
- For example, ***TRSv***, which scales a point, then rotates it, then translates it:

$$\begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

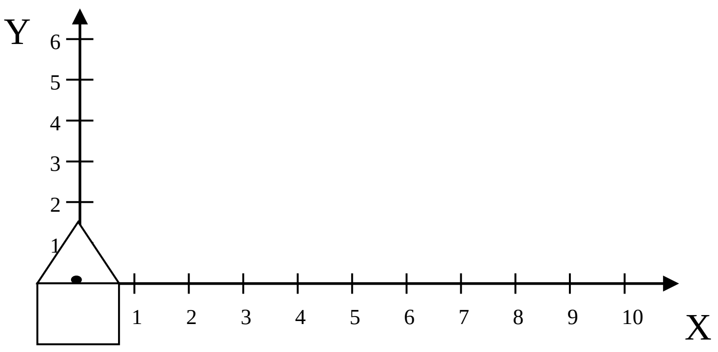
- Important: order matters!
  - Matrix multiplication is NOT commutative
- Note that we apply the matrices in sequence right to left. We can use associativity to compose them first; it is often useful to be able to apply a single matrix if, for example, we want to use it to transform many points at once.

Transformation is not commutative (example)

- Translate by *x = 6, y = 0*, then rotate by 45°



- Rotate by 45°, then translate by *x = 6, y = 0*



The Inverse of Composition of Transformations

- What is the inverse of a sequence of transformations?
  - $(M_1M_2 \ldots M_n)^{-1} = M_n^{-1}M_{n-1}^{-1} \ldots M_1^{-1}$
- Inverse of a sequence of transformations is the composition of the inverses of each transformation in reverse order (why?)
- Say we want to do the opposite transformation of ($T^{-1}RST$) What will our sequence look like?
  - $(T^{-1}RST)^{-1} = T^{-1}S^{-1}R^{-1}T$

3D Transformation

LECTURE 04: TRANSFORMATION

Dimension + 1

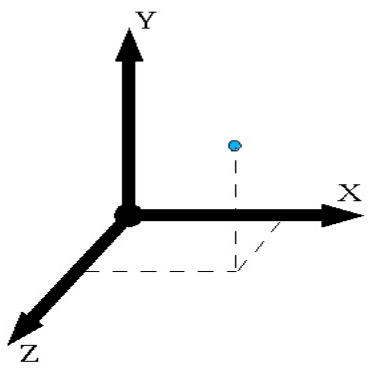
- How should we treat geometric transformations in 3D?
  - Just add one more coordinate axis
- A point is represented as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- A matrix fo a linear transformation T can be represented as

$$\begin{bmatrix} T(e_1) & T(e_2) & T(e_3) \end{bmatrix}$$

- Where e_3 is the standard basic vector along the z-axis



Homogeneous Coordinates

- Remember to use homogeneous coordinates. Embed scale and rotation matrices as upper left submatrices and translation vectors as upper right subvectors of the right column

Translation in 3D

- Similar to the 2D version, just with one more entry *dz*, representing change in the *z*-direction.
- $T\ (dx, dy, dz) = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Rotation in 3D

- In 2D, only one axis of rotation
- In 3D, there are infinitely many, must take all into account

Rodrigues's Formula

- Rotation by angle θ around vector $u = [u_x \;\; u_y \;\; u_z]^T$
  - Note: this is an arbitrary unit vector ***u*** in xyz-space
- Here's a not so friendly-looking rotation matrix

$$\begin{bmatrix} \cos\theta + u_x^2(1-\cos\theta) & u_x u_y(1-\cos\theta) - u_z\sin\theta & u_x u_z(1-\cos\theta) + u_y\sin\theta \\ u_x u_y(1-\cos\theta) + u_z\sin\theta & \cos\theta + u_y^2(1-\cos\theta) & u_y u_z(1-\cos\theta) - u_x\sin\theta \\ u_x u_z(1-\cos\theta) - u_y\sin\theta & u_y u_z(1-\cos\theta) + u_x\sin\theta & \cos\theta + u_z^2(1-\cos\theta) \end{bmatrix}$$

- This is called the coordinate form of Rodrigues's formula

Rotating Axis by Axis

- Every rotation can be represented as the composition of 3 different angles of counter-clockwise rotation around 3 axes, namely
  - x axis in the yz plane by ψ
  - y axis in the xz plane by ϑ
  - z axis in the xy plane by ϕ
- Also known as Euler angles, make problem of rotation much easier

Rotating Axis by Axis (cont.)

- $R_{yz}(\psi)$: rotation about x axis by ψ

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi & 0 \\ 0 & \sin\psi & \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- $R_{zx}(\theta)$: rotation about y axis by θ

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

- $R_{xy}(\varphi)$: rotation about z axis by φ

$$\begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Properties of Rotation Matrix

- Note these differ only in how the 3x3 submatrix is embedded in the homogeneous matrix, but the row-column order is different for R_{zx}
- You can compose these matrices to form a composite rotation matrix
- Columns and rows are mutually orthogonal unit vectors, i.e., orthonormal
- Determinant of M = 1
  - M = $\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- Product of any pair of orthonormal matrices is also orthonormal
- Orthonormality: inverse = transpose ($P^T = P^{-1}$)

Properties of Rotation Matrix (cont.)

- Row vectors: unit vectors which rotate into principal axes
  - i.e., $[1,0,0]^T$, $[0,1,0]^T$, and $[0,0,1]^T$
- Column vectors: unit vectors into which principle axes rotate

Scaling in 3D

- Looks just like the 2D version. We just added an s_z term
- $S\ (s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Shearing in 3D

- In (y,z) w.r.t. x value
  - $SH_{yz} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ sh_y & 1 & 0 & 0 \\ sh_z & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- In (z,x) w.r.t. y value
  - $SH_{zx} = \begin{bmatrix} 1 & sh_x & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & sh_z & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- In (x,y) w.r.t. z value
  - $SH_{xy} = \begin{bmatrix} 1 & 0 & sh_x & 0 \\ 0 & 1 & sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Inverse in 3D

- Inverses are once again parallel to their 2D versions
- Translation

$$\begin{bmatrix} 1 & 0 & 0 & -dx \\ 0 & 1 & 0 & -dy \\ 0 & 0 & 1 & -dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Scaling

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

Inverse in 3D (cont.)

- Rotation $\boldsymbol{R_{yz}^{-1}(\psi)}$ $\boldsymbol{R_{zx}^{-1}(\vartheta)}$ $\boldsymbol{R_{xy}^{-1}(\phi)}$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\psi & \sin\psi & 0 \\ 0 & -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} \cos\varphi & \sin\varphi & 0 & 0 \\ -\sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Composition of Transformations in 3D

- Composition works exactly the same way
- Let’s take a 3D object, say
  - A cube centered at (2,2,2)
  - Rotate clockwise in object’s space by 30º around x axis, 60º around y, and 90º around z
  - Scale in object space by 1 in the x, 2 in the y, and 3 in the z
  - Translate by (2,2,4) in world space
- Transformation sequence: $TT_0^{-1}S_{xy}R_{xy}R_{zx}R_{yz}T_o$, where T_0 translates to (0,0)

Composition of Transformations in 3D (cont.)

- Transformation sequence: $\boldsymbol{T T_0^{-1} S_{xy} R_{xy} R_{zx} R_{yz} T_o}$

$$\begin{bmatrix}1&0&0&2\\0&1&0&2\\0&0&1&4\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&2\\0&1&0&2\\0&0&1&2\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&2&0&0\\0&0&3&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}\cos 90&\sin 90&0&0\\-\sin 90&\cos 90&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}\cos 60&0&-\sin 60&0\\0&1&0&0\\\sin 60&0&\cos 60&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&\cos 30&\sin 30&0\\0&-\sin 30&\cos 30&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&-2\\0&1&0&-2\\0&0&1&-2\\0&0&0&1\end{bmatrix}$$

$\boldsymbol{T}$ $\quad$ $\boldsymbol{T_0^{-1}}$ $\quad$ $\boldsymbol{S_{xy}}$ $\quad$ $\boldsymbol{R_{xy}}$ $\quad$ $\boldsymbol{R_{zx}}$ $\quad$ $\boldsymbol{R_{yz}}$ $\quad$ $\boldsymbol{T_0}$