- Transformations are functions applied to points in space. As function of time, they create animation (program-1)

$$p' = f(p)$$

They provide a mechanism to manipulate models (data).

- The term *homogeneous coordinates* is used in mathematics to refer to the effect of this representation on Cartesian equations. When a cartesian point (x, y) is converted to a homogeneous representation (x_h, y_h, h), equations containing x and y, such as $f(x, y) = 0$, become homogeneous equations in the three parameters x_h, y_h and h (where, $x = x_h/h$, $y = y_h/h$). This just means that if each of the three parameters is replaced by any value h times that parameter, the value h can be factored out of the equations. Similar to above, the homogeneous representation for a point in 3D (x, y, z) is given by (x_h, y_h, z_h, h), where $(x = x_h/h,\ y = y_h/h\ \&\ z = z_h/h)$. Expressing positions in homogeneous coordinates allows us to represent all geometric transformation equations as matrix multiplications.

  There can be infinite ways of representing the homogeneous coordinates for a given point.

- Give two different homogeneous coordinates for a point in 2D given by $[5 \quad 6]$, and a point in 3D given by $[7 \quad 3 \quad 4]$.

  (5, 6, 1) and (10, 12, 2)

  (7, 3, 4, 1) and (14, 6, 8, 2)

- Rotation Matrix properties

$$R^{-1}(\theta) = R(-\theta) = R^T$$

Orthogonal, orthonormal

Commutative

- Reflection in x followed by a reflection in y is the same as rotation by 180 degrees (show?)
- Prove that the multiplication of transformation matrices for each of the following sequence of operations is commutative: i) Two successive rotations, ii) Two successive translations, iii) Two successive scalings.
- Prove that a uniform scaling and a rotation form a commutative pair of operations but that, in general, scaling and rotation are not commutative.
- Show that transformation matrix for a reflection about the line y=x, is equivalent to a reflection relative to the x axis followed by a counter-clockwise rotation of 90 degrees.
- Show that two successive reflections about either of the coordinate axes is equivalent to a single rotation about the coordinate origin.

Affine

- Rotation and Scaling are of the form

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

- A transformation of the form

$$x' = ax + by$$

$$y' = cx + dy$$

is called a linear transformation, represented by the matrix:

- Translation is affine, but not linear (translation is not represented in above matrix form).
- In general, <u>Projective</u> *is superset of* <u>affine</u> *is superset of* <u>linear</u>
- Linear: Transforms straight lines to straight lines or a point. Vector (0,0) is always transformed to (0,0).
- Affine: Preserves parallel lines. Vector (0,0) is <u>not</u> always transformed to (0,0).
- Projective: Parallel lines not necessarily preserved, but lines are sent to lines or points (not curves).
- Translation and rotation are rigid body motions that preserve lengths and angles. i.e, after the transformation, lengths between any two points or angles between any two lines remain unchanged.
- In the below rotation matrix,

$$\begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$$

The rows/columns are i) unit vectors, ii) orthogonal. i.e, they are orthonormal.

- A rotation axis can be defined with two coordinate positions, or with one coordinate point and direction angles (direction cosines) between the rotation axis and two of the coordinate axes.
- Given endpoints, the rotation-axis vector is given by,

$$V = P_2 - P_1 = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$$

  Unit rotation-axis vector u is,

$$u = \frac{V}{|V|}$$

  where the components a, b, c are the direction cosines for the rotation axis:

$$a = \frac{x_2 - x_1}{|V|}, b = \frac{y_2 - y_1}{|V|}, c = \frac{z_2 - z_1}{|V|}$$

- First operation is translation:

$$\begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

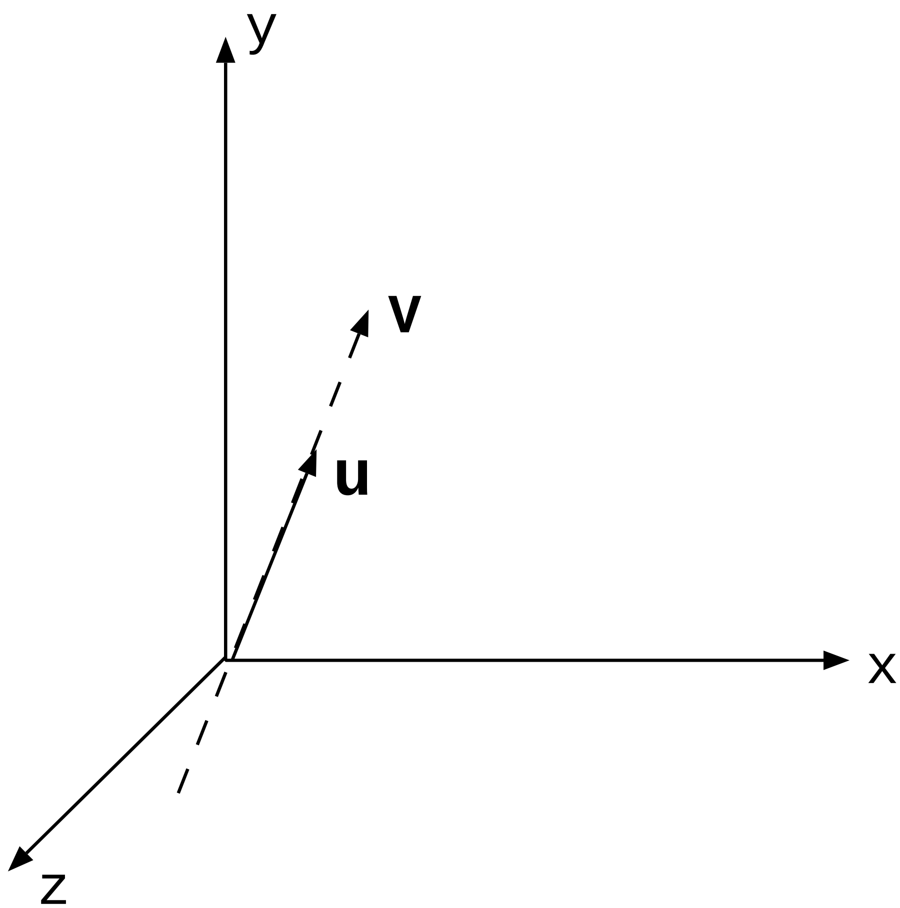
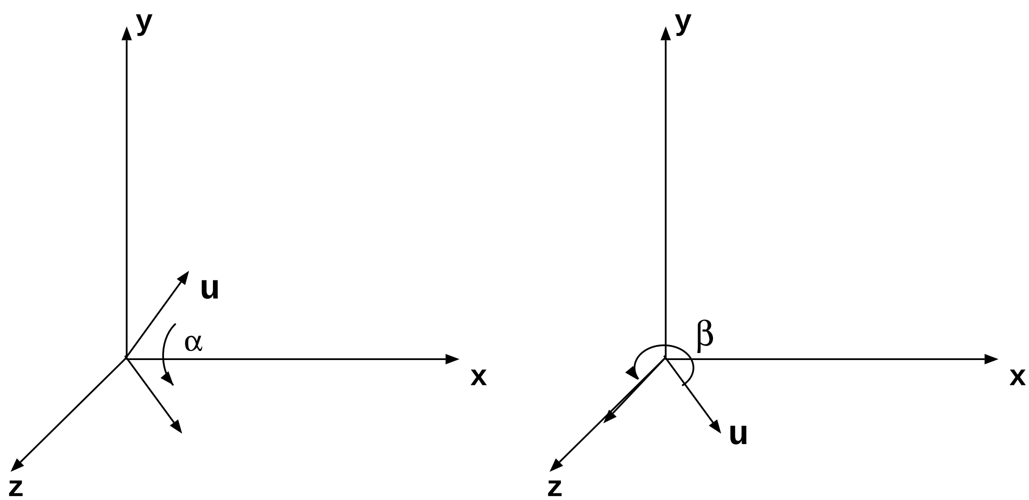


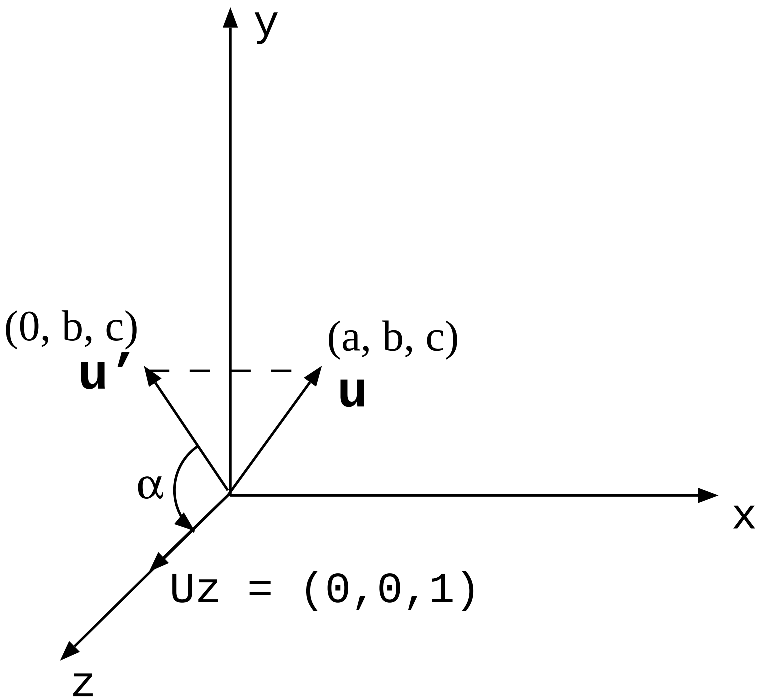
Figure 1:

- We now use coordinate-axis rotations to accomplish alignment of V to z-axis. We first rotate about the x-axis, then rotate about y-axis. The x-axis rotation gets vector u into the xz plane, and the y-axis rotation swings u arouind to the z-axis.
- Rotation angle is the angle between the projection of u in yz plane and the positive z-axis (see Figure 2, 3).



(a) Unit Vector u is rotated about the x-axis to bring it into the xz plane (b) then it is rotated around the y axis to align it with the z axis.

Figure 2:



Rotation of u around the x axis into the xz plane is accomplished by rotating u’ (projection of u in the yz plane) through angle α onto the z axis.

Figure 3:

- Projection of u in the yz plane, u’ = (0, b, c). Then, cosine of the rotation angle α can be determined as dot product of u’ and unit vector on z-axis (0, 0, 1):

$$cos\alpha = \frac{u' \cdot u_z}{|u'||u_z|} = \frac{c}{d}$$

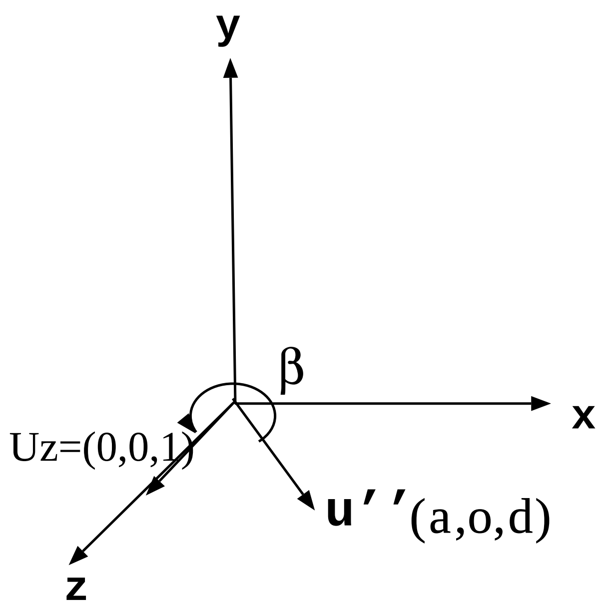
$$d = \sqrt{b^2 + c^2}$$

Similarly, we have

$$sin\alpha = \frac{b}{d}$$

The rotation matrix is then,

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & -b/d & 0 \\ 0 & b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Rotation of unit vector u'' (vector u after rotation into the xz plane) about the y axis. Positive rotation angle β aligns u'' with vector Uz.

Figure 4:

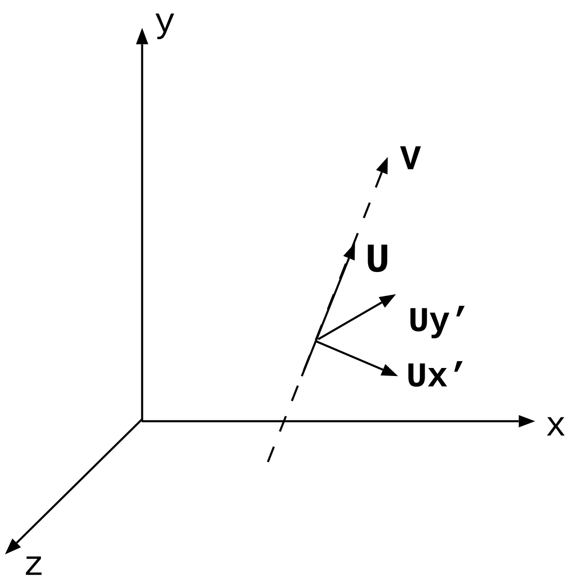
- Next step is to determine the matrix that will swing the unit vector in the xz plane CLOCKWISE (mistake in Hearn and Baker) around the y-axis onto the positive z axis (see Fig. 4).

  The cosβ is given by d, and sinβ is given by a (obtained by dot of x-axis and u", i.e, (1 0 0) $\cdot$ (a 0 d) is cos(90-β) = sin β). Therefore the transformation matrix for rotation of u" about the y-axis is,

$$R_y(\beta) = \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  Thus, the transformation sequence is given by:

$$R(\theta) = T^{-1} \cdot R_x^{-1}(\alpha) \cdot R_y^{-1}(\beta) \cdot R_z(\theta) \cdot R_y(\beta) \cdot R_x(\alpha) \cdot T$$



Local coordinate system for a rotation axis defined by unit vector U.

Figure 5:

- Quicker, less intuitive method is to make use of the fact that the composite matrix for any sequence of 3D rotations is of the form:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  Upper-left 3 by 3 submatrix is orthogonal. i.e, rows/columns of this submatrix form a set of orthogonal unit vectors that are rotated by R onto the x,y,z axes, respectively.

$$R \cdot \begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

  So, we can set up a local coordinate system with one of its axes aligned on the rotation axis.

Then the unit vectors for the three coordinate axes are used to construct the columns of the rotation matrix.

$$u_z' = u$$

$$u_y' = \frac{u X u_x}{|u X u_x|}$$

$u_x' = u_y' X u_z'$

If we express the elements of the unit local vectors for the rotation axis as

$$u_x' = (u_{x1}', u_{x2}', u_{x3}')$$
$$u_y' = (u_{y1}', u_{y2}', u_{y3}')$$
$$u_z' = (u_{z1}', u_{z2}', u_{z3}')$$

Then the required composite matrix, equal to the product $R_y(\beta) \cdot R_x(\alpha)$ is,

$$R = \begin{bmatrix} u_{x1}' & u_{x2}' & u_{x3}' & 0 \\ u_{y1}' & u_{y2}' & u_{y3}' & 0 \\ u_{z1}' & u_{z2}' & u_{z3}' & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Quaternion Methods for 3D Rotations

- Quaternoin: a basic representation

$$q = s + ia + jb + kc$$

- Properties

$$i^2 = j^2 = k^2 = -1 \quad ij = -ji = k$$

- Further

$$jk = -kj = i, \quad ki = -ik = j$$

- Addition of two quaternions

$$q_1 + q_2 = (s_1 + s_2) + i(a_1 + a_2) + j(b_1 + b_2) + k(c_1 + c_2)$$

- Quaternoio: ordered-pair notation

$$q = (s, \mathbf{v}), \text{ where } \mathbf{v} = (a, b, c)$$

- Addition in ordered-pair notation

$$q_1 + q_2 = (s_1 + s_2, \mathbf{v}_1 + \mathbf{v}_2)$$

- Multiplication of two quaternions

$$q_1 q_2 = (s_1 s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2)$$

- Magnitude sqaured of a quaternion

$$|q|^2 = s^2 + \mathbf{v} \cdot \mathbf{v}$$

- Inverse of a quaternion

$$q^{-1} = \frac{1}{|q|^2}(s, -\mathbf{v})$$

$$qq^{-1} = q^{-1}q = (1, 0)$$

Quaternion methods for three-dimensional rotations

- Quaternion: ordered-pair notation

$$q = (s, \mathbf{v}), \text{ where } \mathbf{v} = (a, b, c)$$

$$s = \cos\frac{\theta}{2}, \quad \mathbf{v} = \mathbf{u}\sin\frac{\theta}{2},$$

where $\mathbf{u}$ is a unit vector along the selected rotation axis and θ is the specific rotation angle about this axis.

- Point $\mathbf{P}$

$$\mathbf{P} = (0, \mathbf{p})$$

- After rotation

$$\mathbf{P}' = q\mathbf{P}q^{-1},$$

where $q^{-1} = (s, -\mathbf{v})$ is the inverse of the unit quaternion q.

- Point $\mathbf{P}'$

$$\mathbf{P}' = (0, \mathbf{p}')$$

where

$$\mathbf{p}' = s^2\mathbf{p} + \mathbf{v}(\mathbf{p}\cdot\mathbf{v}) + 2s(\mathbf{v}\times\mathbf{p}) + \mathbf{v}\times(\mathbf{v}\times\mathbf{p})$$

- Basic trigonometric identities

$$\cos^2\frac{\theta}{2} - \sin^2\frac{\theta}{2} = 1 - 2\sin^2\frac{\theta}{2} = \cos\theta,$$

$$2\cos\frac{\theta}{2}\sin\frac{\theta}{2} = \sin\theta$$

- Matrix M_R.

$$\mathbf{M}_R(\theta) = \begin{bmatrix} 1-2b^2-2c^2 & 2ab-2sc & 2ac+2sb \\ 2ab+2sc & 1-2a^2-2c^2 & 2bc-2sa \\ 2ac-2sb & 2bc+2sa & 1-2a^2-2b^2 \end{bmatrix}$$

- Rewrite matrix M_R

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

- Rotation expression

$$\mathbf{R}(\theta) = \mathbf{T}^{-1} \cdot \mathbf{M}_R \cdot \mathbf{T}.$$