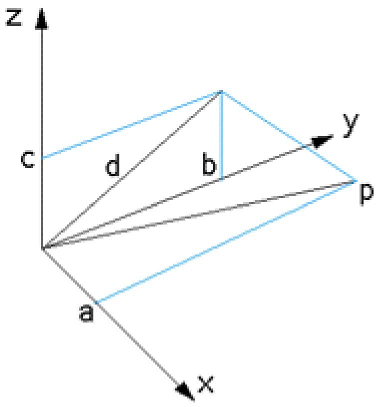
Rotate A Point About An Arbitrary Axis (3 Dimensions)

Written by Paul Bourke

December 1992, Updated August 2002



Illustrative C code that implements the following algorithm is given here. A closed solution attributed to Ronald Goldman is presented as this C function. A contribution by Bruce Vaughan in the form of a Python script for the SDS/2 design software: PointRotate.py.

Rotation of a point in 3 dimensional space by theta about an arbitrary axes defined by a line between two points $P_1 = (x_1,y_1,z_1)$ and $P_2 = (x_2,y_2,z_2)$ can be achieved by the following steps

(**1**) translate space so that the rotation axis passes through the origin

(**2**) rotate space about the x axis so that the rotation axis lies in the xz plane

(**3**) rotate space about the y axis so that the rotation axis lies along the z axis

(**4**) perform the desired rotation by theta about the z axis

(**5**) apply the inverse of step (**3**)

(**6**) apply the inverse of step (**2**)

(**7**) apply the inverse of step (**1**)

Note:

- If the rotation axis is already aligned with the z axis then steps **2**, **3**, **5**, and **6** need not be performed.
- In all that follows a right hand coordinate system is assumed and rotations are positive when looking down the rotation axis towards the origin.
- Symbols representing matrices will be shown in bold text.
- The inverse of the rotation matrices below are particularly straightforward since the determinant is unity in each case.
- All rotation angles are considered positive if anticlockwise looking down the rotation axis towards the origin.

Step 1

Translate space so that the rotation axis passes through the origin. This is accomplished by translating space by $-\mathbf{P}_1$ $(-x_1,-y_1,-z_1)$. The translation matrix $\mathbf{T}$ and the inverse $\mathbf{T}^{-1}$ (required for step 7) are given below

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{T}^{-1} = \begin{pmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & y_1 \\ 0 & 0 & 1 & z_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Step 2

Rotate space about the x axis so that the rotation axis lies in the xz plane. Let U = (a,b,c) be the unit vector along the rotation axis. and define $d = \mathrm{sqrt}(b^2 + c^2)$ as the length of the projection onto the yz plane. If d = 0 then the rotation axis is along the x axis and no additional rotation is necessary. Otherwise rotate the rotation axis so that is lies in the xz plane. The rotation angle to achieve this is the angle between the projection of rotation axis in the yz plane and the z axis. This can be calculated from the dot product of the z component of the unit vector U and its yz projection. The sine of the angle is determine by considering the cross product.

$$\cos(t) = \frac{(0,0,c) \text{ dot } (0,b,c)}{c\ d} = c/d \qquad \sin(t) = \frac{\| (0,0,c) \text{ cross } (0,b,c) \|}{c\ d} = b/d$$

The rotation matrix $\mathbf{R}_x$ and the inverse $\mathbf{R}_x^{-1}$ (required for step 6) are given below

$$\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & -b/d & 0 \\ 0 & b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{R}_x^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & b/d & 0 \\ 0 & -b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Step 3

Rotate space about the y axis so that the rotation axis lies along the positive z axis. Using the appropriate dot and cross product relationships as before the cosine of the angle is d, the sine of the angle is a. The rotation matrix about the y axis $\mathbf{R}_y$ and the inverse $\mathbf{R}_y^{-1}$ (required for step 5) are given below.

$$\mathbf{R}_y = \begin{pmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{R}_y^{-1} = \begin{pmatrix} d & 0 & a & 0 \\ 0 & 1 & 0 & 0 \\ -a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Step 4

Rotation about the z axis by t (theta) is $\mathbf{R_z}$ and is simply

$$\mathbf{R_z} = \begin{pmatrix} \cos(t) & \sin(t) & 0 & 0 \\ -\sin(t) & \cos(t) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The complete transformation to rotate a point (x,y,z) about the rotation axis to a new point (x`,y`,z`) is as follows, the forward transforms followed by the reverse transforms.

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \mathbf{T^{-1}\, R_x^{-1}\, R_y^{-1}\, R_z\, R_y\, R_x\, T} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Using quaternions

To rotate a 3D vector "p" by angle theta about a (unit) axis "r" one forms the quaternion

$$Q_1 = (0,p_x,p_y,p_z)$$

and the rotation quaternion

$$Q_2 = (\cos(\text{theta}/2),\ r_x \sin(\text{theta}/2),\ r_y \sin(\text{theta}/2),\ r_z \sin(\text{theta}/2)).$$

The rotated vector is the last three components of the quaternion

$$Q_3 = Q_2\, Q_1\, Q_2^*$$

It is easy to see that rotation in the opposite direction (-theta) can be achieved by reversing the order of the multiplication.

$$Q_3 = Q_2^*\, Q_1\, Q_2$$

Note also that the quaternian Q_2 is of unit magnitude.