This tutorial teaches you about 4×4 matrices. It begins with multiplication and application to vectors. It also describes how 4×4 matrices represent rotations and translations of three-dimensional space. It includes two ways of making 3×3 rotation matrices. It assumes that you have already studied our 2×2 and 3×3 matrix tutorials and both of our vector tutorials.

1 Multiplication

Multiplication of 4×4 and 4×1 matrices is much like multiplication of smaller matrices. If M is 4×4 and $\vec{v}$ is 4×1, then

$$M\vec{v} = \begin{bmatrix} M_{00} & M_{01} & M_{02} & M_{03} \\ M_{10} & M_{11} & M_{12} & M_{13} \\ M_{20} & M_{21} & M_{22} & M_{23} \\ M_{30} & M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} M_{00}v_0 + M_{01}v_1 + M_{02}v_2 + M_{03}v_3 \\ M_{10}v_0 + M_{11}v_1 + M_{12}v_2 + M_{13}v_3 \\ M_{20}v_0 + M_{21}v_1 + M_{22}v_2 + M_{23}v_3 \\ M_{30}v_0 + M_{31}v_1 + M_{32}v_2 + M_{33}v_3 \end{bmatrix}.$$

If N is also 4×4, then MN is a 4×4 matrix whose jth column is M times the jth column of N. In other words,

$$(MN)_{ij} = \sum_{k=0}^{3} M_{ik}N_{kj} = M_{i0}N_{0j} + M_{i1}N_{1j} + M_{i2}N_{2j} + M_{i3}N_{3j}.$$

Geometrically, $M\vec{v}$ is the vector $\vec{v}$ after being transformed by the transformation M, and MN is the composite transformation resulting from N followed in time by M. As always, matrix multiplication is associative and not commutative. The 4×4 identity matrix

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

satisfies $I\vec{v} = \vec{v}$ and $IM = M = MI$ for all M and $\vec{v}$.

2 Homogeneous coordinates

Now we implement three-dimensional rotation and translation using 4×4 matrices, much as we implemented two-dimensional rotation and translation using 3×3 matrices. Any 3×1 vector $\vec{v}$ gets a 1 appended, and any 3×3 matrix M gets a row and column of 0s and 1s appended:

$$\vec{v} = \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ 1 \end{bmatrix}, \quad M = \begin{bmatrix} M_{00} & M_{01} & M_{02} & 0 \\ M_{10} & M_{11} & M_{12} & 0 \\ M_{20} & M_{21} & M_{22} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Translation by a vector $\vec{t}$ manifests as a 4×4 matrix

$$T = \begin{bmatrix} 1 & 0 & 0 & t_0 \\ 0 & 1 & 0 & t_1 \\ 0 & 0 & 1 & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The composite transformation resulting from M followed by T is

$$TM = \begin{bmatrix} 1 & 0 & 0 & t_0 \\ 0 & 1 & 0 & t_1 \\ 0 & 0 & 1 & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} M_{00} & M_{01} & M_{02} & 0 \\ M_{10} & M_{11} & M_{12} & 0 \\ M_{20} & M_{21} & M_{22} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} M_{00} & M_{01} & M_{02} & t_0 \\ M_{10} & M_{11} & M_{12} & t_1 \\ M_{20} & M_{21} & M_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

In computer graphics, we usually use this framework to express rotation followed by translation. So M is a 3×3 matrix (before it is homogenized) that expresses rotation. Rotations are much more complicated in three dimensions than in two, and there are many systems for how to describe rotations. We describe two systems in later sections of this tutorial.

3 Inverses

Sometimes we need to know the inverse of a composite transformation TM. Intuitively, if one transformation is rotation by M followed by translation by $\vec{t}$, then the inverse transformation should be translation by $-\vec{t}$ followed by rotation by M^{-1}. Conveniently, rotation matrices are easy to invert. Define the *transpose* of a 3×3 matrix M to be the 3×3 matrix $M^\top$ obtained by reflecting M across its diagonal:

$$M^\top = \begin{bmatrix} M_{00} & M_{01} & M_{02} \\ M_{10} & M_{11} & M_{12} \\ M_{20} & M_{21} & M_{22} \end{bmatrix}^\top = \begin{bmatrix} M_{00} & M_{10} & M_{20} \\ M_{01} & M_{11} & M_{21} \\ M_{02} & M_{12} & M_{22} \end{bmatrix}.$$

If the 3×3 matrix M represents a rotation, then the inverse rotation happens to be the transpose: $M^{-1} = M^\top$. So inversion is easy, fast, and numerically robust. And the same calculation works when M is homogenized, too. Skipping some details, it turns out that

$$\begin{aligned}
(TM)^{-1} &= M^{-1}T^{-1} \\
&= \begin{bmatrix} M_{00} & M_{01} & M_{02} & 0 \\ M_{10} & M_{11} & M_{12} & 0 \\ M_{20} & M_{21} & M_{22} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 & t_0 \\ 0 & 1 & 0 & t_1 \\ 0 & 0 & 1 & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \\
&= \begin{bmatrix} M_{00} & M_{10} & M_{20} & 0 \\ M_{01} & M_{11} & M_{21} & 0 \\ M_{02} & M_{12} & M_{22} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -t_0 \\ 0 & 1 & 0 & -t_1 \\ 0 & 0 & 1 & -t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} M_{00} & M_{10} & M_{20} & -M_{00}t_0 - M_{10}t_1 - M_{20}t_2 \\ M_{01} & M_{11} & M_{21} & -M_{01}t_0 - M_{11}t_1 - M_{21}t_2 \\ M_{02} & M_{12} & M_{22} & -M_{02}t_0 - M_{12}t_1 - M_{22}t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} M_{00} & M_{10} & M_{20} & \left(-M^\top \vec{t}\right)_0 \\ M_{01} & M_{11} & M_{21} & \left(-M^\top \vec{t}\right)_1 \\ M_{02} & M_{12} & M_{22} & \left(-M^\top \vec{t}\right)_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned}$$

4 Describing rotations in terms of an orthonormal basis

Suppose that we have two length-1, three-dimensional vectors $\vec{u}$ and $\vec{v}$ that are perpendicular to each other. We also have two length-1, three-dimensional vectors $\vec{a}$ and $\vec{b}$ that are perpendicular to each other. There is a unique rotation of three-dimensional space that transforms $\vec{u}$ to $\vec{a}$ and $\vec{v}$ to $\vec{b}$. We wish to find the 3×3 matrix M that describes that rotation.

We compute the cross product $\vec{w} = \vec{u} \times \vec{v}$ and form the 3×3 matrix R with columns $\vec{u}$, $\vec{v}$, $\vec{w}$. Similarly, we form a 3×3 matrix S with columns $\vec{a}$, $\vec{b}$, $\vec{a} \times \vec{b}$. Then we compute $M = SR^\top$.

That's it. To understand what's going on, try applying M to $\vec{u}$ and to $\vec{v}$, symbolically. You should get $\vec{a}$ and $\vec{b}$.

5 Describing rotations in terms of angle and axis

In this section, we discuss how to compute a 3×3 rotation matrix in terms of an axis of rotation and an angle of rotation about that axis.

First we specify a length-1, three-dimensional vector $\vec{u}$ to serve as the axis of rotation. We form the matrix

$$U = \begin{bmatrix} 0 & -u_2 & u_1 \\ u_2 & 0 & -u_0 \\ -u_1 & u_0 & 0 \end{bmatrix}$$

and compute the square

$$U^2 = UU = \begin{bmatrix} -u_1{}^2 - u_2{}^2 & u_0 u_1 & u_0 u_2 \\ u_0 u_1 & -u_0{}^2 - u_2{}^2 & u_1 u_2 \\ u_0 u_2 & u_1 u_2 & -u_0{}^2 - u_1{}^2 \end{bmatrix} = \begin{bmatrix} u_0{}^2 - 1 & u_0 u_1 & u_0 u_2 \\ u_0 u_1 & u_1{}^2 - 1 & u_1 u_2 \\ u_0 u_2 & u_1 u_2 & u_2{}^2 - 1 \end{bmatrix}.$$

The second piece of information that we specify is the rotation angle α. The matrix M that we build will rotate space about $\vec{u}$, through the angle α, counter-clockwise in a right-handed sense. To understand what this means, hold your right hand in the air, with its fingers curled but its thumb pointing out. The thumb is the axis. If $\alpha > 0$, then the fingers point in the direction of rotation. If $\alpha < 0$, then the fingers point opposite to the direction of rotation. The amount of rotation is $|\alpha|$.

Then, according to Rodrigues' rotation formula, the 3×3 matrix M that represents rotation through the angle α about the axis $\vec{u}$ is

$$M = I + (\sin \alpha) U + (1 - \cos \alpha) U^2.$$

The identity matrix I here is 3×3, not 4×4. The $(\sin \alpha)U$ term is the matrix U with each of its entries multiplied by the number $\sin \alpha$. Similarly, $(1 - \cos \alpha)U^2$ equals U^2 scaled by $1 - \cos \alpha$. Finally, M is the matrix sum of those three terms — meaning that they are added entry-by-entry. In other words,

$$M_{ij} = I_{ij} + (\sin \alpha) U_{ij} + (1 - \cos \alpha)(U^2)_{ij}.$$

To get a sense of how Rodrigues' formula works, you might try computing the special case where $\alpha = 0$. You might also try the special case $u_0 = 0$, $u_1 = 0$, $u_2 = 1$. You might also try computing $M\vec{u}$ symbolically, under no special assumptions. Do the answers make sense?