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

1 Multiplication

Multiplication of 3×3 and 3×1 matrices is much like multiplication of 2×2 and 2×1 matrices. If M and N are 3×3 and $\vec{v}$ is 3×1, then

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

and

$$\begin{aligned} MN &= \begin{bmatrix} M_{00} & M_{01} & M_{02} \\ M_{10} & M_{11} & M_{12} \\ M_{20} & M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} N_{00} & N_{01} & N_{02} \\ N_{10} & N_{11} & N_{12} \\ N_{20} & N_{21} & N_{22} \end{bmatrix} \\ &= \begin{bmatrix} M_{00}N_{00} + M_{01}N_{10} + M_{02}N_{20} & M_{00}N_{01} + M_{01}N_{11} + M_{02}N_{21} & M_{00}N_{02} + M_{01}N_{12} + M_{02}N_{22} \\ M_{10}N_{00} + M_{11}N_{10} + M_{12}N_{20} & M_{10}N_{01} + M_{11}N_{11} + M_{12}N_{21} & M_{10}N_{02} + M_{11}N_{12} + M_{12}N_{22} \\ M_{20}N_{00} + M_{21}N_{10} + M_{22}N_{20} & M_{20}N_{01} + M_{21}N_{11} + M_{22}N_{21} & M_{20}N_{02} + M_{21}N_{12} + M_{22}N_{22} \end{bmatrix}. \end{aligned}$$

It is helpful to recognize that the jth column of MN is M times the jth column of N. Or maybe you would prefer a more concise expression:

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

Geometrically, $M\vec{v}$ is the vector $\vec{v}$ after being transformed by the transformation M. Similarly, MN is the composite transformation resulting from N followed in time by M.

Matrix multiplication is associative; for example, $M(N\vec{v}) = (MN)\vec{v}$. However, matrix multiplication is not commutative: $MN \neq NM$ except in special cases. The 3×3 identity matrix

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

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

2 Optional: Determinant and inversion

I think that we never use these concepts in our code, but I'll leave them here for completeness. The determinant of a 3×3 matrix M is

$$\begin{aligned} \det M &= -M_{20}M_{11}M_{02} + M_{10}M_{21}M_{02} + M_{20}M_{01}M_{12} \\ &\quad - M_{00}M_{21}M_{12} - M_{10}M_{01}M_{22} + M_{00}M_{11}M_{22}. \end{aligned}$$

The inverse matrix M^{-1} exists if and only if $\det M \neq 0$. The inverse satisfies $MM^{-1} = I = M^{-1}M$. To compute the inverse, first compute a matrix N by

$$\begin{aligned} N_{00} &= -M_{12}M_{21} + M_{11}M_{22}, \\ N_{10} &= M_{12}M_{20} - M_{10}M_{22}, \\ N_{20} &= -M_{11}M_{20} + M_{10}M_{21}, \\ N_{01} &= M_{02}M_{21} - M_{01}M_{22}, \\ N_{11} &= -M_{02}M_{20} + M_{00}M_{22}, \\ N_{21} &= M_{01}M_{20} - M_{00}M_{21}, \\ N_{02} &= -M_{02}M_{11} + M_{01}M_{12}, \\ N_{12} &= M_{02}M_{10} - M_{00}M_{12}, \\ N_{22} &= -M_{01}M_{10} + M_{00}M_{11}. \end{aligned}$$

Then $M^{-1} = N / \det M$, meaning that $(M^{-1})_{ij} = N_{ij} / \det M$ for all i, j.

3 Homogeneous coordinates

Suppose that I have a 2×1 point $\vec{v}$. I want to transform it by a 2×2 matrix M and then translate it by a 2×1 vector $\vec{t}$. So the final result will be

$$\vec{t} + M\vec{v} = \begin{bmatrix} t_0 \\ t_1 \end{bmatrix} + \begin{bmatrix} M_{00} & M_{01} \\ M_{10} & M_{11} \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \end{bmatrix} = \begin{bmatrix} t_0 + M_{00}v_0 + M_{01}v_1 \\ t_1 + M_{10}v_0 + M_{11}v_1 \end{bmatrix}.$$

It is not possible to express the translation, let alone the composite transformation, as a 2×2 matrix. To work around this problem, we use a mathematical trick (that is not taught in most introductory linear algebra courses).

We append a 1 to the end of any vector $\vec{v}$, so that it becomes a 3×1 matrix:

$$\vec{v} = \begin{bmatrix} v_0 \\ v_1 \\ 1 \end{bmatrix}.$$

Correspondingly, any 2×2 matrix M gets a row and column of 0s and 1s like this:

$$M = \begin{bmatrix} M_{00} & M_{01} & 0 \\ M_{10} & M_{11} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

We call these the *homogeneous* versions of $\vec{v}$ and M. If we multiply them, then we get the homogeneous version of $M\vec{v}$:

$$\begin{bmatrix} M_{00} & M_{01} & 0 \\ M_{10} & M_{11} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ 1 \end{bmatrix} = \begin{bmatrix} M_{00}v_0 + M_{01}v_1 \\ M_{10}v_0 + M_{11}v_1 \\ 1 \end{bmatrix} = \begin{bmatrix} (M\vec{v})_0 \\ (M\vec{v})_1 \\ 1 \end{bmatrix}.$$

So far, the homogeneous versions don't seem to be hurting us much, but they don't seem to be helping us either. They start helping us when we realize that translation can be expressed in this framework. Let T be the matrix

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

Then, for any $\vec{v}$,

$$T\vec{v} = \begin{bmatrix} 1 & 0 & t_0 \\ 0 & 1 & t_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ 1 \end{bmatrix} = \begin{bmatrix} v_0 + t_0 \\ v_1 + t_1 \\ 1 \end{bmatrix}$$

is the homogeneous version of $\vec{v}$ translated by $\vec{t}$.

4 Rotation followed by translation

For computer graphics, the most important example is rotation followed by translation:

$$TM = \begin{bmatrix} 1 & 0 & t_0 \\ 0 & 1 & t_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_0 \\ \sin\theta & \cos\theta & t_1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Suppose that we want to rotate and translate a vector $\vec{v}$, then rotate and translate it again (by a different rotation and translation), then rotate and translate again, and so on. Suppose that there are d rotations and d translations in all. Here are two strategies:

- Don't use homogeneous coordinates. Just apply each of the rotations and translations to $\vec{v}$ in the ordinary way, using 2×2 and 2×1 matrices.

- Do use homogeneous coordinates. So each rotation and translation is a 3×3 matrix. Don't apply them to $\vec{v}$ immediately. First, multiply them together to get a single 3×3 matrix. Then multiply that matrix by $\vec{v}$.

If we want to transform a single $\vec{v}$, then which strategy is faster? If we want to transform many vectors $\vec{v}$ (all by the same sequence of rotations and translations), then which strategy is faster?