This tutorial teaches you about 3×3 matrices. It begins with transforming vectors and composing transformations. Then it goes on to describe how 3×3 matrices can represent rotations and translations of *two*-dimensional space. (In a later tutorial, we go into more depth about how 3×3 matrices describe transformations of three-dimensional space.)

1 Transforming And Composing

A 3×3 matrix M can transform a three-dimensional vector $\vec{v}$ to a new three-dimensional vector $M\vec{v}$, like this:

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

If that looks crazy to you, it might help to compare it to the two-dimensional case. They follow a single pattern; the three-dimensional case just has more of it.

If M and N are both 3×3 matrices, then their product MN is

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

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, MN is the composite transformation resulting from doing N followed in time by M. Matrix multiplication is associative, so $M(N\vec{v}) = (MN)\vec{v}$. 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 Homogeneous coordinates

Suppose that I have a two-dimensional point $\vec{p}$. 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

$$\vec{t} + M\vec{p} = \begin{bmatrix} t_0 \\ t_1 \end{bmatrix} + \begin{bmatrix} M_{00} & M_{01} \\ M_{10} & M_{11} \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \end{bmatrix} = \begin{bmatrix} t_0 + M_{00}p_0 + M_{01}p_1 \\ t_1 + M_{10}p_0 + M_{11}p_1 \end{bmatrix}.$$

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 two-dimensional vector $\vec{p}$, so that it becomes a three-dimensional vector:

$$\vec{p} = \begin{bmatrix} p_0 \\ p_1 \\ 1 \end{bmatrix}.$$

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 $\vec{p}$ and M. If we multiply them, then we get the homogeneous version of $M\vec{p}$:

$$\begin{bmatrix} M_{00} & M_{01} & 0 \\ M_{10} & M_{11} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ 1 \end{bmatrix} = \begin{bmatrix} M_{00}p_0 + M_{01}p_1 \\ M_{10}p_0 + M_{11}p_1 \\ 1 \end{bmatrix} = \begin{bmatrix} (M\vec{p})_0 \\ (M\vec{p})_1 \\ 1 \end{bmatrix}.$$

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 too. Let T be the matrix

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

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

$$T\vec{p} = \begin{bmatrix} 1 & 0 & t_0 \\ 0 & 1 & t_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ 1 \end{bmatrix} = \begin{bmatrix} p_0 + t_0 \\ p_1 + t_1 \\ 1 \end{bmatrix} = \begin{bmatrix} (\vec{p} + \vec{t})_0 \\ (\vec{p} + \vec{t})_1 \\ 1 \end{bmatrix}$$

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

3 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 two-dimensional point $\vec{p}$, 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 $\vec{p}$ 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 $\vec{p}$ immediately. First, multiply them together to get a single 3×3 matrix. Then multiply that matrix by $\vec{p}$.

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