Animated CGEM: Rotation About an Arbitrary Axis

Dr. John McDonald, DePaul University
jmcdonald@cs.depaul.edu

Abstract: *Rotation matrices are one of the first topics covered in introductory graphics courses, and yet the details of arbitrary rotation matrices often get swept under the rug due to their complexity. This CGEM presents a direct, constructive derivation of the matrix for a rotation about an arbitrary axis, enhanced with animations that help build intuition for the calculation. The details of this derivation can be described in an intuitive manner that builds on the procedure for calculating the matrices for rotations about the coordinate axes. This treatment is suitable for use as a handout for students to use as a supplement to the usual course text.*

Submission Title: Rotation About an Arbitrary Axis
Keywords: Basic Graphics Algorithms, Transformations, Mathematical Foundations
Category: Animation

Introduction

One of the first topics discussed in introductory graphics courses is the matrix representation of linear transformations. They form the core of the traditional rendering pipeline, and are the key to hierarchical modeling techniques. It is therefore imperative that students in computer graphics programming courses develop a thorough intuition for how various linear transformations are written as matrices.

This is especially true for students who will be exposed to modern programmable graphics pipelines. The power of modern graphics hardware puts more detailed control of every aspect of the pipeline into the hands of the programmer, including geometric transformations.

Rotations about the coordinate axes are relatively straightforward, and are indeed one of the primary examples often used to motivate the study of transformations. When, however, the subject turns to rotations about an arbitrary axis in 3D space, the computation becomes complicated enough that introductory students can easily get lost. The following is an approach to deriving the matrix for an arbitrary rotation that builds an intuitive understanding of the components of the final formula.

This module uses a derivation similar to the direct constructive method found in [2] but expands considerably on the details of the calculation, and provides several multimedia aids to help students build intuition. In contrast, other texts, such as [1] and [3] prefer to present methods which reduce the problem to rotation about the *z*-axis by conjugating with a coordinate transformation, although [1] does also briefly outline the constructive method as well.

The animations in this module are stored in an *avi* format using the common "cinepak" codec and so should play normally on both PC and Mac platforms. These animations are also available in Microsoft MPEG-4 and DivX MPEG-4 formats at

- http://mcdonald.cs.depaul.edu/ArbitraryRotation/MicrosoftMPEG4.zip
- http://mcdonald.cs.depaul.edu/ArbitraryRotation/DivxMPEG4.zip

Preliminaries

For the following course module, it is assumed that students have already covered the following topics rigorously:

- The basic vector operations including dot and cross products including the length laws for cross products and dot products which state that if ϕ is the angle between two vectors $\boldsymbol{v}$ and $\boldsymbol{w}$, then $\boldsymbol{v}\bullet\boldsymbol{w} = |\boldsymbol{v}||\boldsymbol{w}|\cos\phi$ and $|\boldsymbol{v}\times\boldsymbol{w}| = |\boldsymbol{v}||\boldsymbol{w}|\sin\phi$.
- The concept of a linear transformation, including the geometric effects of rotation, scale and translation.
- The importance of the choice of a coordinate system in which to measure the effects of linear transformations.
- The fact that the three columns of the matrix for a linear transformation are the images of the three coordinate axes under that transformation. So, if T is the linear transformation and M is the corresponding matrix, and if $\text{col}_i(M)$ represents the i^{th} column of M, then:

$$\text{col}_0(M) = T(\langle 1,0,0\rangle), \quad \text{col}_1(M) = T(\langle 0,1,0\rangle), \quad \text{col}_2(M) = T(\langle 0,0,1\rangle)$$

- The matrices for the rotations about the three coordinate axes.

Rotation about an arbitrary axis

Let $\boldsymbol{a}$ be a *unit* vector in 3D space and let θ be an angle measured in radians. Let R be the rotation about $\boldsymbol{a}$ by the angle θ, as shown in Figure 1.

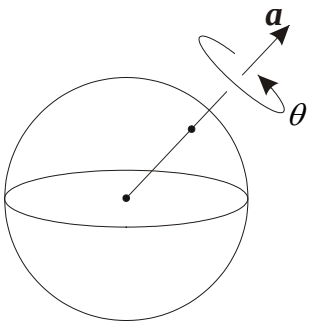

Figure 1: The Angle and Axis of Rotation for *R*

If we fix a reference coordinate system from which to measure the effects of this transformation, then we can calculate the matrix M_R for R by calculating its effects on the three coordinate axes, as shown in Animation 1. To do so, consider the effect of rotating any vector $\boldsymbol{v}$ about $\boldsymbol{a}$. We can be more concrete by looking at the effect of R on the x-axis vector $\langle 1,0,0\rangle$, but it is important to stress that the following calculation will work for any vector.

What effect does the rotation R have on $\boldsymbol{v}$? The vector $\boldsymbol{v}$ will sweep out a cone as shown in Animation 2. As it does so, the tip of $\boldsymbol{v}$ traces out a circle perpendicular to, and centered on the axis $\boldsymbol{a}$, also shown in the animation. Unfortunately, since the rotation axis $\boldsymbol{a}$ is any vector we want to choose, that cone could be pointing in any direction. To quantify the effects of the rotation in a matrix, we need some basis for measuring the path of $\boldsymbol{v}$ as it sweeps out this cone.

As is often true in mathematics, finding an answer to a tricky problem can be made easier by finding the "right perspective". In this case, the right perspective is to rotate our

view so that we are looking straight down the axis ***a*** and with the vector ***v*** pointed towards us and straight out to the right as shown in Figure 2. Animation 3 demonstrates this change of perspective, in the animation the view rotates until the camera is looking straight down the axis ***a,*** the vector in red. We call this perspective the "top-down" view.

The rotation begins to look more familiar. It starts to look like a rotation in 2D, and that is exactly what we have: In a plane perpendicular to ***a***, the motion of a point, or the tip of a vector, under rotation is two-dimensional.

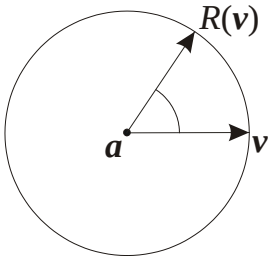

Figure 2: The View Along the Axis

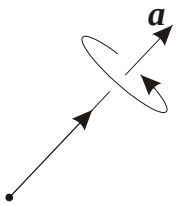

Figure 3: Rotation Parallel to A

To take this further, suppose that ***v*** itself is parallel to ***a***, what happens to it under rotation? Nothing, of course. The rotation happens about the vector ***a,*** and it fixes any vector that is parallel to ***a***, see Figure 3. So, going back to Figure 2, what we are really seeing is the rotation of the part of ***v*** that is perpendicular to ***a***.

So, we decompose ***v*** into two components one $\boldsymbol{v}_{\parallel}$ parallel to ***a*** and one $\boldsymbol{v}_{\perp}$ perpendicular, as shown in Figure 4.

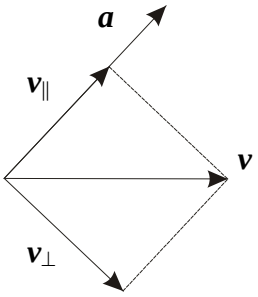

Figure 4: Decomposing *v*

Since rotation is a linear transformation, rotating ***v*** is the same as rotating $\boldsymbol{v}_{\parallel}$ and $\boldsymbol{v}_{\perp}$, as shown in Animation 4. These two components are given by the standard formulas that are made simpler because ***a*** is a unit vector:

$$\boldsymbol{v}_{\parallel} = (\boldsymbol{v} \bullet \boldsymbol{a})\boldsymbol{a}$$

$$\boldsymbol{v}_{\perp} = \boldsymbol{v} - (\boldsymbol{v} \bullet \boldsymbol{a})\boldsymbol{a}$$

Then, since $\boldsymbol{v}_{\parallel}$ is fixed by the rotation, $R(\boldsymbol{v}_{\parallel}) = \boldsymbol{v}_{\parallel}$. So to calculate $R(\boldsymbol{v})$ we can calculate the sum:

$$\begin{aligned} R(\boldsymbol{v}) &= R(\boldsymbol{v}_{\parallel}) + R(\boldsymbol{v}_{\perp}) \\ &= \boldsymbol{v}_{\parallel} + R(\boldsymbol{v}_{\perp}) \end{aligned}$$

To calculate $R(\boldsymbol{v}_{\perp})$, we look again at the top down view, see Figure 5, and we analyze $R(\boldsymbol{v}_{\perp})$ in the same way as for two-dimensional rotations.

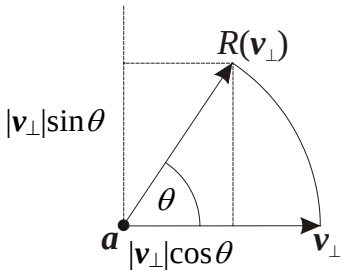

Figure 5: Analyzing the Rotation

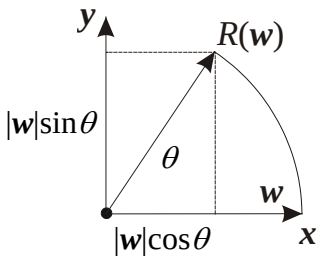

Figure 6: Two Dimensional Rotation

It is clear that, just like the two-dimensional rotation shown in Figure 6, we will write $R(\boldsymbol{v}_\perp)$ as a sum involving cosines and sines of the rotation angle θ. The only question is, what is the direction pointing straight up in Figure 5, the analog of y in the two-dimensional case?

Rotate the view from Figure 5 slightly so that we can see the axis as in Figure 7. Clearly the vector we want is perpendicular to both $\boldsymbol{a}$ and $\boldsymbol{v}_\perp$, that is $\boldsymbol{a} \times \boldsymbol{v}_\perp$.

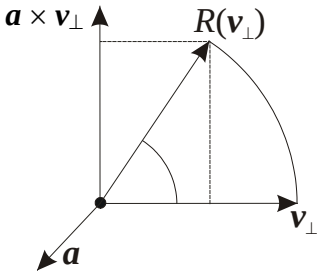

Figure 7: The Vector Perpendicular to $\boldsymbol{v}_\perp$

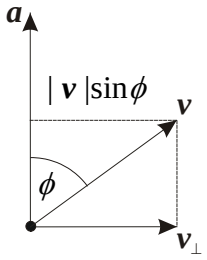

Figure 8: The Relationship Between $\boldsymbol{v}$ and $\boldsymbol{a}$

There are two important facts about $\boldsymbol{a} \times \boldsymbol{v}_\perp$ that are useful. Let ϕ be the angle between the $\boldsymbol{a}$ and $\boldsymbol{v}$ as in Figure 8. Then the length law for cross products gives us two facts:

1. $|\boldsymbol{a} \times \boldsymbol{v}_\perp| = |\boldsymbol{v}_\perp|$, since $\boldsymbol{a}$ and $\boldsymbol{v}_\perp$ are perpendicular and $\boldsymbol{a}$ is a unit vector.
2. $\boldsymbol{a} \times \boldsymbol{v}_\perp = \boldsymbol{a} \times \boldsymbol{v}$. To see this, note that both of these vectors lie perpendicular to $\boldsymbol{a}$, $\boldsymbol{v}$ and $\boldsymbol{v}_\perp$. Further, they have the same length since $|\boldsymbol{a} \times \boldsymbol{v}| = |\boldsymbol{v}|\sin\phi$, and as we can see in Figure 8 this is the same as $|\boldsymbol{v}_\perp| = |\boldsymbol{v}|\sin\phi$, which by our first fact is equal to $|\boldsymbol{a} \times \boldsymbol{v}_\perp|$.

So, we can in fact use $\boldsymbol{a} \times \boldsymbol{v}$ instead of $\boldsymbol{a} \times \boldsymbol{v}_\perp$ in the calculation of the rotation, and all three of the vectors $\boldsymbol{v}_\perp$, $R(\boldsymbol{v}_\perp)$ and $\boldsymbol{a} \times \boldsymbol{v}$ have the same length! Therefore,

$$\begin{aligned} R(\boldsymbol{v}) &= \boldsymbol{v}_\parallel + R(\boldsymbol{v}_\perp) \\ &= \boldsymbol{v}_\parallel + \boldsymbol{v}_\perp \cos\theta + (\boldsymbol{a} \times \boldsymbol{v})\sin\theta \end{aligned}$$

Animation 5 demonstrates this rotation in relationship to all of these vectors.

Of course, we could use this formula to calculate the rotation of any given vector, but our goal is to calculate the matrix for R. This we do by applying the above formula to the coordinate axis vectors. For example, to calculate the first column of the matrix, consider the vector $\boldsymbol{x} = \langle 1, 0, 0 \rangle$, and suppose $\boldsymbol{a} = \langle a_x, a_y, a_z \rangle$, then

$$
\begin{aligned}
\boldsymbol{x}_{\parallel} &= (\boldsymbol{x} \bullet \boldsymbol{a})\boldsymbol{a} \\
&= a_x \boldsymbol{a} = \langle a_x{}^2, a_x a_y, a_x a_z \rangle \\
\boldsymbol{x}_{\perp} &= \langle 1,0,0 \rangle - \boldsymbol{x}_{\parallel} \\
&= \langle 1,0,0 \rangle - \langle a_x{}^2, a_x a_y, a_x a_z \rangle \\
&= \langle 1 - a_x{}^2, a_x a_y, a_x a_z \rangle \\
\boldsymbol{a} \times \boldsymbol{x} &= \langle a_x, a_y, a_z \rangle \times \langle 1,0,0 \rangle \\
&= \langle 0, a_z, -a_y \rangle
\end{aligned}
$$

and so the first column of the matrix M_R is

$$
\begin{aligned}
R(\boldsymbol{x}) &= \boldsymbol{v}_{\parallel} + \boldsymbol{v}_{\perp} \cos\theta + (\boldsymbol{a} \times \boldsymbol{v}) \sin\theta \\
&= \begin{bmatrix} a_x{}^2 \\ a_x a_y \\ a_x a_z \end{bmatrix} + \cos\theta \begin{bmatrix} 1 - a_x{}^2 \\ -a_x a_y \\ -a_x a_z \end{bmatrix} + \sin\theta \begin{bmatrix} 0 \\ a_z \\ -a_y \end{bmatrix} \\
&= \begin{bmatrix} a_x{}^2 (1 - \cos\theta) + \cos\theta \\ a_x a_y (1 - \cos\theta) + a_z \sin\theta \\ a_x a_z (1 - \cos\theta) - a_y \sin\theta \end{bmatrix}
\end{aligned}
$$

it is left as an exercise to calculate the second and third columns of the final matrix, as they have a similar form. To shorten its final form, we let $c = \cos\theta$ and $s = \sin\theta$.

$$
M_R = \begin{bmatrix} a_x{}^2 (1-c) + c & a_x a_y (1-c) - a_z s & a_x a_z (1-c) + a_y s \\ a_x a_y (1-c) + a_z s & a_y{}^2 (1-c) + c & a_y a_z (1-c) - a_x s \\ a_x a_z (1-c) - a_y s & a_y a_z (1-c) + a_x s & a_z{}^2 (1-c) + c \end{bmatrix}
$$

It is important to stress the fact that $\boldsymbol{a}$ must be a unit vector. If it is not, the resulting formula will be incorrect, although it is useful to explore how it will be incorrect. Many computer graphics systems treat this incorrectly for the sake of convenience. They will often multiply the angle of rotation by the length of the vector. If we investigate closely the nature of M_R, it becomes clear that if $\boldsymbol{a}$ is not a unit vector, then M_R is not a rotation at all. It is not even the composition of a scale and a rotation!

References

[1] F.S. Hill, Jr, *Computer Graphics Using OpenGL, Second Edition*, Prentice Hall, 2001.

[2] E. Lengyel, *Mathematics for 3D Game Programming & Computer Graphics, Second Edition*, Charles River Media, 2004.

[3] P. Shirley, *Fundamentals of Computer Graphics*, A.K. Peters Press, 2002.