III.4

FAST RANDOM ROTATION MATRICES

James Arvo
Cornell University
Ithaca, New York

In a previous Gem (Arvo, 1991), I described a method for generating random rotation matrices based on random unit quaternions. As Ken Shoemake points out in his Gem (III.6) that algorithm was flawed in that it did not generate *uniformly distributed* rotation matrices. For a method based on quaternions that corrects this defect see his algorithm. In this Gem I describe a completely different approach to solving the same problem that has the additional benefit of being slightly faster than the previous method. The approach is based on the following fact:

> *To generate uniformly distributed random rotations of a unit sphere, first perform a random rotation about the vertical axis, then rotate the north pole to a random position.*

The first step of this prescription is trivial. Given a random number, x_1, between 0 and 1, the matrix R does the trick:

$$R = \begin{bmatrix} \cos(2\pi x_1) & \sin(2\pi x_1) & 0 \\ -\sin(2\pi x_1) & \cos(2\pi x_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \qquad (1)$$

Here we are assuming that the z-axis is the vertical axis, so the "north pole" will be the point $z = (0, 0, 1)$. The second operation is not quite so obvious, but fortunately it can be carried out quite efficiently. Observe that we can take the point z to any other point p on the sphere via a

ISBN 0-12-409670-0 (IBM)
ISBN 0-12-409671-9 (Mac)

reflection through the plane orthogonal to the line $\overline{zp}$ and containing the origin. Such a reflection is given by the *Householder matrix*

$$H = I - 2vv^{\mathrm{T}}, \tag{2}$$

where v is a unit vector parallel to $\overline{zp}$ (see, for instance, Golub and Van Loan, 1985). To turn this into a rotation we need only apply one more reflection (making the determinant positive). A convenient reflection for this purpose is reflection through the origin—that is, scaling by -1. Thus, the final rotation matrix can be expressed as the product

$$M = -HR, \tag{3}$$

where R is the simple rotation in Eq. (1). The rotation matrix M will be uniformly distributed within SO(3), the set of all rotations in three-space, if H takes the north pole to every point on the sphere with equal probability density. This will hold if the image of z under the random reflection is such that both its azimuth angle and the cosine of its elevation angle are uniformly distributed. The matrix H in Eq. (2) will satisfy these requirements if we let

$$v = \begin{bmatrix} \cos(2\pi x_2)\sqrt{x_3} \\ \sin(2\pi x_2)\sqrt{x_3} \\ \sqrt{1 - x_3} \end{bmatrix}, \tag{4}$$

where x_2 and x_3 are two independent uniform random variables in $[0, 1]$. To show this we need only compute $p = Hz$ and verify that p is distributed appropriately. Using the preceding definition of v, we have

$$p = z - 2vv^{\mathrm{T}}z = \begin{bmatrix} -2\cos(2\pi x_2)\sqrt{x_3(1 - x_3)} \\ -2\sin(2\pi x_2)\sqrt{x_3(1 - x_3)} \\ 2x_3 - 1 \end{bmatrix}. \tag{5}$$

Because the third component of p is the cosine of its elevation angle, we see immediately that it is uniformly distributed over $[-1, 1]$, as required.

random_rotation(x_1, x_2, x_2, M)

x_1, x_2, x_3 : **real**; *Three random variables.*

M : **matrix3**; *The resulting matrix.*

begin

$\theta \leftarrow 2\pi x_1$; *Pick a rotation about the pole.*

$\phi \leftarrow 2\pi x_2$; *Pick a direction to deflect the pole.*

$z \leftarrow x_3$; *Pick the amount of pole deflection.*

Construct a vector for performing the reflection.

$$V \leftarrow \begin{bmatrix} \cos\phi\sqrt{z} \\ \sin\phi\sqrt{z} \\ \sqrt{1-z} \end{bmatrix}$$

Construct the rotation matrix by combining two simple rotations: first rotate about the Z axis, then rotate the Z axis to a random orientation.

$$M \leftarrow (2VV^{\mathrm{T}} - I) \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

end

Figure 1. An efficient procedure for creating random 3×3 rotation matrices.

Similarly, from its first two components we see that the azimuth angle of p is $2\pi x_2$, which is uniformly distributed over $[0, 2\pi]$.

The complete algorithm combining the reflection and simple rotation is shown in Fig. 1, and an optimized version in C appears in the appendix. Procedure "random_rotation" requires three uniformly distributed random numbers between 0 and 1. Supplying these values as arguments has several advantages. First, the procedure can be used in conjunction with your favorite pseudorandom number generator, and there are a great many to choose from. Secondly, if we obtain the three random numbers by *stratified* or *jittered* sampling of the unit cube, the resulting rotation

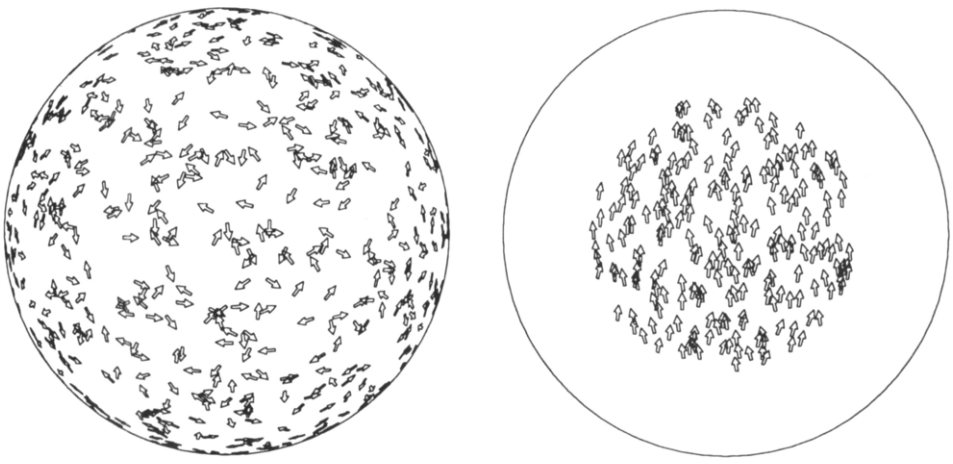

Figure 2.

matrices will inherit the benefits—namely, less clumping. Finally, if we restrict the range of the random input variables (while maintaining their uniformity), we can generate uniformly distributed perturbations or "wobbles" within given limits.

Figure 2a shows the result of applying 1,000 random rotations to a sphere with an arrow painted at one pole. The resulting pattern looks much the same from any vantage point, providing visual confirmation of uniformity. Figure 2b was generated by restricting x_1 and x_3 to the range $[0, 0.1]$.

See also G2, 355; G3, C.6.