

# 3-Dimensional Transformations

Manny Barreto, Will DiBiagio, Nick Morgan, Alex Wong

## 1 Preliminaries

### 1.1 Matrix Inverse

A matrix is said to be invertible if and only if it satisfy the following constraint:

$$AB = BA = I \tag{1}$$

In which case  $B$  is the inverse of  $A$ , and we can write  $B = A^{-1}$ . We can equivalently say that  $A$  is the inverse of  $B$  and write  $A = B^{-1}$ . Hence, we can write:

$$AA^{-1} = BB^{-1} = I \tag{2}$$

Suppose the matrix  $A$  performs a transformation. Since the identity matrix signifies an identity transformation (i.e. nothing has changed  $AI = A$ ),  $AA^{-1} = I$  suggests that  $A^{-1}$  “undos” the transformation.

To compute the inverse of a matrix, we can perform Gauss-Jordan elimination (a method of solving a system of linear equations). Gauss-Jordan elimination allow us to perform a set of actions known as elementary row operations:

1. Adding rows
2. Swapping rows
3. Multiplying rows by a scalar

We begin with a matrix that we would like to find the inverse of:

$$A = \begin{bmatrix} 3 & 0 & 2 \\ 2 & 0 & -2 \\ 0 & 1 & 1 \end{bmatrix} \quad (3)$$

We now concatenate it with the identity matrix  $I$  to create an augmented matrix  $[A|I]$  with the goal of performing a set of elementary row operations to transform the augmented matrix into the form  $[I|A^{-1}]$ :

$$\left[ \begin{array}{ccc|ccc} 3 & 0 & 2 & 1 & 0 & 0 \\ 2 & 0 & -2 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array} \right] \quad (4)$$

We start by eliminating the 2 from the first row by adding the second to the first:

$$\left[ \begin{array}{ccc|ccc} 3+2 & 0 & 2-2 & 1 & 0+1 & 0 \\ 2 & 0 & -2 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array} \right] = \left[ \begin{array}{ccc|ccc} 5 & 0 & 0 & 1 & 1 & 0 \\ 2 & 0 & -2 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array} \right] \quad (5)$$

We can divide the first row by 5 to get the first 1 of the identity:

$$\left[ \begin{array}{ccc|ccc} 5/5 & 0 & 0 & 1/5 & 1/5 & 0 \\ 2 & 0 & -2 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array} \right] = \left[ \begin{array}{ccc|ccc} 1 & 0 & 0 & 1/5 & 1/5 & 0 \\ 2 & 0 & -2 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array} \right] \quad (6)$$

We can now use the first row to eliminate everything in the first column:

$$\left[ \begin{array}{ccc|ccc} 1 & 0 & 0 & 1/5 & 1/5 & 0 \\ 0 & 0 & -2 & -2/5 & 3/5 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array} \right] \quad (7)$$

We can now divide the second row by 2:

$$\left[ \begin{array}{ccc|ccc} 1 & 0 & 0 & 1/5 & 1/5 & 0 \\ 0 & 0 & -2/2 & -2/10 & 3/10 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array} \right] = \left[ \begin{array}{ccc|ccc} 1 & 0 & 0 & 1/5 & 1/5 & 0 \\ 0 & 0 & -1 & -2/10 & 3/10 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array} \right] \quad (8)$$

We can multiply the second row by -1 and swap the third and second row and multiply to get the last diagonal of the identity:

$$\left[ \begin{array}{ccc|ccc} 1 & 0 & 0 & 1/5 & 1/5 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 2/10 & -3/10 & 0 \end{array} \right] \quad (9)$$

And lastly use the third row to eliminate the second row:

$$\left[ \begin{array}{ccc|ccc} 1 & 0 & 0 & 1/5 & 1/5 & 0 \\ 0 & 1 & 0 & -2/10 & 3/10 & 1 \\ 0 & 0 & 1 & 2/10 & -3/10 & 0 \end{array} \right] \quad (10)$$

Our inverse is now:

$$A^{-1} = \begin{bmatrix} 1/5 & 1/5 & 0 \\ -2/10 & 3/10 & 1 \\ 2/10 & -3/10 & 0 \end{bmatrix} \quad (11)$$

We can now apply this method to generically find the inverse of our transformation matrices. For example, let's perform this to a translation matrix  $T_{a,b}$  to find its inverse  $T^{-1}$ :

$$\left[ \begin{array}{ccc|ccc} 1 & 0 & a & 1 & 0 & 0 \\ 0 & 1 & b & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array} \right] \quad (12)$$

We can eliminate the second row by subtracting  $b$  times the third row:

$$\left[ \begin{array}{ccc|ccc} 1 & 0 & a & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & -b \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array} \right] \quad (13)$$

And similarly eliminate the first row by subtracting  $a$  times the first:

$$\left[ \begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & -a \\ 0 & 1 & 0 & 0 & 1 & -b \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array} \right] \quad (14)$$

The inverse of  $T_{a,b}$  is:

$$T^{-1} = \begin{bmatrix} 1 & 0 & -a \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix} \quad (15)$$

## 1.2 Cross Product

The cross ( $\times$ ) of two vectors is defined in 3-dimensional space. The cross of  $\vec{a}$  and  $\vec{b}$  will give us a new vector  $\vec{c}$  that is orthogonal or perpendicular to both  $\vec{a}$  and  $\vec{b}$ .

$$\vec{c} = \vec{a} \times \vec{b} = \|\vec{a}\| \|\vec{b}\| \sin(\theta) \vec{n} \quad (16)$$

where  $\vec{n}$  is the normal vector to  $\vec{a}$  and  $\vec{b}$ , and  $\theta$  is the angle between  $\vec{a}$  and  $\vec{b}$ .

We can redefine this using the components of  $\vec{a} = \langle a_x, a_y, a_z \rangle$  and  $\vec{b} = \langle b_x, b_y, b_z \rangle$  by computing each component using the product of the cross of the other components:

$$\vec{c} = \vec{a} \times \vec{b} = \begin{vmatrix} x & y & z \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} = \begin{bmatrix} a_y b_z - a_z b_y \\ -a_x b_z + a_z b_x \\ a_x b_y - a_y b_x \end{bmatrix} \quad (17)$$

Note that  $\vec{a} \times \vec{b}$  is orthogonal to both  $\vec{a}$  and  $\vec{b}$ ; however,  $\vec{a}$  and  $\vec{b}$  may not be orthogonal themselves. Additionally, the cross between a vector  $\vec{a}$  and itself is 0,  $\vec{a} \times \vec{a} = 0$ .

Note that the cross product of two vectors  $\vec{a} \times \vec{b}$  can be visualized by the right hand rule given by index ( $\vec{a}$ ), middle ( $\vec{b}$ ) and the resulting vector thumb ( $\vec{a} \times \vec{b}$ ).

Note: the computation of the cross product is essentially the determinant. The determinant of a matrix  $\det(A)$  is the volume of a parallelepiped spanned by the column or row vectors of the matrix.

$$\det(A) = x(a_y b_z - a_z b_y) + -y(a_x b_z - a_z b_x) + z(a_x b_y - a_y b_x) \quad (18)$$

## 2 Defining a 3D Coordinate Frame

Similar to a 2D coordinate frame, a 3D coordinate frame can be defined using three vectors (axes) and a point (origin). We begin by defining a 3D coordinate frame using vectors  $\vec{u}$ ,  $\vec{v}$ ,  $\vec{w}$  and the origin  $O$ . To define a point  $p$  in this 3D coordinate frame, we can express  $p$  using coefficients  $x$ ,  $y$  and  $z$ :

$$p = x\vec{u} + y\vec{v} + z\vec{w} + O \quad (19)$$

with  $\vec{u}$ ,  $\vec{v}$  and  $\vec{w}$  as unit vectors. To keep our notation in homogeneous coordinate, we will expand our matrices to 4-dimensional (4D) denoted as  $[x, y, z, 1]$

We now write  $p = x\vec{u} + y\vec{v} + z\vec{w} + O$  into their vector forms using homogeneous coordinates:

$$p = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} \vec{u} \\ \vec{v} \\ \vec{w} \\ O \end{bmatrix} = x\vec{u} + y\vec{v} + z\vec{w} + O \quad (20)$$

### 3 3D Transformations

We may place objects in this 3D space as points or sets of points. Our goal is to perform transformations in 3D space to some point  $p = (x, y, z)$ .

#### 3.1 Translation

We want to move the point  $p = (x, y, z)$  to a new position  $p' = (x+a, y+b, z+c)$ . This naturally becomes a mapping from  $[x, y, c, 1] \rightarrow [x+a, y+b, z+c, 1]$ . We will need a  $4 \times 4$  matrix:

$$T_{a,b,c} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (21)$$

This matrix will perform a 3D translation to any point  $p = (x, y, z)$  to a new location with offset  $(a, b, c)$  for  $p' = (x+a, y+b, z+c)$ :

$$p' = T_{a,b,c} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x+a \\ y+b \\ z+c \\ 1 \end{bmatrix} \quad (22)$$

#### 3.2 Scaling

We can similarly compose the scaling matrix creating a mapping from  $[x, y, c, 1] \rightarrow [ax, by, cz, 1]$ :

$$S_{a,b,c} = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (23)$$

This matrix will perform a 3D scaling to any point  $p = (x, y, z)$  by  $(a, b, c)$

$$p' = S_{a,b,c} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} ax \\ by \\ cz \\ 1 \end{bmatrix} \quad (24)$$

### 3.3 Rotation

Note that both translation and scaling acts along an axis; however rotation in 3D is performed around a given axis, making it a bit more interesting than translation, scaling as well as its 2D counterpart. Given a 3D space, there exists an arbitrary number of possible axes. We will begin by rotating about well defined axes and then around any arbitrary axes.

### 3.4 Rotation (z-Axis)

We can image that the z-axis is the direction in which we view the 2D plane defined by  $x$  and  $y$  in the 2D scenario. To directly extend from the 2D rotational matrix, we first look at the rotation about the z-axis.

$$R_{z,\theta} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (25)$$

If we examine the components in each axis, we see that the z-axis are all 0's which means that if we rotate about the z-axis, the z-coordinate does not change.

Let's see how this would work in an example. Suppose we would like to rotate a point from  $(1, 0, 0) \rightarrow (0, 1, 0)$ . To accomplish this we need a  $90^\circ$  rotation about the z-axis:

$$\begin{aligned}
 p' = R_{z,90^\circ} p &= \begin{bmatrix} \cos(90^\circ) & -\sin(90^\circ) & 0 & 0 \\ \sin(90^\circ) & \cos(90^\circ) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}
 \end{aligned} \tag{26}$$

### 3.5 Rotation (x-Axis)

To begin, rotating about the x-axis is similar that rotating about the z-axis in that the rotation still follows the right-hand-rule. Note that the right hand rule in this case curls towards the extension of the z-axis. To construct the rotational matrix for x-axis, we must observe that the x-component (by definition of rotation about the x-axis) cannot change. Hence, we have the constraint the components affected by the x-component must not change:

$$R_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \\ 0 & \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{27}$$



Now, suppose we would like to rotate a point from  $(0, 1, 0) \rightarrow (0, 0, 1)$ . To accomplish this we need a  $90^\circ$  rotation about the x-axis:

$$\begin{aligned}
 p' = R_{z,90^\circ} p &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(90^\circ) & -\sin(90^\circ) & 0 \\ 0 & \sin(90^\circ) & \cos(90^\circ) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}
 \end{aligned} \tag{28}$$

### 3.6 Rotation (y-Axis)

Now we would imagine that the extension to the y-axis would be similar to that of the x-axis and z-axis. We know that to construct the rotational matrix for y-axis, we must observe that the y-component (again by definition of rotation about the y-axis) cannot change:

$$R_{y,\psi} = \begin{bmatrix} \cos(\psi) & 0 & -\sin(\psi) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\psi) & 0 & \cos(\psi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{29}$$

Now, suppose we would like to rotate a point from  $(0, 0, 1) \rightarrow (1, 0, 0)$ . To accomplish this we need a  $90^\circ$  rotation about the y-axis:

$$\begin{aligned}
 p' = R_{y,90^\circ} p &= \begin{bmatrix} \cos(90^\circ) & 0 & -\sin(90^\circ) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(90^\circ) & 0 & \cos(90^\circ) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix}
 \end{aligned} \tag{30}$$

Notice that the result is actually incorrect; we have turned the wrong direction! Hence technically, we needed to turn  $-90^\circ$ . However, this is inconvenient, so instead, we set the z-component of the first row to be  $\sin(\psi)$  and the z-component of the first column to be  $-\sin(\psi)$  to produce a rotation matrix that adheres to the right-hand-rule.

$$R_{y,\psi} = \begin{bmatrix} \cos(\psi) & 0 & \sin(\psi) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\psi) & 0 & \cos(\psi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{31}$$

Now if we were to reapply the rotation about the y-axis, we expect to get the correct results of  $(0, 0, 1) \rightarrow (1, 0, 0)$ :

$$\begin{aligned}
 p' = R_{y,90^\circ} p &= \begin{bmatrix} \cos(90^\circ) & 0 & \sin(90^\circ) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(90^\circ) & 0 & \cos(90^\circ) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}
 \end{aligned} \tag{32}$$

### 3.7 Rotation About An Arbitrary Axis

Suppose we would like to rotate an arbitrary axis by an angle  $\alpha$  where the axis is defined by a point  $q = (q_x, q_y, q_z)$  and a vector  $\vec{v} = \langle v_x, v_y, v_z \rangle$ . To rotate by  $\alpha$  along this axis, we need to massage the axis to a known axis (namely the z-axis as it is the most standard one). Once it is co-linear with the known axis, we can rotate by  $\alpha$  and massage the to the original position.

(1) As rotations operate with respect to the origin, we begin by translation the axis and the points by the offset of  $\vec{T} = O - q = \langle 0 - q_x, 0 - q_y, 0 - q_z \rangle$ , we will generically refers to this as  $\vec{T} = \langle t_x, t_y, t_z \rangle$ :

$$T_{t_x, t_y, t_z} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{33}$$

(2) Next we need to transform the points and the axis such that it is co-planar with a known plane (namely, y-z plane). We then rotate about the y-axis by some angle  $\psi$  such that it is co-planar to the y-z plane:

$$R_{y,\psi} = \begin{bmatrix} \cos(\psi) & 0 & \sin(\psi) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\psi) & 0 & \cos(\psi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (34)$$

However, what is  $\psi$  exactly? Is there a way to know what  $\psi$  is? If we look at it geometrically, we can decompose the vector  $\vec{v}$  into its components  $\langle v_x, v_y, v_z \rangle$ . We can see that to rotate  $\vec{v}$  to the y-z plane, the angle must be  $\theta = \tan^{-1}(\frac{v_x}{v_z})$  (which also aligns with our conclusion that the y-component should not be affected).

Notice that we have rotated in the opposite direction. In this case, we will need to rewrite  $\psi$  as the following:

$$\psi = 360^\circ - \tan^{-1}(\frac{v_x}{v_z}) = -\theta \quad (35)$$

(3) Now that we are co-planar with the y-z plane, we will need to rotate about the x-axis by an angle  $\phi$  to become co-linear with the z-axis:

$$R_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \\ 0 & \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (36)$$

Again, we will need to figure out what  $\phi$  is. Since we are co-planar with y-z, there is no x-component (which is good since we will be rotating about the x-axis). Note that we have rotated the vector about the y-axis, this is a transformation of  $\vec{v}$ 's z-component using  $v_x$  and  $v_z$ . The new  $v'_z = \sqrt{v_x^2 + v_z^2}$ . Since the y-component remains unchanged, we can again apply the same geometric relation to obtain:

$$\phi = \tan^{-1}(\frac{v_y}{v'_z}) = \tan^{-1}(\frac{v_y}{\sqrt{v_x^2 + v_z^2}}) \quad (37)$$

(4) At this point, we are now co-linear with the z-axis. Hence, we can rotate by  $\alpha$  using the z-axis rotation matrix:

$$R_{z,\alpha} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (38)$$

(5) Since we have centered ourselves at the origin, we must rotate and translate back to the original frame of reference. We can simply use the inverse of our translation  $T_{t_x,t_y,t_z}^{-1}$ ,  $R_{y,\psi}^{-1}$ ,  $R_{x,\phi}^{-1}$ .

We arrive at our final set of transformation  $g$  where  $g$  is some global transformation matrix we may apply generically to any set of points:

$$g = T_{t_x,t_y,t_z}^{-1} R_{y,\psi}^{-1} R_{x,\phi}^{-1} R_{z,\alpha} R_{x,\phi} R_{y,\psi} T_{t_x,t_y,t_z} \quad (39)$$

Note that the order in which we apply the matrices matter! The new set of points  $p'$  will simply be  $p' = gp$ .

## 4 Degrees of Freedom

Now that we are able to translate, scale, and rotate in 3 dimensional space, we notice that there exists six degrees of freedom (DoF). The degrees of freedom of an object refers to the number of ways by which it can move. For 3 dimensions, the six degrees of freedom include:

1. Translating along x-axis
2. Translating along y-axis
3. Translating along z-axis
4. Rotating about x-axis (pitch)
5. Rotating about y-axis (yaw)
6. Rotating about z-axis (roll)

However, the degrees of freedom are not as free as one would think. Suppose we represent the orientation (or pose) of an object (e.g. an airplane) by its position  $(x, y, z)$  and three gimbals (rings representing the rotation about  $x$ ,  $y$ , and  $z$ -axis) in 3D space.

If one were to pitch the object  $90^\circ$  or  $-90^\circ$  (rotation about the  $x$ -axis), the  $z$ -axis of the object would align with its  $y$ -axis. If we now roll the object (rotation about the  $z$ -axis, now aligned with  $y$ -axis), we in fact induce yaw! This phenomenon is known as Gimbal Lock, where two gimbals are aligned parallel with each other causing a loss of a degree of freedom.

This phenomenon is possible for any three parameter representation of rotation (e.g.  $x$ ,  $y$ ,  $z$  rotation matrices, also known as Euler angles). To avoid this phenomenon, one would need a 4 parameter representation (e.g. quaternions).