Chapter 6

3-D Transformational Geometry

In the previous chapter, we showed how 2-D points could be represented by homogeneous three-element vectors, and any affine transformation can be applied by the multiplication of a 3×3 matrix. Similarly we will represent a 3-D point using homogeneous coordinates as a four-element vector, and use a 4×4 transformation matrix to transform them. In general

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} ax+by+cz+d \\ ex+fy+gz+h \\ ix+jy+kz+l \\ 1 \end{bmatrix}. \tag{6.1}$$

By properly selecting the matrix values $a, b, c, \ldots$, these matrices can be configured to apply any affine transformation, to change the size, proportions, position or orientation of a 3-D shape.

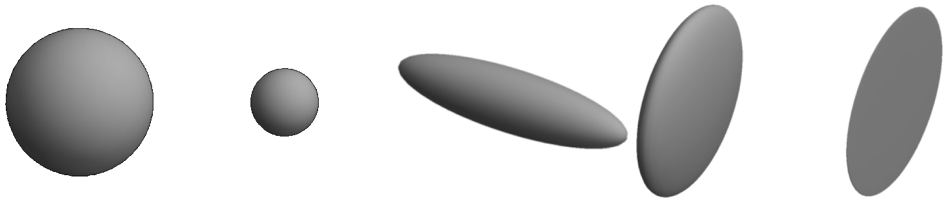
6.1 Scale and Translation

The 4×4 transformation matrices for scaling and translating 3-D points as homogeneous four-element vectors are are simple extensions of the 3×3 transformations used for 2-D points.

We can scale an object in 3-D, by a uniform scale factor s, by multiplying a scale matrix times the homogeneous coordinates of its vertices,

$$\begin{bmatrix} s & & & \\ & s & & \\ & & s & \\ & & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} sx \\ sy \\ sz \\ 1 \end{bmatrix}. \tag{6.2}$$

We use the shorthand $S(s)$ to denote a uniform scale matrix with scale factor s.



$S(1,1,1)$ $S(\frac{1}{2},\frac{1}{2},\frac{1}{2})$ $S(\frac{1}{2},\frac{1}{2},2)$ $S(1,1,\frac{1}{4})$ $S(1,1,0)$

Figure 6.1: A sphere scaled by various factors.

A 3-D stretch and squash transformation is similarly defined using a separate horizontal, vertical and depth scaling factors h, v and d to scale x, y and z,

$$\begin{bmatrix} h & & & \\ & v & & \\ & & d & \\ & & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} hx \\ vy \\ dz \\ 1 \end{bmatrix}. \tag{6.3}$$

We use the shorthand $S(h, v, d)$ to indicate such a matrix. These scaling factors stretch when greater than one, and squash when less than one.

Also similar to the 2-D case, we translate a 3-D space position (x, y, z) by the offset vector (a, b, c) using a matrix-vector product enables by homogeneous coordinates

$$\begin{bmatrix} 1 & & & a \\ & 1 & & b \\ & & 1 & c \\ & & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x+a \\ y+b \\ z+c \\ 1 \end{bmatrix}. \tag{6.4}$$

Hence, as in the 2-D case, homogeneous coordinates enable us to represent any number of scales and translations (and any other affine transformations) in any order with a single transformation matrix that is the product of these transformation matrices.

6.2 Left-Handed v. Right-Handed Coordinate Systems

In 2-D, we use the convention that rotations by positive angles occur counter-clockwise, and that the positive y axis is rotated 90° from the positive x axis. We tend to draw the positive x axis extending horizontally to the right, so the y axis extends vertically upward. This is a convention. One could also define positive rotations to occur clockwise and extend the positive y axis downward from a positive x axis extending right, but such a configuration is non-standard and would seem unintuitive to anyone already familiarized to the counter-clockwise convention.

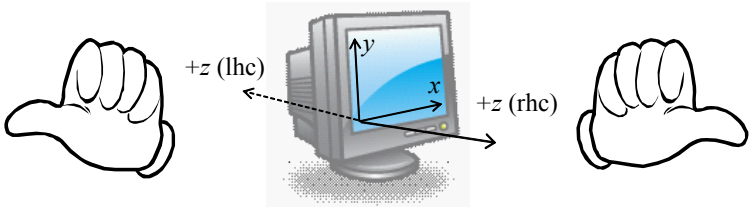


Figure 6.2: Left-handed v. right-handed coordinate systems. If the $+x$ direction extends right, and the $+y$ direction extends up, then the $+z$ direction either extends into the screen in a left-handed coordinate system, or extends toward the viewer in a right-handed coordinate system.

In 3-D, there is also an arbitrary orientation choice. Using the convention that positive x extends right and positive y extends up, then the positive z axis can extend either toward, or away from, the viewer. If the $+z$ axis extends toward the viewer, then we say the coordinate system is *right-handed*. If we can place our right hand at the origin and fold our fingers from the $+x$ axis to the $+y$ axis, then our right-hand's thumb extends in the $+z$ direction. Conversely, if the $+z$ axis extents away from the viewer, then we say the coordinate system is *left-handed*, and we must use our left-hand's fingers to fold from the $+x$ direction to the $+y$ direction, and our left-hand's thumb extends in the $+z$ direction.

Different applications use different coordinate systems, so, unlike the 2-D clockwise convention, there is not a strong convention between left- and right-handed 3-D coordinate systems. For this book (as is mostly the case for modern computer graphics), we will always use right-handed 3-D coordinate systems.

The matrix

$$S(1,1,-1) = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & -1 & \\ & & & 1 \end{bmatrix} \tag{6.5}$$

converts between a left-handed and right-handed coordinate system, by inverting only the z coordinate. Similarly, the inversions $S(-1,1,1)$, $S(1,-1,1)$ and $S(-1,-1,-1)$ would also convert between left-handed and right-handed coordinate systems, though in non-intuitive ways since the primary difference between left-handed and right-handed coordinate systems in most real-world applications is the sign of the z coordinate.

6.3 Vector Arithmetic

As did the previous chapter's discussion of homogeneous coordinates for 2-D graphics, we will use the homogeneous coordinate to differentiate points and vectors in 3-D as well. We will represent points and vectors in 3-

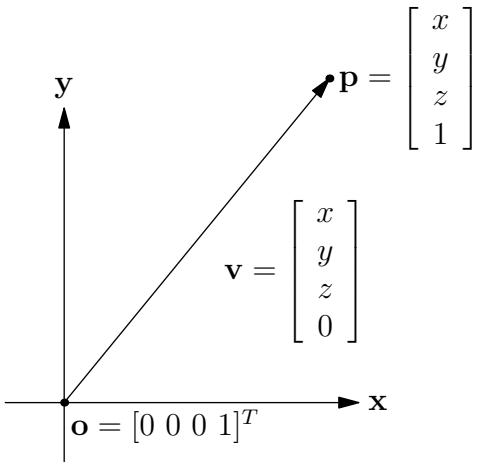


Figure 6.3: Homogenous representations of points and vectors.

D both with 4-element column vectors $[x\ y\ z\ w]^T$. The difference is that the fourth (homogeneous) element w of a vector will be zero whereas the fourth (homogeneous) element w of a point will be non-zero. For now, w will always equal one for points. Hence $\mathbf{p} = [1\ 2\ 3\ 1]^T$ represents the point at 3-D position $(1, 2, 3)$ whereas $\mathbf{v} = [1\ 2\ 3\ 0]$ represents the vector $(1, 2, 3)$. The origin $\mathbf{o}$ is the point $[0\ 0\ 0\ 1]$, so $\mathbf{v} = \mathbf{p} - \mathbf{o}$.

For 3-D computer graphics, we will rely on a lot of vector arithmetic, mostly using the dot product and cross product. These operations are designed to work on ordinary three element vectors. Even though they can be extended to work on homogeneous vectors, the extensions are a bit ugly and cumbersome. Because of that, we tend to work with ordinary vectors when performing the vector arithmetic needed to derive and analyze properties, and then switch to the homogeneous representation for vectors when working on their implementation in the pipeline, or when we need the projective properties of the homogeneous representation. To be able to do this without confusion, we should find a notation to differentiate between ordinary and homogeneous representations.

We'll denote an ordinary 3-D vectors with a lowercase bold variable,

$$\mathbf{v} = (x, y, z) \tag{6.6}$$

whereas its homogeneous representation is denoted with an uppercase roman variable

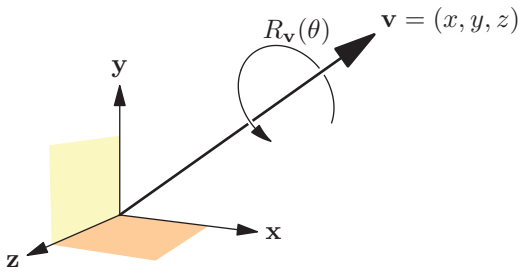
$$V = \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}. \tag{6.7}$$

We use uppercase roman letters for both homogeneous transformation matrices and homogeneous column vectors (which are 4×1 matrices). When we do, definitions and context should make it clear which is which.

How to Rotate about an Arbitrary Axis

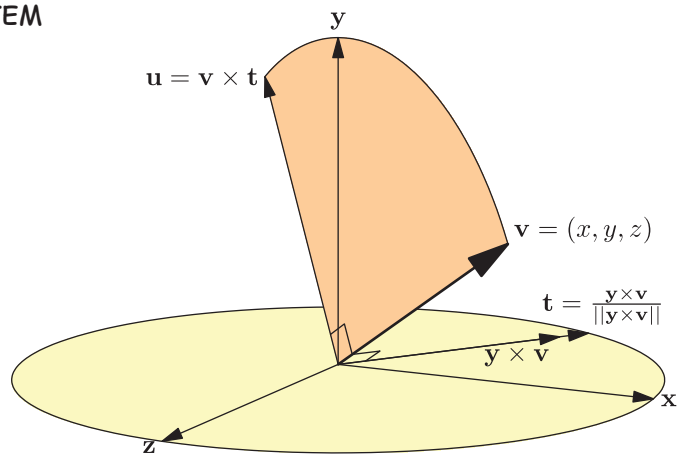
"The Algebraic Way"

Let $R_v(\theta)$ be a rotation by θ about any unit vector v



Step 1: Create a new <T,U,V> coordinate system

(See "How to Make an Orthogonal Coordinate System")



Step 2: Make a matrix out of <T,U,V>,

$$R = \begin{bmatrix} T_x & T_y & T_z & 0 \\ U_x & U_y & U_z & 0 \\ V_x & V_y & V_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

R is a rotation matrix

Since T,U,V are unit length...

$T \cdot T = U \cdot U = V \cdot V = 1$

Since <T,U,V> is orthogonal...

$T \cdot U = T \cdot V = U \cdot V = 0$

$$R\,R^T = \begin{bmatrix} T\cdot T & T\cdot U & T\cdot V & 0 \\ U\cdot T & U\cdot U & U\cdot V & 0 \\ V\cdot T & V\cdot U & V\cdot V & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I$$

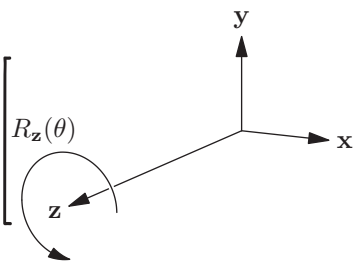
R rotates T,U,V into X,Y,Z

$$R\,T = \begin{bmatrix} T\cdot T \\ U\cdot T \\ V\cdot T \\ 0 \end{bmatrix} = X, \quad R\,U = \begin{bmatrix} T\cdot U \\ U\cdot U \\ V\cdot U \\ 0 \end{bmatrix} = Y, \quad R\,V = \begin{bmatrix} T\cdot V \\ U\cdot V \\ V\cdot V \\ 0 \end{bmatrix} = Z$$

Step 3: Compose rotations: $R_v(\theta) = R^T R_z(\theta) R$

Rotate back | Rotate by θ about Z | Rotate V to Z

$$\begin{bmatrix} T_x & U_x & V_x & 0 \\ T_y & U_y & V_y & 0 \\ T_z & U_z & V_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} T_x & T_y & T_z & 0 \\ U_x & U_y & U_z & 0 \\ V_x & V_y & V_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Dot Product

The dot product on ordinary vectors is denoted $\mathbf{a} \cdot \mathbf{b}$ and is defined algebraically as

$$\mathbf{a} \cdot \mathbf{b} = (a_x, a_y, a_z) \cdot (b_x, b_y, b_z) = a_x b_x + a_y b_y + a_z b_z. \tag{6.8}$$

If $A = [a_x, a_y, a_z, 0]^T$ and $B = [b_x, b_y, b_z, 0]$ represent $\mathbf{a}, \mathbf{b}$ as homogeneous column vectors, then the dot product can be computed as a matrix product

$$\mathbf{a} \cdot \mathbf{b} = A^T B = [a_x \ a_y \ a_z \ 0] \begin{bmatrix} b_x \\ b_y \\ b_z \\ 0 \end{bmatrix}. \tag{6.9}$$

From this definition, we see that $\mathbf{a} \cdot \mathbf{a} = ||a||^2$. It should also be clear that the dot product is commutative $\mathbf{a} \cdot \mathbf{b} = \mathbf{b} \cdot \mathbf{a}$, and distributes over vector addition with a vector $\mathbf{c}$,

$$\mathbf{a} \cdot (\mathbf{b} + \mathbf{c}) = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \cdot \mathbf{c}, \tag{6.10}$$

and scalar multiplication with a real value α,

$$\alpha(\mathbf{a} \cdot \mathbf{b}) = (\alpha \mathbf{a}) \cdot \mathbf{b} = \mathbf{a} \cdot (\alpha \mathbf{b}). \tag{6.11}$$

We can equivalently define the dot product geometrically as

$$\mathbf{a} \cdot \mathbf{b} = ||\mathbf{a}|| \ ||\mathbf{b}|| \cos\theta \tag{6.12}$$

where θ is the angle between $\mathbf{a}$ and $\mathbf{b}$. From this definition, we can see that if $\mathbf{a}$ and $\mathbf{b}$ are perpendicular, then $\mathbf{a} \cdot \mathbf{b} = 0$ and the dot product is positive when two non-orthogonal vectors point (even vaguely) in the same direction and negative when pointing in opposite directions. This definition is also useful to find the angle θ between two vectors $\mathbf{a}$ and $\mathbf{b}$ as

$$\cos\theta = \frac{\mathbf{a}}{||\mathbf{a}||} \cdot \frac{\mathbf{b}}{||\mathbf{b}||}. \tag{6.13}$$

As Fig. 6.5 shows, we can use the dot product to find the component of a vector in the direction of another vector. In this example, the vector $\mathbf{a}$ casts a shadow onto the vector $\mathbf{b}$ and this shadow is cast perpendicular to $\mathbf{b}$ creating a right triangle. The hypotenuse is $||\mathbf{a}||$, so the adjacent length is $||a|| \cos\theta$, which when combined with the geometric definition of the dot product (6.12), yields the length shown. The component of $\mathbf{a}$ in $\mathbf{b}$'s direction is commonly denoted $\mathbf{a_b}$ and is elegantly defined

$$\mathbf{a_b} = \frac{\mathbf{a} \cdot \mathbf{b}}{\mathbf{b} \cdot \mathbf{b}} \mathbf{b}. \tag{6.14}$$

Note that if $||\mathbf{b}|| = 1$, then $\mathbf{a} \cdot \mathbf{b}$ yields the portion of $\mathbf{a}$ in $\mathbf{b}$'s direction. Let $\mathbf{x} = [1\ 0\ 0\ 0]^T, \mathbf{y} = [0\ 1\ 0\ 0]^T$ and $\mathbf{z} = [0\ 0\ 1\ 0]^T$. Then $\mathbf{a} = (\mathbf{a} \cdot \mathbf{x}, \mathbf{a} \cdot \mathbf{y}, \mathbf{a} \cdot \mathbf{z})$. In fact, one can find the coordinates of $\mathbf{a}$ in any coordinate system this way, by redefining $\mathbf{x}, \mathbf{y}$ and $\mathbf{z}$.

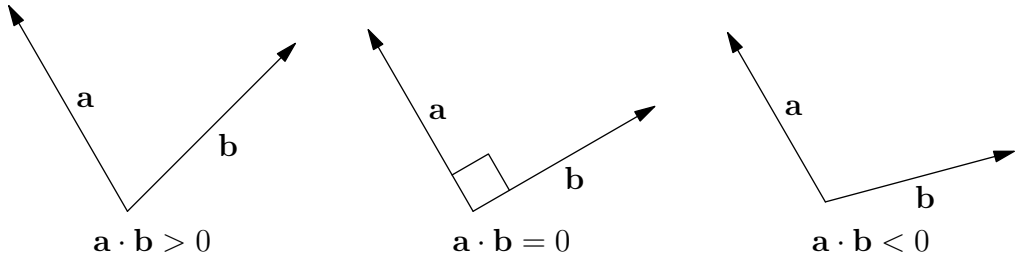


Figure 6.4: Signs of dot products.

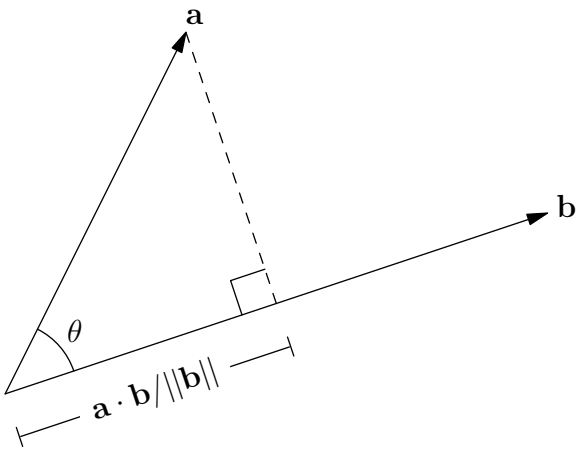


Figure 6.5: The dot product can be used to "cast a shadow" of vector **a** perpendicularly onto vector **b**. The length $\mathbf{a} \cdot \mathbf{b}/||\mathbf{b}||$ is the component of **a** in **b**'s direction.

Cross Product

The cross product is defined algebraically on ordinary 3-D vectors as

$$(a_x, a_y, a_z) \times (b_x, b_y, b_z) = (a_y b_z - a_z b_y, a_z b_x - a_x b_z, a_x b_y - a_y b_x). \quad (6.15)$$

The x coordinate is the difference of the products of the y and z components. The order can be remembered by computing the determinant

$$(a_x, a_y, a_z) \times (b_x, b_y, b_z) = \begin{vmatrix} \mathbf{x} & \mathbf{y} & \mathbf{z} \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} \quad (6.16)$$

We usually do not need to implement a cross product using the homogeneous representation, but we can if necessary. Given the vector $\mathbf{a} = (a_x, a_y, a_z)$, the skew-symmetric matrix

$$\mathbb{X}_{\mathbf{a}} = \begin{bmatrix} & -a_z & a_y \\ a_z & & -a_x \\ -a_y & a_x & \end{bmatrix} \quad (6.17)$$

can be used to generate a homogeneous 4×4 matrix that generates the cross product with $\mathbf{a}$,

$$\begin{bmatrix} & -a_z & a_y & \\ a_z & & -a_x & \\ -a_y & a_x & & \\ & & & 1 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \\ 0 \end{bmatrix} = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \\ 0 \end{bmatrix}. \tag{6.18}$$

The cross product distributes over addition using a vector c

$$\mathbf{a} \times (\mathbf{b} + \mathbf{c}) = \mathbf{a} \times \mathbf{b} + \mathbf{a} \times \mathbf{c}, \tag{6.19}$$

and over scalar multiplication with real value α

$$\alpha(\mathbf{a} \times \mathbf{b}) = (\alpha\mathbf{a}) \times \mathbf{b} = \mathbf{a} \times (\alpha\mathbf{b}). \tag{6.20}$$

The cross product is not commutative, but it is in fact *anti*commutative,

$$\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}. \tag{6.21}$$

When the cross product generates a vector of positive length, the vector is perpendicular to the operand vectors,

$$(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{a} = (\mathbf{a} \times \mathbf{b}) \cdot \mathbf{b} = 0. \tag{6.22}$$

and will follow the right hand rule in the order of the operands

$$\mathbf{x} \times \mathbf{y} = \mathbf{z}, \tag{6.23}$$

$$\mathbf{y} \times \mathbf{z} = \mathbf{x}, \tag{6.24}$$

$$\mathbf{z} \times \mathbf{x} = \mathbf{y}. \tag{6.25}$$

This is the most useful role for the cross product, to find a vector perpendicular to the plane spanned by two other vectors.

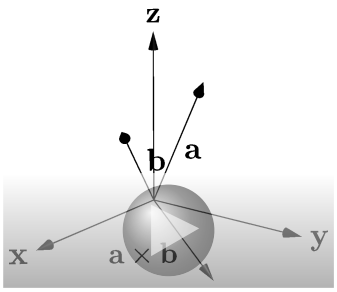


Figure 6.6: The cross product of two vectors generates a new vector perpendicular to them both.

The length of this vector is given by

$$||\mathbf{a} \times \mathbf{b}|| = ||a||\ ||b|| \sin\theta. \tag{6.26}$$

Hence, when $\mathbf{a}$ and $\mathbf{b}$ are parallel $\theta = 0°$ or antiparallel $\theta = 180°$, the cross product yields the zero vector $[0\ 0\ 0\ 0]^T$. If $\mathbf{a}$ and $\mathbf{b}$ are unit length and perpendicular, then $\mathbf{a} \times \mathbf{b}$ will be unit length and perpendicular to both $\mathbf{a}$ and $\mathbf{b}$.

Gramm-Schmidt Orthonormalization

We can use the cross product to set up a new coordinate system, through a method called Gramm-Schmidt orthonormalization. Given a unit vector $\mathbf{v}$ in any direction, we will use the cross product to find two new unit vectors $\mathbf{t}$ and $\mathbf{u}$ such that all three are mutually perpendicular and thus form a right-handed "orthonormal" coordinate system $\langle \mathbf{t}, \mathbf{u}, \mathbf{v} \rangle$. Orthonormal means

$$\mathbf{t} \cdot \mathbf{u} = \mathbf{u} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbf{t} = 0 \tag{6.27}$$

$$||\mathbf{t}|| = ||\mathbf{u}|| = ||\mathbf{v}|| = 1, \tag{6.28}$$

and right-handed means

$$\mathbf{t} \times \mathbf{u} = \mathbf{v}, \tag{6.29}$$

$$\mathbf{u} \times \mathbf{v} = \mathbf{t}, \tag{6.30}$$

$$\mathbf{v} \times \mathbf{t} = \mathbf{u}. \tag{6.31}$$

If $\mathbf{v}$ is not unit length, we first unitize it. Then we let

$$\mathbf{t} = \frac{\mathbf{y} \times \mathbf{v}}{||\mathbf{y} \times \mathbf{v}||} \tag{6.32}$$

where $\mathbf{y} = (0, 1, 0)$. This will find a new direction for $\mathbf{t}$ perpendicular to the plane containing $\mathbf{v}$ and the $\mathbf{y}$ axis. The vectors $\mathbf{v}$ and $\mathbf{y}$ might not be perpendicular, so even though they are both unit length, their cross product might not be. (Recall the magnitude of this cross product will equal the sine of the angle between $\mathbf{y}$ and $\mathbf{v}$.) Hence, we divide the cross product by its magnitude to ensure the result is unit length

Since we know $\mathbf{v} \neq \pm\mathbf{y}$, then $\mathbf{t}$ is a unit vector (somewhere in the xz-plane). All that remains is to compute

$$\mathbf{u} = \mathbf{v} \times \mathbf{t}. \tag{6.33}$$

(Since $\mathbf{v}$ and $\mathbf{t}$ are unit length and perpendicular, their cross product will be unit length so we need not unitize the result.)

There are two special cases we have to worry about. If $\mathbf{v} = \mathbf{y}$ then $\mathbf{y} \times \mathbf{v} = (0, 0, 0)$. In that case it is easy to find an appropriate orthonormal right-handed coordinate system

$$\langle \mathbf{t}, \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{z}, \mathbf{x}, \mathbf{y} \rangle. \tag{6.34}$$

The second case occurs when $\mathbf{v} = -\mathbf{y}$, which similarly yields a null cross product. In this case we assign the coordinate system

$$\langle \mathbf{t}, \mathbf{u}, \mathbf{v} \rangle = \langle -\mathbf{z}, -\mathbf{x}, -\mathbf{y} \rangle. \tag{6.35}$$

(These are arbitrary choices. We can pick any righthanded coordinate system where $\mathbf{v} = \pm\mathbf{y}$ appropriately and $\mathbf{t}, \mathbf{u}$ lie perpendicular to each other in the xz-plane.)

6.4 Rotation about a Coordinate Axis

In 2-D, rotation is simple. A positive rotation in 2-D sends a point counterclockwise around the origin. In 3-D, rotations are more complicated, because one needs both the angle of rotation as well as an "axis" of rotation. The axis of rotation can be one of the three coordinate axes ($\mathbf{x}, \mathbf{y}$ or $\mathbf{z}$), or it could be an axis extending in any other 3-D direction.

The transformation matrix for a rotation about the z axis looks like our original 2-D transformation, since it rotates the x and y coordinates, but leaves the z coordinate alone

$$R(\theta, \mathbf{z}) = \begin{bmatrix} \cos\theta & -\sin\theta & & \\ \sin\theta & \cos\theta & & \\ & & 1 & \\ & & & 1 \end{bmatrix}. \tag{6.36}$$

Note that a positive angle θ rotates counterclockwise when viewed from the $+\mathbf{z}$ direction. As in the 2-D case, the orientation (counterclockwise or clockwise) of the rotation is controlled by the position of the negative sign in (6.36). In row 1, column 2, it yields counterclockwise rotation, whereas in row 2, column 1, it yields a clockwise rotation. The counterclockwise rotation is preferred as a 90° rotation about the $\mathbf{z}$ axis rotates a point on the $\mathbf{x}$ axis into a point on the $\mathbf{y}$ axis

$$\begin{bmatrix} \cos 90^\circ & -\sin 90^\circ & & \\ \sin 90^\circ & \cos 90^\circ & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}. \tag{6.37}$$

We like rotations to occur in alphabetical order.

Hence a 90° rotation about the $\mathbf{x}$ axis should rotate a point on the $\mathbf{y}$ axis to a point on the $\mathbf{z}$ axis. The transformation matrix that implements a rotation by θ about the $\mathbf{x}$ axis is

$$R(\theta, \mathbf{x}) = \begin{bmatrix} 1 & & & \\ & \cos\theta & -\sin\theta & \\ & \sin\theta & \cos\theta & \\ & & & 1 \end{bmatrix}. \tag{6.38}$$

You can verify that $R(90^\circ, \mathbf{x})[0, 1, 0, 1]^T = [0, 0, 1, 1]^T$.

The transformation matrix that implements a rotation by θ about the y axis might look a bit unusual

$$R(\theta, \mathbf{y}) = \begin{bmatrix} \cos\theta & & \sin\theta & \\ & 1 & & \\ -\sin\theta & & \cos\theta & \\ & & & 1 \end{bmatrix}, \tag{6.39}$$

but you can verify that $R(90°, \mathbf{y})[0, 0, 1, 1]^T = [1, 0, 0, 1]^T$. In this case, we have rotated a point on the $\mathbf{z}$ axis back onto a point onto the $\mathbf{x}$ axis (since nothing comes after $\mathbf{z}$ in alphabetical order).

6.5 Properties of Rotation Matrices

In three dimensions, rotations do not commute, which means the order of rotations is important. In 2-D rotations do commute, so a rotation by θ followed by a rotation by ϕ is the same as a rotation by ϕ followed by a rotation by θ, namely a rotation by $\theta+\phi$. In 3-D, rotations about the same axis commute, e.g.

$$R(\phi, \mathbf{z})R(\theta, \mathbf{z}) = R(\theta, \mathbf{z})R(\phi, \mathbf{z}) = R(\theta + \phi, \mathbf{z}). \tag{6.40}$$

However, rotations about different axes do not commute, e.g.

$$R(\phi, \mathbf{y})R(\theta, \mathbf{x}) \neq R(\theta, \mathbf{x})R(\phi, \mathbf{y}). \tag{6.41}$$

Let R be a 4×4 rotation matrix and let Q be its upper-left 3×3 submatrix, the part that actually performs the 3-D rotation,

$$R = \begin{bmatrix} \boxed{Q} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{bmatrix} \tag{6.42}$$

In linear algebra, the matrix Q is a *special orthogonal* matrix, which means that its columns are orthogonal right-handed unit (column) vectors and its rows are also orthogonal right-handed unit (row) vectors. If we focus on the upper-left submatrix (the part of the homogeneous rotation matrix that performs the rotation), we can decompose it into

$$Q = \begin{bmatrix} \boxed{\mathbf{u}_1} \\ \boxed{\mathbf{u}_2} \\ \boxed{\mathbf{u}_3} \end{bmatrix} = \begin{bmatrix} \boxed{\mathbf{v}_1} & \boxed{\mathbf{v}_2} & \boxed{\mathbf{v}_3} \end{bmatrix} \tag{6.43}$$

where

$$||\mathbf{u}_1|| = ||\mathbf{u}_2|| = ||\mathbf{u}_3|| = 1, \tag{6.44}$$

$$\mathbf{u}_1 \cdot \mathbf{u}_2 = \mathbf{u}_2 \cdot \mathbf{u}_3 = \mathbf{u}_3 \cdot \mathbf{u}_1 = 0, \tag{6.45}$$

$$\mathbf{u}_1 \times \mathbf{u}_2 = \mathbf{u}_3, \quad \mathbf{u}_2 \times \mathbf{u}_3 = \mathbf{u}_1, \quad \mathbf{u}_3 \times \mathbf{u}_1 = \mathbf{u}_2, \tag{6.46}$$

and likewise for $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$.

A convenient property of these matrices is that the inverse of a special orthogonal matrix is simply its transpose. Recall that for any invertible

matrix M we have $M^{-1}M = I$. For our 3×3 special orthogonal matrix Q, we have that $Q^TQ = I$, because

$$Q^TQ = \begin{bmatrix} \boxed{\mathbf{v}_1} \\ \boxed{\mathbf{v}_2} \\ \boxed{\mathbf{v}_3} \end{bmatrix} \begin{bmatrix} \boxed{\mathbf{v}_1} & \boxed{\mathbf{v}_2} & \boxed{\mathbf{v}_3} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{v}_1 \cdot \mathbf{v}_1 & \mathbf{v}_1 \cdot \mathbf{v}_2 & \mathbf{v}_1 \cdot \mathbf{v}_3 \\ \mathbf{v}_2 \cdot \mathbf{v}_1 & \mathbf{v}_2 \cdot \mathbf{v}_2 & \mathbf{v}_2 \cdot \mathbf{v}_3 \\ \mathbf{v}_3 \cdot \mathbf{v}_1 & \mathbf{v}_3 \cdot \mathbf{v}_2 & \mathbf{v}_3 \cdot \mathbf{v}_3 \end{bmatrix} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}, \quad (6.47)$$

since its rows are mutually perpendicular. Similarly $QQ^T = I$ because Q consists of three mutually perpendicular column vectors.

Note also that all of these properties work not only with Q but also extend to the original homogeneous 4×4 rotation R that contained it, since the fourth column vector $[0, 0, 0, 1]^T$ is orthogonal to the first three column vectors of R, and $\det R = 1$ since $\det Q = 1$. However, it is not very helpful to think of homogenous 4×4 rotation matrices as four-dimensional rotations, but rather as a three-dimensional rotation implemented in a 4×4 homogeneous transformation matrix. Nevertheless R, like Q, is special orthogonal, and the inverse of any rotation matrix is its transpose.

Cayley's Formula

Recall from the discussion of cross products that given a vector $\mathbf{v} = (v_x, v_y, v_z)$, we can create a (skew-symmetric) matrix

$$\mathbb{X}_{\mathbf{v}} = \begin{bmatrix} & -v_z & v_y \\ v_z & & -v_x \\ -v_y & v_x & \end{bmatrix} \quad (6.48)$$

whose product with a column vector yeilds the cross product of $\mathbf{v}$ with that column vector.

Cayley's formula uses these cross product matrices for $\mathbf{v}$ to create a 3×3 (non-homogeneous) rotation matrix about an arbitrary axis in the direction of $\mathbf{v}$

$$Q(\theta, \mathbf{v}) = (I + \mathbb{X}_{\mathbf{v}})\ (I - \mathbb{X}_{\mathbf{v}})^{-1} \quad (6.49)$$

by some angle θ. The vector $\mathbf{v}$ need not be unit length, and in fact its length indicates the angle of rotation,

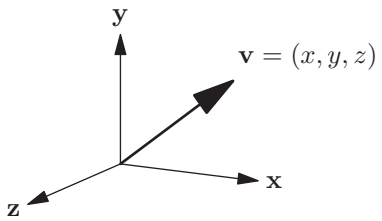
$$\theta = \arccos\left(\frac{1 - \mathbf{v} \cdot \mathbf{v}}{1 + \mathbf{v} \cdot \mathbf{v}}\right). \quad (6.50)$$

A unit vector creates a 90° rotation, a zero vector creates an identity matrix, and a 180° rotation would require an infinite length vector.

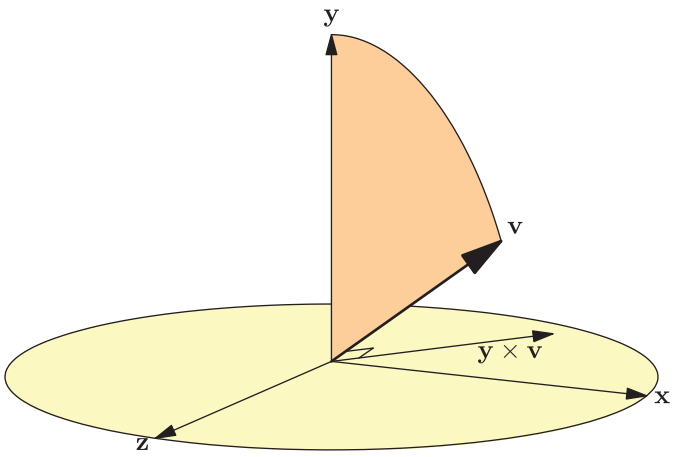
How to Make An Orthogonal Coordinate System

"Gramm Schmidt Orthonormalization"

Given a unit vector V, create an orthogonal coordinate system <T,U,V>



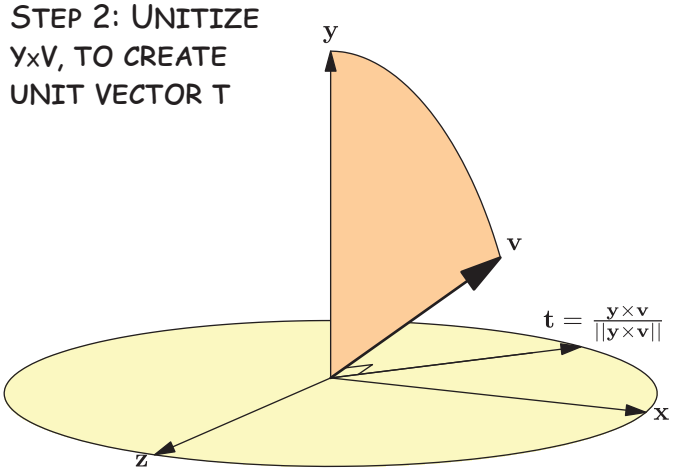
Step 1: Compute YxV, which will be perp. to V



(but not necessarily unit length)

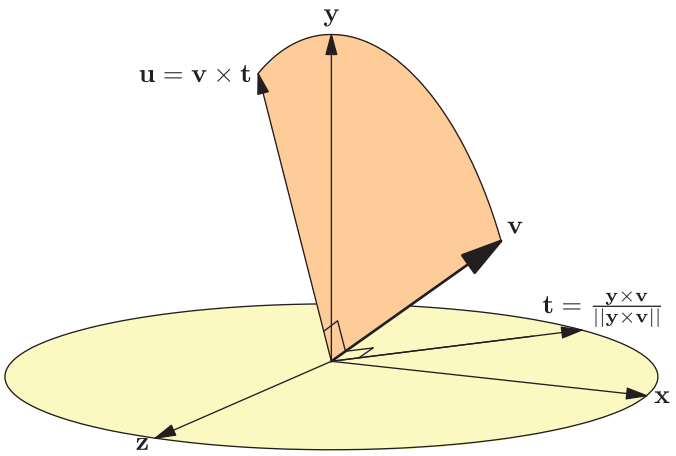
If V is parallel to Y, then VxT = (0,0,0) so
IF V = Y, THEN SET T = X AND U = -Z
IF V = -Y, THEN SET T = X AND U = Z
THEN GO TO STEP 4, YOU ARE DONE.

Step 2: Unitize YxV, to create unit vector T



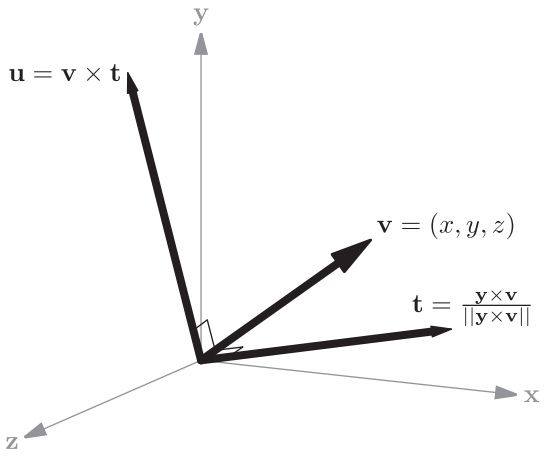
(Note: YxV and T will be in the XZ plane because they are also perp. to Y.)

Step 3: Compute U = VxT, perp. to V and T



(No need to unitize VxT since V and T are perpendicular)

Step 4: Congratulations! <T,U,V> is an orthogonal coordinate system



We can similarly find the rotation axis of a given 3×3 rotation matrix Q,

$$\mathbb{X}_{\mathbf{v}} = (I + Q)(I - Q)^{-1} \tag{6.51}$$

where the rotation axis $\mathbf{v}$ is embedded in $\mathbb{X}_{\mathbf{v}}$ as shown in (6.48). Just hope Q doesn't represent a 180° rotation.

6.6 Rotation about an Arbitrary Axis

In 2-D, there is just one way to rotate: about the origin. In 3-D, we can rotate about each of the three axes $\mathbf{x}, \mathbf{y}$ and $\mathbf{z}$, but there are even more rotations. In fact, we can rotate about an "axis" extending in any direction through the origin. We usually specify this axis with a unit vector $\mathbf{v}$.

We can specify an arbitrary-axis rotation as $R(\theta, \mathbf{v})$, which describes a homogeneous rotation matrix that will rotate a homogeneous point or vector around $\mathbf{v}$ by an angle of θ, counterclockwise when observed from a point further along the direction $\mathbf{v}$ is pointing. We can implement such a rotation as a composition of several rotations

$$R(\theta, \mathbf{v}) = R^{-1}(\phi) R(\theta, \mathbf{z}) R(\phi) \tag{6.52}$$

where $R(\phi)$ is some rotation that rotates the unit vector $\mathbf{v}$ into the z-axis. Thus $R(\theta, \mathbf{v})$ rotates an object about the $\mathbf{v}$ axis by first rotating the object onto the z-axis, then rotates the object around the z-axis, then rotates the result back to the $\mathbf{v}$ axis. The trick is finding $R(\phi)$, the transformation that rotates the $\mathbf{v}$ axis to the $\mathbf{z}$ axis. We have two ways. a geometric way that is tedious but easy to follow, and an algebraic way that is simpler but conceptually more difficult.

The Geometric Way

Given a unit vector $\mathbf{v}$, we want to find $R(\phi)$, a rotation matrix such that $R(\phi)\begin{bmatrix}\mathbf{v}\\0\end{bmatrix} = \begin{bmatrix}\mathbf{z}\\0\end{bmatrix}$. We will do this by first rotating $\mathbf{v}$ into the xz-plane, then rotating that into the z-axis.

We first rotate $\mathbf{v}$ into the xz-plane, by calculating an appropriate rotation about the x-axis, $R(\phi_x, \mathbf{x})$. We find this angle ϕ_x by projecting $\mathbf{v} = (x, y, z)$ onto the yz-plane as $(0, y, z)$. Note that the same rotation $R(\phi_x, \mathbf{x})$ that rotates $\mathbf{v}$ into the xz-plane would also rotate $(0, y, z)$ into the z axis. In the yz-plane, we form the right triangle with vertices $(0, 0, 0), (0, y, z)$ and $(0, 0, z)$, and the angle its hypotenuse makes with the origin is precisely ϕ_x. The length of the adjacent edge is z, of the opposite edge is y and of the hypotenuse is $d = \sqrt{y^2 + z^2}$. We find ϕ_x with the trigonometry

$$\cos\phi_x = z/d, \tag{6.53}$$

$$\sin\phi_x = y/d, \tag{6.54}$$

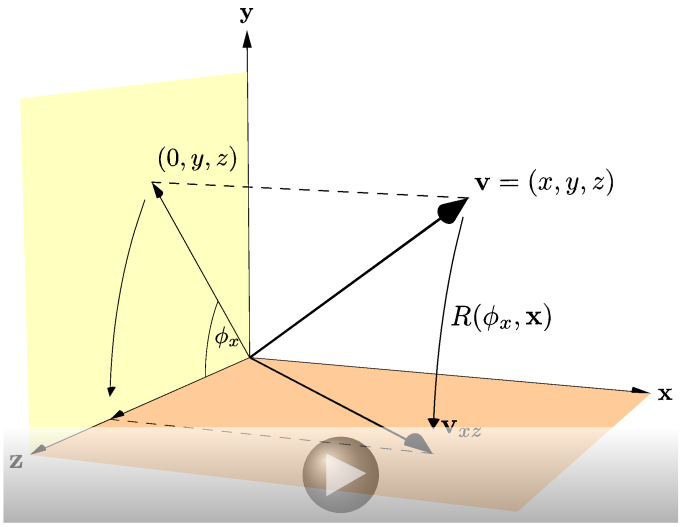
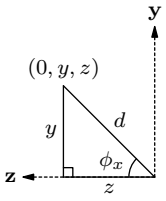


Figure 6.7: Rotation that takes $\mathbf{v}$ to the xz-plane also rotates its projection $\mathbf{v}_{yz}$ on the yz plane to the z-axis.

yielding the homogeneous rotation matrix

$$R(\phi_x, \mathbf{x}) = \begin{bmatrix} 1 & & & \\ & z/d & -y/d & \\ & y/d & z/d & \\ & & & 1 \end{bmatrix}. \qquad (6.55)$$



We will denote the result of this rotation $\begin{bmatrix} \mathbf{v}_{xz} \\ 0 \end{bmatrix} = R(\phi_x, \mathbf{x}) \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix}$.

Figure 6.8: Rotating an arbitrary vector $\mathbf{v}$ into the z-axis.

Next, we find a rotation about the y-axis $R(\phi_y, \mathbf{y})$ that rotates $\mathbf{v}_{xz}$ from the xz-plane into the z-axis. Note that $\mathbf{v}_{xz}$ is a unit vector because it is a

rotated version of $\mathbf{v}$, so when it is rotated into the z-axis, it will be simply $(0,0,1)$. We again create a right triangle, now in the xz-plane with vertices $(0,0,0)$, $\mathbf{v}_{xz}$ and $(0,0,d)$. The point $(0,0,d)$ is the projection of $\mathbf{v}_{xz}$ onto the z-axis, and is the result of rotating the previous hypotenuse by ϕ_x about the x-axis. We use trigonometry again to find the angle ϕ_y of this triangle at the origin, given a hypotenuse of length one ($||\mathbf{v}_{xy}|| = ||\mathbf{v}||$), an adjacent length of d and an opposite length x (since rotation about the x-axis does not change the x coordinate). A positive right-handed rotation would rotate the positive z axis toward the positive x axis, so the angle ϕ_y should be negative when used to specify a rotation[1]. Hence

$$\cos\phi_y = d, \tag{6.56}$$

$$\sin\phi_y = -x, \tag{6.57}$$

and

$$R(\phi_y, \mathbf{y}) = \begin{bmatrix} d & & -x & \\ & 1 & & \\ x & & d & \\ & & & 1 \end{bmatrix}. \tag{6.58}$$

The Algebraic Way

As before, given a unit vector $\mathbf{v}$, we want to find a rotation matrix $R(\phi)$ such that $R(\phi)\begin{bmatrix}\mathbf{v}\\0\end{bmatrix} = \begin{bmatrix}\mathbf{z}\\0\end{bmatrix}$ that rotates $\mathbf{v}$ to the z-axis, so a rotation about $\mathbf{v}$ can then be implemented as a rotation about the z-axis. The algebraic approach to this problem first uses Gramm-Schmidt orthonormalization to create a right-handed coordinate system with $\mathbf{v}$ as one of its axes, then uses the special orthogonal property of rotation matrices to find a rotation matrix that maps this right-handed coordinate system to the x, y and z axes.

Given unit vector $\mathbf{v}$, the first step uses Gramm-Shmidt orthonormalization (Section **??**) to create a coordinate system $\langle \mathbf{t}, \mathbf{u}, \mathbf{v}\rangle$ as

$$\mathbf{t} = \frac{\mathbf{y}\times\mathbf{v}}{||\mathbf{y}\times\mathbf{v}||}, \quad \mathbf{u} = \mathbf{v}\times\mathbf{t}, \tag{6.59}$$

unless $\mathbf{v} = \pm\mathbf{y}$ in which case we set $\langle \mathbf{t}, \mathbf{u}, \mathbf{v}\rangle = \langle \pm\mathbf{x}, \pm\mathbf{y}, \mathbf{z}\rangle$.

Recall from Section 6.5 that rotation matrices are special orthogonal (including 4×4 homogeneous rotation matrices in addition to their upper-left 3×3 submatrices). Given a right-handed coordinate system $\langle \mathbf{t}, \mathbf{u}, \mathbf{v}\rangle$ of orthogonal unit vectors, we can form a homogeneous rotation matrix out of them

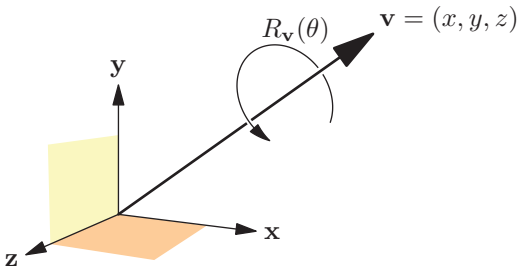
$$R = \begin{bmatrix} \boxed{\mathbf{t}} & \boxed{\mathbf{u}} & \boxed{\mathbf{v}} & \begin{matrix}0\\0\\0\end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{6.60}$$

[1]Thanks to University of Wisconsin student Qisi Wang for pointing that out.

How to Rotate about an Arbitrary Axis

"The Geometric Way"

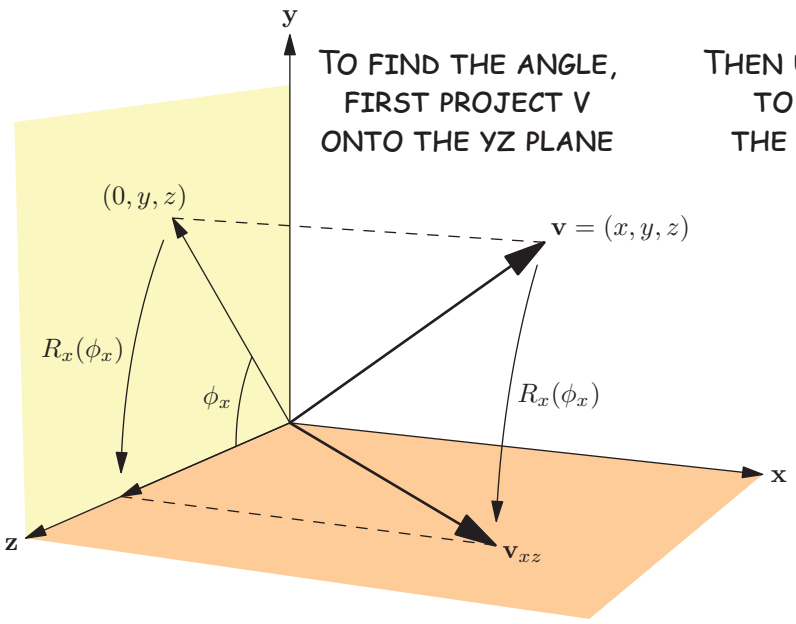
Let unit vector v indicate the direction of the axis

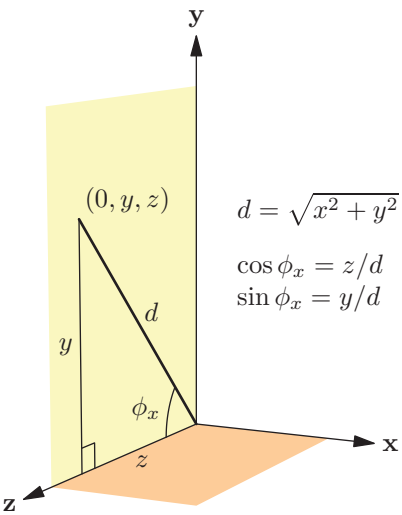


Step 1: Rotate v about x into the xz plane

To find the angle, first project v onto the yz plane

Then use trig to find the angle

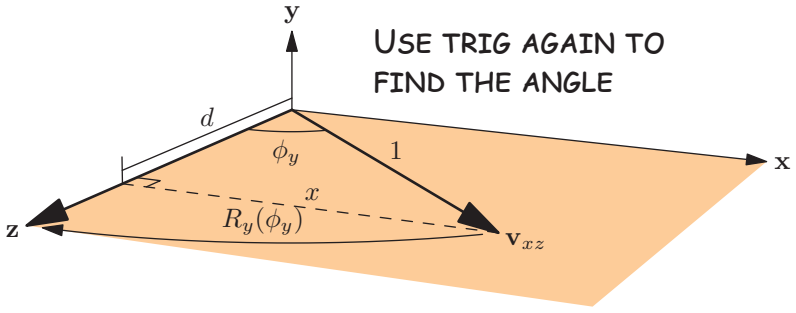




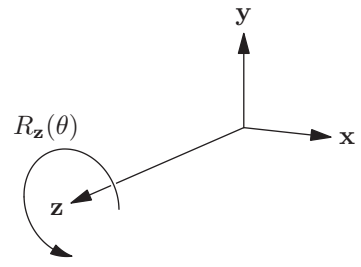
$$R_x(\phi_x) = \begin{bmatrix} 1 & & \\ & z/d & -y/d \\ & y/d & z/d \\ & & & 1 \end{bmatrix}$$

Step 2: Rotate vxz about y into the z axis,

Use trig again to find the angle



Step 3: Rotate by the original angle about the z axis



Step 4: Rotate about y by opposite of Step 2.
Step 5: Rotate About x by opposite of Step 1.

Final rotation is product of rotations in steps 1 thru 5.

Since this rotation is special orthogonal (note all 4-element column vectors are indeed mutually orthogonal and unit length), it represents some rotation by some angle about some axis. We need not be concerned with the rotation angle and axis, because we can look instead at the behavior of the rotation.

The behavior of the rotation R is to rotate the three coordinate axes $\mathbf{x}, \mathbf{y}, \mathbf{z}$ onto the three vector directions $\mathbf{t}, \mathbf{u}, \mathbf{v}$,

$$\begin{bmatrix} \boxed{\mathbf{t}} & \boxed{\mathbf{u}} & \boxed{\mathbf{v}} & \begin{matrix}0\\0\\0\end{matrix} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \boxed{\mathbf{t}} \\ 0 \end{bmatrix}, \tag{6.61}$$

and likewise $R \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{u} \\ 0 \end{bmatrix}$ and $R \begin{bmatrix} \mathbf{z} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix}$.

Since R is a rotation matrix, its inverse is its transpose,

$$R^T = \begin{bmatrix} \boxed{\mathbf{t}} & 0 \\ \boxed{\mathbf{u}} & 0 \\ \boxed{\mathbf{v}} & 0 \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}. \tag{6.62}$$

This matrix rotates the vectors $\mathbf{t}, \mathbf{u}$ and $\mathbf{v}$ onto the $\mathbf{x}, \mathbf{y}$ and $\mathbf{z}$ coordinate axis directions. For example $R^T \begin{bmatrix} \mathbf{t} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix}$,

$$\begin{bmatrix} \boxed{\mathbf{t}} & 0 \\ \boxed{\mathbf{u}} & 0 \\ \boxed{\mathbf{v}} & 0 \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \begin{bmatrix} \boxed{\mathbf{t}} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{t} \cdot \mathbf{t} \\ \mathbf{u} \cdot \mathbf{t} \\ \mathbf{v} \cdot \mathbf{t} \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{6.63}$$

and furthermore $R^T \begin{bmatrix} \mathbf{u} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix}$ and $R^T \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{z} \\ 0 \end{bmatrix}$.

Hence we implement $R(\theta, \mathbf{v})$ using R^T to rotate $\mathbf{v}$ to $\mathbf{z}$, $R(\theta, \mathbf{z})$ to rotate about $\mathbf{z}$, and R to rotate the result back from $\mathbf{z}$ to $\mathbf{v}$,

$$R(\theta, \mathbf{v}) = R \; R(\theta, \mathbf{z}) \; R^T. \tag{6.64}$$