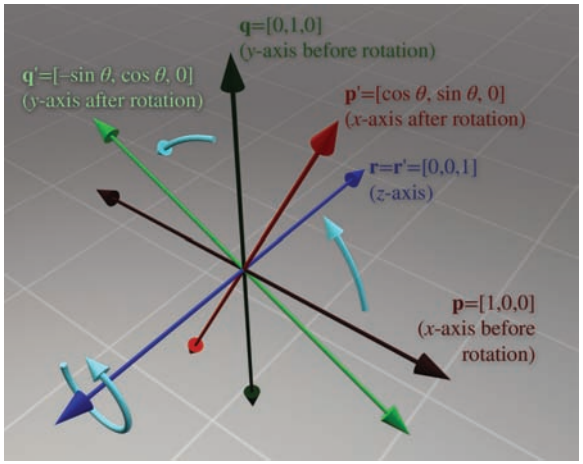

Figure 5.4
Rotating about the z-axis in 3D

And finally, rotation about the z-axis (see Figure 5.4) is done with the matrix

$$\mathbf{R}_z(\theta) = \begin{bmatrix} -\mathbf{p}'- \\ -\mathbf{q}'- \\ -\mathbf{r}'- \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.5}$$

3D matrix to rotate about the z-axis

Please note that although the figures in this section use a left-handed convention, the matrices work in either left- or right-handed coordinate systems, due to the conventions used to define the direction of positive rotation. You can verify this visually by looking at the figures in a mirror.

5.1.3 3D Rotation about an Arbitrary Axis

We can also rotate about an arbitrary axis in 3D, provided, of course, that the axis passes through the origin, since we are not considering translation at the moment. This is more complicated and less common than rotating about a cardinal axis. As before, we define θ to be the amount of rotation about the axis. The axis will be defined by a unit vector $\hat{\mathbf{n}}$.

Let's derive a matrix to rotate about $\hat{\mathbf{n}}$ by the angle θ. In other words, we wish to derive the matrix $\mathbf{R}(\hat{\mathbf{n}}, \theta)$ such that when we multiply a vector $\mathbf{v}$ by $\mathbf{R}(\hat{\mathbf{n}}, \theta)$, the resulting vector $\mathbf{v}'$ is the result of rotating $\mathbf{v}$ about $\hat{\mathbf{n}}$ by the angle θ:

$$\mathbf{v}' = \mathbf{v}\,\mathbf{R}(\hat{\mathbf{n}}, \theta).$$

To derive the matrix $\mathbf{R}(\hat{\mathbf{n}}, \theta)$, let's first see if we can express $\mathbf{v}'$ in terms of $\mathbf{v}$, $\hat{\mathbf{n}}$, and θ. The basic idea is to solve the problem in the plane perpen-

dicular to $\hat{\mathbf{n}}$, which is a much simpler 2D problem. To do this, we separate $\mathbf{v}$ into two vectors, $\mathbf{v}_{\parallel}$ and $\mathbf{v}_{\perp}$, which are parallel and perpendicular to $\hat{\mathbf{n}}$, respectively, such that $\mathbf{v} = \mathbf{v}_{\parallel} + \mathbf{v}_{\perp}$. (We learned how to do this with the dot product in Section 2.11.2.) By rotating each of these components individually, we can rotate the vector as a whole. In other words, $\mathbf{v}' = \mathbf{v}'_{\parallel} + \mathbf{v}'_{\perp}$. Since $\mathbf{v}_{\parallel}$ is parallel to $\hat{\mathbf{n}}$, it will not be affected by the rotation about $\hat{\mathbf{n}}$. In other words, $\mathbf{v}'_{\parallel} = \mathbf{v}_{\parallel}$. So all we need to do is compute $\mathbf{v}'_{\perp}$, and then we have $\mathbf{v}' = \mathbf{v}_{\parallel} + \mathbf{v}'_{\perp}$. To compute $\mathbf{v}'_{\perp}$, we construct the vectors $\mathbf{v}_{\parallel}$, $\mathbf{v}_{\perp}$, and an intermediate vector $\mathbf{w}$, as follows:

- The vector $\mathbf{v}_{\parallel}$ is the portion of $\mathbf{v}$ that is parallel to $\hat{\mathbf{n}}$. Another way of saying this is that $\mathbf{v}_{\parallel}$ is the value of $\mathbf{v}$ *projected onto* $\hat{\mathbf{n}}$. From Section 2.11.2, we know that $\mathbf{v}_{\parallel} = (\mathbf{v} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}$.

- The vector $\mathbf{v}_{\perp}$ is the portion of $\mathbf{v}$ that is perpendicular to $\hat{\mathbf{n}}$. Since $\mathbf{v} = \mathbf{v}_{\parallel} + \mathbf{v}_{\perp}$, $\mathbf{v}_{\perp}$ can be computed by $\mathbf{v} - \mathbf{v}_{\parallel}$. $\mathbf{v}_{\perp}$ is the result of projecting $\mathbf{v}$ onto the plane perpendicular to $\hat{\mathbf{n}}$.

- The vector $\mathbf{w}$ is mutually perpendicular to $\mathbf{v}_{\parallel}$ and $\mathbf{v}_{\perp}$ and has the same length as $\mathbf{v}_{\perp}$. It can be constructed by rotating $\mathbf{v}_{\perp}$ 90° about $\hat{\mathbf{n}}$; thus we see that its value is easily computed by $\mathbf{w} = \hat{\mathbf{n}} \times \mathbf{v}_{\perp}$.

These vectors are shown in Figure 5.5.

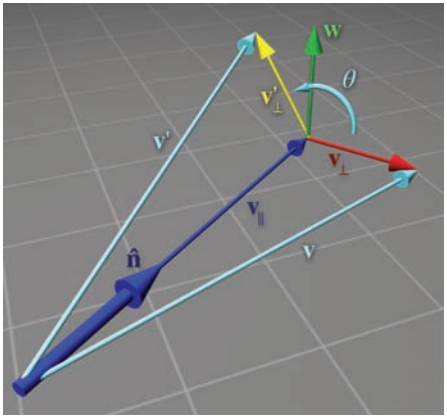

Figure 5.5
Rotating a vector about an arbitrary axis

How do these vectors help us compute $\mathbf{v}'_{\perp}$? Notice that $\mathbf{w}$ and $\mathbf{v}_{\perp}$ form a 2D coordinate space, with $\mathbf{v}_{\perp}$ as the "x-axis" and $\mathbf{w}$ as the "y-axis." (Note that the two vectors don't necessarily have unit length.) $\mathbf{v}'_{\perp}$ is the result of rotating $\mathbf{v}'$ in this plane by the angle θ. Note that this is almost identical to rotating an angle into standard position. Section 1.4.4 showed that the endpoints of a unit ray rotated by an angle θ are $\cos\theta$ and $\sin\theta$. The only difference here is that our ray is not a unit ray, and we are using $\mathbf{v}_{\perp}$ and $\mathbf{w}$ as our basis vectors. Thus, $\mathbf{v}'_{\perp}$ can be computed as

$$\mathbf{v}'_{\perp} = \cos\theta\mathbf{v}_{\perp} + \sin\theta\mathbf{w}.$$

Let's summarize the vectors we have computed:

$$\begin{aligned}
\mathbf{v}_{\parallel} &= (\mathbf{v}\cdot\hat{\mathbf{n}})\,\hat{\mathbf{n}},\\
\mathbf{v}_{\perp} &= \mathbf{v}-\mathbf{v}_{\parallel}\\
&= \mathbf{v}-(\mathbf{v}\cdot\hat{\mathbf{n}})\,\hat{\mathbf{n}},\\
\mathbf{w} &= \hat{\mathbf{n}}\times\mathbf{v}_{\perp}\\
&= \hat{\mathbf{n}}\times\left(\mathbf{v}-\mathbf{v}_{\parallel}\right)\\
&= \hat{\mathbf{n}}\times\mathbf{v}-\hat{\mathbf{n}}\times\mathbf{v}_{\parallel}\\
&= \hat{\mathbf{n}}\times\mathbf{v}-\mathbf{0}\\
&= \hat{\mathbf{n}}\times\mathbf{v},\\
\mathbf{v}'_{\perp} &= \cos\theta\mathbf{v}_{\perp}+\sin\theta\mathbf{w}\\
&= \cos\theta\left(\mathbf{v}-(\mathbf{v}\cdot\hat{\mathbf{n}})\,,\hat{\mathbf{n}}\right)+\sin\theta\left(\hat{\mathbf{n}}\times\mathbf{v}\right).
\end{aligned}$$

Substituting for $\mathbf{v}'$, we have

$$\begin{aligned}
\mathbf{v}' &= \mathbf{v}'_{\perp}+\mathbf{v}_{\parallel}\\
&= \cos\theta\left(\mathbf{v}-(\mathbf{v}\cdot\hat{\mathbf{n}})\,\hat{\mathbf{n}}\right)+\sin\theta\left(\hat{\mathbf{n}}\times\mathbf{v}\right)+(\mathbf{v}\cdot\hat{\mathbf{n}})\,\hat{\mathbf{n}}.
\end{aligned}\tag{5.1}$$

Equation (5.1) allows us to rotate any arbitrary vector about any arbitrary axis. We could perform arbitrary rotation transformations armed only with this equation, so in a sense we are done—the remaining arithmetic is essentially a notational change that expresses Equation (5.1) as a matrix multiplication.

Now that we have expressed $\mathbf{v}'$ in terms of $\mathbf{v}$, $\hat{\mathbf{n}}$, and θ, we can compute what the basis vectors are after transformation and construct our matrix. We're just presenting the results here; a reader interested in following each step can check out Exercise 2.24:

$$\mathbf{p}=\begin{bmatrix}1 & 0 & 0\end{bmatrix},\qquad \mathbf{p}'=\begin{bmatrix}{n_x}^2(1-\cos\theta)+\cos\theta\\ n_xn_y(1-\cos\theta)+n_z\sin\theta\\ n_xn_z(1-\cos\theta)-n_y\sin\theta\end{bmatrix}^{\mathrm{T}},$$

$$\mathbf{q}=\begin{bmatrix}0 & 1 & 0\end{bmatrix},\qquad \mathbf{q}'=\begin{bmatrix}n_xn_y(1-\cos\theta)-n_z\sin\theta\\ {n_y}^2(1-\cos\theta)+\cos\theta\\ n_yn_z(1-\cos\theta)+n_x\sin\theta\end{bmatrix}^{\mathrm{T}},$$

$$\mathbf{r}=\begin{bmatrix}0 & 0 & 1\end{bmatrix},\qquad \mathbf{r}'=\begin{bmatrix}n_xn_z(1-\cos\theta)+n_y\sin\theta\\ n_yn_z(1-\cos\theta)-n_x\sin\theta\\ {n_z}^2(1-\cos\theta)+\cos\theta\end{bmatrix}^{\mathrm{T}}.$$

Note that $\mathbf{p}'$ and friends are actually row vectors, we are just writing them as transposed column vectors to fit on the page.

Constructing the matrix from these basis vectors, we get

3D matrix to rotate about an arbitrary axis

$$\mathbf{R}(\hat{\mathbf{n}}, \theta) = \begin{bmatrix} -\mathbf{p}'- \\ -\mathbf{q}'- \\ -\mathbf{r}'- \end{bmatrix}$$

$$= \begin{bmatrix} {n_x}^2 (1-\cos\theta) + \cos\theta & n_x n_y (1-\cos\theta) + n_z \sin\theta & n_x n_z (1-\cos\theta) - n_y \sin\theta \\ n_x n_y (1-\cos\theta) - n_z \sin\theta & {n_y}^2 (1-\cos\theta) + \cos\theta & n_y n_z (1-\cos\theta) + n_x \sin\theta \\ n_x n_z (1-\cos\theta) + n_y \sin\theta & n_y n_z (1-\cos\theta) - n_x \sin\theta & {n_z}^2 (1-\cos\theta) + \cos\theta \end{bmatrix}.$$

5.2 Scale

We can scale an object to make it proportionally bigger or smaller by a factor of k. If we apply this scale to the entire object, thus "dilating" the object about the origin, we are performing a *uniform scale*. Uniform scale preserves angles and proportions. Lengths increase or decrease uniformly by a factor of k, areas by a factor of k^2, and volumes (in 3D) by a factor of k^3.

If we wish to "stretch" or "squash" the object, we can apply different scale factors in different directions, resulting in *nonuniform scale*. Nonuniform scale does not preserve angles. Lengths, areas, and volumes are adjusted by a factor that varies according to the orientation relative to the direction of scale.

If $|k| < 1$, then the object gets "shorter" in that direction. If $|k| > 1$, then the object gets "longer." If $k = 0$, then we have an *orthographic projection*, discussed in Section 5.3. If $k < 0$, then we have a *reflection*, covered in Section 5.4. For the remainder of this section, we will assume that $k > 0$.

Section 5.2.1 begins with the simple case of scaling along the cardinal axes. Then Section 5.2.2 examines the general case, scaling along an arbitrary axis.

5.2.1 Scaling along the Cardinal Axes

The simplest scale operation applies a separate scale factor along each cardinal axis. The scale *along* an axis is applied *about* the perpendicular axis (in 2D) or plane (in 3D). If the scale factors for all axes are equal, then the scale is uniform; otherwise, it is nonuniform.

In 2D, we have two scale factors, k_x and k_y. Figure 5.6 shows an object with various scale values for k_x and k_y.