# Chapter 6

# Coordinate Systems

The idea of a *coordinate system*, or *coordinate frame* is pervasive in computer graphics. For example, it is usual to build a model in its own *modeling frame*, and later place this model into a scene in the *world coordinate frame*. We often refer to the modeling frame as the *object frame*, and the world coordinate frame as the *scene frame*. Figure 6.1 shows a cylinder that has been first rotated about its own center, then translated to the position $\mathbf{x}$ in the world frame. Such a transformation can be encoded in a transformation matrix from the model frame to world frame,

$$M_{mw} = T_{mw} R_{mw},$$

where $T_{mw}$ encodes the translation and $R_{mw}$ encodes the rotation.

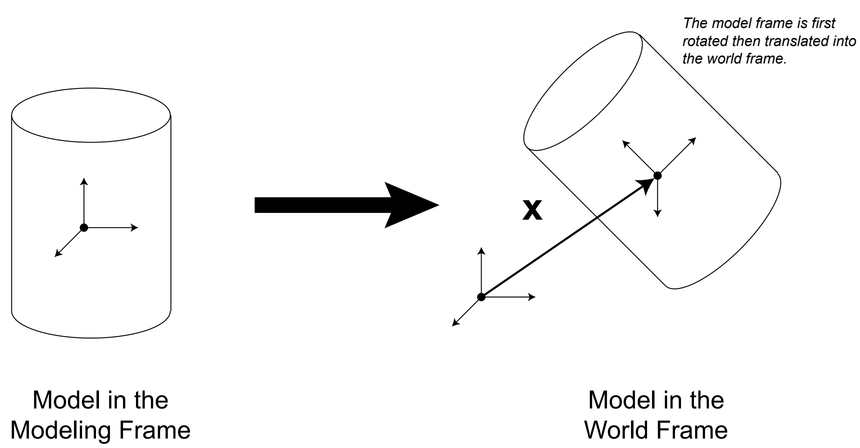
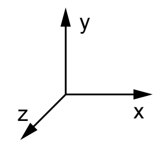



Figure 6.1: Model transformed from *modeling* to *world* coordinate frames.

## 6.1 Left and right handed coordinate frames




Let us develop the idea of a coordinate frame and how we can construct them for use in computer graphics. A 3D coordinate frame might be drawn as shown in the diagram to the right. The three axes are understood to be at right angles (orthogonal) to each other. In the figure, $x$ denotes the horizontal axis, $y$ the vertical axis, and $z$ the depth axis (coming out of the page). This the usual *right-handed* coordinate system seen in Computer Graphics.

The coordinate system shown above is called right handed, since if you place your thumb, index finger and the middle finger of the right hand at right angles to each other, as demonstrated in Figure 6.2, they look like coordinate axes. The thumb represents the $x$ axis, the index finger represents the $y$ axis, and the middle finger represents the $z$ axis. As can be seen from the figure, when done with the left hand, the $z$ axis, is reversed, measuring depth into the page.

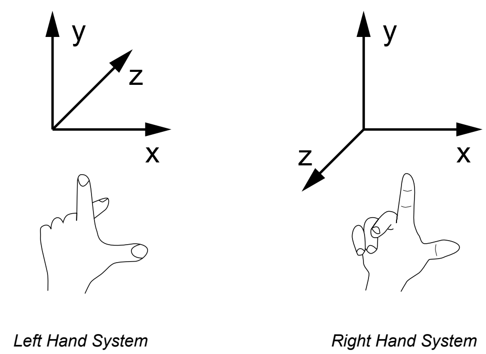
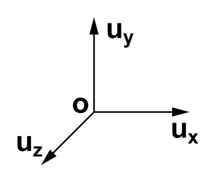



Figure 6.2: Left handed vs. right handed coordinate frames.

## 6.2 Coordinate frame expressed as a point and orthogonal unit vectors




In any coordinate system, the position where the coordinate axes cross is called the *origin*, and by definition has the coordinates $\mathbf{O} = (0, 0, 0)$ in that coordinate system. In order to work with the coordinate frames in the algebraic language of vectors and matrices, we can re-label the axes of the coordinate system with unit vectors directed along the coordinate directions, as shown in the diagram to the right. We use the notation $\mathbf{u}_x$ to represent a unit vector in the $x$ direction, $\mathbf{u}_y$ in the $y$ direction, and $\mathbf{u}_z$ in the $z$ direction. With this notation, a 3D point $\mathbf{p} = (p_x, p_y, p_z)$ in this coordinate frame can be rewritten

$$\mathbf{p} = \mathbf{O} + p_x\mathbf{u_x} + p_y\mathbf{u_y} + p_z\mathbf{u_z}.$$

Now, if we wish to rotate our coordinate frame we can simply apply a rotation matrix to the vectors $\mathbf{u_x}$, $\mathbf{u_y}$, and $\mathbf{u_z}$, and if we wish to translate the frame we can apply a translation matrix to $\mathbf{O}$.

A trick is to represent vectors in homogeneous coordinates on the plane $w = 0$ and points on the plane $w = 1$. For example, a surface normal vector might be written $\mathbf{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \\ 0 \end{bmatrix}$, while a point might be written $\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$. If we are consistent with this notation, then a rotation matrix will affect both points and vectors, but a translation matrix will affect only points. For example

$$\begin{bmatrix} 1 & 0 & 0 & \triangle x \\ 0 & 1 & 0 & \triangle y \\ 0 & 0 & 1 & \triangle z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x \\ n_y \\ n_z \\ 0 \end{bmatrix} = \begin{bmatrix} n_x \\ n_y \\ n_z \\ 0 \end{bmatrix},$$

but

$$\begin{bmatrix} 1 & 0 & 0 & \triangle x \\ 0 & 1 & 0 & \triangle y \\ 0 & 0 & 1 & \triangle z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} p_x + \triangle x \\ p_y + \triangle y \\ p_z + \triangle z \\ 1 \end{bmatrix}.$$

A transformation matrix that first rotates and then translates will then work perfectly for transforming both vectors and points in one frame to another frame. Thus, to transform from one frame to another frame we have a choice. We can either transform the coordinate frame itself, representing this transformation by a matrix, and leave all of the points and normals in the original coordinate frame. Or, we can transform all the points and normals from the original frame to the new frame. The latter approach is referred to as "baking" the transformation.

## 6.3 Creating coordinate frames

If we have two vectors in space and an origin point we can conveniently construct a coordinate frame by using the vector cross product operation. Let us say that we have the vectors $\mathbf{a}$ and $\mathbf{b}$, and compute their cross product $\mathbf{a} \times \mathbf{b}$ as shown in Figure 6.3. The cross product vector is perpendicular to both $\mathbf{a}$ and $\mathbf{b}$.

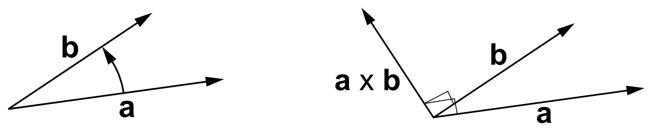



Figure 6.3: $\mathbf{a} \times \mathbf{b}$ is a vector orthogonal to both $\mathbf{a}$ and $\mathbf{b}$.

Figure 6.4 shows a coordinate frame construction made using this observation. We can arbitrarily pick either $\mathbf{a}$ or $\mathbf{b}$ to represent one of the coordinate axes. For example, in the figure we have chosen $\mathbf{a}$ to be aligned with the $x$ direction. Thus, $\mathbf{u_x} = \mathbf{a}/\|\mathbf{a}\|$ is the unit vector in the direction of $\mathbf{a}$. Since $\mathbf{a} \times \mathbf{b}$ will be perpendicular to both $\mathbf{a}$ and $\mathbf{b}$, it must be aligned with one of the other axes, for example the $z$ axis. Thus, $\mathbf{u_z} = (\mathbf{a} \times \mathbf{b})/\|\mathbf{a} \times \mathbf{b}\|$. Finally, the $y$ axis, which must be perpendicular to both the $x$ and $z$ axes, is simply $\mathbf{u_y} = \mathbf{u_z} \times \mathbf{u_x}$.

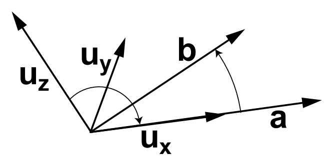



Figure 6.4: A coordinate frame constructed from the vectors $\mathbf{a}$ and $\mathbf{b}$.

Once we have the three unit vectors describing the coordinate frame, it is easy to turn these into a rotation matrix that transforms from the current frame to this new frame. The matrix

$$R = \begin{bmatrix} \mathbf{u}_x & \mathbf{u}_y & \mathbf{u}_z \end{bmatrix},$$

whose columns are the three unit vectors, will do this. You can demonstrate to yourself that this matrix works, since it rotates the three coordinate axes into these new vectors:

$$R\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{u}_x, R\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \mathbf{u}_y, \text{ and } R\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{u}_z.$$

And, we know that this matrix is doing only a pure roation since $\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z$ are mutually orthogonal unit vectors. This is exactly the condition that must hold for a matrix to describe a pure rotation.

To complete the description of the transform from the current frame to the new frame, we also need to provide a translation from the old origin to the new origin. Again, this can be built into a matrix and multiplied into the matrix $R$ on the left.