Conversion of Left-Handed Coordinates to Right-Handed Coordinates

David Eberly
Geometric Tools, LLC
http://www.geometrictools.com/
Copyright © 1998-2012. All Rights Reserved.

Created: December 20, 2003
Last Modified: March 1, 2008

Contents

1 Right-Handed Coordinate Systems **2**

2 Left-Handed Coordinate Systems **3**

2.1 Heading 4

2.2 Pitch 5

2.3 Bank 6

2.4 Order of Rotations 7

3 Conversion from Left-Handedness to Right-Handedness **7**

3.1 Conversion of Points and Translations 7

3.2 Conversion of Rotations 8

3.3 Conversion of Affine Transformations 10

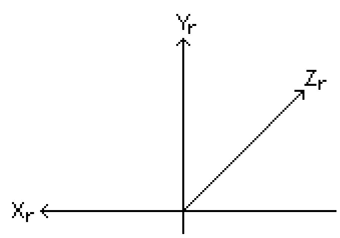
3.4 Conversion of Cameras and Lights 10

This document describes the process by which you convert data in a left-handed coordinate system to that in a right-handed coordinate system. The data includes vertex locations, translational and rotational transformations, camera location and orientation, and light location and orientation.

1 Right-Handed Coordinate Systems

A *right-handed coordinate system* is described as $+X_r$ in the *left* direction, $+Y_r$ in the *up* direction, and $+Z_r$ in the *view* direction. Figure 1.1 shows a typical representation.

Figure 1.1 A right-handed coordinate system. The coordinate names are subscripted with r to stress the fact these are for the right-handed system. The $+Z_r$ axis is intend to have direction into the plane of the page.



The *origin* of the system is $(0, 0, 0)$. A point (x, y, z) is located relative to the origin by moving x units to the left (when $x > 0$) or to the right (when $x < 0$), y units up (when $y > 0$) or down (when $y < 0$), and z units away (when $z > 0$) or towards (when $z < 0$).

The placement of the axes in Figure 1.1 is based on wanting the Y_r-axis to point upward and the Z_r-axis to point into the plane of the page. To make sure the coordinate system is right-handed, the only choice for the X_r-axis is to point to the left. The direction of X_r is actually not relevant on its own. You may choose X_r to point to the right, Y_r to point up, and Z_r to point out of the plane of the page. Or choose X_r to point to the right, Y_r to point down, and Z_r to point into the plane of the page. For that matter, you may rotate any of these configurations to any other configuration you like, but you *still* have a right-handed system. Choose any of the coordinate directions you wish to be the up vector, choose any remaining direction to be the view direction. You have only one choice for the third vector. If this is the world coordinate system you are choosing for an application, you need only position and orient the camera in such a way that your view of the world is what you expect! You can do this regardless of your convention for what is the up direction and what is the view direction.

In this document, the rotation conventions for right-handed coordinates are that positive angles are associated with counterclockwise rotations when viewing the system along the negative direction of the axis of rotation. The rotation about the x-axis is

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{1}$$

The rotation about the y-axis is

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{2}$$

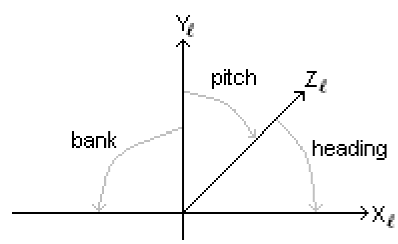
The rotation about the z-axis is

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

2 Left-Handed Coordinate Systems

A *left-handed coordinate system* is described as $+X_\ell$ in the *right* direction, $+Y_\ell$ in the *up* direction, and $+Z_\ell$ in the *view* direction. The prototypical modeling package that uses left-handed coordinates for all objects is LightWave. The discussion here is effectively a description of LightWave's conventions. Figure 2.1 shows a typical representation.

Figure 2.1 A left-handed coordinate system. The coordinate names are subscripted with ℓ to stress the fact these are for the left-handed system. The $+Z_\ell$ axis is intend to have direction into the plane of the page.



The *origin* of the system is $(0, 0, 0)$. A point (x, y, z) is located relative to the origin by moving x units to the right (when $x > 0$) or to the left (when $x < 0$), y units up (when $y > 0$) or down (when $y < 0$), and z units away (when $z > 0$) or towards (when $z < 0$).

Figure 2.1 also shows how coordinate axis rotations are defined in this left-handed system. The direction of the X_ℓ axis is $(1, 0, 0)$, the direction of the Y_ℓ axis is $(0, 1, 0)$, and the direction of the Z_ℓ axis is $(0, 0, 1)$. We may conveniently store the direction vectors as the columns of the identity matrix,

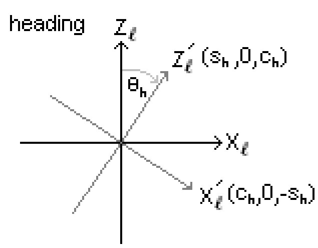
$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Notice that the first column is the right direction, the second column is the up direction, and the third column is the view direction.

2.1 Heading

A rotation about the Y_ℓ-axis represents *heading*. In Figure 2.1, the direction shown by the arrow labeled "heading" corresponds to a positive angle. If you are looking in the $-Y_\ell$ direction, a positive angle is associated with a clockwise rotation. Figure 2.2 shows a view in the $-Y_\ell$ direction.

Figure 2.2 A view in the $-Y_\ell$ direction. A positive angle for heading is associated with a clockwise rotation in the plane of X_ℓ and Z_ℓ.



The *heading angle* is θ_h and the figure illustrates a heading that is a positive angle (clockwise rotation). The Z_ℓ axis is rotated to the Z'_ℓ axis, the latter axis having direction $(s_h, 0, c_h)$ where $s_h = \sin(\theta_h)$ and $c_h = \cos(\theta_h)$. The X_ℓ axis is rotated to the X'_ℓ axis, the latter axis having direction $(c_h, 0, -s_h)$. The Y_ℓ axis is unchanged by the rotation, so the rotated axis Y'_ℓ is the same as the original one. Just as the identity matrix columns are the original directions in the order right, up, and view, the rotated axes may be listed as the columns of another matrix in the same order. This matrix is the *heading matrix*

$$H_\ell(\theta_h) = \begin{bmatrix} c_h & 0 & s_h \\ 0 & 1 & 0 \\ -s_h & 0 & c_h \end{bmatrix}$$

Observe that the first column is the direction of the X'_ℓ axis, the second column is the direction of the Y'_ℓ (and Y_ℓ) axis, and the third column is the direction of the Z'_ℓ axis.

The heading matrix was introduced as a convenient way to store the directions of the rotated coordinate axes. The matrix has double duty in that it may also be used to show how points (x, y, z) in the original coordinate system are rotated to points (x', y', z'). Specifically,

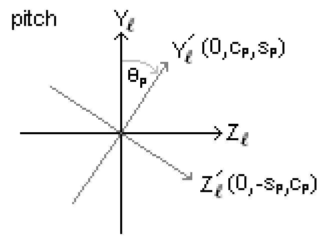
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} c_h & 0 & s_h \\ 0 & 1 & 0 \\ -s_h & 0 & c_h \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} c_h x + s_h z \\ y \\ -s_h x + c_h z \end{bmatrix} \tag{4}$$

The standard rules of matrix multiplication have been applied.

2.2 Pitch

A rotation about the X_ℓ axis represents *pitch*. In Figure 2.1, the direction shown by the arrow labeled "pitch" corresponds to a positive angle. If you are looking in the $-X_\ell$ direction, a positive angle is associated with a clockwise rotation. Figure 2.3 shows a view in the $-X_\ell$ direction.

Figure 2.3 A view in the $-X_\ell$ direction. A positive angle for heading is associated with a clockwise rotation in the plane of Y_ℓ and Z_ℓ.



The *pitch angle* is θ_p and the figure illustrates a pitch that is a positive angle (clockwise rotation). The Z_ℓ axis is rotated to the Z'_ℓ axis, the latter axis having direction $(0, -s_p, c_p)$ where $s_p = \sin(\theta_p)$ and $c_p = \cos(\theta_p)$. The Y_ℓ axis is rotated to the Y'_ℓ axis, the latter axis having direction $(0, c_p, s_p)$. The X_ℓ axis is unchanged by the rotation, so the rotated axis X'_ℓ is the same as the original one. Just as the identity matrix columns are the original directions in the order right, up, and view, the rotated axes may be listed as the columns of another matrix in the same order. This matrix is the *pitch matrix*

$$P_\ell(\theta_p) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_p & -s_p \\ 0 & s_p & c_p \end{bmatrix}$$

Observe that the first column is the direction of the X'_ℓ (and X_ℓ) axis, the second column is the direction of the Y'_ℓ axis, and the third column is the direction of the Z'_ℓ axis.

The pitch matrix was introduced as a convenient way to store the directions of the rotated coordinate axes. The matrix has double duty in that it may also be used to show how points (x, y, z) in the original coordinate system are rotated to points (x', y', z'). Specifically,

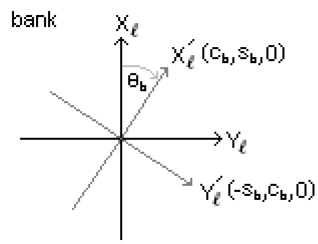
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_p & -s_p \\ 0 & s_p & c_p \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x \\ c_p y - s_p z \\ s_p y + c_p z \end{bmatrix} \tag{5}$$

The standard rules of matrix multiplication have been applied.

2.3 Bank

A rotation about the Z_ℓ axis represents *bank*. In Figure 2.1, the direction shown by the arrow labeled "bank" corresponds to a positive angle. If you are looking in the $-Z_\ell$ direction, a positive angle is associated with a clockwise rotation. Figure 2.4 shows a view in the $-Z_\ell$ direction.

Figure 2.4 A view in the $-Z_\ell$ direction. A positive angle for heading is associated with a clockwise rotation in the plane of X_ℓ and Y_ℓ.



The *bank angle* is θ_b and the figure illustrates a bank that is a positive angle (clockwise rotation). The X_ℓ axis is rotated to the X'_ℓ axis, the latter axis having direction $(c_b, s_b, 0)$ where $s_b = \sin(\theta_b)$ and $c_b = \cos(\theta_b)$. The Y_ℓ axis is rotated to the Y'_ℓ axis, the latter axis having direction $(-s_b, c_b, 0)$. The Z_ℓ axis is unchanged by the rotation, so the rotated axis Z'_ℓ is the same as the original one. Just as the identity matrix columns are the original directions in the order right, up, and view, the rotated axes may be listed as the columns of another matrix in the same order. This matrix is the *bank matrix*

$$B_\ell(\theta_b) = \begin{bmatrix} c_b & -s_b & 0 \\ s_b & c_b & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Observe that the first column is the direction of the X'_ℓ axis, the second column is the direction of the Y'_ℓ axis, and the third column is the direction of the Z'_ℓ (and Z_ℓ) axis.

The bank matrix was introduced as a convenient way to store the directions of the rotated coordinate axes. The matrix has double duty in that it may also be used to show how points (x, y, z) in the original coordinate system are rotated to points (x', y', z'). Specifically,

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} c_b & -s_b & 0 \\ s_b & c_b & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} c_b x - s_b y \\ s_b x + c_b y \\ z \end{bmatrix} \tag{6}$$

The standard rules of matrix multiplication have been applied.

2.4 Order of Rotations

The order of application of heading, pitch, and bank is dependent on the graphics package at hand. LightWave specifies rotations only by a composition of coordinate axis rotations. Moreover, LightWave only tells you the three angles: heading, pitch, and bank. It is important that you apply these in the correct order. In particular, the composite rotation is $R_\ell = H_\ell P_\ell B_\ell$. LightWave uses the convention for transforming vectors, $R_\ell \mathbf{v}$, where the vector $\mathbf{v}$ is a 3×1 column vector. Thus, LightWave applies bank first, pitch second, and heading third.

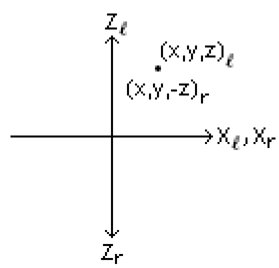
3 Conversion from Left-Handedness to Right-Handedness

The simplest method for conversion is to flip the direction of one of the axes of the left-handed system. It does not matter which one. The analysis here uses the flip of the z-axis direction. You may also flip the direction of all three axes, but no sense in doing more work than you have to.

3.1 Conversion of Points and Translations

This is where the flip of the z-axis direction is obvious. A point (x, y, z) in the left-handed system is converted to a point $(x, y, -z)$ in the right-handed system. Figure 3.1 illustrates this in the xz-plane with the two coordinate systems superimposed.

Figure 3.1 Conversion of a point in left-handed coordinates (subscripted with an ℓ) to a point in right-handed (subscripted with a r). Just change sign on the z-component.



The figure shows that the point in the left-handed coordinate system has a positive z component. In the right-handed coordinate system, observe that the z component must be negative. In matrix-vector form, the conversion from a left-handed point $\mathbf{Q}_\ell$ to a right-handed point $\mathbf{Q}_r$ is

$$\mathbf{Q}_r = \begin{bmatrix} x \\ y \\ -z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = S_z \mathbf{Q}_\ell$$

where S_z is defined to be the diagonal matrix

$$S_z = \mathrm{Diag}(1, 1, -1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

The translational component of a transformation is modified just as points are. The z component is negated. Pseudocode for conversion of a point (or translation) is

```
Vector3 LH_Point(LH_X, LH_Y,  LH_Z);  // left-handed coordinates of point
Vector3 RH_Point(LH_X, LH_Y, -LH_Z);  // right-handed coordinates of point
```

3.2 Conversion of Rotations

Conversion of Heading. Equation 4 shows that an input left-handed point $\mathbf{Q}_\ell = (x, y, z)$ is transformed by the heading matrix to an output left-handed point $\mathbf{Q}'_\ell = (x', y', z')$ by

$$\mathbf{Q}'_\ell = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} c_h & 0 & s_h \\ 0 & 1 & 0 \\ -s_h & 0 & c_h \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = H_\ell \mathbf{Q}_\ell$$

In right-handed coordinates, both the input and output must have negated z components, so the transformation in the right-handed system is $\mathbf{Q}_r = (x, y, -z) \to (x', y', -z') = \mathbf{Q}'_r$, or

$$\mathbf{Q}'_r = S_z \mathbf{Q}'_\ell = S_z H_\ell \mathbf{Q}_\ell = (S_z H_\ell S_z)\mathbf{Q}_r$$

The heading matrix H_r in the right-handed system corresponding to the heading matrix H_ℓ in the left-handed system is

$$H_r = S_z H_\ell S_z \tag{7}$$

For a matrix $M = [m_{ij}]$ with $0 \le i \le 2$ and $0 \le j \le 2$, the operation $S_z M S_z$ is equivalent to changing the signs on m_{02}, m_{12}, m_{20}, and m_{21}.

Conversion of Pitch. Equation 5 shows that an input left-handed point $\mathbf{Q}_\ell = (x, y, z)$ is transformed by the pitch matrix to an output left-handed point $\mathbf{Q}'_\ell = (x', y', z')$ by

$$\mathbf{Q}'_\ell = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_p & -s_p \\ 0 & s_p & c_p \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = P_\ell \mathbf{Q}_\ell$$

In right-handed coordinates, both the input and output must have negated z components, so the transformation in the right-handed system is $\mathbf{Q}_r = (x, y, -z) \to (x', y', -z') = \mathbf{Q}'_r$, or

$$\mathbf{Q}'_r = S_z \mathbf{Q}'_\ell = S_z P_\ell \mathbf{Q}_\ell = (S_z P_\ell S_z)\mathbf{Q}_r$$

The pitch matrix P_r in the right-handed system corresponding to the pitch matrix P_ℓ in the left-handed system is

$$P_r = S_z P_\ell S_z \tag{8}$$

Conversion of Bank. Equation 6 shows that an input left-handed point $\mathbf{Q}_\ell = (x, y, z)$ is transformed by the bank matrix to an output left-handed point $\mathbf{Q}'_\ell = (x', y', z')$ by

$$\mathbf{Q}'_\ell = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} c_b & -s_b & 0 \\ s_b & c_b & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = B_\ell \mathbf{Q}_\ell$$

In right-handed coordinates, both the input and output must have negated z components, so the transformation in the right-handed system is $\mathbf{Q}_r = (x, y, -z) \to (x', y', -z') = \mathbf{Q}'_r$, or

$$\mathbf{Q}'_r = S_z \mathbf{Q}'_\ell = S_z B_\ell \mathbf{Q}_\ell = (S_z B_\ell S_z)\mathbf{Q}_r$$

The bank matrix B_r in the right-handed system corresponding to the bank matrix B_ℓ in the left-handed system is

$$B_r = S_z B_\ell S_z \tag{9}$$

The Composite Rotation. The left-handed composite rotation is $R_\ell = H_\ell P_\ell B_\ell$. The transformation of $\mathbf{Q}_\ell = (x, y, z)$ by R_ℓ to $\mathbf{Q}'_\ell$ is

$$\mathbf{Q}'_\ell = R_P \mathbf{Q}_\ell$$

In right-handed coordinates where $\mathbf{Q}_r = (x, y, -z) = S_z \mathbf{Q}_\ell$ and $\mathbf{Q}'_r = S_z \mathbf{Q}'_\ell$, we have

$$\mathbf{Q}'_r = S_z \mathbf{Q}'_\ell = S_z R_P \mathbf{Q}_\ell = (S_z R_P S_z)\mathbf{Q}_r$$

The right-handed rotation R_r corresponding to the left-handed rotation R_ℓ is therefore,

$$R_r = S_z R_\ell S_z \tag{10}$$

Pseudocode for the conversion of rotations is shown below. The equations for the rotations are (1), (2), and (3).

```
// left-handed rotation angles
float LH_AngleH, LH_AngleP, LH_AngleB;

// left-handed coordinate rotation matrices
Matrix3 LH_H((0,1,0), LH_AngleH);  // heading
Matrix3 LH_P((1,0,0), LH_AngleP);  // pitch
Matrix3 LH_B((0,0,1), LH_AngleB);  // bank

// left-handed composite rotation
Matrix3 LH_Rotate = LH_H * LH_P * LH_B;

// right-handed composite rotation
Matrix3 RH_Rotate = LH_Rotate;
```

```
RH_Rotate[0][2] = -RH_Rotate[0][2];
RH_Rotate[1][2] = -RH_Rotate[1][2];
RH_Rotate[2][0] = -RH_Rotate[2][0];
RH_Rotate[2][1] = -RH_Rotate[2][1];
```

3.3 Conversion of Affine Transformations

Let $A_\ell = [a_{ij}]$ be a 3×3 matrix and let $\mathbf{b}_\ell = [b_i]$ be a 3×1 vector for $0 \le i \le 2$ and $0 \le j \le 2$. The affine transformation of $\mathbf{Q}_\ell = (x, y, z)$ to $\mathbf{Q}'_\ell = (x', y', z')$ by A and $\mathbf{b}$ is given in matrix form by

$$\mathbf{Q}'_\ell = A_\ell \mathbf{Q}_\ell + \mathbf{b}_\ell$$

The input and output points in right-handed coordinates are $\mathbf{Q}_r = (x, y, -z)$ and $\mathbf{Q}'_r = (x', y', -z')$. The conversion is

$$\begin{aligned} \mathbf{Q}'_r &= S_z \mathbf{Q}'_\ell \\ &= S_z (A_\ell \mathbf{Q}_\ell + \mathbf{b}_\ell) \\ &= S_z A_\ell \mathbf{Q}_\ell + S_z \mathbf{b}_\ell \\ &= (S_z A_\ell S_z) \mathbf{Q}_r + S_z \mathbf{b}_\ell \end{aligned}$$

Thus, the corresponding 3×3 matrix for the right-handed system is $A_r = S_z A_\ell S_z$ and the 3×1 vector for the right-handed system is $\mathbf{b}_r = S_z \mathbf{b}_\ell$. This amounts to sign changes on the components a_{02}, a_{12}, a_{20}, a_{21}, and b_2.

3.4 Conversion of Cameras and Lights

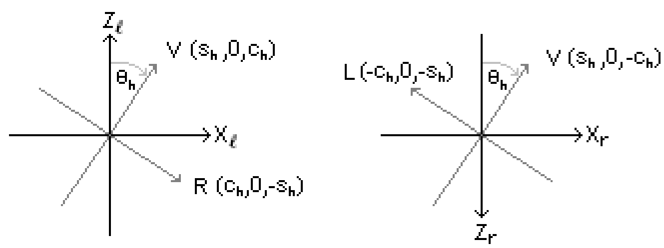
The conversion of cameras and lights are handled in the same manner, so the discussion is presented only for cameras.

The camera has its own coordinate system whose origin is the eye point. In left-handed coordinates, the camera coordinate system is itself left-handed, just as world coordinates are. The camera has a right direction, an up direction, and a view direction. These directions are abstractly stored as the columns of a rotation matrix. Assuming the rotation matrix is represented only as a product of heading, pitch, and bank, we must obtain the camera coordinate directions from the heading, pitch, and bank angles.

A right-handed coordinate system uses a camera that itself uses right-handed coordinates, just like the world does. The camera has a left direction, an up direction, and a view direction. The fact that the handedness of the world and the camera must be changed makes the conversion process for cameras slightly more complicated than the conversion of rotational transformations.

To illustrate, consider a left-handed camera with eye point at the origin and with a small, positive heading angle θ_h. The pitch and bank angles are zero. The left image of Figure 3.2 shows the new view (V) and right (R) directions for the left-handed camera. The right image of the figure shows the new view (V) and left (L) directions for the right-handed camera.

Figure 3.2 Left: The right and view directions for the left-handed camera. Right: The left and view directions for the right-handed camera.



This example illustrates the two steps that must be taken to convert the left-handed camera to a right-handed camera. First, the right, up, and view direction vectors must have their z components negated. This step puts the vectors into the the right-handed coordinate system. Second, the right vector must be negated to make it a left vector.

In matrix notation for this example, the left-handed camera coordinate directions are stored as the columns of the heading matrix H_ℓ. Changing sign on the z components of all three columns is equivalent to the matrix product $S_z H_\ell$, where $S_z = \mathrm{Diag}(1, 1, -1)$. Changing sign on the first column (the right direction) is equivalent to multiplying on the right of the product by the diagonal matrix $S_x = \mathrm{Diag}(-1, 1, 1)$, namely $S_z H_\ell S_x$. To verify for our example,

$$S_z H_\ell S_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} c_h & 0 & s_h \\ 0 & 1 & 0 \\ -s_h & 0 & c_h \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -c_h & 0 & s_h \\ 0 & 1 & 0 \\ -s_h & 0 & -c_h \end{bmatrix}$$

Observe that, in fact, the right-handed camera's left vector is the first column of $S_z H_\ell S_x$, the up vector is the second column, and the view vector is the third column, as illustrated in Figure 3.2.

The general construction is similar. If $R_\ell = H_\ell P_\ell B_\ell$ is the left-handed system's composite matrix built from the heading, pitch, and bank matrices, then the columns of R_ℓ are the right, up, and view vectors for the camera. To convert to a right-handed camera, the z components of the columns must all be negated to place those vectors in the right-handed coordinate system. Then the first column must be negated to convert the right vector to a left vector. The two steps together may be written as

$$C_r = S_z R_\ell S_x$$

The matrix C_r represents the right-handed camera coordinates. The first column of C_r is the left vector, the second column is the up vector, and the third column is the view vector. For a matrix $M = [m_{ij}]$ with $0 \le i \le 2$ and $0 \le j \le 2$, the operation $S_z M S_x$ is equivalent to changing the signs on m_{00}, m_{10}, m_{21}, and m_{22}.

Pseudocode for the conversion is shown below.

```
// left-handed camera location
```

```
Vector3 LH_CameraLocation(LH_X, LH_Y, LH_Z);

// right-handed camera location
Vector3 RH_CameraLocation(LH_X, LH_Y, -LH_Z);

// left-handed rotation angles
float LH_AngleH, LH_AngleP, LH_AngleB;

// left-handed coordinate rotation matrices
Matrix3 LH_H((0,1,0), LH_AngleH);  // heading
Matrix3 LH_P((1,0,0), LH_AngleP);  // pitch
Matrix3 LH_B((0,0,1), LH_AngleB);  // bank

// left-handed camera matrix (colums are right, up, view)
Matrix3 LH_CameraMatrix = LH_H * LH_P * LH_B;

// right-handed camera matrix (columns are left, up, view)
Matrix3 RH_CameraMatrix = LH_CameraMatrix;
RH_CameraMatrix[0][0] = -RH_CameraMatrix[0][0];
RH_CameraMatrix[1][0] = -RH_CameraMatrix[1][0];
RH_CameraMatrix[2][1] = -RH_CameraMatrix[2][1];
RH_CameraMatrix[2][2] = -RH_CameraMatrix[2][2];
```