VII.1

DECOMPOSING A MATRIX INTO SIMPLE TRANSFORMATIONS

Spencer W. Thomas
University of Michigan
Ann Arbor, Michigan

Sometimes, it is useful to be able to extract a sequence of simple transformations (scale, rotate, etc.) that will reproduce a given transformation matrix. This gem provides a way to do that. In particular, given (almost[1]) any 4×4 transformation matrix $\mathbf{M}$, it will compute the arguments to the following sequence of transformations, such that concatenating the transformations will reproduce the original matrix (to within a homogeneous scale factor):

$$\mathrm{Scale}(s_x,\ s_y,\ s_z)\mathrm{Shear}^2_{xy}\ \mathrm{Shear}_{xz}\mathrm{Shear}_{yz}\ \mathrm{Rotate}_x\ \mathrm{Rotate}_y$$

$$\mathrm{Rotate}_z\ \mathrm{Translate}(t_x,\ t_y,\ t_z)$$

$$\mathrm{Perspective}(p_x,\ p_y,\ p_z,\ p_w)$$

This routine has been used for tasks such as removing the shears from a rotation matrix, for feeding an arbitrary transformation to a graphics system that only *understands* a particular sequence of transformations (which is particularly useful when dealing with rotations), or for any other application in which you want just part of the transformation sequence.

[1]The only constraint is that the product of the [4, 4] element with the determinant of the upper left 3 x 3 component of the matrix be nonzero.

[2]Shear_{xy} shears the x coordinate as the y coordinate changes. The matrix corresponding to this transformation is

$$\begin{bmatrix} 1 & 0 & 0 \\ s_{xy} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The Algorithm

The algorithm works by *undoing* the transformation sequence in reverse order. It first determines perspective elements that, when *removed* from the matrix, will leave the last column (the *perspective partition)* as (0, 0, 0, 1)^T. Then it extracts the translations. This leaves a 3 × 3 matrix comprising the scales, shears, and rotations. It is decomposed from the left, extracting first the scaling factors and then the shearing components, leaving a pure rotation matrix. This is broken down into three consecutive rotations.

Extracting the perspective components is the messiest part. Essentially, we need to solve the matrix equation:

$$\begin{bmatrix} \mathbf{M}_{1,1} & \mathbf{M}_{1,2} & \mathbf{M}_{1,3} & \mathbf{M}_{1,4} \\ \mathbf{M}_{2,1} & \mathbf{M}_{2,2} & \mathbf{M}_{2,3} & \mathbf{M}_{2,4} \\ \mathbf{M}_{3,1} & \mathbf{M}_{3,2} & \mathbf{M}_{3,3} & \mathbf{M}_{3,4} \\ \mathbf{M}_{4,1} & \mathbf{M}_{4,2} & \mathbf{M}_{4,3} & \mathbf{M}_{4,4} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{M}_{1,1} & \mathbf{M}_{1,2} & \mathbf{M}_{1,3} & 0 \\ \mathbf{M}_{2,1} & \mathbf{M}_{2,2} & \mathbf{M}_{2,3} & 0 \\ \mathbf{M}_{3,1} & \mathbf{M}_{3,2} & \mathbf{M}_{3,3} & 0 \\ \mathbf{M}_{4,1} & \mathbf{M}_{4,2} & \mathbf{M}_{4,3} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & p_w \end{bmatrix},$$

which reduces to:

$$\begin{bmatrix} \mathbf{M}_{1,4} \\ \mathbf{M}_{2,4} \\ \mathbf{M}_{3,4} \\ \mathbf{M}_{4,4} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{1,1} & \mathbf{M}_{1,2} & \mathbf{M}_{1,3} & 0 \\ \mathbf{M}_{2,1} & \mathbf{M}_{2,2} & \mathbf{M}_{2,3} & 0 \\ \mathbf{M}_{3,1} & \mathbf{M}_{3,2} & \mathbf{M}_{3,3} & 0 \\ \mathbf{M}_{4,1} & \mathbf{M}_{4,2} & \mathbf{M}_{4,3} & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ p_w \end{bmatrix}.$$

Assuming that the upper left 3 × 3 partition of **M** is not singular, this can be solved easily for p_x, p_y, p_z, and p_w Since some of the later steps will not work if this partition is singular, this is not a serious defect.

The next step is to extract the translations. This is trivial; we find $t_x = \mathbf{M}_{4,1}$, $t_y = \mathbf{M}_{4,2}$, and $t_z = \mathbf{M}_{4,3}$. At this point, we are left with a 3×3 matrix, $\mathbf{M}' = \mathbf{M}_{1..3,1..3}$.

The process of finding the scaling factors and shear parameters is interleaved. First, find $s_x = |\mathbf{M}'_1|$. Then, compute an initial value for the xy shear factor, s_{xy}, $= \mathbf{M}'_1$, $\mathbf{M}'_2$ (This is too large by the y scaling factor.) The second row of the matrix is made orthogonal to the first by setting $\mathbf{M}'_2 \leftarrow \mathbf{M}'_2 - s_{xy}\mathbf{M}'_1$. Then the y scaling factor, s_y is the length of the modified second row. The second row is normalized, and s_{xy} is divided by s_y to get its final value. The xz and yz shear factors are computed as in the preceding, the third row is made orthogonal to the first two rows, the z scaling factor is computed, the third row is normalized, and the xz and yz shear factors are rescaled.

The resulting matrix now is a pure rotation matrix, except that it might still include a scale factor of –1. If the determinant of the matrix is –1, negate the matrix and all three scaling factors. Call the resulting matrix **R**.

Finally, we need to decompose the rotation matrix into a sequence of rotations about the x, y, and z axes. If the rotation angle about x is α, that about y is β, and that about z is γ, then the composite rotation is:

$$\mathbf{R} = \begin{bmatrix} \cos\{\beta\}\cos\{\gamma\} & \cos\{\beta\}\sin\{\gamma\} & -\sin[\beta] \\ \sin\{\alpha\}\sin\{\beta\}\cos\{\gamma\} & \sin\{\alpha\}\sin\{\beta\}\sin\{\gamma\} & \\ -\cos\{\alpha\}\sin\{\gamma\} & +\cos\{a\}\cos\{\gamma\} & \sin\{\alpha\}\cos\{\beta\} \\ \cos\{\alpha\}\sin\{\beta\}\cos\{y\} & \cos\{\alpha\}\sin\{\beta\}\sin\{\gamma\} & \\ +\sin\{\alpha\}\sin\{\gamma\} & -\sin\{\alpha\}\cos\{\gamma\} & \cos\{\alpha\}\cos\{\beta\} \end{bmatrix}.$$

Thus, $\beta = \arcsin(-\mathbf{R}_{1,3})$. If $\cos(\beta) \neq 0$, α is derived easily from $\mathbf{R}_{2,3}$ and $\mathbf{R}_{3,3}$, and γ from $\mathbf{R}_{1,2}$ and $\mathbf{R}_{1,1}$. If $\cos(\beta) = 0$, then $\mathbf{R}$ reduces to:

$$\begin{bmatrix} 0 & 0 & \pm 1 \\ \sin\{\alpha \pm \gamma\} & \cos\{\alpha \pm \gamma\} & 0 \\ \cos\{\alpha \pm \gamma\} & -\sin\{\alpha \pm \gamma\} & 0 \end{bmatrix}.$$

In this case, we arbitrarily set γ to 0 and derive α from $\mathbf{R}_{2,1}$ and $\mathbf{R}_{2,2}$. This finishes the decomposition.

See also 7.2 Recovering the Data from the Transformation Matrix, Ronald N. Goldman