Fast Incremental Transformation of Bounding Boxes, with Generalizations to Regular Grids like Terrain and Volume data

Hansong Zhang

Department of Computer Science
UNC-Chapel Hill

9/17/97

1 Introduction

This note explains a fast way of transforming the eight corners of an bounding box with a 4×4 matrix. If the matrix is a composite of the viewing matrix and the projection matrix, then the transformation projects the bounding box into the screen space. Some applications need to do this to estimate the area of projection or find the screen-space bounding rectangle of the bounding box. Further, the method can be applied to transforming any regular grids by a matrix.

2 Incremental Transformation of Axis-Aligned Bounding Boxes

We define an axis-aligned bounding box (AABB) by a base point, (x_0, y_0, z_0), and the increments (dx, dy, dz), so that the eight corners of the bounding box are:

(x_0, y_0, z_0)
$(x_0, y_0 + dy, z_0)$

$(x_0, y_0 + dy, z_0 + dz)$
$(x_0, y_0, z_0 + dz)$
(x_0, y_0, z_0)
$(x_0 + dx, y_0, z_0)$
$(x_0 + dx, y_0 + dy, z_0)$
$(x_0 + dx, y_0 + dy, z_0 + dz)$
$(x_0 + dx, y_0, z_0 + dz)$

The transformations of the corners share many common sub-expressions, which can be computed once and stored for later use.

Let the elements of the transformation matrix, M, be m_{ij}, $0 \leq i, j < 3$:

$$M = \begin{pmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \\ m_{30} & m_{31} & m_{32} & m_{33} \end{pmatrix}$$

Then

$$M \begin{bmatrix} x_0 + dx \\ y_0 + dy \\ z_0 + dz \\ 1 \end{bmatrix} = M \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} + M \begin{bmatrix} dx \\ 0 \\ 0 \\ 0 \end{bmatrix} + M \begin{bmatrix} 0 \\ dy \\ 0 \\ 0 \end{bmatrix} + M \begin{bmatrix} 0 \\ 0 \\ dz \\ 0 \end{bmatrix}$$

By computing the full transformation of the base point, and one increment vector in each dimension ($M(dx, 0, 0, 0)^T$, etc.), the transformtion of the remaining 7 corners is no more than adding proper increment vectors (and a division by the w component). The increment vectors are:

$$\begin{aligned} M(dx, 0, 0, 0)^T &= (m_{00}dx, m_{10}dx, m_{20}dx, m_{30}dx)^T \\ M(0, dy, 0, 0)^T &= (m_{01}dy, m_{11}dy, m_{21}dy, m_{31}dy)^T \\ M(0, 0, dz, 0)^T &= (m_{02}dy, m_{12}dy, m_{22}dy, m_{32}dy)^T \end{aligned}$$

In computer graphics, the last column of M is often not full, a fact that further reduces (slightly) the amount computation in the "full" transformation of the base point.

3 Incremental Transformation of Oriented Bounding Boxes

An oriented bounding box (OBB) can be thought of an rotated axis-aligned bounding box. More precisely, a vertex V_{obb} on the OBB is related to a corresponding vertex V_{aabb} on the AABB by a rotation matrix R of the form

$$M = \begin{pmatrix} r_{00} & r_{01} & r_{02} & 0 \\ r_{10} & r_{11} & r_{12} & 0 \\ r_{20} & r_{21} & r_{22} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

so that $V_{obb} = RV_{aabb}^T$. Thus we have $MV_{obb} = MRV_{aabb}^T$, which says that transforming the OBB by M is equiverlent to transforming the corresponding AABB with a different matrix, RM. This can be done the same way as in the previous section.

Apparently, the OBB transformation (compared to AABB) has the extra cost of computing RM. This computation is that of a 3×3 matrix multiplying a 3×4 matrix. Even with this extra cost, the incremental transformation still provides substantial saving over direct transformations of the eight corners.

4 Generalization

It is obvious that the methods discussed so far apply to any regular $M \times N \times P$ grids. The bounding boxes form a special case as $2 \times 2 \times 2$ grids. Terrain (or height field) data are the $M \times N \times 1$ case. Volume data sets are of course full $M \times N \times P$ grids. Portals, as used in the visiblity culling algorithm with cells and portals for architectural environments, are $2 \times 2 \times 1$ grids. All these types of grids routinely require transformation by matrices, and the incremental transformation methods described above help to accelerate this process.