Tangent Space

Tangent space is a coordinate system that best describes the face of a triangle. Often when reading the latest papers on rendering techniques it is assumed that the reader knows how to calculate the tangent space matrix. Any provided example code skirts the issue by hardcoding values for simple geometry. Without a thorough understanding of what tangent space is and how to correctly calculate it in practice one will simply not get the expected results from any shading technique requiring it.

Coordinate Spaces

Computer graphics deals with three major coordinate spaces: World space, eye space, and tangent space.

World Space

World space is the coordinate system that best describes the entire scene. Imagine you have a floor plan of a building with the lower left hand corner defined as the origin. All points (x,y) in the map will be defined in terms of a unit of distance from the corner much like how all points (x,y,z) in a triangle based rendering engine are defined in relation to some origin arbitrarily chosen by a designer.

Eye Space

Eye space describes the same set of points in terms of how it is viewed by the observer. The eye is always the origin in eye space and points are described in relation to the eye point. As the observer moves around the scene the coordinates for every object in the scene changes. As the observer rotates, all the points in the scene rotate with the observer to maintain the forward eye direction.

Tangent Space

As I said before, this coordinate system best describes the face of a triangle. During the process of rendering a single triangle of thousands that may be within the observers field of view it is convenient to create a coordinate system in terms of the triangle's face and texture coordinates. Often data is stored within the texture, such as height information and/or per-pixel surface normals, this data by definition, is already in tangent space. However, other information, such as the direction of incoming light, is not. By creating the tangent space matrix we can rotate vectors from eye space into our local coordinate system.

Triangle inputs and the normal vector

We are given a triangle composed of three vertices with corresponding per vertex texture coordinates u and v. We can compute the normal of the triangle by performing the cross product of the two vectors spanning the triangle using basic vector algebra.

$$p = v_2 - v_1 \qquad\qquad q = v_3 - v_1 \qquad\qquad n = p \times q$$

Note: This assumes counter clockwise vertex winding.

Tangent and Bitangent

The tangent and bitangent vectors span the u and v coordinate axis by definition. However, the texture may be in any orientation on the plane and must be solved for algebraically.

The given texture coordinates can be rewritten in the following form:
$$p = s_1 t + t_1 b \qquad\qquad q = s_2 t + t_2 b$$
where:
$$s_1 = u_2 - u_1 \qquad\qquad t_1 = v_2 - v_1$$
$$s_2 = u_3 - u_1 \qquad\qquad t_2 = v_3 - v_1$$
In matrix form:

$$\begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} s_1 & t_1 \\ s_2 & t_2 \end{bmatrix} \begin{bmatrix} t \\ b \end{bmatrix}$$

By multiplying both sides by the inverse of the texture coordinate matrix we can solve for both vectors.

$$\begin{bmatrix} t \\ b \end{bmatrix} = \begin{bmatrix} p \\ q \end{bmatrix} \left(\frac{1}{s_1 t_2 - s_2 t_1} \right) \begin{bmatrix} t_2 & -t_1 \\ -s_2 & s_1 \end{bmatrix}$$

Or if you prefer: $\qquad t = \frac{pt_2 - qt_1}{s_1 t_2 - s_2 t_1} \qquad\qquad b = \frac{qs_1 - s_2 p}{s_1 t_2 - s_2 t_1}$

Now we have the three vectors that makeup the final rotation matrix:

$$\begin{bmatrix} t.x & t.y & t.z \\ n.x & n.y & n.z \\ b.x & b.y & b.z \end{bmatrix}$$