Estimating a Tangent Vector for Bump Mapping

David Eberly
Geometric Tools, LLC
http://www.geometrictools.com/

Created: November 10, 2003
Last Modified: March 1, 2008

Contents

1 Discussion

This brief document describes the estimation of a tangent vector for bump mapping in the class `BumpMap`, function `ComputeLightVectors`.

Consider a triangle with vertices $\mathbf{P}_0$, $\mathbf{P}_1$, and $\mathbf{P}_2$ and with corresponding texture coordinates (u_0, v_0), (u_1, v_1), and (u_2, v_2). Any point on the triangle may be represented as

$$\mathbf{P}(s,t) = \mathbf{P}_0 + s(\mathbf{P}_1 - \mathbf{P}_0) + t(\mathbf{P}_2 - \mathbf{P}_0)$$

where $s \geq 0$, $t \geq 0$, and $s+t \leq 1$. The texture coordinate corresponding to this point is similarly represented as

$$\begin{aligned} (u(s,t), v(s,t)) &= (u_0, v_0) + s((u_1, v_1) - (u_0, v_0)) + t((u_2, v_2) - (u_0, v_0)) \\ &= (u_0, v_0) + s(u_1 - u_0, v_1 - v_0) + t(u_2 - u_0, v_2 - v_0) \end{aligned}$$

Abstractly we have a surface defined by $\mathbf{P}(s,t)$ where s and t depend implicitly on two other parameters u and v. The problem is to estimate a tangent vector relative to u or v. We will estimate with respect to u, a process that involves computing the rate of change of $\mathbf{P}$ as u varies, namely the partial derivative $\partial \mathbf{P}/\partial u$.

Using the chain rule from calculus,

$$\frac{\partial \mathbf{P}}{\partial u} = \frac{\partial \mathbf{P}}{\partial s}\frac{\partial s}{\partial u} + \frac{\partial \mathbf{P}}{\partial t}\frac{\partial t}{\partial u} = (\mathbf{P}_1 - \mathbf{P}_0)\frac{\partial s}{\partial u} + (\mathbf{P}_2 - \mathbf{P}_0)\frac{\partial t}{\partial u}$$

Now we need to compute the partial derivatives of s and t with respect to u. The equation that relates s and t to u and v is written as a system of two linear equations in two unknowns

$$\begin{bmatrix} u_1 - u_0 & u_2 - u_0 \\ v_1 - v_0 & v_2 - v_0 \end{bmatrix} \begin{bmatrix} s \\ t \end{bmatrix} = \begin{bmatrix} u - u_0 \\ v - v_0 \end{bmatrix}$$

Inverting this leads to

$$\begin{bmatrix} s \\ t \end{bmatrix} = \frac{1}{(u_1 - u_0)(v_2 - v_0) - (u_2 - u_0)(v_1 - v_0)} \begin{bmatrix} v_2 - v_0 & -(u_2 - u_0) \\ -(v_1 - v_0) & u_1 - u_0 \end{bmatrix} \begin{bmatrix} u - u_0 \\ v - v_0 \end{bmatrix}$$

Computing the partial derivative with respect to u produces

$$\begin{bmatrix} \partial s/\partial u \\ \partial t/\partial u \end{bmatrix} = \tfrac{1}{(u_1-u_0)(v_2-v_0)-(u_2-u_0)(v_1-v_0)} \begin{bmatrix} v_2 - v_0 & -(u_2 - u_0) \\ -(v_1 - v_0) & u_1 - u_0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$= \tfrac{1}{(u_1-u_0)(v_2-v_0)-(u_2-u_0)(v_1-v_0)} \begin{bmatrix} v_2 - v_0 \\ -(v_1 - v_0) \end{bmatrix}$$

Combining this into the partial derivative for $\mathbf{P}$, we have

$$\frac{\partial \mathbf{P}}{\partial u} = \frac{(v_2 - v_0)(\mathbf{P}_1 - \mathbf{P}_0) - (v_1 - v_0)(\mathbf{P}_2 - \mathbf{P}_0)}{(u_1 - u_0)(v_2 - v_0) - (u_2 - u_0)(v_1 - v_0)} = \frac{(v_1 - v_0)(\mathbf{P}_2 - \mathbf{P}_0) - (v_2 - v_0)(\mathbf{P}_1 - \mathbf{P}_0)}{(v_1 - v_0)(u_2 - u_0) - (v_2 - v_0)(u_1 - u_0)}$$

which is the equation for `kTangent` in the source code.