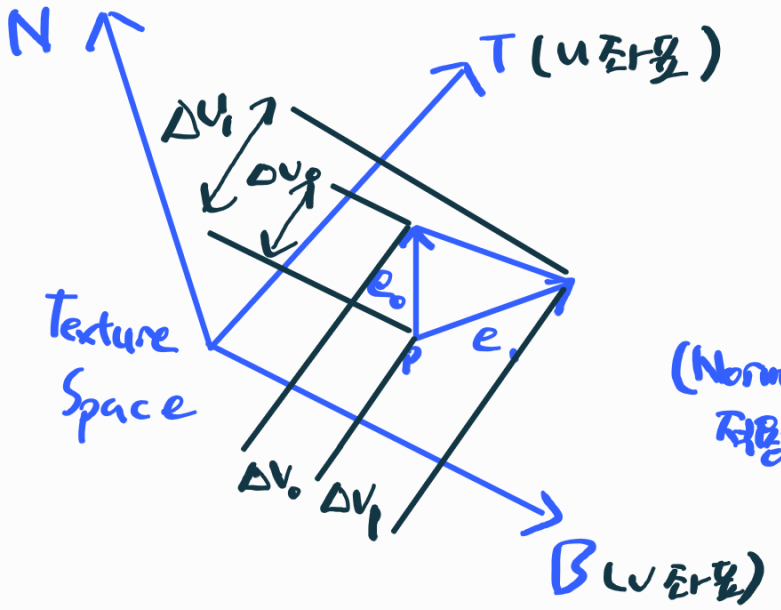
Normal Mapping을 사용하기 위해서 Tangent Vector가 필요하다

임의의 삼각형 Mesh로 부터 Tangent Vector 구하는 법



ΔU_0 = P의 U좌표와 e_0의 U좌표 차이

ΔU_1 = P의 U좌표와 e_1의 U좌표 차이

ΔV_0, ΔV_1 = V좌표의 차이

(Normal Map을 3차원으로 변환해 오브젝에 적용할 수 있게 해줌-)

이 삼각형 Mesh는 3차원 공간에 있는 삼각형이므로 각각의 Vertex들은 3차원 공간상에서의 좌표를 가지고 있고 Texture 좌표(UV)를 가지고있다.

그래서 삼각형의 Vertex가 가진 Texture 좌표로 Texture 좌표계에 그릴 수 있다-

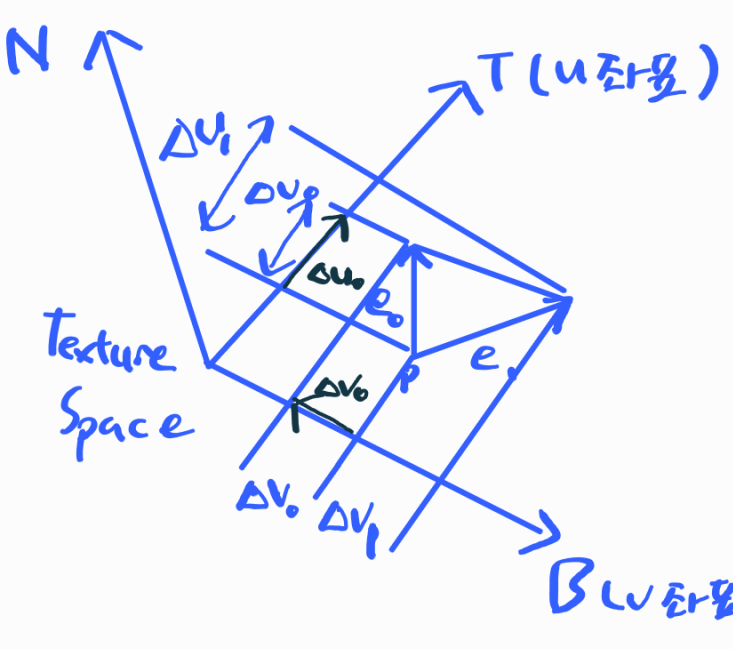
(3차원 벡터)

$e_0 = V_1 - V_0$

$e_1 = V_2 - V_0$

, e 는 3차원 공간상의 Vertex들의 차이값부터 계산되는 벡터다.

구하고자하는건 Tangent Vector와 Bitangent Vector다-.



$e_0 = \Delta U_0 T + \Delta V_0 B$, ΔV_0 = 음수

$e_1 = \Delta U_1 T + \Delta V_1 B$

(3차원 계산 결과)

$$\begin{bmatrix} e_{0,x} & e_{0,y} & e_{0,z} \\ e_{1,x} & e_{1,y} & e_{1,z} \end{bmatrix}$$

행렬로 표현

(정점 벡터)

$$e_0 = v_1 - v_0$$
$$e_1 = v_2 - v_0$$

정점과 같은 것

$$\begin{bmatrix} e_{0,x} & e_{0,y} & e_{0,z} \\ e_{1,x} & e_{1,y} & e_{1,z} \end{bmatrix} = \begin{bmatrix} \Delta u_0 & \Delta v_0 \\ \Delta u_1 & \Delta v_1 \end{bmatrix} \begin{bmatrix} T_x & T_y & T_z \\ B_x & B_y & B_z \end{bmatrix}$$

(vertex의 좌표 변화량)　(texture좌표)　(?)

Tangent Vector 와 Bitangent Vector를 역행렬로 구할 수 있다.

$$\begin{bmatrix} T_x & T_y & T_z \\ B_x & B_y & B_z \end{bmatrix} = \begin{bmatrix} \Delta u_0 & \Delta v_0 \\ \Delta u_1 & \Delta v_1 \end{bmatrix}^{-1} \begin{bmatrix} e_{0,x} & e_{0,y} & e_{0,z} \\ e_{1,x} & e_{1,y} & e_{1,z} \end{bmatrix}$$

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad A^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

DirectX Mesh Library에서 ComputeTangentFrame()을 사용하면 구할 수 있다.