Projective texture mapping - Interpolació en espai projectiu

October 24, 2013

Quan s'utilitza projective texture mapping, cal interpolar les coordenades de textura en espai projectiu, o el que és el mateix, fer la divisió de perspectiva **després** d'interpolar les coordenades, no abans.

Per simplificar aquest exemple, farem aquestes suposicions:

1. En comptes de generar coordenades de textura (s,t) dins l'interval [0,1], generarem unes coordenades de textura que anomenarem (u,v) dins l'interval [-1,1]. No hi ha cap pèrdua de generalitat perquè (s,t) = 0.5(u,v)+(0.5,0.5).
2. No hi ha transformació de modelat.
3. El projector de diapositives està a l'origen de coordenades, mirant en direcció de les Z negatives. Per tant, la viewing transform del projector és la identitat.

Projecció d'un punt P arbitrari Sigui P un punt qualsevol de l'escena. Per calcular les coordenades de textura de P, que anomenarem P_u i P_v, cal projectar el punt P d'acord amb les matrius del projector. Tenint en compte les suposicions anteriors, els passos per calcular P_u són (el cas de P_v és perfectament simètric):

1. $P^{clip} = M_{proj}P$, on M_{proj} és la matriu de projecció associada al projector; hem assumit que no hi ha modeling transform i que la viewing transform del projector és la identitat.
2. $P^{ndc} = P^{clip}/P_w^{clip}$ (divisió de perspectiva)
3. $P_u = P_x^{ndc}$; l'escalat i la translació només caldrien per calcular (s,t), no pas per (u,v)

Observeu que no hi ha cap interpolació.

Anem a expressar P_u en funció de les coordenades originals de P. Per simplificar, assumirem que M_{proj} és la matriu que genera gluPerspective. Suposant un aspect ratio = 1, M_{proj} és de la forma

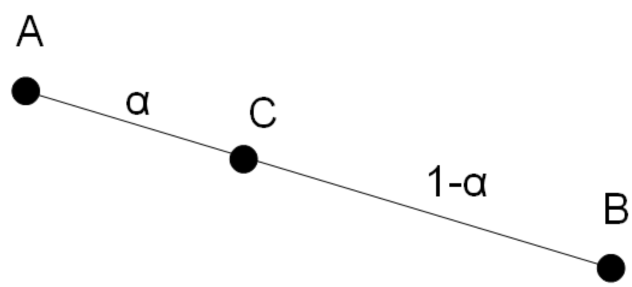
$$M_{proj} = \begin{bmatrix} k & 0 & 0 & 0 \\ 0 & k & 0 & 0 \\ 0 & 0 & ? & ? \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

on $k = cot\frac{fov}{2}$ és una constant que depèn del fov del projector, i els elements amb "?" no son rellevants per aquest exemple.

Per tant, la coordenada P_u d'un punt P arbitrari es pot calcular, en aquest cas, com:

1. $P^{clip} = M_{proj}P = (kP_x, kP_y, ?, -P_z)$
2. $P^{ndc} = (-k\frac{P_x}{P_z}, -k\frac{P_y}{P_z}, ?)$ (divisió de perspectiva)
3. $P_u = -k\frac{P_x}{P_z}$

Imaginem ara un segment de determinat pels seus vèrtexs A i B, i un punt arbitrari C dins el segment definit per A i B. Volem calcular les coordenades de textura al punt C.



Primera opció: projectar directament el punt C Una forma de calcular la coordenada u que li correspon a C és

$$C_u = -k\frac{C_x}{C_z} \tag{1}$$

Aquesta fòrmula és perfectament implementable en un FS, fent servir un varying per que al FS li arrivin les coordenades del punt.

Interpolació Si les distàncies relatives de C als punts A i B són α i $(1-\alpha)$, llavors $C = (1-\alpha)A + \alpha B$ (interpolació lineal entre A i B), o si ho preferiu, $C = A + \alpha(B - A)$, i per tant, una forma alternativa i perfectament correcta de calcular C_u és, aplicant directament l'Eqüació 1,

$$C_u = -k\frac{(1-\alpha)A_x + \alpha B_x}{(1-\alpha)A_z + \alpha B_z} \tag{2}$$

Aquesta darrera eqüació té l'avantatge d'expressar C_u en funció de A, B, i α.

Segona opció: interpolar (u,v) després de la div de perspectiva (incorrecta!) La interpolació lineal de u_Ai u_B, que anomenarem $u^{'}$dóna un resultat diferent a C_u:

$$u^{'} = (1-\alpha)A_u+\alpha B_u = (1-\alpha)(-k\frac{A_x}{A_z})+\alpha(-k\frac{B_x}{B_z}) = -k\frac{(1-\alpha)A_xB_z+\alpha B_xA_z}{A_zB_z} \tag{3}$$

Com a exercici, podeu comprovar que C_u / $u^{'}$ és en general diferent de 1; altrament, podeu provar per exemple amb $A = (1,1,1)$, $B = (2,2,10)$, i $C = 0.5(A+B)$ i veure que surten valors diferents.

La responsable és la divisió de perspectiva del projector, que fa que les coordenades de textura (u,v) no es s'hagin d'interpolar linialment. Per tant, l'eqüació 3 no permet calcular correctament les coordenades de textura.

Tercera opció: interpolar en espai projectiu (correcte) Anem a demostrar que el següent procediment per calcular P_u és vàlid. Observeu que la divisió de perspectiva la farem al final, després d'interpolar les coordenades de textura, la qual cosa implica treballar amb coordenades de textura homogènies (u,v,q):

1. $A_u = kA_x$, $A_q = -A_z$ (igual que abans però hem tret la divisió de perspectiva)
2. $B_u = kB_x$, $B_q = -B_z$
3. Ara interpolem linealment la u, $(1-\alpha)A_u + \alpha B_u = (1-\alpha)kA_x + \alpha kB_x$,
4. Interpolem linealment la q, $(1-\alpha)A_q + \alpha B_q$= -($(1-\alpha)A_z + \alpha B_z$)
5. Finalment, fem la divisió de perspectiva, dividint la u per la q: $u_C = -k\frac{(1-\alpha)A_x+\alpha B_x}{(1-\alpha)A_z+\alpha B_z}$

Observeu que hem arribat al mateix resultat de l'Eqüació 2, per tant el procediment és vàlid. A banda de ser eficient, aquesta tercera opció està soportada pel pipeline fix i per tant és usable sense shaders.