Graphics algorithms

Perspective matrix for horizontal fov $f \in (0, \pi)$, aspect ratio $a = \frac{w}{h}$, near plane n (assuming x is right, y is up, z is forward) is given by

$$\begin{bmatrix} \frac{1}{\tan(f/2)} & 0 & 0 & 0 \\ 0 & \frac{a}{\tan(f/2)} & 0 & 0 \\ 0 & 0 & 0 & n \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

describing the function

$$(x, y, z, 1) \mapsto \left(\frac{x}{\tan(f/2)}, \frac{ay}{\tan(f/2)}, n, z \right)$$

According to this, in clip space, we clip to $-w \leq x \leq w, -w \leq y \leq w, 0 \leq z \leq w, w > 0$. In terms of view-space coordinates, this is equivalent to clipping to

$$-z \tan(f/2) \leq x \leq z \tan(f/2)$$

$$-\frac{z \tan(f/2)}{a} \leq y \leq \frac{z \tan(f/2)}{a}$$

$$0 \leq n \leq z$$

$$z > 0$$

Depth clip simply disables view $0 \leq n \leq z$, but $z > 0$ is still enforced. Note that if depth clip is disabled, then depth is **clamped to the viewport** before any depth testing takes place.

Depth is then given by $\frac{n}{z}$

Now, let $a, b, c \in \mathbb{R}$. Ortho projection is given by

$$\begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

describing the function

$$(x, y, z, 1) \mapsto (ax, by, cz, 1)$$

so in view space, we clip to

$$-1 \leq ax \leq 1$$

$$-1 \leq by \leq 1$$

$$0 \leq cz \leq 1 \text{ (but this one can be disabled with no depth clip)}$$

Screen space ambient occlusion

The technique is to use the depth buffer as an approximation: if a point is inside geometry (note: watertight meshes must be used, i.e. ones with a clear inside and outside), then depth of that point is > final depth value (nearest geometry), but the converse is not true.

To compute the number of additional pixels needed to ensure proper screen edge rendering, divide the ssao radius by the side length of a single pixel at the near plane, a.k.a. $\frac{n \tan(f/2)}{\text{screen width } /2}$ (take ceiling of that).

Deferred rendering

you should store the final color instead of uvs so that the correct mip level is selected.