Conservative Z-Prepass for Frustum-Traced Irregular Z-Buffers

Yusuke Tokuyoshi and Tomohiro Mizokuchi (SQUARE ENIX CO., LTD.)

1. Introduction

Frustum traced irregular z-buffers (IZBs) [Wyman et al. 2016] are used to render accurate hard shadows for real-time applications such as video games, while they are expensive compared to shadow mapping. To improve the performance, we use a two-pass visibility test by integrating a conservative shadow map [Hertel et al. 2009] into the pipeline of the IZB. This poster also presents a more precise implementation of the conservative shadow map than the previous method.

2. Our Hard Shadow Pipeline

Fully shadowed shading points are found using a conservative shadow map, and they are culled before IZB creation.

3. Implementation of Conservative Shadow Mapping

Conservative shadow maps store a biased depth in each texel fully covered by a triangle. This depth is more distant than the blue quadrilateral created by the triangle and texel to give the fully shadowed volume (orange).

void main(float4 p : SV_Position, float2 c : BARYCENTRICS) {
    float2 dx = ddx(c) * 0.5;
    float2 dy = ddy(c) * 0.5;
    float2 a = dx + dy;
    float2 b = dx - dy;
    if(c.x < max(abs(a.x), abs(b.x)) || c.y < max(abs(a.y), abs(b.y)) ||
       1.0 - c.x - c.y < max(abs(a.x + a.y), abs(b.x + b.y))) {
        discard;
    }
}

Our pixel shader for conservative shadow maps (HLSL)

References

4. Future Work

Small triangles Since the conservative shadow map is not efficient for triangles smaller than the texel, only large triangles should be drawn to reduce the overhead.

SV_Barycentrics vs. SV_InnerCoverage A fully covered texel can also be detected using SV_InnerCoverage, though it has to use expensive conservative rasterization unlike our implementation. We would like to evaluate them when they are available in the future.