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

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

Two-pass visibility test for hard shadows. The visibility is first roughly tested (a) using a conservative shadow map, and then an accurate visibility test using an irregular z-buffer (IZB) is performed only for the remaining shading points (b). Using this pipeline, the shadow performance is **improved from 8.52 ms to 3.59 ms** for the 4K screen resolution.

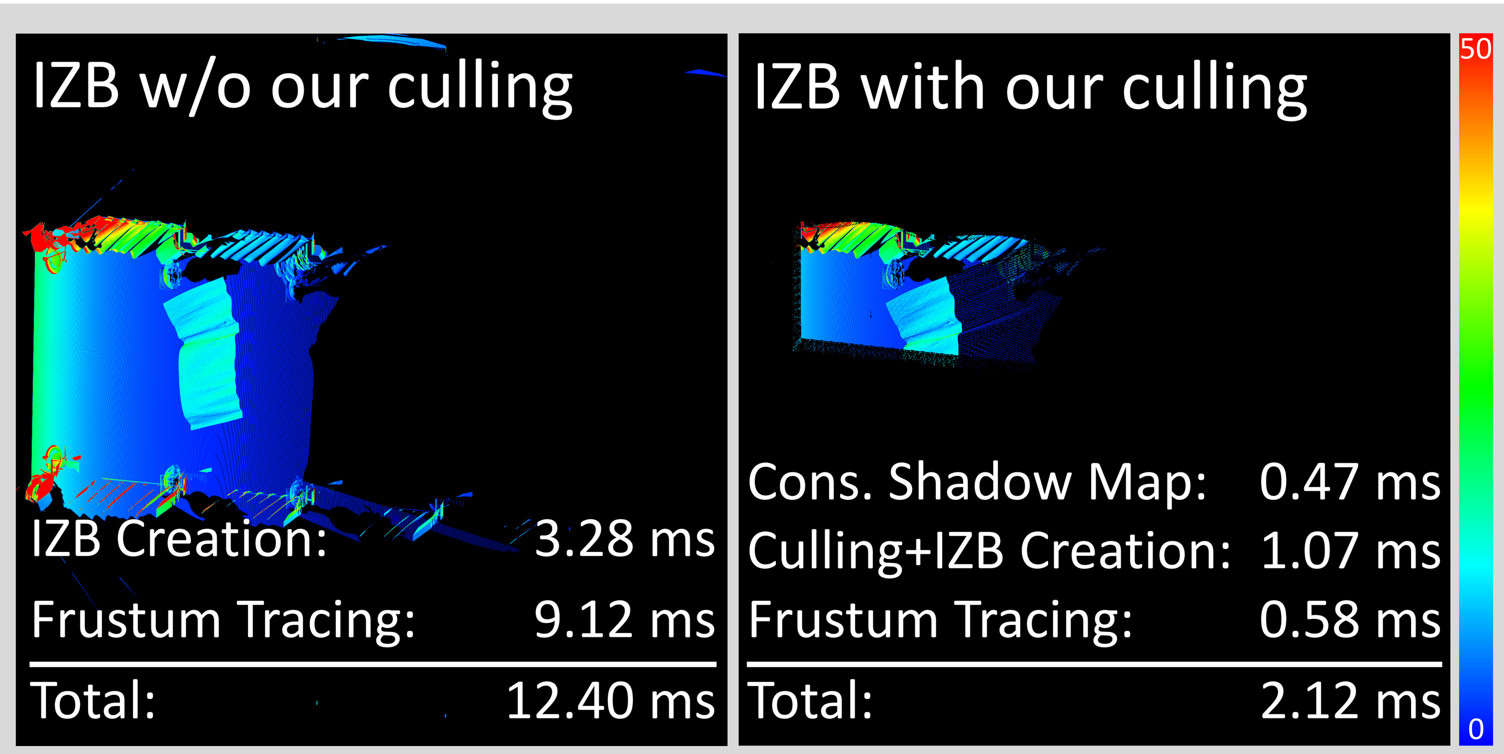
1. Introduction

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

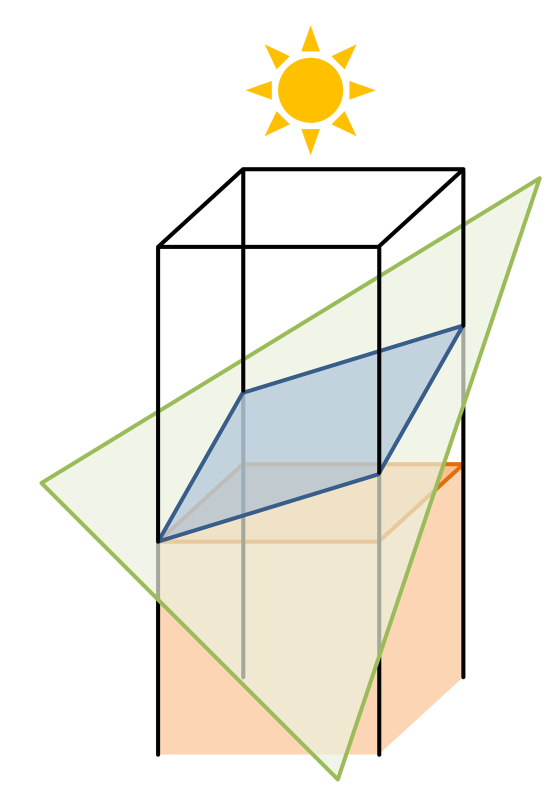
Fully shadowed shading points are found using a conservative shadow map, and they are culled before IZB creation.

Conservative Shadow Map
↓
Culling (1st visibility test)
↓
IZB Creation
↓
Frustum Tracing
(2nd visibility test)



# of shading points in each IZB texel. Our two-pass visibility test is especially effective when many shading points are occluded by multiple triangles.

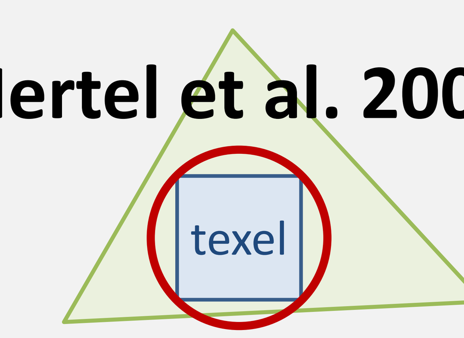
3. Implementation of Conservative Shadow Mapping



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).

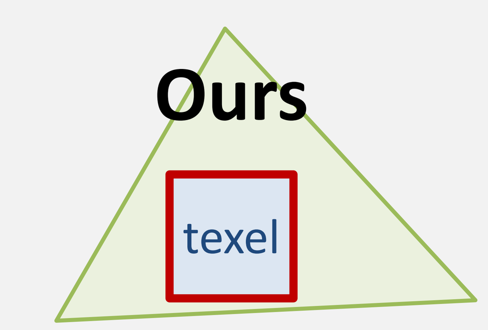
Detection of the fully covered texel

[Hertel et al. 2009]



Rough test using circumscribed circle of the texel

Ours



Strict test whose performance is almost the same as Hertel et al. for a distant light

Our pixel shader for conservative shadow maps (HLSL)

```
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;
  }
}
```

This will be replaced with *SV_Barycentrics* for HLSL 6.1

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.

References

S. Hertel, K. Hormann, and R. Westermann. 2009. A Hybrid GPU Rendering Pipeline for Alias-Free Hard Shadows. In EG '09 Areas Papers.
C. Wyman, R. Hoetzlein, and A. Lefohn. 2016. Frustum-Traced Irregular Z-Buffers: Fast, Sub-Pixel Accurate Hard Shadows. IEEE TVCG. 22, 10, 2249–2261.