Frustum-Traced Irregular Z-Buffers: Fast, Sub-pixel Accurate Hard Shadows

Chris Wyman* Rama Hoetzlein Aaron Lefohn
NVIDIA

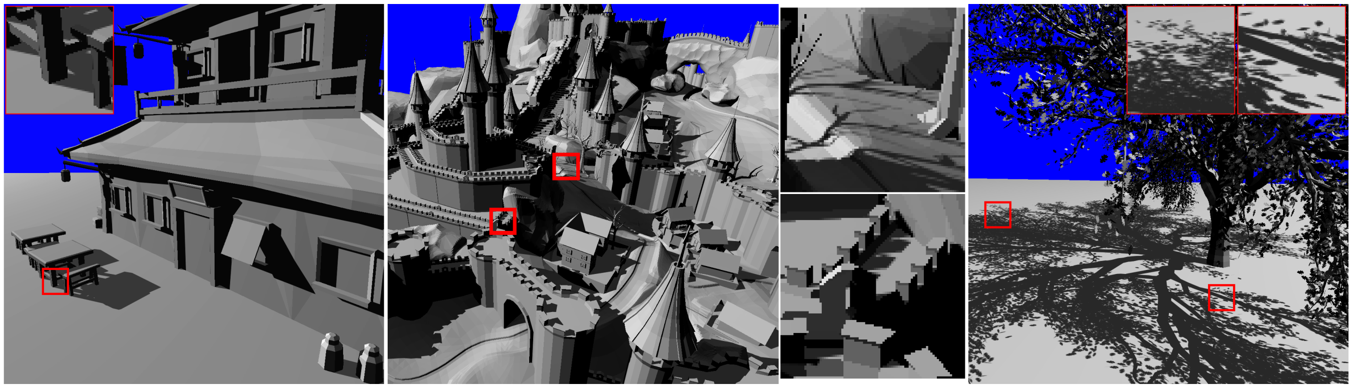

Figure 1: *Our 32 sample per pixel hard shadows on (left) a simple house model (3.2 ms), a more complex citadel scene (10.0 ms), and a 2.4 million polygon tree with alpha-mapped leaves (20.8 ms).*

We present a system to render real-time, antialised hard shadows using irregular z-buffers. We achieve 32 sample, subpixel accuracy at roughly twice the cost of a single visibility sample. Our system remains interactive on complex CAD models and modern game assets while running at 1080p and above, yet imposes no constraints on light, camera, or geometry allowing fully dynamic scenes without precomputation. We introduce no spatial or temporal aliasing, smoothly animating even subpixel shadows from grass or wires.

1 Introduction

Irregular z-buffers (IZBs) generate pixel accurate hard shadows by inserting visible fragments into a light-space A-buffer [Johnson et al. 2005]. Rasterizing geometry over this light-space buffer then tests shadow occlusion at exactly the points needed – those visible on screen. Unfortunately, this process creates irregular workloads as each texel in the light-space A-buffer contains an arbitrary number of samples. Prior research suggested this irregularity led to poor GPU utilization and extremely variable performance (over 100:1 variation between frames), leading many to discard this algorithm.

2 Revisiting Irregular Z-Buffers

Taking a fresh, principled look at hard shadow rendering, we reconverged on irregular z-buffering as the best raster-based approach for alias-free hard shadows. This fresh look brought new insights enabling us to begin addressing IZB's irregular workloads.

Shadow Map Duality. We observed shadow map aliasing occurs at exactly those locations where IZBs exhibit performance problems. Thus, algorithms addressing shadow map *aliasing* (e.g., perspective or cascaded maps) improve irregular z-buffer *performance*.

Culling. Prior IZB prototypes ran in GPU compute kernels, ignoring most of the culling provided in the graphics pipeline. Because rasterization relies on good culling to achieve logarithmic complexity, this led to poor scaling. We demonstrate IZBs can leverage the graphics pipe to regain some advantages of hardware culling.

Early-Outs. Rebuilding an irregular z-buffer each frame provides options to destructively traverse the data structure, removing pixels as they become fully shadowed. This enables an equivalent of "first hit" ray queries (rather than computing a "closest hit" occluder).

Hardware Advances. Recent GPU additions, like hardware conservative rasterization and faster per-primitive geometry shaders reduce overheads in the graphics pipe, providing up to a 3× speedup.

Further Improvements. Beyond our prior work [Wyman et al. 2015] we support alpha-mapped polygons, full hardware acceleration on 32 spp shadows, and other overhead reductions.

Additional Optimizations. Our system includes numerous smaller improvements, each contributing 5–15% speedups. These range from optimizing memory layouts to significantly reducing atomic operations during data structure construction and traversal.

2.1 Results

Our system uses OpenGL, with timings provided on a GeForce GTX 980. Some optimizations run today only on GM20x-class GPUs, though we also have vendor-agnostic implementations.

Scene (poly count)	**32 spp shadow cost**		**1 spp shadow cost**	
Shadow cost per frame *All at 1920×1080*	Full HW accel	Partial HW accel	Full HW accel	Partial HW accel
House (7 k)	3.2 ms	3.2 ms	1.5 ms	1.6 ms
Villa (89 k)	4.6 ms	5.7 ms	2.6 ms	3.1 ms
Citadel (613 k)	9.1 ms	13.5 ms	5.5 ms	8.7 ms
Hairball (2.9 M)	20.3 ms	42.6 ms	10.2 ms	26.9 ms
Tentacles (3.8 M)	15.2 ms	40.9 ms	11.5 ms	32.0 ms
Powerplant (12.3 M)	39.4 ms	120.0 ms	25.5 ms	92.7 ms

*e-mail:chris.wyman@acm.org

SIGGRAPH 2015 Talks, August 09 – 13, 2015, Los Angeles, CA.
ACM 978-1-4503-3636-9/15/08.
http://dx.doi.org/10.1145/2775280.2792511

References

JOHNSON, G. S., LEE, J., BURNS, C. A., AND MARK, W. R. 2005. The irregular z-buffer: Hardware acceleration for irregular data structures. *ACM Trans. Graph. 24*, 4 (Oct.), 1462–1482.

WYMAN, C., HOETZLEIN, R., AND LEFOHN, A. 2015. Frustum-traced raster shadows: Revisiting irregular z-buffers. In *Symp. Interactive 3D Graphics and Games*, 15–23.