Sub-Pixel Shadow Mapping

Pascal Lecocq   Pascal Gautron   Jean-Eudes Marvie   Gael Sourimant
Technicolor *

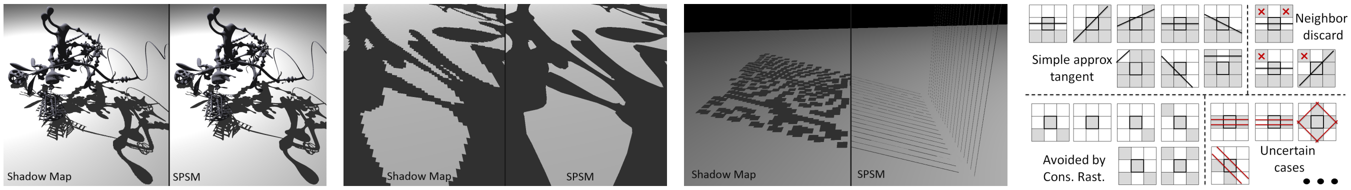

Figure 1: *Our technique provides sub-pixel accuracy to shadow maps, avoiding aliasing artifacts while preserving the rendering speed (left). We reconstruct shadow edges by combining an estimate of the shadow tangent (right) with partial geometry information.*

The limited resolution of shadow maps may result in erroneous shadowing, yielding artificially jagged edges (Figure 1) and temporally crawling shadows even using perspective optimization techniques. Dai *et al.* [2008] propose an explicit storage of geometry within shadow map texels to avoid aliasing. Each texel stores the coordinates of the closest triangle only, potentially leading to false negatives in the intersection computation while incurring large memory consumption. These artifacts are reduced by intersecting the triangles stored in numerous neighboring texels, resulting in significant performance hit while still missing some intersections.

We introduce *Sub-Pixel Shadow Maps* (SPSM) for real-time shadow mapping with sub-pixel precision. Our technique is based on the storage of a fixed-size partial representation of the scene geometry using conservative rasterization, combined with an original reconstruction of shadow edges.

The Sub-Pixel Shadow Map

Storing triangle information within the shadow map introduces two main challenges. First, most hardware rasterizers generate fragments only if the center of the fragment is covered by the considered triangle. A geometry-based shadow map [2008] may thus lack information in the texels corresponding to shadow edges, where precision is crucial. The second challenge is the presence of several triangles rasterized at a same texel position. Ideally, each texel should store an arbitrarily large list of overlapping triangles.

We first enforce the generation of fragments for the entire coverage of the triangles using Conservative Rasterization [Hasselgren et al. 2005]. We then generate the SPSM by rendering the scene from the light viewpoint, storing vertex information for the closest triangle only. We pack the coordinates in SPSM space within 128-bit using a novel compression scheme for compact storage of depth derivatives (Figure 2).

At the rendering stage we project each visible point $\mathbf{p}$ in light space and determine whether $\mathbf{p}$ lies on the triangle stored in the corresponding SPSM texel. If so, conservative rasterization guarantees that $\mathbf{p}$ is not occluded. Fully lit zones are then rendered using a single SPSM lookup. Otherwise, we intersect the light ray from $\mathbf{p}$ with the triangle. While finding an occlusion ensures shadowing, the lack of intersection may be due to missing geometry. We then intersect the light ray with the triangles stored in the 3×3 neighborhood of the texel. If needed, we compensate for occlusion incertainty by analytically reconstructing the shadow edge.

Even though our representation is partial, an accurate shadow edge can be obtained by linking edge vertices. Considering the result of depth comparisons for the point $\mathbf{p}$ in the 3×3 neighborhood, we first deduce an approximate light-space tangent to the shadow edge (Figure 1). We then project the vertices whose bisector faces towards the approximate tangent onto that tangent, and determine their projection distance. Vertices occluded by projected edges are discarded to avoid reconstructing an erroneous edge. Comparing the projection distance $d_{\mathbf{p}}$of $\mathbf{p}$ on the tangent with the projection distances of neighboring vertices determines whether $\mathbf{p}$ is lit or shadowed (Figure 2).

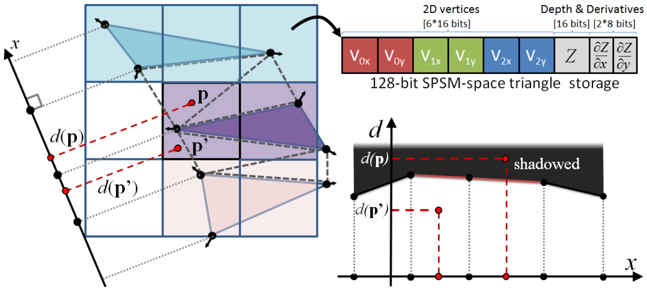
x
d(p)
d(p')
p
p'
2D vertices
[6*16 bits]
Depth & Derivatives
[16 bits] [2*8 bits]
V_{0x} V_{0y} V_{1x} V_{1y} V_{2x} V_{2y} Z $\frac{\partial Z}{\partial x}$ $\frac{\partial Z}{\partial y}$
128-bit SPSM-space triangle storage
d
d(p)
shadowed
d(p')
x

Figure 2: *If point* $\mathbf{p}$ *is not covered by the triangles stored within a close neighborhood (left), their vertices are used to deduce an accurate shadow edge (right).*

Results

Many techniques reduce aliasing and bias issues by projection optimization, each introducing typical artifacts. To avoid confusion we compare SPSM with classical uniform shadow maps, although our approach can also benefit from these optimizations.

We implemented SPSM on a nVidia GeForce GTX580. Comparing 512^2 uniform and sub-pixel shadow maps, our technique effectively avoids aliasing while introducing an average computational overhead of only 14%. The storage of the SPSM uses a single 128-bit RGBA buffer, yielding a total memory occupancy of 4MB.

While bringing significant quality improvements, the storage of a single triangle per texel may result in occluder fusion during edge reconstruction. Future work will consider heuristics for enhanced edges, typically using a combination of conservative and non-conservative rasterization.

Sub-Pixel Shadow Maps are a simple and inexpensive alternative to classical shadow mapping, bringing high quality shadowing capabilities into highly demanding applications such as video games, in which neither image quality nor speed could be compromised.

References

DAI, Q., YANG, B., AND FENG, J. 2008. Reconstructable geometry shadow maps. In *Proceedings of I3D*, 4:1–4:1.

HASSELGREN, J., AKENINE-MÖLLER, T., AND OHLSSON, L. 2005. Conservative rasterization. In *GPU Gems 2*, M. Pharr, Ed. Addison-Wesley, 677–690.

* {pascal.lecocq, pascal.gautron, jean-eudes.marvie, gael.sourimant}@technicolor.com

SIGGRAPH 2013, July 21 – 25, 2013, Anaheim, California.
2013 Copyright held by the Owner/Author.
ACM 978-1-4503-2261-4/13/07