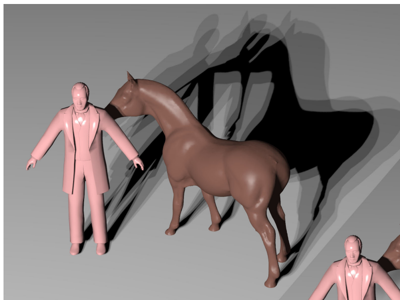
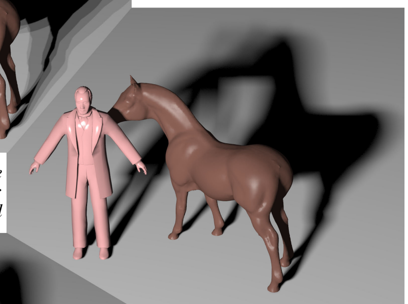
(Talk 0626) Pismo: Parallax-Interpolated Shadow Map Occlusion

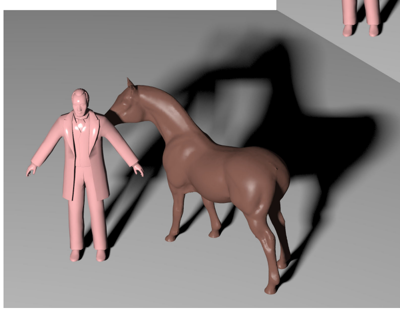
Ivan Neulander
Rhythm & Hues Studios



Four discrete shadows from the corners of an area light, rendered at 1600x1200 using 4 depth maps with PCF antialiasing: 65 sec on a 2133MHz Athlon.



Soft shadow rendered using the same 4 depth maps with Pismo: 323 sec, 128 shadow rays/pixel



Ground truth—soft shadow rendered using a ray tracer: 1732 sec, 128 shadow rays/pixel

Introduction and Motivation

Soft shadows cast by area lights are essential to realistic image synthesis. They are commonly rendered using a ray tracer to perform multiple occlusion tests between a given shading point and a set of sample positions on the light. While this method is reliably accurate, it is often expensive in terms of time and memory.

A potentially less expensive approach is to approximate the soft shadow with a discrete set of hard shadows, each evaluated using a depth map. An example of this is shown in the top image above. In practice this method requires an excessively high number of depth maps to produce an adequately smooth penumbra. Percentage-closer filtering (PCF) can be used to smooth the penumbra by uniformly blurring each hard shadow, but this also blurs detail that should remain crisp, such as contact shadows.

Pismo is a technique developed at Rhythm & Hues for rendering film-quality soft shadows, as shown in the middle image above, using a sparse set of depth maps created from points on an area light. We can test any shadow ray quickly, using only 3 depth map lookups plus a low amortized overhead. Moreover, Pismo is designed to take full advantage of all available depth maps, smoothly converging to an accurate solution as their number is increased.

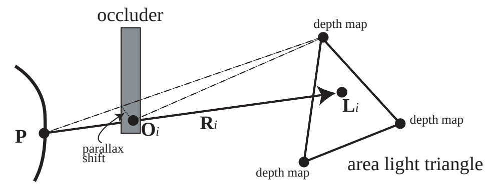
Representation

We represent the area light as a triangle mesh, with a depth map rendered from each vertex. The depth maps are rendered using perspective cameras, all fixated on a common point that is central to the shadow-casting geometry.

Our depth maps are augmented slightly from their traditional definition: they store multiple ordered depth values per pixel, one for each distinct surface covering the pixel, with some proximity-based pruning. The multiple depths are needed to determine the distance from a shadowed point to its nearest occluder, which is useful for short-range shadowing. More traditional, single-channel depth maps (contaning *maximum* depths, however) can also be used with Pismo, but the resulting soft shadows are accurate only outside the casting geometry's convex hull, which precludes good self-shadowing.

Implementation Overview

For any shading point $\mathbf{P}$, we compute fractional area light occlusion by sampling multiple shadow rays. Each shadow ray $\mathbf{R}_i$ extends from $\mathbf{P}$ to a unique point $\mathbf{L}_i$ on one of the area light triangles. The vertices of this triangle contain the 3 corner depth maps we will interpolate to estimate the occlusion of $\mathbf{R}_i$. We use the barycentric coordinates of $\mathbf{L}_i$ with respect to its triangle as linear interpolation weights.



We begin by estimating the distance along $\mathbf{R}_i$ from $\mathbf{P}$ to the nearest *likely* occluder. For this we use a barycentric blend of nearest occluder depths for $\mathbf{P}$ from the 3 corner depth maps. Although this computation requires depth map evaluations, they occur only once per shading point per depth map and are subsequently cached, so their cost is amortized over all shadow rays. The depth maps used at this stage have been bled out to fill empty pixels with nearest nonempty neighbors—a simple, one-time preprocess.

Next, we displace $\mathbf{L}_i$ toward $\mathbf{P}$ (along $-\mathbf{R}_i$) by the above occlusion distance to yield a putative occluder point $\mathbf{O}_i$. Finally, we estimate the occlusion along $\mathbf{R}_i$ by barycentrically blending the occlusion of $\mathbf{P}$ from each corner depth map. But instead of transforming $\mathbf{P}$ into each depth map's screen space, we transform $\mathbf{O}_i$. This alternate projection amounts to a parallax shift of the putative occluding surface from each depth map's original vantage point to that of $\mathbf{L}_i$.

Summary and Conclusions

Pismo samples each shadow ray $\mathbf{R}_i$ by making two estimations of occluder depth along it. The first is a rough approximation based on querying the 3 vertex depth maps at $\mathbf{P}$ (a calculation that is amortized over many shadow rays). The second is a refined approximation that uses the first to account for the parallax shift between each depth map and $\mathbf{L}_i$.

The accuracy of both estimations, and therefore, the quality of Pismo's shadows, depends on the density of depth maps per solid angle of area light and on the occluding geometry's surface complexity. Because both estimations are barycentrically blended, shadow rays hitting an area light triangle's vertices are always evaluated correctly and other shadow rays are evaluated with increasing accuracy as the size of the triangle diminishes. Unlike methods that simulate soft shadows through texture-space PCF, ours is immune to artifacts from distorted depth maps created via [Stamminger and Drettakis 2002].

A somewhat similar approach to ours is the image morphing method presented in [Chen and Williams 1998], which can build many high-quality interpolated depth maps from a sparse initial set. However, hundreds of such interpolated maps may need to be created, stored, and sampled in order to capture a smooth shadow. A hybrid approach that uses morphing to moderately increase the number of depth maps available to Pismo may leverage the best features of both techniques.

References

CHEN, S. E., AND WILLIAMS, L. 1998. View interpolation for image synthesis. 381–390.

STAMMINGER, M., AND DRETTAKIS, G. 2002. Perspective shadow maps. In *SIGGRAPH '02*, ACM, 557–562.