

Fast, Soft Reflections Using Radiance Caches

Apurva Shah Justin Ritter Chris King Stefan Gronsky
Pixar Animation Studios

1 Introduction

In Pixar’s *Ratatouille* a lot of scenes take place inside the kitchen where reflective surfaces like counter tops, stoves, pots and pans abound. Furthermore, these surfaces were often burnished or covered with dents, scratches or other displacements, which meant that the reflections were soft and fuzzy. Physically accurate reflections are most often achieved by tracing reflected rays into the scene. When the ray encounters another object, computationally expensive lighting and shading calculations must be performed to determine the contribution of the reflecting point. Paradoxically, surfaces that have soft or fuzzy reflections are more expensive since they have a larger reflection cone angle requiring more rays to adequately sample the reflected scene. We present a technique that utilizes radiance caches to significantly speed up the reflection calculations and discuss some of the accuracy trade-offs inherent to this approach.

2 Basic Approach

In a pre-pass the reflected objects are rendered without culling or hiding. The resulting radiance values are stored in a brickmap [CHR04] [CHR07], which we refer to as the radiance cache. Although it would seem that having the additional geometry would add significantly to the render time, in practice the grid based Reyes architecture [COO87] is fairly efficient even when the hiding and culling are disabled. In the main render pass, geometric rays are fired into the scene from the reflecting surface, however instead of shading at the end of the ray the intersecting points and normals are instead used to look up the radiance values from the cache.

The brickmap data is structured as a MIP map allowing for efficient filtered reads. For a reflection with cone angle (θ), projected hit distance on to the reflection vector (h) and tracing samples (s) the radiance is read with a filter kernel of:

$$r = \sqrt{h * \tan \theta / s}$$

For soft reflections θ is large which generally requires s to be large as well to provide enough stochastic point samples. However, since our radiance reads are filtered we need s to just be large enough to account for geometric discontinuities within the reflection cone and not texture variations which generally tend to be of a higher frequency and typically the driver for number of samples needed.

2.1 Fur and Hair

In *Ratatouille* we had the added challenge of reflecting fur and hair, which add an order of magnitude geometric complexity that the ray tracing has to deal with. The character grooms were built with a granularity control that could be set independently for camera and ray tracing visibility. Lowering the fur or hair granularity automatically thickened the curves to maintain the original volume. The radiance computation was done using the full granularity so that all the texture and illumination variations were captured in the cache. However, for reflection tracing the granularity was set much lower, often as low as 10%. Essentially we were ray tracing against coarser geometry but resolving reflection illumination and texture at a higher fidelity.

3 Discussion

In most of our tests the radiance cache reflections were considerably faster than the corresponding ray traced reflections in PRMan even after including the additional render time for the radiance cache. Figure 1 provides details for a simple scene. As expected the speedup is more pronounced for higher samples where the fixed radiance cache overhead is prorated and fewer samples are needed.

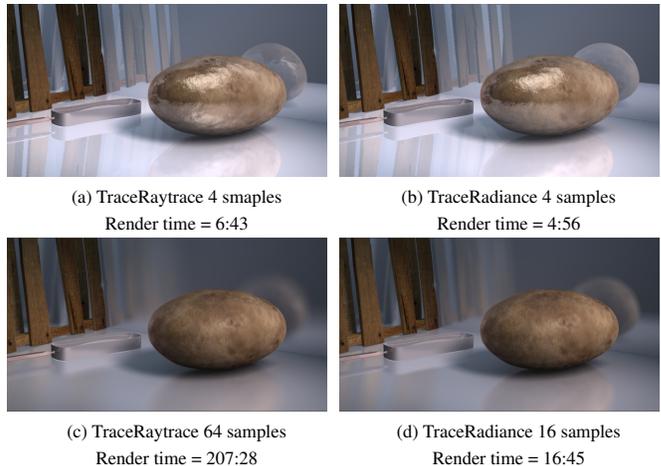


Figure 1: A simple scene that compares normal ray traced reflections to radiance cache reflections. Note that the 64 ray traced samples in (c) are visually equivalent to 16 radiance samples in (d). Radiance cache rendering time of 1:13 for (b) and (d) is included in the total render times. ©Disney/Pixar.

Naturally the biggest downside inherent to this technique is that the view dependent shading in the reflection is incorrect, for example the potato reflection on the wall. The lighters would often setup the radiance pass cameras to match the predominant reflection direction based on the primary reflecting and reflected objects. In our example the reflection of the potato on the table is a better match than the one on the wall. For soft reflections in a typical scene this error was generally imperceptible. However, in a few cases we did revert to traditional ray traced reflections when dealing with particularly sharp, mirror reflections.

It is also worth noting that the filter kernel of the brickmap covers a voxelized sphere rather than a disc, and even though the reads are normalized by volume, there can be errors in the radiance values for multiple, close together surfaces along the ray. Fortunately by increasing the number of samples we can approach the point sampled nature of traditional ray traced reflections when necessary.

References

- COO87 - Robert Cook, Loren Carpenter, Edwin Catmull, The Reyes image rendering architecture, Proceedings of ACM Siggraph 1987.
- CHR04 - Baking 3D Textures: Point Clouds and Brick Maps, PhotoRealistic RenderMan Application Note #39, Pixar 2004.
- CHR07 - Per Christensen, Point Clouds and Brick Maps for Movie Production. Chapter in Point-Based Graphics, Markus Gross and Hanspeter Pfister editors, Morgan Kaufmann Publishers, 2007.