

Adaptive Refinement Promises Faster Rendering on Graphics Chips

October 4 2004

Rendering realistic, computer-generated scenes can be a time-consuming exercise even on the most powerful computers. But computer scientists at the University of California, San Diego have developed a technique that accelerates certain graphics computations **by two or three orders of magnitude, at very little cost.** "It's really simple and requires only a few extra lines of code," says Craig Donner, a fourth-year computer science Ph.D. candidate in UCSD's Jacobs School of Engineering. **"It was just an idea that nobody had thought of or implemented."**

That idea applies adaptive refinement to graphics chips, and was presented in August at SIGGRAPH, the leading academic conference for computer graphics. In a technical sketch (pictured) and companion article, Donner and Computer Science and Engineering professor Henrik Wann Jensen detailed the technique for rendering a scene by tiling the screen and disregarding those tiles that are inactive. The result: significant speedups, depending on tile size and the algorithm used.

"It was three or four times faster as an overall speedup," explains Jensen, who is affiliated with the California Institute for Telecommunications and Information Technology [Cal-(IT)²]. "This technique provides significant speedups by exploiting computational similarity, coherence and locality."

Adaptive refinement is not new, but the UCSD team is the first to apply it to graphics processing units (GPUs), and specifically to the fragment processor, one of two programmable areas on a GPU. The project grew

out of Jensen and Donner's work with colleagues from Stanford University on photon mapping with graphics processors.

Photon mapping simulates the indirect illumination of a scene fairly efficiently, and typically has been done on powerful workstations. But driven by the demands of gamers, chip designers have dramatically improved the power and speed of GPUs. "They've gone from barely being able to run the simple 2D graphics of your desktop graphical interface, to being able to run these really complicated games that you see coming out now at full frame rates, 60 frames per second," says Donner. "It's pretty amazing."

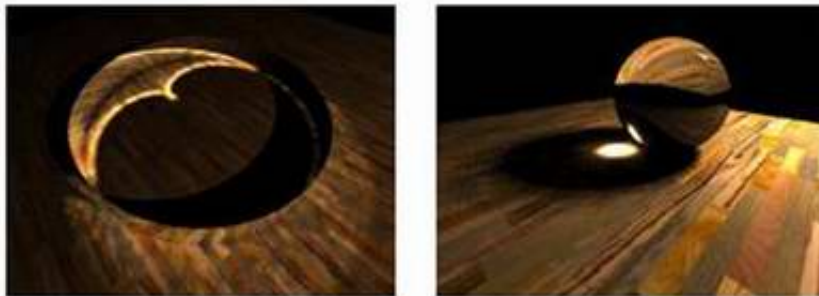


Figure 1: *Examples of scenes from [Purcell et al. 2003] rendered using our methods. Using adaptive refinement, the left image was rendered five times faster, and the right image was rendered four times faster.*

Photon mapping uses multiple passes, and on later passes, the active areas become sparse, i.e., only part of the pixels on the screen are active, and over time some will no longer be active. "This large geometry that most people are using for their computations is fairly wasteful," explains Donner. He and Jensen concluded that it might not be necessary to do

photon mapping on the entire image. "We had to perform this photon mapping computation only on specific parts of the image we were generating," recalls Jensen. "It dawned on us that you really only need to operate on part of the screen at a time."

Doing computation on a GPU typically involves rasterizing a piece of geometry on to the screen so it covers the entire screen, and then doing work on every single pixel in that image. "Yet not every pixel is doing useful work so you are wasting the resources of the hardware," claims Donner. "So what we settled on doing is actually tiling the screen and then culling away tiles that are no longer performing useful computations."

To figure out which tiles are inactive, the researchers took advantage of an existing feature of most recent graphics processors that permits 'occlusion queries'-asking for which tiles are active and have been drawn since the last query.

The researchers tested their technique on both photon mapping (which is very complicated, requiring hundreds of thousands of lines of code) as well as on a simple graphics algorithm known as a Mandelbrot Set (pictured). In both cases, adaptive refinement improved performance, especially in the later stages of each computation. For the easier computation, rendering was over three times faster. But with photon mapping, it was almost 300 times faster. "The more complex the computation, the more work you're wasting on fragments that aren't doing anything useful," explains Donner. "So the more complex your calculation, the more you gain with this method."

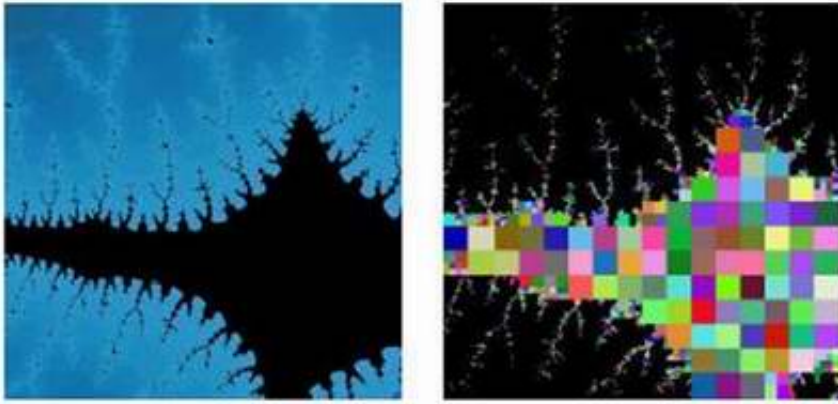


Figure 2: *A fragment shader implementing an iterative Mandelbrot Set solver. The left image is a visualization of the result, and the right image shows the active tiles.*

The optimal tile size in both scenarios was 16-by-16 pixels, but they say it depends on the algorithm itself. "Slightly larger tiles work better with smaller fragment programs that finish quickly," says Jensen, "while smaller tiles may be better suited to complex algorithms with long fragment programs." They also found that the best tile size is from 8 to 32 pixels wide on current hardware, and subdividing gives the best results when 40 to 60 percent of the fragments are active

According to the researchers, adaptive refinement should also be applicable to all sorts of other computations now done on graphics hardware.

In 2002, Craig Donner was the primary programmer on "Smoke and Mirrors," an interactive, multimedia installation funded in part by the institute and designed by UCSD visual arts professor Sheldon Brown, who is Cal-(IT)2's New Media Arts layer leader at the university, and director of the Center for Research in Computing and the Arts (CRCA).

Donner also collaborated with Brown and others on the new visualizations of the Cal-(IT)2 building, now under construction. The interactive 3-D model and multimedia visualizations were created by students Ben Maggos and Matt Hope, while the 3-D environment was controlled with a custom built 'scripter' developed by Donner.

Source: Jacobs School of Engineering

Citation: Adaptive Refinement Promises Faster Rendering on Graphics Chips (2004, October 4) retrieved 24 April 2024 from

<https://phys.org/news/2004-10-refinement-faster-graphics-chips.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.