

Thursday, September 06, 2007

## **Animation for the Masses**

Adobe is developing software to let home users create movie-quality 3-D graphics.

By Kate Greene

Computer-generated effects are becoming increasingly more realistic on the big screen, but these animations generally take hours to render. Now, Adobe Systems, the company famous for tools like Photoshop and Acrobat Reader, is developing software that could bring the power of a Hollywood animation studio to the average computer and let users render high-quality graphics in real time. Such software could be useful for displaying ever-more-realistic computer games on PCs and for allowing the average computer user to design complex and lifelike animations.

Adobe is focusing its efforts on ray tracing, a rendering technique that considers the behavior of light as it bounces off objects. Since it takes so long to render, ray tracing is typically used for precomputed effects that are added to films, computer games, and even still pictures before they reach the consumer, explains [Gavin Miller](http://www.doctorgavin.com/) (<http://www.doctorgavin.com/>), senior principal scientist at Adobe.

With the rise of multicore computing, Miller says, more consumers have machines with the capability to compute ray-tracing algorithms. The challenge now, he says, is to find the best way to divvy up the graphics processes within general microprocessors. "Adobe's research goal is to discover the algorithms that enhance ray-tracing performance and make it accessible to consumers in near real-time form," Miller says.

Consumer computers and video-game consoles compute graphics using an approach called rasterization, explains [John Hart](http://graphics.cs.uiuc.edu/~jch/) (<http://graphics.cs.uiuc.edu/~jch/>), a professor of computer science at the University of Illinois at Urbana-Champaign. Rasterization renders a scene by generating only those pixels that will be visible to a viewer. This process is fast, but it doesn't allow for much realism, explains Hart. "Rasterization is limited in the kinds of visual effects it can produce, and has to be extensively customized just to be able to approximate the appearance of complicated reflective and translucent objects that ray tracing handles nicely." For instance, in real life, if a light is shining at the side of a car, some of that illumination could reflect off metal in the undercarriage, and this would create a reflection on the ground that's visible to a viewer who's looking at the car from above. Rasterization would ignore the pixels that make up the undercarriage, however, and the reflection would be lost.

Ray tracing takes a fundamentally different approach from rasterization, explains Miller. "Rather than converting each object into its pixel representation, it takes all of the geometry in the scene and stores it in a highly specialized database," he says. This database is designed around performing the following fundamental query: given a ray of light, what points on a surface does it collide with first? By following a ray of light as it bounces around an entire scene, designers can capture subtle lighting cues, such as the bending of light through water or glass, or the multiple reflections and shadows cast by shiny three-dimensional objects such as an engine or a car.

Essentially, then, ray tracing tries to find the right information in a database as quickly as possible. This isn't a problem for rasterization, says Miller. Usually, the rendering process is straightforward, and data is cached and ready to go when the processor needs to use it. With ray tracing, however, the brightness of any given point on a surface could have been created from multiple bounces of a light ray, and data about each bounce of light tends to be stored in a separate location in the database. "This is a nightmare scenario for the caching strategy built into microprocessors, since each read to memory is in an entirely different location," says Miller.

He explains that his team is exploring various approaches to making these database queries more efficient. Previous research has produced algorithms that bundle certain types of data together to simplify the querying process. For instance, bundles of data can include information that represents rays of light that start from roughly

the same location, or rays that head in nearly the same direction. Adobe is not releasing the details of its approach, although Miller says that his team is trying to find the most efficient combination of database-management approaches. Once the researchers develop software that can effectively manage the memory of multicore computers, then ray-tracing algorithms can be rendered at full speed, he says.

"Adobe makes software that improves a user's ability to create and communicate visually," says Hart of the University of Illinois. "Software like Photoshop provides methods for processing photographs, but by adding ray tracking, users will have the ability to create photorealistic images of things they didn't actually photograph." One of the biggest obstacles at this point, he says, is making the system work fast enough so that a user can run a ray-tracing program interactively.

The current ray-tracing approach alone won't solve all the problems that computer-graphics researchers are tackling, Hart adds. It's still impossible to perfectly simulate the human face. "This is an elusive goal," he says, "because as we get more realistic ... subtle errors become more noticeable and, in fact, more creepy. Once we get faces right, we will need high-quality methods like ray tracing to render them, and we'll want it in real time."

The system is still just a research project, and the company doesn't provide a timeline for when it might make it to consumers, but technology on all fronts, including advances in multicore architecture, is advancing rapidly. Miller suspects that consumers will start to see real-time ray tracing in products within the next five years.

---

Copyright Technology Review 2007.