TRACING RAYS THROUGH THE CLOUD

Intel® Many Integrated Core Architecture Delivers Interactive Game Experience to Tablets

BY DANIEL POHL

CLOUD-BASED ONLINE STREAMING SERVICES FOR GAMES HAVE BEEN ENTERTAINING GAMERS FOR ALMOST TWO YEARS, and the complex graphics and multi-player capabilities of today's games are enjoyed using both large and small devices. More than just the potential of calculating the current rasterized games that we all know, tracing rays through the cloud opens up the possibility of higher realism in games.

In February 2011, Intel proved the concept of cloud-based ray tracing by demonstrating the ability to execute a high-resolution computer game on a remote server and stream the output to an ordinary consumer device at interactive frame rates. An article published at that time (http://software.intel.com/en-us/articles/cloud-based-ray-tracing/) described how a series of servers using the forthcoming Intel® Many Integrated Core (Intel® MIC) architecture were able to compute the heavy ray-tracing calculations of *Wolfenstein*:* Ray Traced (www.wolfrt.de) and stream the results to a networked laptop, while allowing the laptop to fully control the game.

The results demonstrated how remote servers could provide ordinary consumer devices with the benefits of ray tracing, including the ability to display and interact with:

- Highly detailed geometry with millions of triangles
- Sophisticated reflection effects, such as those created by a shiny car or sniper scope
- Refraction effects, as would appear through glass or through the crystal of a chandelier
- Multi-camera portals, such as a surveillance station showing 12 different parts of a game level

Since then, Intel engineers have demonstrated the ability of the Intel MIC architecture to scale, further proving the concept. Originally implemented on four servers, each with an Intel MIC card inside, engineers eventually supported the same workload with a single server and four—and ultimately eight—32-core Intel MIC subsystem cards in a single server. This configuration can now serve high frame rates to as many as eight tablet computers.

Scaling

Server performance between the single- and multi-card setups was dramatic. Benchmark testing was performed with the *Wolfenstein* courtyard view (level "mp_bank") in a performance-intensive position with all performance-sapping features, such as anti-aliasing, enabled. Intel engineers measured near-linear server performance scaling as the number of Intel MIC architecture cards (code-named Knights Ferry) was increased from one to eight. At peak performance, engineers measured a speedup of 7.73 times when eight Knights Ferry cards were present in the server, a scaling efficiency of 96.6 percent (see Figure 4).

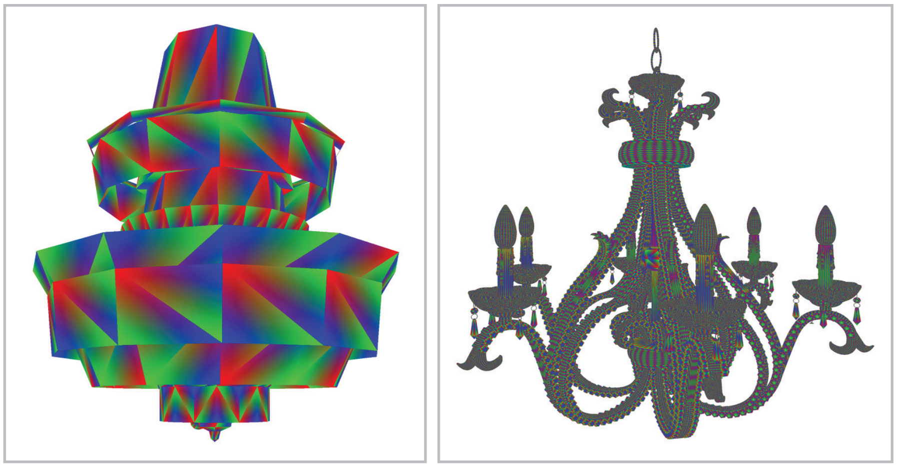


Figure 1. Original chandelier model with a few hundred triangles (LEFT); model with one-million triangles (RIGHT).

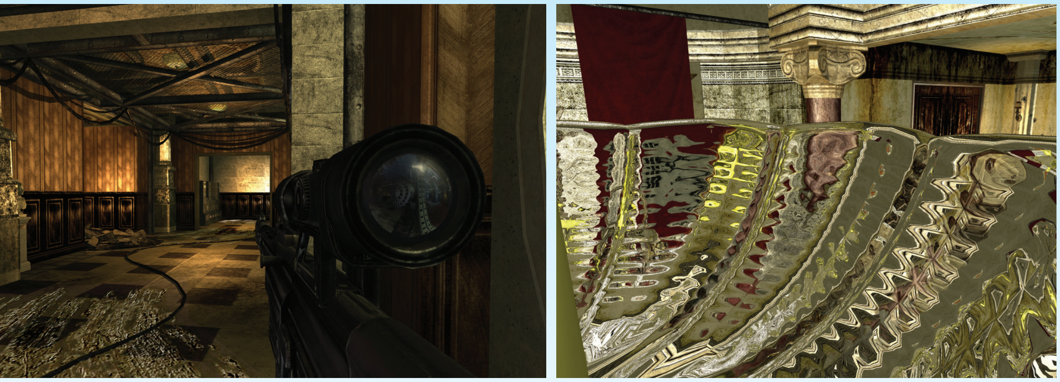


Figure 2. Reflecting sniper rifle scope (LEFT) and glass refractions (MIDDLE).

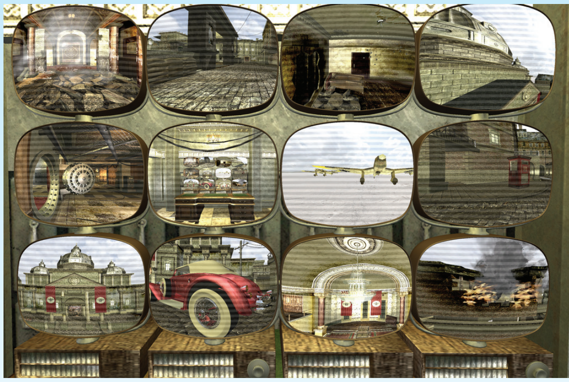


Figure 3. Surveillance station showing 12 parts of the level simultaneously (RIGHT).

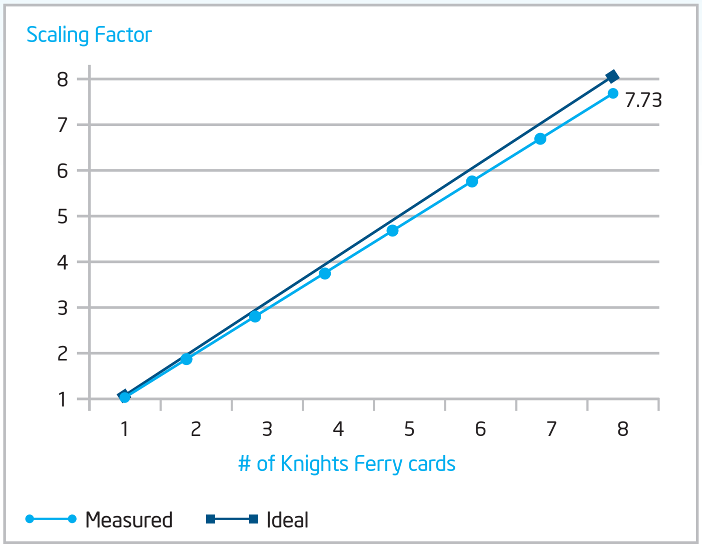


Figure 4. Internal benchmark tests showed that Intel® Many Integrated Core (Intel® MIC) architecture cards (code-named Knights Ferry) had a scaling efficiency at peak performance of 96.6 percent.

Tablet Testing

Having proved that laptop computers are a successful client device for cloud-based games that employ ray tracing, Intel engineers set out to test the technology on less powerful x86-based tablet devices. Test subjects were the Intel® Atom™ processor-based Viliv S5* Entertainment mobile Internet device (MID) EXOPC Slate* and Lenovo's IdeaPad* S10-3t. In tablet tests, engineers used the wireless Rii* mini-keyboard (www.riiminikeyboards.com/), with an integrated touch pad taking the place of the tablets' native touch controls.

To adapt the demo for tablet computing, engineers had to first select the technology that would receive and display the ray-traced images. For tablets, DirectDraw*/DirectX* was found to be the faster option compared to OpenGL* used in the test laptop. Engineers found that the Lenovo and Viliv tablets, with their 1024 x 600 pixel resolution, could each be serviced by a single Knights Ferry card, which delivered frame rates between 20 and 30 frames per second (fps). For the EXOPC Slate (and its native resolution of 1366 x 768 pixels), two Knights Ferry cards were needed.

Turning back to the laptop tests, engineers found that once the cloud server was equipped with eight Knights Ferry cards, they were able to increase the laptop resolution from 1280 x 720 to 1920 x 1080 pixels while maintaining interactive frame rates in the area of 60 fps. What's more, they were able to add post-processing effects and perform advanced anti-aliasing to further improve the user experience.

Post-Processing Special Effects

Ben Segovia, a research scientist at Intel Labs, implemented several post-processing effects into *Wolfenstein:* Ray Traced. These effects operate on the pixels of the rendered image (as opposed to the 3D scene) and are processed in real time by the Knights Ferry card. Although these effects have been seen in games before and are not specific to ray tracing, they improve the perception of the rendered scene dramatically in this scenario.

Depth of Field

The depth-of-field effect, one that's well known to photographers, can be used to focus attention on a certain area of an image. This effect puts the object of interest in sharp focus while the less-relevant parts appear blurred (Figure 5). The performance-intensive calculations were applied in the Knights Ferry-equipped system with just a 3-percent performance drop.

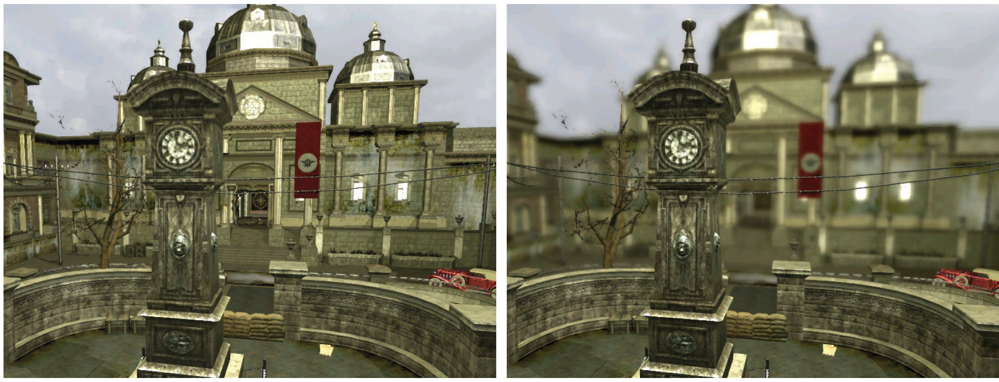


Figure 5. Comparison of depth-of-field effect (applied, at right).

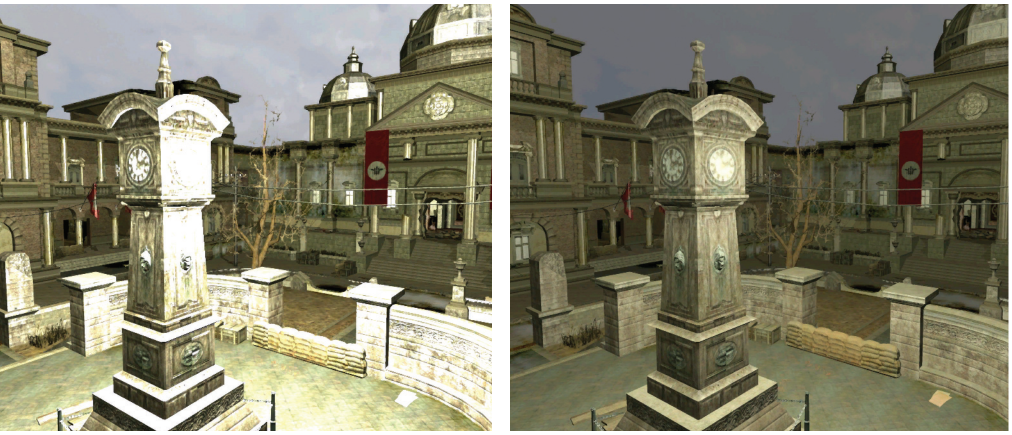


Figure 6. Comparison of high-dynamic range (HDR) (applied, at right).

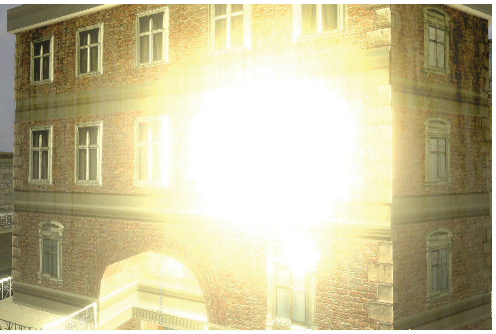


Figure 7. An example of the bloom effect.

HDR Bloom

When traversing in the real world from a dark room into bright sunlight, the eyes see an overexposed scene as they gradually adjust to the brightness. The same effect can be observed with digital (video) cameras, which adapt for brightness until a pleasant image is seen. During the transition, cameras might produce a bloom that can "bleed" into other objects, a phenomenon that is mimicked by the high-dynamic range (HDR) bloom effect. Also fairly compute-intensive, the HDR bloom exhibited a mere 2-percent cost.

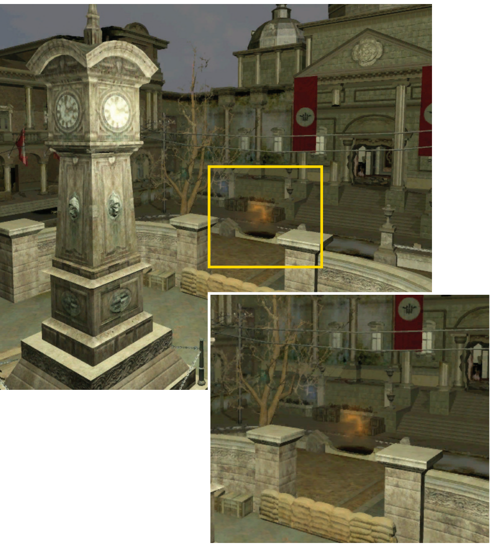


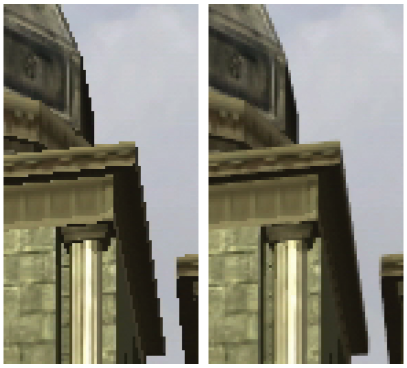
Figure 8. In an image-based, inter-lens reflection, a subtle effect casts an orange hue near the center of the image, also shown larger. This effect carries a performance cost of around 0.1 percent.

Inter-lens Reflections

While camera manufacturers strive to avoid lens flares, makers of computer games and movies often add them as an artistic element. In the implementation shown, several smaller-sized versions of the image, shifted to a specific color (for example, green, blue, and orange), have been blended into the original image. Intel engineers measured a 0.1 percent lag when applying this effect.

Smart Anti-Aliasing

Segovia and fellow Intel research scientist Ingo Wald are developing a more effective, less compute-intensive method of performing anti-aliasing, one that is applied after the image has been rendered. Because ray tracing allows the shooting of just a few rays for refinement, it's also relatively easy to analyze each pixel to determine if it requires more anti-aliasing based on the object's polygon mesh ID and the angle of the polygon that got hit at that pixel.



Zoomed-in view that shows the difference between smart anti-aliasing off (LEFT) and on (RIGHT).

Games generally save their geometric content in meshes, which are in turn defined by their use of a different shader or simply by being a different object. There might be one mesh for a column, a different one for a fence, another for a certain tree, and so on. In Intel's ray tracer (Figure 9), every mesh gets an internal ID.

Figure 10 shows a visualization of the second criteria that Intel uses in its algorithm. The orientation of the surface is visualized in the red, green, and blue color channels.

Next the algorithm looks at neighboring pixels. If there is a large enough variation in the orientation (for example, going from one roof that faces left to one that faces front) or a different mesh ID is found, the algorithm shoots 16 more rays (also called "supersampling") for that specific pixel and averages the resulting color into that pixel.

Future implementations could include the addition of criteria such as the color of the pixel itself or the colors between neighboring pixels. Morphological anti-aliasing (MLAA), an image-based technique described in 2009 by Intel Labs' Alexander Reshetov (http://visual-computing.intel-research.net/publications/papers/2009/mlaa/mlaa.pdf), could be used to first reduce aliasing before any color comparison takes place. After the MLAA process, fewer pixels would need to be refined by shooting new rays. Tweaking the number of additional rays to four or eight might lead to better trade-offs in performance and quality.

Industry Adoption

As powerful as they might be, most of today's high-end gaming PCs are unable to render complex, dynamic, fully ray-traced games at high resolutions and frame rates. So from an economic standpoint, there hasn't been a large enough potential customer base to justify the commercial development of a purely ray-traced game. But Intel work to offload complex ray-tracing calculations to the cloud could eventually enable highly realistic games to be played on smaller, more mobile systems.

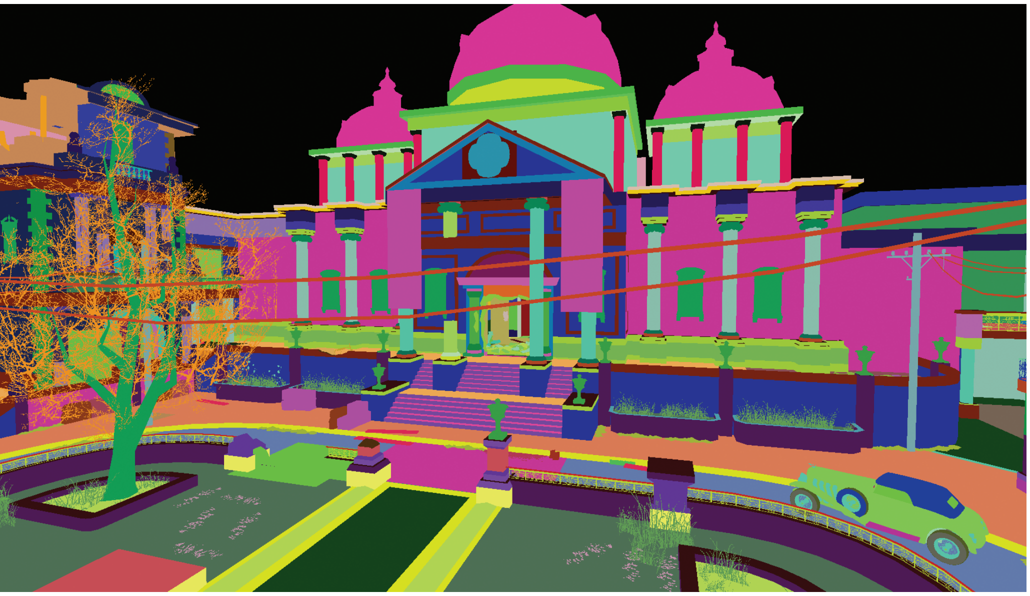


Figure 9. Intel's ray tracer displays a color-coded visualization of the mesh IDs (ABOVE).

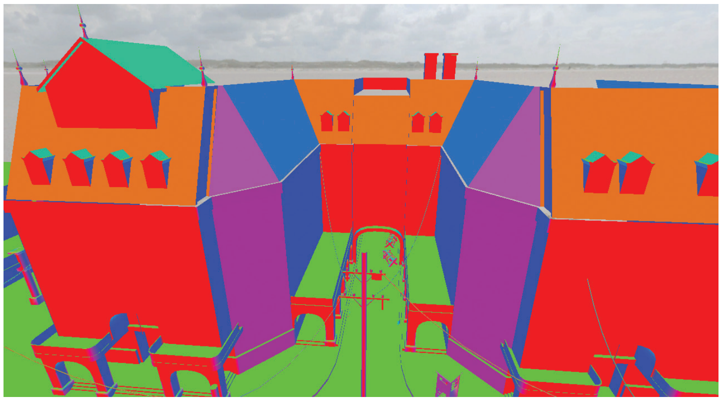


Figure 10. Intel's ray tracer color-codes orientation of the geometry (LEFT).

At the same time, as the computational power of all consumer devices continues to increase, game developers are presented with new target devices and customers. So it stands to reason that forward-looking game developers are paying close attention to advanced approaches to rendering.

One such notable figure is John Carmack, developer of games such as *Doom** and *Quake** and founder of id Software. In an August 2011 interview with *PC Perspective* (www.pcper.com/reviews/Editorial/John-Carmack-Interview-GPU-Race-Intel-Graphics-Ray-Tracing-Voxels-and-more/), Carmack said he was confident that ray tracing would eventually win out over rasterization as a rendering technology. "There are too many things [about rasterization] that we've suffered with, especially for shadows and environment mapping. We live with hacks that ray tracing can let us do much better."

And in a 2008 interview with *TG Daily*, (www.tgdaily.com/business/36410-tim-sweeney-part-2-%E2%80%9Cdirectx-10-is-the-last-relevant-graphics-api%E2%80%9D), Epic Games CEO Tim Sweeney called ray tracing ". . . a cool direction for future rendering techniques." The creator of the Unreal* game engine, Sweeney also noted the potential of other rendering schemes involving micro-polygons.

And Cevat Yerli, president, CEO, and founder of German game developer Crytek, mentioned ray tracing and voxel rendering as possible future directions in his keynote speech at the 2009 Game Developers Conference in Europe titled, "The Future of Gaming Graphics."

Rays in the Future

Intel has been making steady progress in its research and development of the use of real-time ray-tracing technology for the advancement of the graphics-rendering industry. Intel engineers have demonstrated the effectiveness of using the Intel MIC architecture as a cloud-based rendering platform for ray-traced gaming on a variety of lightweight client devices. And Intel's software efforts have made strides in the areas of anti-aliasing and related visual effects.

Looking ahead, there are still more interesting research questions to explore. Intel will continue its work on advancing and improving the efficiency of its anti-aliasing approach, and it will continue to study volumetric effects such as smoke and fire. Thanks to Intel's 22-nanometer process and the more than 50 cores per chip it allows, the next implementation of Intel MIC architecture (code-named Knights Corner) might lead to interesting advancements in cloud-based ray tracing. These are exciting times, indeed, both for the real world and for the fantastic worlds that Intel® technology make possible. ■

Acknowledgments

Special thanks to the following contributors: id Software and Raven Software for the Wolfenstein content; Alexander Reshetov, Benjamin Segovia, Alexey Soupikov, Ingo Wald, and Sven Woop for working on the internals of the ray-tracing engine; and Ram Nalla for working on the demo and enabling the cloud-based gaming setup. Thanks also to Nathaniel Hitchborn for his support on Knights Ferry and to Hans-Christian Hoppe and Sven Woop for their help in reviewing this article. And many thanks to editor Edward J. Correia.*

About the Author

Daniel Pohl started researching real-time ray tracing for games in 2004 during his study of computer science at Erlangen-Nuremberg University in Germany. As his master's thesis, he developed a ray-traced version of Quake 4**. In 2007, he joined Intel's ray tracing group, where he continues to research game-related ray tracing.*