White Paper

Relighting Technology

Sorbetto Relighting Technology

The Problem

Lighting a scene for 3D visual effects production is time consuming, and is also one of the most complex tasks in digital film production. Like live action production, proper lighting in computer-generated graphics can make or break a scene. To engage and entertain the audience, the lighting must set the mood, highlight important elements, and look natural.

In a traditional film set or studio, changes to lighting are visible instantaneously; however, in a computer-generated scene, the image must be rerendered each time a change is made to the scene. The iterative rendering process consumes valuable time, both from computer resources and highly paid human resources.

So far this problem has defied an innovative and integrated solution. Various lighting tools have addressed this task, but offered solutions that did not shorten the process or did not provide the true final image on the screen.

NVIDIA now introduces, NVIDIA Sorbetto™, a breakthrough solution for lighting based on its NVIDIA® Gelato™ renderer, a solution that saves time and increases production capabilities.

The Solution

Beginning with Gelato 2.0, users have the option of adding Sorbetto relighting technology to the core renderer. This addition allows rerendering the scene considerably faster after changes have been made to the lighting—fast enough in many cases to be called "interactive."

Sorbetto was designed with the following principles:

❑ **Relighting is done on the "final" pixels**. No approximations or unshaded models are used. What you see is the final image, just a whole lot faster.

❑ **Support is offered for a full range of lighting changes**. Lights can be added, deleted, or moved; any lighting parameter can be changed; and shadows can be recomputed.

❑ **No special preparation or shaders are required**. Any scene that can be rendered in Gelato can be relit in Sorbetto.

❑ **Its functionality is resident in the renderer and is fully documented in the Gelato API**. NVIDIA's Mango™ plug-in (for Maya) and Frantic Films' Amaretto plug-in (for 3ds Max) will both support Sorbetto. What's more, any application can use Sorbetto via the API provided by NVIDIA.

How Sorbetto Works

To free up system resources and keep the render going at maximum efficiency, professional film software renderers aggressively discard data as soon as it is no longer required. Gelato, for example, tosses out most of the data as soon as a "bucket" or "tile" is complete, retaining only the RGB data for the final pixel.

When running in Sorbetto mode, though, Gelato retains much of the data it would normally discard, caching it in system memory. Sorbetto then uses this interim data on subsequent rerenders to accelerate the iteration cycle and provide faster feedback.

Sorbetto can determine whether a pixel will be affected by the lighting change, and then recalculate only the pixels it needs (Figure 1 and Figure 2).

Figure 1.   Initial Render, 3 min. 40 sec. at 2 K Resolution

Sorbetto initially caches this data in system RAM by running a full render at the beginning of the lighting session. This initial render takes slightly longer than a normal renderer, but the difference is probably not noticeable to the human operator waiting for the process to complete.

Further efficiencies can be achieved by using other Sorbetto features. Sorbetto can focus on a crop window or how lighting affects a particular character or object, ignoring the rest of the scene. Sorbetto is also interruptible; if lighters do not like a change, they can restart the render using new parameters, before the last render completes. And the cached data can be saved to disk, letting the lighter stop work and resume later at the same point, or allowing the initial renders to be run overnight so the scenes are ready for the lighter to work on the next morning.

Figure 2. Rerender (Garage Door Light Reoriented and Color Changed), 1 min. 14 sec.

Additional system RAM is required to run Sorbetto, but since Gelato has a small memory footprint, the additional RAM required is not much. NVIDIA recommends that Sorbetto run on systems with 2 GB of RAM or more. And when memory limits are reached, Sorbetto degrades gracefully, taking longer to complete the rerendered image.

Sorbetto Functions

Rerendering. Rapidly recompute changes to lighting much faster than an ordinary render.

Relighting API. All Sorbetto functions are exposed in the Gelato API. They're not dependent on any particular modeling or animation software. Plug-ins can be created for any production tool to allow complete access to Sorbetto's functionality.

What you see is what you get. Relighting is on the "final pixels," including full antialiasing, motion blur, transparency, displacement, and production shaders. The rerendered image is always identical to the final rendered image.

Capability to change anything to do with lighting and then rerender.

❑ Add/delete lights
❑ Move/reorient lights
❑ Change any light shader parameter
❑ Change light linking (what lights shine on what surfaces)

Reflections and shadows are recomputed as well.

Ray tracing is fully supported.

Selective relighting. Recompute lighting for a crop window or specified object for even faster results.

Interruptible. Make changes on the fly before the rerender is finished.

Ability to save state to disk and continue relighting session at a later time. Or batch the initial Sorbetto renders during the night and have the scenes waiting for you to relight in the morning.

Support for Maya and Max through NVIDIA's Mango plug-in for Maya and Frantic Film's Amaretto plug-in for 3ds Max.

The Current Implementation

Sorbetto's architecture has a few limitations that NVIDIA is actively working on to resolve. The main shortcoming is that changes affecting the visibility of objects cannot be made. For example:

❑ Camera and geometry moves, as well as changes to displacement shaders, cannot be rerendered by Sorbetto. Sorbetto does not cache data about objects not seen in the initial frame (because they were offscreen or behind opaque objects).

❑ Adjusting the opacity of an object that was initially transparent is allowed, but you cannot make an initially opaque object transparent. Gelato has "forgotten" what was behind the object, so you will not see what's there when you rerender with Sorbetto.

These limitations may be removed in new releases of Sorbetto, but not in the near future.

Another limitation, which will be resolved sooner, is that Sorbetto only works with lighting shaders and parameters. NVIDIA continues to develop Sorbetto to allow rerendering of changes to surface shaders. Sorbetto in Gelato version 2.0 does not support this function, but an upcoming release will.

Finally, scenes that are extremely complex and have large number of elements will hit a point of diminishing returns, where the rerendering time could approach that of a normal render. To solve this situation, you can focus relighting on a portion of the scene by using a crop window or by relighting individual objects or small groups of objects (Figure 3). You can also split the scene; light, relight, and render; and then reassemble the frame in compositing.

Figure 3.   Crop Window—Only Rerendered (Left: Shadows Softened in
              Rerendered Image; Right: Original Image), 6 sec.

Truly Revolutionary

Gelato was launched with the tagline, "Revolutionary Rendering." With the release
of Gelato 2.0 and Sorbetto, this promise is paying off. Sorbetto allows on-the-spot
feedback from the director during dailies. Changes can be made and approved
immediately, without waiting to review the next iteration.

Sorbetto is revolutionary not just because it is faster, but because its speed will
change how filmmakers work and how the digital production pipeline is structured.