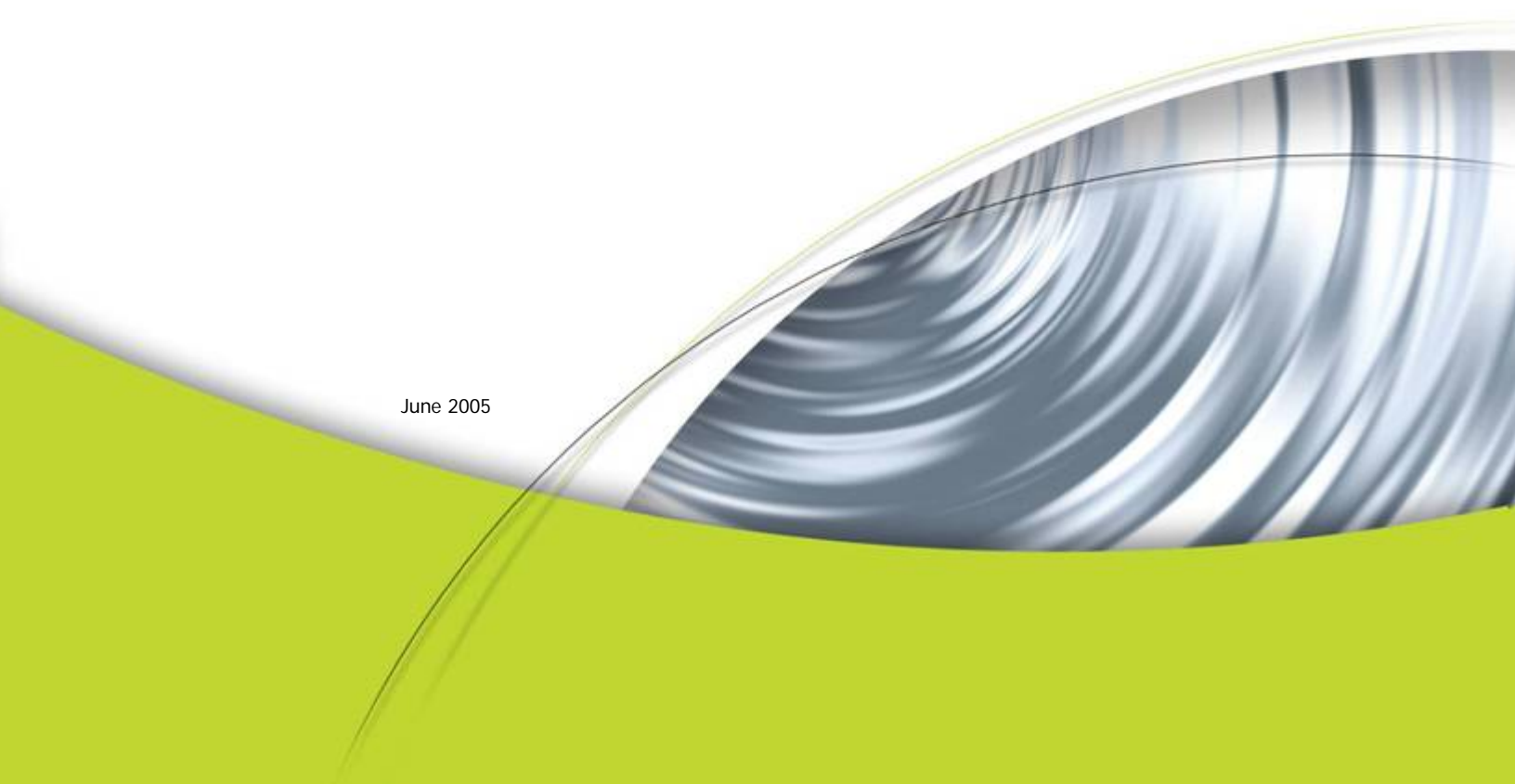




# Evaluation Notes

**NVIDIA Gelato**  
Final Frame Renderer

June 2005

A decorative graphic at the bottom of the page consists of a green curved shape on the left, a white curved shape in the middle, and a blue and white striped curved shape on the right, all overlapping and curving upwards.

# Read Me First

Thank you for evaluating the NVIDIA® Gelato™ final-frame renderer. This guide is primarily intended for those evaluating Gelato for the purposes of purchase or just curiosity. Those in the press who wish to review Gelato for publication should first consult the press contacts at the end of the guide.

Included in these Evaluation Notes are:

- ❑ Product background material
- ❑ System configuration information
- ❑ Some specific test cases
- ❑ Contact information for public relations (PR), sales, and product information

Gelato is a complex, professional program that is highly sensitive to system configuration. While Gelato should install and run “right out of the box,” you may encounter issues and problems when first using the software. Because a product like Gelato is so dependent on a variety of factors outside of our control, you may encounter unexpected set up or performance issues upon installation. If this occurs, please contact Gelato technical support at [gelatosupport@nvidia.com](mailto:gelatosupport@nvidia.com). Our support engineers can walk you through the system configuration and get you up and running.

Also, Gelato’s performance curves are different from other renderers you may be familiar with. If you attempt to run a scene that has been created and optimized for another renderer, Gelato’s performance may not appear to be much higher. However, with a few tweaks to the scene file or to the shaders, Gelato’s performance will usually outstrip that of other renderers. Therefore, we encourage you to read this entire document prior to testing and to refer to it during testing and while writing your test report. You can obtain further information or assistance from NVIDIA:

- ❑ By contacting [gelatosupport@nvidia.com](mailto:gelatosupport@nvidia.com)
- ❑ By visiting the public Gelato discussion forums at <http://forums.nvidia.com>
- ❑ (For press) By logging into the NVIDIA press FTP site at <ftp://press.nvidia.com> with your username/password.
- ❑ Through the appropriate NVIDIA contact listed at the end of this document.

If you have not informed NVIDIA of your intent to evaluate Gelato, please do so before requesting technical support. This will greatly streamline the response process—requests and questions from unknown sources will be given a low priority in the support queue; if we know who you are, the response will be faster. So please contact either the Digital Film Group sales (if you are a potential user) or NVIDIA public relations (for press inquiries) and let us know who you are and why you want to test Gelato. The contact information is at the end of this guide.

This reviewer’s guide does not take the place of personal contact. We welcome your e-mails and phone calls if you need additional information, or if you encounter technical difficulties or less-than-expected results.

# Table of Contents

<b>Read Me First .....</b>	<b>i</b>
<b>Table of Contents .....</b>	<b>ii</b>
<b>Gelato Final Frame Renderer .....</b>	<b>1</b>
What is Gelato? .....	1
Features .....	1
What’s New in Gelato 1.2? .....	1
Complete Feature List .....	2
<b>System Configuration .....</b>	<b>3</b>
Recommended Configuration.....	3
System Requirements .....	3
Gelato Version.....	3
Other Software .....	4
CPU .....	4
Graphics Board.....	4
NVIDIA Driver .....	5
RAM.....	5
Operating System .....	6
Gelato License Key.....	6
<b>Setting Up Test Cases .....</b>	<b>7</b>
General Configuration .....	7
Test Case Set Up .....	7
General Information on Test Cases.....	8
Test Cases.....	9
Simple Render ( <i>killeroo-party</i> ).....	9
Preview Mode ( <i>killeroo-party</i> ).....	10
iv Image Comparison Tools ( <i>killeroo-party</i> ).....	12
Progressive Mode & Priority Bucket Order ( <i>killeroo-party</i> ) .....	12
Alternative Shading Modes ( <i>killeroo-party</i> ).....	13
Simple Motion Blur ( <i>killeroo-mb</i> ).....	15

Multi-Segment Motion Blur ( <i>msmb-points-explode</i> ) .....	15
Production Scene ( <i>trooperoo</i> ) .....	16
Fur ( <i>hairyMonster</i> ).....	18
Depth of Field ( <i>DOF</i> ).....	19
Ambient Occlusion ( <i>ambocc</i> ).....	20
Mango Render ( <i>SloppySam</i> ).....	21
Subsurface Scattering in Mango ( <i>SloppySam</i> ).....	22
Caustics in Mango ( <i>Caustics</i> ).....	22
Mango Render ( <i>Lab</i> ) .....	23
Running Your Own Test Cases.....	24
<b>Technical Support</b> .....	<b>25</b>
<b>NVIDIA Contacts</b> .....	<b>26</b>
<b>Credits</b> .....	<b>27</b>

# Gelato

## Final Frame Renderer

---

### What is Gelato?

Gelato is a software rendering system designed to create beautiful imagery for film and other high-end applications. The software is accelerated with the NVIDIA Quadro® FX professional-quality hardware to create this imagery at unprecedented speeds. The two stated goals at the outset of the design of Gelato were:

- ❑ To accelerate the process with revolutionary new programmable graphics hardware applied to rendering operations. The combination of Gelato and NVIDIA Quadro FX finally allows rendering to take advantage of the faster-than-Moore's-law rate of improvement in graphics hardware over time. While at the same time, Gelato introduces unprecedented features such as real interactive previewing of lighting, shading, and final renderings.
- ❑ To never compromise on image quality, robustness, power, programmability, flexibility, or any other property necessary for a top-of-the-line film rendering system. In particular, we will never let Gelato's use of graphics hardware result in inferior rendering output.

We believe that we have achieved both of these goals, and are proud to present the Gelato renderer.

---

### Features

#### What's New in Gelato 1.2?

##### Image Quality

- ❑ Ray tracing and ambient occlusion interpolation quality is greatly improved

##### Geometry Support

- ❑ Wide curves: supports ribbon-like curves that can be textured in width as well as length

##### Programmable Shading & Lighting

- ❑ Caustics
- ❑ Subsurface scattering
- ❑ Layered light shaders
- ❑ Average-Z ("Woo") shadow maps

- ❑ Cube-faced shadow maps—look up shadows from any direction with a single query

## Production-Ready

- ❑ More flexible ConnectShaders
- ❑ User-priority buckets—set which buckets should be rendered first
- ❑ Spiral bucket order—render an area of interest on command
- ❑ Interleave utility—allows “field rendering” by combining alternate scanlines from two images into one

## High Performance

- ❑ Ray tracing speed improved by 30%
- ❑ Ambient occlusion speed improved by 50%

## Tool Set

- ❑ Mango: Point light shadows now supported
- ❑ iv: new and improved image viewer

## System Requirements

- ❑ Now supports all NVIDIA Quadro FX GPUs

# Complete Feature List

## Image Quality

Gelato was designed with the philosophy of “no compromises” on image quality and to produce the highest quality images without sacrificing quality for the sake of performance. The following features are implemented to provide this quality:

- ❑ Unlimited antialiasing via supersampling and user-selected filters
- ❑ High-quality motion blur
- ❑ Depth of field
- ❑ Automatic adaptive tessellation
- ❑ Faster and higher quality ray tracing, ambient occlusion, and global illumination
- ❑ Image output: 8-bit, 16-bit, and float
- ❑ Output image channels for any value computed in shaders

## Geometry Support

- ❑ NURBS, bicubic, and bilinear patches
- ❑ Polygon meshes
- ❑ Subdivision surfaces
- ❑ Curves (hair)
- ❑ Wide curves (feathers and ribbons)

- ❑ Particles
- ❑ Procedural geometry and plug-ins

## Programmable Shading and Lighting

Gelato uses its own shading language, based on C, to create surface textures and lighting for scenes using the renderer. Gelato Shading Language provides the flexibility required for the most complex scenes.

- ❑ Caustics
- ❑ Subsurface scattering
- ❑ True displacement on all primitive types
- ❑ Layered surface and light shaders
- ❑ Anti-aliased texture, environment, and shadow mapping
- ❑ Atmospheric effects
- ❑ Vertex variables
- ❑ Unlimited number of lights
- ❑ Global illumination
- ❑ Ambient occlusion
- ❑ Ray traced reflections, refractions, and shadows
- ❑ Average-Z (“Woo”) shadow maps
- ❑ Cube-faced shadow maps
- ❑ DSO shadeops
- ❑ Sample shaders and shader function library

## Production Ready

Gelato is designed to fit into just about any production pipeline and comes with features to support an efficient workflow:

- ❑ Unlimited image resolution
- ❑ Fully selective lighting
- ❑ Preview mode
- ❑ Network Parallel Rendering
- ❑ Holdout matte objects
- ❑ No eyesplits
- ❑ More flexible ConnectShaders
- ❑ User-priority bucket order (spiral and crop window)
- ❑ Interleave utility (field rendering)

- ❑ Flexlm licensing—floating or node-locked
- ❑ Open, documented, royalty-free formats

## High Performance

Gelato was designed from the ground up using the NVIDIA Quadro FX line of commodity programmable graphics hardware to speed up various internal functions. No special shaders, coding, or configuration are required to use the hardware; this does not affect flexibility or image quality in any way.

Subsequent releases of Gelato will take greater advantage of the graphics hardware and future graphics hardware will be even faster and more capable. For the past few years, graphics hardware has been doubling in speed every 6-12 months, whereas CPUs have been doubling in speed roughly every 18 months. So renderers based on graphics hardware will not only perform well now, but will over time rapidly out-strip the performance of CPU-only renderers.

Other performance features include:

- ❑ Hardware-accelerated
- ❑ 30% faster ray tracing than 1.1
- ❑ 50% faster ambient occlusion than 1.1
- ❑ Multithreaded on CPU
- ❑ Efficient handling of complex scenes
- ❑ Efficient memory use

## Tool Set

Gelato includes a variety of tools and utilities, including Mango, Gelato's plug-in for the Alias Maya modeling and animation software package. It enables Maya users to choose Gelato as the renderer for Maya scenes. Other tools are:

- ❑ Powerful C++ API
- ❑ Python binding
- ❑ Plug-ins for scene file formats:
  - Python scene description
  - Gelato byte-stream scene description
  - RIB scene description (third-party plug-in)
  - Mango tool for Maya
- ❑ Image & texture I/O plug-ins:
 

➤ TIFF	➤ PNG
➤ OpenEXR	➤ DDS
➤ JPG	➤ IFF
➤ HDR	➤ TARGA
➤ PPM	
- ❑ Improved iv image viewer
- ❑ gslc shader compiler



- ❑ maketx—convert image files to textures
- ❑ Shader developer libraries
- ❑ Mango:
  - Render to either Maya's render window or iv (user choice)
  - Translates Hypershade networks to Gelato shaders
  - Replicates all Maya's light shaders
  - Assign any existing Gelato shader to an object/scene from within the Maya GUI
- ❑ Third party plug-ins and tools (available, but not part of the Gelato package):
  - 3ds Max plug-in
  - RIB scene file I/O plug-in
  - Renderman to Gelato shader converter

## Support

- ❑ Technical Reference manual
- ❑ Mango User's Guide
- ❑ Source code examples of plug-ins and shaders
- ❑ Online support forums
- ❑ Annual support available from NVIDIA
  - Online, phone, fax support direct from NVIDIA
  - All product releases and upgrades throughout the year

# System Configuration

---

## Recommended Configuration

Gelato is designed to run on a wide variety of system configurations. For optimal results, however, we recommend standardizing test systems based on the following configuration:

- ❑ Gelato 1.2R2 or higher
- ❑ 2.4 GHz AMD Opteron Model 250 CPU or better
- ❑ Graphics Board
  - AGP Systems: NVIDIA Quadro FX 3000
  - PCI Express Systems: NVIDIA Quadro FX 3400 or 4400
- ❑ 2 GB DDR400 ECC (4 DIMMS) RAM
  - Suse Linux 9.1 (32bit)

The NVIDIA Quadro drivers do not yet support multiple OpenGL contexts which will allow Gelato to use multiple GPUs in a single rendering node (“SLI”). The required change to the NVIDIA driver is planned.

This is a recommendation for testing and evaluating the system; it is not a requirement for operation. You may modify the configuration so long as it meets the system requirements described below. For example, you may choose to use Windows XP instead of Linux.

If you are using Gelato’s network parallel rendering feature (using multiple machines to render a single image) the machines do not require identical configurations. You can mix and match CPUs, NVIDIA Quadro FX boards, and even operating systems, so long as each machine meets the minimum system requirements.

---

## System Requirements

### Gelato Version

We strongly recommend that you evaluate the latest version of Gelato so that you have all the latest bug fixes and performance optimizations. You can download the latest version from: [http://film.nvidia.com/page/gelato\\_download.html](http://film.nvidia.com/page/gelato_download.html)

## Other Software

### Renderman-to-Gelato Scene File Plug-in

Many of the scenes included here require the Renderman-to-Gelato plug-in that enables Gelato to read scene files written in the Renderman (RIB) format. This plug-in, developed by a third party, is open source and available free of charge. A link to the download site is on:

[http://film.nvidia.com/page/gelato\\_download.html](http://film.nvidia.com/page/gelato_download.html)

Ensure that you have the latest version of the plug-in installed.

### Maya

You need Alias Maya if you want to evaluate Gelato's Mango plug-in. We recommend Version 6.5 or higher, although Mango will function with Maya 6.0.

### Renderman-to-Gelato Shader Translator (RSL2GSL)

This tool is not required to render any of the test scenes provided with these Evaluation Notes. If you wish to render your own Renderman scenes, however, this tool will assist you in translating your Renderman shaders to the Gelato Shading Language—translating the shaders automatically in most cases. This tool is another third-party application that is available at no cost. A link to the download site is on [http://film.nvidia.com/page/gelato\\_download.html](http://film.nvidia.com/page/gelato_download.html).

## CPU

- Recommended:
  - AMD Opteron
- Qualified:
  - AMD Athlon
  - Intel Xeon
  - Intel Pentium 4

## Graphics Board

Gelato requires an NVIDIA Quadro FX graphics board. Gelato is certified on all NVIDIA Quadro FX boards.

Gelato is not certified or supported on the GeForce line of products, although we have done nothing to deliberately disable it on those graphics cards. Because GeForce cards are manufactured and marketed by a variety of firms to different specifications and quality control standards, NVIDIA cannot ensure that Gelato will properly function throughout the GeForce line. Therefore we do not recommend, certify, or support those cards with Gelato.

Which NVIDIA Quadro FX product you should test with depends on your needs and what systems you intend to run Gelato on. Generally, higher-end boards are best suited for workstations and mid-range boards for servers.

- ❑ Ultra-High Performance
  - NVIDIA Quadro FX 4400 (PCI Express)
- ❑ High Performance
  - NVIDIA Quadro FX 3400 (PCI Express)
  - NVIDIA Quadro FX 4000 (AGP)
  - NVIDIA Quadro FX 3000 (AGP)
- ❑ Price-Performance
  - NVIDIA Quadro FX 1400 (PCI Express)
  - NVIDIA Quadro FX 1100 (AGP)
- ❑ Laptop
  - NVIDIA Quadro FX Go1400 (PCI Express)
- ❑ Legacy Servers
  - NVIDIA Quadro FX 600 (PCI; for those machines without AGP or PCI Express slots)
- ❑ Gelato also runs on the following graphics boards. But previous generation or entry-level boards may not provide the performance you require when using them with Gelato. Availability may also be limited. But if they already exist in your infrastructure you can certainly take advantage of them:
  - NVIDIA Quadro FX 330 (PCI Express)
  - NVIDIA Quadro FX 500 (AGP)
  - NVIDIA Quadro FX 540 (PCI Express)
  - NVIDIA Quadro FX 700 (AGP)
  - NVIDIA Quadro FX 1000 (AGP)
  - NVIDIA Quadro FX 1300 (PCI Express)
  - NVIDIA Quadro FX 2000 (AGP)
  - NVIDIA Quadro FX Go700 (AGP)
  - NVIDIA Quadro FX Go1000 (AGP)

## NVIDIA Driver

You should update your NVIDIA driver to the latest released version before testing Gelato. The latest driver is included in the Gelato distribution and we recommend that you install this version before beginning testing. For optimal performance on all features, you should have at least:

- Driver Release 7184

## RAM

Gelato has no minimum memory requirements, but will perform better as you add memory. We recommend a minimum of 1GB RAM if you are rendering complex or large scenes.

## Operating System

The Windows and Linux versions of Gelato have identical feature sets. Because of differences between the Windows and Linux file systems, Gelato runs marginally slower on Windows than it does on Linux, but you should find that Windows performance is well within the acceptable range.

Qualified:

- ❑ Microsoft Windows XP
- ❑ Linux Redhat Enterprise 3 or higher
- ❑ Linux Redhat 7.2 or higher
- ❑ Linux SUSE 8.0 or higher

## Gelato License Key

Without a license key from NVIDIA, Gelato will run in evaluation mode. In evaluation mode Gelato runs at full performance, so you can evaluate and benchmark it, but certain features are restricted or disabled. The restrictions are as follows:

- ❑ The rendered image and shadow maps will be watermarked with the word “GELATO” in small letters, repeated periodically on the image.
- ❑ Image output is restricted to RGB, RGBA, and Z, 8-bits per channel only.
- ❑ DSO shadeops will not be loaded.
- ❑ Network parallel rendering will be disabled.
- ❑ Other minor features may be disabled.

If you would like to evaluate the disabled features or need un-watermarked images for the publication of your review, you may obtain a temporary license key from NVIDIA. Contact [gelatosupport@nvidia.com](mailto:gelatosupport@nvidia.com) and provide them with the following information:

- ❑ Flexlm hostid/MAC address of the system(s) to be used in testing. If you are running a Flexlm server, you should provide the hostid of the server and the number of copies of Gelato you need for the test.
- ❑ The period during which testing will be conducted.

If you have not informed NVIDIA of your intent to evaluate Gelato, please contact us before requesting a key or other technical support. This will greatly streamline the response process—requests and questions from unknown sources will be given a low priority in the support queue; if we know who you are, the response will be faster. So please contact either the Professional Solutions Group sales (if you are a potential user) or NVIDIA public relations (for press inquiries) and let us know who you are and why you want to test Gelato. The contact information is at the end of this guide.

# Setting Up Test Cases

---

## General Configuration

The files needed to run the test cases in this guide are available on either the accompanying CD-ROM or at [http://film.nvidia.com/page/gelato\\_download.html](http://film.nvidia.com/page/gelato_download.html). The files for each test case are in an individual scene directory which contains the scene files, shaders, and textures needed to render the scene with Gelato.

Approximate rendering times for Gelato are given for each scene. The render times you achieve will differ somewhat depending on version, operating system, hardware configuration, and NVIDIA driver, but the published times can be used to see if you are “in the ballpark.” If your render times are significantly slower, there is probably a configuration problem with your system. In such a case, please contact [gelatosupport@nvidia.com](mailto:gelatosupport@nvidia.com).

You are not restricted to running the tests given in the batch files for each scene. You can alter various rendering parameters such as bucket size and final resolution simply by altering the batch files. You may also alter the scene files, shaders, and textures if you wish. Or you can run your own scenes. The files presented here are simply a jumping off point for your testing.

## Test Case Set Up

The scene files are located on the accompanying CD-ROM or in the zip file downloaded from the internet.

Windows users should copy these files to the c:\ directory (i.e., c:\rguide). You can place them elsewhere, but if you do so you will also have to alter the Maya shader and texture paths in Render Globals window in order to render the Maya scenes correctly.

Linux installation is a bit more complicated as the Maya scenes require a known path for shaders and textures. Linux users should copy the files to any location on the hard drive, keeping the relative file structure intact and then, after opening the Maya scenes, alter the texture and shader paths in the Render Globals window to include the directories where the files reside.

The test cases can be launched by running the .bat (Windows) or .sh (Linux) files contained in each scene directory.

## General Information on Test Cases

The test cases are:

- ❑ Simple render (killeroo-party)
- ❑ Preview modes (killeroo-party)
- ❑ Simple motion blur (killeroo-mb)
- ❑ Multi-segment motion blur (msmb-points-explode)
- ❑ Production-type scene (trooperoo)
- ❑ Fur (hairyMonster)
- ❑ Depth of field (DOF)
- ❑ Ambient occlusion (ambocc)
- ❑ Mango render (SloppySam)
- ❑ Mango/caustics render (Caustics)
- ❑ Mango render (Lab)

For each test case, this guide lists the approximate time it should take Gelato to render the image. The time it takes you to render the image will vary somewhat from the time stated depending on a variety of factors, such as the type of NVIDIA Quadro FX board you use, your operating system, your CPU, and the NVIDIA driver version you have installed.

Note that the Windows .bat files will render images with a gamma of 2.2. You can change this setting by altering the .bat files. The Linux .sh files will render images with whatever gamma setting is set in your environment variables. The reference .tif files were rendered at gamma 1.0, the reference .jpg files at gamma 2.2. If the brightness of your images differs from the reference images, this is probably due to the gamma setting. If you are using Gelato's `iv` image viewer to view the images, you can easily alter the gamma setting via the View menu or the `-/=` keys.

If the times you see are significantly slower than those listed here, please contact [gelatosupport@nvidia.com](mailto:gelatosupport@nvidia.com). You most likely have a configuration problem and chances are this can be easily corrected.

Unless otherwise stated, the render times given were achieved on a system with the following specifications:

- ❑ Gelato Version: 1.2R2
- ❑ 2 GHz AMD Opteron 246
- ❑ 2 GB RAM
- ❑ Windows XP
- ❑ NVIDIA Quadro FX 3000
- ❑ NVIDIA Driver Version 7184

---

## Test Cases

### Simple Render (*killeroo-party*)

This is a straightforward render producing an image of several mutant kangaroos, or *killeroos*. The model is NURBS-based with true displacement.



Figure 1. Killeroo-party

There are two batch files for each operating system that produce this image:

- ❑ killerooparty-1024.bat (Windows, 1024x768 resolution)
- ❑ killerooparty-1920.bat (Windows, 1920x1200 resolution)
- ❑ killerooparty-1024.sh (Linux, 1024x768 resolution)
- ❑ killerooparty-1920.sh (Linux, 1920x1200 resolution)

These files produce an image file named `killerooparty.tif`. (Note: all the batch files in this directory produce a file with this name. If you run another batch file, the old image will be overwritten. To save an image, rename it before running a new one.)

The reference images at the two resolutions are `kp-1024ref.tif` and `kp-1920ref.tif`.

Rendering times:

- ❑ 1024x768 resolution: 26 seconds
- ❑ 1920x1200 resolution: 68 seconds



## Preview Mode (*killeroo-party*)

### Preview at 10% Quality

Gelato has two types of preview modes that reduce the quality of the shading in order to increase speed to near-real time. This allows the artist to render the image in a “quick and dirty” fashion in order to check the geometry and lighting.

The first type of preview allows the user to reduce the quality settings of several shading attributes, resulting in a lower-quality render. This mode uses the same Gelato shaders as the full-quality render, but reduces the spatial anti-aliasing, motion blur, depth of field, number of points and curves rendered, etc.

Figure 2 shows the killeroo party scene rendered using Gelato’s preview mode. The defects in the image at Figure 2 are visible at a glance, but the speed of the preview mode gives the artist quick feedback.



Figure 2. Killeroo-party in preview mode at 10% quality

The following batch files render the image at 10% (0.1) quality at 1024x768 or 1920x1200 resolution:

- ❑ preview1-1024.bat (Windows)
- ❑ preview1-1024.sh (Linux)
- ❑ preview1-1920.bat (Windows)
- ❑ preview1-1920.sh (Linux)

The batch files produce an image file named `killerooparty.tif`. (Note: all the batch files in this directory produce an image file with this name. If you execute another batch file, the old image file will be overwritten. If you wish to save the image, rename it before running another batch file.)

The directory includes reference images at the two resolutions, `preview1-1024ref.tif` and `preview1-1920ref.tif`.

Rendering times:

- ❑ 1024x768 resolution: 4.5 seconds
- ❑ 1920x1200 resolution: 7.5 seconds

## Preview at 50% Quality

Increasing the quality to 50% (0.5) yields the following image.



Figure 3. Killeroo-party in preview mode at 50% quality

The following batch files render the image at 50% (0.5) quality at 1024x768 or 1920x1200 resolution:

- ❑ preview2-1024.bat (Windows)
- ❑ preview2-1024.sh (Linux)
- ❑ preview2-1920.bat (Windows)
- ❑ preview2-1920.sh (Linux)

The batch files produce an image file named killerooparty.tif. (Note: all the batch files in this directory produce an image file with this name. If you execute another batch file, the old image file will be overwritten. If you wish to save the image, rename it before running another batch file.)

The directory includes reference images at the two resolutions, preview2-1024ref.tif and preview2-1920ref.tif.

Note that at 50% quality, the defects in the image are difficult to spot by the casual viewer, but become apparent upon close examination.

Rendering times:

- ❑ 1024x768 resolution: 10.0 seconds

- ❑ 1920x1200 resolution: 21.4 seconds

## iv Image Comparison Tools (*killeroo-party*)

If you don't close the iv window between renders. The differences between the two preview renders (10% and 50%) provide a good opportunity to look at iv's image comparison tools. You can toggle between the two most recently rendered images using the command in the Tools menu (or by hitting the letter "t" on the keyboard). The Wipe command on the Tools menu allows you to reveal portions of the older image. You can also assign random colors (View menu) to help identify subtle, pixel-by-pixel differences.

## Progressive Mode & Priority Bucket Order (*killeroo-party*)

Gelato also has a "progressive rendering" mode that executes a preview render and then a full quality render to the same iv window. This gives you an immediate, but rough, feel for what the final image will look like before "filling in the details" of the shading.

The progressive mode is especially useful when used in conjunction with the user priority bucket order feature. If you are interested in a particular section of the image, but aren't exactly sure where it is, you can use the preview image to guide you in drawing the crop window for priority rendering.

The following batch files will render the image in progressive mode:

- ❑ progressive-1920.bat (Windows)
- ❑ progressive-1920.sh (Linux)

You can also use this scene to experiment with Gelato's user-priority bucket order feature. Render the image in progressive mode. When iv opens, go to the Tools menu and change the "Zoom" setting to "Select." Then go to the Edit menu and ensure the "Prioritize Selection" button is toggled.

If the preview image has not completed, wait for it to do so. When the full-quality render begins, click and drag out a crop window box anywhere in the image window where the full-quality buckets have not yet been rendered. Gelato will render the area within the crop window next. You can create multiple crop windows and Gelato will render the buckets in the most recently created before going back to the previous one in sequence. You can also simply click on a spot instead of dragging out a crop window and Gelato will render the buckets spirally, outward from that point.

Priority bucket order works on any render, not just those done in progressive mode. If rendering a Maya scene with Mango, you must render to the iv window, and not to Maya's Renderview, for the priority bucket order to work.

## Alternative Shading Modes (*killeroo-party*)

### KeyFillRim Shader

You can also substitute one of two alternative shading modes for the Gelato shaders to achieve faster preview renders. The first of these alternative shaders is the keyfill-rim shader. This shader uses three, automatically placed lights with no shadows. This is useful for determining the placement and view of geometric objects in the scene, but is (obviously) not useful for previewing lighting or shadows.

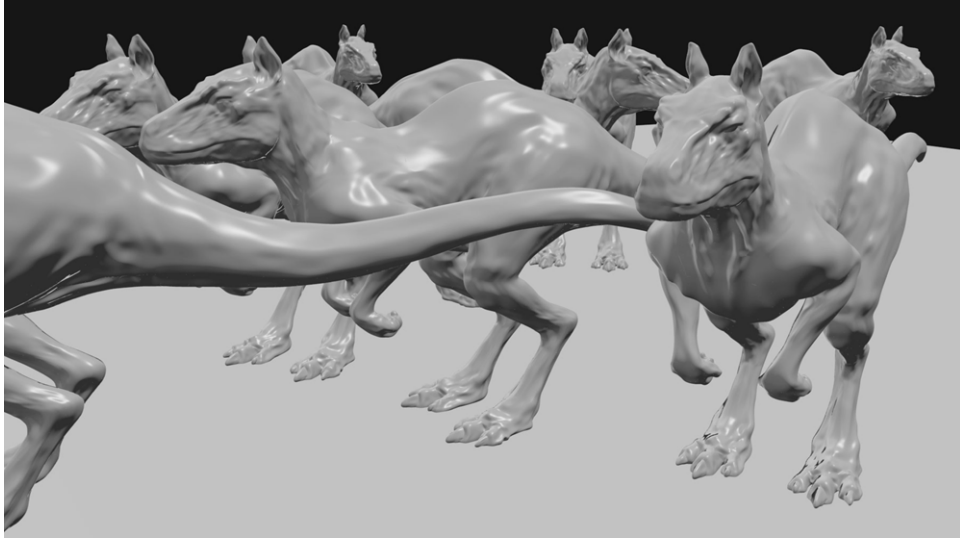


Figure 4. Killeroo-party rendered using the keyfillrim shader

The following batch files render the image using the default surface shader at 1024x768 or 1920x1200 resolution:

- ❑ unshaded1-1024.bat (Windows)
- ❑ unshaded1-1024.sh (Linux)
- ❑ unshaded1-1920.bat (Windows)
- ❑ unshaded1-1920.sh (Linux)

The batch files produce an image file named `killerooparty.tif`. (Note: all the batch files in this directory produce an image file with this name. If you execute another batch file, the old image file will be overwritten. If you wish to save the image, rename it before running another batch file.)

The directory includes reference images at the two resolutions, `unshaded1-1024ref.tif` and `unshaded1-1920ref.tif`.

Rendering times:

- ❑ 1024x768 resolution: 12.9 seconds
- ❑ 1920x1200 resolution: 35.7 seconds

## DefaultSurface Shader

The second alternative substitutes a single, default hardware shader for all the surface shaders used in the scene. This renders the image very quickly.

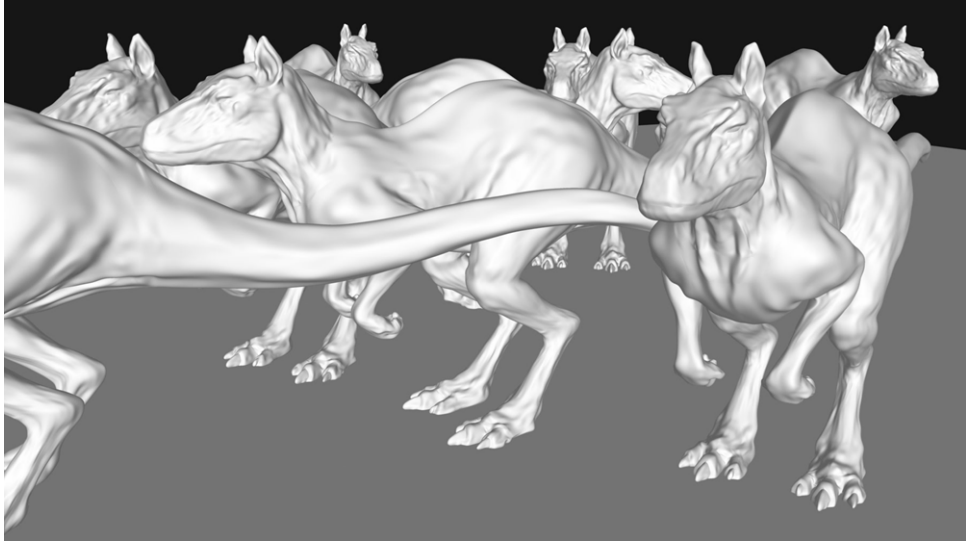


Figure 5. Killeroo-party rendered using the defaultsurface shader

The following batch files render the image using the defaultsurface shader at 1024x768 or 1920x1200 resolution:

- ❑ unshaded2-1024.bat (Windows)
- ❑ unshaded2-1024.sh (Linux)
- ❑ unshaded2-1920.bat (Windows)
- ❑ unshaded2-1920.sh (Linux)

The batch files produce an image file named killerooparty.tif. (Note: all the batch files in this directory produce an image file with this name. If you execute another batch file, the old image file will be overwritten. If you wish to save the image, rename it before running another batch file.)

The directory includes reference images at the two resolutions, unshaded2-1024ref.tif and unshaded2-1920ref.tif.

Both the defaultsurface shader and the keyfillrim shader can be used in conjunction with the regular preview mode for even faster renders.

Rendering times for defaultsurface shader:

- ❑ 1024x768 resolution: 12.9 seconds
- ❑ 1920x1200 resolution: 34.2 seconds

## Simple Motion Blur (*killeroo-mb*)

This scene is a simple example of Gelato's motion blur. It is a single killeroo moving horizontally across the frame.



Figure 6. Killeroo-mb

Killeroo-mb.bat (Windows) or killeroo-mb.sh (Linux) will launch the render. They will create an image file called killeroo-mb.tif. A reference image, killeroo-mbref.tif is in the directory as well.

Rendering time:

□ 46.0 seconds

## Multi-Segment Motion Blur (*msmb-points-explode*)

The scene in Figure 7 demonstrates multi-segment motion blur, a method for rendering motion blur for objects that are not moving linearly. This technique allows the user to slice up the motion blur time sample to correctly blur objects that are traveling in an arc or other non-linear path from shutter open to shutter close.

Unlike other renderers, Gelato has no limits on the number of segments you may subdivide the time sampling. Also note that increasing the samples, set to 64 in this case, has only a marginal impact on the time it takes to render due to Gelato's architecture that makes sampling a low-cost operation.



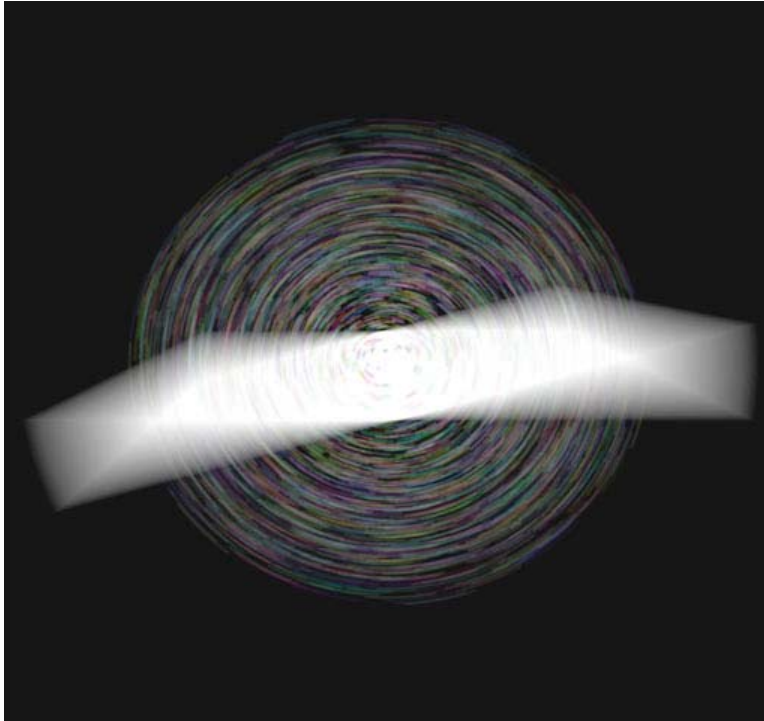


Figure 7 MSMB

Executing either MSMB.bat (Windows) or MSMB.sh (Linux) in the msmb-points-explode directory will generate the image file MSMB.tif. A reference image, MSMB-ref.tif is in the directory as well. The run.pyg file contains the scene information and settings.

Rendering time:

■ 10.0 seconds

## Production Scene (*trooperoo*)

This scene, consisting of 8000 killeroos ready for battle, represents the geometric complexity you might find in a production CG film frame. Each killeroo is the full model. There is no instancing or use of low-res models for the killeroos in the background. This image also uses vertical bucket order instead of the usual horizontal order. This feature of Gelato reduces memory footprint and other system demands for frames that have a horizon or a lot of geometry along the X-axis.



Figure 8. Trooperoo

The image can be rendered using the following batch files. The resolution of the images will be 1024x768 or 1920x1200 pixels respectively. The batch files called “no-display” render the image to a file without running Gelato’s image viewer.

- ❑ trooperoo-1024.bat
- ❑ trooperoo -1920.bat
- ❑ trooperoo -1920-nodisplay.bat
- ❑ trooperoo -1024.sh
- ❑ trooperoo.1920.sh
- ❑ trooperoo -1920-nodisplay.sh

These batch files do not generate the shadowmap; instead they use a pre-generated shadowmap file (key.sm). If you wish, you can regenerate the shadowmap with `make_shadowmap.bat` or `make_shadowmap.sh`. This will approximately double the rendering time.

The batch files will create an image file `trooperoo.tif`. Also contained in the directory is a reference image, `trooperoo-ref.tif`.

Rendering times:

- ❑ 1024x768 resolution: 14 minutes, 55 seconds
- ❑ 1920x1200 resolution: 17 minutes, 47 second



## Fur (*hairyMonster*)

This image of a monster is a good demonstration of Gelato's ability to render hair and fur.



Figure 9 HairyMonster

The hairy monster image can be rendered by running the `hairymonster.bat` (Windows) or `hairymonster.sh` (Linux) files in the `hairyMonster` directory. These will generate an image file `hairymonster.tif`. The image `hairymonster-ref.tif` is the reference image.

The batch files use the textures in the `\texture` subdirectory. If you are conducting comparison tests with other renderers, these textures should work with those renderers.

There are also textures in the `\geltx` subdirectory. You may alter the `.rib` file to use these with Gelato. Since they have 64x64 tiles they will be more efficient than the ones used by the batch files, but they may not work with other renderers.

If you are conducting comparison tests, see the `makefile` file in the `\shaders` subdirectory for information on how to build shaders for other renderers.

Rendering time:

- 15 minutes, 18 seconds

## Depth of Field (*DOF*)

This image shows Gelato's depth of field feature.



Figure 10 DOF

The DOF image can be rendered by running the DOF.bat (Windows) or DOF.sh (Linux) files in the DOF directory. These will generate an image file called DOF.tif. There is also a reference image, DOF-ref.tif, in the directory.

There is no shadowmap in this render because the shadows are raytraced, as are the reflections.

Rendering time:

□ 19 minutes, 45 seconds

## Ambient Occlusion (*ambocc*)

This test demonstrates Gelato's ambient occlusion capabilities.



Figure 11 ambocc

The ambient occlusion image can be rendered by running the `ambocc.bat` (Windows) or `ambocc.sh` (Linux) files in the `ambocc` directory. These will generate an image file called `ambocc.tif`. There is also a reference image, `ambocc-ref.tif`, in the directory.

Rendering time:

- 1 minute, 4 seconds.

## Mango Render (*SloppySam*)

This test demonstrates Gelato's Mango plug-in to Alias Maya.



Figure 12 SloppySam

This test is run by opening SloppySam.mb within Maya. The image is ready to render with Gelato and will render to Maya's Render View window. You can also change the render settings to have it render to Gelato's image viewer (iv). The render can also be launched without using Maya by running the batch file SloppySam.bat (Windows) or SloppySam.sh (Linux).

SloppySam was created entirely with Maya assets. He can be rendered with any Maya-compatible renderer.

Note that in Windows the brightness of the image you render may appear different than the one shown here, especially if you render to the Maya Render View window. This is due to the gamma setting.

Mango gives you the option of choosing one of several options for Anti-Aliasing quality in the Render Global settings. The preset options are:

- Rough Preview Quality (low quality, fast renders)
- Preview Quality (better quality, somewhat slower)
- Intermediate Quality (mid-quality, somewhat slower still)
- Production Quality (best quality, slowest speed)
- Custom (user configured)

The default setting for SloppySam.mb is production quality. You can change this, and any other Render Global settings, in the Maya Render Global window, under the Gelato tab. The current options will be preserved when you save the scene.

Mango ships with Gelato versions of many of the Maya Hypershade nodes. Shader networks constructed entirely of supported nodes will render using Gelato. Not all nodes that ship with Maya are supported, however. We are continually adding new nodes with each release of the product. Check the Gelato release notes for the version you have installed to see the current list of supported nodes.

Rendering time:

- 35 seconds

## Subsurface Scattering in Mango (*SloppySam*)

You can use the SloppySam files to test subsurface scattering. Select Sam's skin surface and go to the Blinn shader tab in the Attribute editor. Open Gelato → Subsurface Scattering and check the box to turn on Subsurface Scattering. Render the image and you should get the following with the default parameters:

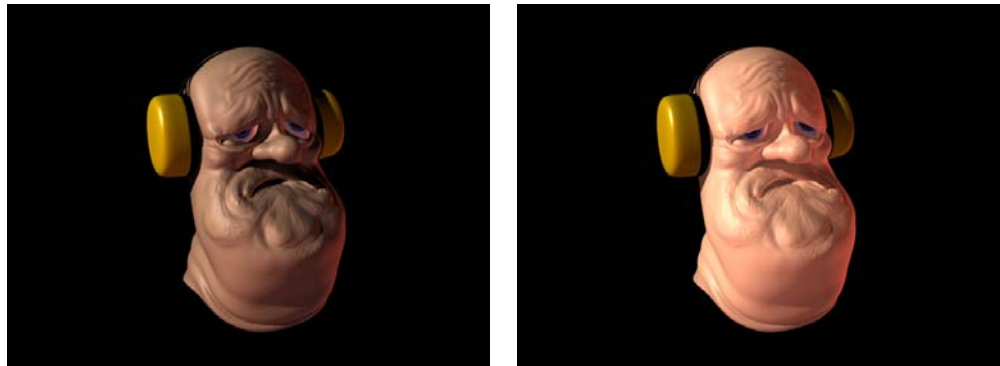


Figure 13 Subsurface Scattering Off (left) and On (right)

You can adjust the parameters to get different results.

Rendering time:

- First Pass: 40 seconds
- Beauty Pass: 2 minutes, 21 seconds

## Caustics in Mango (*Caustics*)

This test demonstrates Gelato 1.2's new caustics feature as well as provides another example of a Maya scene. Unlike the Sloppy Sam scene, this one uses a few Gelato shaders and will not render properly using other Maya-compatible renders.

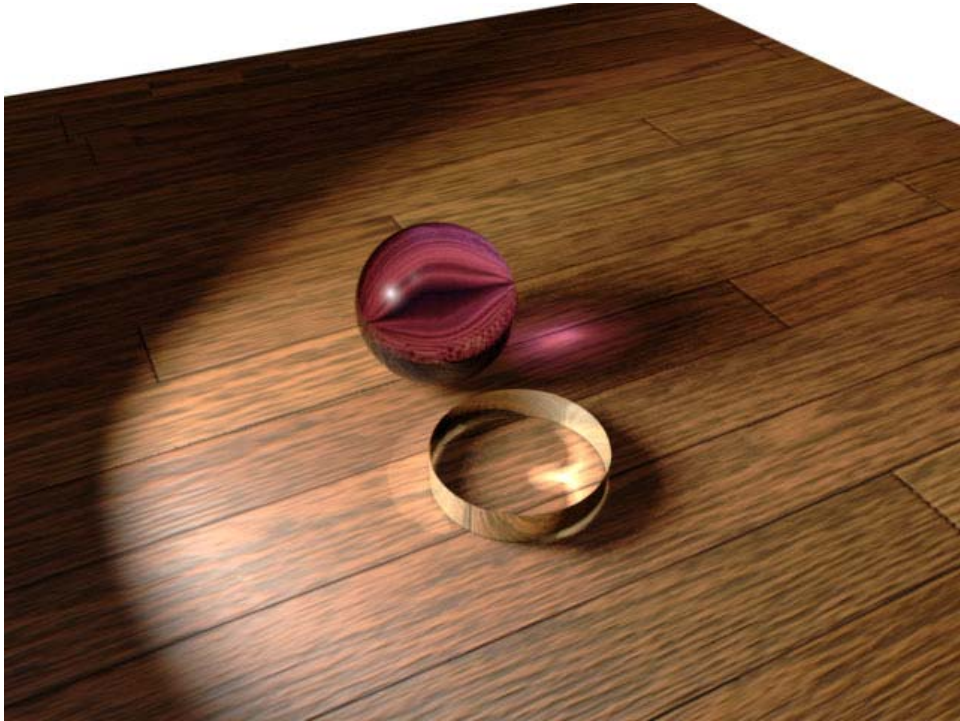


Figure 14 Caustics

This test is run by opening `caustics.mb` within Maya. The image is ready to render with Gelato and will render to Gelato's image viewer (iv). The caustics feature does not work with Maya's Render View window, which will only display the first pass of the renderer, not the completed image. The render can also be launched without using Maya by running the batch file `caustics.bat` (Windows) or `caustics.sh` (Linux). Again, you may need to adjust the gamma setting if the image is too dark—a likely situation in Windows where you will want to adjust the gamma from 1.0 to 2.2. This is simple to do in Gelato iv; simply hit the `-/=` buttons.

Rendering times:

- ❑ First Pass: 8.5 seconds
- ❑ Second Pass: 25 seconds

## Mango Render (*Lab*)

This test is a more complex example of a Maya scene, with multiple lights, transparency, and larger texture files.

This test is run by opening `lab.mb` within Maya. The image is ready to render with Gelato and will render to Gelato's image viewer (iv). You can also change the render settings to have it render to Maya's Render View window. Rendering to Gelato's iv is faster than rendering to Maya's Render View. The render can also be launched without using Maya by running the batch file `caustics.bat` (Windows) or `caustics.sh` (Linux). Again, you may need to adjust the gamma setting if the image is too dark—

a likely situation in Windows where you will want to adjust the gamma from 1.0 to 2.2. This is simple to do in Gelato iv; simply hit the  $\neq$  buttons.

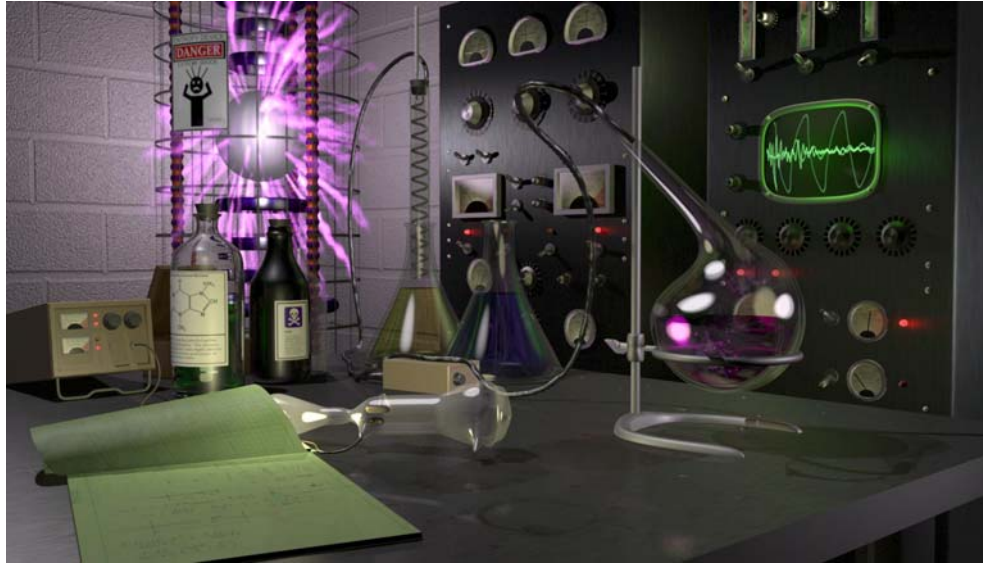


Figure 15 Lab

Rendering time:

- ❑ 4 minutes, 18 seconds

---

## Running Your Own Test Cases

You are not restricted to the above test cases and can create your own test cases for Gelato. When doing so, however, we recommend you adhere to the following guidelines:

When comparing Gelato's performance with that of other renderers:

- ❑ Use identical systems (or better yet the same system) for all the renderers.
- ❑ Do not benchmark performance comparisons on extremely simple scenes (e.g., a handful of simple shapes and textures, and a single light). The overhead in loading assets such as texture files often exceeds the render time for such scenes and you will not see a meaningful performance differential between Gelato and other renderers.

# Technical Support

A detailed technical description of Gelato, including the complete API specification, is available in the Gelato Technical Reference. This document is part of the Gelato download. A Mango User's Guide is also included in the documentation package.

If you experience problems, errors, bugs, or inadequate performance when testing Gelato, do not hesitate to contact [gelatosupport@nvidia.com](mailto:gelatosupport@nvidia.com). Often, such problems can be quickly resolved with simple changes to the system settings.

We recommend that you contact us before you begin testing Gelato. This will help minimize response time when you contact us with questions or for assistance.

If you experience slow performance or low quality images, please contact Technical Support.



# NVIDIA Contacts

## **Press**

David Higham  
Professional Products PR Manager  
[dhigham@nvidia.com](mailto:dhigham@nvidia.com)  
+1 (408) 486 8287  
[http://www.nvidia.com/page/press\\_room.html](http://www.nvidia.com/page/press_room.html)

## **Professional Solutions Group Sales**

[gelatosales@nvidia.com](mailto:gelatosales@nvidia.com)

## **Technical Support**

[gelatosupport@nvidia.com](mailto:gelatosupport@nvidia.com)

## **Gelato Product Questions**

[gelatoinfo@nvidia.com](mailto:gelatoinfo@nvidia.com)

## Credits

The killeroo model was created by Headus studios in Australia and is used with permission.

Some textures used in the included scenes are taken from dvGarage's Surface Toolkit and its Reflection Toolkit and are distributed here with permission. These and other 3D development tools are available from <http://www.dvgarage.com/index.php>.



## Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. NVIDIA Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

## Trademarks

NVIDIA, the NVIDIA logo, NVIDIA Quadro, Gelato, and Mango are trademarks or registered trademarks of NVIDIA Corporation in the United States and other countries. Windows® is a trademark of Microsoft Corporation. Maya® is a trademark of Alias Systems Corp. 3ds Max® is a trademark of Autodesk, Inc. Other company and product names may be trademarks of the respective companies with which they are associated.

## Copyright

© 2005 by NVIDIA Corporation. All rights reserved



**NVIDIA.**

NVIDIA Corporation  
2701 San Tomas Expressway  
Santa Clara, CA 95050  
[www.nvidia.com](http://www.nvidia.com)