

# Accelerad Documentation

*You'll want to read this!*

## Contents

### Getting Started

- [Before you begin](#)
- [Confirm that Accelerad runs on your computer](#)
- [Use Accelerad in your workflow](#)

### Release Notes

- [Radiance versions](#)
- [Missing files from RAYPATH](#)
- [Missing library files \(.dll, .so, and .dylib\)](#)
- [Graphics driver crashes](#)
- [Supported GPUs](#)
- [Disabling GPU acceleration](#)
- [Slow first run](#)
- [Distance to origin](#)
- [Unexpected high and low values](#)
- [Patchy or blotchy rendering](#)
- [BSDF Use](#)
- [Antimatter](#)

### Information and Help

#### Command Line Arguments

#### Updates

- [Since Version 0.6 beta](#)
- [Since Version 0.5 beta](#)
- [Since Version 0.4 beta](#)
- [Since Version 0.3 beta](#)
- [Since Version 0.2 beta](#)
- [Since Version 0.1 beta](#)
- [Since Version 0 beta](#)

#### Current Limitations

#### AcceleradRT

##### Features

##### AcceleradVR

##### Command Line Arguments

## Welcome to Accelerad!

Accelerad is a free suite of programs for fast and accurate lighting and daylighting analysis and visualization. It was originally developed by Nathaniel Jones at the MIT Sustainable Design Lab and continues to be developed independently. It is modeled after the industry standard Radiance software suite developed by Greg Ward at Lawrence Berkeley National Laboratory. These algorithms are accelerated up to twenty times faster using OptiX™, a ray tracing engine built for the graphics processor unit (GPU). In order to allow for smooth adoption among Radiance users and software developers, Accelerad maintains compatibility with Radiance file formats, materials, and command-line arguments.

# Getting Started

## Before you begin

The Accelerad programs are intended to replace the standard versions that come with the default installation of Radiance. Before you install Accelerad:

1. Verify that you have a CUDA-enabled GPU with compute capability 5.0 to 7.5. A list of currently-supported GPUs is available at <https://developer.nvidia.com/cuda-gpus>. If you are unsure what GPU you have:
  - a. On Windows, right click on *Computer* and select *Properties > Device Manager > Display adapters*.
  - b. On Linux, use the command `lshw -C display` to list devices.
  - c. On Mac, choose *About this Mac* from the Apple menu and go to the *Displays* tab.
2. Download and install Radiance if it is not already present. The latest version of Radiance for all platforms can be found at <https://github.com/NREL/Radiance/releases>.
3. Download and install the latest graphics driver for your GPU. Drivers can be downloaded from <http://www.nvidia.com/Download/index.aspx>.
  - a. On Windows, driver version 418.81 or later is required.
  - b. On Linux, driver version 418.30 or later is required.
  - c. On Mac, no supported driver is currently available.

## Confirm that Accelerad runs on your computer

To confirm that Accelerad is successfully installed on your computer, run the included Windows *.bat* or Unix *.sh* files in the *demo* folder. By default, this folder is located at *C:\Program Files\Accelerad\demo* on Windows and */usr/local/accelerad/demo* on Linux.

- Running *test\_accelerad\_rpict* should produce a high dynamic range image file named *test\_rpict.hdr*.
- Running *test\_accelerad\_rtrace* should produce tabular data in a file named *test\_rtrace.txt*.
- Running *test\_accelerad\_rcontrib* should produce tabular data in a file named *test\_rcontrib.txt*.
- Running *test\_AcceleradRT* should open an interactive false color view of a simple scene (available only on Windows).

On Linux and Mac, it will first be necessary to add the Accelerad *bin* folder to the PATH and LD\_LIBRARY\_PATH environment variables, and to add the Accelerad *lib* folder to the RAYPATH environment variable. On Windows, this will be done automatically by the installer. On Linux or Mac, add the following to the end of your *~/.profile* file and edit the paths as necessary if you choose a custom install location:

```
export PATH=/usr/local/accelerad/bin:$PATH
export RAYPATH=/usr/local/accelerad/lib:$RAYPATH
export LD_LIBRARY_PATH=/usr/local/accelerad/bin:$LD_LIBRARY_PATH
```

## Use Accelerad in your workflow

This suite contains Accelerad versions of several Radiance programs:

The Accelerad program ...	... replaces the Radiance program ...
<i>accelerad_rpict</i>	<i>rpict</i>
<i>accelerad_rtrace</i>	<i>rtrace</i>
<i>accelerad_rcontrib</i>	<i>rcontrib</i>
<i>accelerad_rfluxmtx</i>	<i>rfluxmtx</i>
<i>accelerad_genBSDF</i>	<i>genBSDF</i>

For Windows, these programs are installed by default to *C:\Program Files\Accelerad\bin*. However, different Radiance tools have different expectations about where Radiance programs will be found. Depending on the tools you use, **you may need to move or rename some files**. You can run Accelerad in one of three ways:

**Within an application:** In your applications that use one of the above Radiance programs, change the paths to refer to the corresponding Accelerad program.

**Replace the original programs:** If your application does not allow you to edit the Radiance application names:

1. Copy the contents of the Accelerad *bin* folder into your Radiance *bin* folder.
2. In the Radiance *bin* folder, rename the Accelerad programs with the names of the corresponding Radiance programs and replace the original program files. You may want to keep backup copies of the original Radiance files elsewhere.
3. Copy the .ptx files from the Accelerad *lib* folder to the Radiance *lib* folder.

**From the command line:** If you use Radiance from the command line, you can simply address your commands to the Accelerad programs instead of the corresponding Radiance programs. Be sure that the Accelerad *bin* and *lib* folder are found in your PATH and RAYPATH environment variables, respectively. On Linux, the Accelerad *bin* folder must also be found in the LD\_LIBRARY\_PATH environment variable.

# Release Notes

## Radiance versions

Accelerad is designed to mimic the behaviour of Radiance and has been tested in comparison with Radiance version 5.2. A number of other sources exist for the Radiance executables, including DIVA, DAYSIM, and IES VE. However, these versions may not be updated as frequently and have not been tested with Accelerad.

## Missing files from RAYPATH

Accelerad locates certain files using Radiance's RAYPATH environment variable. The message "File <filename> not found in RAYPATH" indicates that the folder containing the required file has not been added to the RAYPATH.

In Windows, this error may occur when RAYPATH has been defined in both the system and current user environment variable sets. In this case, the current user's RAYPATH will override the system RAYPATH created by the Accelerad installer. It is recommended to merge all paths from the current user's RAYPATH into the system RAYPATH and then delete RAYPATH from the current user's environment variables.

To edit the RAYPATH in Windows, right click on *This PC > Properties > Advanced system settings > Environment Variables*. The RAYPATH variable may be in either the user variables or system variables set, but should not be in both. On Linux and Mac, environment variables can be edited in the hidden `~/.profile` file. Make sure that the RAYPATH variable contains paths to both the Accelerad *lib* and Radiance *lib* folders. The exact appearance will vary depending on the Radiance installations present, but a typical RAYPATH value looks like:

- Windows  
.;C:\Program Files\Accelerad\lib;C:\Program Files\Radiance\lib
- Linux and Mac  
.:usr/local/accelerad/lib:usr/local/radiance/lib

## Missing library files (.dll, .so, and .dylib)

If you do not have Visual Studio 2013 installed on your computer, you may see a message that "The program can't start because MSVCR120.dll is missing from your computer." If you see this error, visit <http://www.microsoft.com/en-us/download/details.aspx?id=40784> to download an installer for the missing file.

Linux and Mac computers may complain of missing .so or .dylib files if the Accelerad *bin* folder has not been added to the LD\_LIBRARY\_PATH environment variable. The path can be added by editing the `~/.profile` file.

## Graphics driver crashes

If you use a GPU that is not in Tesla Compute Cluster (TCC) mode with Windows, you may experience an unresponsive screen leading to timeout detection and recovery (TDR) while

running the software. This will be accompanied by a message saying “Display driver stopped responding and has recovered.”

By default, the Windows OS will end processes on the GPU (essentially rebooting it) after two seconds of unresponsiveness, which can happen when the GPU is processing a large amount of data. Possible remedies include:

- If more than one graphics card is available, set `CUDA_VISIBLE_DEVICES` to use only the GPUs that are not connected to monitors.
- Increase the TDR delay by adding a registry key. This is not recommended for novices.
- Test on a more powerful graphics card or a card which is not connected to a monitor. Tesla cards or other cards with TCC mode enabled are not affected by TDR.

## Supported GPUs

Accelerad is targeted toward newer graphics hardware with RTX capability. It is compatible with graphics hardware with CUDA compute capability 5.0 or greater (“Turing”, “Volta”, “Pascal”, or “Maxwell”). As of version 0.7 beta, older hardware (“Kepler” and “Fermi”) is no longer supported.

If you receive an error stating “A supported NVIDIA GPU could not be found” even though your machine has a supported GPU, you must update to the latest graphics driver. Drivers can be downloaded from <http://www.nvidia.com/Download/index.aspx>.

## Disabling GPU acceleration

You can reproduce the normal behavior of Radiance on the CPU in Accelerad by providing the argument `-g-` to most Accelerad programs. When this argument is given, Accelerad will not use the GPU for any calculations, and normal Radiance algorithms will be used.

AcceleradRT requires use of the GPU and does not support `-g-`.

## Slow first run

In order to achieve high speeds, Accelerad compiles certain GPU code specifically for the GPUs available on your computer and caches it for later use. This may result in longer startup times the first time Accelerad is run or a new model is created. On subsequent runs, the cached programs are used, resulting in a much shorter startup time.

## Distance to origin

Because ray intersection calculations on the GPU use single-precision floating point arithmetic, some impression is likely for geometry that is located far from the origin. This can lead to missing geometry or trace depth exceeded errors, which appear as warning messages. Additionally, warnings will appear as red pixels in *accelerad\_rpict* or white pixels in AcceleradRT. Accelerad makes an effort to avoid these issues, but the result is that surfaces far

from the origin may appear incorrectly. As a solution, scale the model in meters and position the model so that relevant geometry is less than 100 meters from the origin.

## Unexpected high and low values

Accelerad flags computation errors with brightly-colored results. If you see results with much higher radiance values than expected, check the command-line output for error warnings.

Low or patchy radiance results may indicate insufficient ambient coverage. You may correct this by varying the ambient parameters, including the additional Accelerad command line arguments.

## Patchy or blotchy rendering

Patchy or blotchy rendering may occur if the irradiance cache provides insufficient coverage of the scene. This algorithm may occasionally to undersample diffuse lighting in many-bounce paths. The size of the irradiance cache in *accelerad\_rtrace* or *accelerad\_rpict* is fixed by the *-ac* parameter. To increase ambient coverage, increase the *-ac* or *-aa* parameter value, or decrease the *-ar* argument. To turn off irradiance caching, set the ambient accuracy *-aa* to zero.

Irradiance caching is not used in *accelerad\_rcontrib* or AcceleradRT.

## BSDF Use

Accelerad includes *accelerad\_rcontrib*, which can be used to create bidirectional scattering distribution functions (BSDFs) by *rtfluxmtx* or *genBSDF*. However, the current Accelerad version does not take BSDF files as input. Three-phase method simulations using Accelerad produce accurate results because this simulation type does not cast rays through the BSDF material. However, five-phase method simulations may produce lower illuminance results in Accelerad because interreflection of the direct solar component within the BSDF material is not taken into account.

## Antimatter

Accelerad allows the use of up to 32 antimatter materials. However, the use of antimatter can lead to slow-downs in rendering, reduce the number of ray reflections that occur, or cause trace depth exceeded errors. Low ambient lighting values may result inside antimatter volumes. It is recommended to model holes in surfaces explicitly rather than use antimatter in models.

## Information and Help

The Accelerad homepage can be found at:

<http://nljones.github.io/Accelerad/>

A forum for questions, bug reporting, and update announcements can be found at:

<https://groups.google.com/forum/?hl=en#!forum/accelerad-users>

## Command Line Arguments

In addition to the command line arguments typically used by [rpict](#), [rtrace](#), and [rcontrib](#), Accelerad introduces optional new command line arguments required for the GPU implementation of certain algorithms. These new parameters are summarized below:

Argument	Description	Default
-g	Enable or disable GPU ray tracing. A value of -g- will cause the Accelerad programs to revert to normal Radiance behavior without using the GPU.	-g+
-gv <i>level</i>	Set the verbosity of GPU debugging to <i>level</i> . Level 0 produces the fastest output but will not display GPU errors which could affect results. Level 1 reports GPU errors. Levels 2 and 3 provide additional GPU stats but result in longer computations.	0
-al <i>stride</i>	Set the spacing between seed point pixels for ambient sampling to <i>stride</i> in rtrace only. A value of zero will cause all pixels to be considered. This option is ignored when the -az option is used.	0
-ag <i>N</i>	Set number of ambient divisions for final gather infill to <i>N</i> . When -aa is non-zero, <i>N</i> ambient samples will be taken at points not covered by the precomputed irradiance cache. A value of -1 will cause the value to be copied from -ad.	-1
-az <i>res</i>	Set the number of seeds points for ambient samples to take around the circumference of a sphere based at the view point to <i>res</i> in rtrace only. A value of zero will cause view-dependent seeding to be used instead. Thus, zero should not be used in combination with the -S option in which a view file changes the view direction between frames.	0
-ac <i>N</i>	Set the number of k-means clusters for ambient calculation to <i>N</i> .	4096
-an <i>N</i>	Set the maximum number of k-means iterations to <i>N</i> . Larger values can cause k-means calculation to take longer but will generate more accurate ambient results.	100
-at <i>thresh</i>	Set the k-means threshold to <i>thresh</i> . This is the fraction of seeds that must change cluster in order for k-means iteration to continue. Smaller values can cause k-means calculation to take longer but will generate more accurate ambient results.	0.05
-ax <i>wt</i>	Set the weighting factor for position in k-means error calculation to <i>wt</i> . Small values concentrate more ambient calculations around edges where ambient gradients are likely to be large.	1.0

# Updates

## Since Version 0.6 beta

- The ray tracing engine is updated to OptiX™ 6.0.0 using CUDA 10.0 with support for RTX graphics cards.
- AcceleradRT is included in the Windows distribution of Accelerad.
- Added support for instances.
- Reduced memory overhead for meshes.
- Added support for *reinhartb.cal*, *disk2square.cal*, *klems\_half.cal*, *klems\_quarter.cal*, and all *rfluxmtx* h options.
- Fixed program name checking in *rfluxmtx* and *genBSDF* to avoid errors on some machines.
- Added support for *rtrace* -orRxX outputs.
- Number of allowed antimatter materials increased to 32.
- Various improvements to error handling.
- Fixed compatibility issue with Honeybee standard error formatting.
- Fixed bug in which ray tracing fails for empty scenes.
- Fixed bug in which end of file message appeared erroneously in *rcontrib*.
- Fixed bug in which some contribution indices might not be recorded in *rcontrib*.
- Fixed bug in ambient super sampling.
- Fixed bug in aft clipping plane for scenes with transparent surfaces.
- Adjustable color scale added to AcceleradRT.
- Radiance and irradiance visualization added to AcceleradRT.
- Walking mode added to AcceleradRT.
- Fixed bug that caused sky to render too bright in AcceleradRT.
- Fixed bug in DGP calculation when no task area is selected in AcceleradRT.
- Use of *-g* parameter to set GPU stack size is deprecated and has no effect.
- Use of *-t* parameter to set timeout callback is deprecated and has no effect.
- Support discontinued for SM 2.X (Fermi) and SM 3.X (Kepler) GPUs.
- Support discontinued for Mac due to lack of available drivers.
- Support discontinued for VCA.

## Since Version 0.5 beta

- Added Accelerad versions of *rcontrib*, *rfluxmtx*, and *genBSDF* programs.
- Added support for Russian roulette ray extinguishing with negative *-lr* arguments. Using Russian roulette may have a negative effect on speedup and require a larger *-g* argument.
- Added support for *src\_phi2* and *src\_phi4* to be used in *brightdata* calls with *source.cal*.
- Slight performance improvement by ignoring materials that are defined but not used.
- Fixed bug in shadows in antimatter regions.
- Fixed bug in ambient calculations at low angles.
- Fixed bug that allowed ray weights greater than one in irradiance calculations.



### Since Version 0.4 beta

- The ray tracing engine is updated to OptiX™ 3.9.1 final release using CUDA 7.5, which provides support for the Pascal architecture.
- Includes bug fixes from Radiance version 5.0.a.11.
- Added support for antimatter, although first string argument must be void.
- Added support for ambient inclusion and exclusion (*-ae*, *-aE*, *-ai*, *-aI*).
- Added support for *.cal* files created with GenCumulativeSky.
- Added support for Mark Stock's *utah.cal*.
- Fixed bug in coordinate mapping for data from *.dat* files.
- Fixed bug in rendering glass with nothing behind it.
- Fixed bug in rendering shadows of trans materials.
- Fixed bug in rpict images with odd numbers in dimensions.

### Since Version 0.3 beta

- The ray tracing engine is updated to OptiX™ 3.8.0 final release using CUDA 7.0.
- Includes bug fixes from Radiance version 5.0 final release.
- Added support for ambient supersampling (*-as*) so long as *-aa > 0*.
- Added support for Monte Carlo random seeds (*-u*).
- Added support for motion blur in rpict (*-pm*).
- Added support for depth of field blur in rpict (*-pd*).
- Added *-ag* parameter for faster final gather step for debugging.
- Added support for cone, cup, cylinder, tube, and ring objects in *.rad* files.
- Added support for illum materials with non-opaque or void string arguments.
- Added support for *corr*, *boxcorr*, and *cylcorr* functions from *source.cal*.
- Improved handling of normals modified by texfunc objects.
- Brightdata from *source.cal* can now be applied as a material to sources.
- Ignored triangulation errors with a warning.
- Printed an extra line in rpict to handle a bug found in IES<VE>.
- Fixed bug in display of GPU global memory size.
- Added support for Linux and Mac.

### Since Version 0.2 beta

- The ray tracing engine is updated to OptiX™ 3.8.0 beta.
- Includes bug fixes from Radiance version 5.0a.
- Major speed improvements for ambient calculations in rpict and rtrace.
- Added support for spheres and bubbles in *.rad* files.
- Added support for alias objects in *.rad* files.
- Function objects no longer need to be direct parents of the objects they modify, but still only one function object is supported per modified object.
- Fixed issues in creation of non-square images that caused fatal errors.

- Renamed *-am* argument to *-an* to avoid conflict with new Radiance version 5.0a parameter.
- The *-x* and *-y* arguments are no longer required for *rtrace*.
- Removed limit on *-ad*.
- Removed limit on the number of TCC-enabled graphics cards that can be used.
- Reduced default OptiX stack size hint *-g* to 4096 bytes due to smaller stack size requirement.
- Improved reporting of errors.
- Fixed incorrect reporting on number of rays cast by *rpict* when GPU is used.

### Since Version 0.1 beta

- The ray tracing engine is updated to OptiX™ 3.7.0 beta 3, which provides support for CUDA 6.5 and SM 3.7 GPUs, including the Tesla K80.
- Users are no longer required to download and install CUDA.
- Includes bug fixes from Radiance version 4.3.a.2.
- Radiance mesh objects are supported.
- Changed new *-at* parameter to work independently of *-aa* and *-ar*.
- Fixed a bug that prevented caching of ambient values for certain specular parameters.
- Fixed a bug in Gaussian transmission for trans materials.
- Changed error messages and return values to match those of Radiance.

### Since Version 0 beta

- The ray tracing engine is updated to OptiX™ 3.6.3, which provides improved support for Maxwell-based SM 5.2 GPUs, including the GeForce GTX 980.
- Includes bug fixes from Radiance version 4.2.2.
- Enabled an experimental algorithm for parallel irradiance caching in *rtrace*.
- Implemented *-l* and *-ld* arguments from *rtrace*.
- Improved handling of not-a-number (NaN) errors in both GPU and standard implementations of *rpict*.
- Fixed a bug in Gaussian transmission for trans materials.
- Added ability to assign geometric transformations to skies defined with *skybright.cal* or *perezlum.cal*.
- Fixed a bug that caused out-of-resources error (cuda error 7) on some machines.
- Fixed a bug that caused stack overflow errors in some cases for geometry located far from the origin.

## Current Limitations

The table below lists currently supported parameters. Unsupported parameters will be ignored.

Category	Supported Parameters	Unsupported Parameters
Direct	-dj, -ds, -dv	-dt, -dc, -dr, -dp
Specular	-ss, -st	
Visibility	-bv, -i, -l	
Ambient	-av, -aw, -ab, -ar, -aa, -ad, -as, -af, -ae, -ai, -aE, -al	-ap, -am
Medium		-me, -ma, -mg, -ms
Rays	-x, -y, -lr, -lw, -ld	
View (rpict only)	-vt, -vp, -vd, -vu, -vh, -vv, -vo, -va, -vs, -vl, -vf	
Pixel (rpict only)	-pa, -pj, -pm, -pd	-ps, -pt
Modifiers (rcontrib only)	-m, -M, -b, -bn, -f followed by <i>disk2square.cal</i> , <i>klems_full.cal</i> , <i>klems_half.cal</i> , <i>klems_quarter.cal</i> , <i>tregenza.cal</i> , <i>reinhart.cal</i> , or <i>reinhartb.cal</i>	
Control	-e, -u, -w rpict only: -S, -o, -z, -t rtrace only: -h, -o followed by o, d, v, w, l, r, R, x, X rcontrib only: -h, -o, -c, -V	rpict only: -r, -ro rtrace only: -n, -o followed by V, W, L, c, p, n, N, s, m, M, t, T

The following Radiance primitive types are currently supported. Unsupported surface types, textures, and patterns will be ignored. Surfaces assigned unsupported materials will also be ignored. Alias types are supported for all categories.

Surfaces	Materials	Textures and Patterns
<ul style="list-style-type: none"> <li>• Polygon</li> <li>• Sphere</li> <li>• Bubble</li> <li>• Cone</li> <li>• Cup</li> <li>• Cylinder</li> <li>• Tube</li> <li>• Ring</li> <li>• Mesh</li> <li>• Instance</li> <li>• Source</li> </ul>	<ul style="list-style-type: none"> <li>• Plastic</li> <li>• Metal</li> <li>• Trans</li> <li>• Glass</li> <li>• Light</li> <li>• Illum</li> <li>• Glow</li> <li>• Spotlight</li> <li>• Antimatter</li> </ul>	<ul style="list-style-type: none"> <li>• Texfunc using <i>tmesh.cal</i></li> <li>• Brightfunc using <i>skybright.cal</i>, <i>perezlum.cal</i>, <i>utah.cal</i>, <i>isotrop_sky.cal</i>, or output from GenCumulativeSky</li> <li>• Colorfunc using <i>utah.cal</i></li> <li>• Brightdata using <i>corr</i>, <i>flatcorr</i>, <i>boxcorr</i>, or <i>cylcorr</i> from <i>source.cal</i></li> </ul>

## AcceleradRT

AcceleradRT, the next generation in daylighting simulation, is an interactive interface for real-time daylighting, glare, and visual comfort analysis. AcceleradRT combines the capabilities of other programs such as rpict, rvu, evalglare, and wxfalsecolor with GPU acceleration.

### Features

In AcceleradRT, you can:

- Change the view position and direction using intuitive mouse and scroll wheel navigation.
- Switch between realistic, greyscale, and false color visualizations.
- Switch between luminance, illuminance, radiance, and irradiance visualizations.
- View live updates of vertical eye illuminance ( $E_v$ ) and daylight glare probability (DGP).
- Select task areas and contrast areas for luminance or illuminance measurement.
- Adjust exposure settings, rendering settings, and camera settings.
- Automatically update the model whenever a new octree file is created.
- Save high-dynamic range (HDR) images at any point while progressive rendering runs.

### AcceleradVR

AcceleradRT includes an experimental feature, AcceleradVR, for experiencing the luminous environment in virtual reality. AcceleradVR uses third party software to stream AcceleradRT's real-time rendering to a smartphone. The stereo image can be viewed with any VR headset, including Google cardboard. To use AcceleradVR:

1. Install Accelerad if it is not already present
2. Install Trinus CB VR server (or a similar screen sharing program) on computer
3. Install Trinus CB VR for iPhone or Android on phone
4. Launch Trinus CB VR on both computer and phone
5. In the Trinus CB VR server window:
  - a. Set compression high using the slider
  - b. Turn off "Fake 3D"
  - c. Adjust the yaw and dead zone as necessary
6. Enter phone IP address (shown in app) into Trinus CB VR
7. Click the triangle in the Trinus CB VR phone app
8. Click the triangle in the Trinus CB VR computer app
9. Launch *C:\Program Files\Accelerad\demo\test\_AcceleradRT.bat*
10. Click the VR button in AcceleradRT app and select the window that opens

Use head movement, your keyboard, or any USB game controller to move through your virtual space. On game controllers and joysticks, hold down Button 2 to switch from conventional navigation to vertical movement. For keyboard navigation, use forward (W), back (S), left (A), right (D), and arrow keys to move up, down, or spin.

## Command Line Arguments

AcceleradRT does not need to be run from the command line, as most settings can be adjusted in the exposure settings, rendering settings, or camera settings. For convenience, advanced users may still adjust parameters via the command line.

For example, specify an initial hemispherical camera view and octree file as follows:

```
AcceleradRT -vp 2.5 -3 2.5 -vd 0 1 0 -vu 0 0 1 -vta -vv 180 -vh 180 -ab  
3 -x 768 -y 768 model.oct
```

AcceleradRT supports *rpict* command line arguments. Because AcceleradRT uses progressive rendering, it will ignore any *-aa* or *-ad* parameters it receives. Additional parameters for specifying the false color scale and visual comfort indicators are summarized below:

Argument	Description	Default
-f	Set whether to use the false color scale. Setting <i>-f+</i> enables the false color scale, while <i>-f-</i> enables true colors.	-f+
-s <i>maximum</i>	Set the color scale maximum to <i>maximum</i> . Set the value to zero to enable auto-selection.	10000.0
-log <i>decades</i>	Set the number of logarithmic scale increments to <i>decades</i> . Set the value to zero to disable logarithmic scaling.	3
-base <i>b</i>	Set the base for the logarithmic color scale to <i>b</i> .	10
-mask <i>minimum</i>	Set the minimum value on the false color scale to <i>minimum</i> . Pixels below this value will not be shaded. Set the value to zero to disable masking.	0.0
-pe <i>exposure</i>	Set the exposure used for saving HDR image output to <i>exposure</i> .	1.0
-T x y <i>angle</i>	Set the task area used for task area luminance and glare calculations. The area will be a circle centered on (x, y) in screen space with an opening angle of <i>angle</i> radians. Set <i>angle</i> to zero when no task area is defined.	0 0 0.0
-C xh yh ah xl yl al	Set the regions to use for measuring contrast. The ratio will be taken between the high circle centered on (xh, yh) in screen space with an opening angle of <i>ah</i> radians and the low circle centered on (xl, yl) in screen space with an opening angle of <i>al</i> radians. Set <i>ah</i> and <i>al</i> to zero when no contrast regions are defined.	0 0 0.0 0 0 0.0