NAME

rtrace - trace rays in RADIANCE scene

SYNOPSIS

rtrace [**options**] [**$EVAR**] [**@file**] **octree**
rtrace [options] −defaults

DESCRIPTION

Rtrace traces rays from the standard input through the RADIANCE scene given by *octree* and sends the results to the standard output. Input for each ray is:

xorg yorg zorg xdir ydir zdir

If the direction vector is (0,0,0), a bogus record is printed and the output is flushed if the *-x* value is unset or zero. (See the notes on this option below.) This may be useful for programs that run *rtrace* as a separate process. In the second form, the default values for the options (modified by those options present) are printed with a brief explanation.

Options may be given on the command line and/or read from the environment and/or read from a file. A command argument beginning with a dollar sign ('$') is immediately replaced by the contents of the given environment variable. A command argument beginning with an at sign ('@') is immediately replaced by the contents of the given file. Most options are followed by one or more arguments, which must be separated from the option and each other by white space. The exceptions to this rule are the boolean options. Normally, the appearance of a boolean option causes a feature to be "toggled", that is switched from off to on or on to off depending on its previous state. Boolean options may also be set explicitly by following them immediately with a '+' or '-', meaning on or off, respectively. Synonyms for '+' are any of the characters "yYtT1", and synonyms for '-' are any of the characters "nNfF0". All other characters will generate an error.

-f*io* Format input according to the character *i* and output according to the character *o*. *Rtrace* understands the following input and output formats: 'a' for ascii, 'f' for single-precision floating point, and 'd' for double-precision floating point. In addition to these three choices, the character 'c' may be used to denote 4-byte floating point (Radiance) color format for the output of values only *(−ov* option, below). If the output character is missing, the input format is used.

Note that there is no space between this option and its argument.

-o*spec* Produce output fields according to *spec*. Characters are interpreted as follows:

o	origin (input)
d	direction (normalized)
v	value (radiance)
w	weight
l	effective length of ray
L	first intersection distance
p	point of intersection
n	normal at intersection (perturbed)
N	normal at intersection (unperturbed)
s	surface name
m	modifier name

If the letter 't' appears in *spec*, then the fields following will be printed for every ray traced, not just the final result. Spawned rays are indented one tab for each level.

Note that there is no space between this option and its argument.

-te *mat* Append *mat* to the trace exclude list, so that it will not be reported by the trace option *(−o*t*).* Any ray striking an object having *mat* as its modifier will not be reported to the standard output with the rest of the rays being traced. This option has no effect unless the 't' option has been given as part of the output specifier. Any number of excluded materials may be given, but each must appear in a separate option.

-ti *mat* Add *mat* to the trace include list, so that it will be considered during the indirect calculation. The program can use either an include list or an exclude list, but not both.

-tE *file* Same as *−te,* except read materials to be excluded from *file.* The RAYPATH environment variable determines which directories are searched for this file. The material names are separated by white space in the file.

-tI *file* Same as *−ti,* except read materials to be included from *file.*

−i Boolean switch to compute irradiance rather than radiance values. This only affects the final result, substituting a Lambertian surface and multiplying the radiance by pi. Glass and other transparent surfaces are ignored during this stage. Light sources still appear with their original radiance values, though the *−dv* option (below) may be used to override this. This option is especially useful in conjunction with ximage(1) for computing illuminance at scene points.

−I Boolean switch to compute irradiance rather than radiance, with the input origin and direction interpreted instead as measurement point and orientation.

−h Boolean switch for information header on output.

-x *res* Set the x resolution to *res.* The output will be flushed after every *res* input rays. A value of zero means that no output flushing will take place.

-y *res* Set the y resolution to *res.* The program will exit after *res* scanlines have been processed, where a scanline is the number of rays given by the *−x* option, or 1 if *−x* is zero. A value of zero means the program will not halt until the end of file is reached.

If both *−x* and *−y* options are given, a resolution string is printed at the beginning of the output. This is mostly useful for recovering image dimensions with *pvalue(1),* and for creating valid Radiance picture files using the color output format. (See the *−f* option, above.)

-dj *frac* Set the direct jittering to *frac.* A value of zero samples each source at specific sample points (see the *−ds* option below), giving a smoother but somewhat less accurate rendering. A positive value causes rays to be distributed over each source sample according to its size, resulting in more accurate penumbras. This option should never be greater than 1, and may even cause problems (such as speckle) when the value is smaller. A warning about aiming failure will issued if *frac* is too large.

-ds *frac* Set the direct sampling ratio to *frac.* A light source will be subdivided until the width of each sample area divided by the distance to the illuminated point is below this ratio. This assures accuracy in regions close to large area sources at a slight computational expense. A value of zero turns source subdivision off, sending at most one shadow ray to each light source.

-dt *frac* Set the direct threshold to *frac.* Shadow testing will stop when the potential contribution of at least the next and at most all remaining light sources is less than this fraction of the accumulated value. (See the *−dc* option below.) The remaining light source contributions are approximated statistically. A value of zero means that all light sources will be tested for shadow.

−dc *frac* Set the direct certainty to *frac.* A value of one guarantees that the absolute accuracy of the direct calculation will be equal to or better than that given in the *−dt* specification. A value of zero only insures that all shadow lines resulting in a contrast change greater than the *−dt* specification will be calculated.

-dr *N* Set the number of relays for secondary sources to *N.* A value of 0 means that secondary sources will be ignored. A value of 1 means that sources will be made into first generation secondary sources; a value of 2 means that first generation secondary sources will also be made into second generation secondary sources, and so on.

-dp *D* Set the secondary source presampling density to D. This is the number of samples per steradian that will be used to determine ahead of time whether or not it is worth following shadow rays through all the reflections and/or transmissions associated with a secondary source path. A value of 0 means that the full secondary source path will always be tested for shadows if it is tested at all.

−dv Boolean switch for light source visibility. With this switch off, sources will be black when viewed directly although they will still participate in the direct calculation. This option is mostly for the program *mkillum(1)* to avoid inappropriate counting of light sources, but it may also be desirable in conjunction with the −*i* option.

-sj *frac* Set the specular sampling jitter to *frac*. This is the degree to which the highlights are sampled for rough specular materials. A value of one means that all highlights will be fully sampled using distributed ray tracing. A value of zero means that no jittering will take place, and all reflections will appear sharp even when they should be diffuse.

-st *frac* Set the specular sampling threshold to *frac.* This is the minimum fraction of reflection or transmission, under which no specular sampling is performed. A value of zero means that highlights will always be sampled by tracing reflected or transmitted rays. A value of one means that specular sampling is never used. Highlights from light sources will always be correct, but reflections from other surfaces will be approximated using an ambient value. A sampling threshold between zero and one offers a compromise between image accuracy and rendering time.

-bv Boolean switch for back face visibility. With this switch off, back faces of opaque objects will be invisible to all rays. This is dangerous unless the model was constructed such that all surface normals on opaque objects face outward. Although turning off back face visibility does not save much computation time under most circumstances, it may be useful as a tool for scene debugging, or for seeing through one-sided walls from the outside. This option has no effect on transparent or translucent materials.

-av *red grn blu*
Set the ambient value to a radiance of *red grn blu* . This is the final value used in place of an indirect light calculation. If the number of ambient bounces is one or greater and the ambient value weight is non-zero (see -*aw* and -*ab* below), this value may be modified by the computed indirect values to improve overall accuracy.

-aw *N* Set the relative weight of the ambient value given with the -*av* option to *N*. As new indirect irradiances are computed, they will modify the default ambient value in a moving average, with the specified weight assigned to the initial value given on the command and all other weights set to 1. If a value of 0 is given with this option, then the initial ambient value is never modified. This is the safest value for scenes with large differences in indirect contributions, such as when both indoor and outdoor (daylight) areas are visible.

-ab *N* Set the number of ambient bounces to *N*. This is the maximum number of diffuse bounces computed by the indirect calculation. A value of zero implies no indirect calculation.

-ar *res* Set the ambient resolution to *res.* This number will determine the maximum density of ambient values used in interpolation. Error will start to increase on surfaces spaced closer than the scene size divided by the ambient resolution. The maximum ambient value density is the scene size times the ambient accuracy (see the −*aa* option below) divided by the ambient resolution. The scene size can be determined using *getinfo(1)* with the −*d* option on the input octree.

-aa *acc* Set the ambient accuracy to *acc.* This value will approximately equal the error from indirect illuminance interpolation. A value of zero implies no interpolation.

-ad *N* Set the number of ambient divisions to *N.* The error in the Monte Carlo calculation of indirect illuminance will be inversely proportional to the square root of this number. A value of zero implies no indirect calculation.

-as *N* Set the number of ambient super-samples to *N*. Super-samples are applied only to the ambient divisions which show a significant change.

-af *fname* Set the ambient file to *fname*. This is where indirect illuminance will be stored and retrieved. Normally, indirect illuminance values are kept in memory and lost when the program finishes or dies. By using a file, different invocations can share illuminance values, saving time in the computation. The ambient file is in a machine-independent binary format which can be examined with *lookamb(1)*.

The ambient file may also be used as a means of communication and data sharing between simultaneously executing processes. The same file may be used by multiple processes, possibly running on different machines and accessing the file via the network (ie. *nfs(4)*). The network lock manager *lockd(8)* is used to insure that this information is used consistently.

If any calculation parameters are changed or the scene is modified, the old ambient file should be removed so that the calculation can start over from scratch. For convenience, the original ambient parameters are listed in the header of the ambient file. *Getinfo(1)* may be used to print out this information.

-ae *mat* Append *mat* to the ambient exclude list, so that it will not be considered during the indirect calculation. This is a hack for speeding the indirect computation by ignoring certain objects. Any object having *mat* as its modifier will get the default ambient level rather than a calculated value. Any number of excluded materials may be given, but each must appear in a separate option.

-ai *mat* Add *mat* to the ambient include list, so that it will be considered during the indirect calculation. The program can use either an include list or an exclude list, but not both.

-aE *file* Same as *−ae*, except read materials to be excluded from *file*. The RAYPATH environment variable determines which directories are searched for this file. The material names are separated by white space in the file.

-aI *file* Same as *−ai*, except read materials to be included from *file*.

-me *rext gext bext*
Set the global medium extinction coefficient to the indicated color, in units of 1/distance (distance in world coordinates). Light will be scattered or absorbed over distance according to this value. The ratio of scattering to total scattering plus absorption is set by the albedo parameter, described below.

-ma *ralb galb balb*
Set the global medium albedo to the given value between 0 0 0 and 1 1 1. A zero value means that all light not transmitted by the medium is absorbed. A unitary value means that all light not transmitted by the medium is scattered in some new direction. The isotropy of scattering is determined by the Heyney-Greenstein parameter, described below.

−mg *gecc* Set the medium Heyney-Greenstein eccentricity parameter to *gecc*. This parameter determines how strongly scattering favors the forward direction. A value of 0 indicates perfectly isotropic scattering. As this parameter approaches 1, scattering tends to prefer the forward direction.

−ms *sampdist*
Set the medium sampling distance to *sampdist*, in world coordinate units. During source scattering, this will be the average distance between adjacent samples. A value of 0 means that only one sample will be taken per light source within a given scattering volume.

-lr *N* Limit reflections to a maximum of *N*.

-lw *frac* Limit the weight of each ray to a minimum of *frac*. During ray-tracing, a record is kept of the final contribution a ray would have to the image. If it is less then the specified minimum, the ray is not traced.

-ld Boolean switch to limit ray distance. If this option is set, then rays will only be traced as far as the magnitude of each direction vector. Otherwise, vector magnitude is ignored and rays are

traced to infinity.

-e *efile* Send error messages and progress reports to *efile* instead of the standard error.

−w Boolean switch to suppress warning messages.

−P *pfile* Execute in a persistent mode, using *pfile* as the control file. Persistent execution means that after reaching end-of-file on its input, *rtrace* will fork a child process that will wait for another *rtrace* command with the same *−P* option to attach to it. (Note that since the rest of the command line options will be those of the original invocation, it is not necessary to give any arguments besides *−P* for subsequent calls.) Killing the process is achieved with the *kill(1)* command. (The process ID in the first line of *pfile* may be used to identify the waiting *rtrace* process.) This option may be used with the *−fr* option of *pinterp(1)* to avoid the cost of starting up *rtrace* many times.

−PP *pfile* Execute in continuous-forking persistent mode, using *pfile* as the control file. The difference between this option and the *−P* option described above is the creation of multiple duplicate processes to handle any number of attaches. This provides a simple and reliable mechanism of memory sharing on most multiprocessing platforms, since the *fork(2)* system call will share memory on a copy-on-write basis.

EXAMPLES

To compute radiance values for the rays listed in samples.inp:

```
rtrace -ov scene.oct < samples.inp > radiance.out
```

To compute illuminance values at locations selected with the 't' command of *ximage(1):*

```
ximage scene.pic | rtrace -h -x 1 -i scene.oct | rcalc -e '$1=47.4*$1+120*$2+11.6*$3'
```

To record the object identifier corresponding to each pixel in an image:

```
vwrays -fd scene.pic | rtrace -fda `vwrays -d scene.pic` -os scene.oct
```

To compute an image with an unusual view mapping:

```
cnt 640 480 | rcalc -e 'xr:640;yr:480' -f unusual_view.cal | rtrace -x 640 -y 480 -fac scene.oct > unusual.pic
```

ENVIRONMENT

RAYPATH the directories to check for auxiliary files.

FILES

/usr/tmp/rtXXXXXX common header information for picture sequence

DIAGNOSTICS

If the program terminates from an input related error, the exit status will be 1. A system related error results in an exit status of 2. If the program receives a signal that is caught, it will exit with a status of 3. In each case, an error message will be printed to the standard error, or to the file designated by the *−e* option.

AUTHOR

Greg Ward

SEE ALSO

getinfo(1), lookamb(1), oconv(1), pfilt(1), pinterp(1), pvalue(1), rpict(1), rview(1), vwrays(1), ximage(1)