

**NAME**

rsensor - compute sensor signal from a RADIANCE scene

**SYNOPSIS**

```
rsensor [ -n nprocs ][ -h ][ render options ] [ $EVAR ] [ @file ] { [ -rd nrays ][ -dn nsrc ][ sensor_view
] sensor_file .. } octree
rsensor [ -h ] { [ -rd nrays ][ sensor_view ] sensor_file .. } .
rsensor [ options ] --defaults
```

**DESCRIPTION**

*Rsensor* traces rays outward from one or more specified illumination sensors into the RADIANCE scene given by *octree*, sending the computed sensor value to the standard output. (The octree may be given as the output of a command enclosed in quotes and preceded by a '!'.) In the second form, a single period ('.') is given in place of an octree, and the origin and directions of the specified number of rays will be printed on the standard output. If these rays are later traced and added together, the results will sum to a signal proportional to the given sensor distribution. In the third 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.

The sensor files themselves will be searched for in the path locations specified by the *RAYPATH* environment variable, similar to other types of Radiance auxiliary files. If the sensor file path begins with '/', '.' or '~', no search will take place. Before each sensor file, a separate view may be specified. In this case, the view origin and direction will correspond to the position and orientation of the sensor, and the view up vector will determine the zero azimuthal direction of the sensor. The fore clipping distance may be used as well, but other view options will be ignored. (See *rpict(1)* for details on how to specify a view.) The actual data contained in the sensor file corresponds to the *SPOT* tab-separated matrix specification, where the column header has "degrees" in the leftmost column, followed by evenly-spaced azimuthal angles. Azimuth angles must start at 0 and proceed to at least 270 degrees. No symmetry is applied to missing values. Each row begins with the polar angle, and is followed by the relative sensitivity values for each direction. Polar angles must start from 0 and continue to the edge of the sensor's view. Polar angles above the last row listed are assumed to have zero sensitivity. A low-resolution example of a sensor file is given below:

degrees	0	90	180	270
0	.02	.04	.02	.04
45	.01	.02	.01	.02
90	.001	.002	.001	.002

As well as different views, the number of samples may be changed between sensors, where the *-rd* option controls the number of ray samples sent at random, and the *-dn* option controls the number of rays sent to each light source per sensor.

The *-h* option toggles header output, which defaults to "on." The *-n* option may be used to specify multiple calculation processes on systems with more than one CPU. For additional options, consult the *rtrace(1)* man page. The final *octree* argument must be given, as the octree cannot be read from the standard input.

**EXAMPLES**

To compute values for the same sensor with two different positions:

```
rsensor -ab 2 -vf posA.vf mysens.dat -vf posB.vf mysens.dat scene.oct
```

To generate a set of rays corresponding to a given sensor and compute the resulting signal with *rtrace*:

```
rsensor -h -vf posC.vf mysens.dat . | rtrace -h scene.oct | total -m
```

RSensor(1)

RSensor(1)

## **ENVIRONMENT**

RAYPATH                      the directories to check for auxiliary files.

## **AUTHOR**

Greg Ward for Architectural Energy Corporation

## **SEE ALSO**

oconv(1), rpict(1), rcontrib(1), rtrace(1)