

NAME

rfluxmtx - compute flux transfer matrix(es) for RADIANCE scene

SYNOPSIS

rfluxmtx [**-v**] [**rcontrib options**] { **sender.rad** | **-** } **receivers.rad** [**-i system.oct**] [**system.rad ..**]

DESCRIPTION

Rfluxmtx samples rays uniformly over the surface given in *sender.rad* and records rays arriving at surfaces in the file *receivers.rad*, producing a flux transfer matrix per receiver. A system octree to which the receivers will be appended may be given with a **-i** option following the receiver file. Additional system surfaces may be given in one or more *system.rad* files, which are compiled before the receiver file into an octree sent to the *rcontrib(1)* program to do the actual work. If a single hyphen ('-') is given in place of the sender file, then *rfluxmtx* passes ray samples from its standard input directly to *rcontrib* without interpretation. By default, all resulting matrix data are interleaved and sent to the standard output in ASCII format, but this behavior is typically overridden using inline options as described below.

The **-v** option turns on verbose reporting for the number of samples and the executed *rcontrib* command. All other supported options are passed on to *rcontrib(1)*. However, the **-f**, **-e**, **-p**, **-b**, **-bn**, **-m**, and **-M** options are controlled by *rfluxmtx* and may not be set by the user. Also, the **-x**, **-y**, and **-ld** options are ignored unless *rfluxmtx* is invoked in the pass-through mode, in which case they may be needed to generate RADIANCE views from *vwrays(1)*. The sample count, unless set by the **-c** option, defaults to 10000 when a sender file is given, or to 1 for pass-through mode.

VARIABLES

The sender and receiver scene files given to *rfluxmtx* contain controlling parameters in special comments of the form:

```
#@rfluxmtx variable=value ..
```

At minimum, both sender and receiver must specify one of the hemisphere sampling types, and there must be at least one surface in each file.

h=u Set hemisphere sampling to "uniform," meaning a single bin of (cosine-distributed) samples. In the case of distant "source" primitives, this is the only sampling method that supports arbitrary receiver sizes. The other methods below require a full hemispherical source.

h=kf Divide the hemisphere using the LBNL/Klems "full" sampling basis. (Use "h=kf" for left-handed coordinates.)

h=kh Divide the hemisphere using the LBNL/Klems "half" sampling basis. (Use "h=kh" for left-handed coordinates.)

h=kq Divide the hemisphere using the LBNL/Klems "quarter" sampling basis. (Use "h=kq" for left-handed coordinates.)

h=rN Divide the hemisphere using Reinhart's substructuring of the Tregenza sky pattern with *N* divisions in each dimension. If it is not given, *N* defaults to 1, which is just the Tregenza sky. (Use "h=rN" for left-handed coordinates.)

h=cie Divide the hemisphere into CIE sky scanner directions, which is similar to Tregenza but with different starting azimuths and reversing row direction at each new altitude. (Use "h=cie" for left-handed coordinates.)

h=scN Subdivide the hemisphere using the Shirley-Chiu square-to-disk mapping with an *N*×*N* grid over the square. (Use "h=scN" for left-handed coordinates.)

u=[-]{X|Y|Z|ux,uy,uz}

Orient the "up" direction for the hemisphere using the indicated axis or direction vector.

o=output_spec

Send the matrix data for this receiver to the indicated file or command. Single or double quotes may be used to contain strings with spaces, and commands must begin with an

exclamation mark (!). The file format will be determined by the command-line *-fio* option and will include an information header unless the *-h* option was used to turn headers off. (The output file specification is ignored for senders.)

In normal execution, only a single sender surface is sampled, but it may be comprised of any number of subsurfaces, as in a triangle mesh or similar. The surface normal will be computed as the average of all the constituent subsurfaces. The subsurfaces themselves must be planar, thus only *polygon* and *ring* surface primitives are supported. Other primitives will be silently ignored and will have no effect on the calculation.

In the receiver file, the *source* primitive is supported as well, and multiple receivers (and multiple output matrices) may be identified by different modifier names. (Make sure that surfaces using the same modifier are grouped together, and that the modifiers are unique and not used elsewhere in the scene description.) Though it may be counter-intuitive, receivers are often light sources, since samples end up there in a backwards ray-tracing system such as RADIANCE. When using local geometry, the overall aperture shape should be close to flat. Large displacements may give rise to errors due to a convex receiver's larger profile at low angles of incidence.

Rays always emanate from the back side of the sender surface and arrive at the front side of receiver surfaces. In this way, a receiver surface may be reused as a sender in a subsequent *rfluxmtx* calculation and the resulting matrices will concatenate properly. (Note that it is important to keep receiver surfaces together, otherwise a "duplicate modifier" error will result.)

EXAMPLES

To generate a flux transfer matrix connecting input and output apertures on a light pipe:

```
rfluxmtx int_aperture.rad ext_aperture.rad lpipe.rad > lpipe.mtx
```

AUTHOR

Greg Ward

SEE ALSO

genBSDF(1), getinfo(1), pvsum(1), rcalc(1), rcollate(1), rcomb(1), rcontrib(1), rcrop(1), rmtxop(1), vwrays(1), wrapBSDF(1)