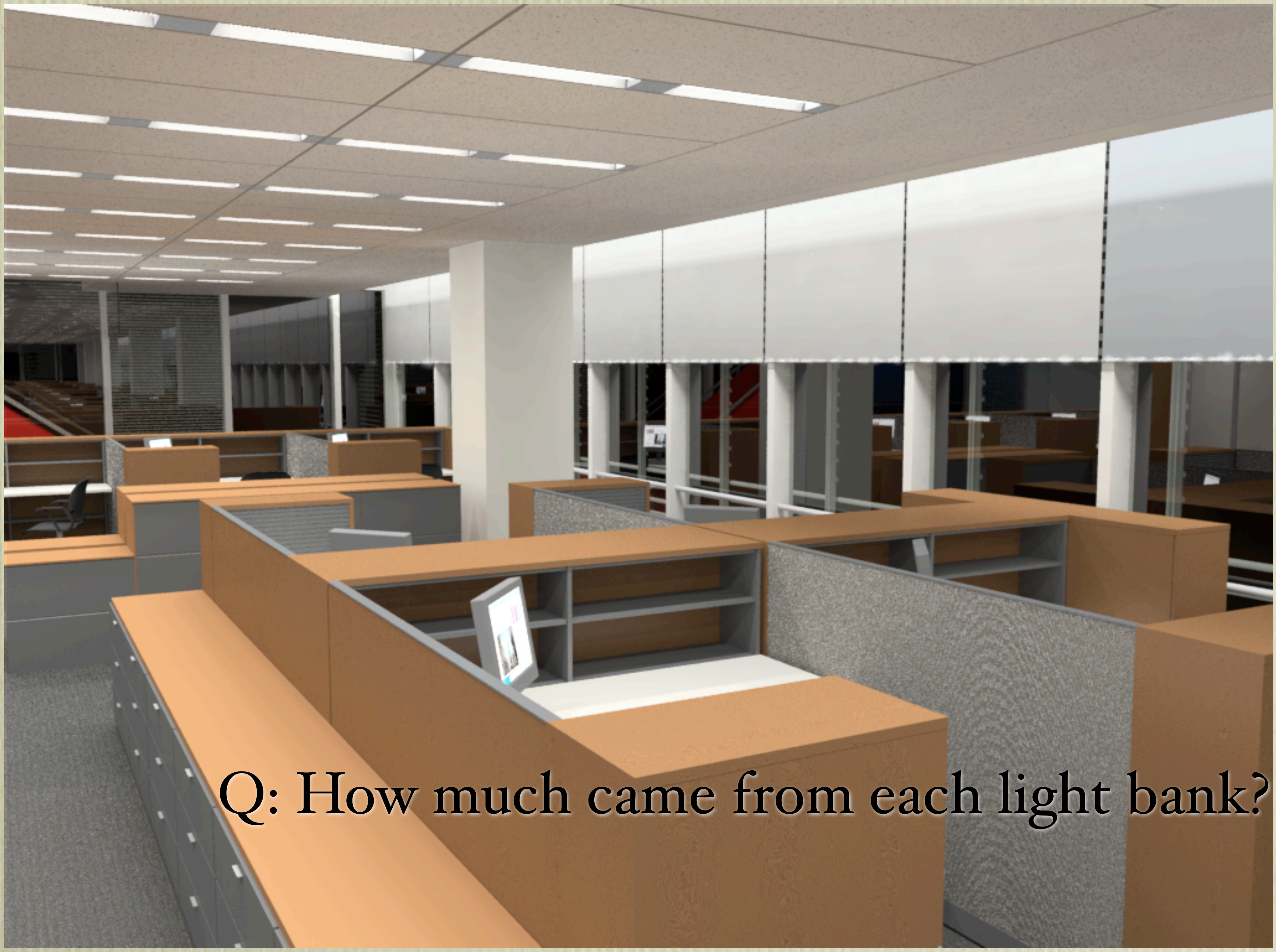
The *Radiance* **rtcontrib** Program

Greg Ward
Anyhere Software

Basic Idea

- Normally, details about what sources contributed to a pixel are thrown away
- Saving this information allows different source contributions to be substituted
- More generally, the flux transfer between a point and any other point may be quantified

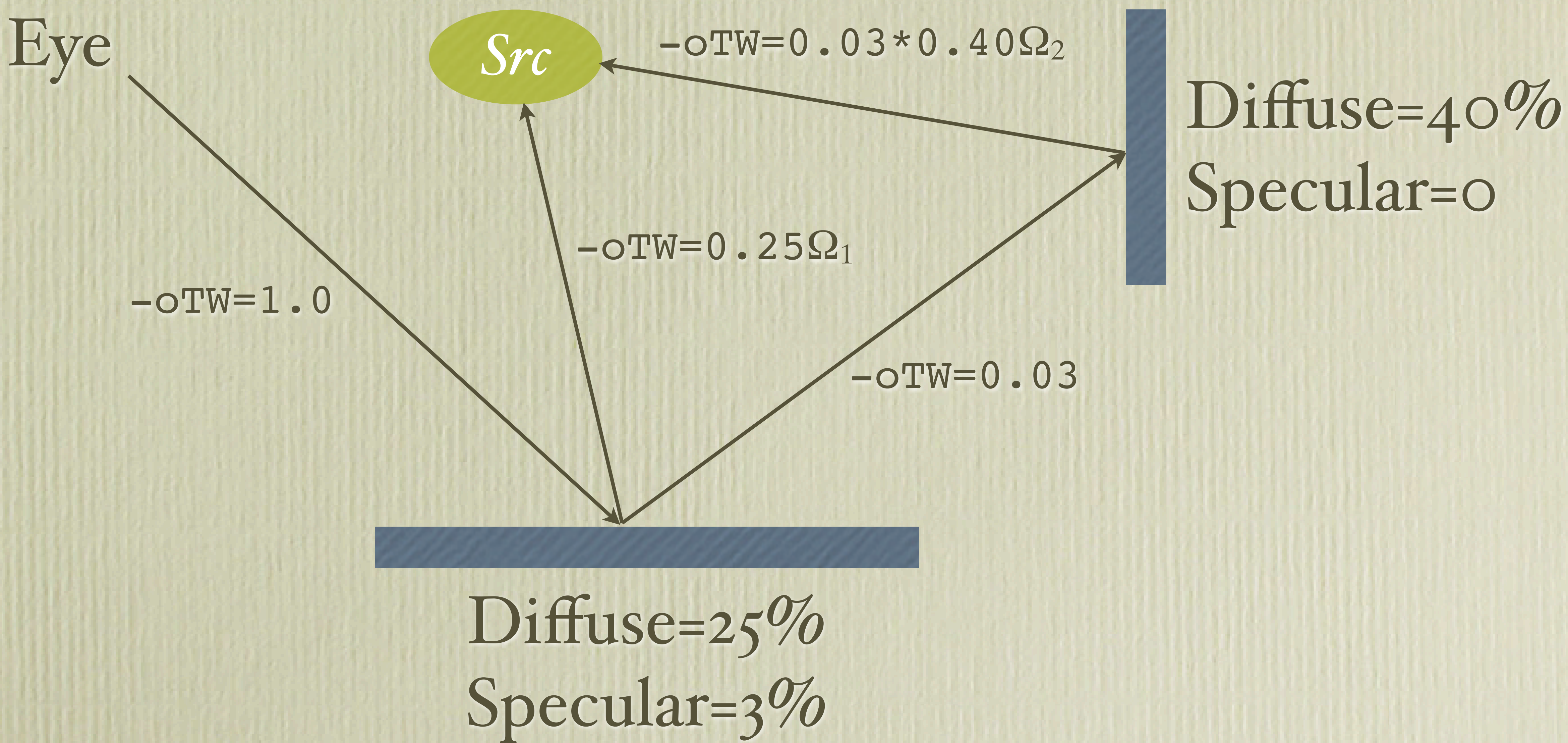
Quantifying Contributions



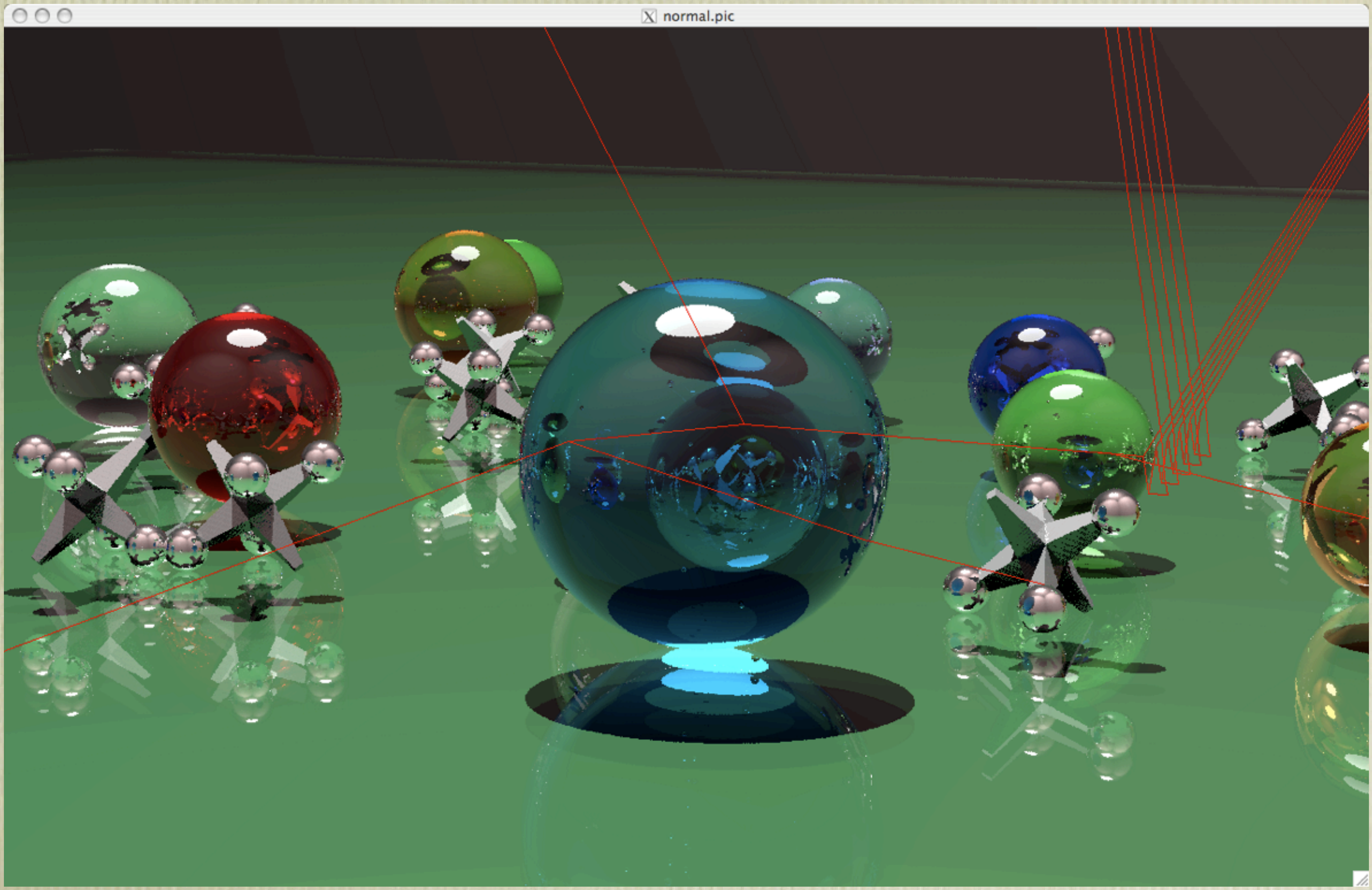
Background

- Core *Radiance* rendering routines recursively evaluate radiance, hence the name
- Potentially useful information about where light originates is lost during this process
- Prior to version 3.7, there were two solutions:
    - Repeat rendering for each source (costly)
    - Switch to *Daysim* (daylight coefficients only)

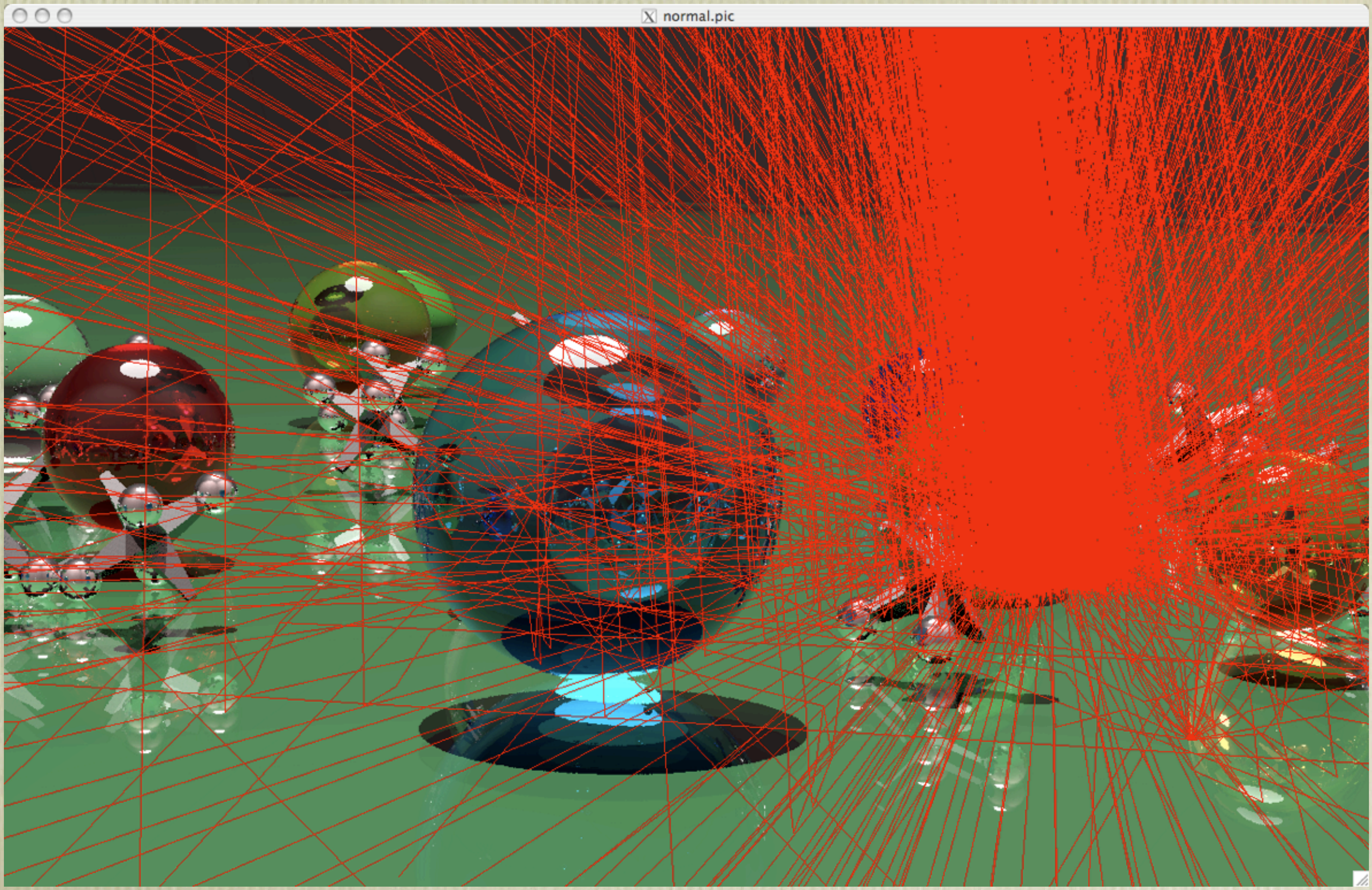
A Simple Example



Problem: Daughter Rays



Diffuse Interreflections



Central Concept
Understand This First!

- Rays of interest are fed as input, as for **rtrace**
  - view rays for image, illuminance points, etc.
- Output is determined by "catch surfaces"
  - defined via *modifier names* and *bin expressions*
  - this part is new, and can be a bit tricky

rtcontrib: Gathers Rays

- Gather contributions & coefficients and sum them up logically
- Different applications require different sums:
    - Daylight coefficients sum at sky patches
    - Luminaire model may sum at lamp surface
- **rtcontrib** provides general mechanism, while **rtrace** handles actual ray-tracing

General Operation

- Looks like **rtrace** command, similar options
- Options tell **rtcontrib** where to collect values
  - required modifier name(s)
  - optional bin number based on ray direction and intersection point
- Output sent to one or more files or commands
  - specified by modifier name and bin number

Lighting Example

```
vwrays -ff -x 1024 -y 1024 -vf model.vp \
| rtcontrib -V+ -o part_%s.pic -m fluor1 -m fluor2 \
-ffc `vwrays -d -x 1024 -y 1024 -vf model.vp` -u+ model.oct
```

rtcontrib Options

General options:

- `-n` *N* start *N* **rtrace** processes
- `-r` recover previously aborted calculation
- `-e` *expr* compile definitions string
- `-f` *source* compile definitions file

Used by `-b`

Modifier options:

- `-o` *ospec* output specification — May contain '`!`' and '`%d`' or '`%s`'
- `-b` *binv* bin number — Integer expression, or '`0`' to disable

Modifier specification:

- `-m` *mod* modifier name
- `-M` *file* modifier list from file

Lighting Example Dissection

```
vwrays -ff -x 1024 -y 1024 -vf model.vp \
| rtcontrib -o part_%s.pic -m fluor1 -m fluor2 \
-ffc `vwrays -d -x 1024 -y 1024 -vf model.vp` model.oct
```

Lighting Example Dissection

vwrays provides primary ray origins and directions (in floating point) for pictures to be generated by rtcontrib

```
vwrays -ff -x 1024 -y 1024 -vf model.vp \
| rtcontrib -o part_%s.pic -m fluor1 -m fluor2 \
-ffc `vwrays -d -x 1024 -y 1024 -vf model.vp` model.oct
```

Second invocation reports actual resolution (-x 1024 -y 690)

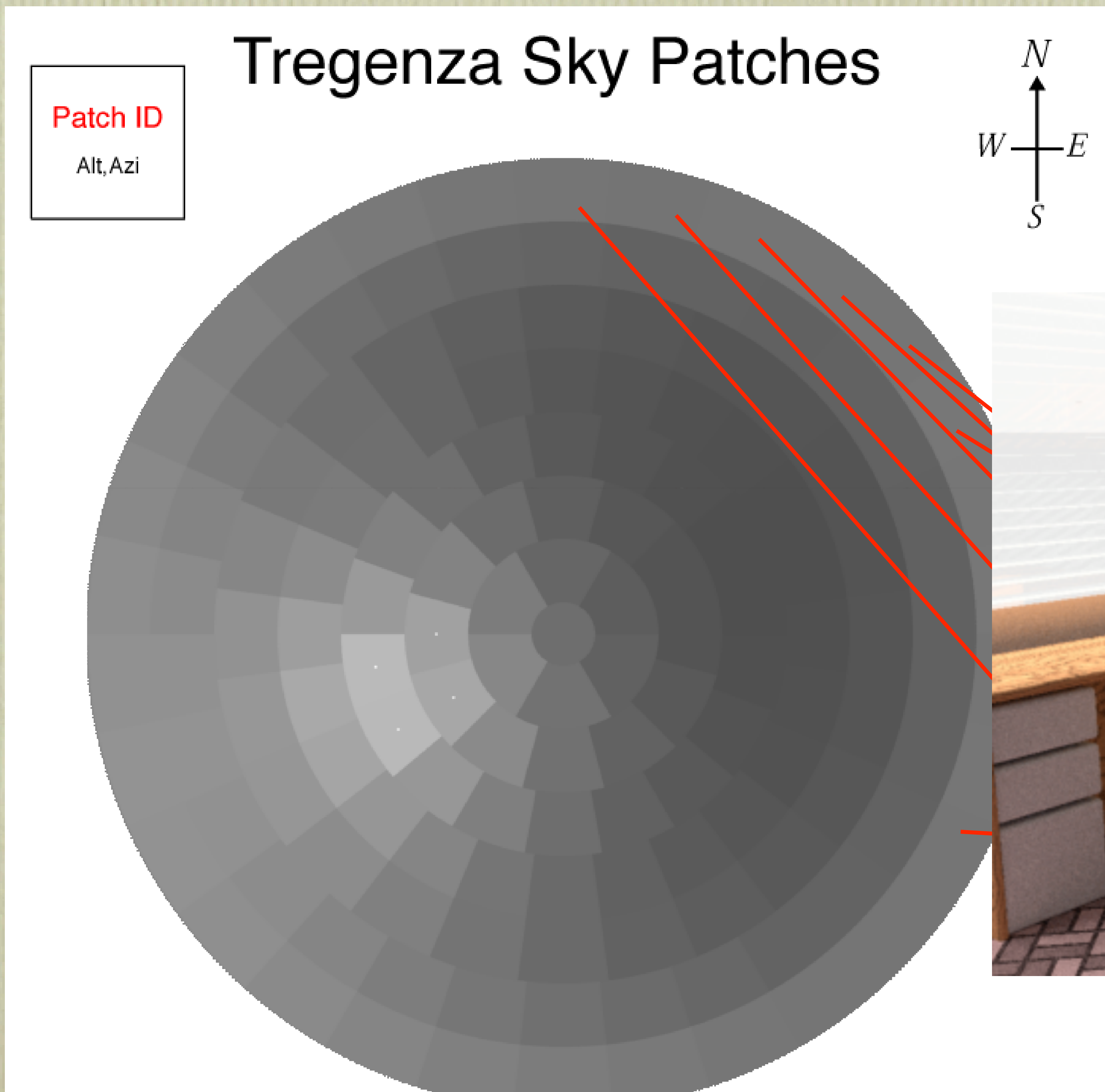
Lighting Example Dissection

Specifies output files and associated modifiers, creating `part_fluor1.pic` and `part_fluor2.pic`.

```
vwrays -ff -x 1024 -y 1024 -vf model.vp \
| rtcontrib -o part_%s.pic -m fluor1 -m fluor2 \
-ffc `vwrays -d -x 1024 -y 1024 -vf model.vp` model.oct
```

The `-ffc` option is an **rtrace** option telling **rtcontrib** to expect single-precision floats on input and produce RGBE colors on output.

Daylight Coefficients



From John Mardaljevic's thesis

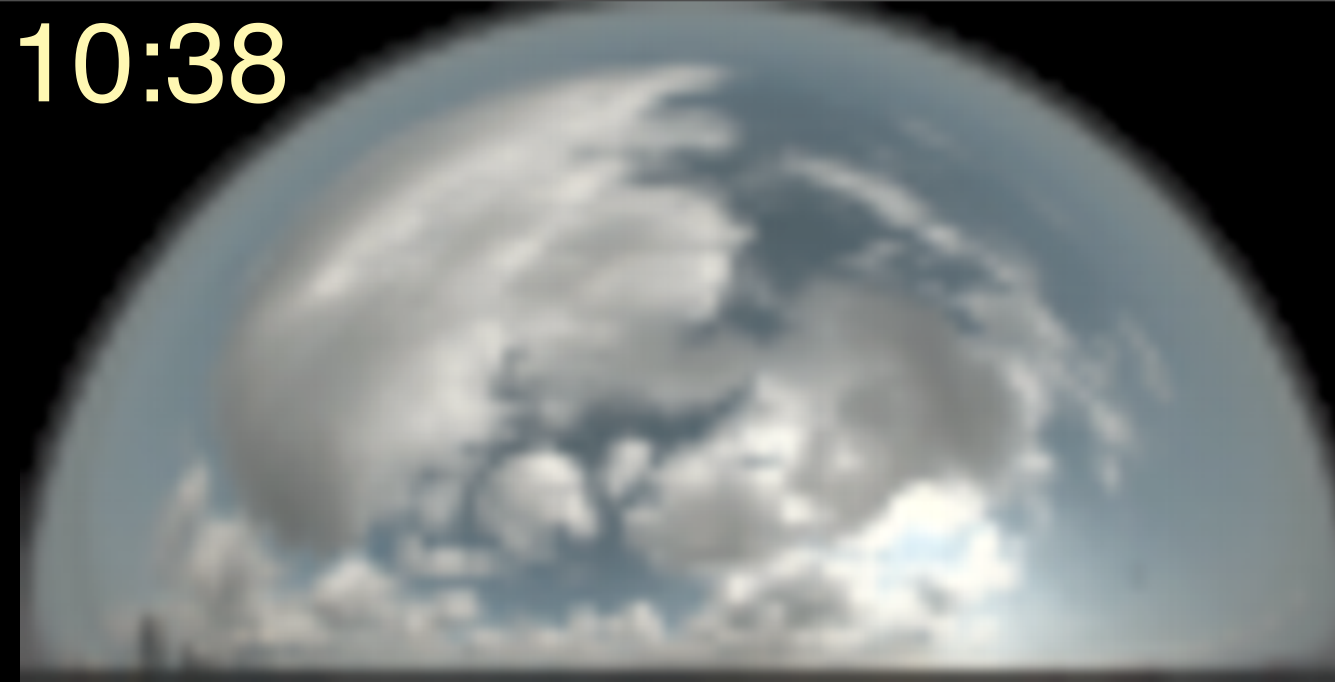
Example with Captured Sky

- Parthenon model created by Paul Debevec & Co. at USC's ICT from 100's of site laser scans
- Los Angeles sky captured by Jessi Stumpfel on the one cloudy day she could find
- **rtcontrib** run plus **pcomb** sums for 690 animation frames done in 6 hours on G5 Quad

Parthenon Rendering Method

- Run **rtcontrib** to compute each Tregenza sky patch DC contribution as a partial image
  - Takes the time of a single MC rendering
- Use low-resolution version of captured sky to compute Tregenza patch radiances
- Use radiance as coefficients in **pcomb** command to combine partial images

10:38



Tregenza
Patch 001

Actual Commands

```
rtcontrib -x 2048 -y 1361 -ffc -ab 2 -ad 1024 -f
tregenza.cal -b tbin -o p%d.hdr -m white -n 4
parth.oct < rays.flt
```

Then, for each frame time:

```
mksky.csh $time | oconv - > capsky.oct

rtrace -h -w -dv- capsky.oct < tregsamp.dat |
total -64 -m > pval.dat

pcomb -h `rcalc -o '-s ${$1} p${recno-1}.hdr'
pval.dat` > f$time.hdr
```

Example **pcomb** Command

```
pcomb -h -s 26.679 p0.hdr -s 17.0417 p1.hdr -s 16.7292 p2.hdr -s 17.2695 p3.hdr -s 17.0158 p4.hdr -s
16.9859 p5.hdr -s 17.3179 p6.hdr -s 17.3057 p7.hdr -s 16.9012 p8.hdr -s 17.1711 p9.hdr -s 17.005
p10.hdr -s 16.4502 p11.hdr -s 16.9732 p12.hdr -s 17.3672 p13.hdr -s 17.5727 p14.hdr -s 18.3379
p15.hdr -s 19.6837 p16.hdr -s 20.945 p17.hdr -s 22.1034 p18.hdr -s 23.4713 p19.hdr -s 24.4605 p20.hdr
-s 24.9192 p21.hdr -s 24.7394 p22.hdr -s 23.8799 p23.hdr -s 22.8255 p24.hdr -s 21.4208 p25.hdr -s
20.2332 p26.hdr -s 18.811 p27.hdr -s 17.9236 p28.hdr -s 17.5551 p29.hdr -s 16.943 p30.hdr -s 11.3134
p31.hdr -s 10.9923 p32.hdr -s 11.0595 p33.hdr -s 10.8431 p34.hdr -s 10.6877 p35.hdr -s 10.8235
p36.hdr -s 10.9221 p37.hdr -s 10.6787 p38.hdr -s 10.8518 p39.hdr -s 11.0184 p40.hdr -s 11.1377
p41.hdr -s 11.4171 p42.hdr -s 12.0989 p43.hdr -s 12.8039 p44.hdr -s 13.9147 p45.hdr -s 15.4106
p46.hdr -s 17.1459 p47.hdr -s 19.1204 p48.hdr -s 20.8152 p49.hdr -s 22.3375 p50.hdr -s 23.096 p51.hdr
-s 22.8274 p52.hdr -s 21.7096 p53.hdr -s 20.0354 p54.hdr -s 18.0616 p55.hdr -s 16.4738 p56.hdr -s
14.6816 p57.hdr -s 13.3963 p58.hdr -s 12.473 p59.hdr -s 11.6952 p60.hdr -s 9.06285 p61.hdr -s 8.58478
p62.hdr -s 8.28409 p63.hdr -s 8.16134 p64.hdr -s 8.02032 p65.hdr -s 8.10924 p66.hdr -s 8.21599
p67.hdr -s 8.4734 p68.hdr -s 8.77964 p69.hdr -s 9.56698 p70.hdr -s 10.5602 p71.hdr -s 12.0353 p72.hdr
-s 14.007 p73.hdr -s 16.7768 p74.hdr -s 19.7968 p75.hdr -s 22.7948 p76.hdr -s 24.3699 p77.hdr -s
23.7876 p78.hdr -s 21.216 p79.hdr -s 18.0321 p80.hdr -s 15.0926 p81.hdr -s 12.7871 p82.hdr -s 11.1515
p83.hdr -s 9.92927 p84.hdr -s 8.60844 p85.hdr -s 7.95365 p86.hdr -s 7.53902 p87.hdr -s 7.30563
p88.hdr -s 7.22913 p89.hdr -s 7.25486 p90.hdr -s 7.41717 p91.hdr -s 7.75376 p92.hdr -s 8.26077
p93.hdr -s 9.08944 p94.hdr -s 10.3667 p95.hdr -s 12.1716 p96.hdr -s 14.6882 p97.hdr -s 18.2808
p98.hdr -s 22.5925 p99.hdr -s 27.2209 p100.hdr -s 30.3359 p101.hdr -s 28.7542 p102.hdr -s 24.6012
p103.hdr -s 20.1185 p104.hdr -s 16.1143 p105.hdr -s 13.1023 p106.hdr -s 10.951 p107.hdr -s 9.62028
p108.hdr -s 9.18923 p109.hdr -s 8.22889 p110.hdr -s 7.61537 p111.hdr -s 7.45216 p112.hdr -s 7.52448
p113.hdr -s 7.91023 p114.hdr -s 8.79739 p115.hdr -s 10.3087 p116.hdr -s 12.6263 p117.hdr -s 16.4565
p118.hdr -s 22.6238 p119.hdr -s 31.9131 p120.hdr -s 41.3573 p121.hdr -s 36.2196 p122.hdr -s 25.4606
p123.hdr -s 18.206 p124.hdr -s 13.4968 p125.hdr -s 10.87 p126.hdr -s 10.6187 p127.hdr -s 9.24827
p128.hdr -s 8.69842 p129.hdr -s 9.03493 p130.hdr -s 10.3831 p131.hdr -s 13.2038 p132.hdr -s 18.6265
p133.hdr -s 29.4887 p134.hdr -s 43.2829 p135.hdr -s 32.612 p136.hdr -s 20.4952 p137.hdr -s 13.678
p138.hdr -s 13.1981 p139.hdr -s 11.1899 p140.hdr -s 12.8601 p141.hdr -s 19.2883 p142.hdr -s 27.1508
p143.hdr -s 19.9292 p144.hdr -s 16.5438 p145.hdr > f10-38.hdr
```

Using **dctimestep**

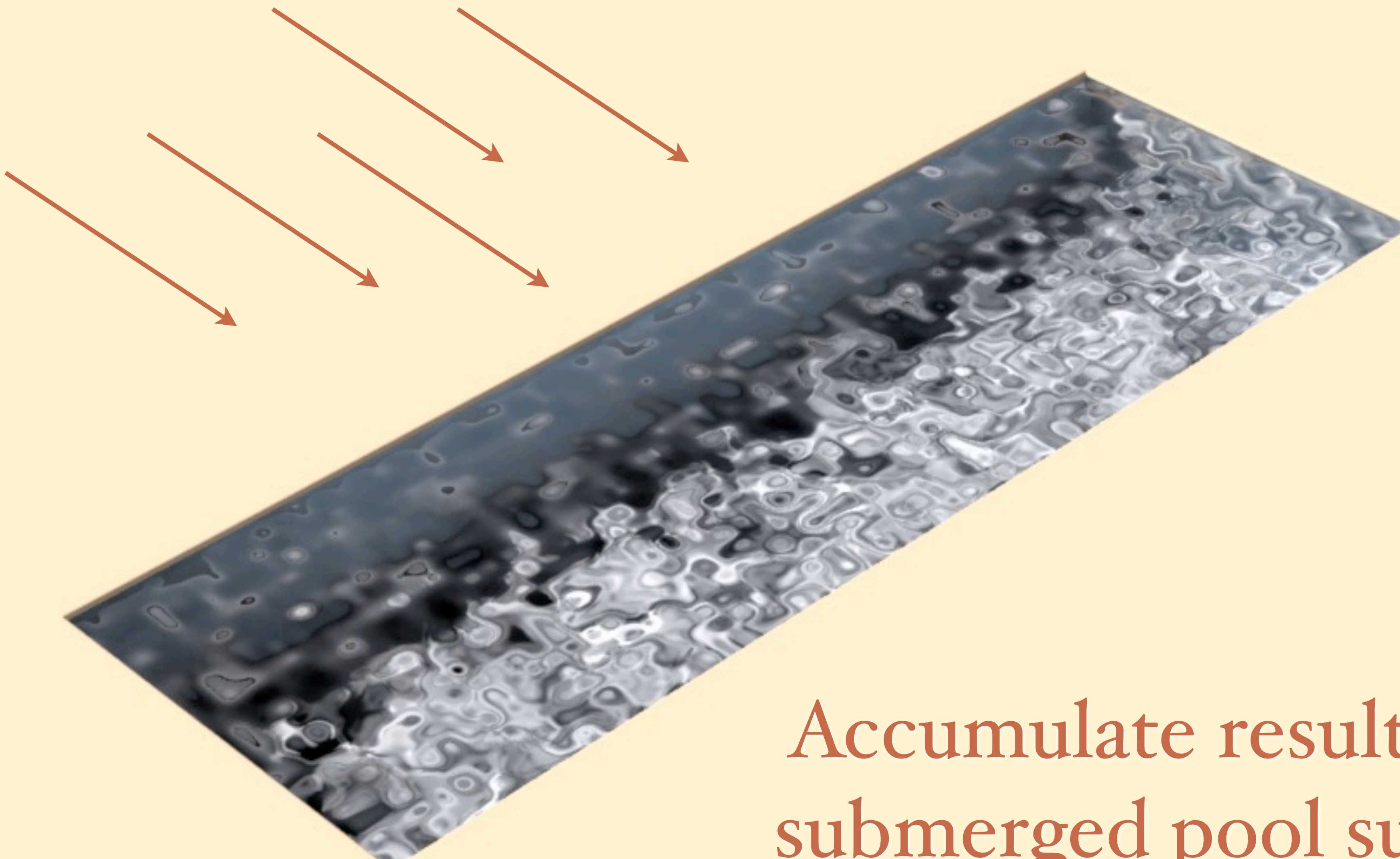
```
dctimestep p%d.hdr pval.dat > f10-38.hdr
```

Much faster!

This didn't exist when the animation was done.

Computing Caustics (1)

Send rays from “sun”



Accumulate results on 5 submerged pool surfaces

Computing Caustics (2)

```
vwrays -pj .6 -vf pool_par.vf -x 500 -y 500 -ff \
                | rtcontrib -w -ffc -c 0 \
                        -f pool_coords.cal -o %s_caustics.pic \
                        -b floor_bin -bn 7500 -x 50 -y 150 -m floor \
                        -b s_wall_bin -bn 1500 -x 50 -y 30 -m s_wall \
                        -b n_wall_bin -bn 1500 -x 50 -y 30 -m n_wall \
                        -b e_wall_bin -bn 4500 -x 150 -y 30 -m e_wall \
                        -b w_wall_bin -bn 4500 -x 150 -y 30 -m w_wall \
                        catchscene.oct
```

Accumulate results

```
        { south wall bins }
  SWallXres : 50;
  SWallZres : 30;
  SWallWidth : 5 {meters};
  SWallOrigX : -2.5;
  SWallHeight : 3 {meters};
  SWallOrigZ : -3.08;
  s_wall_bin = floor(SWallXres*(Px - SWallOrigX)/SWallWidth) +
     floor(SWallZres*(1 - (Pz - SWallOrigZ)/SWallHeight))*SWallXres;
```

part of pool_coords.cal

Computing Caustics (3)

```
void colorpict s_wall_pat
7 red green blue s_wall_caustics.pic . (Px--2.5)/3 (Pz--3.08)/3
0
0

s_wall_pat glow s_wall
0
0
4 5 5 5 0
```

“Forward Ray-Tracing”

- The caustics example shows one way **rtcontrib** can act like a photon-mapper
- It would be nice to extend this to more general situations
- **genBSDF** does this for surfaces, what about light pipes, etc?
- Likely future work

The Care and Feeding of **rtcontrib**

- If you use **`-b`**, must use **`-bn`** as well
- Careful where your place **`-m`** and **`-M`** options
- Tracking absolute flux values using **`-c 0`** requires careful distribution of incident rays
- Remember that contributions are triplet values
- Learn the difference between **`-V-`** and **`-V+`**
- Options must be appropriate to pure MC

What Is Pure Monte Carlo?

- No indirect cache (**`-aa 0`**)
- Uncorrelated sampling (**`-u+`**)
- Russian roulette termination (**`-lr`** ≤0)
- Other recommended settings:
    - **`-as 0 -dt 0 -dj 0.9 -sj 1`**
- **`-lw`** setting very important: `1/#paths`

No Ambient Cache?

- **rtcontrib** needs to know at any point what contribution will ultimately be made
- Ambient values are stored and reused later in an untraceable way
- *Daysim* avoids this issue by storing daylight coefficients in each ambient value
- **rtcontrib** more general and memory-efficient

Multiprocessing in **rtcontrib**

- Supported with **-n** option
- This has been greatly sped up and improved in the 4.2 replacement command, **rcontrib**
- No calls to **rtrace** and little overhead
- Using **-c** option further improves scalability

Number of Open Files

- Since many components are often desired, the number of open files may be a constraint
- Raise max. open descriptors/process:
  - `ulimit -n 1024`, or
  - `limit descriptors 1024`
- This may still be inadequate in some cases....

Recovery Options

- If output is to binary file(s) via `-o` option, &...
- Exact command is re-issued with `-r` option,
- Then **rtcontrib** attempts to append data
- So make sure old process is dead, first!
- If `-c 0` option is in effect, sums are updated
  - very useful for progressive calculations