

**NAME**

`pmapdump` - generate RADIANCE scene description or point list representing photon positions and (optionally) flux

**SYNOPSIS**

```
pmapdump [-a] [-n num1] [-r radscale1] [-f | -c rcol1 gcol1 bcol1]
          pmap1
          [-a] [-n num2] [-r radscale2] [-f | -c rcol2 gcol2 bcol2]
          pmap2 ...
```

**DESCRIPTION**

`pmapdump` takes one or more photon map files generated with `mkpmap(1)` as input and, by default, sends a RADIANCE scene description of their photon distributions to the standard output. Photons are represented as spheres of material type `glow`. These can be visualised with e.g. `objview(1)`, `rpict(1)`, or `rvu(1)` to assess the location and local density of photons in relation to the scene geometry. No additional light sources are necessary, as the spheres representing the photons are self-luminous.

Alternatively, photons can also be output as an ASCII point list, where each line contains a photon's position and colour. This point list can be imported in a 3D point cloud processor/viewer to interactively explore the photon map.

An arbitrary number of photon maps can be specified on the command line and the respective photon type is determined automatically. Per default, the different photon types are visualised as colour coded spheres/points according to the following default schema:

*Blue:* global photons  
*Cyan:* precomputed global photons  
*Red:* caustic photons  
*Green:* volume photons  
*Magenta:* direct photons  
*Yellow:* contribution photons

These colours can be overridden for individual photon maps with the `-c` option (see below). Alternatively, photons can be individually coloured according to their actual RGB flux with the `-f` option (see below); while this makes it difficult to discern photon types, it can be used to quantitatively analyse colour bleeding effects, for example.

**OPTIONS**

Options are effective for the photon map file immediately following on the command line, and are reset to their defaults after completion of each dump. As such they must be set individually for each photon map.

**-a** Boolean switch to output photons as a point list in ASCII (text) format instead of a RADIANCE scene. Each output line consists of 6 tab-separated floating point values: the X, Y, Z coordinates of the photon's position, and the R, G, B colour channels of its flux. These values, notably the flux, are expressed in scientific notation if necessary to accommodate their high dynamic range.

As `pmapdump` groups its options per photon map, this option must be specified per photon map for consistent output. This prevents erroneously dumping RADIANCE scene descriptions along with point lists, which will fail to load in the 3D point cloud processor/viewer.

**-c rcol gcol bcol**

Specifies a custom sphere/point colour for the next photon map. The colour is specified as an RGB triplet, with each component in the range (0..1]. Without this option, the default colour for the corresponding photon type is used. This option is mutually exclusive with `-f`.

- f** Boolean switch to colour each sphere/point according to the corresponding photon's RGB flux instead of a constant colour. The flux is adjusted for the fraction of dumped photons to maintain the total flux contained in the dumped photon map. Note that no exposure is applied, and as such the resulting colours can span several orders of magnitude and may require tone mapping with *pcond(1)* for visualisation. This option is mutually exclusive with **-c**.
  
- n num** Specifies the number of spheres or points to dump for the next photon map. The dump is performed by random sampling with *num* as target count, hence the number actually output will be approximate. *Num* may be suffixed by a case-insensitive multiplier for convenience, where  $k = 10^3$  and  $m = 10^6$ , although the latter may lead to problems when processing the output geometry with *oconv(1)*. The default number is 10k.
  
- r radscale** Specifies a relative scale factor *radscale* for the sphere radius. The sphere radius is determined automatically from an estimated average distance between spheres so as to reduce clustering, assuming a uniform distribution. In cases where the distribution is substantially nonuniform (e.g. highly localised caustics) the radius can be manually corrected with this option. The default value is 1.0. This option is ignored for point list output in conjunction with **-a**.

## NOTES

The RADIANCE scene output may contain many overlapping spheres in areas with high photon density, particularly in caustics. This results in inefficient and slow octree generation with *oconv(1)*. Generally this can be improved by reducing *num* and/or *radscale*.

## EXAMPLES

Visualise the distribution of global and caustic photons superimposed on the scene geometry with 5000 pale red and 10000 pale blue spheres, respectively:

```
pmapdump -n 5k -c 1 0.4 0.4 global.pm -n 10k -c 0.4 0.4 1 caustic.pm | oconv - scene.rad > scene_pm.oct
```

Visualise the caustic photon distribution superimposed on the scene geometry with 10000 spheres coloured according to the photons' respective RGB flux:

```
pmapdump -n 10k -f caustic.pm | oconv - scene.rad > scene_pm.oct
```

But Capt. B wants 'em bigger:

```
pmapdump -r 4.0 bonzo.pm > bonzo_bigballz.rad
```

RADIANCE scene dumps may also be viewed on their own by simply piping the output of *pmapdump* directly into *objview(1)* (using the default number of spheres in this example):

```
pmapdump zombo.pm | objview
```

Instead of a RADIANCE scene, dump photons as a (really long) point list to an ASCII file for import into a 3D point cloud processor/viewer:

```
pmapdump -a -f -n 1m lotsa.pm > lotsa_pointz.txt
```

## AUTHOR

Roland Schregle (roland.schregle@hslu.ch, gmail.com)

## COPYRIGHT

(c) Fraunhofer Institute for Solar Energy Systems,  
(c) Lucerne University of Applied Sciences and Arts,  
(c) Tokyo University of Science.

## ACKNOWLEDGEMENTS

Development of the RADIANCE photon mapping extension was supported by:

*Fraunhofer Institute for Solar Energy Systems* funded by the German Research Foundation (*DFG LU204/10-2*, "Fassadenintegrierte Regelsysteme (FARESYS)"),

*Lucerne University of Applied Sciences and Arts* funded by the Swiss National Science Foundation (*SNSF 147053*, "Daylight redirecting components"),

*Tokyo University of Science* funded by the JSPS Grants-in-Aid for Scientific Research Programme (*KAKENHI JP19KK0115*, "Three-dimensional light flow").

Many thanks also to the many individuals who tested the code and provided valuable feedback. Special greetz to Don Gregorio, PAB and Capt. B!

## SEE ALSO

mkpmap(1), objview(1), oconv(1), rpict(1), rvu(1),  
*The RADIANCE Photon Map Manual*,  
*Development and Integration of the RADIANCE Photon Map Extension: Technical Report*,  
*The RADIANCE Out-of-Core Photon Map: Technical Report*,  
*Bonzo Daylighting Tool a.k.a. EvilDRC [TM]*