NAME

mkpmap - generate RADIANCE photon map

SYNOPSIS

mkpmap **−apg|−apc|−apv|−apd|−app|−apC** *file nphotons* [*bwidth*] ...
[options] *octree*

DESCRIPTION

Mkpmap takes a RADIANCE scene description as an octree and performs Monte Carlo forward path tracing from the light sources, depositing indirect ray hitpoints along with their energy (flux) as "photons". The resulting localised energy distribution represents a global illumination solution which is written to a file for subsequent evaluation by *rpict(1), rtrace(1)* and *rvu(1)* in a backward raytracing pass. The photon map(s) can be reused for multiple viewpoints and sensor locations as long as the geometry remains unchanged.

OPTIONS

Mkpmap can generate different types of photon maps depending on the materials present in the scene. In most cases, these can be specified independently or in combination on the command line. If multiple photon maps of the same type are specified, the last instance takes precedence.

−apg *file nphotons*

Generate a global photon map containing approximately *nphotons* photons, and output to *file*. This accounts for all indirect illumination, from both specular and diffuse scattering, on surfaces with a diffuse component. This is the most general type of photon map and replaces the ambient calculation in *rpict(1), rtrace(1)* and *rvu(1)*.

−apc *file nphotons*

Generate a separate caustic photon map containing approximately *nphotons* photons, and output to file *file*. This is a subset of the global photon map intended for direct visualisation at primary rays, This accounts for all indirect illumination on diffuse surfaces from specular scattering, which usually exhibits a large gradient and requires a higher resolution than the global photon map, typically containing the tenfold number of photons.

−apv *file nphotons*

Generate a volume photon map containing approximately *nphotons* photons, and output to file *file*. These account for indirect inscattering in participating media such as **mist** and complement the direct inscattering computed by *rpict(1), rtrace(1)* and *rvu(1)*. See also the **−me**, **−ma** and **−mg** options below.

−apd *file nphotons*

Generate a direct photon map containing approximately *nphotons* photons, and output to file *file*. This only accounts for direct illumination and is intended for debugging and validation of photon emission from the light sources, as the quality is too low for actual rendering.

−apC *file nphotons*

Generate a contribution photon map containing approximately *nphotons* photons, and output to file *file*. This may then be used by *rcontrib(1)* to compute light source contributions. When used with *rtrace(1)* or *rpict(1)*, contribution photon maps behave as regular global photon maps and yield cumulative contributions from all light sources.

With this option, *mkpmap* uses a modified photon distribution algorithm that ensures all light sources contribute approximately the same number of photons. Each photon indexes a primary hitpoint, incident direction, and emitting light source which can be used to bin contributions per light source and direction.

Mkpmap cannot generate a contribution photon map in combination with others in a single run, as it uses a different distribution algorithm. Other photon maps specified on the command line will be ignored.

−app *file nphotons bwidth*

Generate a precomputed global photon map containing a fraction of *nphotons* photons (specified with the **−apP** option, see below), and output to file *file*. This is a special case of the global photon map where the irradiance is evaluated for a fraction of the photon positions using *bwidth* nearest photons, and stored as photon flux; the remaining photons are discarded as their contributions have been accounted for.

This obviates the explicit irradiance evaluation by *rpict(1), rtrace(1)* and *rvu(1)*, thus providing a speedup at the expense of accuracy. The resulting error is tolerable if the indirect illumination has a low gradient, as is usually the case with diffuse illumination.

−apD *predistrib*

Photon predistribution factor; this is the fraction of *nphotons* which are emitted in a distribution prepass in order to estimate the remaining number of photons to emit in the main pass to approximately yield a photon map of size *nphotons*.

Setting this too high may yield more than *nphotons* in the initial pass with highly reflective geometry. Note that this value may exceed 1, which may be useful if the resulting photon map size greatly deviates from *nphotons* with a very low average reflectance.

−api *min_x min_y min_z max_x max_y max_z*

Define a rectangular region of interest within which to store photons exclusively; photons will only be stored within the volume bounded by the given minimum and maximum coordinates. Multiple instances of this option may be specified with cumulative effect to define compound regions of interest. This is useful for constraining photons to only the relevant regions of a scene, but may increase the photon distribution time.

WARNING: this is an optimisation option for advanced users (an elite group collectively known as *Ze Ekspertz***) and may yield biased results. Use with caution!**

-apI *pos_x pos_y pos_z rad*

Similar to **−api**, but with a spherical region of interest of given radius, centred at the given coordinates.

−apm *maxbounce*

Synonymous with **−lr** for backwards compatibility. May be removed in future releases.

−apM *maxprepass*

Maximum number of iterations of the distribution prepass before terminating if some photon maps are still empty. This option is rarely needed as an aborted prepass may indicate an anomaly in the geometry or an incompatibility with the specified photon map types (see **NOTES** below).

−apo[**+**|**-**|**0**] *mod*

Specifies a modifier *mod* to act as a *photon port*. All objects using this modifier will emit photons directly in lieu of any light sources defined with the *source* material. This greatly accelerates photon distribution in scenes where photons have to enter a space which separates them from the emitting light source via an aperture (e.g. fenestration, skylight) acting as a port.

In a typical daylight simulation scenario, a fenestration acts as a port to admit photons into an interior after emission from sky and solar sources. Multiple instances of this option may be specified.

By default, ports are oriented to emit in the halfspace defined by their associated surface normal. This can be overridden by specifying a trivalent suffix as follows:

+: Forward emission; this is equivalent to the abovementioned default behaviour.

-: Backward emission; the port is reversed and photons are emitted into the halfspace facing away from the surface normal.

0: Bidirectional emission; photons are emitted from both sides of the port.

Some typical situations that call for a reversed photon port include, for example:

(a) Using fenestrations as ports that were (for whatever reason) defined with outward facing normals,

(b) Using a **mist** primitive as a port, since this requires outward facing normals in order to register the photons as having entered the volume,

(c) Reorienting a port associated with a **bsdf** modifier, since inverting its normal would also reorient the BSDF and alter its behaviour.

Other oddball scenarios are conceivable. If in doubt, specify a bidirectional port orientation for a slight performance penalty, as photon emission is attempted from both sides. For well-defined port geometry with inward-facing normals, just use the default; doan' mess with da normalz.

Photon port geometry is discretised according to the **−dp** and **−ds** options. These parameters aid in resolving spatially and directionally varying illuminance received by the port from distant light sources, e.g due to partial occlusion or when using climate-based sky models.

−apO *modfile*
Read photon port modifiers from the file *modfile* as a more convenient alternative to multiple instances of **−apo**.

−apP *precomp*
Fraction of global photons to precompute in the range]0,1] when using the **−app** option.

−apr *seed*
Seed for the random number generator. This is useful for generating different photon distributions for the same octree and photon map size, notably in progressive applications.

−aps *mod*
Specifies a modifier *mod* defined as *antimatter* material to act as a virtual (i.e. invisible) receiver surface. Photons will be deposited on all surfaces using this modifier, just like regular materials, but will then be transferred through the surface without undergoing scattering; the surface therefore does not affect the light transport and simply acts as an invisible photon receiver. This is useful when photon irradiance is to be evaluated at points which do not lie on regular geometry, e.g. at workplane height with *rtrace*'s **-I** option. Without this workaround, photons would be collected from parallel but distant planes, leading to underestimation. Note that photons are only deposited when incident from the front side of the sensor surface, i.e. when entering the *antimatter*, thus the surface normal is relevant. *Mkpmap* reports an error if the specified modifier is not an *antimatter* material.

−apS *modfile*
Read virtual receiver surface modifiers from the file *modfile* as a more convenient alternative to multiple instances of **−aps**.

−ae *mod*

Add *mod* to the ambient exclude list, so that it will be ignored by the photon map. Objects having *mod* as their modifier will not have photons deposited on them. Multiple modifiers may be given, each as separate instances of this option.

WARNING: this is an optimisation option for advanced users and may yield biased results. It may also significantly increase photon distribution times. Use with caution!

−aE *file*

Same as *-ae*, except modifiers to be exluded are read from *file*, separated by whitespace. The RAYPATH environment variable determines which directories are searched for this file.

−ai *mod*

Add *mod* to the ambient include list, so that it will contribute to the photon map. Only objects having *mod* as their modifier will have photons deposited on them. Multiple modifiers may be given, each as separate instances of this option. Note that the ambient include and exclude options are mutually exclusive.

WARNING: this is an optimisation option for advanced users and may yield biased results. It may also significantly increase photon distribution times. Use with caution!

−aI *file* Same as *-ai*, except modifiers to be included are read from *file*, separated by whitespace. The RAYPATH environment variable determines which directories are searched for this file.

−bv[**+**|**-**]

Toggles backface visibility; enabling this causes photons to be stored and possibly scattered if they strike the back of a surface, otherwise they are unconditionally absorbed and discarded.

−dp *sampleres*

Angular resolution for sampling the spatial emission distribution of a modified light source or photon port (e.g. via *brightfunc*), in samples per steradian. This is required to numerically integrate the flux emitted by the light source and construct a probability density function for photon emission. The accuracy of photon emission from a modified source or port therefore depends on this parameter. The resolution may need to be increased with complex emission distributions in combination with caustics.

−ds *partsize*

Light source partition size ratio; a local light source object (or photon port in case of a distant source) is spatially partitioned to distribute the photon emission over its surface. This parameter specifies the ratio of the size (per dimension) of each partition to the scene cube, and may need to be reduced for modified light sources (e.g. via *brightfunc*) with high spatial variance, or for partially occluded photon ports.

−e *file* Redirect diagnostics and progress reports to *file* instead of the console.

−fo[**+**|**-**]

Toggles overwriting of output files. By default, *mkpmap* will not overwrite an already existing photon map file. This is to prevent inadvertently destroying the results of potentially lengthy photon mapping runs.

−ld *maxdist*

Limit cumulative distance travelled by a photon along its path to *maxdist*. Photon hits within this distance will be stored, and the photon is terminated once its path length exceeds this limit. This is useful for setting radial regions of interest around emitting/reflecting geometry, but may increase the photon distribution time.

WARNING: this is an optimisation option for advanced users (an elite group collectively known as *Ze Ekspertz*) **and may yield biased results. Use with caution!**

−lr *maxbounce*

Limit number of bounces (scattering events) along a photon path to *maxbounce* before being considered "runaway" and terminated. Photons paths are normally terminated via *Russian Roulette*, depending on their albedo. With unrealistically high albedos, this is not guaranteed, and this option imposes a hard limit to avoid an infinite loop.

WARNING: this is an optimisation option for advanced users (an elite group collectively known as *Ze Ekspertz*) **and may yield biased results. Use with caution!**

−ma *ralb galb balb*

Set the global scattering albedo for participating media in conjunction with the **−apv** option. See *rpict(1)* for details.

−me *rext gext bext*

Set the global extinction coefficient for participating media in conjunction with the **−apv** option. See *rpict(1)* for details.

−mg *gecc*

Set the global scattering eccentricity for participating media in conjunction with the **−apv** option. See *rpict(1)* for details.

−n *nproc*

Use *nproc* processes for parallel photon distribution. There is no benefit in specifying more than the number of physical CPU cores available (so doan’ even try). This option is currently not available on Windows -- so there, tuff luck.

−t *interval*

Output a progress report every *interval* seconds. This includes statistics about the currently emitting light source (including number of partitions), the total number of photons emitted, the number of each type stored, the percentage of the completed pass (pre or main), and the elapsed time.

NOTES

Parametrisation

Mkpmap recognises multiplier suffixes (k = 1000, m = 1000000) to facilitate the specification of *nphotons*, both in upper and lower case.

Distribution Algorithm

The photon distribution algorithm estimates the number of required photons to emit to arrive at the specified target count *nphotons* per photon map using a distribution prepass followed by a main pass. As a result, *mkpmap* generates the **approximate** number of photons specified, which can vary by up to 10% for typical scenes, but can be higher for scenes with unusually high or low reflectance. In this case, the predistribution factor **−apD** should be increased for scenes with low reflectance, and reduced for those with high reflectance.

There are situations which may prevent certain (or any) photon types from being generated, depending on the light source and material configuration. This typically occurs when attempting to generate a caustic photon map without specular materials present in the scene, or a volume photon map without participating media. Ill-configured light sources may also prevent indirect rays from reaching a surface, and thus no photons being deposited. In these cases, *mkpmap* will make a number of distribution attempts before terminating with an error. This can be adjusted with the **−apM** option.

Material Support

Not all materials are fully supported by the photon map extension. The *plasfunc*, *metfunc*, *transfunc*, *plasdata*, *metdata* and *transdata* materials currently only scatter photons diffusely, and will not produce caustics. The *brtdfunc* material only produces caustics via ideal (mirror) specular reflection and transmission. For more realistic scattering behaviour, use the newer *bsdf* material instead.

Virtual light sources (normally enabled with the *mirror* material) are disabled with the photon map, as the resulting caustics are already accounted for.

Virtual Receiver Surfaces

Since photons are surface bound, the density estimate is only asymptotically correct when performed at points which lie on the scene geometry. The irradiance is underestimated for arbitrarily placed points when photons are collected from distant surfaces. *Mkpmap* offers a workaround with a virtual receiver surface using the *antimatter* material; see the **-aps** and **-apS** options for details.

EXAMPLES

The following command generates a global photon map *bonzo.gpm* and a caustic photon map *bonzo.cpm* containing approximately 10000 and 100000 photons, respectively, with progress report every 5 seconds:

```
mkpmap −apg bonzo.gpm 10k −apc bonzo.cpm 100k -t 5 bonzo.oct
```

Generate a global photon map containing 80000 photons, then precompute the diffuse irradiance for 1/4 of these with a bandwidth of 40 photons:

```
mkpmap −app bonzo-precomp.gpm 80k 40 −apP 0.25 bonzo.oct
```

Generate 1 million global photons by emitting them from external light sources of type *source* into a reference room via a fenestration with modifier *glazingMat* acting as photon port, with inward-facing normal:

```
mkpmap −apg refRoom.gpm 1m −apo glazingMat refRoom.oct
```

Generate a contribution photon map containing 10 million photons to bin light source contributions with *rcontrib(1)*:

```
mkpmap −apC bonzo-contrib.gpm 10m bonzo.oct
```

BUGS

The focus of a spotlight source, as defined by the length of its direction vector, is ignored by the photon map; photons are unconditionally emitted from the light source surface, which can lead to deviations from standard RADIANCE.

Light sources simply absorb incoming photons.

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 LU-204/10-2*, "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

rpict(1), rtrace(1), rvu(1), rcontrib(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]