Radiance Course Day 1

DAYLIGHT SIMULATION

Based on Chapter 6 of ***Rendering with Radiance***

Dr. John Mardaljevic

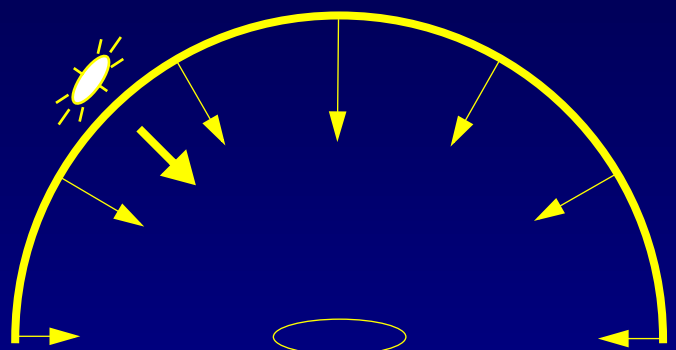
jm@dmu.ac.uk

Institute of Energy and Sustainable Development
De Montfort University
Leicester, UK

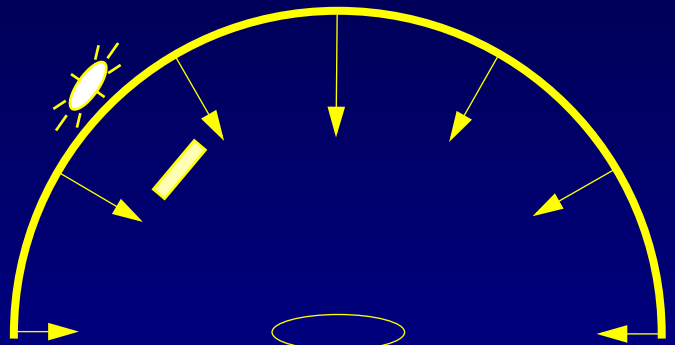
http://www.iesd.dmu.ac.uk/~jm

MEASURING DAYLIGHT

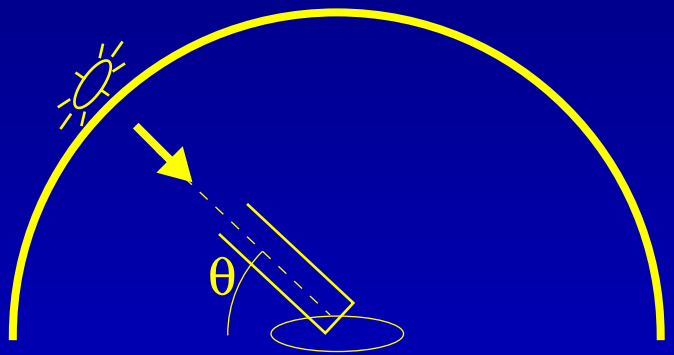
Real skies and suns are usually characterised using these quantities



(a) Global horizontal (sun & sky)



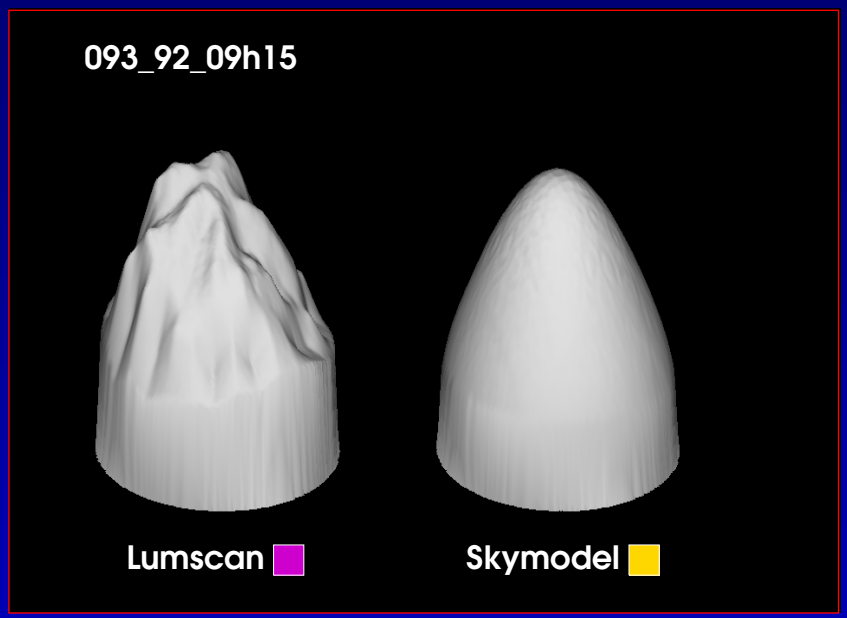
(b) Diffuse horizontal (sky only)

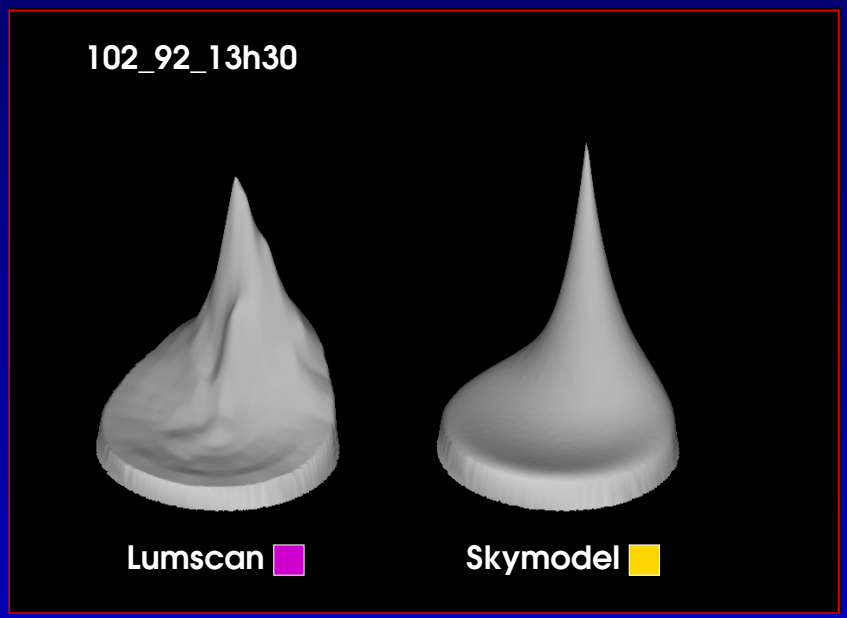


(c) Direct normal (sun only)

Real and Modelled Sky Patterns

Actual skies vary from moment to moment and it is impossible to reproduce actually occurring sky patterns without detailed meteorological data. However, sky models do a reasonable job of estimating patterns from basic data - we use these in *Radiance* to 'create' skies.

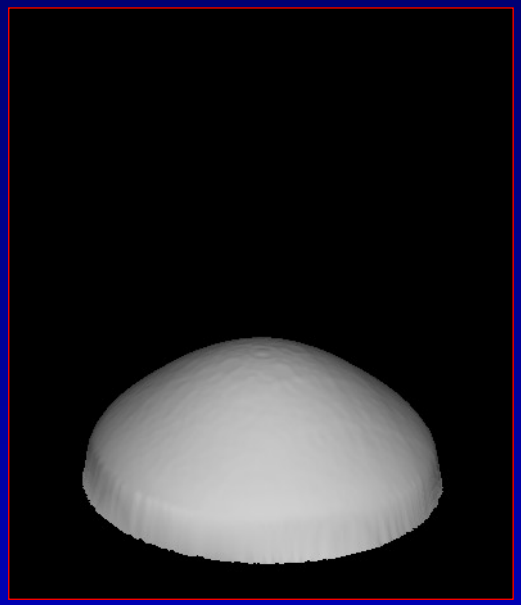




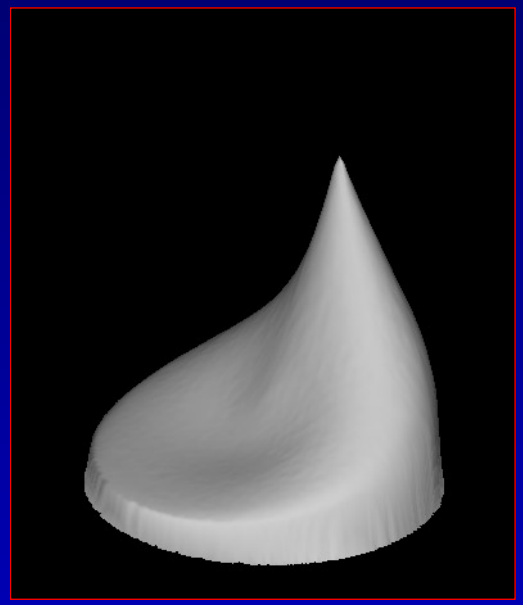
Real skies compared with sky models

Sky Models

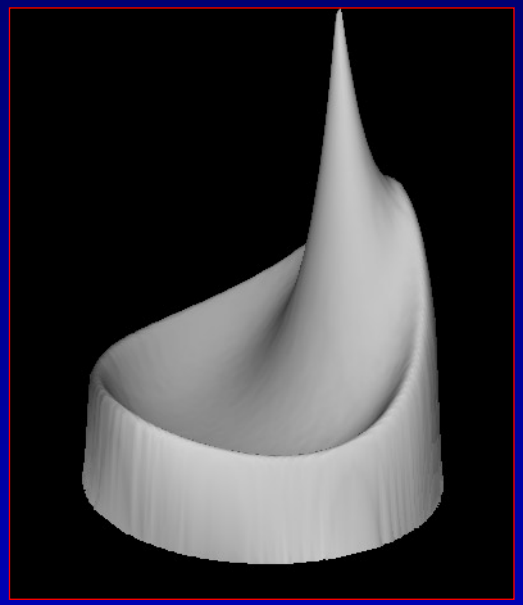
The standard overcast sky pattern is three times brighter at the zenith than at the horizon, and it does not vary in brightness with azimuth. An intermediate or clear pattern is brightest about the sun position - this is the circumsolar region.



Overcast



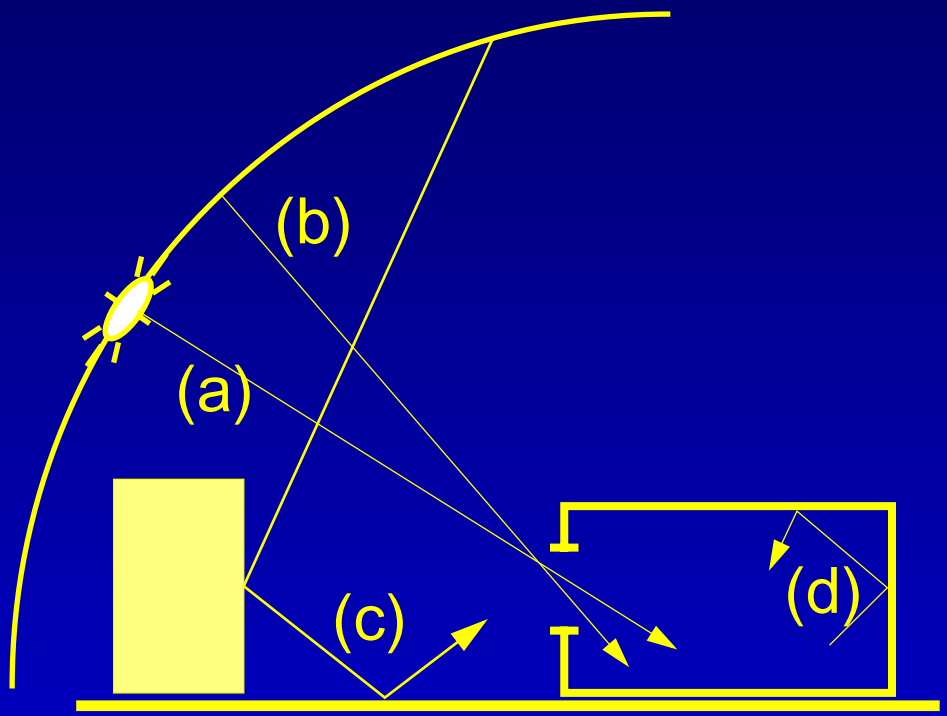
Intermediate



Clear

DAYLIGHT INDOORS

It helps to characterize the daylight entering a space by its origin - sun or sky - and the path by which it has arrived - directly from the source or by reflection.



(a) direct sun
(b) direct sky
(c) externally reflected
(d) internally reflected

Evaluation Techniques and Accuracy

Daylight simulation for interior spaces can be divided into two modes of evaluation:

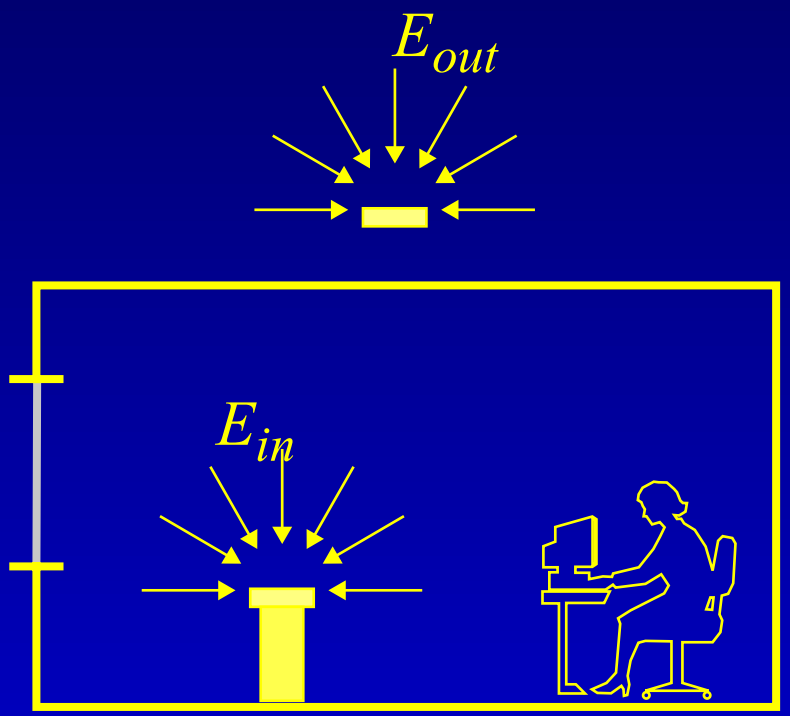
- Quantitative, or numerical
- Qualitative, or visual.

The CIE Standard Overcast Sky is mostly used for quantitative analysis, e.g. daylight factor prediction.

Skies with sun are mostly used for qualitative analysis, e.g. visual impression, shading studies.

The Daylight factor Approach

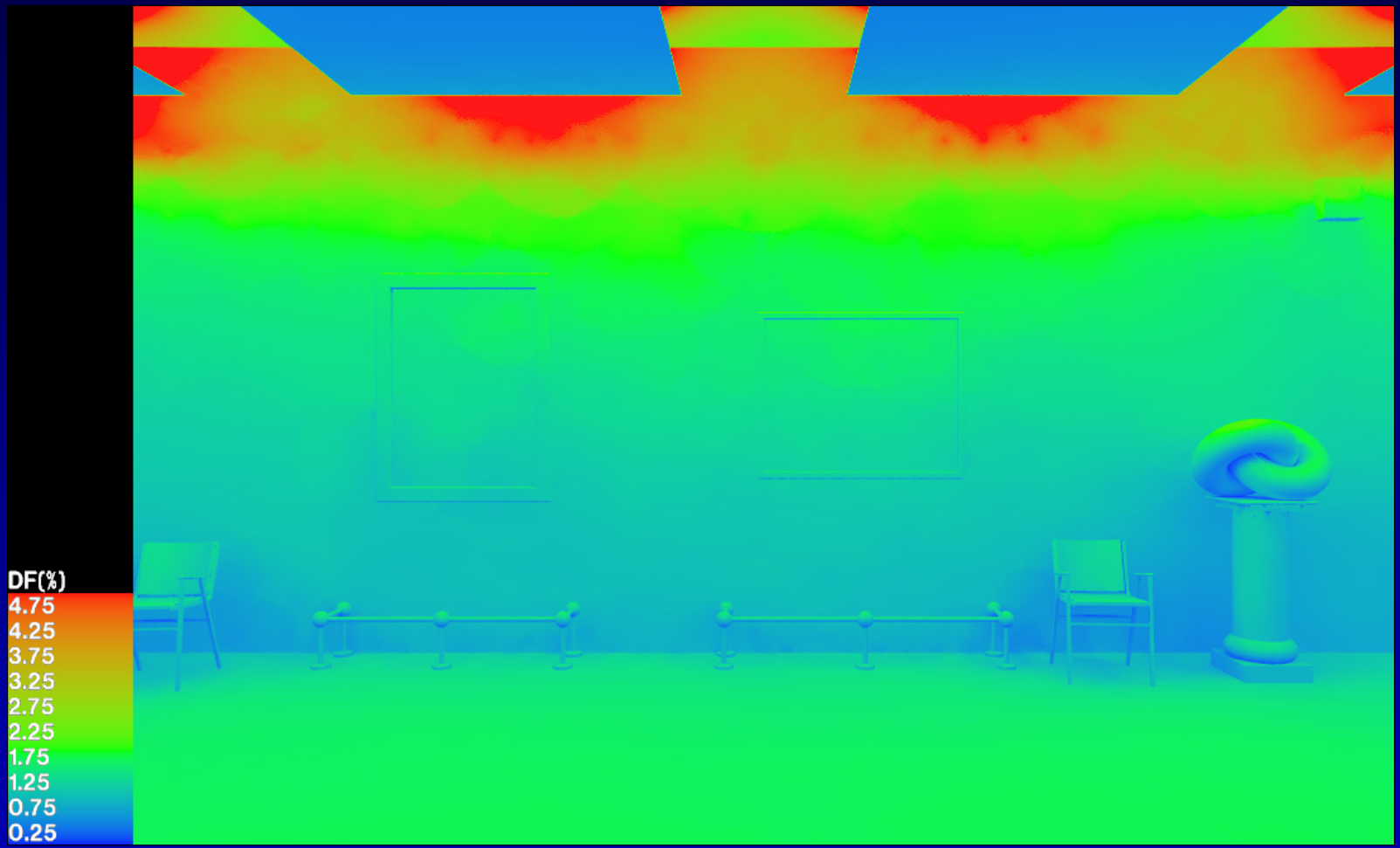
The daylight factor at any point is the ratio of the interior illuminance at that point to the global horizontal illuminance under CIE standard overcast sky conditions (normally expressed as a %).



$$DF = \frac{E_{in}}{E_{out}} \cdot 100$$

Daylight Factor Example - Scene

DAYLIGHT FACTOR EXAMPLE - DF MAP



Pictures, Numbers, and Accuracy

For a conventional office scene constructed with typical materials, an accurate (±10%) illuminance prediction usually requires four or more ambient bounces.

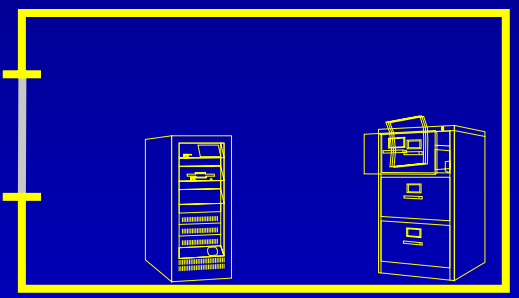
For visual impression it is the directionality of the ambient shading that lends realism, for example the brightening of surfaces near a sun patch. This can be largely achieved with just one or two ambient bounces.

Empty And Cluttered Scenes

The more cluttered the scene (from the view point), the harder the inter-reflection calculation has to work. This means more frequent sampling if we wish to avoid blotches, with the resulting computational overhead.

Use uncluttered scenes for DF predictions to speed up computation

Add clutter to scenes for renderings to lend realism



Case Study I: Creating The Luminous Environment

The sky and sun are, on an architectural scale, very distant from the local scene. So, the unobstructed view of the sky will be identical for all observers placed anywhere in the scene. In *Radiance*, the sky is specified as a **source solid angle** rather than a dome of actual extent. From our local "flat Earth," the sky appears to be a luminous hemisphere. Thus, we model it as a source whose angle is 180 degrees, and we aim the centre of the source directly upward, that is, toward the zenith.

EXAMPLE: UNIFORM SKY

The scene file, *sky_uni.rad*, describes our entire scene, which is simply a hemispherical sky of unit radiance:

```
# uniform brightness sky (B=1)
void glow sky_glow
0
0
4 1 1 1 0
sky_glow source sky
0
0
4 0 0 1 180
```

CREATE THE OCTREE AND RUN `rtrace`

```
% oconv sky_uni.rad > sky_uni.oct
```

Now execute the **`rtrace`** program to determine the horizontal irradiance due to the uniform sky. A typical command might look like this:

```
% echo "0 0 0 0 0 1" | rtrace -h -I+ -w -ab 1
sky_uni.oct
```

which writes to the standard output the simulated spectral (RGB) irradiance values:

```
3.141593e+00  3.141593e+00  3.141593e+00
```

Note, **`-ab 1`** is used so that rays are sent to sample the **`glow`** sky.

The Same In Greek

For any sky of radiance $B(\theta, \phi)$ the horizontal irradiance is given by:

$$I = \int_0^{2\pi} \int_0^{\pi/2} B(\theta, \phi) \sin\theta \cos\theta \, d\theta \, d\phi$$

where for a uniform sky, $B(\theta, \phi) = B$, giving:

$$I = B \int_0^{2\pi} \int_0^{\pi/2} \sin\theta \cos\theta \, d\theta \, d\phi = \pi B$$

which, for a sky of unit radiance, gives

$$I = \pi$$

EXAMPLE: CIE OVERCAST SKY

This sky varies in brightness with altitude. We use the **gensky** program to create this sky:

```
% gensky -ang 45 0 -c -b 1
```

which writes the following to the standard output:

```
# gensky -ang 45 0 -c -b 1
# Ground ambient level: 0.8
void brightfunc skyfunc
2 skybr skybright.cal
0
3 2 1.00e+00 1.56e-01
```

The output from the **gensky** program provides a brightness function (**skyfunc**) that we can apply as a *modifier* to the **glow** material.

WE NEED TO ADD A SKY DESCRIPTION...

...to give **skyfunc** something to modify:

```
# cie_ovc.rad
# CIE overcast sky (Bz = 1)
!gensky -ang 45 0 -c -b 1
skyfunc glow sky_glow
0
0
4 1 1 1 0
sky_glow source sky
0
0
4 0 0 1 180
```

The (RGB) radiance of the sky now is **skyfunc** *multiplied* by the **glow** radiance (1 1 1).

Compute Irradiance As Before

Create the octree for this scene:

```
% oconv cie_ovc.rad > cie_ovc.oct
```

and then calculate the horizontal irradiance using **rtrace** (pipe the output through **rcalc** to obtain the achromatic irradiance directly):

```
% rtrace -w -h -I+ -ab 1 cie_ovc.oct < samp.inp
| rcalc -e '$1=$1*0.265+$2*0.670+$3*0.065'
```

which produces the value:

```
2.434001
```

The exact theoretical value for irradiance from the CIE overcast sky is $7\pi B_z/9$ = 2.443451.

Example: CIE Overcast Sky Defined by Its Horizontal Illuminance

Before we can tackle real-world problems, we need to be able to relate the more usual daylighting quantities of luminance and illuminance to the radiance and irradiance inputs required by `gensky`.

Although *Radiance* calculates in units of radiance/irradiance, we use a constant value (179 lm/W) to convert to luminance/illuminance, or vice versa.

A realistic horizontal illuminance for a (brightish) overcast sky is 10,000 lux. A convenient figure to work with; e.g. 500 lux corresponds to a DF of 5%.

FIRST MODIFY gensky...

...to produce a 10,000 lux sky. The irradiance that corresponds to this illuminance is 10,000/179 = 55.866 W/m^2. The line for the **gensky** command should now look like this:

```
!gensky -ang 45 0 -c -B 55.866
```

The rest of the file as before. Run **oconv** as before, then execute a slightly modified **rtrace** command:

```
% rtrace -w -h -I+ -ab 1 cie_ovc.oct < samp.inp
| rcalc -e '$1=($1*0.265+$2*0.670+$3*0.065)*179'
```

The calculation returns the value

```
9977.17002
```

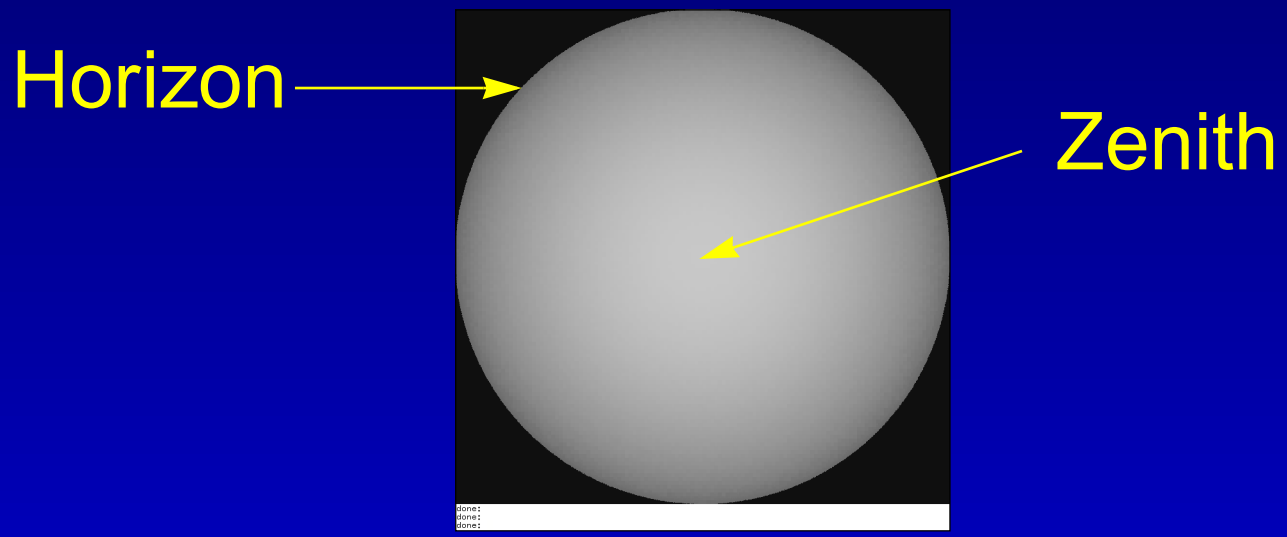
Pretty close to our starting value of 10,000 lux.

VIEWING THE SKY

Use the interactive viewing program **rview**:

```
% rview -vta -vp 0 0 0 -vd 0 0 1 -vu 0 1 0 -vh
180 -vv 180 sky_ovc.oct
```

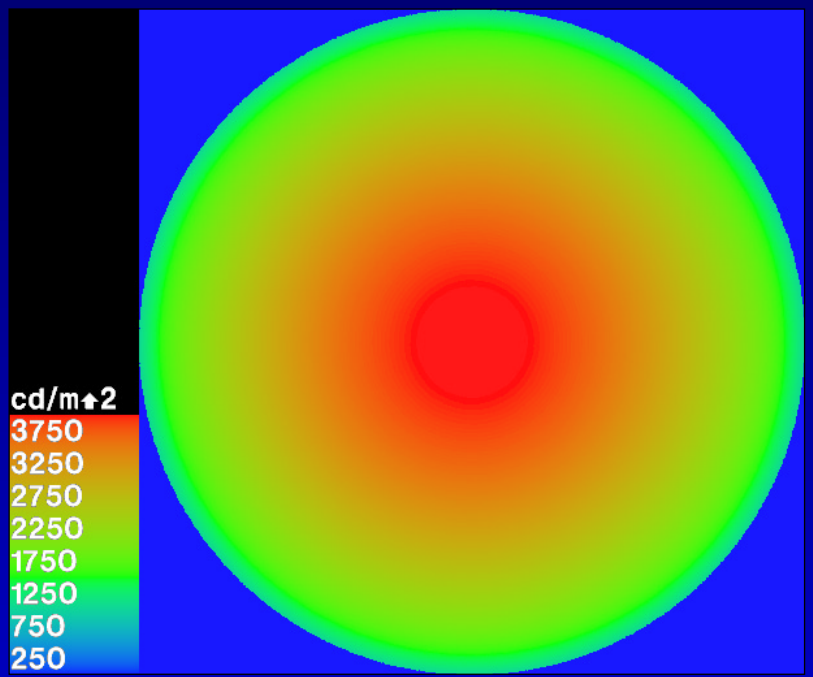
This command starts rview 'looking' up with an angular fisheye view.



FALSECOLOUR IMAGE OF CIE SKY

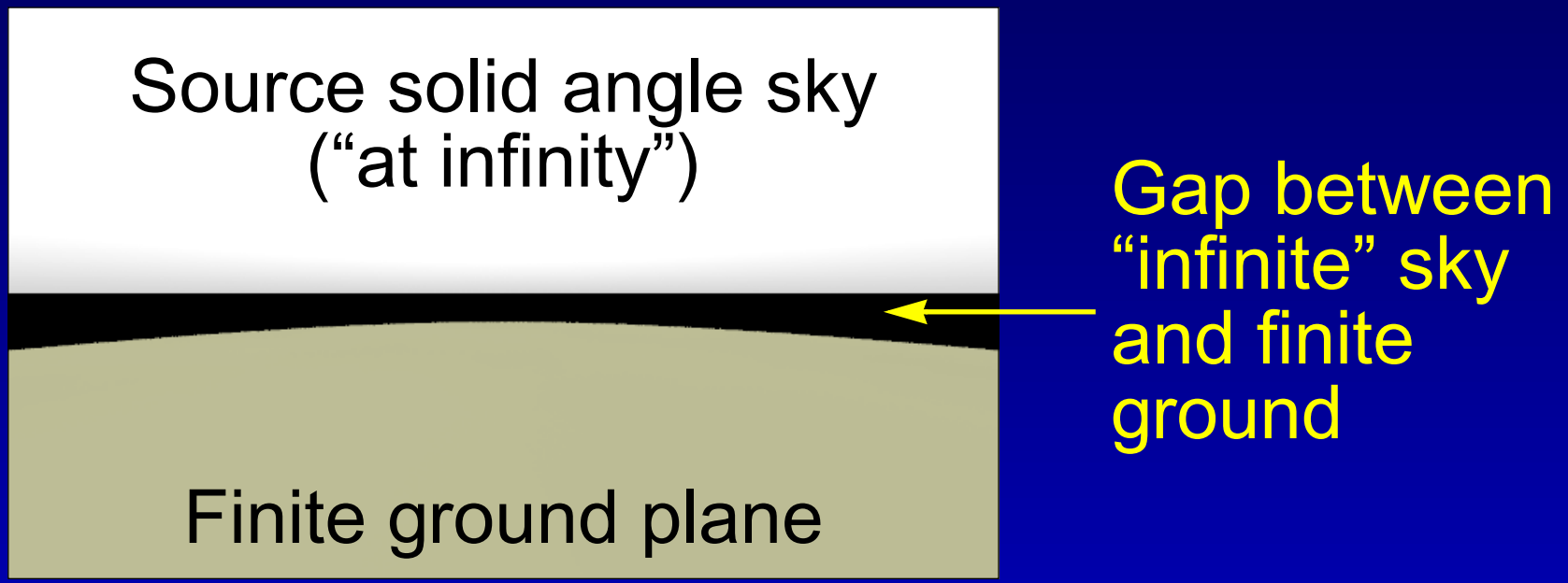
Execute the following command to create a rendering (i.e. radiance map) of the view of the sky and pipe it directly into the **falsecolor** program:

```
% rpict -vf ang180.vf sky_ovc.oct   \
    | falsecolor -s 4000 -l cd/m^2 > ovc_lum.pic
```



Ground "Glow": An "Upside-Down" Sky

At the horizon, the sky "meets" the ground. An actual ground plane of finite extent, say, a disc, will always fall short of an "infinite" horizon.



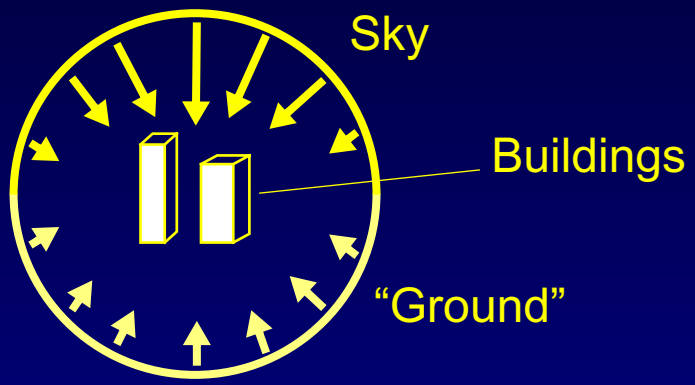
TO AVOID GAPS AT THE HORIZON...

...we use an upside-down sky to represent a luminous ground. To do this, we apply the **skyfunc** modifier to a 180-degree glow source, where the direction vector is pointing *downward*, e.g:

```
skyfunc glow ground_glow
0
0
4 1 1 1 0
ground_glow source ground
0
0
4 0 0 -1 180
```

Unlike sky, the ground has a constant brightness.

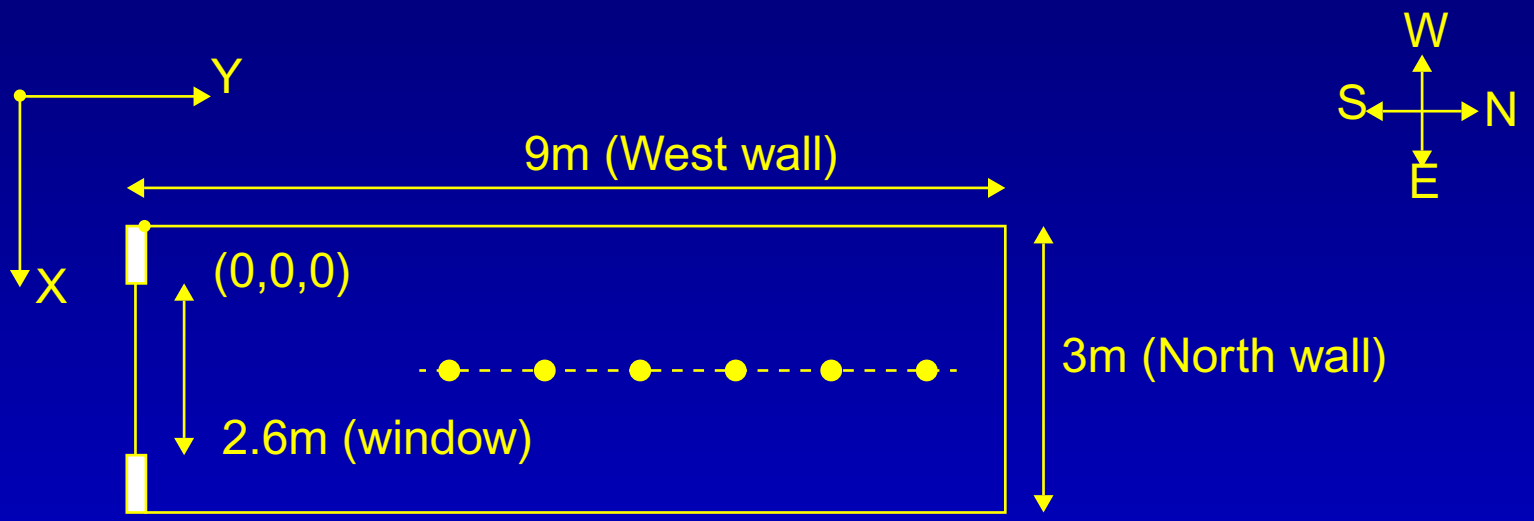
The Luminous "Envelope"



We now have a seamless sky and ground. Although the ground radiance is based on the sky's horizontal irradiance, putting something between the sky and the ground will not affect the brightness of either. In other words, no matter how built-up the model becomes, with nearby tall structures and so on, the ground radiance (where it is visible) will be the same as for an empty scene. Therefore, all scenes should include a local ground plane that participates in the inter-reflection calculation.

Case Study II: Predicting Internal Illuminances

Here we demonstrate how to predict DF levels for a simple scene, how to automate the execution of the `rtrace` program and how to test for convergence in the ambient calculation.



COMPUTING DAYLIGHT FACTOR VALUES

Executing the **rtrace** command from a shell script is a convenient way to automate systematic explorations of parameter settings:

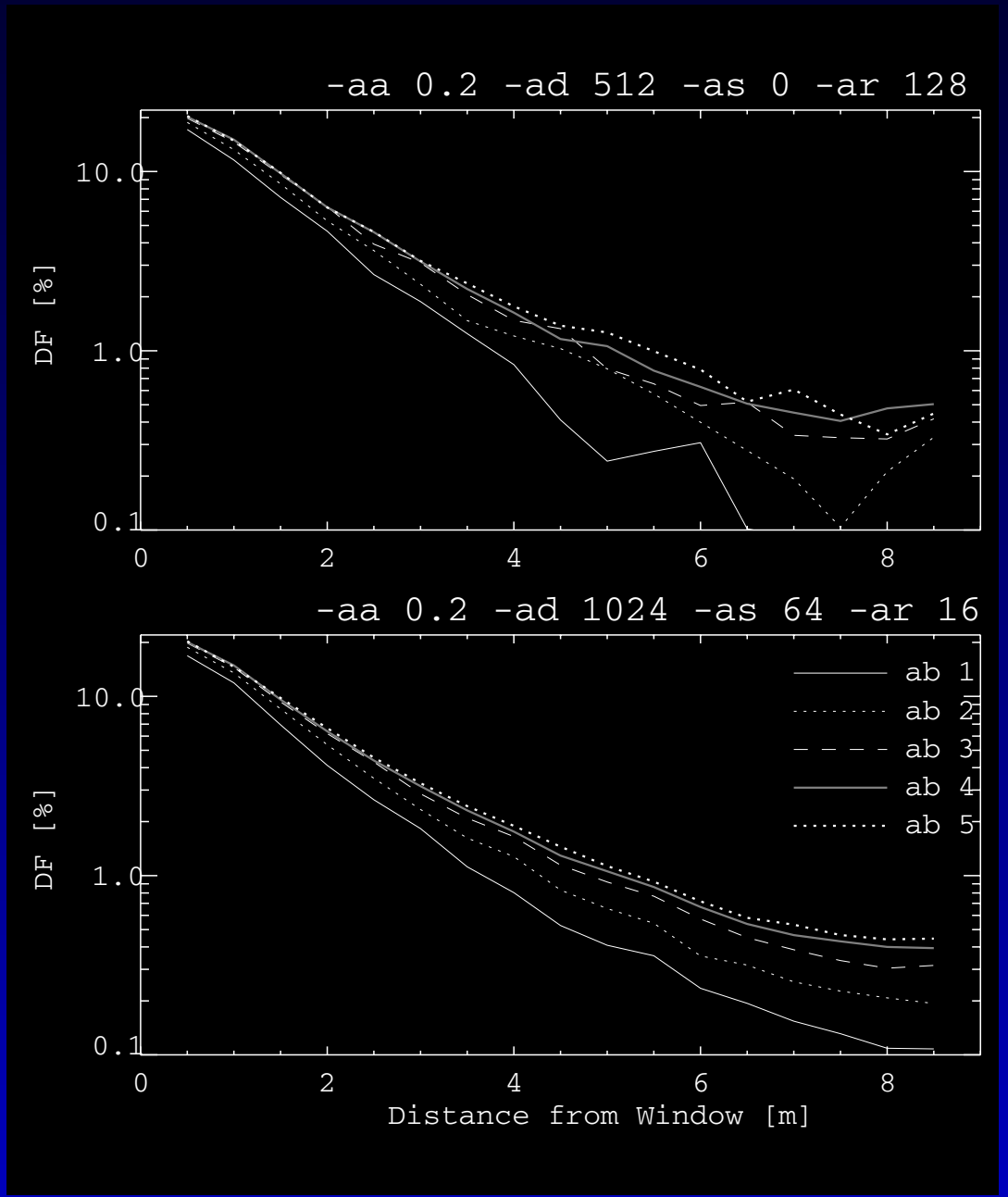
```
#!/bin/csh -f
# loop through ab
foreach ab (1 2 3 4 5)
echo "Doing ambient bounces" $ab
# Calculate DF
rtrace -w -h -I+ -ab $ab -aa 0.2 -ad 512 \
-as 0 -ar 128 scene.oct \
< samp1.inp | rcalc -e\
'$1=($1*0.265+$2*0.670+$3*0.065)*179/10000*100'
end
```

DF Plots

For **-ad 512** the DF lines don't look right - they should be smooth and rank order maintained.

Doubling to **-ad 1024** produces more plausible DF lines.

(Notice **-ar** reduced)



ANALYSIS

Examination of the data for `-ad 512 -ab 1` reveals that, for several points at the back of the room, the DF was predicted to be zero. This tells us that too few rays were spawned to guarantee adequate sampling of the window from all points in the DF plane. To remedy this, we should set `-ad` to a higher value, say `1024`. We can further improve our estimates at `-ab 1` by enabling the ambient supersampling option `-as`. The value we set for `-as` is the number of extra rays that will be used to sample areas in the divided hemisphere that appear to have high variance.

The basic tenets for setting the ambient parameters are:

1. Set **-ad** high enough to capture the visible luminous features at the first bounce.

2. Give sufficient ambient bounces to redistribute the light.

3. Set the remaining ambient parameters to sufficiently high resolution to deliver *acceptably* smooth results.

Note - Use the minimum number of ambient bounces for tests, **-ab 1** is often enough. If the DF plot (or image) for **-ab 1** is lumpy, just setting a higher **ab** will not fix it.

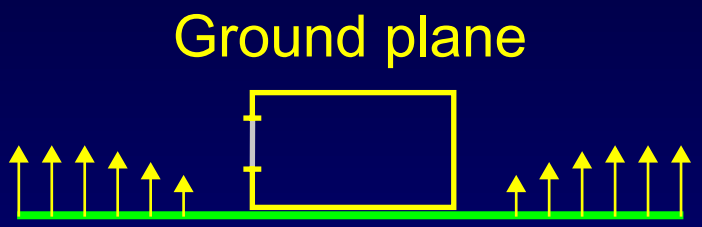
The Dayfact Script

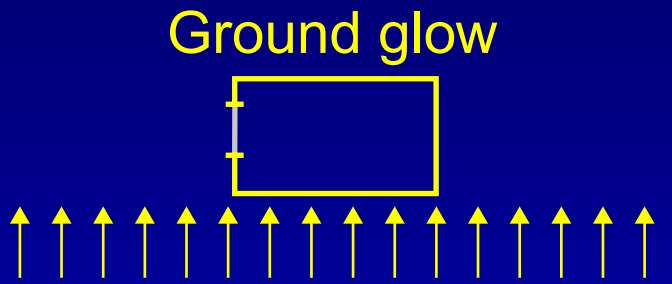
A user-friendly interface to the illuminance prediction capabilities of `rtrace`. The script essentially performs the same `rtrace` illuminance calculation shown above, but in addition it can create contour plots of:

- Workplane illuminance.
- Workplane daylight factors.
- Potential savings resulting from daylight illumination based on a given lighting design level.

Case Study III: Introducing Complexity

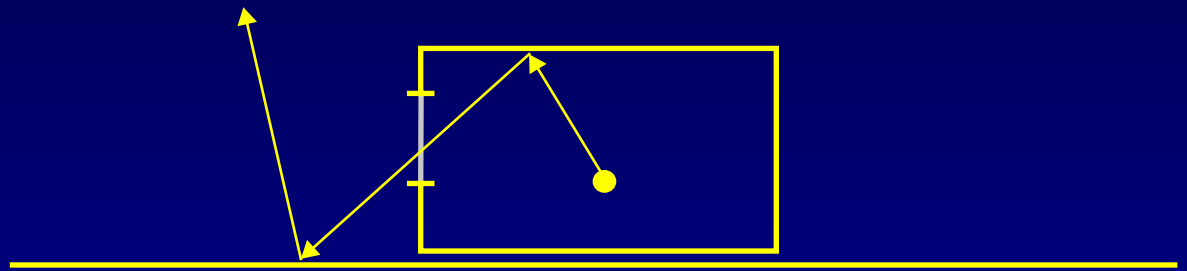
Add a ground plane and nearby building. The diffuse reflectance for the ground material is the same as the ground plane reflectance used in the **gensky** command (0.2, or 20%). The effect of the ground plane will be to slightly lower the DFs calculated in the preceding example, because we are replacing (locally) a ground **glow** of constant radiance with a material whose brightness now depends on the geometry and reflectance of nearby objects as well as the sky.



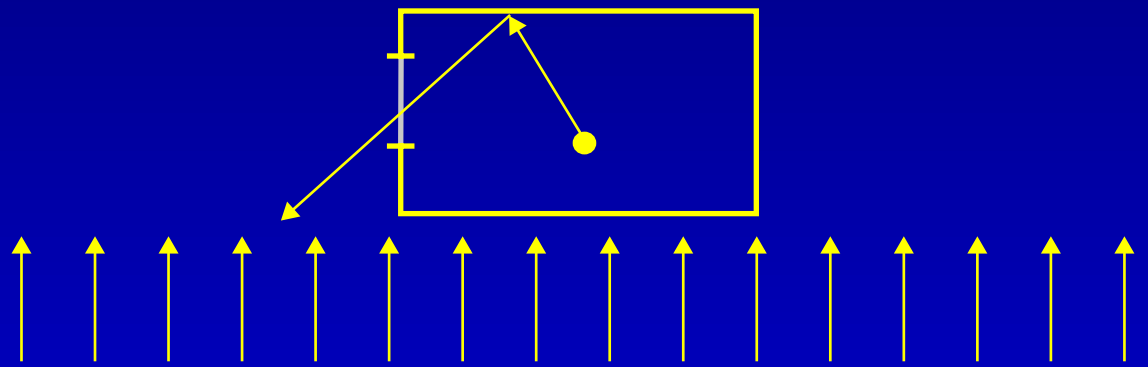


THE SIMULATION HAS TO WORK HARDER...

...because **rtrace** now has to evaluate the ground plane brightness.



For **-ab 3** ray samples ground plane radiance calculated from sky brightness



For **-ab 2** ray samples ground glow radiance

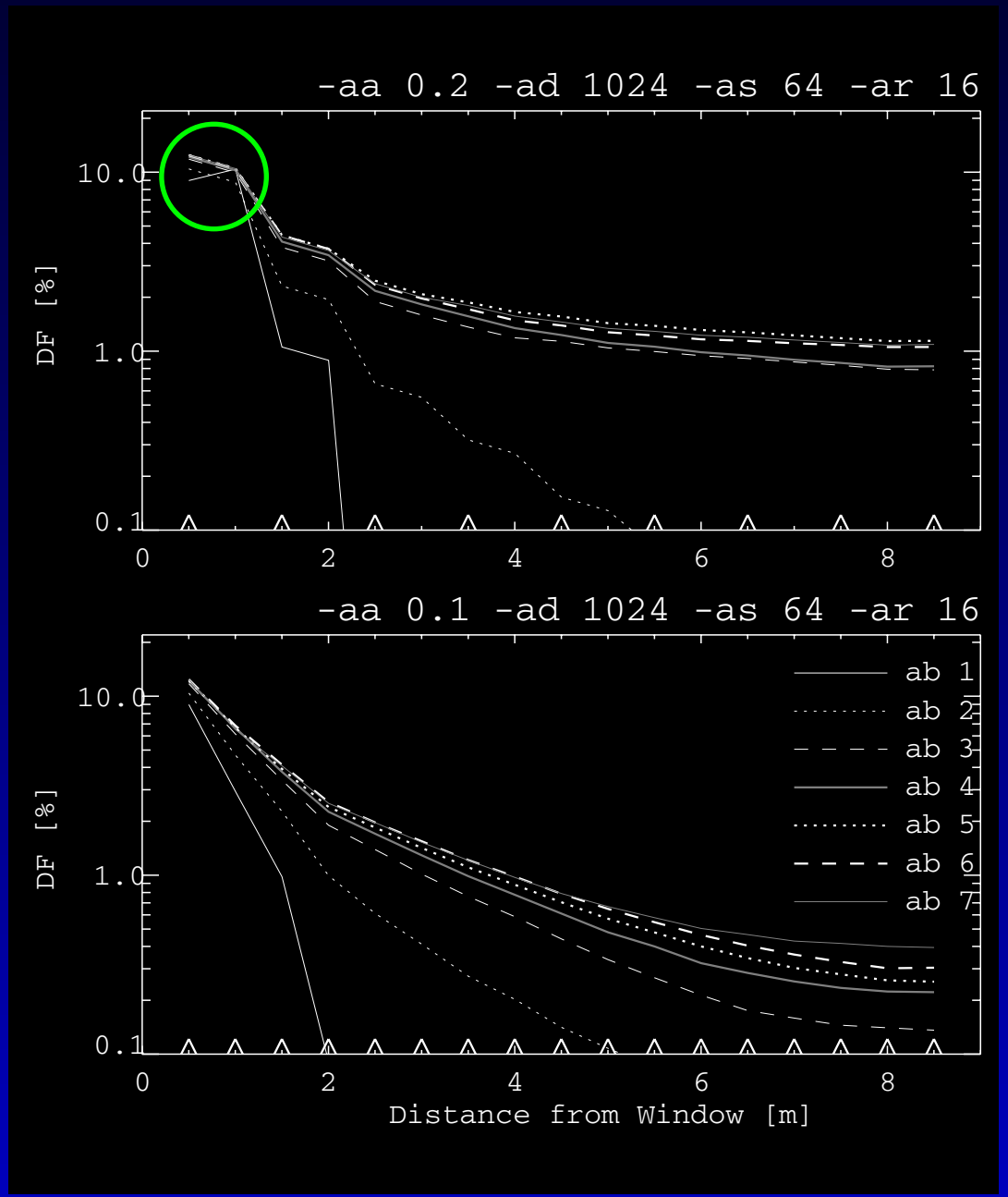
DF Plots For The New Scene

Results for two ambient accuracy (`-aa`) settings.

The upper plot is surely unsound. For `-ab 1`, the DF *increases* away from the window.

For `-aa 0.1`, the results look plausible.

Why the dramatic difference?



ANALYSIS

This example was contrived to create the circumstances under which the irradiance interpolation algorithm would, for certain parameter combinations, perform relatively poorly. To appreciate why this has happened, we need to recognize that irradiance interpolation can occur across the points supplied to rtrace in the same way that it can across the surfaces (i.e., pixels) computed by **`rpict`**. In other words, hemispherical sampling (at the first level) will not necessarily be initiated from every point in the DF plane supplied to **`rtrace`**.

ANALYSIS (CONT.)

Hemispherical sampling at the first level will always be initiated from the first point supplied to rtrace. From these HS rays, the ambient calculation will predict the way the indirect irradiance is changing about that point - this is the indirect irradiance gradient. The calculation also evaluates an estimation of error associated with the prediction for the irradiance gradient. These quantities, together with the ambient accuracy parameter, are used to determine a “radius of validity” for the gradient estimate.

ANALYSIS (CONT.)

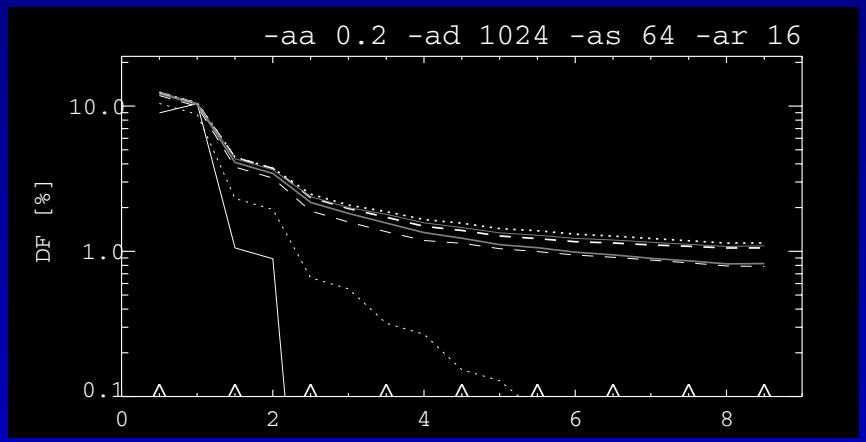
If the next point supplied to **`rtrace`** is within this radius, the indirect irradiance is evaluated from the gradient estimate and not from further HS. In other words, the value is obtained by a form of interpolation rather than by actual sampling.

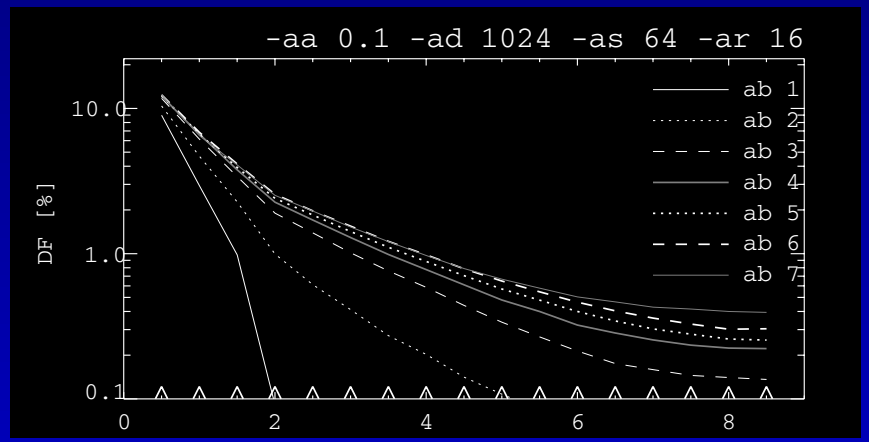
Factors that influence the scale over which interpolation may occur are:

- The ambient accuracy (**`-aa`**).
- The ambient resolution (**`-ar`**).
- The maximum scene dimension.

Analysis (Cont.)

The minimum possible spacing between HS points is the maximum scene dimension multiplied by the ambient accuracy divided by the ambient resolution. We confirm that the poor results for -aa 0.2 arose from interpolation by plotting on the abscissa the points in the DF plane from which HS was initiated (Δ markers).





ANALYSIS (CONT.)

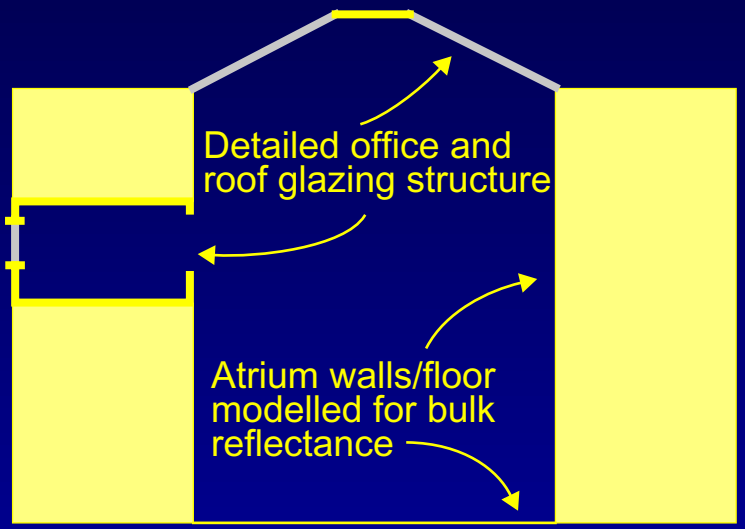
For `-aa 0.1`, sampling was initiated from all the points supplied to `rtrace`; for `-aa 0.2`, it was from every other point. Note that a doubling of the value for ambient resolution (i.e., from 16 to 32) would not necessarily have effected the same cure. This is because the `-ar` parameter acts as a limiting device. If you are already running up against the `-ar` limit, increasing the setting will result in a higher density of sampling. If the limit has not been reached, then increasing `-ar` should have no effect.

DF Prediction: Tricks of the Trade

(Applies to most work where exacting results are required)

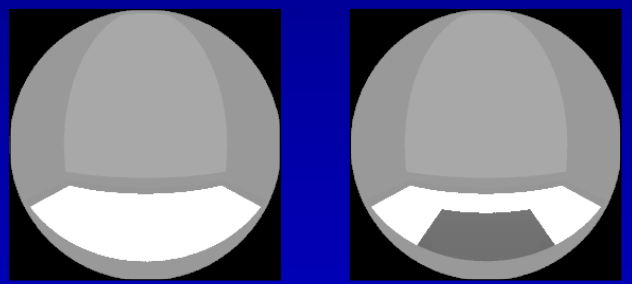
Appropriate Complexity

- Cut down on the clutter.
- Only model those surfaces not in view that are important for light transfer.



Views From The DF Plane

- Visualize the scene from one or more points of interest.



CASE STUDY IV: CREATING SKIES WITH SUN

There are two *Radiance* sky generator programs. The "official" program, which is part of the standard *Radiance* release, is called **`gensky`**; it offers a selection of sky model types based on CIE standards. The other is called **`gendaylit`**; it is one of the many *Radiance* extension programs, that is, it is not part of the standard release but is freely available to all users.

`gensky` can generate CIE standard clear, overcast and intermediate skies, with/without sun.

`genday lit` generates the Perez All Weather sky.

USING `gensky`

The only way to be certain of the sky and sun brightness is to supply them as **`gensky`** arguments. The sky brightness can be specified in terms of either the zenith radiance (**`-b`** option) or the horizontal diffuse irradiance (**`-B`** option). The sun brightness is either given directly (**`-r`** option) or evaluated from the horizontal direct irradiance (**`-R`** option). Most users will want to generate sun and skies based on either measured or yardstick values for global horizontal and diffuse horizontal illuminance.

gensky EXAMPLE

We want to generate a sun and intermediate sky description from these measured quantities: a global horizontal illuminance of 66,110 lux and a diffuse horizontal illuminance of 41,881 lux. The sun position was recorded as altitude 49.6 degrees and azimuth 222.5 degrees.

The altitude is the angle in degrees above the horizon and the azimuth is measured as degrees east of north. Note that this azimuth convention is different from the one used in *Radiance*, which is degrees west of south, so we need to subtract 180 degrees from the measured azimuth value.

gensky Example (Cont.)

From the illuminance quantities, we need to deduce the correct gensky arguments for the -B and -R options—they are the easiest to figure out from what we have.

$$\text{horizontal diffuse irradiance} = \frac{\text{horizontal diffuse illuminance}}{\text{luminous efficacy}}$$

$$\mathbf{233.97} = \frac{41881}{179}$$

$$\text{horizontal direct irradiance} = \frac{\text{hor. global ill.} - \text{hor. diffuse ill.}}{\text{luminous efficacy}}$$

$$\mathbf{135.35} = \frac{(66110 - 41881)}{179}$$

gensky Example (Cont.)

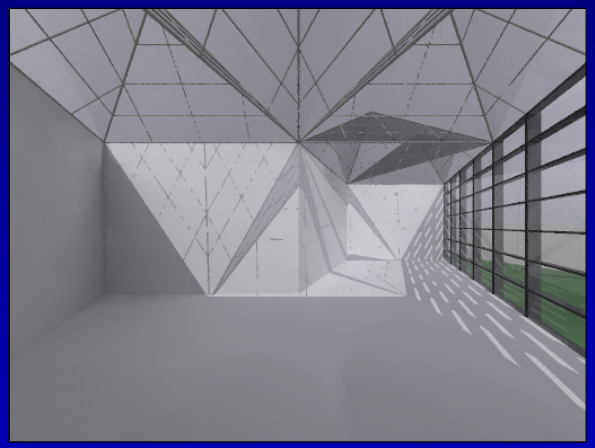
Thus, our **gensky** command, executed in a scene file, would look like this:

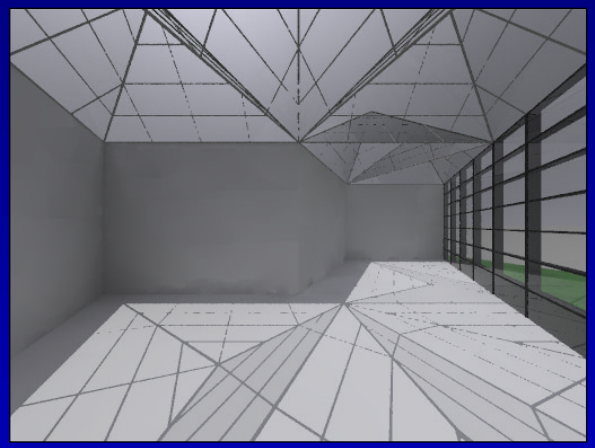
```
# Intermediate sky with sun
# Igh=66,110 lux, Idh=41,881 lux.
!gensky -ang 49.6 42.5 +i -B 233.97 -R 135.35
skyfunc glow sky_glow
0
0
4 0.986 0.986 1.205 0
sky_glow source sky
0
0
4 0 0 1 180
```

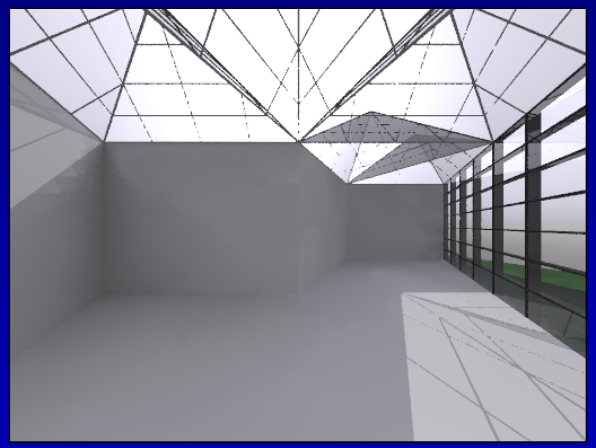
Notice the sky is given a blueish hue.

Time Of Day Image Sequence

The progression of the solar beam in a space can be shown by images generated for different times of day. The creation of these can be automated by treating the `gensky` time parameters as shell variables. [See shell script code in RwR].







Rendering Scenes Illuminated by Sunny Skies

The bad news

You will by now be aware that it is impossible to recommend a single set of rendering parameters that can guarantee an efficient solution for every conceivable design type. **Bummer!**

The good news

It should, however, be possible to anticipate from the actual design and lighting conditions the best approach to solving the problem. **Okay, so...**

The Simple Space Lit by a Sunny Sky

Recreate the simple room scene octree using the intermediate sky description. Include the ground plane but leave out the external obstruction. Use the `rview` interactive renderer to view the scene from somewhere at the back of the room, looking toward the window at about eye-level height. All that you will see at first is the sky through the window and the sun patch on the floor/wall.

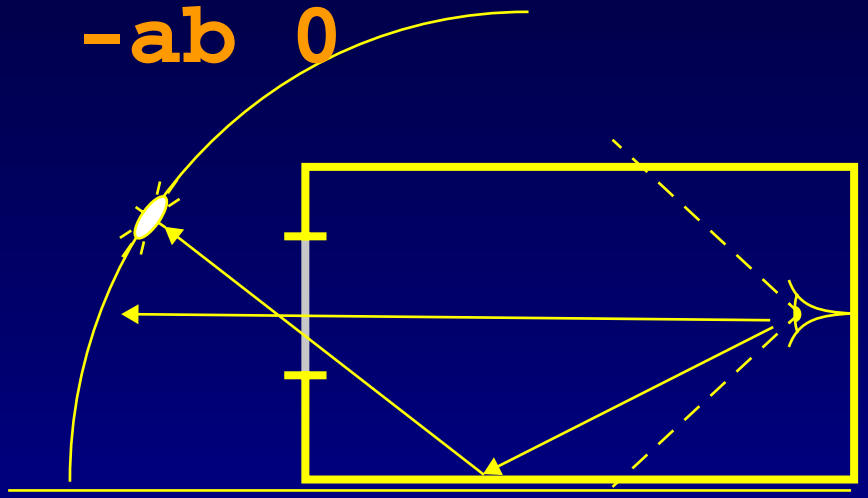
Initiate the inter-reflection calculation by setting the number of ambient bounces to 1 (`set ab 1`). Restart the image with the command `new`.

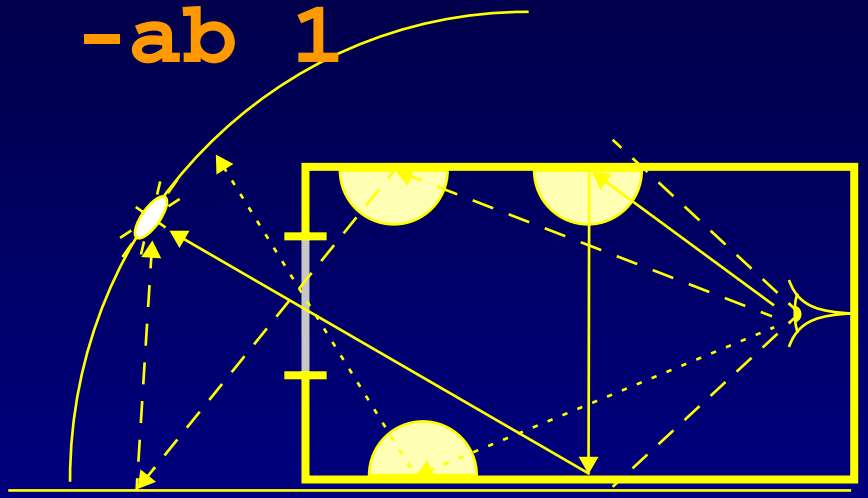
WHAT WE SEE IS...

...the sky through the window and the sun patch on the floor/wall (`ab 0`). For `ab 1` we begin to see more of the room, but it is blotchy because the default ambient parameter settings for `rview` are fairly coarse.

Note that `rview` does not flush the ambient cache when ambient parameters are changed - you'll need to restart `rview` to see the effect.

POSSIBLE LIGHT TRANSFERS ARE:





With the ambient calculation switched off, we see the sky (glow) through the window and whatever sun patches are directly visible from the view point.

With the inter-reflection calculation switched on, several other routes to the eye (that is, the camera) become possible via hemispherical sampling. Three are shown.

WITH INTER-REFLECTION SWITCHED ON...

...it is important to appreciate the element of chance at work whenever hemispherical sampling is used. If the number of initial sampling rays (**-ad**) were set too small, the calculation might, for example, “miss” the sun patch even though it was “visible” from the point at which the rays were spawned. By the same token, an unrepresentative chance “hit” of a small sun patch by one of the sampling rays can produce a gross overestimate for indirect irradiance. In a rendering, the artifacts associated with ambient undersampling are all too apparent - bright and dark blotches.

TO MINIMISE BLOTCHES...

...we need to set a sufficiently high value for the number of initial sampling rays. HS is generally too expensive to initiate at every surface visible from the eyepoint (i.e. pixel). The calculation needs good indirect irradiance estimates from sampling at a limited number of locations. We then rely on the irradiance interpolation algorithm to estimate the in-between, or missing, values. To generate a fairly smooth rendering for the sunlit space, accounting for the first level of inter-reflection, we would need to set moderately high resolution values for the ambient parameters.

TO APPROXIMATE THE EFFECT...

...of the higher-level reflections, we should set a value for the **-av** parameter. A rough guess for a daylit scene (internal) would be something in the range of 1/50 to 1/200 of the ground ambient value (obtained by executing the **gensky** command).

You may decide that **-ab 1** is insufficient to model the major light transfers, and that **-ab 2** is needed. In fact, this is almost certainly the case, because by using only one ambient bounce, we fail to account for the externally reflected component of sky light.

THE `mkillum` APPROACH

We can somewhat reduce the element of chance in our calculations for important light transfers by treating the window opening in a special way.

The *Radiance* system allows you to select known sources of light (windows, skylights, and so on) and pre-compute light output distributions for them.

They are then moved from the indirect (stochastic) calculation to the direct (deterministic) calculation.

The program we use for this task is called **`mkillum`**.

TO ILLUSTRATE THE EFFECTIVENESS OF...

...the **`mkillum`** approach, consider HS spawned at the rear wall of the room and also at the window plane. At the rear wall, the window subtends a solid angle that accounts for about 5% of the hemispherical “view” normal to the wall surface.

Therefore, only about 5 in every 100 rays spawned from this point will directly sample the luminous environment through the window - even though we know the window to be the only “source” of illumination.

THE SAME SAMPLING STRATEGY AT...

...the window plane, however, will cause about half the rays to sample the sky and the remainder to sample the ground. This is how `mkillum` works; you direct the program to determine a light output distribution for the window based on the sampling of incident radiation and the glazing transmission properties. In any subsequent calculation or rendering, the glazing elements are treated as "secondary light sources." Note that `mkillum` can account only for the diffuse component of light that passes through the glazing; the direct and specular components are unaffected.

USING `mkillum`

`mkillum` parameters can be specified in the scene description file, but on first encountering the technique, you may prefer to control all aspects of the calculation from the command line.

In this case, you must keep the window description, materials, and surfaces in a separate file.

To create a scene octree with the modified window description usually requires three stages.

THESE ARE...

(1) Prepare scene octree in the normal way.

(2) Use `mkillum` to compute the light output distribution of named glazing elements, usually one or more polygons. On completion, the program will have created new window description(s) using a special light source material called `illum`. There will be data files, one for each illum surface, that contain the material's light output distribution.

(3) Recreate the scene octree, replacing the original window description with the modified light source window.

THE COMMANDS MIGHT BE AS FOLLOWS:

Create octree as usual:

```
% oconv room.rad window.rad sky.rad out.rad >
scene.oct
```

Execute `mkillum` process:

```
% mkillum [rtrace options] scene.oct <
window.rad > mkiwin.rad
```

Re-create scene octree replacing the normal glazing window with the secondary source window:

```
% oconv room.rad mkiwin.rad sky.rad out.rad >
mkiscene.oct
```

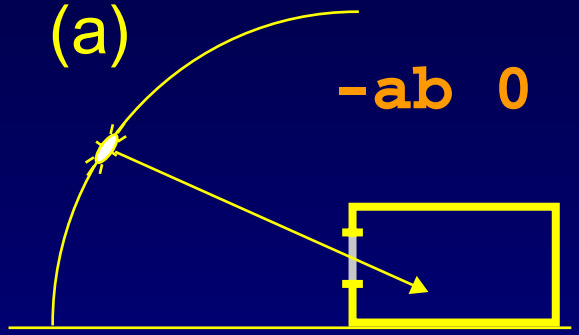
WHAT `rtrace` SETTINGS YOU USE...

...will depend on which light transfers you think need to be modelled, and on the complexity of the external scene. A series of simplified ray diagrams shows what `ab` settings will account for these components of diffuse radiation incident at the window:

- The diffuse component of light from the `glow` sky (b)
- The diffuse component of the first-order reflection of light from outside surfaces, for example the ground plane (c)
- The diffuse component of the first-order reflection of sky radiation from outside surfaces, for example the ground plane (d)

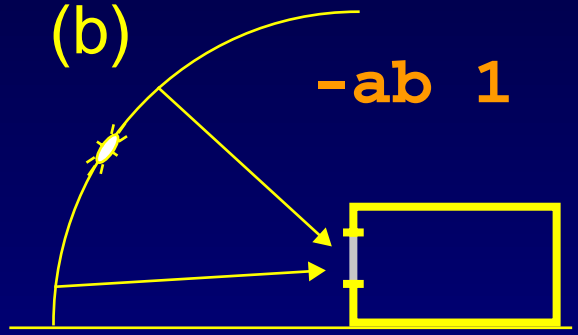
Components Of Illumination Accounted For By `mkillum`

(a)



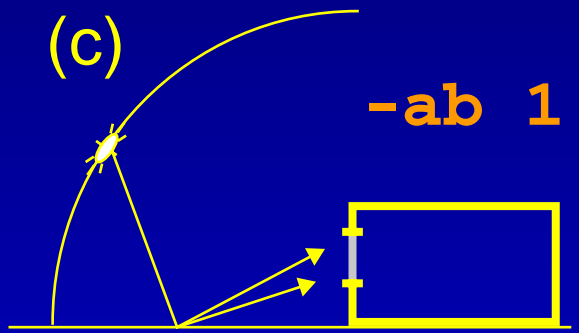
Direct only, `mkillum` blocks diffuse

(b)



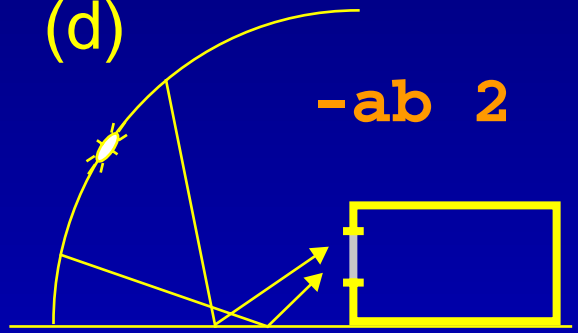
The diffuse component of light from the sky

(c)



Diffuse component of reflected sun

(d)



Diffuse component of reflected sky

`mkillum` - SUMMARY

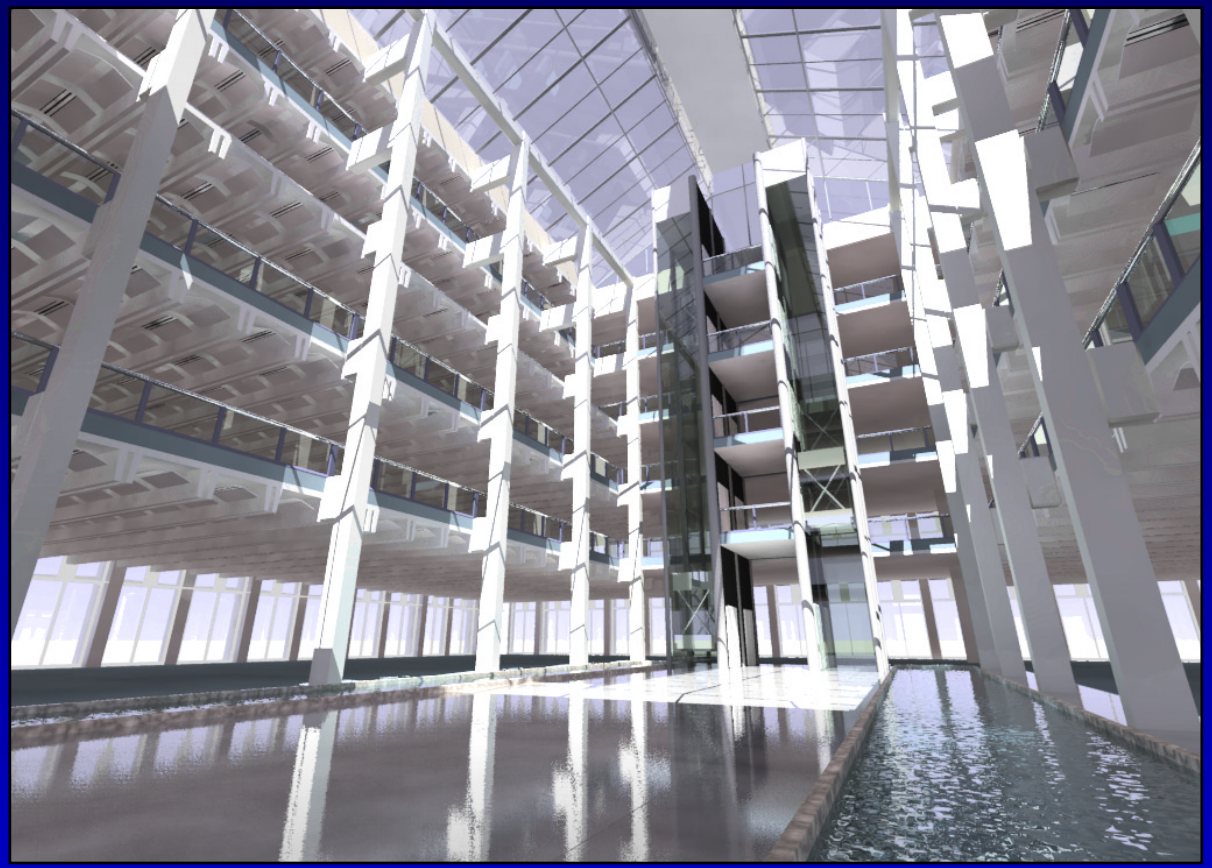
For the majority of scenes, setting `-ab 2` is usually sufficient to account for most of the diffuse light transfer paths to a window.

Ordinary *Radiance* light sources are opaque; if this were the case with the `illum` window, we would not be able to see through it. To avoid this, the `illum` sources have a dual nature. When treated in the direct component calculation, they behave like ordinary light sources, but when viewed directly, they revert to the original material description.

Examine the “Drafting Office” example in the `obj/virtual` directory of the standard *Radiance* release.

Visualizing A Highly Detailed Atrium Scene

A *Radiance* image of the atrium shows the degree of complexity that was achieved for this model.



THE ENTIRE *RADIANCE* SCENE...

...description was created from the command line.

This task was not as horrendous as it might first appear. Once a basic office module had been worked up, it was easy to generate much of the structure using the repeated-transformation option in **`xform`**.

In fact, the scene description consists of hierarchies of repeated transforms at various scales - for example, ceiling lights, a single office module, a row of office modules, and so on.

Ambient Calculation Parameter Values

First, we need to decide what light transfers are needed to produce the major illumination components for the rendering we have in mind. This will depend to some degree on how we choose to illuminate the model, and on the view parameters. For open scenes, it is invariably the case that some direct solar illumination greatly enhances the impact of the rendering.

From what we know of the model geometry and orientation, we can decide on a viewpoint and make a good guess at the solar altitude and azimuth positions.

THIS ATRIUM HAS NUMEROUS...

...facade windows and many roof glazing elements. With so many potential sources of light, it would be very inefficient to calculate their contribution in the deterministic domain.

Preprocessing of glazing elements to secondary light sources is therefore not advised for this type of building.

Consequently, we will rely exclusively on the ambient calculation to model the inter-reflection.

MAKE INFORMED CHOICES FOR...

...ambient parameter values *before you begin any batch rendering*.

Trial and error can be an instructive process. However, when, as here, the number of possibilities is nearly infinite, we need to drastically reduce the options before we do any exploring.

We need good starting values for **-ab** (ambient bounces), **-av** (ambient value), **-aw** (ambient weight), **-ad** (ambient divisions), **-as** (ambient super samples), **-aa** (ambient accuracy) and **-ar** (ambient resolution).

SETTING `-ab`

We could go for a low-cost rendering and set `-ab 0`, but the final result, we know, would not be very convincing. At one ambient bounce, the sky and sun patch become potential sources of indirect illumination. At two ambient bounces, we have the potential to calculate indirect illumination for surfaces that have no direct line of sight to either sky or sun patch. This should be sufficient to give most of the surfaces that we can see a calculated diffuse irradiance. We approximate the effect of subsequent ambient bounces with a constant ambient value.

SETTING -av AND -aw

The constant ambient value option serves two functions.

The first is to participate in the inter-reflection calculation, where it approximates the contribution of the higher-order reflections.

The other function is as sole provider of indirect illumination to surfaces excluded from the ambient calculation. More on this later.

SETTING -av AND -aw (CONT.)

For the majority of scenes it is likely that you will need to base the estimate on calculated values. Here, we demonstrate how `rview` can be used to make a reasonable estimate for a constant ambient value. Where in the scene should we determine this value? The average radiance in the middle of the office floor at level 2 will be very different from that at the top of the elevator shaft. We decide by anticipating where in the scene the ambient calculation will expend the greatest effort. This is most likely to be for the office ceilings, many of which are visible from our viewpoint.

SETTING -av AND -aw (CONT.)

Consequently, a "good" ambient value for the office spaces is what we should determine. This can be achieved in the following way:

(1) Start the previewer `rview` with the irradiance option (`-i`) enabled, `-ad 1`, and maybe `-ad` set to higher than the default.

(2) Wait a while for some detail to appear, then select a region in shade to refine (`frame` option). In this case, a bit of the ceiling at level 2 would be suitable.

(3) After some further refinement, pick out and display the irradiance evaluated at a surface on the ceiling (use the `trace` option). We call this value *I*.

(4) Recall that a uniform radiance that produces an irradiance, *I*, is simply I/π.

SETTING -av AND -aw (CONT.)

Try this value ($1/\pi$) with the ambient bounces set to zero. Does it give similar indirect illumination for the same surface? If yes, this is the value to use.

The ambient weight parameter **-aw**, if enabled (i.e., non-zero), will modify the default ambient value in a moving average as new indirect irradiances are computed. This may produce more accurate renderings for scenes where the luminance extremes, and therefore the indirect contributions, are not too great. However, this is rarely the case for renderings with daylight, and it is usually safest to disable this option, setting **-aw 0**.

SETTING -ad AND -as

The sun patches on the floor and structure of the atrium will be significant sources of indirect illumination.

To capture these potential sources, we should use a relatively large number of ambient divisions, in this case **-ad 1024**.

Ambient supersampling (**-as**) should therefore be set to about one half or one quarter of this value.

SETTING -aa AND -ar

Our view of the atrium will reveal an enormous amount of fine-scale detail, e.g. ceiling lights and acoustic baffles. None of these objects are seen really close up, but we still want to calculate values for them to see in the shading the local illumination effect of the sun patch.

Exact shading for each and every surface, however, is not really necessary; moderate irradiance interpolation errors over the scale size of a ceiling fixture should not be too conspicuous in the final image. Thus, a moderately accurate value should suffice. For this rendering, `-aa 0.3` was used.

SETTING -aa AND -ar (CONT.)

Having settled on a value for **-aa**, we can base the ambient resolution on a minimum separation for indirect irradiance values in the cache. In other words, for distances less than this minimum, the calculation will always resort to interpolation, rather than initiate more sampling, regardless of the error estimate associated with that interpolation. This prevents the calculation from expending massive effort resolving irradiance gradients over negligible scales. Strictly speaking, this distance gives the scale at which the interpolation accuracy begins to deteriorate from the **-aa** setting.

SETTING -aa AND -ar (CONT.)

How do we decide on a magnitude for this scale? It often helps to evaluate this scale for a range of **-ar** and then to choose the value that gives the best compromise between speed and accuracy. The scene dimension, D_{max}, is found from the scene octree using the **-d** option of **getinfo**. For this atrium, it was 99.2 meters. The minimum separation for cached irradiances, S_{min}, is:

$$S_{min} = \frac{D_{max} \times \text{-aa}}{\text{-ar}}$$

SETTING -aa AND -ar (CONT.)

For `-aa 0.3`, the S_{min} for a range of `-ar` are:

S_{min} [m]	-ar	Relative cost
0.93	32	1
0.47	64	4
0.23	128	16
0.12	256	64

The third column gives the approximate relative cost of the calculation based on a minimum ambient resolution of 32. From these values, we can make a reasonably informed choice for `-ar` and anticipate the trade-off between accuracy and speed.

SETTING -aa AND -ar (CONT.)

For the minimum -ar listed, the potential exists for poor irradiance interpolation over scales of about 1 meter.

These could be quite conspicuous from our view point, whereas (potentially) inaccurate shading over scales smaller than about 0.25 meter is far less likely to impair image quality.

Higher resolution is of course possible, but at some cost. With this in mind, an ambient resolution of 128 seems a reasonable compromise.

AMBIENT EXCLUDE/INCLUDE OPTIONS

Excluded materials will use the ambient value approximation directly, rather than a calculated indirect irradiance. We can make significant savings in rendering time by applying this option. Exclusion criteria could be any of the following:

Surfaces not visible from our view point (and unimportant in terms of light transfer).

High-detail areas (the -ar parameter may already impose a partial restriction here).

Surfaces that have a small diffuse reflectance (<5%).

Surfaces that will appear very small in the final image.

"Sticks" - surfaces that will appear as thin lines in the final image.

AMBIENT EXCLUDE/INCLUDE (CONT.)

Some of the surfaces of the atrium model that did not participate in the rendering ambient calculation (and the reasons for their exclusion) were

External facade detail including light shelf surfaces (not visible)

Window frames (sticks)

Windowsills (small)

Atrium roof vent slats (detail and small)

Atrium roof glazing bars (sticks)

Black handrail supports (low diffuse reflectance)

Best to segregate the materials into include and exclude types when you first construct the scene. In CAD terms, build the model layer by layer, with these requirements in mind.

Ambient File Use And The "Overture" Calculation

For a daylight rendering, the lion's share of the computation is invariably taken up by the ambient calculation. It makes sense, therefore, to save the cached indirect irradiance values to a file so they can be reused for later renderings.

With a well-populated ambient file, it can be surprising how little time additional renderings take to complete, especially when there is significant overlap between views.

Ambient File Use (cont.)

You must always set the same combination of ambient parameters for every rendering that uses the ambient file. There is a special exception to this (see below). Also, the ambient exclude (or include) list must not change.

Interpolation accuracy can be improved if the "presentation" (i.e., large) image is rendered using an already partially populated ambient file. The creation of the initial ambient file is known as an "overture" calculation.

AMBIENT FILE USE (CONT.)

Having created the ambient file with the “overture” calculation, you can, with caution, relax *some* of the ambient parameters for the larger renderings. The parameter revisions could be one or both of the following:

- Reduce **-ad** and **-as** by about 50%
- Slightly increase **-aa** (i.e., by 0.05 or 0.10)

The other ambient parameter settings should not be changed.

If you do decide to change any of the **-ad**, **-as**, or **-aa** settings after the “overture” calculation, the modifications will not be reflected in the header of the ambient file.

Batch Rendering

Starting with the “overture” calculation, we generate a small image and save the ambient file. The “presentation” image we have in mind is a rendering at approximately the resolution of the monitor display: about 1000 pixels square. We rarely show images at the resolution at which they were rendered; alias artifacts always look unpleasant and greatly detract from the impression of realism.

The highest quality is achieved by creating the rendering at two or three times the eventual size, then scaling it down using the `pfilt` program.

BATCH RENDERING (CONT.)

For certain scenes with multiple ambient bounces, you may find that it is the “overture” calculation that takes the longest, and that subsequent renderings, regardless of size, are completed relatively quickly. In this case, don’t be too concerned if the “overture” calculation seems to be taking a long time to generate a small image.

Batch-render a series of images, say, overnight or over the weekend. Automate the rendering from shell scripts and keep track of the progress by setting the `-e` and `-t` options of `rpict`.

An Appraisal Of The Atrium Rendering

The viewpoint and lighting were chosen to create a striking impression rather than to show a typical view. The low view point was deliberately chosen to reveal specular reflections from the “terrazzo” floor and the nearby water feature. This effect is perhaps too exaggerated, and the floor itself has something of the appearance of calm water in a murky pool.

It is in fact the uniformity of the floor that is the problem, rather than the specular component. If the floor had been divided up into slabs or tiles, and these given slightly different material properties, the final result would be much more convincing. If each tile had a small random component applied to its surface normal, giving us a slightly uneven floor, the rendering would be better still.

CONCLUSION

The *Radiance* system can be used to predict illumination levels and visual appearance under daylight conditions for virtually any building design. In this chapter, we have looked at just some of the ways in which *Radiance* can be applied to solving daylight problems. We hope that daylight designers will find the techniques of value and use them to solve their own lighting problems.

More important, we hope that the majority will be inspired to take a closer look at the system and the possibilities it offers.