**John Mardaljevic**                                           *e-mail*  **jm@dmu.ac.uk**
Institute of Energy and Sustainable Development (IESD), De Montfort University, Leicester, UK

# GENAMBPOS: A SCRIPT TO GENERATE MARKERS WHERE AMBIENT SAMPLING OCCURRED

## Background

The ambient calculation is one of the keystone features of *Radiance*. Used carefully, it can impart a tremendous impression of realism to a synthetic image. Mastery of the technique is vital for exacting, quantitative work such as daylight factor prediction. The *Daylight Simulation* chapter of the *Rendering with Radiance* book (just re-released) gives an overview of the ambient calculation and shows the user how to select values for the various parameters.

Effective use of the ambient calculation requires some understanding of how it progresses. Chiefly, how the density of hemispherical sampling is related to both the scene geometry <u>and</u> the settings for the key ambient parameters. This is described at length in the book, and will not be repeated here. This Technical Note is a supplement the material on ambient sampling in the book. It shows how a script can be used to visualise the locations in the scene where ambient sampling took place. The script is called **genambpos**, i.e. ***gen*erate *amb*ient *pos*ition** markers. The script is used as follows:

1. Run simulation (e.g. rendering, illuminance calculation, etc.) as normal with the ambient file option enabled (**-af fname**).

2. Execute the **genambpos** script in the same directory as the saved ambient file. The script will write out a brief description of its function and a list of the colour coding used, e.g. red spheres for level 0 sampling. It then asks you for two pieces of information. The first is the name of the ambient file for which you want to generate markers (i.e. **fname**). The second is the radius of the marker spheres in whatever units you scene is constructed from. Just hit return if the default value of **0.05** is desired.

3. The script will then interrogate the ambient file using the **lookamb** program and generate a scene description of coloured spheres where hemispherical sampling took place. It will also echo the number of locations (i.e. generated spheres) for each of the levels of ambient sampling.

To view the spheres in the context of the original scene (e.g. **scene.rad**), enter the command:

```
% objview amb_pos.rad scene.rad
```

and use the **l** command in **rview** to load-up any previously created views.[1] Alternatively, recreate the scene octree adding **amb_pos.rad** to the list of geometry files. Do not to attempt to reuse the ambient file with the markers present - the calculation won't like it!

Some example output from the script follows.

---

1. You could also view the spheres on their own, i.e. **objview amb_pos.rad**.

## Example Scenario

The scenario used for these examples is a precariously positioned box intersecting with a horizontal plane, Figure 1. Illumination is provided by a CIE standard overcast sky (i.e. a `glow` source). There is no ground glow in this model and the view parameters are set so that the dark space above the plane is in actuality below the horizon, i.e. the sky is present but not seen. Because the only illumination is a `glow` source, we need at least **`-ab 1`** to be able to compute some luminance for the visible surfaces in this scene (the constant ambient value is set to zero). The arrangement of the box above the plane makes for some, potentially, subtle shading on the horizontal plane. The renderings A and B in Figure 1 were generated using slightly different ambient parameter settings. The parameters common to both renderings were:

```
-ad 2048 -as 128 -ab 1 -aa 0.15 -av 0 0 0
```

The **`-ar`** parameter was set to 4 and 64 for renderings A and B respectively. Examine closely the shading under the box in both renderings (zoom in on the PDF). Notice how rendering B (**`-ar 64`**) better achieves the shading pattern we might expect from this arrangement. Renderings A' and B' were created with the makers generated by **genambpos** in the scene description.[2] Now we clearly see what we previously suspected - the density of hemispherical sampling points was greater for B. Also evident is the way the density increases closer in to the 'crease' between the box and the plane. Whilst **`-ar 4`** prevents the ambient sampling from
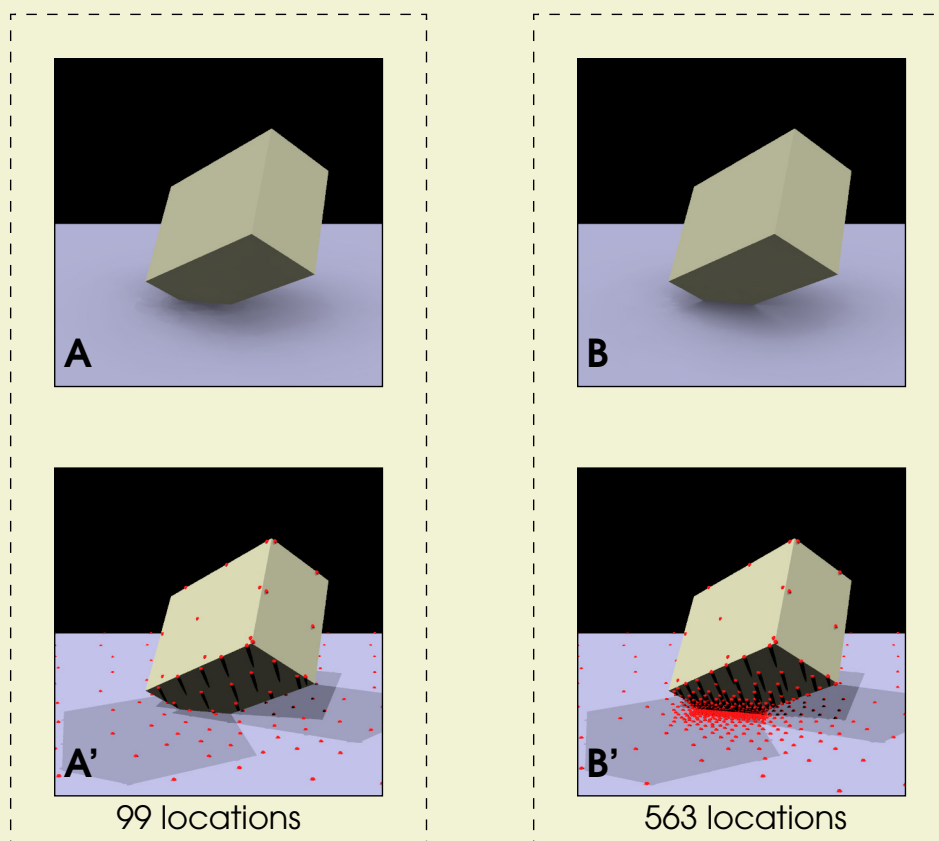


A        B

A'        B'

99 locations        563 locations

**Figure 1. Shading and sampling locations under a glow sky**

---

2.  Direct lighting from three sources was used to illuminate the scene with makers.

adequately resolving the irradiance gradient in A, it is probable that **-ar 64** is too high (i.e. sampling points too closely packed) and this could have been relaxed to 32 or even 16.

Repeating the above procedure with three ambient bounces (and **-ar 32**) produces the pattern of sampling shown in Figure 2. The explanation for some of the features in this pattern is probably evident. For example, one possible ray path across the sampling levels is drawn on the rendering. Other features are perhaps a little harder to explain - these are left to the reader.
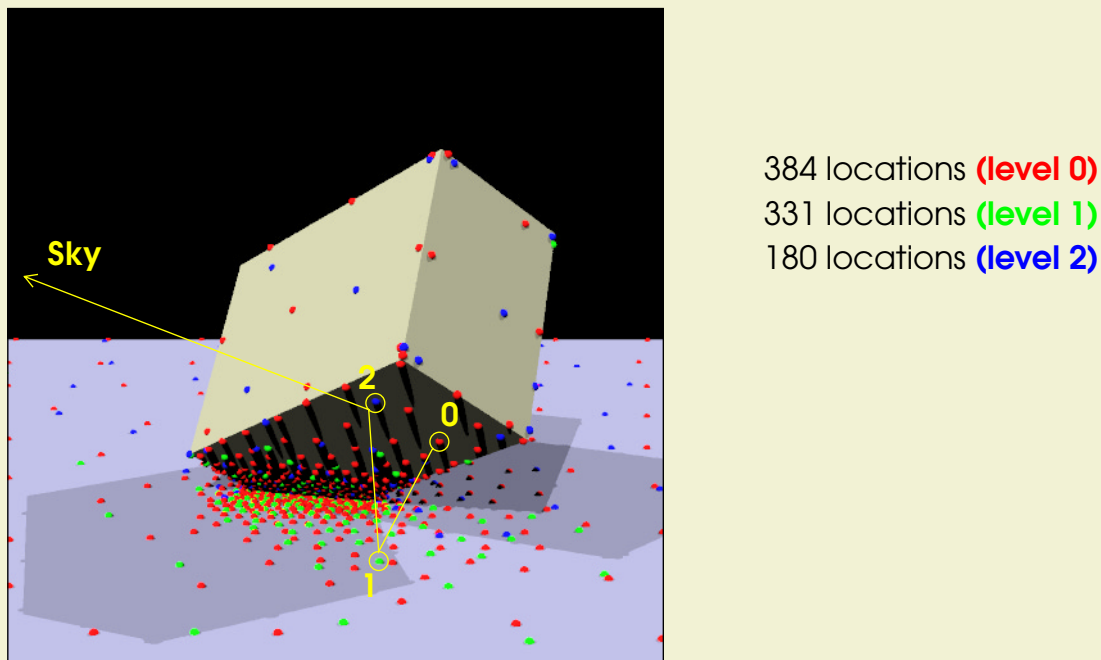


384 locations **(level 0)**
331 locations **(level 1)**
180 locations **(level 2)**

Figure 2. Sampling pattern with  **-ab  3**

Please bear in mind the following when using **genampos**:

- For thin walls etc. the marker spheres may look the same from the outside as they do on the inside; be careful not to misinterpret the patterns.
- The generated files (**amb_pos.rad** and **amb_pos_lev0** etc.) will need to be deleted manually when no longer needed.

## Summary

The **genampos** script is a utility to generate markers where ambient sampling took place. It may prove useful for anyone learning or teaching *Radiance*. Experienced users may find it handy as an occasional "ambient parameter debugging aid" for those particularly tricky lighting problems.

## Acknowledgements

Greg Ward assisted with the creation of the **genampos** script.

## Notes

The **genampos** script usually resides in **ray/src/util** as **genampos.csh** but is not installed by default. There is a bug in the current (3.5) and previous versions. Get the fixed version from the development source tree or the author's website.