



5th International Radiance Scientific Workshop

13-14 September 2006

IESD, De Montfort University, Leicester, UK

5th International *Radiance* Scientific Workshop

Organised by John Mardaljevic and Greg Ward

Hosted by the

Institute of Energy and Sustainable Development

De Montfort University, Leicester, UK

Programme Outline

	AM	PM
Wednesday	Rendering	Human factors, validation & materials
	Rendering	High Dynamic Range Imaging
Thursday	Rtcontrib: theory & application	Design case studies
	Design case studies	Radiance in “end-user packages” & Close

Workshop Dinner Menu

Appitisers

ONION BHAJI (v)

VEGETABLE PATTIS (v)

CHICKEN SHASHLIC

MASALA CHICKEN WINGS

HASINA LAMB

Main Course

CHICKEN TIKKA MASALA

LAMB ROGAN GOSH

KADAHI CHICKEN

MATTER PANEER (v)

SAAG ALOO (v)

PESAWARI NAN / PLAIN NAN (v)

PULLAU RICE (v)

Cost £20.50p (includes dessert + coffee)

Please sign-up if you wish to go

Radiance - a brief history, with some pictures

We'll skip the very early days - ask Greg about those. For many researchers and practitioners the promise was: **the ability to accurately predict daylight levels in buildings.**

Also, few (if any) restrictions on material type

And, the facility to model almost any light transfer e.g. diffuse-specular-specular-diffuse-cat-diffuse-specular etc. etc.

But you had to get your
hands dirty...



Instead of a “friendly” GUI

Users were
presented
with the UNIX
prompt:
jm 45%



NAME

rpict - generate a RADIANCE picture

SYNOPSIS

```
rpict [ options ] [ $EVAR ] [ @file ] [ octree ]
rpict [ options ] -defaults
```

DESCRIPTION

Rpict generates a picture from the RADIANCE scene given in octree and sends it to the standard output. If no octree is given, the standard input is read. (The octree may also be specified as the output of a command enclosed in quotes and preceded by a `!'.) Options specify the viewing parameters as well as giving some control over the calculation. Options may be given on the command line and/or read from the environment and/or read from a file. A command argument beginning with a dollar sign ('\$') is immediately replaced by the contents of the given environment variable. A command argument beginning with an at sign('@') is immediately replaced by the contents of the given file.

In the second form shown above, the default values for the options (modified by those options present) are printed with a brief explanation.

Most options are followed by one or more arguments, which must be separated from the option and each other by white space. The exceptions to this rule are the -vt option and the boolean options. Normally, the appearance of a boolean option causes a feature to be "toggled", that is switched from off to on or on to off depending on its previous state. Boolean options may also be set explicitly by following them immediately with a '+' or '-', meaning on or off, respectively. Synonyms for '+' are any of the characters "yYtT1", and synonyms for '-' are any of the characters "nNfF0". All other characters will generate an error.

-vtt Set view type to t. If t is 'v', a perspective view is selected. If t is 'l', a parallel view is used. A cylindrical panorama may be selected by setting t to the letter 'c'. This view is like a standard perspective vertically, but projected on a cylinder horizontally (like a soupcan's-eye view). Two fisheye views are provided as well; 'h' yields a hemispherical fisheye view and 'a' results in angular fisheye distortion. A hemispherical fisheye is a projection of the hemisphere onto a circle. The maximum view angle for this type is 180 degrees. An angular fisheye view is defined such that distance from the center of the image is proportional to the angle from the central view direction. An angular fisheye can display a full 360 degrees. Note that there is no space between the view type option and its single letter argument.

-vp x y z Set the view point to x y z . This is the focal point of a perspective view or the center of a parallel projection.

-vd xd yd zd Set the view direction vector to xd yd zd . The length of this vector indicates the focal distance as needed by the -pd option, described below.

-vu xd yd zd Set the view up vector (vertical direction) to xd yd zd .

-vh val Set the view horizontal size to val. For a perspective projection (including fisheye views), val is the horizontal field of view (in degrees). For a parallel projection, val is the view width in world coordinates.

-vv val Set the view vertical size to val.

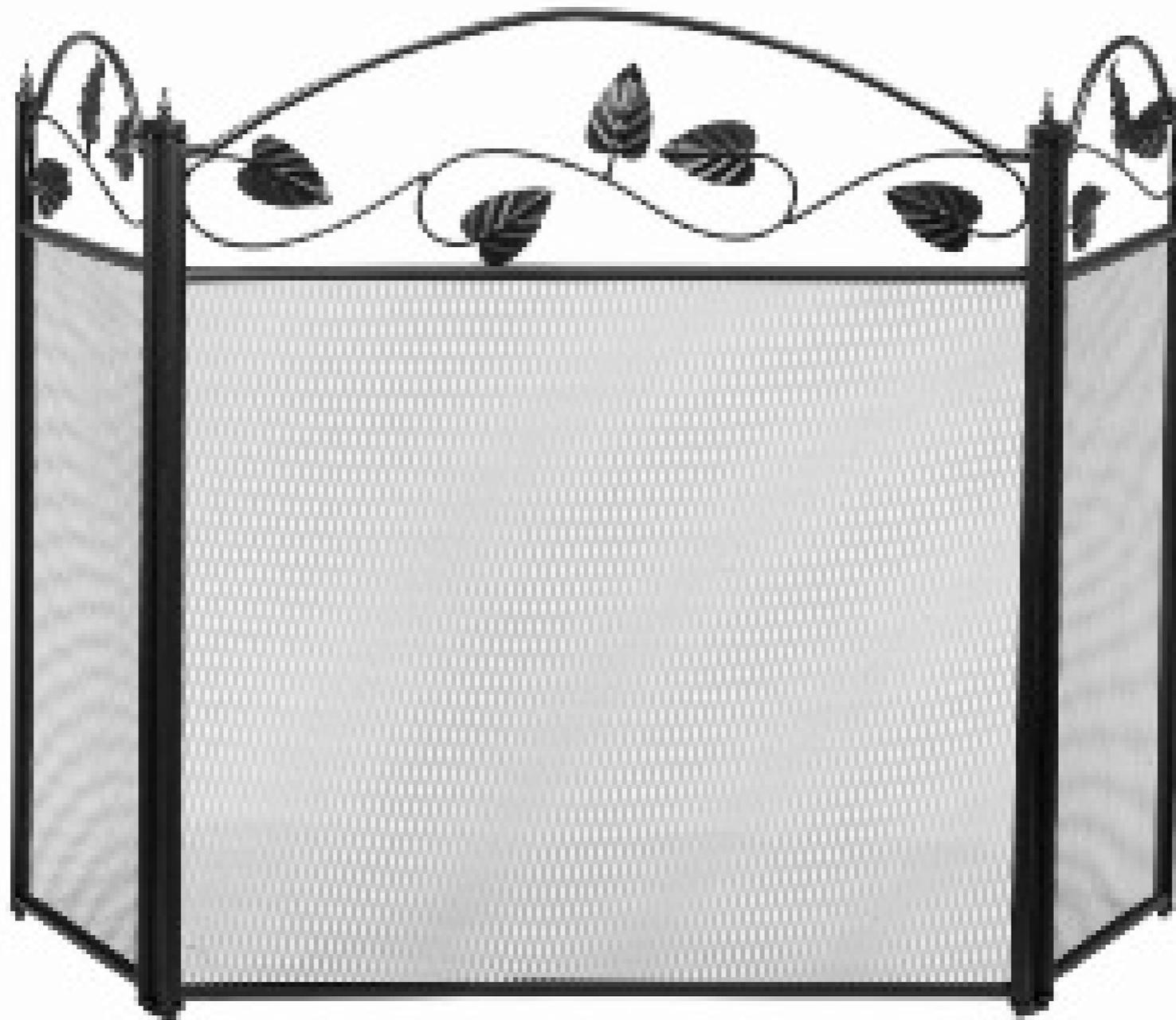
-vo val Set the view fore clipping plane at a distance of val from the view point. The plane will be perpendicular to the view

But that was
OK because
Greg had
provided
UNIX Man
pages

The Man pages were great because they explained how *Radiance* worked.

Unfortunately, the Man pages made sense only if you already knew pretty well how *Radiance* worked and just needed a gentle reminder to do with this or that parameter.

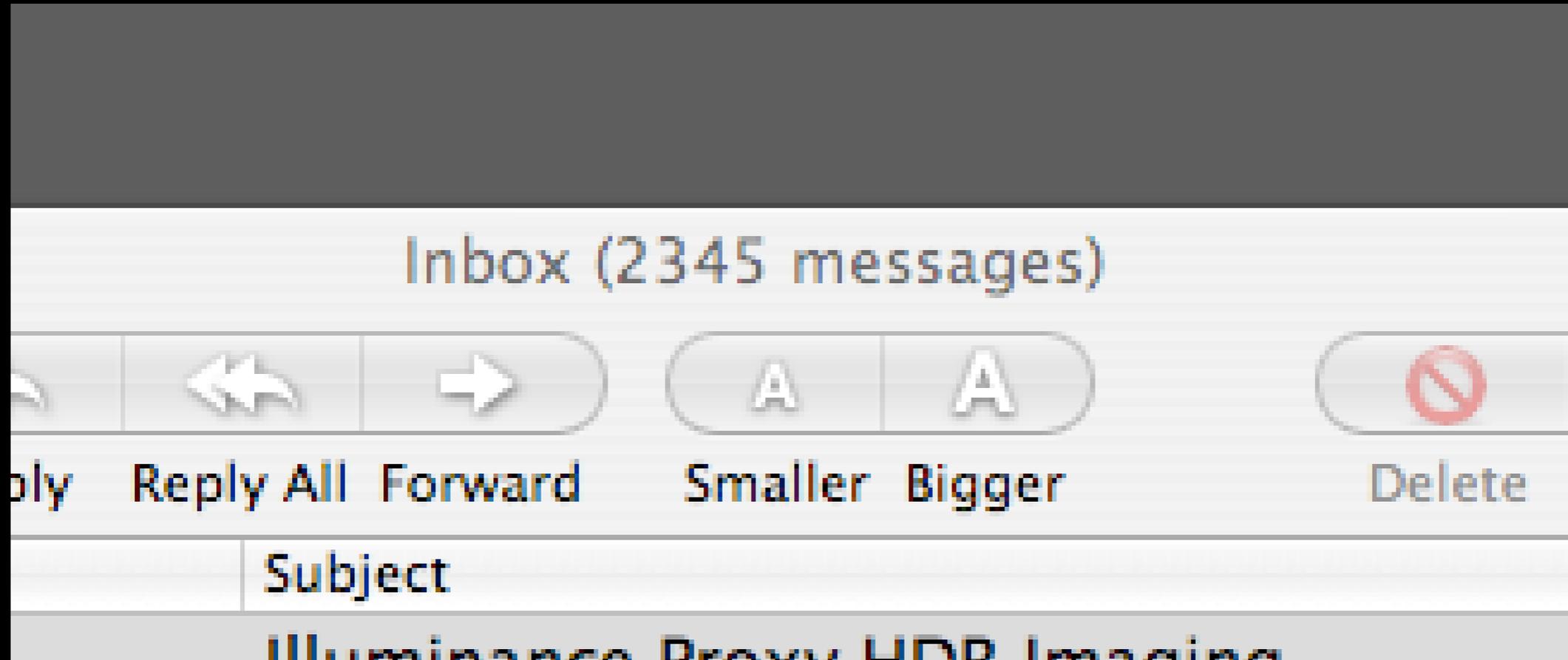
**So, for many of us, at least at first,
the Man pages were about as useful as
a . . .**



...made of

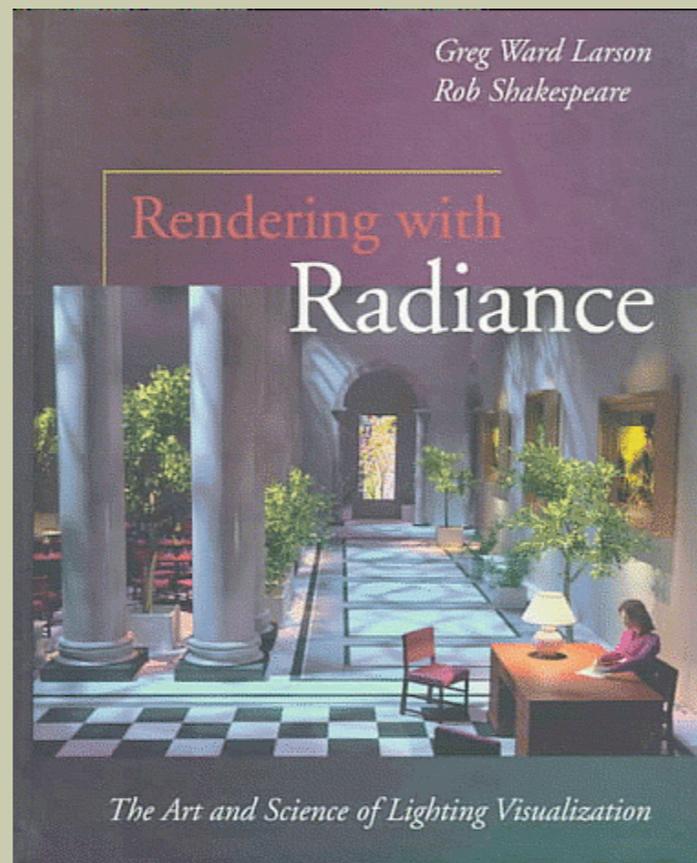


So Greg received lots of e-mails



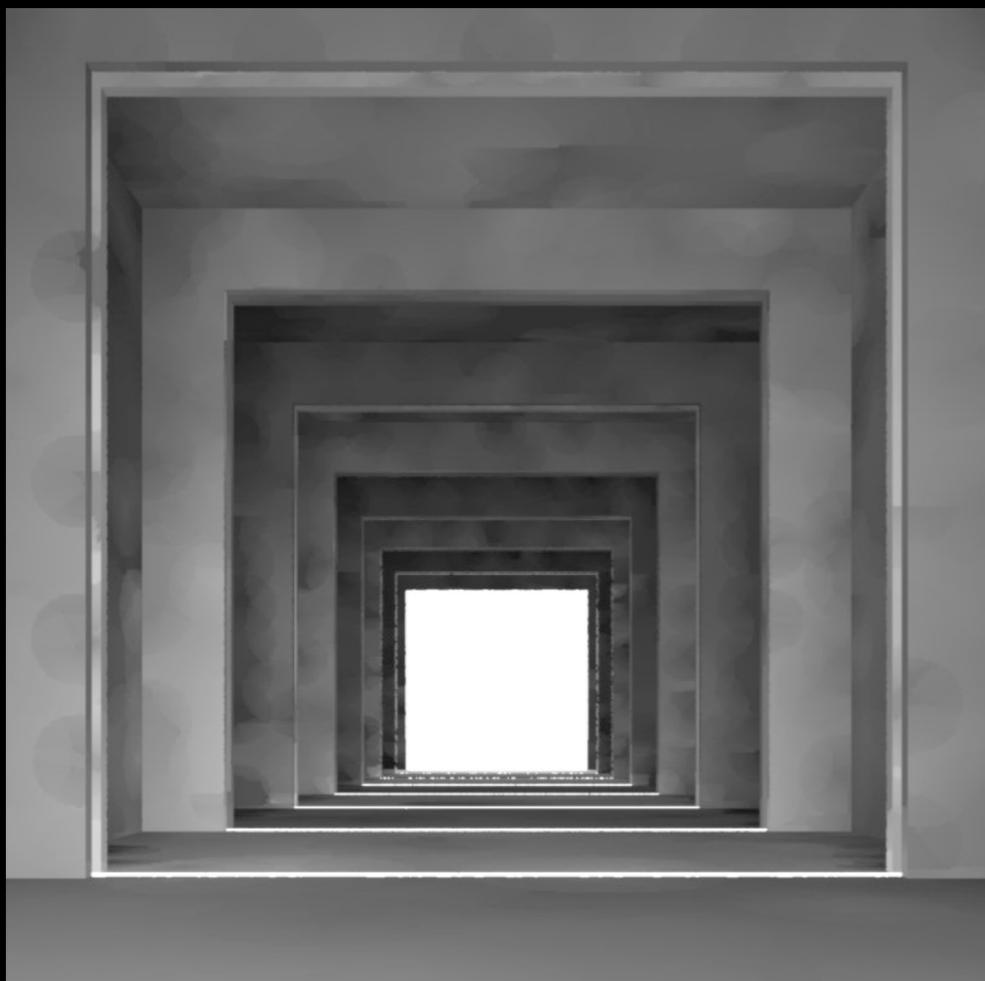
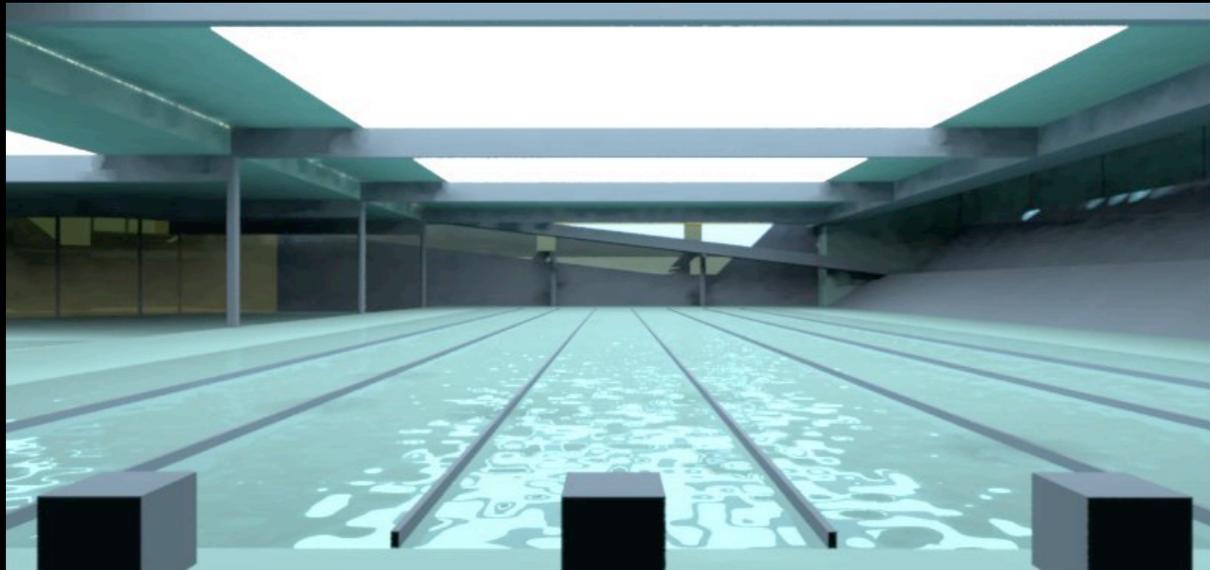
And somehow managed to respond to most (all?) of them

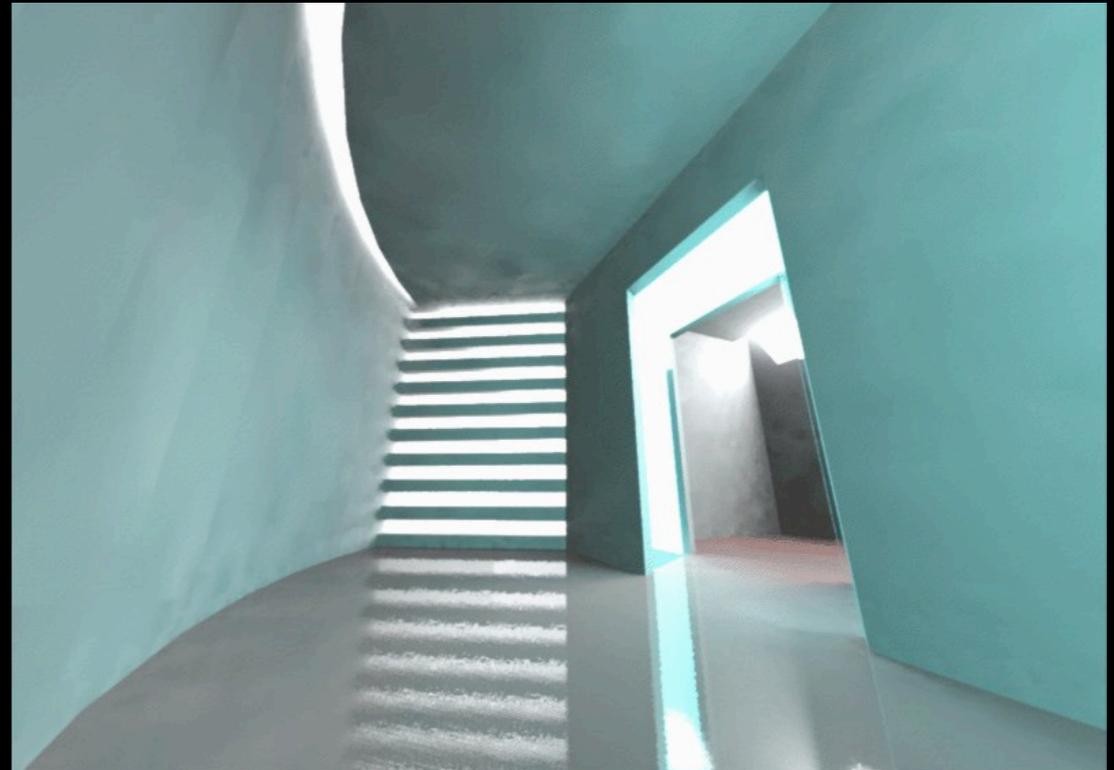
Fed up with answering the same question for the umpteenth time, Greg thought that a book on the *Radiance* system would help (at least to reduce the size of his inbox).



The promise of huge royalties was sufficient to attract a slew of co-authors

The *Radiance*-online website
<http://www.radiance-online.org/>
is another major resource that has been of
enormous help to users - new and
experienced alike.





And so lots of folk busied themselves with the (mostly) pleasurable task of predicting a veritable universe of lighting phenomena using the *Radiance* lighting simulation system.

They're still doing so today -
Ladies & Gentlemen welcome to the
**5th International *Radiance* Scientific
Workshop**