Integrated Lighting System
Design And Visualisation.

"The simulation and presentation of lighting effects."

RENDERING REALITY

VOLUME ONE

SIMON CRONE

ARCHITECTURAL DISSERTATION

NOVEMBER 1992

Iɴᴛᴇɢʀᴀᴛᴇᴅ Lɪɢʜᴛɪɴɢ Sʏsᴛᴇᴍ

Dᴇsɪɢɴ ᴀɴᴅ Vɪsᴜᴀʟɪsᴀᴛɪᴏɴ.

The simulation and presentation of lighting effects.

Volume One

RENDERING REALITY

Simon Michael Dalrymple Crone.

Tutor   Neville D'Cruz
Co Tutor   Terry McMinn

Preface.

The idea of totally perceiving and visualising three dimensional design has always intrigued me. To be able to know exactly how the finished design will look and feel requires a unique insight to which all designers aspire. It is this insight that enables designers to exhibit control in the evolution of their designs.

It is one thing to have this gift of insight but without the ability to communicate the vision to others the final goal cannot be reached. Design is about ideas and communicating those ideas.

One increasingly accessible application available to designers is the use of the computer as a visualisation and communication tool. It is here that my personal interest lies. The computer is both a design tool for the designer to evaluate designs and as a communication device for showing the design.

My quest for total visualisation has led me through the experience of three dimensional modelling systems and into realistic rendering. The current rendering system platform in the Curtin Architecture department has provided, through the CAN Raytrace program and EAGLE 3D modeler, an impressive integrated design tool. However it does not incorporate realistic lighting values based on physical models.

So my search began...

Simon Crone.

Acknowledgments

I wish to thank the following for providing all their expertise and support;

Neville D'Cruz   (Senior Tutor)

Terry McMinn   (Co Tutor)

Andrew Marriott   (Co Tutor)

Greg Ward   (RADIANCE Author, LBL)

and all my luddite friends and family.

Contents

1.   Introduction.

> " A picture is an intermediate something between a
>   thought and a thing. "

> - Samuel Taylor Coleridge

The design of artificial lighting is currently a very specialised field. Artificial lighting designers rely heavily on their own experience and on what has previously worked well in similar situations. There is currently no real way to accurately visualise the complexities of their design without full scale testing.

The aesthetic quality of a lightning system depends primarily upon the experience of the designer. The large diversity in types of available artificial lighting and their respective quality and colour, creates problems when trying out new mixes of specific lighting. Large numbers of possible configuration go untried purely because they cannot be accurately visually assessed. The risk of designing untried lighting systems means that often most lighting systems are under utilised.

This dissertation explores a possible solution to this dilemma by utilising a photo realistic image rendering system. The system used, called **RADIANCE**, is a computer software package that is able to create a photo realistic image of a physical environment by accurately calculating and predicting the visible radiation in the space.

By using accurate input into the program, such as manufacturers photometric data for specific lighting fixtures, designers are able to evaluate innovative, experimental lighting designs, lowering the risk of failure. The program accounts for both specular and diffuse interreflections thus allowing both the designer and client to perceive a genuine view of a finished space.

1.1. Overview

This dissertation has been conducted in three stages. They deal with;

a)   An investigation into the way in which the RADIANCE package works.

b)   Testing the package on two case studies - a gallery and a foyer - and comparing the images generated with other methods of realism.

c)   The production of a design platform (interface) with which students can model a physical space using the EAGLE 3D modelling system and begin to accurately evaluate different lighting designs within the model.

2. Background to realistic rendering

2.1. Representation or Simulation of reality?

When it comes to representing reality in photo-realistic images there are two avenues of approach.

1) Illusion based rendering.

2) Physical based rendering.

Although the end results often look similar the process that create them are vastly different.

2.1.1. Illusion based rendering

Illusion based rendering is the process of developing elements of reality only to the degree necessary to convey an idea. It places total emphasis on the final image itself, rather than the techniques used to produce it. There is no attempt, for example, to correlate any physical values with the model being rendered. Often materials or lights sources are given physically impossible values. For example a light source may be made invisible and placed in a corner that previously appeared too dark. Ambient light that originates from nowhere, often plays a large role in many illusion based images. The advantages of this technique is that results may be obtained very quickly and easily without fully describing the exact physical characteristics of material. Also it is possible to render images that are beyond the realm of physical techniques.

2.1.2. Physical based rendering

Physical based rendering however, involves the strict adherence to rules of physics and geometry. It traces as closely as possible the physical behaviour of light over or through physical materials to generate an image. The

closer the simulation is to the real world physical model, the less flexible the processed image becomes. The emphasis is on numeric simulation not on an artistic impression. This technique has the advantage of producing images that are able to be analysed. Useful meaningful data can be extracted (not only presented as a picture,) and used to evaluate the performance of a specific scene.

This rigorous simulation, however raises a question, as put by Hall (1989, pp 6);

"How do we tolerate the need for effect - the need to create imagery that is outside the visual reality of the physical world in order to convey a message?"

He does not answer this question in his book but goes on to say that

" Successful use of computer graphics as a communication tool comes from understanding the process of modeling reality and extracting from it elements that best convey the essence of an idea".

2.2. The Illumination Process

The illumination process that is used to create a realistic image of a physical environment depends largely upon the rendering technique employed. However, there are two illumination paradigms that must be dealt with when dealing with any rendering technique as described in Hall (1989, pp11). The first is how the boundary between materials interact with light. The second is how light is scattered and absorbed as it passes through a material.

2.2.1. Boundaries.

All materials have boundaries with other materials. It is these boundaries which are the surfaces of an environment. Generally boundaries may lie between two dielectric (

transparent) materials, a dielectric and an opaque material or two opaque materials (although this is never seen). All light that strikes a boundary is either reflected and or transmitted. Light may be absorbed after it passes through a boundary.

Smooth boundaries scatter very little reflected and transmitted light, while rough boundaries scatter light in all directions. Most light is transmitted at boundaries between dielectic materials, while boundaries with opaque materials reflects most light.

2.2.2. Reflections

There are two types of reflection commonly used in rendering techniques. They are;

1)   Lambertian diffuse reflection

2)   Specular reflection

2.2.2.1.   Lambertian Diffuse Reflection

Lambertian diffuse reflection is the simplest type of reflection where a ray of light, after a small amount of absorption, is scattered back into the environment with equal intensity in equal directions. The amount of energy reflected per unit area is proportional to the cosine of the angle between the normal to the surface at that point, and the direction to the light source, as described by Rogers (1991, pp 60).

Lambertian reflection can be written as;

$$I_d = I_i K_d \cos\theta$$

Where I_i is the intensity of the light source;

K_d is a constant of reflection dependant on the surface material;

θ is the angle of reflection between the incident light direction and the surface normal.

Lambertian diffuse surfaces appear dull and do not produce regular reflection. It is important to note (for radiosity) that the intensity the viewer sees reflected of a diffuse surface is independent of the viewer position, while energy is not.

2.2.2.2.   Specular Reflection

Specular reflection is due to the shininess of a surface. Unlike diffuse reflections it is highly dependant upon the positioning of the viewer and the light source at each point on the specular surface. The simplest form of specular reflection is the *mirror specular.*

Mirror Specular

This reflection model produces perfect mirror reflections. ie. The angle of reflection, θ_r, of a light ray on a planar surface is equal to the angle of the incidence, θ_i.   (Hall, 1989 pp 20)

$$\theta_r = \theta_i$$

More complex specular reflection occurs when a surface is not mirror like but still shiny such that the specular reflection is scattered about the mirror direction This type of specular model is known as the Phong model as discussed by Rogers (1991, pp 62).

The Phong Model

In this model the intensity of the reflection is proportional to the cosine (raised to some power) of the angle between the mirror direction and viewer direction.

$$I_s = I_i K_s (\cos\theta)^n$$

where I_i is the intensity of the light source;

K_s is a constant representing the fraction of energy reflected specularily, which is a function of the surface properties;

θ is the angle between the mirror direction and the viewer direction.

n is the specular exponent, also a function of the surface properties.

Small values of n result in a not very shiny surface while large values result in shiny surfaces. A value of n = 0 will result in a Lambertian diffuse reflectance and a value of n = *infinity* will result in a mirrored surface.

2.2.3. Transmission

In the case of transparent or semi-transparent materials, light rays may be transmitted through the surface. The transmission may be specular such as in normal glass, or diffuse such as frosted glass, but rarely a mixture. The angle of refraction of a transmitted ray through a boundary is given by Snell's Law as described in the IES Lighting Handbook (1981).

$$n_t \sin \theta_t = n_i \sin \theta_i$$

where n_t is the index of   refraction of the   first medium.

θ_t is the angle between the incident light ray and the surface normal

n_t is the index of refraction of the second medium.

θ_t is the angle between the refracted light ray and the surface normal

3.   Raytracing

In the real world light rays (photons) are emitted from a light source and are reflected and transmitted around an environment until they are finally absorbed. Those rays that enter the eyes will produce a picture on the retina. However only a minute fraction of all the rays emitted at a particular time ever enter the eyes.

If this rendering model was chosen to produce a photo-realistic picture it would take an incredibly long time for any sort of image to be created. If the rays were traced backwards from the eye into the environment until they hit a light source then there would still be an infinite amount of ray paths to trace (as a large majority of them would not hit a source).

Thus the only rays that are of any interest are the ones that emanate from the eye and end up hitting a light source. Thus the basic raytracing technique involves tracing a ray backwards from the eye to a light source and arbitrarily sending a shadow ray towards the light at every intersection between the ray and an object. This process stops when either the ray hits a diffuse surface or a specific number of "bounces" has occurred. Since the rays stop at diffuse surfaces, this causes a quantity of diffusely interreflected light to be missed. This is usually accounted for by the addition of ambient light as discussed by Rodgers (1991, pp 65).

The department of Architecture at Curtin University of Technology is currently using the Computer Animation Negus (CAN) Raytrace system. A full description can be found in "Image generation using the Computer Animation Negus Raytracing System" by A. Marriott and P. Dench (1990).

4.   Radiosity

Radiosity is a technique for accurately modeling area light sources, diffuse reflections, colour bleeding between surfaces, penumbrae along shadow boundaries and detailed shading within shadows. Since raytracing is more effective at modeling point sources, specular reflections and refractions, it is usually desirable to apply both techniques to a physical environment to produce a photo-realistic image.

Ashdown (1992), gives a general account of the radiosity technique. He states that radiosity requires all surfaces to be ideally diffuse reflectors and that the scene to be rendered is a closed environment. That is a model in which every light ray emitted must eventually end up intersecting a surface. The main advantage of radiosity over raytracing is that the technique creates view independent solution. This solution is in the form of a precalculated surface illuminance for every part of a surface in the scene. To do this, ever surface is subdivided into small planar polygons called patches. These patches are then further subdivided into elements. The illuminance for these elements is estimated and then recursively calculated until it converges to a solution as described by Rogers (1991, pp 66 -70). This recursive nature of the radiosity technique allows a geometrically correct scene to be almost instantly displayed while the illuminance for each element is being refined. One drawback of this technique is that in order for so many elements to be calculated, large amounts of memory and CPU time will be used.

As mentioned above, once the illuminance values for the scene have been refined, any view of the scene may be created with very little effort, for any eye position. This has allowed for the development of real time interactive walk through of a realistic simulated environment.

5.   RADIANCE Description.

RADIANCE is a computer software package developed by the Lighting Systems Research group at Lawrence Berkeley Laboratory under the direction of Greg Ward (Ward, 1992). It is a research tool for accurately calculating and predicting the visible radiation in a space by using a combination of raytracing and radiosity techniques. The program uses three dimensional (3D) geometric models as input, to generate spectral radiance values in the form of photo realistic images. The package though is more than just a photo-realistic renderer.

By using accurate input into the program, such as manufacturers photometric data for specific lighting fixtures, designers are able to confidently evaluate their designs without the risk of being led astray by visually appealing yet totally inaccurate images. The RADIANCE software package is of most use when dealing with innovative, experimental lighting designs. The program can account for both specular and diffuse interreflections thus allowing both the designer and client a genuine view of a finished space.

There are three steps to producing such an image. The first involves creating or converting a three dimensional description of a physical environment or scene (ie an office interior; rooms, furniture lights etc) into simple geometric elements that can be interpreted by the RADIANCE package. Such elements include polygons, spheres, cylinders and cones. These must then be assigned a specific material or property, for example metal, glass, wood, marble etc. This second step also includes the setting up of specific light sources, their strength, type and distribution if necessary. The final step is to render the scene to produce an image. This image may then be "cleaned", "analysed", and "filtered" in a variety of ways depending on the required application. This process is of course an interactive one. The designer can easily go back and change the geometry or material specifications until the required design has been reached. For a complete description of the RADIANCE package refer to the second volume (Users Manual) of this dissertation.

RADIANCE is a free raytracing software for lighting simulation and rendering. It is not public domain and no one may resell it under their own name.

The RADIANCE package can be picked up by an anonymous ftp from its original site at;   hobbes.lbl.gov (128.3.12.38)

6. RADIANCE Evaluation

In order to evaluate the RADIANCE package and come to terms with its capabilities, two case studies were conducted upon existing spaces. These spaces were the Underwood Gallery situated in the Curtin University Administration Building and the Zurich House foyer situated at 66 Kings Park Road, West Perth. These two very different scenes offered a variety of materials and lighting conditions with which to simulate.

6.1. Case study one -   Underwood Gallery.

This was the first attempt at using the RADIANCE package on an original scene. The gallery offered the opportunity of three different types of integrated lighting. That is, natural daylighting, general strip lighting and focussed spot lighting. The scene geometry was also relatively simple so that more time could be spent on rendering pictures than creating the geometric model.

6.1.1. Autocad converter

As there was, at the beginning of the project, no utility to convert EAGLE 3D geometry into RADIANCE 3D geometry the gallery was modeled on Autocad. This was because the RADIANCE package came supplied with a conversion program to create RADIANCE geometry from an Autocad dxf file.

This conversion program however did not work well. Adjustments were made to the source code to produce a working converter. This program however still has major problems. The greatest being that it tessellates 3D polyline faces into 5mm wide strips. Thus many of the walls of the gallery (which measure 10 by 5 meters) contained large number of faces. The east painted brick wall for example contained 2000 faces when only one was required.

This problem was overcome by manually editing the converted RADIANCE geometry files. This was a straight forward task as the geometry of the faces in question could be simply obtained from the autocad model. The fixing (if not completely rewriting) of this conversion program is something that would be worth while in the future.

6.1.2. Brick texture patterns

The second problem faced was that of the supplied brick texture pattern. It did not produce realistic brick patterning. The problem lay in the *brick.cal* function file. This procedural texture produced bricks with doubled up perp joints. The solution to this problem was to write my own brick texture function. The first of my procedural brick texture function produced a too detailed pattern. Each mortar joint had a highlight, a shadow and a mortar colour. This detail was lost when viewed from more than a few meters away. Version number two of the brick texture pattern only had a highlight and a shadow, which increased the distance from which it could be seen. For quite a while, however, the brick texture when mapped onto a wall appeared to have bright white spots no matter what the internal lighting conditions were. The origin of this problem ended up not being the brick texture patterning at all, as detailed below.

6.1.3. Daylighting.

The problem with the brick spots lay with the specification of daylight. RADIANCE sets up a large "source" that emits light depending upon the time of the year and location in the world. This source, which was being used to model daylight through the gallery's door, was illuminating the backs of all the internal walls. Subsequently the brick procedural texture map was letting in light through "holes" in the wall. By effectively throwing a "blanket" of faces over the gallery,

leaving the door uncovered, the spotty brick pattern disappeared.

RADIANCE has two distinct ways of modeling exterior windows, in respect to daylighting. Both ways require the *gensky* program that creates CIE sky distribution for a particular time and place. The first model tried on the foyer used a plain glass window and relied upon interreflection calculations to produce distributions from the sky into the room. The second model used the program *mkillum* to generate a set of distribution data that enables the window to be used as a secondary light source to illuminate the room. Of the two models the second proved to be more efficient and visually acceptable.

6.1.4. RADIANCE to EAGLE

In order to check that the geometry converted from autocad and subsequently modified by hand was indeed correct, an EAGLE macro was written converting RADIANCE geometries into a 3D EAGLE model. The macro reads all RADIANCE geometries and even expands RADIANCE object generation programs into its 3D components. This macro showed that the geometry in the foyer was correct and has been used since to convert other RADIANCE descriptions into EAGLE models.

6.1.5. Gallery Materials

A number of the materials that are used in the RADIANCE description were taken from a still video camera and scanned into the computer as images. These images were then cropped and colour adjusted and used as tiling patterns. All of the pictures hanging on the walls in the gallery were existing scanned images that were able to be mapped onto canvases placed around in the scene. The

ministrip ceiling was a tiled picture that was created with a paint box program.

The only draw back with using scanned pictures is that they must be transformed from the origin and the xy plane to their new position. The images also cannot be stretched and thus always remain with their original aspect ratio.

At present it is difficult to match colours. Unless the user is very experienced with R G B colour values, a lot of guess work is required to obtain respectable colours.

6.1.6. Gallery Lights

The gallery had a large general purpose strip lighting ring that had an upward facing surface normal when converted from the dxf file. This caused the lights to shine up, thus requiring the surfaces to be flipped.

The spot lights used the IES distribution type 06 (R-40 Flood) and as such did not end up being as focused as the real thing. The positioning and directing of the spotlights was done in EAGLE by converting the RADIANCE geometry into EAGLE geometry and rotating the lights to the required direction. An EAGLE macro then extracted the angles of rotation which were placed into the RADIANCE scene description files. Half way through the second case study it was discovered that the initial radiance values for the gallery lights were out by around 300 % and when this was changed more realistic images resulted.

6.1.7. Trees

In the gallery scene a number of small trees were modeled using the standard tree model provided by RADIANCE. These trees although looking very impressive drastically increased the amount of time taken to render the scene. One particular view where the trees were dominant took 25

hours to render a 600 x 900 scene. This would not appear to be a realistic option in scene rendering.

6.1.8. Conclusion

The rendering of the gallery was a success in terms of reaching the goal of producing a recongisable photo-realistic image of the gallery. As this was the first attempt to render an original scene with RADIANCE, the final image is not as complete as it might be. However it has raised interest in the RADIANCE package as a design tool for lighting galleries. With ever changing pieces of art work all requiring different lighting intensities and lamp colour types, the RADIANCE package seems an ideal platform with which to administer gallery lighting design. (See appendix B for colour copies of the actual galley and RADIANCE generated images.)

6.2. Case study two - Zurich House Foyer.

The second case study involved the testing of the EAGLE to RADIANCE conversion with a slightly more dynamic scene. This particular foyer had previously been Raytraced (three years earlier) and thus offered an interesting scene for comparison.

6.2.1. EAGLE Conversion

The EAGLE model of the foyer already existed but the design had slightly changed when the foyer was actually built. Thus there was a modification needed to the model to bring it up to date. The model was divided up into its separate materials parts which were then labelled as particular fragments. The model was then converted into RADIANCE description files as detailed in section 7.1.

6.2.2. IES lighting lamps

The foyer scene used a combination of IES standard lamp distributions. The first attempt at using the IES no. 10 standard luminaire produced too concentrated patches on

the floor, so this was changed to IES no 06. This demonstrates the ability to use RADIANCE as a design tool. It should be noted that the foyer is a night scene as no sunlight, sky distribution was modelled.

6.2.3. Interreflection

The completed foyer scene was used to investigate the impact of ambient light on the final image. The RADIANCE package can either use ambient light to compensate for loss of light due to diffuse interreflection or actually calculate the interreflection value itself. This however greatly effects the rendering time of the image. An image with no ambient light and no interreflection calculation appeared dark and shadows appeared black. There also appeared to be no colour bleeding between surfaces. The image with ambient light and no interreflection simply had lighter shadows. There was still no colour bleeding or variation within the shadows. With the interreflection calculation turned on, the ceiling showed colour from the surrounding walls and floor and the shadows were varied. The cost of this higher quality picture was a much longer rendering time.

6.2.4. Conclusion

The rendering of the foyer produced some very spectacular results. The images produced are very similar to the actual foyer. The times taken to render the images (up to 13 hours) are at the top end of acceptable times to wait for an image. However rendering times must be sacrificed if the highest possible quality images are required. (See appendix B for colour copies of the actual foyer and RADIANCE generated images.)

7. Design Platform

One of the aims of this dissertation was to create a platform with which designers (students especially) could begin to incorporate lighting design into their work. This platform has been established through the integration of the EAGLE 3D Modeling system and the RADIANCE rendering program. The platform consists of two parts. The first part, which is invisible to the designer, deals with converting an existing 3D geometric model from EAGLE into a geometric scene description that the RADIANCE program can interpret. The second part is the up front user interface where the designers are able to assign a number of different materials and lighting fixtures to their models and produce a photo-realistic image with which they can use to evaluate their design.

7.1. EAGLE to RADIANCE conversion

The EAGLE to RADIANCE conversion is based on the existing EAGLE to Raytrace system. This system takes an EAGLE model in ASCII format and converts its geometry into a MOG   (Dench, 1989) geometry file.

7.1.1. MOG Files.

This MOG file is an intermediate file in a standard geometry format that is used by a whole series of utilities to convert from one geometry format into another. The MOG utilities use the Unix "redirection" and "filter" facilities to string together complex operations without the creation of temporary files. Most MOG utilities use standard input and standard output to read in and write out converted geometry data.

Thus by creating a new MOG utility to handle RADIANCE files, any geometry that can be converted into MOG format can now be converted into RADIANCE geometry format.

7.1.2. Mog2rad Utility

The new MOG utility program, mog2rad, takes MOG format files as its standard input and writes out the RADIANCE geometry to a number of files . The MOG format has a unique group code attached to each geometrical object which was assigned in the original description of the geometric model using the original modeler. For the EAGLE modeler this group code is either specified through the fragment number or attribute string. The *mog2rad* program uses this group code as names for its output files. In this way different geometric objects can then be assigned to different materials.

Thus to convert an EAGLE model in to RADIANCE geometry description files the following can be used;

eagle2mog model.asc | *mog2rad*

If the EAGLE model consisted of say a pipe, two spheres and a box, all on different fragment numbers the output files of the mog2rad program might be as follows;

polygon-10.rad   sphere-11.rad   sphere-12.rad reducer-13.rad

See volume three of this dissertation for the *mog2rad* source code.

7.2. User interface

The RADIANCE user interface is also based around the EAGLE, Raytrace user interface. It in fact has been mirrored to use the same material library. This is to maintain a common standard in order to minimise the problems of learning a new system.

7.2.1. Eagle2rad Macros

The *eagle2rad* macros offer the designer the ability to select parts of their 3D model and assign predefined material properties to them. This is achieved through a set of mouse driven menus. A more detailed description of this process

may be found in "Using Raytrace while running EAGLE" by Oliver (1991).

Once the designers have finished their material selection, which include the selection of light sources, they are able to convert the model into RADIANCE scene description files and render a view (The conversion process as described in section 7.1 happens in the background out of sight of the user).

The user has the opportunity to either display the scene using RADIANCE's interactive viewer (*rview*) and modify the viewing parameters or to render a high quality picture. Other options include checking how much of an image has been generated and displaying the image on the screen.

7.2.2. Materials

As mentioned above the materials available in the RADIANCE package are to be modeled as closely as possible to the raytrace material list. This has its disadvantages as some materials are easier to define in raytrace properties than in RADIANCE, (water for example), and some attractive RADIANCE materials are not defined in the raytrace material list. So far only a small number of materials have been modeled so as to demonstrate the user interface.

7.2.3. Lights specification

The RADIANCE interface does differ from the raytrace interface when it comes to the assigning of light sources. Since RADIANCE is being used here as a lighting design tool, it is important that a number of different lighting types are offered. Currently the following lamp types are offered;

Sphere globes

60 Watt	GLS incandescent
100 Watt	GLS Incandescent
150 Watt	GLS incandescent

Down Lighters

75 Watt	R-40 Flood w/o shielding (ies05)
150 Watt	R-40 Flood w reflector skirt (ies06)
150 Watt	PAR/FL 140 mm dia. recessed baffled downlight (ies10)

Strip lights

Two lamp prismatic wrap around (ies35)
Four lamp fluorescent unit dropped diffuser,   610
          mm wide (ies 40)

7.3.   Other Programs

As well as the *mog2rad* program a few other small programs have been written. They are;

7.3.1. Trans program

This small program converts a transmissivity value into a transmission value. This is used in the definition of the glass material.

7.3.2. Colour program

This simply converts three colour values between 0 and 255 into a percentage.

7.3.3. Pic2dat program

This program extracts x and y data from a RADIANCE picture data file into two separate files to be used by the texdata function.

7.3.4. Standard RADIANCE Makefile

A standard Unix makefile has been developed that can be used to automatically create RADIANCE pictures. It reads all of the materials description files from a *mats* subdirectory, all of the scene geometry description files from a *rads* subdirectory and viewing files from a *vp* subdirectory . The options in the makefile are to create an octree (the default), view the picture through the interactive package *rview*, render a picture using *rpict*, filter an image using *pfilt* and view an image using *ximage*.

8. Conclusion and areas for future research

The RADIANCE rendering package has shown itself to be an excellent tool for the design and evaluation of lighting. If used by lighting designers it would offer much more freedom in the approach and evaluation of innovative new lighting designs.

The RADIANCE package as a whole lacks a user friendly up front interface but is designed primarily as a research tool. The number of features and its versatility however far outweigh its rough appearance. Thus when coupled with a 3D modeling system a very powerful design tool is created.

The platform that was created by linking the EAGLE 3D modeling system to the RADIANCE package via a user friendly interface has created a system where designers are able to start to consider lighting design more intensively than they may have done in the past. It is a perfect tool for students to be introduced to the concepts of lighting and to quickly and efficiently obtain hands on experience.

The following is a list of subjects that have not been tackled by this dissertation and that offer possibly future directions of research in this area.;

> Not all the RADIANCE materials have been closely evaluated. Such materials as prism1 and prism2, dielectrics and bidirectional reflectance distribution functions (BRDF's) could be the subject of further investigation.

> At present no daylighting has been incorporated into the EAGLE macro user interface. The actual setting up of the sky source is not difficult but in order for exterior windows to act as secondary light sources more complex selection and analysis procedures of the 3D EAGLE model would be required.

> A more complete set of lighting types could be set up for the user interface. At present only a handful of standard lights are given.

There is the opportunity for a macro to extract data from the EAGLE model (such as surface area) so that the user can design their own light fixtures.

Also more local, up to date photometric data could be converted into distribution functions to increase the current lighting library.

At present there is only a small number of defined materials (for demonstration purposes) in the user interface library. This should be increased. If matching the CAN Raytrace material library is of importance, then it may even be worth while writing a new texture primitive to read textures from RADIANCE picture files as "bump maps". At present this is only possible by converting the picture file into text data. The resultant data files are quite large and cumbersome.

One rather interesting and useful project would be to create a function that would enable procedural functions to be specified without having to transform the pattern to match the orientation of the surface. Currently a number of the same material must be defined for different surface orientations of a surface. For example a north - south brick texture would look stretched if the surface was rotated 45 degrees about the Z axis.

The RADIANCE package offers some animation options that have not been explored.

There are currently limitations on producing hardcopies of the final photo-realistic image. The technique used involves simply taking photographs of the screen. This area has not currently been investigated in the school. See appendix B.

9.   References

G.Ward   *"RADIANCE reference manual"*
                                    Lighting Systems Research Group
                                    Lawerence Berkeley Laboratory 1988-92

C.Larson   *"RADIANCE users manual (draft)"*
                                    199Lawerence Berkeley Laboratory
                                    Berkeley, California,   1991

R.Hall   *"Illumination and Color in Computer*
                                    Generated Imagery"
                                    Springer-Verlag,   New York 1989

D. Rogers   *"State of the Art in Computer Graphics"*
                                    Springer-Verlag,   New York 1991

A. Marriott & P. Dench   *"Image Generation using the Computer*
                    Animation Negus Raytracing System"
                                    Technical report, no. 3, Curtin University of
                                    Technology 1990.

I. Ashdown   *"Radiosity and Realism"*
                                    The C Users Journal, pp 33-41, Aug 1992

Illuminating Engineering   *"IES Lighting handbook 1981 reference*
Society   *volume"*
                                    Illuminating Engineering Society,
                                    New York   1981.

A. Oliver   *"Using RAYTRACE while running EAGLE"*
                                    Technical Report. no1. Curtin University of
                                    Technology. 1991

G.Ward   *"RADIANCE software system"*
                                    Lighting Systems Research Group
                                    Lawerence Berkeley Laboratory 1988-199

P.Dench   *"MOG Software system"*

Computer Animation Negus, 1989
CAN Computer Software.

Appendix A

LISTING OF PROGRAMS

RADIANCE PROGRAM LIST.

aedimage	- RADIANCE driver for AED 512 color graphics terminal
arch2rad	- convert Architrion text file to RADIANCE description
calc	- calculator
cnt	- index counter
dayfact	- compute illuminance and daylight factor on workplane
ev	- evaluate expressions
falsecolor	- make a false color RADIANCE picture
findglare	- locate glare sources in a RADIANCE scene
genbox	- generate a RADIANCE description of a box
genprism	- generate a RADIANCE description of a prism
genrev	- generate a RADIANCE description of surface of revolution
gensky	- generate a RADIANCE description of the sky
gensurf	- generate a RADIANCE description of a functional surface
genworm	- generate a RADIANCE description of a functional worm
getbbox	- compute bounding box for RADIANCE scene
getinfo	- get header information from a RADIANCE file
glare	- perform glare and visual comfort calculations
glarendx	- calculate glare index
ies2rad	- convert IES luminaire data to RADIANCE description
lam	- laminate lines of multiple files
lampcolor	- compute spectral radiance for diffuse emitter
lookamb	- examine ambient file values
mkillum	- compute illum sources for a RADIANCE scene
neat	- neaten up output columns
normpat	- normalize RADIANCE pictures for use as patterns.
oconv	- create an octree from a RADIANCE scene description
pcomb	- combine RADIANCE pictures.
pcompos	- composite RADIANCE pictures.

pfilt	- filter a RADIANCE picture
pflip	- flip a RADIANCE picture.
pinterp	- interpolate/extrapolate view from pictures
protate	- rotate a RADIANCE picture.
psign	- produce a RADIANCE picture from text.
pvalue	- convert RADIANCE picture to/from alternate formats
ra_bn	- convert RADIANCE picture to/from Barneyscan image
ra_pixar	- convert RADIANCE picture to/from PIXAR picture
ra_ppm	- convert RADIANCE picture to/from a Poskanzer Portable Pixmap
ra_pr	- convert RADIANCE picture to/from pixrect rasterfile
ra_pr24	- convert RADIANCE picture to/from 24-bit rasterfile
ra_rgbe	- change run-length encoding of a RADIANCE picture
ra_t16	- convert RADIANCE picture to/from Targa 16 or 24-bit image file
ra_t8	- convert RADIANCE picture to/from Targa 8-bit image file
ra_tiff	- convert RADIANCE picture to/from a TIFF color or greyscale image
rcalc	- record calculator
replmarks	- replace triangular markers in a RADIANCE scene description
rpict	- generate a RADIANCE picture
rtrace	- trace rays in RADIANCE scene
rview	- generate RADIANCE images interactively
thf2rad	- convert GDS things file to RADIANCE description
total	- sum up columns
ttyimage	- RADIANCE driver for X window system
xform	- transform a RADIANCE scene description
xglaresrc	- display glare sources under X11
ximage	- RADIANCE driver for X window system
xshowtrace	- interactively show rays traced on RADIANCE image under X11

EAGLE CONVERSION PROGRAM LIST

(See volume three of this dissertation for full source listing)

EAGLE2mog	- convert EAGLE model to MOG geometry format
mog2rad	- convert MOG geometry to RADIANCE description
pic2dat	- convert RADIANCE picture into data files
trans	- calculate transmission value from transmissivity
colour	- convert R G B colour values into percentages.
Makefile	- standard make file for RADIANCE image generation

EAGLE MACROS LIST

(See volume three of this dissertation for full source listing)

rad2egl	- convert RADIANCE description to EAGLE model
radprops	- assign RADIANCE materials to an EAGLE model
radconv	- convert EAGLE model and generate RADIANCE image

Appendix B

INDEX TO COLOUR IMAGES

Underwood gallery.

Zurich house foyer.

Note: The images presented here have been produced by taking a photograph of an image off a monitor, which has then been enlarged to a4 size with a colour copier. Thus there is a large error factor in the colours of the images.

Original picture colour information is altered:

1)   From the image data to the monitor (as monitor colours differ).
2)   From the monitor to the film (depending on type of film and exposure etc.).
3)   From the film to the print (depending upon the developing)
4)   From the print to the colour copy (especially with enlargement)