

School of Architecture, Construction and Planning

**SEEING THE LIGHT:
AN INTEGRATED DESIGN APPROACH FOR
AUSTRALIAN CONDITIONS**

SIMON MICHAEL DALRYMPLE CRONE

**“This thesis is presented as part of the requirements for
the award of the Degree of
MASTER OF SCIENCE (BUILDING STUDIES)
of the
Curtin University of Technology”**

December 1997

TABLE OF CONTENTS

ABSTRACT	1
PREFACE	2
Conventions	2
Acknowledgments	3
1. INTRODUCTION	4
1.1 In the beginning....	4
1.2 Explanation of title	6
1.3 Aim	7
1.4 Issues to be discussed	7
1.5 Method	7
1.6 Results	8
1.7 Format of thesis	9
2. BACKGROUND	11
2.1 Overview	11
2.2 Rendering techniques	11
2.2.1 Representation or simulation of reality?	11
2.2.2 Definitions of materials	14
2.2.3 Ray-tracing and Radiosity	17
2.3 Current CAD platform	20
2.3.1 EAGLE description	20
2.3.2 RAYTRACE Description	22
2.3.3 REAGLE description	23
2.3.4 RADIANCE description	23
2.4 Daylight design	25
2.4.1 Daylight background	26
2.4.2 Standard daylight design (CIE)	27
2.4.3 More advanced daylight models	29
2.5 Summary	30
3. LOCAL DAYLIGHT MODEL	32

3.1 Overview	32
3.2 Model calculations	32
3.2.1 Time	33
3.2.2 Solar Position	34
3.2.3 Solar radiation	35
3.2.4 Sunlight	36
3.3 Kittler's model calculations	38
3.3.1 Solar irradiance	40
3.3.2 Daylight	40
3.4 Summary	41
4. DAYLIGHT INTEGRATION	43
4.1 Overview	43
4.2 Existing RADIANCE sky distributions	43
4.2.1 Gensky program	43
4.2.2 Gendaylit program	46
4.3 Genrad program integration	48
4.3.1 Genrad usage	49
4.3.2 Genrad Examples	50
4.4 Summary	52
5. THE PLATFORM INTEGRATION	54
5.1 Overview	54
5.2 REAGLE system design	54
5.2.1 Geometry conversion	57
5.2.2 Material and environment control	59
5.3 RADIANCE software additions	60
5.3.1 Bump maps	61
5.3.2 Material orientation	64
5.3.3 Light sources and sky backgrounds	65
5.4 REAGLE - RADIANCE integration	68
5.4.1 mog2rad - Torus addition.	69
5.4.2 xeagle2rad	70
5.5 Summary	71
6. THE PLATFORM'S GRAPHICAL USER INTERFACE (GUI)	74
6.1 Overview	74
6.2 REAGLE top level system design	74
6.2.1 The REAGLE GUI (reagle)	75
6.2.2 The stand-alone GUI programs	76

6.3 REAGLE RADIANCE GUI	78
6.3.1 xradconvert	78
6.4 Summary	93
7. RESULTS AND DISCUSSION	94
7.1 Overview	94
7.2 Daylight Model	94
7.2.1 The rendered skies - test room sky validation	94
7.2.2 Validation - calculated vs. measured	102
7.3 Daylight results	105
7.4 Platform process example	105
7.4.1 The problem	105
7.4.2 The materials	107
7.4.3 The lighting	108
7.4.4 The options	110
7.4.5 The solutions	118
7.5 Platform results	120
7.6 Summary	122
8. CONCLUSIONS	124
8.1 Thesis summary	124
8.2 Future work	125
8.3 The end ?	126
9. REFERENCES	128
APPENDICES	132
APPENDIX A	132
APPENDIX B	139
APPENDIX C	141
APPENDIX D	155
APPENDIX E	161

TABLE OF TABLES

TABLE 3-1- EXAMPLES OF TURBIDITY VALUES	37
TABLE 3-2 SHOWING PARTLY CLOUDY AND OVERCAST TURBIDITY VALUES KITTLER (1981).	39
TABLE 6-1 RGB COLOUR MIXES	90

TABLE OF IMAGES

IMAGE 1-1 RAYTRACE VISUALISATION	5
IMAGE 1-2 RADIANCE VISUALISATION	5
IMAGE 2-1 SIMPLE EXAMPLE OF ILLUSION BASED RENDERING	12
IMAGE 2-2 SIMPLE EXAMPLE OF PHYSICAL BASED RENDERING	13
IMAGE 2-3 TYPICAL EAGLE HIDDEN LINE PERSPECTIVE VIEW	21
IMAGE 2-4 AWARD WINNING RAYTRACE IMAGE CREATED BY PHIL DENCH 1990	22
IMAGE 2-5 EXAMPLE OF RADIANCE IMAGE SHOWING NATURAL DIFFUSE DAYLIGHT.	25
IMAGE 2-6 SKY ANGLES	27
IMAGE 4-1 GENSKY CIE CLEAR SKY DISTRIBUTION	45
IMAGE 4-2 GENSKY CIE CLEAR SKY DISTRIBUTION USING MEASURED LOCAL IRRADIANCE DATA	45
IMAGE 4-3 GENSKY CIE OVERCAST SKY DISTRIBUTION	46
IMAGE 4-4 GENDAYLIT CLEAR SKY DISTRIBUTION USING MEASURED LOCAL IRRADIANCE DATA	48
IMAGE 4-5 CIE CLEAR SKY DISTRIBUTION USING CALCULATED IRRADIANCE DATA.	50
IMAGE 4-6 PEREZ CLEAR SKY DISTRIBUTION USING CALCULATED IRRADIANCE DATA.	51
IMAGE 4-7 PEREZ OVERCAST SKY DISTRIBUTION USING CALCULATED IRRADIANCE DATA	52
IMAGE 5-1 WATER BUMP MAP	63
IMAGE 5-2 EXAMPLE OF WATER MATERIAL	63
IMAGE 5-3 METAL PANEL CLAD ROOM	64
IMAGE 5-4 METAL RELIEF BUMP MAP	64
IMAGE 5-5 SKY MAPS NOW AVAILABLE TO THE RADIANCE PACKAGE.	68
IMAGE 6-1 REAGLE PROGRAMM GRAPHICAL USER INTERFACE	75
IMAGE 6-2 REAGLE PROGRAMM PULL DOWN MATERIAL LIBRARIES	76
IMAGE 6-3 XCONVERT USER INTERFACE	77
IMAGE 6-4 XHOWMUCH USER INTERFACE	78
IMAGE 6-5 XRADCONVERT INTERFACE	79
IMAGE 6-6 FILES PULL-DOWN MENU	81
IMAGE 6-7 FILE SELECTION INTERFACE	82
IMAGE 6-8 PROCESS PULL-DOWN MENU	83
IMAGE 6-9 DAYLIGHT INTERFACE	85
IMAGE 6-10 DAYLIGHT INTERFACE (USER SPECIFIED DATE AND TIME)	86
IMAGE 6-11 DAYLIGHT INTERFACE - NO DIRECT SOLAR RADIATION.	87
IMAGE 6-12 DETAIL INTERFACE	88
IMAGE 6-13 BACKGROUND INTERFACE	89
IMAGE 6-14 MISCELLANEOUS OPTIONS INTERFACE	91
IMAGE 6-15 VIEW OPTIONS INTERFACE	92
IMAGE 7-1 CIE CLEAR SKY	95
IMAGE 7-2 CIE CLEAR SKY TEST ROOM	95
IMAGE 7-3 CIE CLEAR SKY - MEASURED DATA	96
IMAGE 7-4 CIE CLEAR SKY MEASURED DATA - TEST ROOM	96
IMAGE 7-5 PEREZ SKY	97

IMAGE 7-6 PEREZ SKY - TEST ROOM	97
IMAGE 7-7 GENRAD CIE SKY	98
IMAGE 7-8 GENRAD CIE - TEST ROOM	98
IMAGE 7-9 GENRAD PEREZ SKY	98
IMAGE 7-10 GENRAD PEREZ - TEST ROOM	98
IMAGE 7-11 CIE OVERCAST SKY	99
IMAGE 7-12 CIE OVERCAST SKY - TEST ROOM	99
IMAGE 7-13 CIE OVERCAST SKY - MEASURED DATA	100
IMAGE 7-14 CIE OVERCAST SKY MEASURED DATA - TEST ROOM	100
IMAGE 7-15 PEREZ OVERCAST SKY	100
IMAGE 7-16 PEREZ OVERCAST SKY - TEST ROOM	100
IMAGE 7-17 GENRAD CIE OVERCAST SKY	101
IMAGE 7-18 GENRAD CIE OVERCAST SKY - TEST ROOM	101
IMAGE 7-19 GENRAD PEREZ OVERCAST SKY	102
IMAGE 7-20 GENRAD PEREZ OVERCAST SKY - TEST ROOM	102
IMAGE 7-21 OFFICE RECEPTION MODEL USED FOR THE LIGHTING DESIGN PLATFORM EXAMPLE.	106
IMAGE 7-22 REAGLE SYSTEM BEING USED TO ASSIGN MATERIALS TO THE EXAMPLE MODEL.	107
IMAGE 7-23 "CRINKLY FROSTY GLASS" MATERIAL SAMPLE USING BUMP MAP.	108
IMAGE 7-24 EXAMPLE MODEL SHOWING RAYTRACE LIGHT SOURCES AND ORIENTATION.	109
IMAGE 7-25 XCONVERT OPTIONS FOR THE RAYTRACE RECEPTION MODEL	111
IMAGE 7-26 XCONVERT PROCESS - CONVERT AND RAYTRACE	112
IMAGE 7-27 XRADCONVERT - MISC OPTIONS FOR RADIANCE EXAMPLE	113
IMAGE 7-28 XRADCONVERT - VIEW OPTIONS FOR RADIANCE EXAMPLE	113
IMAGE 7-29 XRADCONVERT - DETAIL OPTIONS FOR RADIANCE EXAMPLE	114
IMAGE 7-30 XRADCONVERT - DAYLIGHT OPTIONS FOR RADIANCE EXAMPLE	115
IMAGE 7-31 XRADCONVERT - SAVE OPTIONS FOR RADIANCE EXAMPLE	116
IMAGE 7-32 XRADCONVERT PROCESS - CONVERT AND RENDER	117
IMAGE 7-33 RAYTRACE RENDERING OF THE EXAMPLE MODEL	118
IMAGE 7-34 RADIANCE RENDERING OF THE EXAMPLE MODEL	119
IMAGE 7-35 INTERIOR DESIGN IMAGE PRODUCED WITH THE RADIANCE / REAGLE PLATFORM	120
IMAGE 7-36 STUDENT WORK PRODUCED USING RAYTRACE BEFORE ACCESS TO RADIANCE	121
IMAGE 7-37 RADIANCE RENDER OF MAIN EXHIBITION SPACE SHOWING ISOLUX LINES.	122

Abstract

The field of lighting design is a highly specialised one. The control and manipulation of both natural and artificial light is a difficult aspect of building design. Proficient architects, interior designers and engineers must currently draw on their own experience and resources to evaluate their lighting designs. However, most often lighting design is overlooked and not fully understood by building designers. Those lighting designers that do have a good understanding of lighting are hampered by current conventional design methods when dealing with alternative and creative lighting designs. This is especially the case when designing with natural daylight which tends to be under utilised, particularly in climatic areas that differ greatly from the European standard daylight design skies.

There is clearly a need for a tool that enables building designers to quickly and accurately evaluate their use of localised natural light and specific artificial light in their building designs.

The recent development of accurate, physically based computer models and the resulting photo realistic computer rendered images, can provide the basis of a platform for lighting designers to visualise their lighting designs. Such a lighting platform enables designers to make more informed design decisions when looking at new approaches and unusual situations.

The aim of this thesis is to explore and formulate a working example of such a lighting design platform, where by designers can easily evaluate lighting and daylighting designs based upon real world lighting and localised atmospheric conditions. The designer is presented with an easy to use graphical user interface, that is able to produce a photometrically accurate photo realistic computer image.

This platform was achieved through the development of three key areas. The first was the creation of a process of integrating a specific physical based rendering program, RADIANCE (Ward 1996) into an existing three dimensional computer aided design modelling package. The second area consisted of creating a localised daylight model suitable to improve the accuracy of the physical based renderer when dealing with specific local atmospheric conditions. This daylight model was based on Kittler's sky irradiance model (Kittler 1982a). The final area was the development of a graphical user interface that provides an easy to use, front end to the lighting design platform.

The results of this thesis exist in a working suite of programs and graphical user interfaces where local daylight and atmospheric condition can be specified and a photo realistic image of a design produced. The success of this platform can be demonstrated by the quality of the images that are produced. A designer can begin to use the interface with ease after very little instruction and thus start to quickly evaluate their design in terms of lighting manipulation and control. This ability to visualise and assess lighting and its effects, both natural daylight and artificial, will inevitably lead to a higher quality of successful building designs.

Preface

Conventions

A number of typographical conventions have been used in this thesis to aid in the ease of reading. They are primarily concerned with specific computer related systems, commands and user interface controls. These items often appear within normal paragraphs and, as such, have been typographically altered in order to stand out from the normal paragraph text. The conventions adopted are outlined below.

The `courier new` font is used to indicate that the text is a command line program that may be executed within a UNIX shell. Options that are also needed by these programs are printed in the `courier new` font and usually follow the command text. It is also used to indicate raw text from a computer file, such as may be inputted into a program.

The capitalisation of certain words is used to indicate a reference to a complete software system, not just an specific individual program. This applies only to the RADIANCE, RAYTRACE, REAGLE, MOG and EAGLE software systems.

The use of `helvetica` font refers to an item that is part of a graphical user interface, such as a text label. Specifically words formatted in an *italic helvetica font* indicate a particular setting of a graphical user interface option, while text outlined in “*quoted italics helvetica*” indicates an individual button.

Acknowledgments

This thesis would not have been possible without the support, direction, assistance and encouragement of a number of people.

I would, firstly, like to thank my supervisor, Neville D'Cruz for his patience in waiting for this written text and his faith in knowing that it would indeed eventuate.

I would also like to thank my associate supervisor, Terry McMinn, for sharing his valuable computing knowledge and technical assistance and for his perceptive comments on the final draft.

Thanks must also go to Andrew Marriott and Phil Dench, for keeping me away from the “dark side” of computing science .

Finally I would like to thank my family and friends for their support and encouragement throughout the thesis, especially Gaby for keeping me together in mind, body and spirit and who will never have to do the washing again.

1. Introduction

1.1 In the beginning....

This thesis has developed from my fascination with computer based visualisation techniques and their application to the discipline of architectural design. The utilisation of the computer as a visualisation tool has freed architects and designers to explore new avenues of design of the built environment. It allows them the opportunity to experiment and evaluate new or complex creations in a much faster and inexpensive way than traditional approaches. Also, as a communication medium, photo-realistic images of the final design are extremely effective at conveying the designer's concept to clients. As the saying goes, "a picture tells a thousand words."

As a design tool in the architectural world, the computer first made its presence felt with the running of three dimensional modelling programs. Suddenly architects were able to design and evaluate their three dimensional creations in a way that was much more powerful than the traditional methods they employed. They could easily visualise the three dimensional spaces they were creating, and communicate their ideas in much more accessible ways. With the addition of surface shading and simple rendering techniques, a very powerful visualisation tool was created. Finally, with the use of near photo-realistic visualisation techniques, clients no longer had to understand lines drawn on a page. They could see exactly what was up for offer.

The work in my Bachelor of Architectural dissertation (1992) centered around comparisons between a number of such photo-realistic visualisation techniques. The following images (from the dissertation) show an office foyer rendered using two different visualisation packages.



Image 1-1 RAYTRACE Visualisation



Image 1-2 RADIANCE Visualisation

The first image (Image 1-1) was rendered with a computer software package called RAYTRACE (Marriott 1989) developed by the School of Computing Science at Curtin University. This package is an “illusion” based renderer, as it does not strictly adhere to the laws of physics when tracing light through the scene. The second image (Image 1-2) was produced using the Lawrence Berkeley Laboratory’s RADIANCE (Ward 1988a) rendering software. This is a physical based rendering package, as it follows the passage of light through a scene, according to physical laws. One of the outcomes of the dissertation work showed that the key differences in the types and overall success of the visualisation techniques depended upon how each renderer modelled lighting. For example, in the second image, the ceiling is slightly coloured

due to the diffuse light being radiated off the marble floor up onto the ceiling, while in the first image, the ceiling remains grey. Techniques that attempt to physically model the way light behaves in the real world produced superior realistic results, producing more photo-realistic images. Lighting emerged as the critical element.

Accurate visualisation of physical lighting has a direct practical application to architects and building designers in the area of lighting design. The control of light is one aspect of building design that is either overlooked, or not fully understood and encompassed, by most designers in the design process.

1.2 Explanation of title

One possible reason for this limited emphasis on lighting is that building designers have immense trouble in visualising and evaluating the effects of light on and in their buildings. This is especially the case when it comes to daylight.

Daylight is under-utilised by architects, particularly in climatic areas that differ greatly from the standard daylight design skies. This is a major area of concern, as correct use of daylight has many advantages. Daylight is a free light source, it can be used to aid a building's thermal dynamics and has positive psycho-physical effects on the building's occupants (Kittler 1982b).

However, architects often place a low priority on daylight design. Their buildings are designed in ways that ignore the usefulness, or the problems associated with, daylight. There is clearly a need for a tool that enables designers to quickly evaluate the successful use of localised natural light and specific artificial light in their building designs. A picture of the interaction of light, on or in the building, is the most effective output of such a tool. The designer can then see instantly where any problems may arise. Such advanced physical based realistic rendering techniques, as mentioned above, can now be applied to specific lighting tools and used by designers. It is this opportunity to visualise the design in its real world lighting conditions that can lead to a more successful building. Thus the title of this thesis: "Seeing the light: an integrated design approach for Australian conditions."

1.3 Aim

The aim of this thesis is to discuss the issues involved in, and finally formulate a working model of, a specific physical based lighting design platform for Australian-based architects or other building designers. This platform, or suite of tools, allows these designers to easily evaluate lighting and daylighting designs based on real world lighting and atmospheric conditions. The designer is presented with an easy to use and understand graphical user interface, that outputs a photometrically accurate photo-realistic computer image for Australian lighting conditions.

1.4 Issues to be discussed

In order to create this platform the shortcomings, possible solutions and integration of elements that might make up such a tool are discussed. Such elements include common visualisation techniques, the computer graphics user interface currently in use at the School of Architecture, Construction and Planning, as well as general localised daylighting models.

1.5 Method

The process of creating such a tool was broken down into three separate areas.

The first consisted of exploring the process, and resolving the problems of how the physical based rendering program RADIANCE, could be integrated into the current computer modelling and rendering platform used by students in the Department of Architecture. This rendering platform, called REAGLE (Marriott 1991), consists of a graphical user interface that links together a three dimensional computer aided design (3D CAD) package, EAGLE and the illusion based rendering program RAYTRACE. The REAGLE package enables students to build a three dimensional model, assign materials from a material library, place lights and render a close to photo-realistic image. However since RAYTRACE is not a physical based renderer, it cannot be used as a lighting tool. The situation then existed to seamlessly integrate the RADIANCE software into the REAGLE platform, thus allowing the students access to the new physical based renderer, without the need to learn a completely new

computer program. The integration of RADIANCE into REAGLE involved finding a way of converting EAGLE geometry into RADIANCE geometry, converting the existing RAYTRACE material library into RADIANCE materials, creating a library of real world light fixtures, and adding extra functionality to the RADIANCE program to provide compatibility with the RAYTRACE program.

The second area consisted of creating a localised daylight model suitable to improve the accuracy of the physical based renderer in dealing with specific local atmospheric conditions. This was due to the fact that one of the major problems with the RAYTRACE program is that it has no daylight rendering capabilities. Although RADIANCE has the ability to calculate daylight designs, they are for the most part, based on standard European (CIE) skies, and often do not relate well to localised Australian atmospheric conditions. More accurate designs can be created if accurate local direct and diffuse irradiance values are used as input to the RADIANCE sky distribution. This need for a more localised model led to the creation of a computer program based on Kittler's sky irradiance model (Kittler 1982a) that calculates these direct and diffuse horizontal irradiance values.

The final area involved creating a graphical user interface, familiar to students, that integrates the running of RADIANCE and the daylight model in a seamless way into the REAGLE system, thus providing a front end to the physical based lighting design platform. The user interface is consistent with the current REAGLE procedure of producing a rendered image, but has had some extra functionality added. This includes the ability to control the daylight parameters, overall quality of the final image, diffuse lighting parameters and adds some new viewing types.

1.6 Results

The results of this thesis exist in a fully working suite of programs and a graphic user interface where local daylight and atmospheric conditions can be specified, and a photo realistic image of the 3D CAD design produced. The success of this platform can be demonstrated by the ease and quality of the images produced. A student can start to use the interface after very little instruction, not only to produce the photo-

realistic image, but to start evaluating their design in terms of successful lighting control.

1.7 Format of thesis

The format of this thesis is in line with the three areas mentioned above.

Chapter two provides an introduction and review of the background involved in all three areas. This includes the concepts inherent in current rendering techniques, their shortcomings, a discussion of the software used in the Department of Architecture, Construction and Planning at Curtin University, as well as the issues involved in daylight models.

Chapter three deals with the background and process of creating a local daylight computer model that extend the capabilities of the RADIANCE package.

Chapter four explores the issues involved with the initial integration of daylighting program detailed in Chapter 3 into the RADIANCE software package.

Chapter five provides the detail of the process involved and problems associated with the integration of the RADIANCE software into the REAGLE visualisation system.

Chapter six details the implementation of issues presented in chapters three to five into the lighting design tool via a graphical user interface.

Chapter seven gives examples of, and discusses, the results obtained from this platform.

Chapter eight provides a closing discussion, looking at outstanding issues and scope for future work.

Appendix A contains graphical results of the sky model calculation program along with the raw calculated data.

Appendix B contains Perth daylight data.

Appendix C contains a collection of appropriate RADIANCE manual pages.

Appendix D contains modified RADIANCE calculation files.

Appendix E contains high quality printed versions of many of the images incorporated in the thesis.

2. Background

2.1 Overview

This chapter provides an introduction, review and identification of shortcomings of the major areas involved in the creation of a computer based lighting tool for designers. It looks at photo-realistic rendering techniques, the computer aided design platform currently in use by the Department of Architecture at Curtin University, and finally at daylighting models that are used in current day lighting design.

2.2 Rendering techniques

The production of a photo realistic picture is one of the more appropriate outputs of a lighting design tool for non-engineering designers to evaluate the success of lighting designs. This is due to the high level of visual feedback available from an image, as compared to more traditional lighting tools outputting numerical tables and charts. A successful lighting installation is not only measured quantitatively but also qualitatively. The quality or ambiance of such an installation cannot readily be obtained from numerical data, where as a photo-realistic image quickly conveys the quality of the lighting. This photo-realistic image however must represent the three dimensional scene as realistically as possible. Thus the type of rendering process that produces the image is critical.

2.2.1 Representation or simulation of reality?

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

- illusion based rendering; and
- physical based rendering.

Although the end results often look similar, the processes that create them are vastly different. One can be used as an evaluation tool while the other cannot. Thus only one is applicable for use as a lighting design tool.

2.2.1.1 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. Image 2-1 shown a pictorial example of such an illusion based renderer.

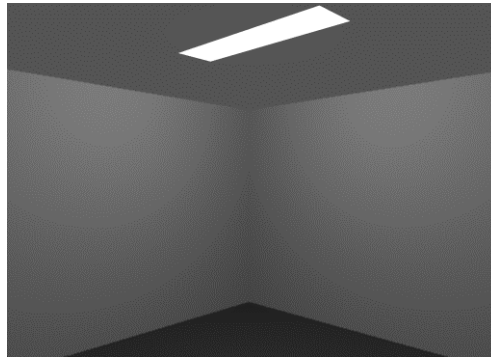


Image 2-1 Simple example of illusion based rendering

The advantages of this technique are that results may be obtained very quickly and easily without fully describing the exact physical characteristics of materials. It is possible to adjust all the scene's material and light parameters until a visually pleasing image is created. It is also possible to render images that are beyond the realm of physical techniques.

The disadvantages are that the images that are produced cannot be used to evaluate a real world situation. For example the ceiling of the room in Image 2-1 is evenly lit and does not accurately show any diffuse light bouncing back onto the ceiling from the walls, as would be apparent in true lighting conditions. Illusion based techniques can use a variety of different ways of specifying lighting and materials and, as such, do not accurately depict a real world scenario.

2.2.1.2 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 behavior 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 in terms of misinterpreting the scene. The emphasis is on numeric simulation not on artistic impression. Image 2-2 shows an example of such a physical based renderer.

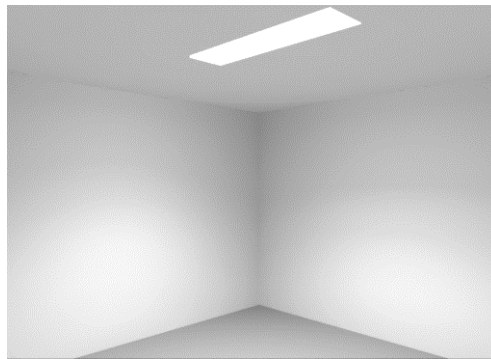


Image 2-2 Simple example of physical based rendering

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 lighting in a specific scene. The critical element in producing these images is the lighting representation. Both direct and diffuse lighting must be correctly modelled to produce a realistic and accurate result. This can be seen in Image 2-2 where the correct dark shadowing occurs in the top corner of the room. Specific luminaire photometric distribution data can be obtained directly from lighting manufactures and used to provide even more accurate rendering results.

The disadvantage of using this technique is often the long time taken to render the image (rendering time). Since diffuse or global illumination must be calculated, computational time taken to produce the image is very long. Often the need to specify all correct material and lighting physical characteristics also adds extra time to the image production process. However, it is critical to obtain realistic results, to make sure that the type of reflections and transmissions of light between materials are

correctly specified. It is the complex way that light interacts with these materials that determines overall effectiveness of the final picture.

2.2.2 Definitions of materials

All rendering techniques used to create realistic images of a physical environment must be able to accommodate two types of light and surface interactions. These two surface or material interactions are clearly summarised by Hall (1989). They consist of how the boundaries between materials interact with light, and how light is scattered and absorbed as it passes through a material.

2.2.2.1 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 dielectric materials, while boundaries with opaque materials reflect most light.

2.2.2.2 Reflection and transmission

There are two types of reflection and one of transmission commonly used in rendering techniques. They are:

- Lambertian diffuse reflection;
- specular reflection; and
- transmission.

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). Lambertian reflection (I_d) (Equation 1) can be written as;

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

Equation 1

Where I_i is the intensity of the light source measured in candelas;

K_d is a constant of reflection dependent on the surface material; and

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 from a diffuse surface is independent of the viewer position.

Specular reflection

Specular reflection is due to the shininess of a surface. Unlike diffuse reflections, it is highly dependent 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*. This reflection model produces perfect mirror reflections: 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) as shown in Equation 2:

$$\theta_r = \theta_i$$

Equation 2

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 (1975) and is discussed by Rogers (1991). In this model, the intensity of the reflection (I_s) (Equation 3) is

proportional to the cosine (raised to some power) of the angle between the mirror direction and viewer direction.

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

Equation 3

where I_j is the intensity of the light source in candelas;

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

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

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 $n = \text{infinity}$ will result in a mirrored surface.

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 (Equation 4) as described in the Illuminating Engineering Society (IES) Lighting Handbook (1993).

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

Equation 4

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

θ_i is the angle between the incident light ray and the surface normal;

n_i is the index of refraction of the second medium; and

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

2.2.3 Ray-tracing and Radiosity

Defining the interaction of light on or through a material's surface is only part of the total rendering technique. The light itself must be calculated through the scene. All rendering techniques, both illusion based and physical based, deal with direct lighting. Direct lighting is the light that falls on a surface directly from a light source specified in the scene. However, to physically model a scene, indirect lighting must also be calculated. This indirect lighting occurs from surface to surface inter-reflections. In the real world, surfaces are lit, not only directly from light sources, but also indirectly from the diffuse reflections off other surfaces. The modelling of this indirect light is referred to as global illumination and is of great importance for realistic image rendering. The algorithms of such global illumination techniques were introduced by Goral, Torrance, Greenberg and Battaile (1984) and has led to two separate areas of research. These areas concentrate on a ray-tracing and a radiosity approach to global illumination solutions.

2.2.3.1 Ray-tracing

The use of ray-tracing in the computer image rendering field has been around the computer graphics industry since 1980 when it was introduced by Whitted (1980). However, it was some time later that it was developed to be able to be used to determine global illumination solutions (Ward, Rubinstein, Clear 1988).

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 enter our eyes and stimulate the retina, which result in us seeing a picture of the world. However only a minute fraction of all the rays emitted from a light source at a particular time ever enter our eyes. Thus if this forward ray-tracing rendering model was used to produce a photo-realistic picture, it would take an unacceptably long time for any sort of image to be generated, as millions of lighting calculations would be computed that were never needed or used in the creation of the final image.

The basic direct lighting ray-tracing technique (ignoring the calculating global illumination) is to simulate the way the eye sees the world but in reverse. It involves

tracing rays backwards from the eye out into a scene and following the reflected and transmitted light rays down to a specified level. When a ray from the eye hits a surface, any number of events may occur. If the surface is diffuse then the ray stops and the brightness of the surface point is calculated. However if the surface has some reflectivity, then a new ray is created and traced off in the direction of the reflection. If the surface also has some transparency then an additional ray is created and traced off through the surface. These newly created rays travel through the scene until they strike a surface where the same process is repeated. This could go on *ad infinitum* so must be stopped after a certain number of reflections / transmissions. The end user usually has control over the number of these reflections. Shadow testing is also done for every ray intersecting with a surface. To test if a point on a surface is in shadow from any light source, a number of rays are created and traced directly towards all defined light sources. If the rays intersect a surface before they reach their specific light source, then the original point is in shadow from that particular light source.

The problem with this direct only ray-tracing technique is that the rays stop at diffuse non reflective / transparent surfaces, causing a quantity of diffusely inter-reflected light to be missed. This has traditionally been accounted for by the addition of ambient light as discussed by Rodgers (1991). This loss of diffuse light is unacceptable in a lighting simulation rendering system.

One solution to this problem is to use recursively ray-tracing to evaluate the energy transfer integral equation for specified surface points as defined by Kajiya (1986). This process determines the outgoing radiance in terms of incoming radiance over the projected sphere. However, this cannot be applied to every surface point, as the convergence of the solution would require tracing a huge number of rays and become too slow to be useful. One practical implementation of this, however, is to only sample a defined number of points on a surface and interpolate the results between the points. This is acceptable in terms of accuracy, as diffuse inter-reflections only change gradually over surfaces as discussed by Shirley (1990). One advantage of this method is that the diffuse calculations need only be done for view-dependent surfaces and not

completely over the scene, as with many radiosity techniques, thus saving time taken to generate a single image.

2.2.3.2 Radiosity

The second approach to determining a global illumination model is the radiosity technique. The basic radiosity imaging method is based on radiative heat transfer principles from thermal engineering, adapted to visual radiation. The radiosity method calculates the spectral radiance, or visual energy, leaving a surface incorporating all energy reflecting from every other surface in a scene. The technique can be used for accurately modelling area light sources, diffuse reflections, colour bleeding between surfaces, penumbras along shadow boundaries and detailed shading within shadows. Since ray-tracing is more effective at modelling 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 global illumination solution is in the form of a pre-calculated surface illuminance for every part of a surface in the scene. This is done by subdividing every surface 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). 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. Another benefit of radiosity, especially over ray-tracing, is that the technique creates view independent solutions. That is it pre-calculates all surface radiance values before it starts to render an image. Once these illuminance values for the scene have been refined, any view of the scene may be created with very little computational effort for any perspective view. This has allowed for the development of a real time interactive walk-through of a realistic simulated environment. One of the main drawbacks of this technique however, is that in order

for so many elements to be calculated, large amounts of memory and CPU time will be used. The recursive nature of subdivision of patches into elements and the subsequent convergence into a diffuse lighting solution also mean that very large or complex geometric models take considerably longer time to compute. The preference for totally diffuse surfaces also limits the sophistication of realistic materials in the production of photo-realistic images.

2.3 Current CAD platform

Curtin University's Department of Architecture has been at the forefront in use of computer visualisation as a design tool. One of the reasons for this is the close relationship it has with Curtin's School of Computing Science. In 1985 the computing school implemented a ray-tracing program and system called RAYTRACE, that became the visualisation tool used by students in the architecture department. Over the next seven years the program was substantially developed and tailored to suit an architectural market with input from architecture students. This section describes the existing visualisation platform, its current limitations and an introduction to the RADIANCE program that will be used as the basis for the lighting design platform.

2.3.1 EAGLE description

EAGLE is the name of the three dimensional computer aided design package currently in use by the department's architecture and interior design students. It is an advanced, command line driven, X window interface based, 3D surface modelling package, running on Sun workstations and in its latest versions also on PCs. The software is capable of building up complex models of any 3D object from simple primitives, such as multi-edged extruded solids, faces, cylinders (pipes), toruses (curved pipes), to sub models, annotation, lines, arcs and circles. Many attributes may be assigned to each of these elements, such as edge colour, surface colour, and surface materials. Each element or collection of elements are able to be created and manipulated while being viewed from any direction. Such manipulations include rotations, scalings and transformations, all about any axis. It does not conform to the paradigm of working in 2D orthographic sections and extruding the 3rd dimension. This gives the students a very powerful three dimensional grasp of their model.

The software also has an advanced macro scripting language enabling complex batch processing to be performed. Currently EAGLE is a command line driven program meaning it does not have a graphical user interface. This is in fact very useful as command line user input is most often faster than traditional menu-based forms of program input. The software also has limited rendering capabilities, such as hidden line surface removal and simple surface shading.

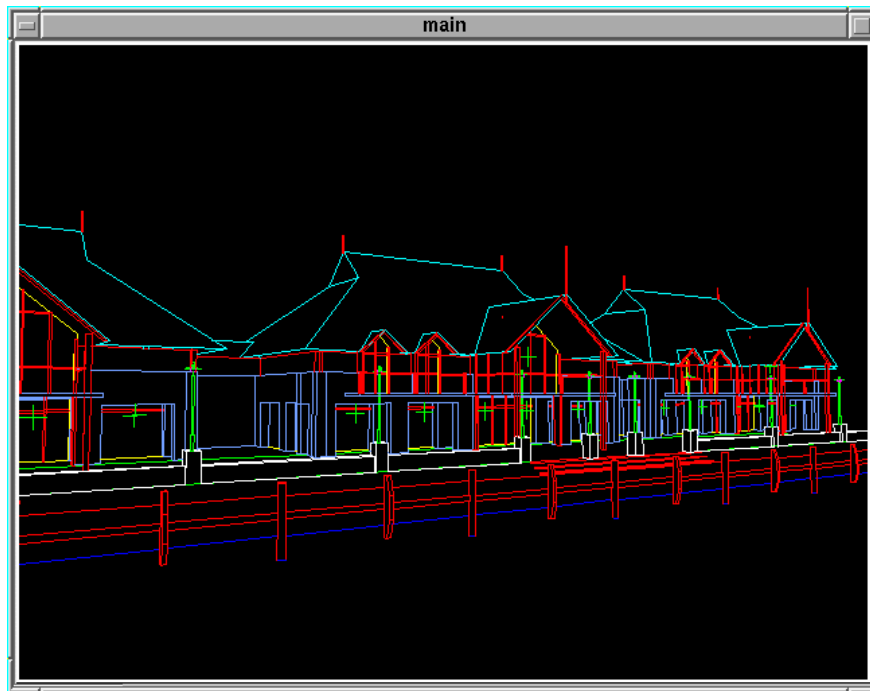


Image 2-3 Typical EAGLE hidden line perspective view

Image 2-3 shows a typical hidden line perspective view of a small cluster of buildings. This lack of realistic rendering capabilities encouraged the integration of the RAYTRACE ray-tracing image software program with EAGLE.

The process of creating a computer model involves the construction of 3D building elements using whatever primitives are applicable for the type of element. These building elements can be saved to disk and then arranged in an hierarchical way to build up the total design. Each one of the primitives or elements can be assigned a particular material, either as they are being created or later on in the model construction process. Thus RAYTRACE rendering can be carried out during the

model building process, acting as a design tool for the designer rather than just as a presentation tool.

2.3.2 RAYTRACE Description

The RAYTRACE program (Marriott 1985) was developed by the School of Computing Science at Curtin University. It is an application of backward ray tracing that deals only with direct lighting. Its geometric surfaces can only be defined by combinations of diffuse reflection, specular reflection, or regular transmission. However it does have a detailed recursive system of defining materials with such properties as texture maps, bump maps and functions that can be used to create the appearance of very realistic materials. It also has a variety of global environments that are used to give some background to an otherwise floating-in-space scene. It is for these reasons that it has been used so extensively by architecture students. Image 2-4 shows an example of a RAYTRACE rendering created by P. Dench of K. Stewart-Fox's fifth year (1989) architecture design dissertation.



Image 2-4 Award winning RAYTRACE image created by Phil Dench 1990

It clearly illustrates the ability of the program to be used as a tool for illustrating the spatial quality and style of a particular design. A full description of the ray-tracing software can be found in "Image generation using the Computer Animation Negus Raytracing System," (Marriott and Dench 1990).

Although it produces near photo-realistic images it cannot be used as a lighting tool, as it is not a physical based rendering program. Its definition of light sources do not correspond to real life lighting situations, and it does not calculate diffuse inter-reflections. Diffuse lighting is catered for by the use of an average ambient factor that is added to all surfaces. Because of this it cannot be used for daylight design, although it has the ability to do simple sun shading simulations. One of the biggest problems with the RAYTRACE program is that its interface is very difficult to use. The conversion of EAGLE geometry and inclusion of surface materials into a format compatible with the RAYTRACE program must all be done manually by the end user, and is a complicated process.

2.3.3 REAGLE description

Because of this difficulty in using the RAYTRACE software as documented by Oliver (1991), the School of Computing Science created a system called REAGLE (Marriott 1991). REAGLE is simply an X window interface based, graphical user interface, that is a bridge between the 3D CAD package EAGLE, and the RAYTRACE rendering program. This REAGLE package enables students to build a 3D model, assign materials to elements in the model from an extensive material library, place pre-defined lights, and render a close to photo-realistic image all within the EAGLE environment. This platform has been running for a number of years, and students have learnt to quickly create and render their designs, mostly for presentations. In order to use this existing platform as a lighting evaluation tool rather than just a presentation tool, a physical based rendering system must be integrated into the REAGLE platform. Rather than replacing the existing RAYTRACE renderer, it is beneficial to give the user the choice of which rendering technique to use. The physical based rendering package chosen to be added was RADIANCE.

2.3.4 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 1988a). It is a research tool for accurately calculating and predicting the visible radiation in a space by using a combination of ray-tracing and radiosity techniques.

The program uses three dimensional geometric models as input, to generate spectral radiance values in the form of photo realistic images. However, it is more than a photo-realistic presentation 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 inter-reflections, 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, such as an office interior; including rooms, furniture and lights, 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 setting up specific light sources, including 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", "analyzed", 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. A complete description of the RADIANCE package has been discussed by Crone (1992).

The disadvantages to the RADIANCE software is that it has no graphical user interface, is difficult to use, and does not accept the EAGLE 3D CAD model geometry. The situation then existed to integrate the RADIANCE software into the REAGLE platform, thus allowing students to choose the type of render they want, while not having to learn a new computer program. To do this, many integration problems had to be solved, the majority of which were done through the creation of computer programs. Specifically, a geometry converter was needed to convert the

EAGLE geometry output files into RADIANCE geometry input files; a scene and material generator was needed to create the controlling RADIANCE input files; and a new graphical user interface was vital to drive the RADIANCE package.



Image 2-5 Example of RADIANCE image showing natural diffuse daylight.

One of the benefits of RADIANCE that is currently lacking in the REAGLE package, is that it has good daylight rendering capabilities. This is demonstrated in Image 2-5 showing daylight entering an interior through window and sky lights. It is able to model the luminance distribution of the overcast, partly cloudy and clear CIE (Commission Internationale de l'Éclairage) standard skies (see section 2.4.2). Since the CIE skies are not related to localised atmospheric conditions, it also has the ability to create more accurate daylight distributions based on Perez, Michalsky and Seals (1993), when given accurate local direct and diffuse irradiance measurements. The need for these accurate localised irradiance values for non-measured sites, led to the investigation of sky irradiance models that produce these direct and diffuse horizontal irradiance values from site specific data.

2.4 Daylight design

As mentioned in the introduction, daylight is under-utilised by architects and building designers, particularly in climatic areas where the sky differs greatly from the standard daylight design skies. There is clearly a need for a tool that enables designers to

quickly evaluate the successful use of localised natural light in their building designs. This section looks at the problems related to daylight design models, current design solutions, and a local daylight model's integration into such a computer lighting tool.

2.4.1 Daylight background

One of the most desirable though most frequently misapplied sources of light is daylight. Architects often place a low priority on daylight design. Their buildings are designed in ways that ignore the usefulness, or the problems associated with, daylight. This is a major area of concern as correct use of this light has many advantages. Daylight is a free light source, it can be used to aid a building's energy usage, conservation, and thermal dynamics. It also has many positive psycho-physical effects on the building's occupants as discussed by Kittler (1982b).

This lack of sensitivity towards using natural light to advantage, is due to the difficulty in determining both the quantity and quality of daylight that may enter a building, as the levels of illuminance are so unpredictable from one moment to the next, depending upon local atmospheric conditions. This is particularly a problem in Australia as there are vastly differing climatic conditions and resulting localised sky conditions.

The difficulty for architects designing for daylight is compounded by the fact that the current daylight design methods are based on overcast or worst case skies conditions that produce the minimum daylighting levels. As such, the daylight models ignore sunlight and variable directional characteristics of different combinations of sunlight and skylight under typical or extreme levels of daylight. The Commission Internationale de l'Éclairage (CIE) overcast sky is the adopted standard used to correspond to this worst case condition. This overcast sky will provide minimum lighting levels available for a lighting design, but is totally inappropriate for maximum levels. Thus problems such as glare from bright skies are not considered. For example, with the CIE overcast or cloudy sky condition, the sky is brightest at its zenith and decreases to a value of one third at the horizon. However, in a typical Australian clear blue sky, the horizon is brighter than the zenith. Thus quite serious glare problems can

occur from external daylight filtrations. There is a need for a more realistic localised daylight model for daylight design.

2.4.2 Standard daylight design (CIE)

The standard CIE overcast sky model was adopted by the CIE in 1955 based on Moon and Spencer's (1942) empirical equation (Equation 5) which is:

$$L_{\zeta,\alpha} = \frac{L_z}{3} (1 + 2 \cos(\zeta))$$

Equation 5

where $L_{\zeta,\alpha}$ is the sky luminance at point P with spherical coordinates ζ and α in kcd/m^2 ;

L_z is the sky zenith luminance in kcd/m^2 ; and

ζ is the zenithal point angle in radians.

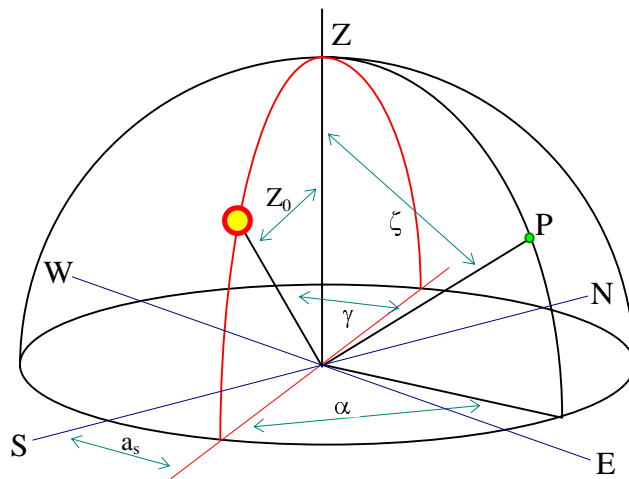


Image 2-6 Sky angles

The angles used in the daylight equations are clearly shown in Image 2-6, as adapted from the IES Lighting Handbook (1993). This overcast model was further developed by Pierpoint (1983) as (Equation 6);

$$L_{\zeta,\alpha} = L_z \left(0.864 \frac{e^{-0.52/\cos \zeta}}{e^{-0.52}} + 0.136 \frac{1 - e^{-0.52/\cos \zeta}}{1 - e^{-0.52}} \right)$$

Equation 6

The constants were still derived to fit with the original Moon and Spencer data. This model, although designed for the worst case sky conditions, is still not appropriate for Australian daylight design as this typically European overcast sky seldom occurs in most Australian climates. In Australia, the design criteria problem is often too much natural light, not too little.

The standard sky luminance distribution, modelling the extreme condition (maximum daylight levels), has been defined in the CIE (1973) clear sky function as originally developed by Kittler back in 1967. This equation (Equation 7) provides a distribution incorporating the position of the sun in the sky, unlike the symmetrical overcast sky model, and as such provides a more accurate model, especially when dealing with the orientations of building fenestration.

The CIE clear sky equation is;

$$L_{\zeta,\alpha} = L_z \frac{(0.91 + 10e^{-3\gamma} + 0.45\cos^2\gamma)(1 - e^{-0.32/\cos\zeta})}{(0.91 + 10e^{-3Z_0} + 0.45\cos^2 Z_0)(1 - e^{-0.32})}$$

Equation 7

where γ is the angle between the sun and sky point P in radians;

α is the azimuth angle from the sun in radians;

Z_0 is the zenithal sun angle in radians; and

a_s is the solar azimuth angle in radians (see Image 2-6).

This model does not, however, take into account the turbidity conditions of the atmosphere at the time, and as such, the sky distribution does not reflect the true state of the sky condition. A clear (totally cloudless) sky, may be slightly polluted or hazy which will increase the brightness of the sky in certain areas. A sophisticated daylight model is still required to produce realistic extremes of daylighting conditions for localised climates.

2.4.3 More advanced daylight models

Designing for worst case scenarios does not necessarily mean that the finished design will be an attractive, pleasant day-lit space, catering to human needs. The majority of the time these worst case skies do not occur, instead there are more typical or average skies that are constantly dynamically changing. It is desirable for a lighting design tool to be able to simulate these most favorable conditions as well as the extreme, undesirable ones.

There are a number of sky models that attempt to define the sky illuminance distribution in a more accurate way. Some approaches, such as formulated by Littlefair (1981), recommend using an average sky distribution approach to daylight design. Once again, this average approach seldom occurs in nature and does not take into account possible instantaneous atmospheric conditions.

More accurate models are available that account for the changing light distribution over the sky in terms of the local insolation conditions, that is, according to the solar radiation that is received at a specific site. Perez, Michalsky and Seals (1992), evaluated six such models, and concluded that empirically-based models were satisfactory and that a model's ability to parameterise insolation conditions was a major key to accurately simulating daylight distribution across the sky dome. Perez then went on to develop a model for describing the mean instantaneous sky luminance distribution patterns for all sky conditions, based on routine measured or calculated irradiance levels (Perez *et al* 1993). This model was validated and tested against the six models in the 1992 study and was found to produce the most accurate sky illumination distribution simulations. It is this model that has been incorporated into the RADIANCE package as a more accurate model than the standard CIE skies (Delaunay 1994).

The required input to this model is in the form of the direct and diffuse solar irradiance for a particular time of day at a specific site. Architects do not often have this sort of data readily available, or may be designing for a site location that does not have solar irradiance records. To overcome this limitation with the lighting platform,

it is possible to use physically derived calculation models to determine global and indirect sky irradiances. One such model is presented by Kittler (1982a).

Kittler's model calculates the solar radiation at a specific site by considering how solar radiation, reaching the earth from the sun, is scattered and absorbed, as it passes through the atmosphere depending upon the position of the sun and the local sky conditions. The model is described in more detail in Chapter 3. The advantages of this model is that it can consistently be used for any site knowing only a turbidity factor describing the current state of the sky. These turbidity values can be assessed from climate type or cloud cover type if no local values are available. Both the direct solar irradiance and diffuse solar irradiances can be calculated for any specific time and date, together with any local sky condition. Once these values are calculated, an accurate localised sky distribution model can be obtained, and daylighting evaluations can be created via the lighting tool.

2.5 Summary

The ideal base for the creation of a useful lighting design tool for architects and other building designers is to expand the current visualisation platform available in the Department of Architecture at Curtin University of Technology. This platform is effective, as it has a powerful underlining commercial 3D modelling package and a strongly developed, tested and extensively used graphical user interface aimed specifically at building designers.

The extension of this platform consists in the addition of a physical based rendering software capable of accurately rendering photo-realistic images of a 3D scene under all lighting conditions, including direct, diffuse and daylight. The rendering program chosen, RADIANCE, has been well validated, is ray-trace based, and originates from Lawrence Berkeley Laboratories. The daylighting distribution models that are already integrated into the fabric of the program allow the addition of localised daylighting data to be freely utilised in a daylight evaluation.

It is possible to determine localised daylighting data (direct and diffuse irradiance values) from the use of Kittler's mathematical solar irradiance model, if not readily available, thus producing more accurate results than previously possible.

The next three chapters show the application and integration of these ideas into the lighting design platform.

3. Local Daylight Model

3.1 Overview

In order to obtain the most accurate representation of the availability of daylight to be used in the evaluation of a lighting design, localised daylighting data is required. If this is not available then the situation exists to mathematically model this data, as in most situations this is more desirable than using the standard worst case design skies.

This chapter sets out the workings of a local daylight model based on a combination of IES adopted equations and Kittler's (1982a) radiation model, to produce localised daylighting data (direct and diffuse irradiance values) when only the state of the current sky conditions are known. These equations are the basis of a newly created software program *genrad*, as described in chapter 4, that extends the capabilities of the lighting design platform.

3.2 Model calculations

The basis of all the daylight and sunlight calculations depends on the ability to describe the position of the sun and its subsequent influence on the site of interest, dependent upon the instantaneous local atmospheric conditions. The following equations describe such a calculation. The description of the sun's position in the sky and its subsequent direct solar radiation levels are as prescribed by the IES (1993). It is only the indirect solar radiation levels that are needed to be generated through Kittler's model. The IES equations for calculating the sun's location and strength have been chosen as they are more advanced and more accurate than the ones suggested by Kittler in his 1981a paper. Since the use of the computer has enabled more complex equations to be easily and quickly calculated, this has made redundant some of the approximation methods, as suggested by Kittler, that enabled daylight calculations to be performed on the designer's hand held calculator. However, Kittler's indirect daylight equations are still valid, as they interpret localised climatic conditions in more detail than the IES formula.

3.2.1 Time

In order to evaluate the position of the sun in the sky, solar time needs to be calculated from the local clock time. Clock time must be corrected for:

- the eccentric nature of the earth's orbit around the sun (the equation of time);
- the variation between a site's longitude and the site's standard time zone (standard meridian); and
- any daylight saving or summer time.

The equation of time (Equation 8) is described by Lamm (1981) where:

ET is the equation of time in 24 hour decimal format; and

J is the Julian date (i.e. the number of the day of the year):

$$ET = 0.170 \sin\left(\frac{4\pi(J - 80)}{373}\right) - 0.129 \sin\left(\frac{2\pi(J - 8)}{355}\right)$$

Equation 8

Each degree of a site's longitude away from its standard meridian relates to four (4) minutes correction of solar time.

Thus solar time t_s (Equation 9) can then be expressed as:

$$t_s = t_l + ET + \frac{12(SM - L)}{\pi} - t_d$$

Equation 9

where t_l is local time;

ET is the equation of time;

t_d is daylight saving correction;

SM is the site's time zone standard meridian in radians; and

L is the site longitude in radians.

Both longitude and standard meridian are measured with positive angles being west of Greenwich (IES 1993).

3.2.2 Solar Position

Solar altitude and solar azimuth are used to describe the position of the sun in the sky. Solar altitude is the vertical angle between the center of the sun and the horizon. It is not precise due to the refraction of the sun in the atmosphere, especially as the sun approaches the horizon and sets. This is not critical in terms of building science, as typically the early morning and evening sun are blocked by other buildings or vegetation, and do not play a vital part in daylighting design. Solar azimuth is the horizontal angle of the sun measured usually in geodetic form: that is from the north with positive angles heading clockwise or towards the east, from 0 radians to 2π radians.

The sun's altitude and azimuth are functions of the observer's site latitude, the solar time and the tilt of the earth towards the sun. This tilt, or solar declination, is the angle between the sun's rays and the earth's equatorial plane at a specific date of the year, for the latitude at which the sun is directly overhead at midday. The solar declination ranges from +23.45 degrees (+0.4093 radians) on the 21st June through to -23.45 degrees (-0.4093 radians) on the 22nd of December. Solar declination δ (Equation 10), measured in radians, can be approximated as (IES 1993):

$$\delta = 0.4093 \sin\left(\frac{2\pi(J - 81)}{368}\right)$$

Equation 10

where J is the Julian date.

Solar altitude s_a (Equation 11), also measured in radians, can be given as (CIE 1990):

$$s_a = \arcsin\left(\sin(l) \sin(\delta) - \cos(l) \cos(\delta) \cos\left(\frac{\pi t_s}{12}\right)\right)$$

Equation 11

where l is the site latitude in radians (-ve for the southern hemisphere);

δ is the solar declination in radians; and

t_s is solar time in decimal hours.

Solar azimuth s_z (Equation 12), measured in radians, can be expressed as (IES 1993):

$$s_z = \arctan \left(\frac{-\left(\cos(\delta) \sin\left(\frac{\pi t_s}{12}\right) \right)}{-\left(\cos(l) \sin(\delta) + \sin(l) \cos(\delta) \cos\left(\frac{\pi t_s}{12}\right) \right)} \right)$$

Equation 12

where l is the site latitude in radians;

δ is the solar declination in radians; and

t_s is solar time in decimal hours.

Positive solar azimuth angles correspond to a direction west of south, while negative solar azimuth angles correspond to a direction east of south.

3.2.3 Solar radiation

The rays emitted from the sun can be assumed to be parallel, as they hit our atmosphere due to the large distance between the sun and the earth. Since the sun's rays are not assumed to be diffused or degraded due to minimal gaseous and particle content of space, the amount of solar radiance hitting the outer atmosphere is dependent upon the sun's distance from the earth. This distance ranges from a maximum of 152 million kilometers on the 4th of July to a minimum of 147 million kilometers on the 3rd of January. The mean distance is 149.5 million kilometers which occurs on the 3rd of April and the 5th of October. For this mean distance, the solar radiation incident on the outer atmosphere has been internationally accepted as 1350 W/m² and is called the solar constant E_{sc} (IES 1993).

For a particular date in the year, this value must be corrected due to the elliptical nature of the earth's orbit. The extraterrestrial solar irradiance E_{xt} (Equation 13) measured in W/m², can be expressed as (IES 1993):

$$E_{xt} = E_{sc} \left(1 + 0.034 \cos\left(\frac{2\pi(J-2)}{365}\right) \right)$$

Equation 13

where J is the Julian date; and

E_{sc} is the solar constant.

3.2.4 Sunlight

To calculate the direct radiation at a particular site location, the extraterrestrial solar radiation value must be reduced, due to the attenuation effects of the atmosphere.

This degradation is dependent upon:

- the optical air mass (distance ratio of the sun beams path);
- the mean extinction coefficient of the atmosphere; and
- a turbidity factor describing the particular local sky conditions at the time.

3.2.4.1 Optical air mass

The optical air mass is a ratio of the path length of the direct solar beam radiance through the atmosphere at a particular time, relative to the solar beam path length at sea level when the sun is directly overhead. It is dependent upon the height above sea level or air pressure and position of the sun. The ratio of mean atmospheric pressure at a particular site and to that at sea level can be expressed as (Equation 14 and Equation 15) (Kasten 1989):

$$\frac{P}{P_o} = 1.0 - \frac{h}{10} \text{ where } h < 4$$

Equation 14

and:

$$\frac{P}{P_o} = \exp(-0.1174h - 0.0017h^2) \text{ where } h \geq 4$$

Equation 15

where h is the site's height above sea level in km.

The relative air mass m can then also be described as (Equation 16) (Kasten 1989):

$$m = \frac{P}{P_o} \left(\frac{1}{\sin(s_a) + 0.50572 \left(\frac{180s_a}{\pi} + 6.07995 \right)^{-1.6364}} \right)$$

Equation 16

where s_a is the solar altitude in radians.

3.2.4.2 Mean extinction coefficient

The mean extinction coefficient of light through clean dry air can be defined as (Equation 17) (Navvab 1984):

$$a_r = \left(\frac{0.1}{1 + 0.0045m} \right)$$

Equation 17

where m is the relative optical air mass.

3.2.4.3 Linke's turbidity factor

Linke's turbidity factor is a measure of the state of local atmospheric conditions. It is usually drawn from either measured meteorological records or average data from a particular location. Average turbidity factors according to Kittler (1982a) are shown in Table 3-1 and can be used as an approximation if no recorded turbidity data is readily available.

Site type (pollution)	Desert Climate		Temperate Climate			Tropical Climate
Rural	2.5	2.8	3.1	3.2	3.3	3.4
Urban	3.3	3.6	3.9	4.0	4.1	4.3
Industrial	4.9	5.2	5.6	5.7	5.8	6.0

Table 3-1- Examples of turbidity values

Table 3-1 shows how the turbidity factor varies with the atmospheric water vapor content and pollution characteristic of a particular site. The amount of cloud coverage and cloud type also dramatically effect the turbidity factor.

3.3 Kittler's model calculations

Kittler (1982a) has used the turbidity factor in a double role as both a turbidity factor of the relatively clear atmosphere while the sun is shining, and as a transmission and turbidity factor of the cloudy atmosphere when the sun is hidden. When the sun is shining, the direct solar radiation increases with decreasing values of the atmospheric turbidity. The diffuse horizontal solar radiation increases, however, with an increasing turbidity factor under simultaneous sunshine. For clear skies or partly cloudy skies the turbidity value T has the limits of (Equation 18):

$$1 < T^{clear} < 10$$

Equation 18

When there is no sunshine, that is the horizontal solar irradiance is 0, the turbidity factor must also approximate the influence of the thickness filtering effect of the clouds. Thus when the sun has just disappeared so that no shadows can be the detected, behind thin cirrus, cirrostratus or altostratus clouds, the diffuse turbidity value can be quite high (e.g. between 5 and 11). Under dense cloudiness the diffuse turbidity values can range from -4.5 to 4.5. The usual limits of the turbidity value T under overcast skies is (Equation 19):

$$-4.5 < T^{cloudy} < 10$$

Equation 19

Values of the turbidity factor for use under overcast skies have been summarized by Kittler (1981b) and are shown in Table 3-2 in respect to cloud coverage, cloud type and CIE standardization of the dense multi-layer status nebulosity (St 10).

Nebulosity type	Cover	St substitute	T value
Cirrus	1 - 4	St 0	6.5
Cirrostratus	4 - 6	St (-1)	7.6
Cirrocumulus	6 - 9	St (-2)	8.7
clouds	9 - 10	St (-4)	10.9
Alto cumulus clouds	1 - 4	St 7	-1.2
	4 - 6	St 5	1.0
	6 - 9	St 3.5	2.65
	9 - 10	St 2.5	3.75
Stratocumulus clouds	1 - 4	St 9	-3.4
	4 - 6	St 7	-1.2
	6 - 9	St 5	1.0
	9 - 10	St 5	1.0
Cumulus Cumulonimbus clouds	1 - 4	St 7.5	-1.75
	4 - 6	St 7	-1.2
	6 - 9	St 7	-1.2
	9 - 10	St 7	-1.2
Altostratus clouds	9 - 10	St 0	6.5
Nimbostratus clouds	9 - 10	St 8	-2.3
Stratus clouds	9 - 10	St 10	-4.5

Table 3-2 showing partly cloudy and overcast turbidity values Kittler (1981b).

Table 3-2 shows turbidity values for typical partly cloudy and overcast skies. The use of these fictitious partly cloudy and overcast turbidity values have been independently verified from measurements in Japan by Nakamura and Oki (1979) and by Mikler (1980) in Czechoslovakia.

3.3.1 Solar irradiance

Once the optical air mass, the mean extinction coefficient and the turbidity factor have been determined, the direct normal solar irradiance E_d^{clear} in W/m^2 , can then be expressed from the classical Bouger-Lambert-Linke's formula (Equation 20), as described by Navvab (1984):

$$E_d^{clear} = E_{xt} e^{(-a_r m T^{clear})}$$

Equation 20

where E_{xt} is the extraterrestrial solar irradiance in W/m^2 ;

a_r is the atmospheric extinction coefficient;

m is the optical air mass; and

T^{clear} is Linke's turbidity factor.

The direct horizontal irradiance, E_h^{clear} (Equation 21) is then just the direct normal solar irradiance E_d^{clear} adjusted for the height of the sun in the sky. This dependence upon the solar altitude is simply expressed as:

$$E_h^{clear} = E_d^{clear} \sin(s_a)$$

Equation 21

where E_d^{clear} is the direct normal solar irradiance in W/m^2 ; and

s_a is the solar altitude in radians.

This direct horizontal irradiance value can then be used as the input for the RADIANCE sky distribution programs. The other required value being the diffuse sky irradiance or daylight.

3.3.2 Daylight

The diffuse sky irradiance is dependent on the direct horizontal irradiance and can be specified differently in these two typical separate conditions.

- a) A cloudless or partly cloudy sky where there is a definite solar corona pattern associated with sunshine.
- b) A cloudy to overcast sky with no sunshine.

In the case of sunshine (a) the diffuse horizontal irradiance $E_{h,s}^{clear}$ in W/m^2 (Equation 22), can be expressed as (Kittler 1982a):

$$E_{h,s}^{clear} = (E_{xt} \sin(s_a) - E_h^{clear})(0.025T^{clear} + 0.25)$$

Equation 22

Where E_{xt} is the extraterrestrial solar irradiance in W/m^2 ;

s_a is the solar altitude in radians;

E_h^{clear} is the direct horizontal irradiance in W/m^2 ; and

T^{clear} is Linke's turbidity factor for clear to slightly cloudy sky conditions.

In the case of no sunshine i.e. $E_h^{clear} = 0$, the diffuse horizontal irradiance $E_{h,s}^{cloudy}$ (Equation 23), in W/m^2 , can be determined by (Kittler 1982a):

$$E_{h,c}^{cloudy} = E_{xt} \sin(s_a)(0.025T^{cloudy} + 0.25)$$

Equation 23

Where E_{xt} is the extraterrestrial solar irradiance in W/m^2 ;

s_a is the solar altitude in radians; and

T^{cloudy} is Linke's turbidity factor for overcast sky conditions.

3.4 Summary

By using the above equations based the IES adopted formula and on Kittler's 1982a daylight model, it is possible to produce both a direct horizontal irradiance value and a diffuse horizontal irradiance value for a particular site location, at a specific time, knowing only the site's instantaneous atmospheric conditions. These conditions are described by either previously measured turbidity factors or turbidity factors based on the site's specific cloud coverage type and cloud densities. These calculated diffuse

and direct irradiance values can then be used to produce more accurate localised daylight distributions that give the building designer more useful feedback in evaluating their design.

In order to extend the accuracy of daylight simulations offered by the RADIANCE package, the above equations were coded into a computer program called `genrad`, developed by the author.

The next chapter documents the way the two irradiance values produced by the `genrad` program are integrated into the existing RADIANCE software package.

4. Daylight Integration

4.1 Overview

This chapter looks at the issues involved in integrating the `genrad` program, written around the sky model described in Chapter 3, into the RADIANCE software package. Currently, in the existing RADIANCE suite of software, there are two separate programs that are used to simulate daylight. They are both capable of simulating direct solar radiation as well as a diffuse sky distribution. Both of the programs optionally allow the input of measured direct and diffuse irradiance values to obtain more accurate simulations for specific individual daylight design studies. It is this feature offered by the RADIANCE programs that allows the `genrad` program to be conveniently integrated into the RADIANCE package.

4.2 Existing RADIANCE sky distributions

The two daylight simulation programs available for use in the RADIANCE package are the `gensky` program (Ward 1994a), supplied with RADIANCE and the `gendaylit` program (Delaunay 1994), available via file transfer protocol (ftp) from <ftp://ise.fhg.de/ftp/pub/radiance>. Both programs take, as standard input, such information as site location, date and time, to produce a RADIANCE scene description (a rendering input file). This scene description may contain both a sky distribution and a sunlight source, depending upon the desired sky model chosen.

4.2.1 Gensky program

The `gensky` program is able to model all of the major CIE standard skies. These sky condition variations include (where the letters in the brackets are the options used to obtain the appropriate distributions):

- a standard CIE clear day sky distribution without a sun - no direct solar radiation (`-s`);
- a standard CIE clear day sky distribution with a sun (`+s`);
- a standard CIE overcast sky distribution (`-c`);

- a standard CIE intermediate sky distribution without a sun (`-i`);
- a standard CIE intermediate sky distribution with a sun (`+i`); and
- a uniform cloudy sky, where the sky distribution is totally uniform across the sky dome (`-u`).

The output of the program produces a sky distribution brightness function in W/sr/m^2 that is used to model the sky. If a sun source (solar radiance) is to be generated, it is normally computed from the solar altitude, while the zenith radiance is computed from the sun angle and sky turbidity (for a sunny sky) using a formula developed from data gathered in San Francisco by Ward (1994a).

The input parameters needed to fully describe the conditions of a specific site at a specific time, are passed to the `gensky` program by the inclusion of these parameters on the command line that is used to run the program. These inputs include;

- the site latitude with +ve angles for northern latitudes (`-a <lat>`);
- the site longitude with +ve angles for western longitudes (`-o <lon>`);
- the site's standard meridian where +ve angles are west of Greenwich (`-m <mer>`);
- the turbidity factor for scattering in a sunny sky (`-t <trb>`);
- the average ground reflectance (`-g <rfl>`); or
- the solar altitude and solar azimuth. instead of giving a latitude, longitude and meridian (`-ang <alt> <azm>`).

A complete description of the parameters available to the `gensky` program are referenced in the `gensky` manual page included in appendix C.

For example, to produce a CIE standard clear sky distribution with a sun source, on the 4th of March at 11:00 am at Perth, Western Australia, using the default turbidity value of 2.75, the following `gensky` input parameters would be specified:

```
gensky 3 4 11 +s -a -31.95 -o -115.85 -m -120
```

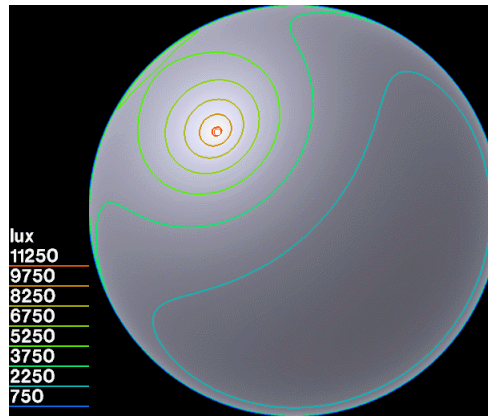


Image 4-1 Gensky CIE clear sky distribution

Image 4-1 shows a RADIANCE rendered picture of the sky dome when using the above CIE clear sky distribution.

However more accurate results could be obtained by providing local measured solar irradiance and zenith irradiance values directly to the program. This achieved by using the -R (direct normal irradiance) and -B (indirect horizontal irradiance) gensky parameters. For example, to produce a corresponding clear CIE sky distribution with a sun source for Perth using measured irradiance data, the following gensky input parameters could be specified:

```
gensky 3 4 11 +s -a -31.95 -o -115.85 -m -120 -R 720 -B 120
```

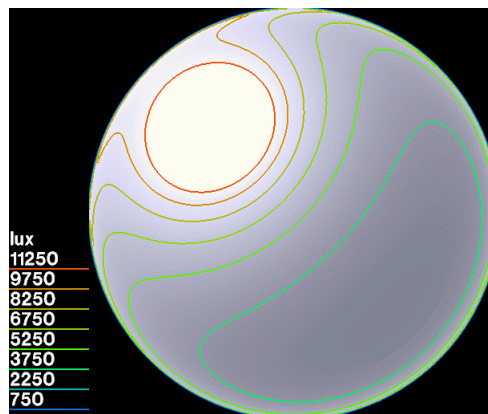


Image 4-2 Gensky CIE clear sky distribution using measured local irradiance data

Image 4-2 shows a rendered picture of the sky dome when using the above CIE clear sky distribution and measured irradiance data for Perth from Spencer's 1976 radiation study.

Even with local measured radiation data, the use of the `gensky` program has the limitations inherent in the standard CIE sky models, as discussed in chapter 2.

Often it is the typical worst case design scenario that is used to evaluate lighting designs. The `gensky` program can easily model the CIE overcast sky by specifying the following;

```
gensky 3 4 11 -c -a -31.95 -o -115.85 -m -120
```

and is shown in Image 4-3.

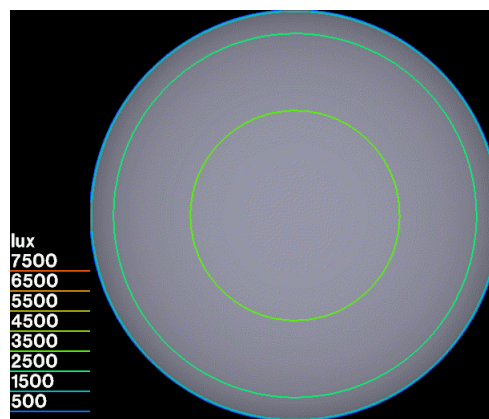


Image 4-3 Gensky CIE overcast sky distribution

A full discussion of the differences and implication of these CIE skies is presented later, in Chapter 7.

4.2.2 Gendaylit program

The `gendaylit` program overcomes some of these limitations by using a more localised set of parameterised values to create its sky distribution. Like the `gensky` program, `gendaylit` produces a RADIANCE scene description based on an angular distribution of direct and diffuse daylight at a chosen location, date, time and local atmospheric conditions. These atmospheric conditions are calculated from the direct and diffuse component of the solar radiation. The output is the direct radiance of the sun and the diffuse radiance of the sky, integrated over the visual spectral range: 380-780 nm (Delaunay 1994).

The diffuse angular distribution of the sky is based upon the Perez et al. (1993) sky luminance distribution model. All sky conditions (overcast, partly cloudy and clear) can be modelled to produce a mean instantaneous sky illuminance angular distribution pattern. The resulting sky radiance values are corrected through the normalization of the modelled sky diffuse irradiances to the measured sky diffuse irradiances.

The parameters needed for the `gendaylit` program are the direct and diffuse solar irradiance or illuminances values for a specific time, date and site location. This is the recommended input to produce the most accurate RADIANCE sky models. It is also possible to input just the Perez, Ineichen and Seals (1990) parameterisation values corresponding to atmospheric conditions, but this is not recommended by the `gendaylit` author.

The following `gendaylit` input options (as documented in the `gendaylit` manual pages included in appendix C), are available:

- the epsilon and delta Perez parameters (`-P <eps> <dlt>`); or
- the direct horizontal irradiance and diffuse horizontal irradiance values (`-G <dir> <dif>`); or
- the direct normal irradiance and diffuse horizontal irradiance values (`-W <dir> <dif>`); or
- the direct normal illuminance and diffuse horizontal illuminance values (`-L <dir> <dif>`).

The output options for specifying the units of the sky distribution can be either:

- $\text{W/m}^2/\text{sr}$ visible radiation (`-O 0`); or
- $\text{W/m}^2/\text{sr}$ solar radiation (`-O 1`); or
- lumens/ m^2/sr (i.e. luminance) (`-O 2`).

Other miscellaneous `gendaylit` options available are:

- the site latitude with +ve angles for northern latitudes (`-a <lat>`);
- the site longitude with +ve angles for western longitudes (`-o <lon>`);
- the site's standard meridian where +ve angles are west of Greenwich (`-m <mer>`);

- the option to not generate a sun source (-s);
- the average ground reflectance (-g <rfl>); or
- the solar altitude and solar azimuth instead of giving a latitude, longitude and meridian (-ang <alt> <azm>).

For example, to produce a sky description that corresponds to a clear sunny sky distribution on the 4th of March at 11:00 am at Perth, Western Australia using the same measured irradiance values as in Image 4-2, the following `gendaylit` parameters would be specified:

```
gendaylit 3 4 11 -a -31.95 -o -115.85 -m -120 -G 720 120
```

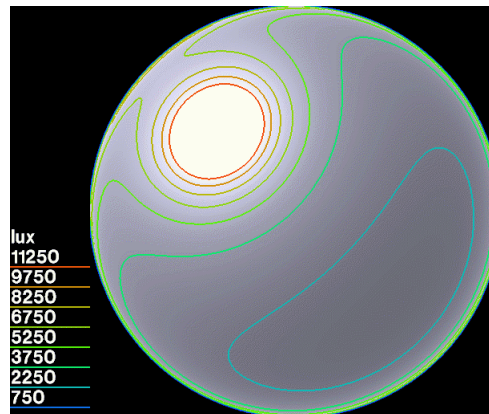


Image 4-4 Gendaylit clear sky distribution using measured local irradiance data

Image 4-4 shows the resulting sky dome from such a `gendaylit` sky description. Once again this daylight distribution is discussed in detail in Chapter 7.

4.3 Genrad program integration

The `genrad` program has been written to supply calculated direct and diffuse radiation to either of the two RADIANCE sky distribution programs, where no measured data is available. This leads to more accurate modelling for localized daylight modelling than the standard CIE skies. This local modelling enables designers to evaluate their designs with more emphasis on actual site conditions. The `genrad` program is integrated into the RADIANCE software package through the use of the existing features of both the `gendaylit` and `gensky` RADIANCE programs. Those

features depend upon the ability of being able to incorporate measured radiance values into the existing sky distribution programs.

The design of the `genrad` program has incorporated the same type of command input as both the `gendaylit` and `gensky` RADIANCE programs. This enables the program to become a replacement for the existing programs, while still retaining exactly the same usage. Thus the `genrad` program can easily be used if the operator is familiar with the existing RADIANCE sky distribution programs. The `genrad` program calculates the horizontal direct and diffuse irradiance values for a specific site and sky conditions, and then produces an output line that will run either the `gendaylit` or `gensky` with the appropriate input values.

4.3.1 Genrad usage

The input parameters to the `genrad` program are similar to the input to the `gendaylit` and `gensky` RADIANCE programs. The `genrad` program looks at the input that has been specified on the command line and then decides on the appropriate RADIANCE sky distribution program to use. If the command line input is ambiguous then `genrad` will default its output, producing a `gensky` format sky distribution.

There is one parameter to the `genrad` program that is not part of the other RADIANCE sky programs, and is needed to calculate the local horizontal direct and diffuse irradiance values. This parameter is:

- the site height above sea level in kilometers (`-h <hgt>`).

If it is not given, it defaults to 0.06 km.

All of the default input values can be listed by specifying the `-defaults` input to the `genrad` program. Currently they are hard coded into the `genrad` program and are set for Perth, Western Australia. These default parameters are:

- `-a -31.95` # Site latitude (deg);
- `-o -115.85` # Site longitude (deg);
- `-m -120.0` # Site meridian (deg);
- `-h 0.06` # Site altitude (km);

- `-t 3.4` # Site turbidity

The sky distribution output of the `genrad` program will always be in the CIE format, that is through using the `gensky` program, unless the any of the following `gendaylit` output parameters are specified to the `genrad` program. These output parameters are:

- $\text{W/m}^2/\text{sr}$ visible radiation (`-o 0`);
- $\text{W/m}^2/\text{sr}$ solar radiation (`-o 1`); or
- $\text{lumens/m}^2/\text{sr}$ (luminance) (`-o 2`).

4.3.2 Genrad Examples

The following show examples of typical usage of the `genrad` program.

The first example shows a clear sky CIE distribution using calculated irradiance data from the `genrad` program for the 4th March at 11:00 am under a sky turbidity factor of 2.75. The command line looks like:

```
genrad 3 4 11 -a -31.95 -o -115.85 -m -120 -t 2.75
```

and when executed, will in turn produce:

```
gensky -ang 57.707 -135.991 -R 841.474 -B 101.689
```

resulting in a sky distribution as shown in Image 4-5.

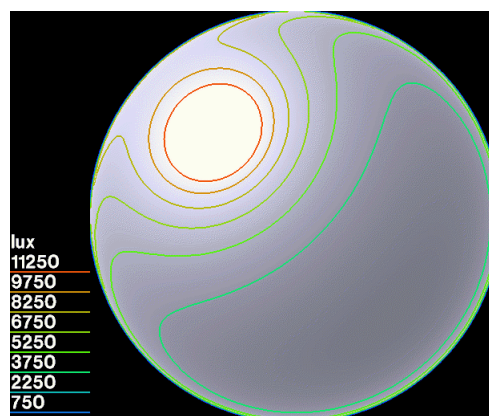


Image 4-5 CIE clear sky distribution using calculated irradiance data.

The second example shows the same clear sky distribution for the same date and time, but using the more accurate Perez sky distribution model. The `genrad` program is still used to calculate the local horizontal direct and diffuse irradiance data. The command line is specified as:

```
genrad 3 4 11 -a -31.95 -o -115.85 -m -120 -t 2.75 -O 0
```

and will, in turn, produce:

```
gendaylit -ang 57.707 -135.591 -W 995.439 101.689 -O 0
```

resulting in a sky distribution as shown in Image 4-6.

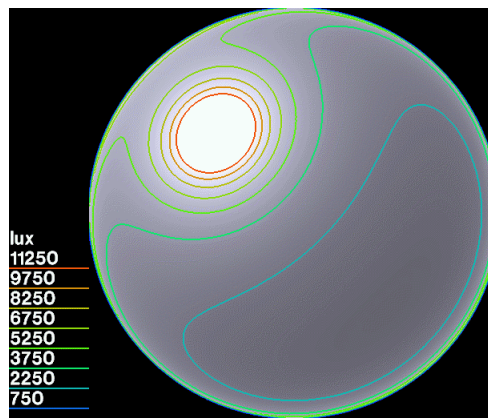


Image 4-6 Perez clear sky distribution using calculated irradiance data.

The final example shows an overcast sky distribution also using the Perez sky distribution model based on calculated irradiance data from the `genrad` program. The turbidity is set to -4.5 based on a totally overcast sky (St 10) consisting of stratus cloud and is obtained from Table 3-2. The command line is specified as;

```
genrad 3 4 11 -c -a -31.95 -o -115.85 -m -120 -t -4.5 -O 0
```

and will in turn produce;

```
gendaylit -ang 57.707 -135.591 -W 0.000 159.569 -O 0 -c
```

resulting in a sky distribution as shown in Image 4-7.

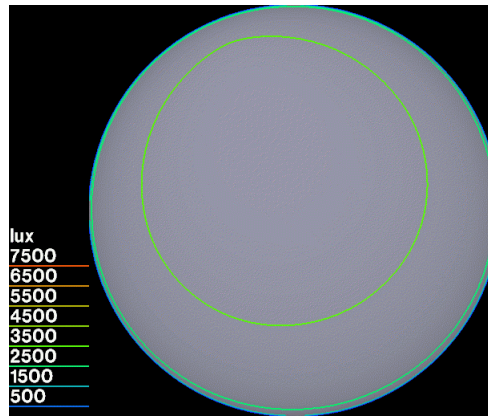


Image 4-7 Perez overcast sky distribution using calculated irradiance data

All these genrad sky examples are discussed and compared to the CIE and Perez sky distribution models fully in chapter 7.

4.4 Summary

By using the genrad program to calculate diffuse and direct irradiance values, and automatically passing these values on to the existing RADIANCE sky distribution models, much more accurate simulations can be created to assist the designer in evaluating their daylighting designs. The common forms of input parameters used to control all of the sky distribution programs enhance the integration of the genrad program into the RADIANCE software package. By using the genrad program as a transparent front end to the gensky and gendaylit programs, the process of producing RADIANCE daylight sky distributions is simplified with the advantage of producing more accurate localised daylight models. This ensures that useable tools to accurately model natural light are available to building designers.

Even with the new genrad program, there is still the need to quickly and effortlessly model an accurate three dimensional geometric representation of the design, since it is the physical scene that the designer wishes to evaluate. Currently the geometric input needed to run the RADIANCE software must be created and edited by hand in a text editor. This tends to be a long process, limiting the complexity and size of the scene to be evaluated. The next chapter explores a solution to this by integrating the RADIANCE programs directly into a 3D CAD system.

5. The Platform Integration

5.1 Overview

As introduced in the first chapter, the centre of the lighting design platform is the integration of the RADIANCE software, complete with the local daylight model (genrad), into an existing 3D modelling (EAGLE) and rendering platform (REAGLE). This integration involved the development of an EAGLE to RADIANCE geometry conversion system, the creation of a material and luminaire library, as well as adding extra functionality to the RADIANCE software to provide a compatibility with the REAGLE renderer (RAYTRACE).

This chapter focuses on the development of such an integrated platform that allows the designer to experiment, validate and present their designs in a highly visual format. The chapter is broken down into the following sections:

- REAGLE system design. The REAGLE system, developed by Marriott (1991), is a system that links the CAN RAYTRACE program with the EAGLE modelling system. It is to this system that the RADIANCE rendering functionality has been added. The section describes the underlying processes involved in the production of an image.
- RADIANCE software additions. The section describes the additions made to the RADIANCE suite of programs in order to create a unified functionality and enable the creation of a material library common to both rendering systems.
- REAGLE - RADIANCE integration. The section describes the way in which new computer programs enable the RADIANCE program to be integrated into the REAGLE platform, thus turning the existing rendering platform from a presentation and visualisation tool into a lighting evaluation tool.

5.2 REAGLE system design

The REAGLE system is a rendering platform that consists of conversion programs, a material library and a UNIX X window based graphical user interface that wraps

around the EAGLE 3D modelling program, all acting together to simplify the process of creating a RAYTRACE rendered image.

The REAGLE system has been designed upon a number of levels (as shown in Figure 5-1). The top level and most visible is the graphical user interface (GUI) that sits on top of the EAGLE modelling program. This is used to create, modify or assign materials to elements within the EAGLE model, as well as control all the subsequent levels of the rendering process, thus making the complex process of creating an image invisible to the end user. This GUI is briefly described in the next chapter.

The next level down consists of two stand alone X programs (`xconvert` and `xhowmuch`) that are used to set the user's rendering options or display existing images. They are graphical user interfaces themselves and can be launched by the REAGLE top level interface or individually from a UNIX shell.

The third level involves a controlling program (`xeagle2rin`) that takes the user's rendering input options, as specified by the higher level programs, and builds the specified material and geometry files ready for the rendering program.

The bottom level involves the rendering program RAYTRACE and geometry conversion programs. These are the critical programs of the REAGLE rendering interface as without them, no photo-realistic images could be created. The RAYTRACE program has its own very terse set of input files used to describe a scene to render. Although they are plain ASCII (text) files, their contents are not in an easily decipherable form and as such, their creation is best left up to software rather than the end user themselves. These RAYTRACE input files are a combination of geometry files, material files and environmental settings / options files.

In short, the hierarchical process is shown below.

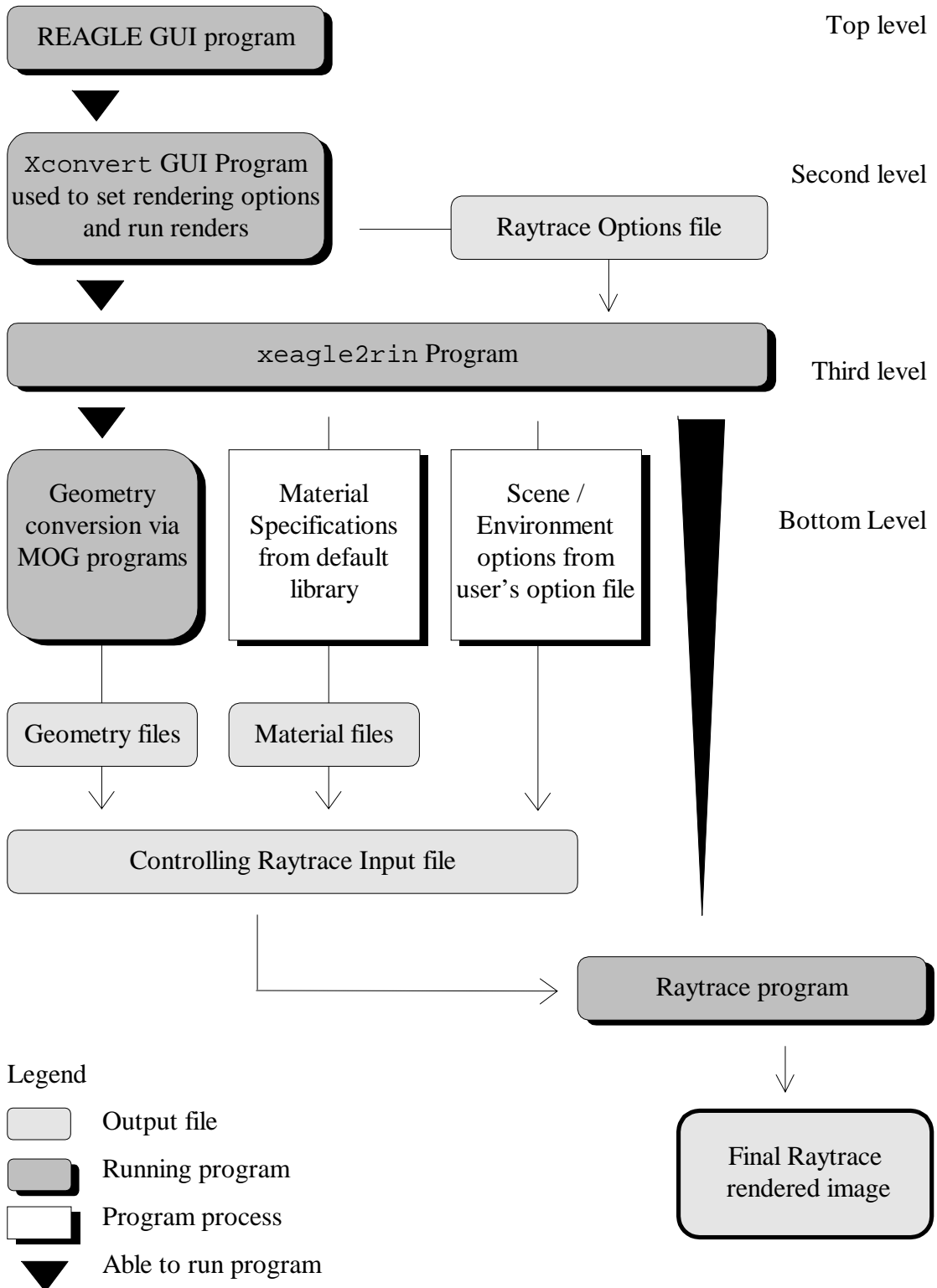


Figure 5-1 REAGLE - RAYTRACE rendering process

5.2.1 Geometry conversion

The underlying process for the creation of the RAYTRACE geometry files is set out below.

In order to extract the primitive geometry from an EAGLE model for use by the RAYTRACE program, the EAGLE model itself must not contain any other sub-models (instances to other EAGLE models). To do, this the original EAGLE model is recursively traversed, expanding any sub models found along the way, into their individual primitive elements (faces, solids etc.). This expanded model is then saved in an ASCII format: a non binary, text readable file, describing all the geometry and material codes of the scene. This is all done in the EAGLE program itself using EAGLE's custom scripting macro language.

The EAGLE ASCII file, containing all geometry and material codes, is then converted into a more structured, readable file format for users using the `eagle2mog` program. This file format is known as MOG format (My Own Geometry), and was developed by Dench (1989) to facilitate the conversion between multiple geometry file formats. The MOG system consists of many small utility programs that convert, sort, edit and combine any number of standard 3D geometric model files, by the use of a common, easily readable text file format (MOG).

The scene geometry, now in the MOG format, is split into separate files using the `mogsplit` program. The files that are created correspond to the different materials codes that had been attached to the geometry in the EAGLE program, and the geometry type. These files are then filtered and cleaned to improve storage, and finally converted, using the `mog2rt` program, into files that are suitable for input into the RAYTRACE program. Although the files have been split up into files depending upon their material code, no actual materials files have as yet been created, as this is the responsibility of the `xeagle2rin` program. The advantage of the MOG utility programs is that they can be chained (or piped) together, removing the need for temporary files.

The following is an example of the creation of RAYTRACE geometry files from an EAGLE model via the MOG utility programs. A simple model (tree.mod), consists of a lollipop style tree (a sphere on a cylinder), sitting on a flat ground plane, with all elements having been assigned the colour white (material code of '00001'). After the model has been saved in the EAGLE ASCII file format (tree.asc), the RAYTRACE geometry is created by executing the following MOG programs;

```
% eagle2mog tree.asc | mogsplit
```

This converts the EAGLE ASCII model primitives into the MOG format and creates separate MOG files based upon primitive types and material code. These MOG files contain the specification for the ground polygon, the tree trunk cylinder (reducer) and the tree canopy sphere;

```
00001_polygons.mog 00001_reducers.mog 00001_spheres.mog
```

To convert these MOG files into RAYTRACE input files (.rin), the following could be performed;

```
% cat 00001_polygons.mog | mogpunch | mogtri | mogshare |  
    mog2rt > 00001_polygons.rin
```

Here, the raw MOG polygon data is piped into the mogpunch program which checks for, and reorders, punched holes. It is then piped into the mogtri program which breaks the polygon data up into triangular patches, which are then passed into the mogshare program. This utility rearranges the MOG data into a shared coordinate format. Finally it is piped into the mog2rt program that converts the MOG data into the RAYTRACE geometry format and outputs it into the RAYTRACE input file (.rin).

```
% cat 00001_reducers.mog | mog2rt > 00001_reducers.rin
```

```
% cat 00001_spheres.mog | mog2rt > 00001_spheres.rin
```

The conversions of the cylinder and sphere primitives from the MOG format into the RAYTRACE format do not need the elaborate MOG filtering as the polygon primitive requires. Instead the raw MOG cylinder and sphere data is simply passed

straight into the `mog2rt` conversion program and out into a RAYTRACE input file format.

These `rin` files only contain the RAYTRACE geometry input. The default white colour (00001) material file still needs to be created.

5.2.2 Material and environment control

The RAYTRACE material and environment option input files are all created by a controlling program named `xeagle2rin`. This program is in charge of executing the above MOG geometry conversions, creating the RAYTRACE material files, and using the user options files (constructed by the higher graphical user interface programs) to produce the final coherent set of input files that the RAYTRACE program needs to render an image. The material files are created by copying an existing material definition from a standard material library, as indexed by the scene's geometry material codes.

The `xeagle2rin` program can be run in three separate modes. The first mode only performs the conversion of the EAGLE model from the ASCII file to the RAYTRACE inputs (all the files necessary for a render), but does not start the rendering process. The second mode is to start running a render after doing the conversion; and the third mode is to only start a render on a previously converted model. This allows the user defined options to be changed without the need to perform the time consuming geometry conversion process. The steps that `xeagle2rin` process follows in the convert and render mode are:

1. Firstly take the EAGLE ASCII file (`tree.asc` for example) and convert it into the RAYTRACE geometry files (`'00001_*.rin'`) using the MOG filters as described above in section 5.2.1.
2. For each one of the material coded geometry files (`'00001_*.rin'` in the above example), look up the material code (`'00001'`) in the material library index and make a copy of that material from the standard library area.

3. Read the user option files and extract information such as the viewing parameters, environmental conditions and any other specific options. Write out this information into the controlling RAYTRACE input file (this would be named 'tree.rin' in the above example). Then include references to all the other RAYTRACE material and geometry input files into this controlling file.
4. Finally, feed the controlling file to the RAYTRACE program. When the RAYTRACE has finished rendering the image, the image is converted into an image format suitable for viewing.

A more detailed account of the `xeagle2rin` program is described by Marriott (1994).

5.3 RADIANCE software additions

In order to integrate the RADIANCE software into the REAGLE system, there were a number of additions that were required to the RADIANCE software to enable the same user functionality to be applied for both rendering programs. These were added to the RADIANCE package at source level to be totally invisible to the end users. The main additions centered around the desire to have a common material library. Due to the integration goal of being able to render a modelled scene using either of the two rendering programs, there had to be a consistent material library between the two rendering programs. Since both rendering programs use completely different formatted material input files, a single material library common to both programs was not applicable. Instead, two separate material libraries are maintained describing exactly the same materials. Thus when an object is assigned a material in EAGLE, depending upon the rendering type chosen, either a RADIANCE material file or RAYTRACE material file is used with no visible difference in the appearance of the surface material in the final render.

In order for the photo-realistic images to be produced by the platform, the materials defined in the REAGLE systems RAYTRACE material library and RADIANCE material library must be of a high quality and detail. Generally, the most detailed

materials are defined by a combination of surface colour pattern and a surface bump texture. In the RAYTRACE material library, these colour patterns and surface textures are able to be stored in the form of colour images. These images can be scanned samples of materials or, as in the case of the surface textures, created digitally. In the RADIANCE material library only colour patterns can be stored in the form of colour images. The use of images defining both a material's colour pattern and surface texture would have the benefit of those images being able to be transferred and shared between rendering systems, thus simplifying the maintenance of the two material libraries.

5.3.1 Bump maps

RAYTRACE materials make use of bump maps to add surface textures to materials. These textures are stored as an image file and are used to modify the surface normal of an object, thus producing the effect of a bump on that surface. RADIANCE also has the ability to alter the surface normal of a surface to create a texture, but only by a mathematical function, as opposed to an image file. The public release version (v2.4) of RADIANCE does not have a bump map picture primitive and as such this limits the ability to define the same material in both the rendering programs. It does, however, have the ability to use images to modify an object's surface colour but not an object's surface texture. In order to use the same material texture and bump maps for both rendering programs (as the REAGLE only has the one material library), a bump map function had to be added to the RADIANCE rendering program. Currently this addition has only been applied to the local version of the RADIANCE program, and is yet to be officially incorporated into future RADIANCE releases. The result is the ability to create an identically appearing material from a separate RADIANCE and RAYTRACE material specification.

This newly created function, introduces a new picture texture modifier to the RADIANCE scene description format. It takes a RADIANCE picture and uses its red and green picture channels as a texture map. The red channel perturbs the x normal and the green channel perturbs the y normal. A pixel channel value of 0.5 (i.e. 128, in a range of 0 to 255) will result in no perturbation. The calculation file `texpict.cal`

is used to scale the resulting texture, "multiply" its bumpiness and return the correct surface normal modification with respect to the surface's orientation.

An example RADIANCE scene description of the new texture picture modifier consists of four lines and is shown below. The first line defines the type of geometric, material or pattern/texture primitive. In this case the new *texpict* modifier is to be used. The second line specifies five (5) string based arguments consisting of the name of the texture picture and texture calculation file to be used. The third line specifies that no (0) integer values are required. The fourth line specifies four (4) real values that are used in the texture calculation file.

```
void taxpict mat_texture
5    red grn blue pics/water.pic cal/t taxpict.cal
0
4    1.0 500.0 500.0 0.75
```

Where the 4 real arguments used are described by:

- Argument 1 is the ratio of height to width for tiles (i.e. 1);
- Argument 2 is the X scale (i.e. 500);
- Argument 3 is the Y scale (i.e. 500);
- Argument 4 is the multiplier for the surface perturbations (i.e. 1).

The taxpict.cal calculation file contains:

```
{    Calculation of 2d bump map picture perturbations.
    Picture is automatically mapped correctly to surface
    orientation.

A1 - Ratio of height to width for tiles.
A2 - X scale
A3 - Y scale
A4 - multiplier
}

pic_aspect = if(AC-.5, A1, 1);
xscale = if(AC-1.5, A2, 1);
yscale = if (AC-2.5, A3, xscale);
mult = if(AC-3.5, A4, 1);
Xp(red, gre, blue) = dx(colour_to_pert(red), colour_to_pert(gre));
Yp(red, gre, blue) = dy(colour_to_pert(red), colour_to_pert(gre));
Zp(red, gre, blue) = dz(colour_to_pert(red), colour_to_pert(gre));
pic_u = U / xscale;
pic_v = V / yscale;
```

```

match_u = tri(pic_u,max(1,1/pic_aspect));
match_v = tri(pic_v,max(1,1/pic_aspect));
tile_u = mod(pic_u,max(1,1/pic_aspect));
tile_v = mod(pic_v,max(1,1/pic_aspect));
absx = abs(Nx);
absy = abs(Ny);
absz = abs(Nz);
colour_to_pert(v) = (((v - 0.5) * 2) * mult);
dx(red, gre) = if(absx - absy, if(absx - absz, 0, if(Nz,red,red)),
    if(absy - absz, if(Ny, -red, red), if(Nz, red, red)));
dy(red, gre) = if(absx - absy, if(absx - absz, if(Nx, red, -red),
    if(Nz, gre, gre)), if(absy - absz, 0, if(Nz, gre, gre)));
dz(red, gre) = if(absx - absy, if(absx - absz, if(Nx, gre, gre), 0),
    if(absy - absz, if(Ny, gre, gre), 0));

```

By using the image show in Image 5-1 as a textural bump image, the standard RAYTRACE water material can now easily be rendered in the RADIANCE package as shown in Image 5-2.

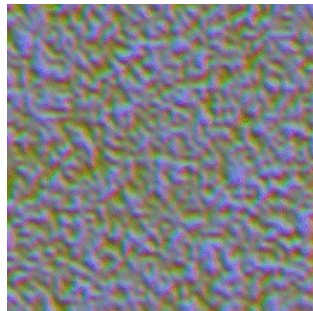


Image 5-1 Water bump map

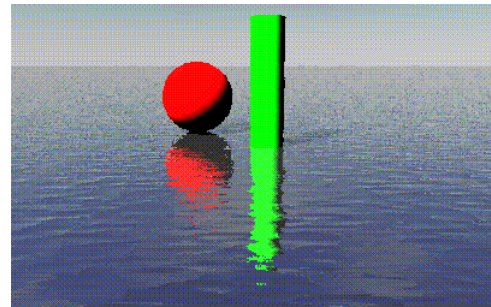


Image 5-2 Example of water material

A more interesting example is shown in Image 5-3. Here a detailed bump map (Image 5-4) has been used to create a gold metal relief material panel in a room. This complexity of bump map would be extremely difficult to represent as a mathematical function, which is how the RADIANCE software currently defines textures. With the *texpict* functionality added to the RADIANCE this complexity is now straight forward.

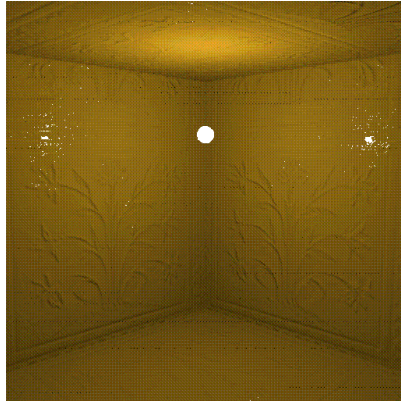


Image 5-3 Metal panel clad room



Image 5-4 Metal relief bump map

Thus the introduction of this new RADIANCE material description primitive enables the use of any of the RAYTRACE material texture map images to be defined and used by the RADIANCE program.

5.3.2 Material orientation

Although the existing RADIANCE software makes use of images to define the colour pattern of a material, it has one major limiting restriction: the colour image or pattern is automatically mapped onto the horizontal plane. The surface material must then be manually transformed to match the orientation of the actual surface of the object. This is not desirable in the REAGLE platform as there are often a large number of surfaces created having the same material, yet numerous surface orientations (a curved brick wall for example). This limitation does not occur in the RAYTRACE image pattern material definitions, as it employs automatic image transforming and orientating mappings. That is to say that a single image of a brick texture would be automatically rotated so that the bricks remained running in a horizontal direction no matter which way the geometry of the surface tilted.

A way of representing a RADIANCE colour image texture material had to be developed to mimic the raytrace way of mapping. This was achieved by adding extra functionality to the information passed to the colour image calculation files, enabling this automatic rotation to occur. Rather than using just the X and Y intersection point of a ray onto the surface in question to determine the position of the image file on the surface, the X, Y and Z intersection point is used and transformed via a

transformation matrix to determine the final image position. An example of the additions added to the RADIANCE main calculation file to achieve this automatic rotation can be found in appendix D.

5.3.3 Light sources and sky backgrounds

Once the problems of the ability to correctly map both colour pattern images and surface texture images on to a surface had been overcome, the standard RAYTRACE material library could be easily adapted and a RADIANCE material library created. There were, however, two parts of the RAYTRACE library that could not be directly transferred to the RADIANCE library. These two parts consist of all light source material definitions and global environment backgrounds.

Since RADIANCE is a lighting simulation package that simulates reality, it needs true accurate descriptions of light sources and their distributions. The light source material descriptions in the RAYTRACE software do not correspond to any real life lighting properties and thus it is not appropriate to try and create a RADIANCE material specification of them. So, in order for the lighting platform to be able to evaluate lighting designs, a new library of light sources was created from scratch to form part of the RADIANCE material library. The light sources that were defined for the lighting library, consisted of a number of differing wattages of the common incandescent light globe, as well as the standard 45 IES distribution (IES 1996) luminaire types, with their corresponding North American IES photometric data files that are included in the RADIANCE software package.

The definition of light source materials raised a problem in that, since RADIANCE is modelling reality, the power assigned to a light source depends upon the surface area of the geometry that is used to represent it in the scene. This means that each light emitting surface in a scene that is of a different area must have an individual light material allocated to it even though it may be of the same light type (incandescent). This problem was resolved by creating a library of light models, that contain both a geometric description and material definition. This way the end user of the EAGLE software can insert lights from a pre-defined library without having to worry about

trying to match their geometry correctly with a specific lighting specification. An advantage of this is that the library model, not only has the specific manufacturers correct lighting distribution, but also correct detailed luminaire geometry. This removes the risk of designers creating their own light sources that do not exist in reality (i.e. producing an exceptionally small powerful light source) and producing visually attractive yet misleading renderings.

The second part of the RAYTRACE material library that was not directly transferable to the RADIANCE library was that of global backgrounds. The RAYTRACE program has the ability to use six separate images mapped onto a global cube, to produce an overall background in an image. This background is usually of the sky in a particular state (for example, stormy, night or sunset), but the images can be of anything. The RADIANCE software also has the ability to produce a global background, but not directly from the six RAYTRACE background images. A way of converting the standard RAYTRACE backgrounds into the RADIANCE software had to be developed.

This was achieved by writing a skymap function that provided the ability to map a single background image onto a RADIANCE source primitive to create a world environment bubble. The six separate background images were combined into a single image by using the RAYTRACE program to render a single 360 degree panoramic image of the overall background. This RAYTRACE picture, after being converted into the RADIANCE image format, was used as the single image mapped onto the RADIANCE background environment. This mapping is controlled by the skymap function that simply stretches and contorts the image so as to appear normal when mapped onto a sphere / bubble, or in the RADIANCE case a source.

The most effective use of these background environments is to represent a more realistic image of the sky dome. These sky environments do not deal with any sky illuminance distributions as they are simply mapped colour images. However `genrad` can be used to further modify these sky environment as the colour values can be automatically adjusted to maintain accuracy, by using the `illred`, `illgrn` and `illblu` variables in the calculation function file.

An example of a RADIANCE scene description of the new sky environment consists of four lines and is shown below. Once again five (5) string arguments are required, no (0) integer arguments and only one (1) real argument.

```
skyfunc colorpict midday_sky
7    illred illgrn illblu
    pics/midday.pic cal/skymap.cal
    map_u map_v
0
1    0.5
```

The background material definitions take 1 real argument.

- A1 is the picture aspect ratio. Typically 0.5

The calculation file for mapping these pictures onto a sphere is skymap.cal, and contains:

```
{
    skymap.cal - Map a pre skewed sky picture onto a sphere
    A1 = aspect ratio (height / width)
}

u_scale = if(A1 -1, 1, A1);
v_scale = if(A1 -1, A1, 1);

hang = if(abs(Dx) - FTINY,
    atan2(Dy, Dx),
    if(Dy, PI/2, -PI/2));

map_u = ((hang + PI) / (PI*2)) / u_scale;
map_v = ((asin(Dz) + (PI/2)) / PI) / v_scale;

illred(r,g,b) = r / grey(r,g,b);
illgrn(r,g,b) = g / grey(r,g,b);
illblu(r,g,b) = b / grey(r,g,b);
```

By using this function, all of the RAYTRACE sky backgrounds can now be successfully defined in the RADIANCE library. Image 5-5 shows a picture of the currently available sky backgrounds in the RADIANCE library. The images are of chrome spheres reflecting each of the individual skies. The midday_lr sky uses a low resolution version of the midday background image file for use on machines where memory is at a premium.

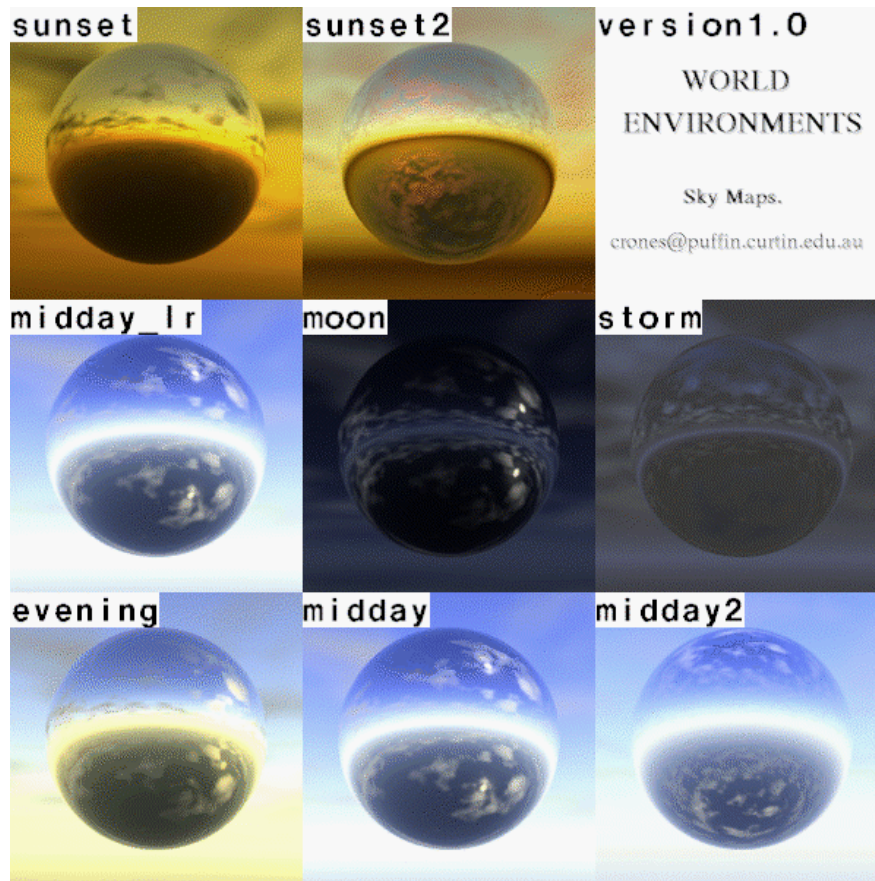


Image 5-5 Sky maps now available to the RADIANCE package.

It is now possible to have separate yet complete material, lighting and background libraries that can be used by each of the rendering programs.

5.4 REAGLE - RADIANCE integration

This section briefly describes the development of the new workhorse programs that are used to convert an EAGLE model so that it may be rendered using the RADIANCE package by the REAGLE system.

The underlying process is the same as the REAGLE - RAYTRACE system, in that the EAGLE geometry is converted and filtered into the MOG format according to the geometry's material specification. However, from this point, instead of using the `mog2rt` program, a new program called `mog2rad` is used to generate the geometry files that are suitable for input into the RADIANCE rendering software. This whole process is started by the new `xeagle2rad` program which is also used to control

and monitor the building of suitable material files and running of actual RADIANCE renders. A graphical user interface to this program is discussed in the next chapter.

5.4.1 mog2rad - Torus addition.

A new program was needed to transform the converted EAGLE geometry (in MOG format) into a geometry format suitable for input directly into the RADIANCE software. This program called mog2rad was initially developed by Crone in 1992. That first version was able to convert polygons, cylinders, cones and spheres successfully into the RADIANCE format. It could not, however, handle the conversion of the EAGLE curved cylinder primitive or torus section. To accommodate this, a much more complex and improved mog2rad program was needed.

The original program lacked this torus conversion, as the curved cylinder primitive inherent in the EAGLE CAD system does not exist as a geometric primitive in the RADIANCE geometry definitions. Thus it was not able to be directly transferred into the RADIANCE system as it was with the RAYTRACE system. The RAYTRACE program has a torus primitive but the RADIANCE rendering program does not. This could be over come in two ways. The first way available was to add a torus primitive to the RADIANCE scene description. This would involve detailed changes to the RADIANCE source code and result in a “special” version of the RADIANCE software. This was not desirable as it would reduce the ability to share RADIANCE models between different users, as each user would require the specialised RADIANCE software. In short the RADIANCE model would not be portable.

So rather than adding a new primitive to the RADIANCE source code, it was decided to represent these curved cylinders as a series of straight cylinder runs and spheres. Since most of these curved cylinders spanned a maximum angle of 90 degrees, a relatively smooth pipe curve could be generated by a small number of geometric primitives. In fact, the number of segments used to approximate the curve is calculated from the actual angle subtended by the curved pipe, currently one segment for each 10 degree step. This approximated curved pipe geometry is created with a

RADIANCE utility program called `genworm`. `Genworm` is used to produce a tubular “worm” from a set of parameteric equations defined by $x(t)$, $y(t)$, $z(t)$ with a radius of $r(t)$ where t varies from 0 to 1 in increments of 1 divided by the number of segments. Thus by passing appropriate values to the `genworm` program, an approximate curved cylinder geometry is created. For example a 90 degree ($\pi/2$ radians) curved piped with diameter of 0.1 and radius of 5 could be created in 9 segments by the following:

```
genworm A B 'sin((1-t)*pi/2)*5' 'cos((1-t)*pi/2)*5' 0 0.1 9
```

This resulting RADIANCE geometry must then be transformed and rotated into its correct position. It is this final transformation that required careful coding of the new `mog2rad` program.

With this program complete it is now possible to render all parts of the EAGLE model successfully using the original RADIANCE software. This is demonstrated by the example test scene in chapter 7.

5.4.2 `xeagle2rad`

With the EAGLE geometry conversion now complete and a defined RADIANCE material and light library, the only part left to finish the RADIANCE integration into the REAGLE package was the creation of the overall controlling program, `xeagle2rad`. This underlying driving program is based on the `xeagle2rin` program and follows closely its process and usage. It takes an EAGLE ASCII file, converts it into RADIANCE geometry, extracts the material information, builds the appropriate RADIANCE material files and sets off a render of the final scene. It is responsible for extracting the material definitions from the library and coupling them together with the geometry files to produce a RADIANCE input file. Along with this input file, a controlling RADIANCE “rad” file (`.rif`) is also created based upon specific settings, such as the viewing points of the scene, daylighting conditions and world environments. This “rad” file (`.rif`) is used by the RADIANCE `rad` program to control the rendering of a series of images (see appendix C for the `rad` manual page). Although it is automatically created, it is dependent upon information specified by the

user through a higher level graphical user interface (GUI) program. This GUI is fully described in the next chapter. With all of these files the `xeagle2rad` program is able to start running a RADIANCE render to produce a photo-realistic image of the final scene.

5.5 Summary

This chapter demonstrates the overall low level and mid level structure and process involved in integrating the RADIANCE rendering program into the REAGLE system. This process is summarised on the next page in Figure 5-2.

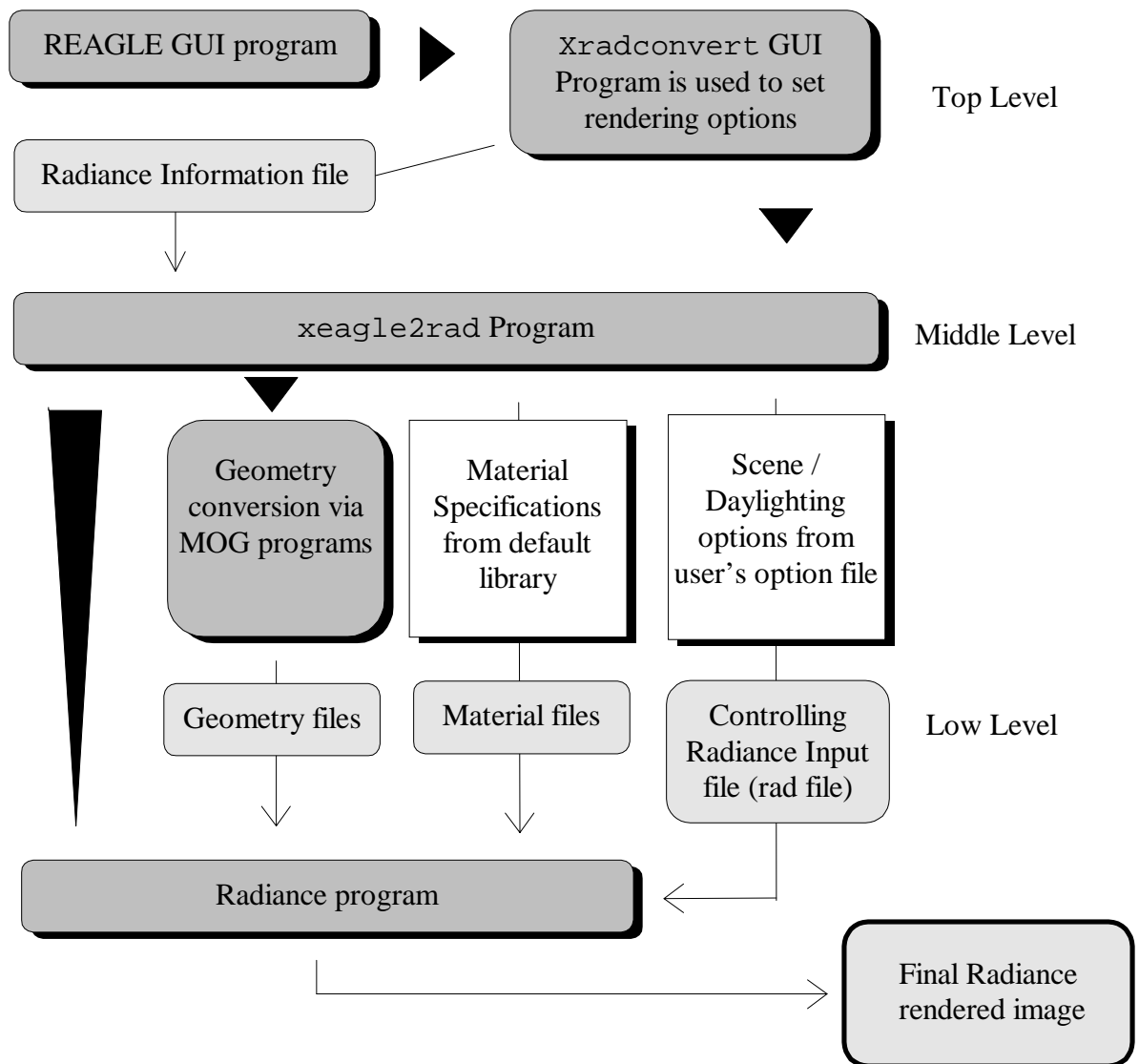


Figure 5-2 Radiance rendering Process

The major changes involved in the integration of the RADIANCE package into the REAGLE rendering system were with the low level processes and the production of the mid level processes whilst maintaining compatibility with the RAYTRACE system. These low level changes included:

- The creation of the `mog2rad` program enabling the EAGLE model geometry to be converted into RADIANCE geometry. i.e. ASCII file > mog filters > `mog2rad` > Radiance geometry file;
- The creation of a RADIANCE material library that included:
 - New bump map material definition
 - New automatic material alignment

- New world environment backgrounds
- The formation of a RADIANCE light library that also included new EAGLE light models.

The middle level process development consisted of the `xeagle2rad` program that controls or runs the lower level programs as listed above. The program is responsible for:

- Running the geometry converters;
- Copying materials;
- Creating environment description;
- Creating daylighting descriptions;
- Creating the controlling rad file; and
- Controlling rendering.

In the next chapter a new high level program, `xradconvert`, will be described, as it is the graphical user interface that is used to:

- Set up user options (daylight views etc.); and
- Control the `xeagle2rad` program.

6. The Platform's Graphical User Interface (GUI)

6.1 Overview

While the previous chapter examined the underlying low level processes involved in producing either a RADIANCE or RAYTRACE picture, this chapter deals with the top level Graphical User Interface (GUI) that has been developed to control the low level RADIANCE rendering process. This has been done in such a way as to ensure the lighting platform is easy to use from the user's point of view. The chapter begins with a review of the current REAGLE GUI system and discusses its advantages and limitations. The successful elements of this interface are the basis of the new GUI program (`xradconvert`) that manages the RADIANCE rendering of a scene. The rest of the chapter then deals with a detailed description of this RADIANCE GUI program.

6.2 REAGLE top level system design

The top level REAGLE system is an X-based graphical user interface that runs concurrently with the EAGLE modelling program. It is the top end user interface for the Computer Animation Negus RAYTRACE System and MOG Geometry System. This program, `reagle`, is responsible for managing a variety of user tasks. Primarily these management tasks include:

- the starting of the geometry and material conversion process;
- the setting up of rendering options;
- the starting of the actual rendering process;
- the viewing of an existing rendered image;
- the reporting on a render in progress;
- the cleaning up after a rendered project is finished.

The program also interacts with the EAGLE program, allowing a user to:

- assign pre-defined system and user-defined local materials to EAGLE models;

- create and define individual user material colours and reflectances;
- use images from a variety of formats as pictures/paintings/textures within the EAGLE model;
- incorporate pre-defined and user defined light-models into the EAGLE model;
- manage the system, user and local material libraries;
- create ASCII geometry files; and
- exit both EAGLE and the `reagle` program itself.

The following sections look at the REAGLE system's graphical user interface programs and discusses their advantages and shortcomings.

6.2.1 The REAGLE GUI (`reagle`)

The `reagle` program interface is shown in Image 6-1. The program itself consists of two lines of buttons. The top line contains names of the sections from the material libraries that are automatically loaded on startup. These are used to assign materials to various objects in the EAGLE model.



Image 6-1 reagle programm graphical user interface

The interface is designed in such a way that the section material lists drop down automatically whenever the mouse pointer passes over the button. This is an advantage as it allows the end user to make material selections faster than having to press buttons to activate a drop down panel. An example of such a drop down material selection panel is shown in Image 6-2. By holding down the mouse and releasing it over a specific material, that material is automatically applied to the selected objects in the EAGLE model.

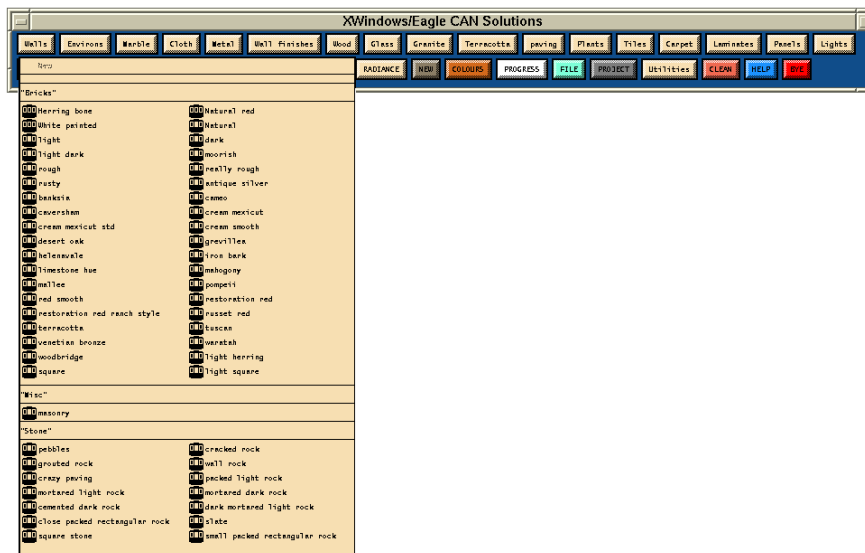


Image 6-2 reagle programm pull down material libraries

The buttons on the second line of the reagle program interface are normal push buttons and are used to perform the management functions of the program. One of the major shortcomings of the reagle program interface is the lack of a unified look to the program. The button colours and overall shape and size could be redesigned, not only in more visually pleasing format but also in a more consistent intuitive form. The software has started to do this by the constant use of a red button to exit the program and a blue button to obtain on-line help. However this idea of colour coded buttons has not been continued. This has been addressed in the new RADIANCE GUI program.

One of the major advantages of the reagle program is that it is able to control and call up standalone or external programs that are part of the REAGLE system. This means the user only has to learn how to start the reagle program itself, rather than a number of separate programs.

6.2.2 The stand-alone GUI programs

The stand-alone GUI programs are used primarily to set up the user options for the REAGLE rendering process. There are a number of user interface programs that are part of the REAGLE system, but only the mostly commonly used programs will be

investigated. These programs are the `xconvert` program and the `xhowmuch` program.

The `xconvert` program is an X-based, graphical interface to the CAN EAGLE to RAYTRACE conversion system and specifically to the `xeagle2rin` program. It allows an EAGLE user to define the image size, quality and orientation for a ray-traced render of their design. Numerous configurable options for the ray-traced environment can be set, including sky or coloured backgrounds, type of shadows (none, lights only or lights and sun at any time of the year and from any latitude). The `xconvert` program can be used to either convert, render or convert and render EAGLE models, as well as launching the `xhowmuch` program. Its design, as shown in Image 6-3, like the `reagle` program, lacks attractive colour and detailed button / field alignment.

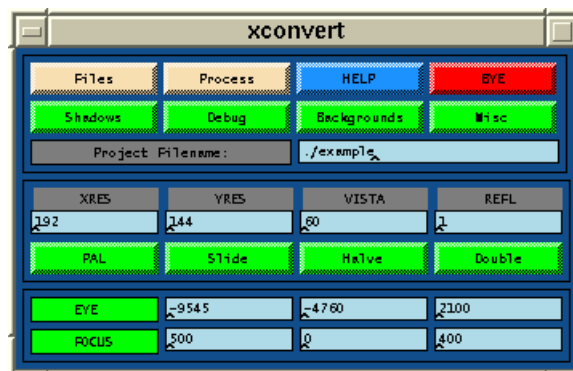


Image 6-3 Xconvert User Interface

The most commonly used buttons, (the “*Files*” button and the “*Process*” button) are similar to the `reagle` top line material buttons, as they automatically drop down when the mouse pointer passes over them. This, once again, speeds up the use of the program as there is less mouse clicking to be done by the user.

The `xhowmuch` program is used to monitor, abort and display the currently rendering raytrace image. It can also be used to display previously rendered images that are stored in the users disk. Its user interface is show in Image 6-4.

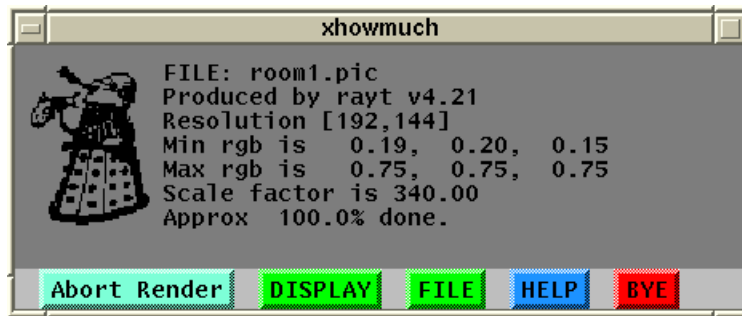


Image 6-4 Xhowmuch User Interface

Once again, with the exception of the red “*bye*” button and blue “*help*” button, the colouring of the buttons does not help in differentiating the functions of each button and detracts from the appearance of the interface. The red and blue buttons are an example of good programming technique as the end user is able to associate colours to the same functions through a number of separate stand-alone programs.

6.3 REAGLE RADIANCE GUI

The main aim of the RADIANCE GUI program was its ability to be used by experienced REAGLE users without the need to learn a totally new program or interface. This has been achieved by keeping the positive aspects of the REAGLE GUI programs, while also improving on the interface as a whole. The process that a user must go through is the same whether they wish to produce a RAYTRACE image or a RADIANCE image.

6.3.1 xradconvert

The xradconvert program is a stand alone X-based graphical user interface based on the CAN EAGLE to RAYTRACE conversion system xconvert, but for the RADIANCE system. It lets the user of the 3D EAGLE modelling system easily set common RADIANCE options, including image size, quality and views, and to render a photo-realistic picture of their model. It also lets the user specify individual environmental conditions for the RADIANCE environment, including the advanced modelling of skylight and sunlight from any time of the year and from any site position.

The main `xradconvert` program window consists of a number of push buttons, pull down menus and text input fields and is shown in Image 6-5.



Image 6-5 `xradconvert` Interface

The format of the interface is the very similar to the `xconvert` program, enabling users to quickly adapt to the new found ability of producing a RADIANCE rendered image. The most noticeable change to the interface is that a single overall colour has been applied to the interface. Only the red “*bye*” and blue “*help*” buttons are kept constant. Along with the lining up of the buttons and fields, this gives the interface a smarter look while not diminishing its function. Buttons are still easily distinguished from labels by their raised three dimensional look. User changeable text fields are shown in the window border grey. This limiting of the colours from eight, as currently set in the `xconvert` program, down to four, draws far less on the X window system resources. All of the text fields in the interface have active error checking associated with them. That is, they are able to check that the user’s input is within a sensible range. For example, values may be checked to see they are always positive (in the *Indirect* field) or between say zero and one (Matt RGB values).

The next sections describe in detail the use of each of the field options and buttons available in the initial `xradconvert` user interface, followed by the “*Files*”, “*Process*”, “*Daylight*”, “*Detail*”, “*Backgrounds*”, “*Misc*” and “*View*” buttons.

6.3.1.1 Project Filename

The current model being rendered is referred to the Project Filename. Once a model has been selected, its name is displayed here.

6.3.1.2 Image Size Options

The XRES and YRES text fields define the final pixel size of the rendered image. The “*Halve*” and “*Double*” buttons, when pressed, simply double or half the x and y image resolution values currently displayed. The user also may specify an image size that has the same aspect ratio as a 35 mm slide by pressing the “*Slide*” button. The final image size resolution is independent of the screen that the program is running on. This enables high resolution images to be professionally printed out on high dot per inch (DPI) resolution printers.

6.3.1.3 Exposure Value

The Exposure text field is used to specify the overall exposure of the final image. The exposure can be thought of in the same way that the human eye or camera can adjust its iris to vary the amount of light that is recorded. A value of *1* (one) indicates no change from the original image.

6.3.1.4 Indirect Value

The Indirect text field specifies the number of ambient bounces for the RADIANCE render to calculate. The ambient bounce refers to the number of diffuse reflections that will take place in a render. For example, to render a picture showing the effect of sunlight bouncing off a floor illuminating a ceiling, then an Indirect value of *1* (one) would be needed.

6.3.1.5 Files Options

By moving the mouse pointer over the “*Files*” button, a drop down menu is available for files, as shown in Image 6-6, and is used to either read existing or save new xradconvert options. This enables common settings to be quickly loaded, or saved for later use.



Image 6-6 Files Pull-down Menu

Both the read and save options open up a file selection interface (Image 6-7). A highlight of the xradconvert program is that the single interface window is able to enlarge and shrink to contain the new input options (in this case the file selection interface), rather than opening them in a completely new window. As there is only the single window ever open, this reduces the possibility of confusing a user with which is the next step to follow or button to press. Also, there are no windows to lose behind other windows (a common problem with some GUI systems).

As seen in Image 6-7, the file selection interface is typical of most file selection interfaces found in graphical user interfacing programs. It contains the standard directory lists, file lists, scroll bars for traversing these lists, text input field and “*Cancel*” and “*Ok*” buttons. The interface is used only to read and save files (with an extension of “.rdf” - RADIANCE Description File) that contain RADIANCE project options.

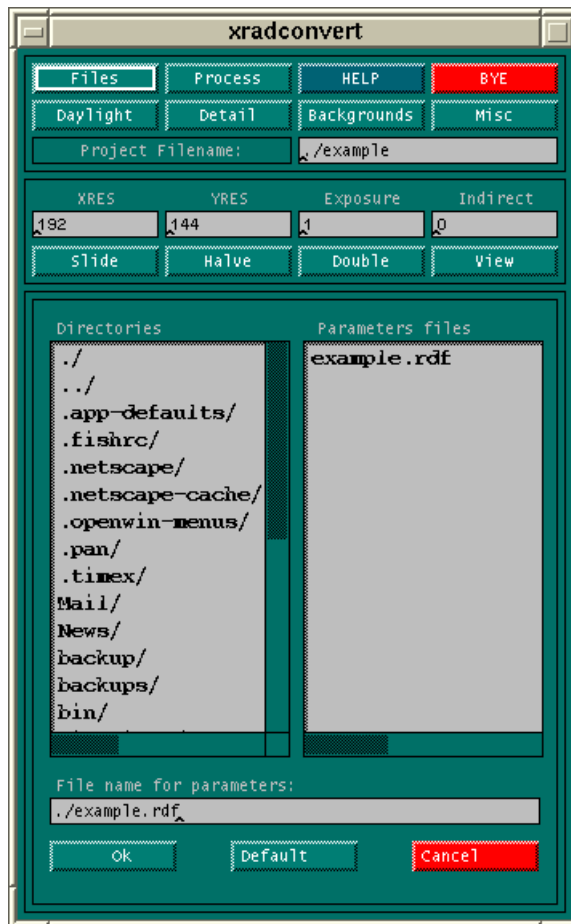


Image 6-7 File Selection Interface

The file selection interface is similar for both the read and save processes, except for an extra button in the read options window. When pressed, this extra button - the “*Default*” button - loads in a default set of image parameters, as all the xradconvert options must be set before a render can take place. After the default option has been chosen, or an options file has been selected from the listed files, or, in the case of the save options window, a new options file has been saved, the file selection interface shrinks, returning the user back to the original xradconvert interface. The current project filename is then automatically updated to this last saved or opened, options file filename.

6.3.1.6 Process Options

The next major button is the “*Process*” button. By moving the mouse pointer over the “*Process*” button with the left mouse button held down, a drop down menu is available as shown in Image 6-8. This has the following options:

- Convert;
- Radiance Render;
- Convert and Render;
- Display current image; and
- Display existing image.

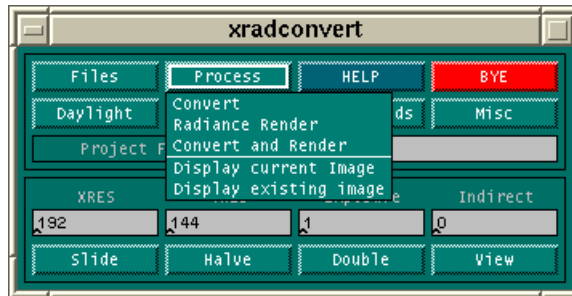


Image 6-8 Process Pull-down Menu

The *Convert* option (selected by releasing the left mouse button over the *Convert* option) starts the conversion process. This process involves converting the geometry from the EAGLE model into RADIANCE geometry files. The materials present in the EAGLE model are extracted and RADIANCE materials files are created, with reference from the current RADIANCE library of materials. Finally all of the currently displayed RADIANCE options in the xradconvert interface are collated, and the controlling RADIANCE input file (.rif) is created. This is where the process stops. No image is rendered.

When the *Radiance Render* option is selected, the program assumes the model has been previously converted, and that no materials or geometry have been changed since the last conversion, so the above conversion process can be skipped and the render process started. This involves collating all of the currently displayed xradconvert options and writing out the controlling RADIANCE input file (.rad). This file is used by the RADIANCE rad program and specifies view, image size and

quality, along with which material and geometry files to use. From this “rad” file the image is then rendered.

The *Convert and Render* option combines both the previous processes into one single option, thus converting the EAGLE model and immediately starting the rendering process.

The *Display current image* option is used to display part of the image as it is being rendered. It uses the RADIANCE ximage program to open up a new window to display the image.

The *Display existing image* option opens up a file selection interface as described above, listing all the previous rendered images. Once an existing image file has been selected, it will be displayed in a separate window.

6.3.1.7 Help Button

The “*Help*” button starts up the system HTML browser program (netscape), which automatically loads information on the use of the program.

6.3.1.8 Bye Button

The “*Bye*” button quits the program and displays a list of any problems and errors associated with that particular running of the program.

6.3.1.9 Daylight Button

Clicking on the “*Daylight*” button opens up the daylight user interface as shown in Image 6-9. Here the user controls the lighting available from the daylight environment.

xradconvert

Files Process **HELP** BYE

Daylight Detail Backgrounds Misc

Project Filename: ./example

XRES: 192 YRES: 144 Exposure: 1 Indirect: 0

Slide Halve Double View

Use Daylight? **Yes** No

Sun shining? **Yes** No

Use Today's time & date? **Yes** No

Sky distribution option

Clear Inter Overcast Perez

Latitude: -31.95 Elevation: 06

Longitude: -115.85 Turbidity: 3.4

Std Meridian: -120

MONTH

January	February	March
April	May	June
July	August	September
October	November	December

DAY

1	2	3	4	5	6	7	8	9	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31					

HOUR

1	2	3	4	5	6	7	8	9	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24

MINUTE

00	05	10	15	20	25	30	35	40	45	50	55
----	----	----	----	----	----	----	----	----	----	----	----

Image 6-9 Daylight Interface

When the Use Daylight option is set to *no*, then no RADIANCE daylight description files are created. This is used for either night images or interior only images.

When the Use Today's Time & Date option is set to *no*, as shown in Image 6-10, then it is possible to specify exactly the time of year and time of day to create a render for. As soon as this option is selected, the month, day, hour and minute fields become active, and a date and time can be selected.

The screenshot shows the 'xradconvert' application window. At the top is a menu bar with 'Files', 'Process', 'HELP', and 'BYE'. Below it is a toolbar with 'Daylight', 'Detail', 'Backgrounds', and 'Misc'. The 'Daylight' tab is active. The 'Project Filename' field contains './example'. Below this are input fields for 'XRES' (192), 'YRES' (144), 'Exposure' (1), and 'Indirect' (0). There are buttons for 'Slide', 'Halve', 'Double', and 'View'. A section titled 'Use Daylight?' has three rows, each with 'Yes' and 'No' buttons. The 'Sky distribution option' section has four buttons: 'Clear', 'Inter', 'Overcast', and 'Perez'. Below this are input fields for 'Latitude' (-31.95), 'Longitude' (-115.85), 'Elevation' (06), and 'Turbidity' (3.4). A 'MONTH' section shows a calendar grid with 'March 21' selected. An 'HOUR' section shows a grid with '12' selected. A 'MINUTE' section shows a grid with '00' selected.

Image 6-10 Daylight Interface (User specified date and time)

If this option is left on *yes* then the current date and time are used. Care must be taken that this is not the case when working late at night with the xradconvert program (as shown in Image 6-9, where the time was automatically set to 22:00 - 10:00 pm).

When the Sun Shining option is set to *no* then no RADIANCE sun source file is created. The eagle model is to be rendered without any direct sun present (for example, in a cloudy sky condition).

The next option available is the Sky distribution option or daylight type that the EAGLE model is to be rendered with. The first three options are the standard CIE *Clear*, *Intermediate* or *Overcast* skies. The fourth option is the *Perez* sky

distribution model. To produce the most accurate results, the `genrad` program, as discussed in earlier chapters, is used to generate the correct RADIANCE sky descriptions. It should be noted, as shown in Image 6-11, that when the overcast sky is chosen the **Sun Shining** option automatically changes to *no* and becomes inactive, as the CIE overcast sky is defined as to have no direct solar radiation.

The screenshot shows the `xradconvert` application window. The **Daylight** tab is selected. The **Project Filename** is `./example`. The **XRES** is 192, **YRES** is 144, **Exposure** is 1, and **Indirect** is 0. The **Slide**, **Halve**, **Double**, and **View** buttons are available. The **Use Daylight?** option is set to **Yes**, **Sun shining?** is set to **No**, and **Use Today's time & date?** is set to **Yes**. The **Sky distribution option** is set to **Overcast**. The **Clear**, **Inter**, **Overcast**, and **Perez** buttons are visible. The **Latitude** is -31.95, **Longitude** is -115.85, **Std Meridian** is -120, **Elevation** is 0.06, and **Turbidity** is 3.4. The **MONTH** is set to **October**, **DAY** is 29, **HOUR** is 22, and **MINUTE** is 00.

Image 6-11 Daylight Interface - No direct solar radiation.

The final options specify the global location of the site, and the instantaneous atmospheric turbidity factor. The **Latitude** is described in degrees where negative values denote southern latitudes. The **Longitude** and standard **Meridian** are also described in degrees, where east of Greenwich is also given in negative values. The **Elevation** above sea level is described in kilometers. The final value needed is the **Turbidity** factor as described in Chapter 3.

Clicking on the “*Daylight*” button a second time will close up the daylight interface options and return to the initial xradconvert user interface.

6.3.1.10 Detail Button

Clicking on the “*Detail*” button opens up the image detail user interface as shown in Image 6-12. This is used to set RADIANCE options that control the quality and accuracy of the final rendered image. They correspond directly to the Detail, Variability, Quality and Penumbra settings that are specified in the RADIANCE rad program as described in the rad manual page by Ward (1994b), shown in appendix C.



Image 6-12 Detail Interface

The Detail options, *Low*, *Medium* and *High* are used to specify the visual detail in the model. If a model is very simple, containing only a few similar surfaces, then the value should be set to *Low*. For a slightly more complex model with say, simple furniture and fixtures, then a value of *Medium* might be chosen. For a complex model with a great deal of detail and textures the *High* setting should be used. This value is used by the RADIANCE rad program to determine image sampling rates. The higher the setting the longer the rendering time of the image.

The Quality option specifies the overall quality of the rendered image. This has a direct effect on the accuracy and rendering time of the final image. A value of *Medium* will result in an image twice the resolution specified by the user being rendered, which is then filtered down to the originally specified size. This results in a

much higher quality image. If a value of *High* is given, the image is rendered three times the size for highest possible quality. A *High* value also results in inter-reflection calculations being automatically performed, where the Indirect setting is set to be greater than 1 (one). This type of calculation will also take place if the Quality is set to *Medium* and the Indirect value is greater than 0.

The Variability option specifies the type of light variability over the surfaces in the model. This would range from a *Low* setting for a uniformly lit room, to a *Medium* setting for a high contrasting scene, say with spot lighting or bright window, and to a *High* setting for direct sunlight falling on a few separate areas in a scene.

Clicking on the “*Detail*” button a second time will close up the detail interface window and return to the initial xradconvert user interface.

6.3.1.11 Background Button

Clicking on the “*Background*” button opens up the background user interface as shown in Image 6-13. This is used to control the appearance of the sky background or sky dome of the model.

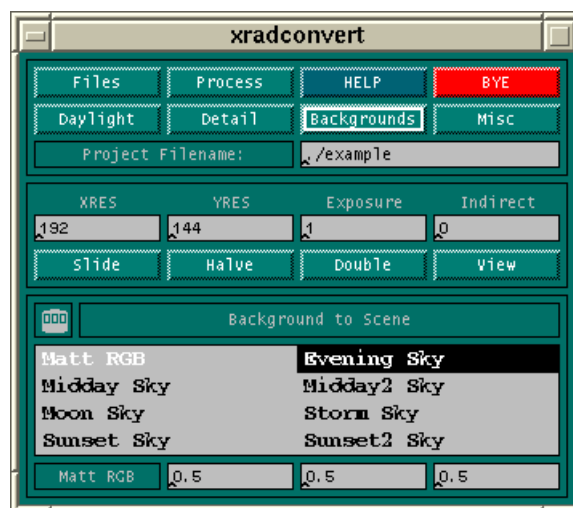


Image 6-13 Background Interface

The options available are either a solid colour for the background (*Matt RGB*) or a mapped sky image. If a plain colour is required then the Matt RGB option should be highlighted from the list, and the colour specified in the three text fields at the bottom

of the window. The three fields correspond to the amount of red, green, and blue that make up the matt colour. Each value can range from 0.0 to 1.0, with 0.0 meaning no colour and 1.0 being full intensity. Table 6-1 shows a sample of a few colours:

Colour	RGB Mix
Black	R=0.0, G=0.0, B=0.0
White	R=1.0, G=1.0, B=1.0
Red	R=1.0, G=0.0, B=0.0
Yellow	R=1.0, G=1.0, B=0.0
Sky Blue	R=0.5, G=0.5, B=0.9

Table 6-1 RGB Colour Mixes

The sky maps are 360 degree pictures of artificial skies at different times of the day. They are automatically mapped across the whole sky dome and are shown in chapter 5 in Image 5-5. The picture of the sun in the maps does not match the actual position of the sun defined in the daylight interface. A sample image of the sky maps, displayed in a new window, can be obtained by pressing the middle or right mouse button on the desired sky map listing.

Clicking on the “*Background*” button a second time will close up the background interface window and return to the initial `xradconvert` user interface.

6.3.1.12 Misc Button

Clicking on the “*Misc*” button opens up the miscellaneous options interface as shown in Image 6-14. This is used to set a few special and advanced options.



Image 6-14 Miscellaneous Options Interface

The **Zone** option is used to specify the particular type of scene. Either an *interior* or *exterior* scene. The **Bounding Box** fields specify the maximum and minimum extents of the RADIANCE scene. The **Zone** and **Bounding Box** values are used by the RADIANCE rad program to determine sensible rendering parameters. The **Bounding Box** values can easily be determined by using the EAGLE “status 1” command to display the current EAGLE model bounding box (maximum and minimum extents of the EAGLE model). With these values highlighted using the mouse, clicking on the “*Min XYZ*” and “*Max XYZ*” buttons automatically copies these EAGLE bounding box values into the correct xradconvert fields.

Clicking on the “Misc” button a second time will close up the miscellaneous interface window and return to the initial xradconvert user interface.

6.3.1.13 View Button

Clicking on the “*View*” button opens the View options interface as shown in Image 6-15. This is used to define the precise viewing parameters of the model or scene.

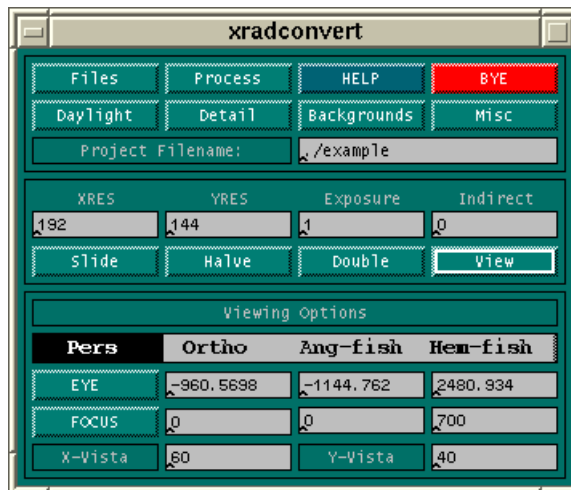


Image 6-15 View Options Interface

The user has the choice of four types of camera lenses to view the scene through. These are a normal lenses (*Pers*), a special orthographic view (*Ortho*) and two types of fish eye lenses (*Ang-fish* and *Hem-fish*). To set up the exact perspective view of the scene, the X, Y and Z coordinates of the eye (or camera) point and the focus (or target) point must be entered into the appropriate text fields. Once again this can be aided by using the EAGLE “status 3” command to display the current EAGLE perspective view positions. With these values highlighted using the mouse, clicking on the “*Eye*” and “*Focus*” buttons automatically copies the perspective values into the correct X, Y and Z fields. The final two options specify the X-Vista and Y-Vista of the image. This represents the horizontal and vertical fields of vision. A typical field of view is 60 degrees for the X and 40 degrees for the Y. Distorted images occur when greater vistas are used. For fish eye lenses, up to 360 degrees can be used in the angular fish eye and up to 180 degrees in the hemispherical fish eye. If the type of lens is orthogonal, then the X-Vista and Y-Vista relate to actual measured model dimensions each side of the eye point to display, for example, the elevation, plan or section.

Clicking on the “*View*” button a second time will close the view options interface window and return to the initial xradconvert user interface.

6.4 Summary

The creation of a graphical user interface to control the low level RADIANCE integration programs is the last piece needed in the development of a lighting design platform. It is now possible to take an EAGLE model and render a photo-realistic image representing true lighting applications. This adaptation to the REAGLE system has been performed in such a way as to encourage the use of the new RADIANCE rendering system through its ease of use.

Successful elements of the REAGLE system graphical user interface have been adapted to produce the RADIANCE interface software. This new software is consistent in use, function and process to the REAGLE system and as such has been shown to be learnt and used with minimal effort.

The next chapter provides a working example of this new lighting design platform and in the process validates its results against measured data.

7. Results and Discussion

7.1 Overview

This chapter consists of two main sections that discuss the success and validity of specific aspects of the new lighting design platform. The first section examines the validity of the daylight model program, `genrad`, by looking at test examples and comparisons with measured data. The second section documents the evaluation of the newly-created lighting interface, as presented in the previous chapter, that is used to produce more realistic computer simulations for the lighting designer. This chapter demonstrates the achievement of the aim of this thesis to provide a working model of a specific physical based lighting design platform for localised Australian conditions. Using a real world example, it shows the process of the integration of the physical based rendering package RADIANCE into the REAGLE rendering platform. This platform is then able to provide better predictive capabilities, especially in the area of daylight design, and produce more realistic computer simulated images than previously possible.

7.2 Daylight Model

The daylight model described in detail in chapters 3 and 4, has been integrated into the REAGLE - RADIANCE rendering platform, will be validated in two ways. The first is a simple comparison between the new daylight model program `genrad` and the existing RADIANCE daylight programs `gensky` and `gendaylit` on a test room. The second is a more quantitative approach comparing the output of the `genrad` program with measured irradiance data for a specific Australian site.

7.2.1 The rendered skies - test room sky validation

A small test room was modelled in EAGLE and rendered with a number of differing sky models and sky types. The three daylight models used are the RADIANCE `gensky` program, used to generate the standard CIE sky distributions; the `gendaylit` program that models the Perez sky distributions; and the newly created

genrad that is used to calculate diffuse and direct irradiance values for localised climatic conditions. By closely examining the test room rendered with these models, the differences and limitations of each of the sky distributions can be easily seen.

7.2.1.1 *Clear skies*

The first sky type examined was the clear sky type. That is, no cloud cover and thus a direct solar component along with a diffuse sky component. The following section shows these sky domes, as images previously shown in chapter four along with images of the accompanying test room. The test room consists of a 4m x 4m x 3m room with a ceiling reflectance of 80%, wall reflectances of 60% and a floor reflectance of 20%. The daylight models were set up to render on the 4th of March at 11:00 am at Perth, Western Australia with a clear sky turbidity factor of 2.75. The window is sized at 2m by 2m and faces south so there is no direct sun light entering the room, thus the daylight sky distribution can easily be compared. The piano is included to add a sense of scale to the room.

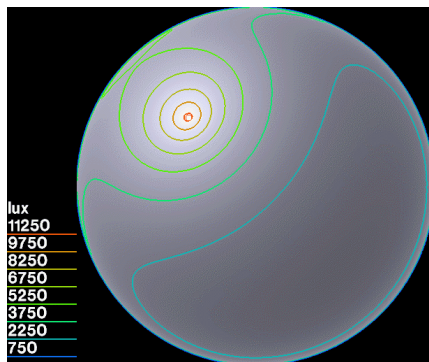


Image 7-1 CIE clear sky

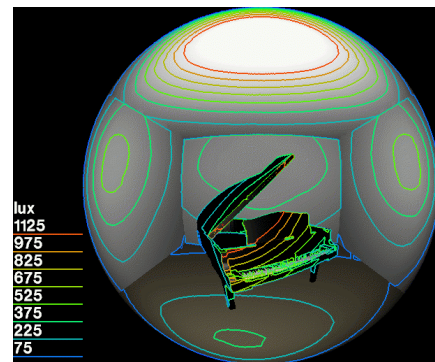


Image 7-2 CIE clear sky test room

The standard CIE sky (Image 7-1) distribution shows a small bright area of only 11250 lux surrounding the sun which fades away to a relatively dark sky dome of less than 2250 lux. As a result, the test room (Image 7-2) shows a dark floor and rear wall as there is only a small contribution from the sky dome itself, but with a bright ceiling that has resulted from the reflected ground component of the small concentrated bright sun source. This does not correspond well to a sunny Perth March day as the sky component is typically a lot brighter.

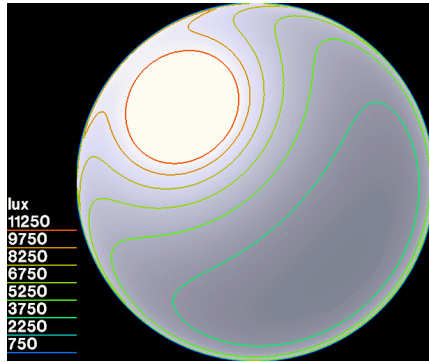


Image 7-3 CIE clear sky - measured data

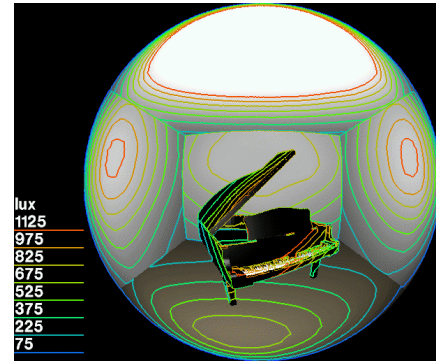


Image 7-4 CIE clear sky measured data - test room

A slightly more realistic result is achieved by providing actual measured sky irradiance values to the CIE sky model as shown in Image 7-3. However this CIE model tends to generate a sky dome that is too bright, especially around the horizon in the direction of the sun. The test room (Image 7-4) shows this as much more light is reaching the rear, side walls and floor, with still a strong contribution from the reflected ground component.

There is almost a discrepancy of 60% between the two CIE sky model domes (calculated from the raw data as presented in appendix A and B) as well as the generation of particularly dark horizons that do not represent the bright horizons often existing in bright clean sky locations. This discrepancy can be put down to the gensky program using a formula developed from data gathered in San Francisco to calculate the zenith brightness. The San Francisco model does not transfer well to other locations on the globe.

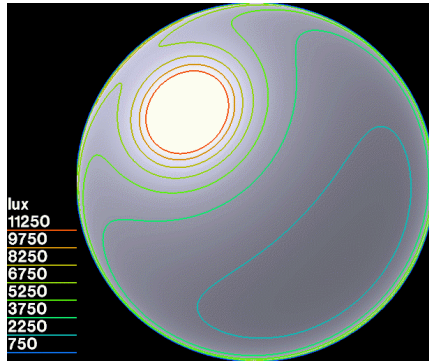


Image 7-5 Perez Sky

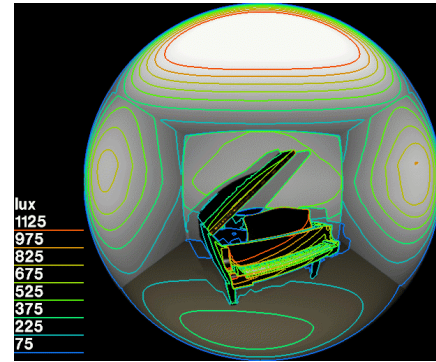


Image 7-6 Perez Sky - test room

The Perez sky distribution model (Image 7-5), using the same measured data as used in CIE model (Image 7-3), shows a much more realistic simulation of the sky dome. This is due to the superior Perez sky modeling technique as described back in Chapter 4. In this simulation, the horizons are brighter, as shown by the larger spacing of the contour lines at the edges of the sky dome; the corona or area surrounding the sun is well defined and the rest of the sky dome is consistent to typical clear sky conditions. This is also demonstrated in the test room (Image 7-6) as there is more light hitting the rear wall which is originating from the sky dome horizon. This is the preferred sky distribution model to be used in the RADIANCE lighting platform. The drawback to the Perez model is that measured data must always be used. If measured data for a particular site for a specific atmospheric condition is not available, then the genrad program is required. The genrad program is used to calculate instantaneous direct and diffuse irradiance values for a specific time and site. It then uses these values to automatically run either the CIE or Perez sky distribution programs to produce a RADIANCE description of the sky dome.

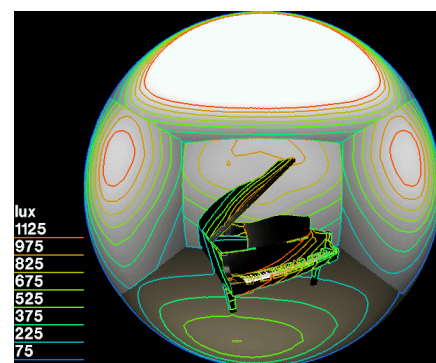
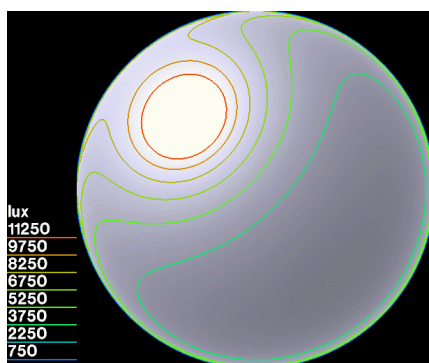


Image 7-7 Genrad CIE sky

Image 7-8 Genrad CIE - test room

Image 7-7 shows the sky dome of the CIE distribution of the genrad calculated irradiance values. It can be seen that the resulting sky dome distribution more closely corresponds to the Perez distribution (Image 7-5) than the CIE measured distribution (Image 7-3). It does, however, still suffer from the over brightness or increased magnitude problem inherent in the CIE sky model program. This is demonstrated by the high lux levels shown in the test room walls (Image 7-8).

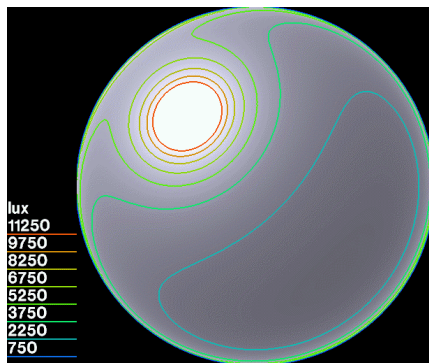


Image 7-9 Genrad Perez sky

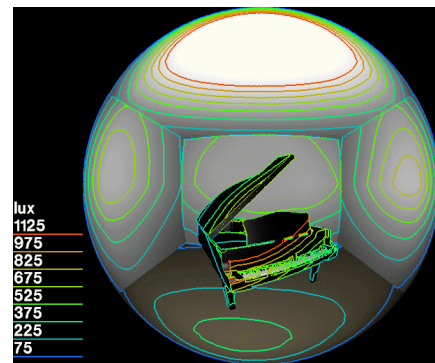


Image 7-10 Genrad Perez - test room

These problems disappear when the genrad program is used with the Perez sky distribution. As shown in Image 7-9, a more realistic sky distribution is produced with bright horizons, a well defined and modelled corona, and a consistent sky dome for typical clear sky conditions. It can be seen, however, in the test room of the Perez genrad model (Image 7-10), that the rear wall is slightly less illuminated than the Perez model alone (Image 7-6). This is due to the values given to the Perez sky distribution which were obtained from average monthly irradiance values and thus do not represent the instantaneous conditions of the local atmosphere. genrad tends to produce a slightly more realistic representation of the sky distribution as it is generating instantaneous irradiance values for the given site and atmospheric conditions.

7.2.1.2 Overcast skies

The second sky type examined was the overcast sky type. This consists of a fully overcast sky with no direct solar component, only a diffuse component. Similar to the

above examples, the following section shows these sky domes, as images previously shown in chapter 4, along with images of the accompanying test room. The test room has remained the same. The daylight models were set up to render on the 4th of June at 11:00 am at Perth, Western Australia with an overcast sky turbidity factor of -4.5. The month of June was chosen as this is the month with the maximum amount of overcast days for Perth (see Figure 7-1) and as the diffuse irradiance data that is given to the CIE and Perez models was, again, obtained from average monthly records (Spencer 1976).

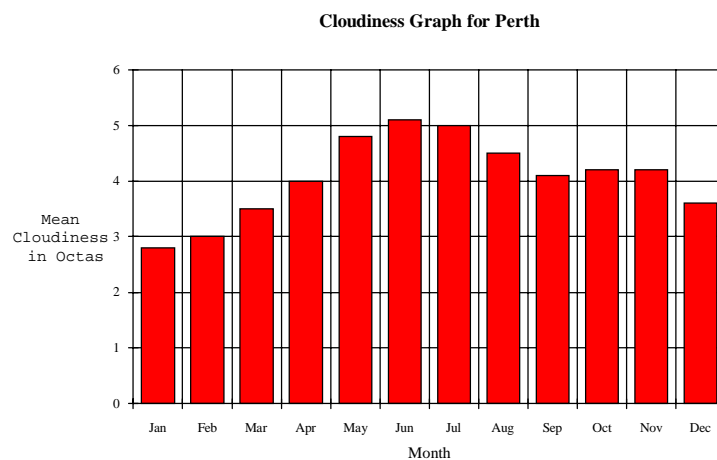


Figure 7-1 Cloudiness graph for Perth (Kendrick 1980)

This results in more realistic sky distribution with which to compare to the instantaneous diffuse irradiance values generated from the genrad program.

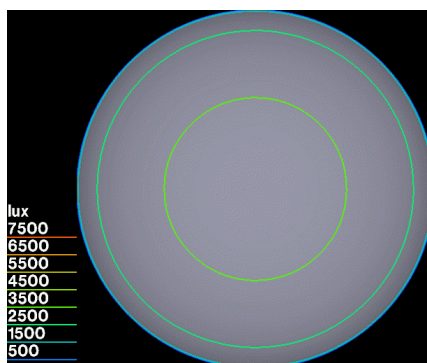


Image 7-11 CIE overcast sky

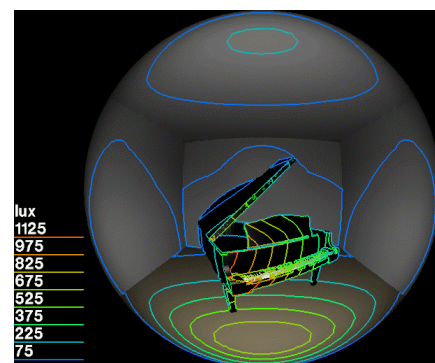


Image 7-12 CIE overcast sky - test room

The standard CIE overcast sky (Image 7-11) shows the distribution of diffuse daylight over the sky dome. It can be seen that the distribution is symmetrical about the zenith

with the horizon being one third of the intensity of the zenith (<4500 lux). This distribution results in a dark test room (Image 7-12) with only a small amount of light hitting the floor near the window and very little daylight penetration into the room. As such, the CIE dark sky dome corresponds to a worst possible case scenario.

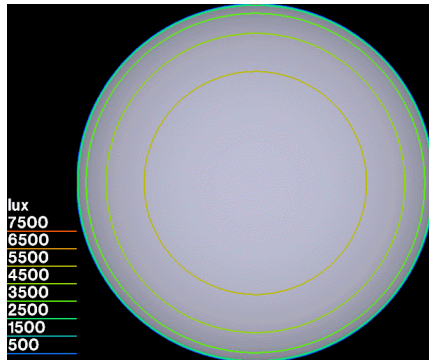


Image 7-13 CIE overcast sky - measured data

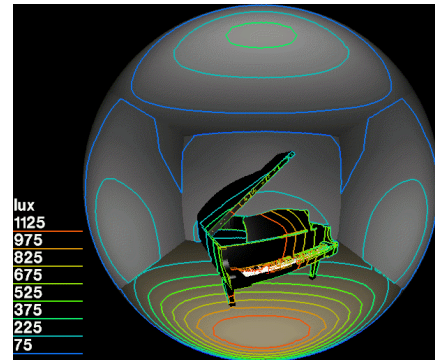


Image 7-14 CIE overcast sky measured data - test room

A slightly more realistic local simulation is produced by using measured diffuse irradiance data in the CIE model as shown in Image 7-13. Here the sky dome is much brighter and still in a symmetrical distribution. The increase in sky illuminance is reflected in the test room (Image 7-14) with a brighter floor patch near the window and limited light penetration into the room. Here the model simulates quite a bright overcast sky that is too intense if the lighting designer is requiring to model minimum light levels.

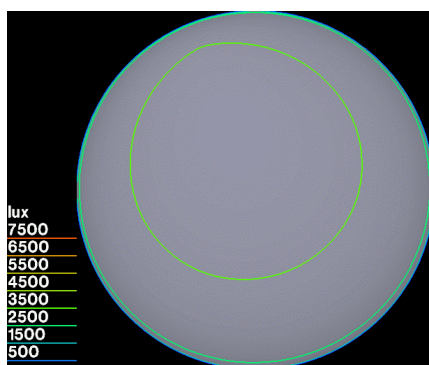


Image 7-15 Perez overcast sky

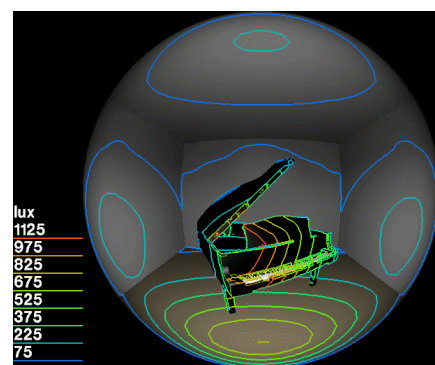


Image 7-16 Perez overcast sky - test room

The Perez overcast sky distribution model (Image 7-15), using the same measured data as used in CIE model (Image 7-13), shows a much more realistic simulation of the sky dome. The most obvious difference is the asymmetrical nature of the sky light

distribution, leading to more accurate lighting analysis as the orientation of the window towards the sky dome now plays a part in the design. A slightly more subtle difference is the brighter horizon. This change is clearly shown in the test room (Image 7-16) as more light is penetrating into the room and onto the side walls. Once again the Perez distribution is the preferred sky distribution model to be used in the RADIANCE lighting platform as it most accurately models the sky dome with measured data. However when measured data is not available, the `genrad` program can be utilised to calculate the sky diffuse irradiance based on the local sky conditions.

When dealing with overcast skies, the `genrad` program is used to calculate only the instantaneous diffuse irradiance values for a specific time and site. It then passes this value to either the CIE or Perez sky distribution programs which, in turn, produce a RADIANCE description of the sky dome.

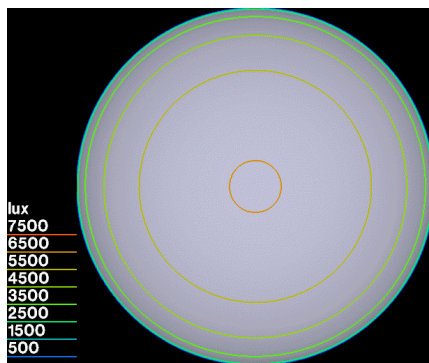


Image 7-17 Genrad CIE overcast sky

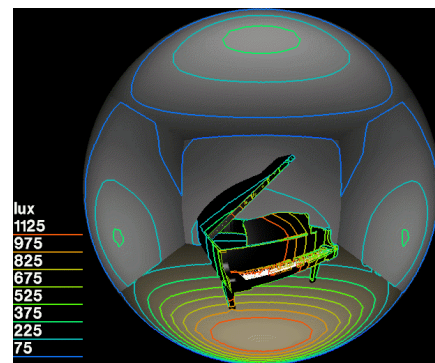


Image 7-18 Genrad CIE overcast sky - test room

Image 7-17 shows the sky dome of the CIE distribution using the `genrad` calculated diffuse irradiance value. It can be seen that the resulting sky dome distribution is typical of the CIE measured distribution (Image 7-13), that is, a very bright azimuth within a symmetrical distribution. The calculated azimuth is, in fact, brighter than the CIE overcast sky using measured data. This is because the measured diffuse irradiance data was 96 W/m^2 as opposed to 97.5 W/m^2 as calculated by the `genrad` program. 1.5 W/m^2 is enough of a difference to produce the brighter sky zenith illuminance. In any case, the resulting test room still suffers from the over brightness problem inherent in the RADIANCE CIE sky model program. The use of this sky distribution,

corresponds to the relatively high lux levels visible on the test room floor beneath the window (Image 7-18).

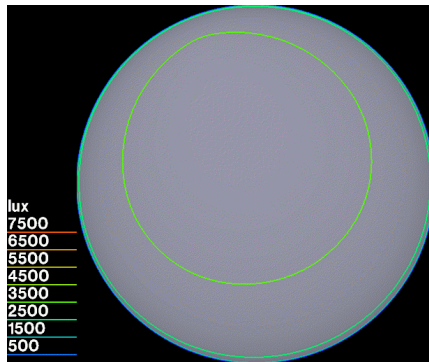


Image 7-19 Genrad Perez overcast sky

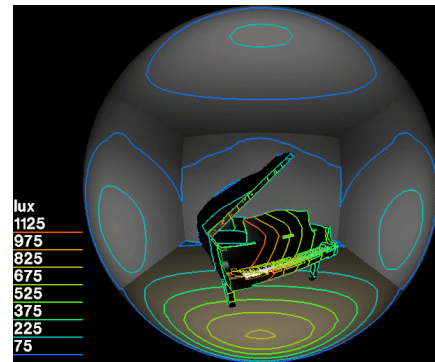


Image 7-20 Genrad Perez overcast sky - test room

When the genrad program is used with the Perez sky distribution, as shown in Image 7-19, a more realistic sky distribution is produced with brighter horizons, an asymmetrical distribution and an appropriate zenith brightness. The test room (Image 7-20) reveals slightly brighter walls, resulting from the brighter horizons and a slightly brighter floor region than the Perez test room (Image 7-16). These more appropriate values are advantageous for the lighting designer working on a design for the worst case scenario, as the designer is not needing to specify additional lighting to compensate for the extreme CIE overcast sky that may never occur at their particular site. Thus the use of this sky model reduces the danger of over-designing artificial lighting to correct for the overcast skies. This an example of how using the genrad program to generate instantaneous irradiance values for a given site and atmospheric conditions, can produce a more accurate representation of the sky distribution.

7.2.2 Validation - calculated vs. measured

Part two of the validation consists of determining the validity of the genrad day lighting output. It does this by comparing the output of the genrad program with measured data from a specific site. The data consists of average monthly direct and diffuse irradiance values for each sunlight hour of the day. The data comes from a Building Management Authority of Western Australia daylight utilization study (Gabriels 1986), originally collected by Spencer (1976) and is reproduced in appendix

B. In order to compare these average monthly values with the instantaneous values produced by the `genrad` program, a modified version of the `genrad` program was written. The modifications consisted of a series of incremental loops used to calculate, total and then average, hourly irradiance values. The modified `genrad` program, when run, produces average monthly direct horizontal and indirect horizontal irradiance data for each sunshine hour of the day for any specified month. These average hourly values were then able to be compared with actual measured data. Additionally, a modified version of the `RADIANCE gensky` program was also written, once again with incremental loops, to calculate average hourly direct horizontal and indirect horizontal irradiance values for any specified month, using the `RADIANCE CIE` sky model. These calculated values have been charted and are shown in full in appendix A. Figure 7-2 to Figure 7-5 show the direct solar irradiance graph for December and the diffuse solar irradiance graph for June. Both 2D and 3D charts have been reproduced demonstrating the results.

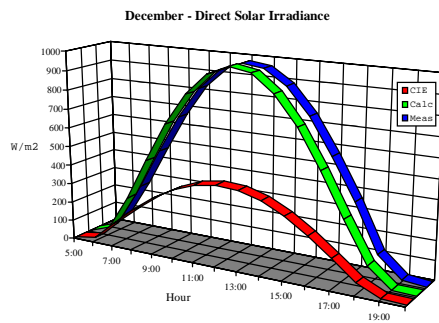


Figure 7-2 December direct solar radiation

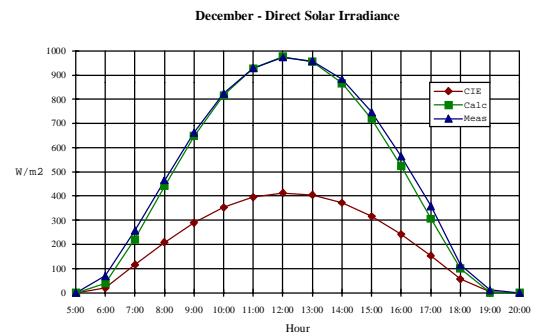


Figure 7-3 December direct solar radiation

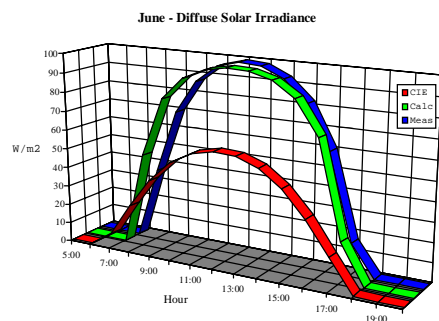


Figure 7-4 June diffuse solar radiation

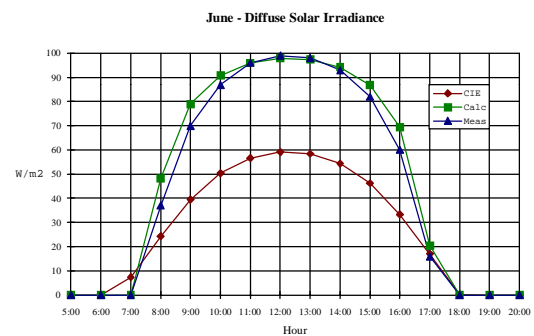


Figure 7-5 June diffuse solar radiation

It can be seen from the above graphics that the genrad calculated results are quite close to the measured data. However it must be noted that no average monthly turbidity factors have been used in the production of these values, as the turbidity data was unavailable for the specific site. A turbidity value of 3.1 was used throughout the test runs as this represented a typical patchy light cloud sky for Perth's temperate climate (Table 3-1). Also, the modified programs do not take into account relative sunshine hours, and as such are only calculating average clear sky values. Because of this, the winter direct irradiance values calculated diverge from the measured data. The genrad results as demonstrated, are more accurate than the CIE standard clear sky values when compared to the measured data. It must be noted however that discrepancies will occur as the genrad calculation is an instantaneous calculation for specific atmospheric conditions (turbidity values with or without the sun shining) and can vary greatly from the average monthly measured data. Without specific instantaneous data available, validation with average measured data is sufficient.

7.3 Daylight results

It can be seen from both the test room renderings and comparison with measured data that the *genrad* program is able to produce more accurate RADIANCE sky distribution models than the RADIANCE *gensky* program when local radiation data is limited. The most accurate sky representation would occur using actual instantaneous measured data, but as this data may not be readily available to the lighting designer, (especially when designing in non-standard locations), the *genrad* program offers a good sky simulation replacement. It is now much easier to specify and design for specific localised Australian daylighting conditions, than having to rely on the typical CIE worst case scenarios. The added benefit of the *genrad* program is that it has been seamlessly integrated into the RADIANCE lighting platform. This is demonstrated by the complete working example of the lighting design platform in the next section.

7.4 Platform process example

The following section documents, through a working example, the process of using the lighting design platform to produce more realistic computer simulations of the lighting design. It demonstrates the realisation of the aim of this thesis, to provide a working model of a specific physical based lighting design platform for localised Australian daylighting conditions. The example shows, step by step, the way in which the new additional programs integrated into the REAGLE rendering system are used to provide better predictive capabilities, and to produce more realistic computer simulated images.

7.4.1 The problem

The model example chosen was an un-built design of an office reception area with meeting room as shown in perspective in Image 7-21, and as shown with the sections and plan included in Appendix E. The lighting design consists of a natural skylight running across the rear wall, with general direct and diffuse fluorescent luminaires lighting a meeting area and the general reception area, and a number of small spot

lights highlighting wall pictures and the main reception desk. There are a number of design details that make the traditional simulation of this space difficult.

The first problem is with the two white dropped ceiling planes, one above the reception area and one making up the ceiling of the meeting room. These multi level planes leave the area above them, containing the roof trusses, to be totally lit with diffuse or ambient light that most traditional simulation techniques would fail to evaluate.

The second problem is the natural skylight above a very brightly coloured wall. The effects of the strong daylight falling on this wall, and the resultant colour bleeding into the scene, is also a difficult concept with which to traditionally visualise.

The third problematic area is the clear glass and semi opaque glass encasing the meeting room. Lighting from both inside the meeting room area and outside, in the reception area, will have an impact upon the quality of the lighting in the final design.

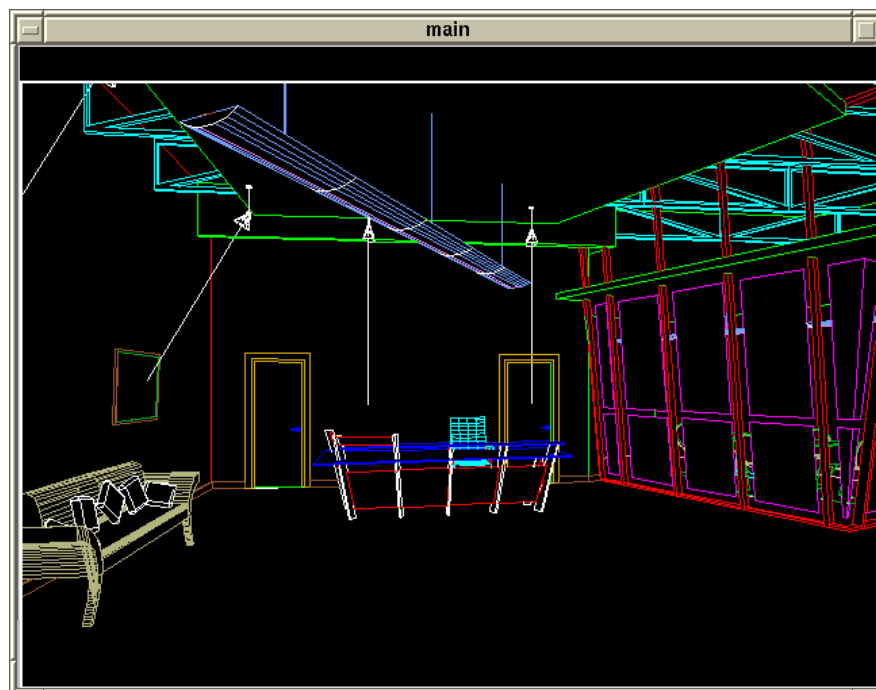


Image 7-21 Office reception model used for the lighting design platform example.

7.4.2 The materials

The first step in using the new REAGLE system to produce a rendered image of the above scene is to assign materials to the surfaces created in the model. This assignment of the materials does not need to happen after the final model is created, but can occur while it is being constructed. Materials are allocated from the same common material library. Image 7-22 shows the assigning of “crinkly frosty glass” to the currently selected glass panels of the meeting room in the example model.

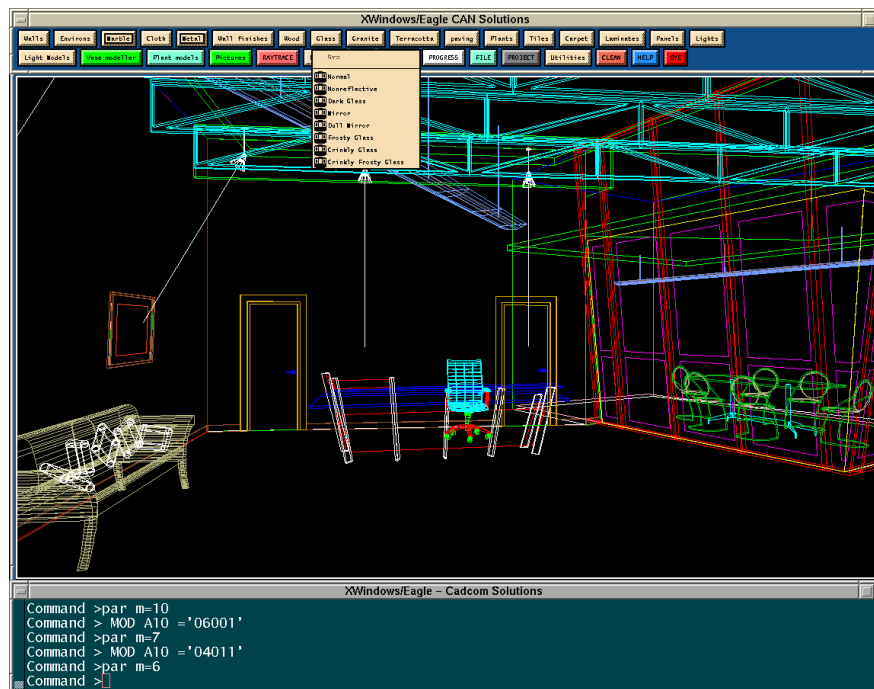


Image 7-22 REAGLE system being used to assign materials to the example model.

The “crinkly frosty glass” material is one example of where an addition to the RADIANCE program was necessary to be able to create materials that are common to both libraries (see section 5.3.1). The RAYTRACE program uses a bump map to create the crinkles on the surface of the glass (a sample is shown in Image 7-23). Modifications to the RADIANCE rendering program enabled this bump map to be used in the definition of the RADIANCE “crinkly frosty glass” material.

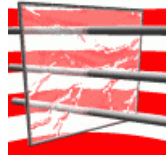


Image 7-23 “Crinkly frosty glass” material sample using bump map.

Another example of an addition to the RADIANCE rendering software was the ability to automatically orientate materials directly onto surfaces correctly. This ensures that the brick material assigned to the left side wall runs horizontally and not, say, vertically. This automatic material transformation applies to all of the materials in the RADIANCE material library. Once the materials have been assigned, the next step is to add lights to the model.

7.4.3 The lighting

Here is the point where the user must separately model two different lighting models if they wish to render their model using both the RAYTRACE and RADIANCE renderers, which is the case in this example. The need for two separate models is due to the fact that the RAYTRACE lights are not capable of physically simulating a real world light source, where as the RADIANCE light are, and there is no sensible way of converting between the two rendering systems. Image 7-21 shows the reception scene with the RADIANCE lights already set up.

The RADIANCE spot lights chosen model a MR-16 50W halogen low voltage lamp with a 30 degree beam spread. They were created by inserting a pre-defined MR-16 light source EAGLE model (a simple sphere), from the RADIANCE standard lighting library, into the spot light model created by the user. The line leading away from the centre of the spot light is used to check the direction that the spot light is facing.

The overhead strip lighting was created using the pre-defined polygon light source, once again from the RADIANCE standard lighting library, that has the distribution of a STAFF “TRIAS”, 2x36W downward direct and 1x36W upward indirect fluorescent light source.

The skylight itself, consists of a number of polygons that have been assigned a special glass material. This material, that is specific to RADIANCE, allows the glass to act as a secondary light source when illuminated from daylight and is how the natural daylight produced by the genrad program is utilised in the scene. In fact any exterior glass in an EAGLE model should be assigned this special glass material. In the example simulation an overcast sky is used and thus no direct sun source will be produced. It is possible to assign the skylight polygons as normal glass, but the result of the lighting component from the exterior daylight would be reduced and not as accurate. When this material is used in a RAYTRACE render it defaults to normal reflective glass.

The lighting setup needed for the RAYTRACE render is very different, since the REAGLE / RAYTRACE process does not allow for polygonal light sources. Thus in order to specify a light source for the tubular lighting, a wide RAYTRACE spotlight model (beam spread of 165 degrees) is used for each fluorescent up light and down light. These are shown in Image 7-24, as each of the spotlights, imported from the RAYTRACE standard lighting library, have lines that show the orientation and beam angle of the spot light source.

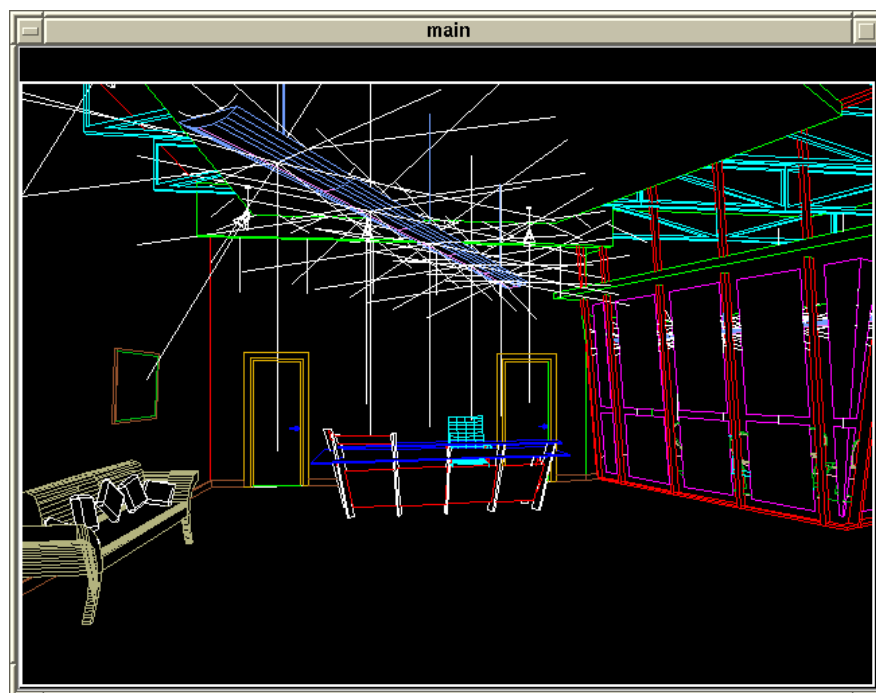


Image 7-24 Example model showing RAYTRACE light sources and orientation.

RAYTRACE is unable to simulate daylight penetration into the scene through the skylight as it lacks the ability to model daylight. The RAYTRACE program can generate a direct sun source, but as an overcast day is being modelled, this will not be used. So, in order to simulate the sky light, it is necessary to place a wide angle RAYTRACE spotlight source in the centre of each polygon that makes up the skylight. These can just be seen in Image 7-24 as the vertical lines from the spotlights pointing down the real wall. Normal RAYTRACE narrow spotlights (beam angle of 30 degrees) have been used to highlight the picture and the reception desk. Once the model has been saved it is ready to be converted and a final image produced.

7.4.4 The options

The initial process for starting a REAGLE render is the same whether producing either a RADIANCE picture or a RAYTRACE picture. That is, the REAGLE “*mog->asc*” button is used to recursively expand the EAGLE model and sub models that it may contain, to produce a single ASCII file containing all the model geometry and material labels. Once this is done the user can then either choose to click on the REAGLE “*raytrace*” button or the REAGLE “*radiance*” button to begin the option setup process. The RAYTRACE rendered example is examined here first.

7.4.4.1 RAYTRACE options

In this example, the RAYTRACE options are only examined briefly as they are not the focus of this thesis and are described elsewhere (Marriott 1994). By clicking on the “*raytrace*” button the `xconvert` program is executed. This allows the user to set all the options for the RAYTRACE image. Image 7-25 shows the options that have been set up for the RAYTRACE rendering of the example model. This default values for the “*misc*”, “*background*” and “*debug*” buttons have been accepted with no change. The final image resolution is selected by clicking on the “*slide*” button, thus automatically setting the XRES and YRES to 1032 and 689 pixels respectively. The new viewing vista is set to 75 degrees and the number of specular reflections (REFL) to 1 (one).

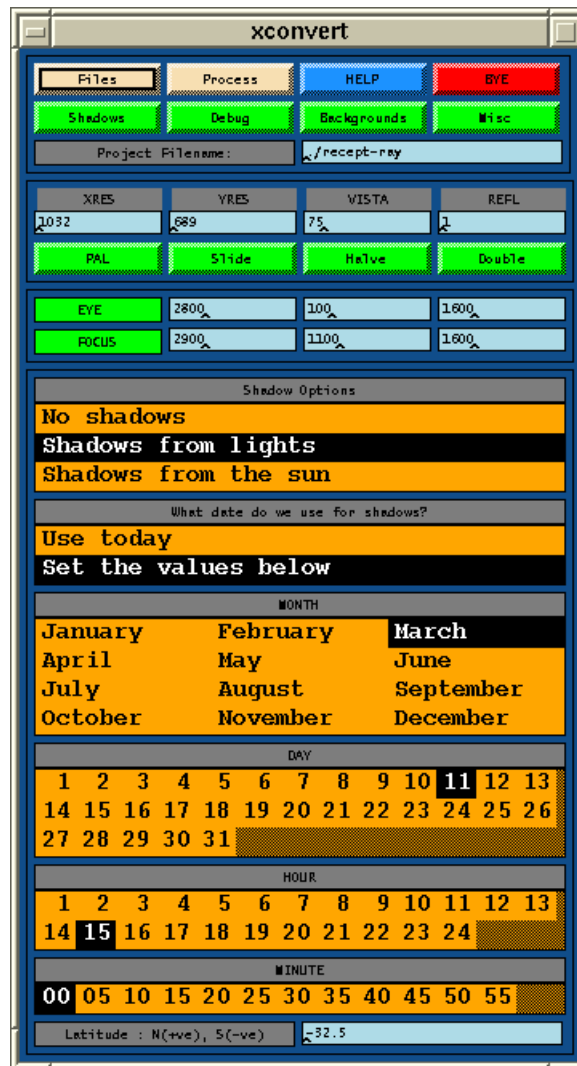


Image 7-25 Xconvert options for the RAYTRACE reception model

The EYE and FOCUS points are obtained from the perspective information given by the EAGLE program. Finally the shadow options are set up. Here the shadow options have been set to *shadows from lights*, as the RAYTRACE image is to be as realistic as possible and thus needs to cast shadows. By setting the shadow option to *no shadows*, a picture is still generated with light sources but none of those sources would cast shadows, and consequently vastly decreases the rendering time. The shadow option has not been set to *shadows from sun* as no direct solar source is desired, as the scene is to be rendered in overcast conditions. This also means that the date and time values set for calculating the position of the sun source are ignored.

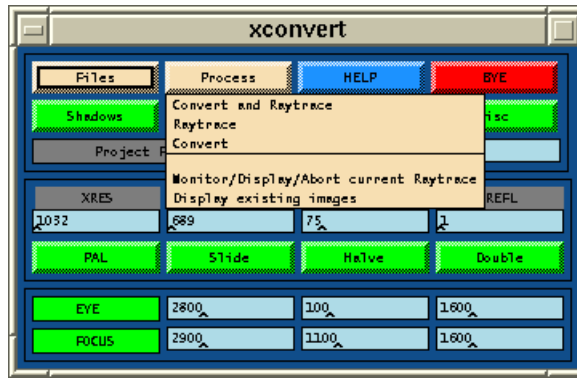


Image 7-26 Xconvert Process - Convert and Raytrace

Once these options have been saved, the “*Convert and Raytrace*” option of the “*Process*” menu button (Image 7-26) can be used to execute the `xeagle2rin` program that starts the automatic geometry conversion, material creation and final RAYTRACE rendering process. The results of this RAYTRACE render shall be shown and discussed later in this chapter.

7.4.4.2 RADIANCE options

The process of setting up the appropriate RADIANCE options for the example model render is begun by clicking on the REAGLE “*Radiance*” button, thus executing the newly developed `xradconvert` program. This is the graphical user interface program that allows the user to set all the options and control the rendering of the RADIANCE image.

The first step, once the `xradconvert` program is open, is to load in the default set of options. This is done by selecting the *read options* part of the “*Files*” pull down menu, which opens the file selection window, revealing a “*default*” button down at the bottom of the window. By clicking on this button the default values will be automatically loaded. It is then up to the user to go through the options and change the values according to their particular needs. In the example RADIANCE render, the following options were changed.



Image 7-27 Xradconvert - misc options for RADIANCE example

In the misc window (Image 7-27), opened by clicking on the “Misc” button, the *Interior* Zone must be selected, as the render is of a fully enclosed scene, and the bounding box information inserted. This bounding box information is used by the RADIANCE programs to control the automatic calculation of various accuracy settings. It can be obtained directly from the EAGLE program. The *Radiance options* and *Conversion options* can be left as is.



Image 7-28 Xradconvert - View options for RADIANCE example

In the view window (Image 7-28), opened by clicking on the “View” button, the selection of the *Pers* viewing type results in a normal perspective view of the scene. By setting the X-Vista to 75 and Y-vista to 53, a horizontal viewing angle of 75

degrees and a vertical viewing angle of 53 degrees will result. As with the `xconvert` program the viewing **EYE** and **FOCUS** points can be set by obtaining the information from the **EAGLE** program.

Although it is now possible to set up a background picture in a **RADIANCE** image, it is not necessary here as there is no outlook into the outside world in our example. Thus the “*backgrounds*” button can be left untouched and as the default background is a plain matte grey colour.



Image 7-29 *Xradconvert - Detail options for RADIANCE example*

The options that need to be set in the detail window, opened by clicking on the “*Detail*” button, are important as they control the overall quality and rendering time taken to create the **RADIANCE** image as described in chapter 6. As the example scene is quite geometrically detailed, the **Detail** setting has been set to *Medium*. The **Variability** setting has been left on *Low* as there are no direct patches of sunshine in the scene. Hard shadows are wanted, thus the **Soft Shadows** setting is left as *False*. Finally the **Quality** setting has been set on *Medium* as a good quality image is required, with an indirect diffuse calculation necessary to fully resolve the lighting installation. At this point in the process, while the issues of quality are being addressed, the image resolution and indirect settings in the main **xradconvert** window should be set. As a final image is to be produced, clicking on the “*slide*” button results in an **XRES** setting of 1032 and a **YRES** setting of 689. Since a diffuse calculation is required the **Indirect** value is set to 1 (one).

xradconvert

Files Process HELP **BYE**

Daylight Detail Backgrounds Misc

Project Filename: /default

XRES: 1032 YRES: 689 Exposure: 1 Indirect: 1

Slide Halve Double View

Use Daylight? **Yes** No

Sun shining? **Yes** No

Use Today's time & date? **Yes** No

Sky distribution option

Clear Inter Overcast **Perez**

Latitude: -31.95 Elevation: .06

Longitude: -115.85 Turbidity: -4.5

Std Meridian: -120

MONTH

January February March

April May **June**

July August September

October November December

DAY

1 2 3 **4** 5 6 7 8 9 10 11 12

13 14 15 16 17 18 19 20 21 22 23 24

25 26 27 28 29 30 31

HOUR

1 2 3 4 5 6 7 8 9 10 **11** 12

13 14 15 16 17 18 19 20 21 22 23 24

MINUTE

00 05 10 15 20 25 30 35 40 45 50 55

Image 7-30 Xradconvert - Daylight options for RADIANCE example

The last options to set in the xradconvert program are the daylight options, which are accessed by clicking on “*Daylight*” button as shown in Image 7-30. These options control the amount and quality of light that flows through the skylight in the example model. As stated earlier, an overcast daylight model is required. This results in the Use daylight option to set to *Yes* and the Sun shining option to be set to *No*. For this example the simulation is to be run on the 4th of June at 11:00 AM and as such the Use today's date and time option is set to *No* and the specified date and time are set. The sky distribution type is set to *Perez* which will result in genrad calculated diffuse irradiance data being passed onto the gendaylit sky distribution program. This will produce the most accurate daylight description to be used by the example model skylight for the specified site. This site information is the last of the

options left to set. The site location is Perth, Western Australia and a turbidity factor of -4.5 is used.



Image 7-31 Xradconvert - Save options for RADIANCE example

The final step, before rendering can begin, is to save the RADIANCE options. This is done by selecting the **save options** part of the “*Files*” pull down menu, which opens the file selection window as shown in Image 7-31. In the example, the new file name *recept-rad* is entered, the “*Ok*” button pressed, and all the RADIANCE option parameters are saved. The Project Filename field in the main xradconvert window is then updated to reflect the new model ready to be rendered.

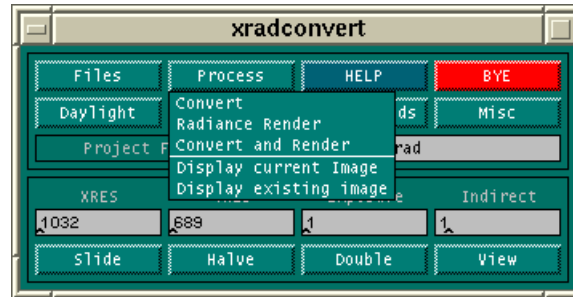


Image 7-32 Xradconvert Process - Convert and Render

Once these options have been saved, the **Convert and Render** option of the “*Process*” menu button (Image 7-32) can be used to execute the new *xeagle2rad* program. This program controls the automatic geometry conversion, material creation and final RADIANCE rendering process.

The final results of this RADIANCE render, which consists of an image of the example model, are compared to the final RAYTRACE rendered image in the next section.

7.4.5 The solutions

The final result of the RAYTRACE rendered example scene is shown in Image 7-33. Although an impressive dramatic image, it lacks realism in certain areas when compared to the final RADIANCE rendered example model as shown in Image 7-34.

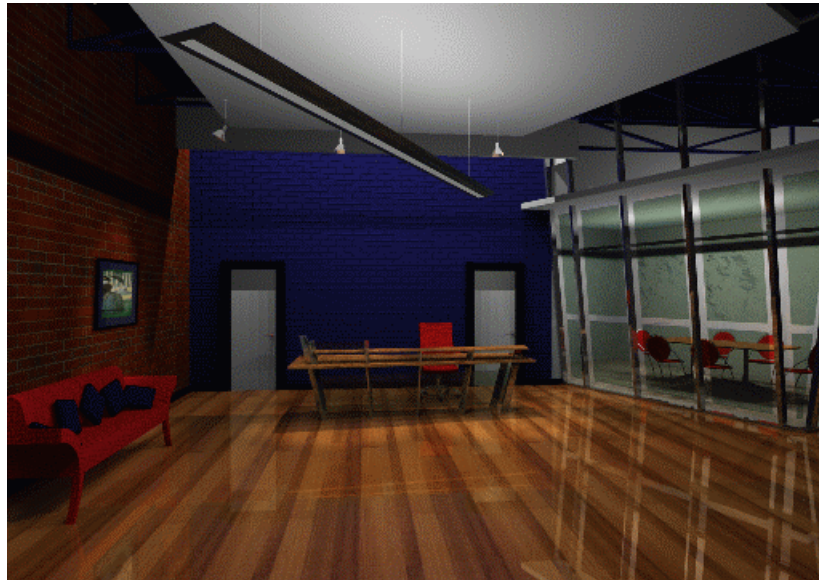


Image 7-33 RAYTRACE rendering of the example model

This initial difference is in the overall brightness of the scene. Since the RADIANCE render is able to compute the diffuse reflected component of the lighting, the image is much more evenly lit. This is particularly the case with the lighting of the brick wall, where the RAYTRACE brick wall appears very dark as there is no diffuse light being reflected from the white reception ceiling plane. However in the RADIANCE image the wall is very evenly lit. Here is clear evidence that the RAYTRACE renderer cannot be used to evaluate lighting designs as is the reason behind the integration of the RADIANCE software into the REAGLE system.



Image 7-34 RADIANCE rendering of the example model

Another problem in the RAYTRACE image is the dark band that crosses the rear coloured wall, as well as being visible in the meeting room. The band is caused by the cut off angle of the RAYTRACE spotlights that are being used to model the strip lights. The band is forming because no light is being projected sideways from the strip lights (the RAYTRACE spotlights). The strip lights themselves also appear black as there is no light shining onto them. In the case of the RADIANCE image, indirect diffuse light reaches the surfaces of the strip light and enables the surface of the strip lights to be seen, even though no direct light is shining onto them. This diffuse light is coloured, depending upon the surface that it has been reflected from, and as such the strip lights appear slightly yellow (the colour of the floor).

A more subtle difference between the two images is the daylight filtering in through the skylight is much more pronounced in the RADIANCE image and does not result in the hard shadow lines across the rear doors as it does in the RAYTRACE image. The lack of shadow line in the RADIANCE image is due to the RADIANCE skylight is acting as a diffuse secondary light source, as opposed to the direct spotlights being used to simulate the daylight in the RAYTRACE image.

A common aspect of the two images is the appearance of the surface materials. Although rendered with two different packages they appear very similar due to the two RADIANCE and RAYTRACE rendering material libraries mirroring each other.

There are some small differences, mainly due to the scaling of the materials themselves. For example, the size of the wooden floor boards differs slightly, as well as the roughness of the “frosty crinkly glass” material. This is just a small maintenance issue that appeared in the creation of the RADIANCE material library and can be easily remedied.

7.5 Platform results

It can be seen from the example model that it is now possible for building designers to take EAGLE 3D models and produce either a RAYTRACE rendered image, that can be used for initial presentation purposes, or a RADIANCE rendered image, that can be used to evaluate a realistic lighting design concept, as well as presentations. There is still the need for the RAYTRACE render as it is faster than the RADIANCE render and useful for more conceptual presentations rather than photo realistic building simulations. For example the RAYTRACE image took just over 4 hours to render on a Sun SparcStation while the RADIANCE image took just over 12 hours. With the ability to now simulate and evaluate daylighting added to the REAGLE platform, there exists a tool that offers better predictive capabilities and more realistic images to building designers and their clients.



Image 7-35 Interior Design image produced with the RADIANCE / REAGLE platform

This new platform has been tested by architecture and interior design students at the School of Architecture, Construction and Planning at Curtin University. The new `xradconvert` interface has been shown to be easily used by students, especially those who were familiar with the old REAGLE system. As a result, the quality and realism of imaging has increased. This is clearly demonstrated through a comparison of student work using the new RADIANCE / REAGLE system (Image 7-35) and students work generated with the existing RAYTRACE / REAGLE system (Image 7-36).

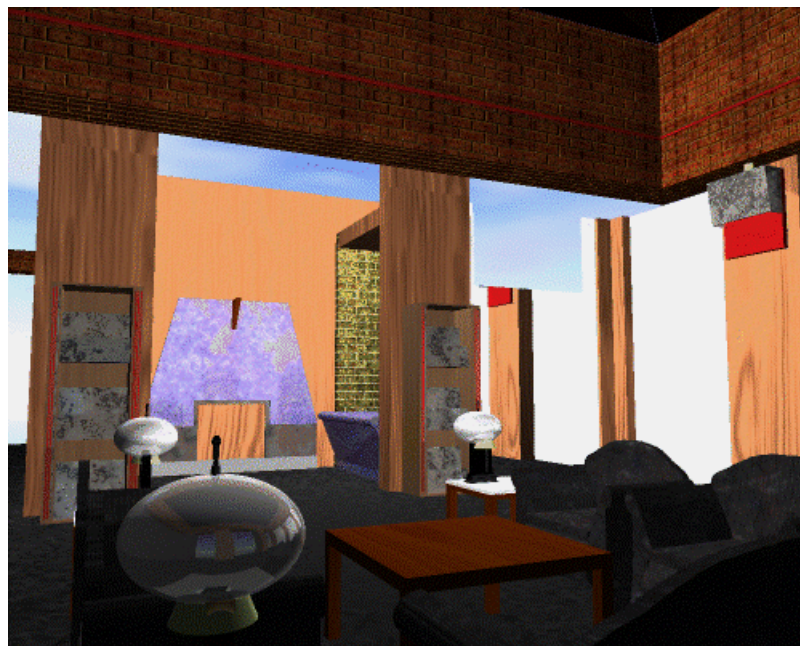


Image 7-36 Student work produced using RAYTRACE before access to RADIANCE

A note of caution is needed here. The new RADIANCE / REAGLE system is not going to produce superb quality images straight away for a first time user. There is still a learning curve required, to be able to understand and choose the appropriate rendering options to maximise rendering quality with speed. The options and controls of the RADIANCE rendering process can only be simplified to a certain extent. However this learning curve is small for students who are comfortable with the existing RAYTRACE / REAGLE system.

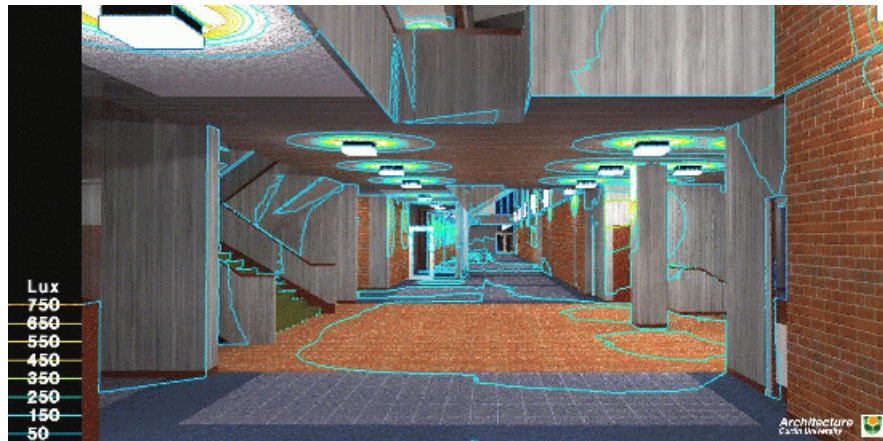


Image 7-37 RADIANCE render of main exhibition space showing isolux lines.

The quantitative aspects of the lighting platform have also been tested by the modelling of the main exhibition floor of the Architecture Building at Curtin University (Image 7-37). Here RADIANCE software tools were used to add illuminance isolux lines to the final RADIANCE rendered image. A calibrated light meter was then used to measure the actual lighting levels in the building. The results were very close to the RADIANCE calculated values. For example the brick floor area directly below the lights opposite the stairs (to the right in the image) was measured to receive around 400 lux which is well within the range that the RADIANCE render calculated (between 350 and 450 lux).

The above images demonstrate the success of the lighting design platform in terms of its ability to be used by students to produce accurate visual evaluations of their building designs. Even more images that show evidence of this quality can be found in appendix E.

7.6 Summary

This chapter has discussed the success and validity of the developed daylight model program and RADIANCE integration programs that aim to extend the capabilities of the REAGLE system into a lighting design platform.

It has shown that the new genrad program provides a better predictive representation of daylight for specific local climatic conditions. It has done this by comparisons with the existing RADIANCE daylight models as well as being verified

with measured data. The use of the new `genrad` program, through its refined input into the existing RADIANCE daylight models, enables building designers to develop integrated natural and artificial lighting solutions using more appropriate models than the traditional CIE worst case scenario skies. The program has been integrated into the lighting design platform for ease of use.

This chapter has also shown, through an illustrated example, how the integration via a graphical user interface of the underlying RADIANCE CAD translators, material libraries and conversion programs are utilized in the production of a rendered image. The result of which is a platform that designers can use to produce more realistic images and simulations, that themselves, become evaluation tools rather than just presentation pictures.

The next chapter draws conclusions along these lines and also presents possibilities of areas for future work.

8. Conclusions

8.1 Thesis summary

The aim of this thesis was to discuss, formulate and create a working model of a physical based lighting design platform that enables designers to easily evaluate lighting and daylighting designs based on specific Australian site, climatic and atmospheric conditions. This evaluation platform consists of a set of computer software tools that ultimately produce a photo-realistic computer generated image. It is this image portraying the accurate visualisation of physical lighting that has a direct practical application to designers in enhancing their ability to control light and improve the quality of their building designs.

The creation of this platform has been achieved, through the integration of the physical based RADIANCE rendering program with the existing illusion based REAGLE / RAYTRACE rendering system, incorporating the development of a localised daylighting model for Australian conditions. The success of this platform is clearly shown in chapter seven by the higher quality and more realistic images that can be produced using the new lighting tools. These improved images can now be used to evaluate and provide better predictive simulations of real world lighting conditions.

The integration of the RADIANCE rendering program has involved the development of:

- i) A CAD translator, to take the geometry from the EAGLE 3D CAD modelling software and convert it into a format suitable for inclusion with the RADIANCE rendering program;
- ii) A RADIANCE material library, that mirrors the materials currently defined in the REAGLE / RAYTRACE rendering system, including global environment sky maps;
- iii) A RADIANCE library of light models, suitably defined from real world manufactures luminaires and their light distribution data;

- iv) Modifications to the RADIANCE rendering program to allow for the addition of image based bump (texture) maps and automatic material orientation;
- v) The `xeagle2rad` software program that controls the REAGLE to RADIANCE conversion and rendering process;
- vi) The `xradconvert` software program that offers a consistent graphical user interface to designers with which to set up scene rendering options and conditions, and ultimately control the rendering process (using the `xeagle2rad` program); and
- vii) The `genrad` software program, that is provided to calculate the direct and diffuse radiation values for a specific site, time, date and localised climatic / atmospheric conditions, for inclusion into the RADIANCE daylighting distribution models.

The combination of the development of these tools, the provision of a better user interface, the ability to simulate more realistic local daylighting conditions and a process for accurate visual simulation, together will help designers produce an improved quality of building design in terms of integrated lighting.

8.2 Future work

Although in its current state the lighting design platform is in a useful working condition, there is scope for future development. The first area should involve the tidying up of both the RAYTRACE and RADIANCE material libraries to reflect more realistically scaled materials. Some of the materials that may be specified do not accurately reflect their real world size and would not require much effort to correct. The addition to the number of RAYTRACE light models could also be considered, as currently these are very limiting.

In terms of the daylight modelling the `genrad` program could be more rigorously tested using measured average and measured instantaneous turbidity factors. Further testing and validation could also be carried out in relation to artificial lighting simulations.

Opportunities for more advanced future developments lie in the porting of the platform to the PC environment. As personal desktop computers are becoming faster and more accessible, there is the opportunity to rewrite the lighting design software for this computing platform and thus be utilized by a larger number of designers. Currently both the EAGLE 3D modelling software and RADIANCE rendering software are available on the PC. The RAYTRACE software is not available on this platform, but as the software is written in C, the porting of this should not be difficult.

Other areas of future work include developing the CAD translator to allow the nesting of EAGLE sub models that can then be easily handled by the RADIANCE instance primitive. This would dramatically reduce the memory needed and time taken to generate a RADIANCE rendered image. The RAYTRACE program itself would also need to be developed to incorporate this instancing feature. A feature that is lacking in the REAGLE rendering system is a quality material editor. Giving the user the ability to create their own materials would be of enormous benefit. Currently a REAGLE expert must be convinced to create any new materials in both the RAYTRACE and RADIANCE material libraries.

One of the most interesting areas for future work is the extension of the lighting design platform to enable integration into other CAD packages. The process involved is quite simple, as only a new geometry conversion program is required. This new conversion program would become part of the MOG suite of programs and would convert the native geometry of the new CAD package, including material assignments, into the MOG geometry format. From here the existing `mog2rad`, and `mog2rt`, and modified `xeagle2rad`, `xconvert` and `xeagle2rin`, `xradconvert` programs could be used to generate both RAYTRACE and RADIANCE rendered images.

8.3 The end ?

This thesis is another step in my personal journey of exploring the power of computer visualisation as applied to the discipline of architectural design. With computing speed continuing to increase, and computers becoming more and more accessible in the

architectural working office, tools providing the true simulation of a building, both visually and now environmentally, are becoming more of an achievable goal. Only through access to such tools can designers begin to excel in the creation of more energy-efficient, occupier focused, attractive and stimulating built environments.

9. References

- Ashdown, I. (1992), "Radiosity and Realism", *The C Users Journal*, August 1992, pp. 33-41.
- CIE Editor Ruck, N. (1990), "Commission Internationale de l'Éclairage Guide to daylighting of building interiors, draft 1990", Paris, (to be published by Commission Internationale de l'Éclairage).
- Commission Internationale de l'Éclairage (CIE) (1970), "International recommendations for the calculation of natural daylight", *CIE Publication No. 16*, Paris: Bureau Central de la CIE.
- Commission Internationale de l'Éclairage (CIE) (1973), "Standardisation of luminance distribution of clear skies", *CIE Publication No. 22*, Paris: Bureau Central de la CIE.
- Crone, S. (1992), "Integrated Lighting System Design and Visualisation", *Architectural Dissertation*, School of Architecture, Construction and Planning, Curtin University of Technology.
- Delaunay, J. (1994), "RADIANCE Gendaylit" *manual page*, Fraunhofer Institute for Solar Energy Systems, Freiburg, Germany.
- Dench, P. (1989), "MOG Software system", Computer Animation Negus, [CAN Computer Software], Curtin University of Technology.
- Dench, P. (1993), "Raytrace V4.9 Overview of changes", *Technical Report No: 27*, School of Computing Science, Curtin University of Technology.
- Gabriels, N. (1986), "Utilisation of solar beam energy in daylight design", SERIWA Project No. 901/286, Building Management Authority of Western Australia, pp. 23,27.
- Goral, C. Torrance, K. Greenberg, D. Battaile, B. (1984) "Modeling the Interaction of Light Between Diffuse Surfaces", *Computer Graphics*, Vol. 18, No. 3, pp. 213-222.
- Hall, R. (1989), "Illumination and Color in Computer Generated Imagery", Springer-Verlag, New York.
- Illuminating Engineering Society (IES) (1984), "Calculation Procedures Committee Recommended Practice For The Calculation of Daylight Availability", *Journal of the Illuminating Engineering Society of North America*, Vol. 13, No. 4, pp 381-392.

- Illuminating Engineering Society (IES) (1993), "IES Lighting Handbook 8th Edition 1993 - reference and application", Illuminating Engineering Society, New York.
- Kasten, F. and Young, A. T. (1989), "Revised optical air mass tables and approximation formula", *Applied Optics*, Vol. 28, No. 22, pp 4735-4738.
- Kayiya, J. (1984), "The Rendering Equation", *Computer Graphics*, Vol. 20, No. 4, pp. 143-150.
- Kendrick, J., Skinner, S. (1980), "Dynamic Aspects of Daylight", *Paper presented at Symposium on Daylight, - Physical, Psychological and Architectural Aspects*, CIE, Berlin.
- Kittler, R. (1981a), "A universal calculation method from simple predetermination of natural radiation on building surfaces and solar collectors", *Building and Environment*, Vol. 16, No. 3, pp 177-182.
- Kittler, R. (1981b), "Skylight availability under typical nebulosity conditions", *Lighting Research and Technology*, Vol. 13, No. 4, pp 199-202
- Kittler, R. (1982a), "A provisional Consistent Model Defining Variable Exterior Daylight Illuminances", *Proceedings of the Daylighting and Energy Conservation conference*, University of New South Wales, Paper No 5.
- Kittler, R. (1982b), "The Importance of Daylight", *Daylighting and Energy Conservation proceedings*, University of New South Wales, Paper No. 1.
- Lane, L. O. (1981), "A new analytic expression for the equation of time", *Solar Energy*, Vol. 26, No. 5, p. 465
- Larson, C. (1991), "RADIANCE users manual (draft)", Lawrence Berkeley Laboratory, Berkeley, California.
- Littlefair, P. J., (1981), "The luminance distribution of an average sky", *Lighting Research and Technology*, Vol. 13, No. 4, pp. 192-198.
- Marriott, A. & Dench, P. (1990), "Image Generation using the Computer Animation Negus Raytracing System", *Technical Report, No. 3*, Curtin University of Technology.
- Marriott, A. (1985-1996), "RAYTRACE rendering system", Version 4.21, Computer Animation Negus, [CAN Computer Software], Curtin University of Technology.
- Marriott, A. (1991-1996), "REAGLE system", Version 1.2, Computer Animation Negus, [CAN Computer Software], Curtin University of Technology.

- Marriott, A. (1994), "REAGLE Programmers Documentation", *Technical Report*, URL, <http://puffin.curtin.edu.au/documentation/REAGLE/REAGLE.html>
- Mikler, J., Kittler, R., (1980), "Research on unsteady daylight states in building ensembles", *Private research report*, Institute of Construction and Architecture, Bratislava.
- Moon, P., Spencer D. E. (1942), "Illumination from a non-uniform sky", *Illuminating Engineer*, Vol. 37, No. 12, pp 707-726.
- Nakamura, H., Oki, M. (1979), "Study on the statistic estimation of the horizontal illuminance from unobstructed sky", *Light and Visual Environment*, Vol. 3, No. 1, pp 23.
- Navvab, M. et al. (1984), "Analysis of atmospheric turbidity for daylight calculations", *Energy and Buildings*, Vol. 6, pp 293-303.
- Oliver, A. (1991), "Using RAYTRACE while running EAGLE", *Technical Report No. 1*, Curtin University of Technology.
- Perez, R., Ineichen, P., and Seals, R., (1990), "Modeling daylight availability and irradiance components from direct and global irradiance", *Solar Energy*, Vol. 44, No. 5, pp. 271-289.
- Perez, R., Michalsky, J., and Seals, R., (1992), "Modeling sky luminance angular distribution from real sky conditions; Experimental evaluation of existing algorithms", *Journal of the IES*, Vol. 21, No. 2, pp. 84-92.
- Perez, R., Michalsky, J., and Seals, R., (1993), "All-weather model for sky luminance distribution - Preliminary configuration and validation", *Solar Energy*, Vol. 50, No. 3, pp. 235-245.
- Phong, Bui-Tuong, (1975), "Illumination for computer-generated pictures", *Communications of the ACM*, Vol. 18, No. 3, pp 311-317.
- Pierpoint, W. (1983), "A Simple sky model for daylighting Calculations", *Proceedings of the International Daylight Conference*, Phoenix, USA.
- Rogers, D. (1991), "State of the Art in Computer Graphics", Springer-Verlag, New York.
- Shirley, P. (1990), "A Ray Tracing Method for Illumination Calculation in Diffuse-Specular Scenes", *Proceedings of Graphics Interface '90*, pp. 205-212.
- Spencer, I. W. (1976), "Perth Solar Tables", C.S.I.R.O. Technical Paper No. 10.

- Ward, G. (1988a-1996), "RADIANCE software system", Version 2.4, Lighting Systems Research Group, [Computer Software] Lawrence Berkeley Laboratory.
- Ward, G. (1994a), "Gensky software program", Version 2.14, Lighting Systems Research Group, [Computer Software] Lawrence Berkeley Laboratory.
- Ward, G. (1994b), "RADIANCE manual pages". Lawrence Berkeley Laboratory, Berkeley, California.
- Ward, G. Rubinstein, F. Clear, R. (1988), "A Ray Tracing Solution for Diffuse Interreflection", *Computer Graphics*, Vol. 22, No. 4, pp 311-320.
- Ward, G. (1988b-1992), "RADIANCE reference manual", Lighting Systems Research Group, Lawrence Berkeley Laboratory.
- Whitted, T. (1980), "An Improved Illumination Model for Shaded Display", *Communications of the ACM*, Vol. 23, No. 6, pp 343-349.

APPENDICES

APPENDIX A

The following appendix contains the raw data obtained from the modified gensky and modified genrad programs. It also contains this data in a charted form as discussed in Chapter 7.

Average diffuse horizontal radiation data as calculated by the genrad program in W/m^2 for Perth W.A. (turbidity 3.1).

Time:	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
5:00	0	0	0	0	0	0	0	0	0	0	0	0
6:00	40	7	0	0	0	0	0	0	0	30	59	60
7:00	84	72	54	28	2	0	0	9	53	80	90	91
8:00	101	95	88	77	62	48	49	67	86	98	103	104
9:00	108	105	101	94	85	78	79	88	98	105	109	110
10:00	113	110	107	101	95	90	91	97	104	109	112	113
11:00	115	113	110	105	99	95	96	101	107	111	114	115
12:00	116	114	111	106	101	97	98	103	108	112	115	116
13:00	116	114	111	106	100	97	98	103	107	111	114	116
14:00	115	113	109	104	97	94	95	100	105	110	112	114
15:00	112	110	106	99	91	86	89	95	101	106	109	112
16:00	108	105	99	88	75	69	74	84	92	98	103	107
17:00	99	95	85	62	32	20	33	55	71	83	91	97
18:00	80	72	44	2	0	0	0	0	9	37	61	76
19:00	29	8	0	0	0	0	0	0	0	0	0	18
20:00	0	0	0	0	0	0	0	0	0	0	0	0

Average Direct Horizontal Radiation Data as calculated by genrad the program in
W/m² for Perth W.A. (turbidity 3.1).

Time:	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
5:00	0	0	0	0	0	0	0	0	0	0	0	0
6:00	9	0	0	0	0	0	0	0	2	6	36	39
7:00	154	83	30	3	0	0	0	0	34	138	219	220
8:00	372	289	207	122	53	19	21	75	203	352	440	441
9:00	587	508	418	308	203	141	149	242	401	559	645	648
10:00	769	697	602	473	349	277	291	400	568	724	807	815
11:00	897	833	733	590	453	377	398	515	683	831	911	928
12:00	962	904	799	643	498	423	450	571	731	870	948	976
13:00	957	905	793	628	480	410	442	560	710	837	915	955
14:00	883	834	717	546	401	339	374	486	621	735	814	867
15:00	746	699	576	405	271	220	256	355	472	574	655	719
16:00	559	511	386	226	115	81	110	188	281	370	452	525
17:00	341	292	175	54	5	0	5	34	88	155	231	306
18:00	126	85	17	0	0	0	0	0	0	8	45	99
19:00	2	0	0	0	0	0	0	0	0	0	0	0
20:00	0	0	0	0	0	0	0	0	0	0	0	0

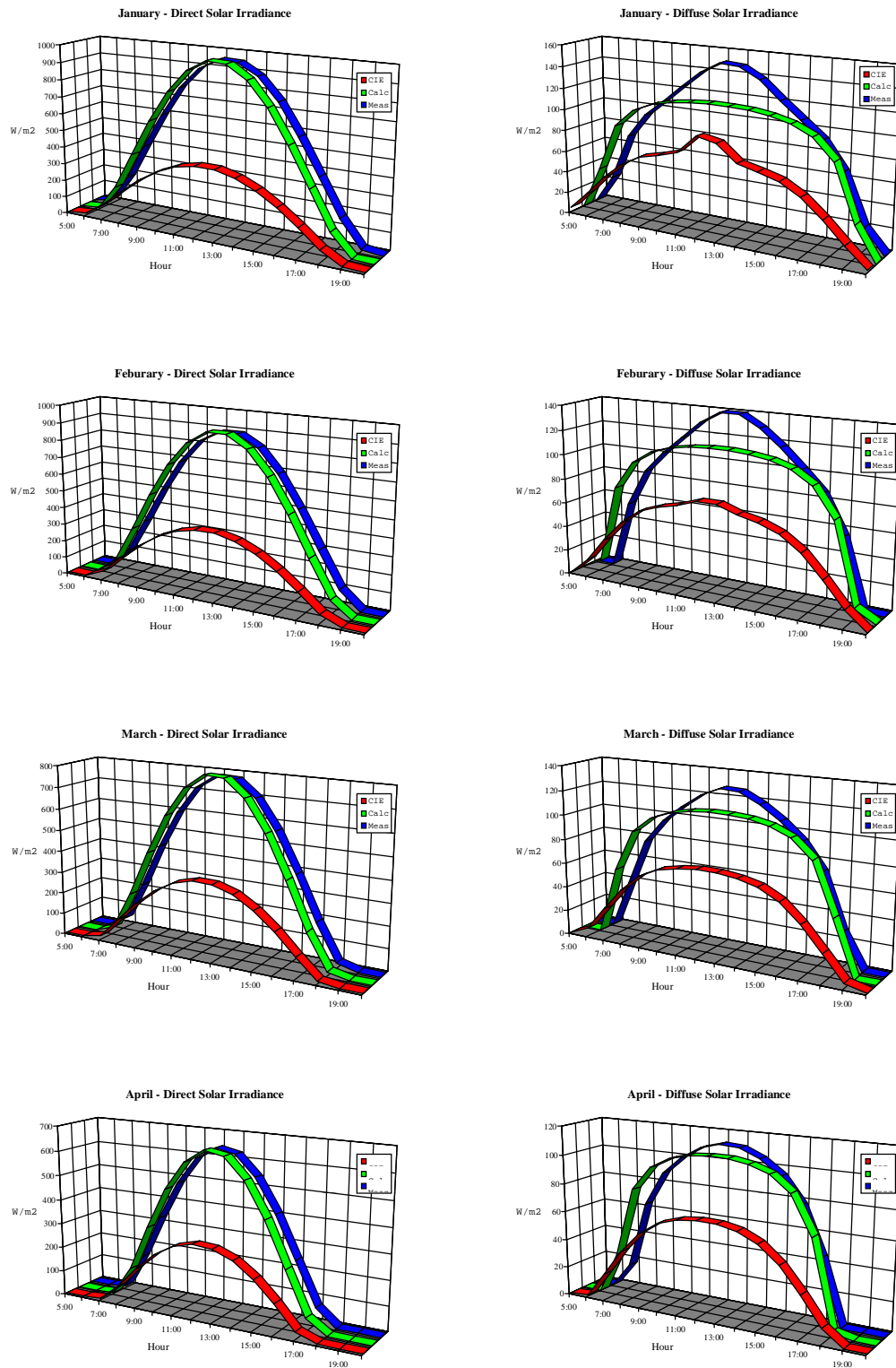
Average Diffuse Horizontal Radiation Data as calculated by the gensky program in
W/m² for Perth W.A. (turbidity 3.1).

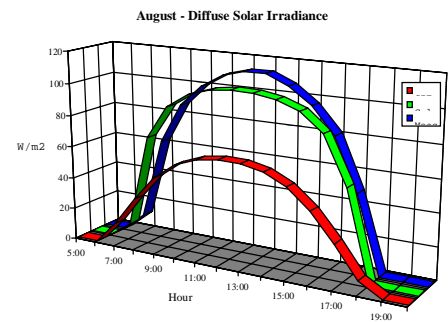
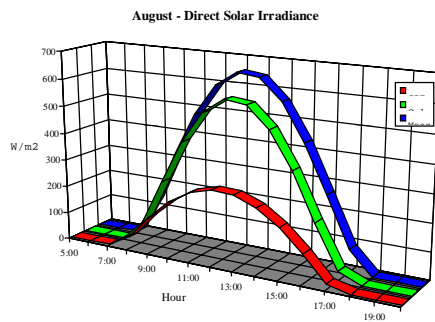
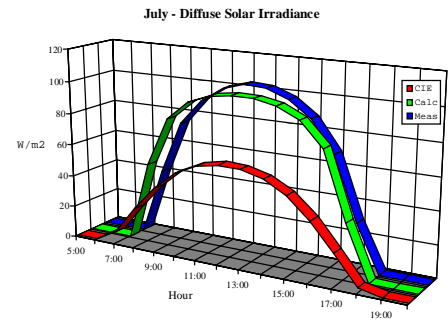
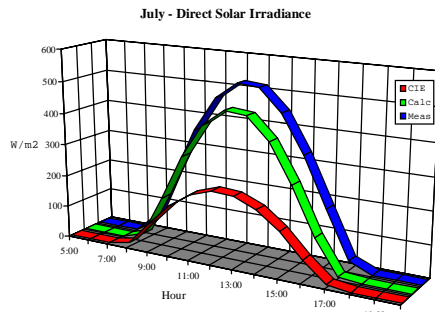
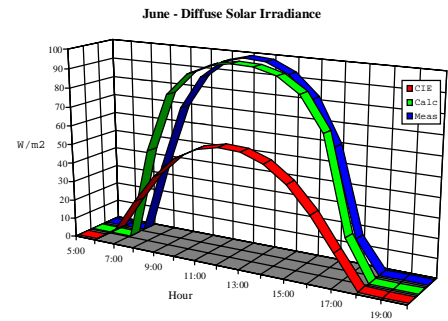
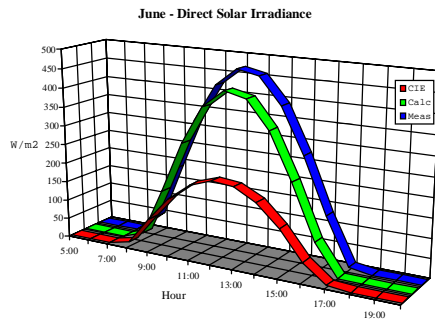
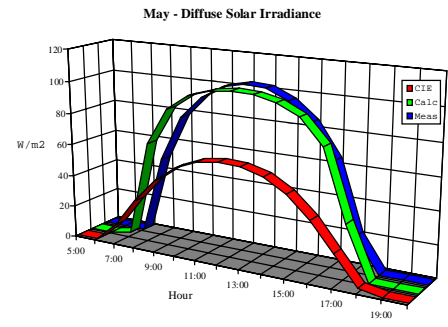
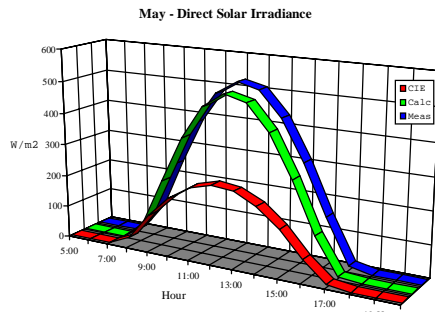
Time:	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
5:00	3	0	0	0	0	0	0	0	0	1	8	9
6:00	21	13	6	0	0	0	0	0	6	19	27	27
7:00	39	32	25	18	11	7	7	13	26	38	45	45
8:00	55	50	44	37	29	24	24	32	44	54	58	58
9:00	64	61	57	51	44	39	40	47	57	64	66	66
10:00	69	67	65	60	54	50	51	57	64	68	70	70
11:00	74	71	68	65	60	56	57	63	67	71	77	78
12:00	91	76	70	66	62	59	60	65	69	74	89	106
13:00	87	76	70	66	61	58	60	64	68	72	78	87
14:00	73	71	68	63	57	54	56	61	66	68	70	72
15:00	68	67	64	57	49	46	48	55	60	64	66	68
16:00	63	61	56	46	36	33	36	43	50	55	59	62
17:00	53	50	41	29	19	16	20	26	33	40	46	51
18:00	37	33	22	10	2	0	2	8	14	20	28	34
19:00	18	13	3	0	0	0	0	0	0	2	9	16
20:00	1	0	0	0	0	0	0	0	0	0	0	0

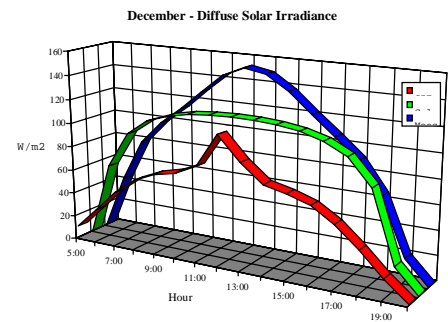
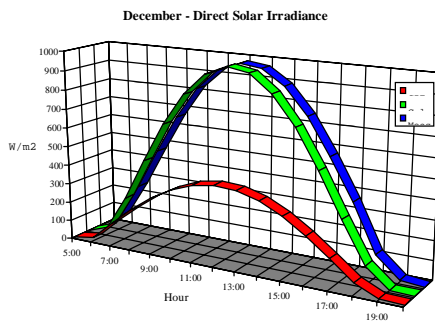
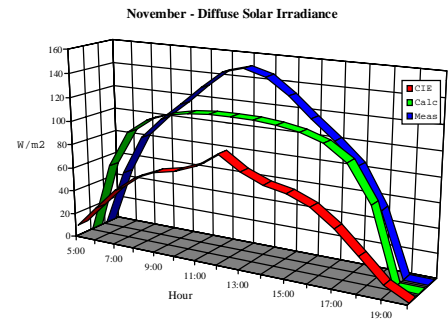
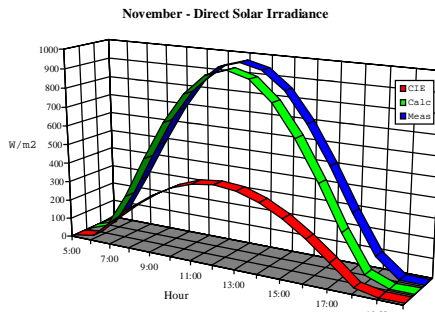
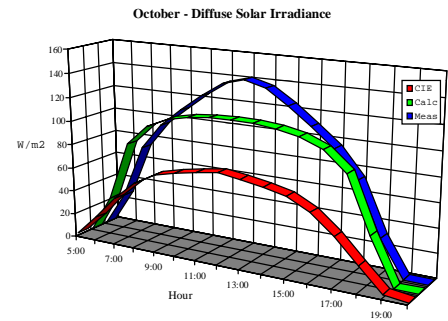
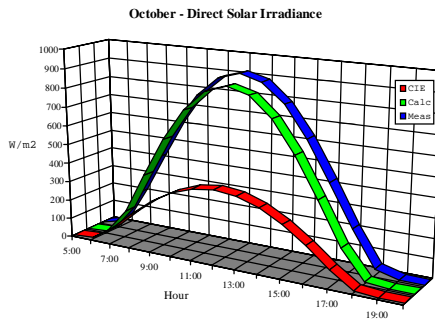
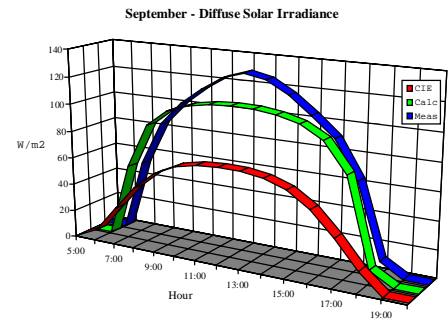
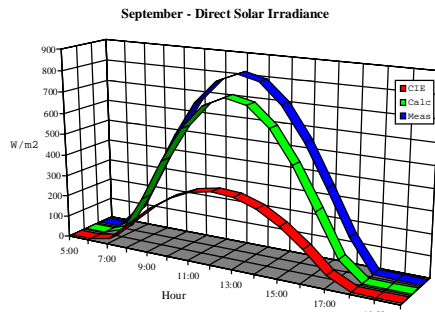
Average Diffuse Horizontal Radiation Data as calculated by the gensky program in
W/m² for Perth W.A. (turbidity 3.1).

Time:	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
5:00	0	0	0	0	0	0	0	0	0	0	0	0
6:00	9	1	0	0	0	0	0	0	0	7	18	19
7:00	85	47	16	6	0	0	0	2	20	78	116	116
8:00	180	147	112	71	30	12	13	44	111	176	209	208
9:00	266	236	203	159	113	83	87	131	199	260	290	289
10:00	335	309	276	228	179	148	155	201	268	325	352	352
11:00	383	361	327	275	223	192	202	249	313	366	391	395
12:00	407	388	353	297	242	212	224	272	332	381	405	412
13:00	405	388	351	291	234	207	221	268	324	368	393	404
14:00	378	362	321	258	201	176	192	237	289	329	355	372
15:00	326	310	266	200	145	122	139	182	228	266	294	316
16:00	255	237	189	122	68	49	66	106	148	183	214	241
17:00	168	148	96	30	7	4	8	18	52	87	122	153
18:00	71	48	11	0	0	0	0	0	2	8	23	57
19:00	6	1	0	0	0	0	0	0	0	0	0	4
20:00	0	0	0	0	0	0	0	0	0	0	0	0

Charts of the raw data shown above, together with measured data from appendix B. All graphs are for Perth W.A. based on a turbidity of 3.1.







APPENDIX B

The following tables contains measured average diffuse and direct radiation data.

DIFFUSE HORIZONTAL RADIATION DATA - (Gabriels 1986 p23)

Location : West Perth W.A.
 Longitude : 115 degrees 52 minutes
 Reference Longitude : 120 degrees
 Latitude : 31 degrees 57 minutes SOUTH

Time	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
5.00												
6.00	26									28	48	45
7.00	70	55	39	17				11	52	73	82	80
8.00	93	85	77	64	49	37	41	62	84	95	100	99
9.00	108	102	96	88	78	70	74	87	101	110	115	113
10.00	124	116	109	100	92	87	90	100	113	124	131	129
11.00	139	130	120	108	100	96	99	108	123	136	145	144
12.00	149	139	127	112	103	99	103	112	128	141	150	152
13.00	148	139	126	111	102	98	102	112	125	136	145	149
14.00	138	130	118	105	96	93	97	106	116	124	132	137
15.00	122	117	107	95	85	82	87	96	104	110	116	121
16.00	107	102	93	78	64	60	68	80	90	95	101	106
17.00	92	86	72	45	21	16	29	48	63	73	84	90
18.00	67	55	28						10	30	52	65
19.00	21											19
20.00												
Total	1404	1256	1112	923	790	738	790	922	1109	1275	1401	1449

DIRECT HORIZONTAL RADIATION DATA - (Gabriels 1986, p27)

Location : West Perth W.A.
 Longitude : 115 degrees 52 minutes
 Reference Longitude : 120 degrees
 Latitude : 31 degrees 57 minutes SOUTH

Time	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
5.00												
6.00	22									28	80	70
7.00	183	102	47	9				4	99	207	275	256
8.00	392	303	227	150	84	46	59	142	301	420	488	465
9.00	596	510	425	327	241	188	214	323	499	620	684	663
10.00	770	690	596	481	382	324	359	482	665	780	839	824
11.00	890	819	720	590	480	421	466	597	775	879	934	929
12.00	950	884	780	638	523	468	519	651	819	913	968	973
13.00	946	885	774	622	506	456	512	640	794	881	940	958
14.00	882	820	701	542	429	388	445	564	702	784	851	883
15.00	756	691	567	408	305	272	328	432	550	626	701	748
16.00	579	512	388	239	150	128	176	262	358	427	508	565
17.00	373	305	190	67	14	8	30	83	153	213	295	358
18.00	165	103	24						4	32	96	155
19.00	14											11
20.00												
Total	7518	6624	5439	4073	3114	2699	3108	4180	5719	6810	7659	7858

APPENDIX C

This appendix contains various manual pages from the RADIANCE software suite that provide specific and detailed information in the usage and availability of options that are given to the RADIANCE programs.

GENSKY(1)

GENSKY(1)

NAME

gensky - generate a RADIANCE description of the sky

SYNOPSIS

```
gensky month day hour [ options ]
gensky -ang altitude azimuth [ options ]
gensky -defaults
```

DESCRIPTION

Gensky produces a RADIANCE scene description for the CIE standard sky distribution at the given date and local standard time. If the hour is preceded by a plus sign ('+'), then it is interpreted as local solar time instead of standard time. The second form gives the solar angles explicitly. The altitude is measured in degrees above the horizon, and the azimuth is measured in degrees west of South. The third form prints the default option values.

The output sky distribution is given as a brightness function, skyfunc. Its value is in watts/steradian/meter². The x axis points east, the y axis points north, and the z axis corresponds to the zenith. The actual material and surface(s) used for the sky is left up to the user. For a hemispherical blue sky, the description might be:

```
!gensky 4 1 14

skyfunc glow skyglow
0
0
4 .9 .9 1 0

skyglow source sky
0
0
4 0 0 1 180
```

Often, skyfunc will actually be used to characterize the light coming in from a window.

In addition to the specification of a sky distribution function, gensky suggests an ambient value in a comment at the beginning of the description to use with the -av option of the RADIANCE rendering programs. (See rview(1) and rpict(1).) This value is the cosine-weighted radiance of the sky in watts/steradian/meter².

Gensky supports the following options.

- s Sunny sky without sun. The sky distribution will correspond to a standard CIE clear day.
- +s Sunny sky with sun. In addition to the sky distribution function, a source description of the sun is generated.
- c Cloudy sky. The sky distribution will correspond to a standard CIE overcast day.
- I Intermediate sky without sun. The sky will correspond to a standard CIE intermediate day.
- +I Intermediate sky with sun. In addition to the sky distribution, a (somewhat subdued) sun is generated.
- u Uniform cloudy sky. The sky distribution will be completely uniform.
- g rfl Average ground reflectance is rfl. This value is used to compute skyfunc when Dz is negative. Ground plane brightness is the same for -s as for +s. (Likewise for -i and +i, but see the -r option below.)
- b brt The zenith brightness is brt. Zenith radiance (in watts/steradian/meter²) is normally computed from the sun angle and sky turbidity (for sunny sky). It can be given directly instead, using this option.
- B irrad Same as -b, except zenith brightness is computed from the horizontal diffuse irradiance (in watts/meter²).
- r rad The solar radiance is rad. Solar radiance (in watts/steradian/meter²) is normally computed from the solar altitude. This option may be used to override the default calculation. If a value of zero is given, no sun description is produced, and the contribution of direct solar to ground brightness is neglected.
- R irrad Same as -r, except solar radiance is computed from the horizontal direct irradiance (in watts/meter²).
- t trb The turbidity factor is trb. Greater turbidity factors correspond to greater atmospheric scattering. A turbidity factor of 1.0 indicates an ideal clear atmosphere.

The following options do not apply when the solar altitude and azimuth are given explicitly.

- a lat The site latitude is lat degrees north. (Use negative angle for south latitude.) This is used in the calculation of sun angle.
- o lon The site longitude is lon degrees west. (Use negative angle for east longitude.) This is used in the calculation of solar time and sun angle. Be sure to give the corresponding standard meridian also! If

solar time is given directly, then this option has no effect.

-m mer The site standard meridian is mer degrees west of Greenwich. (Use negative angle for east.) This is used in the calculation of solar time. Be sure to give the correct longitude also! If solar time is given directly, then this option has no effect.

EXAMPLE

To produce a sunny sky for July 4th at 2:30pm standard time at a site latitude of 42 degrees, 108 degrees west longitude, and a 110 degree standard meridian:

```
gensky 7 4 14.5 +s -a 42 -o 108 -m 110
```

To produce a sunny sky distribution for a specific sun position but without the sun description:

```
gensky -ang 23 -40 -s
```

FILES

```
/usr/local/lib/ray/skybright.cal
```

AUTHOR

Greg Ward

SEE ALSO

```
rpict(1), rview(1), xform(1)
```

RADIANCE 11/15/93 3

NAME

gendaylit - generates a RADIANCE description of the daylit sources using Perez models for diffuse and direct components

SYNOPSIS

```
gendaylit month day hour [-P|-W|-L|-G] direct value diffuse
value [ options ]
gendaylit -ang altitude azimuth [-P|-W|-L|-G] direct value
diffuse value [ options ]
```

DESCRIPTION

Gendaylit produces a RADIANCE scene description based on an angular distribution of the daylight sources (direct+diffuse) for the given atmospheric conditions (direct and diffuse component of the solar radiation), date and local standard time. The default output is the radiance of the sun (direct) and the sky (diffuse) integrated over the visible spectral range (380-780 nm). We have used the calculation of the sun's position and the ground brightness models which were programmed in gensky.

The diffuse angular distribution is calculated using the Perez et al. sky luminance distribution model (see Solar Energy Vol. 50, No. 3, pp. 235-245, 1993) which, quoting Perez, describes "the mean instantaneous sky luminance angular distribution patterns for all sky conditions from overcast to clear, through partly cloudy, skies". The correctness of the resulting sky radiance/luminance values in this simulation is ensured through the normalization of the modelled sky diffuse to the measured sky diffuse irradiances/illuminances.

The direct radiation is understood here as the radiant flux coming from the sun and an area of approximately 3 degrees around the sun (World Meteorological Organisation specifications for measuring the direct radiation. The aperture angle of a pyrhelimeter is approximately 6 degrees). To simplify the calculations for the direct radiation, the sun is represented as a disk and no circumsolar radiation is modelled in the 3 degrees around the sun. This means that all the measured/evaluated direct radiation is added to the 0.5 degree sun source.

The direct and diffuse solar irradiances/illuminances are the inputs needed for the calculation. These quantities are the commonly accessible data from radiometric measurement centres, conversion models (e.g. global irradiance to direct irradiance), or from the Test Reference Year. The use of such data is the recommended method for achieving the most accurate simulation results.

The atmospheric conditions are modelled with the Perez et al. parametrization (see Solar Energy Vol. 44, No 5, pp. 271-289, 1990), which is dependent on the values for the direct and the diffuse irradiances. The three parameters are epsilon, delta and the solar zenith angle. "Epsilon variations express the transition from a totally overcast sky (epsilon=1) to a low turbidity clear sky (epsilon>6); delta variations reflect

the opacity/thickness of the clouds". Delta can vary from 0.05 representing a dark sky to 0.5 for a very bright sky. Not every combination of epsilon, delta and solar zenith angle is possible. For a clear day, if epsilon and the solar zenith angle are known, then delta can be determined. For intermediate or overcast days, the sky can be dark or bright, giving a range of possible values for delta when epsilon and the solar zenith are fixed. The relation between epsilon and delta is represented in a figure on page 393 in Solar Energy Vol.42, No 5, 1989, or can be obtained from the author of this RADIANCE extension upon request. Note that the epsilon parameter is a function of the solar zenith angle. It means that a clear day will not be defined by fixed values of epsilon and delta. Consequently the input parameters, epsilon, delta and the solar zenith angle, have to be determined on a graph. It might be easier to work with the measured direct and diffuse components (direct normal irradiance/illuminance and diffuse horizontal irradiance/illuminance) than with the epsilon and delta parameters.

The conversion of irradiance into illuminance for the direct and the diffuse components is determined by the luminous efficacy models of Perez et al. (see Solar Energy Vol. 44, No 5, pp. 271-289, 1990). To convert the luminance values into radiance integrated over the visible range of the spectrum, we divide the luminance by the white light efficacy factor of 179 lm/W. This is consistent with the RADIANCE calculation because the luminance will be recalculated from the radiance integrated over the visible range by :

```
luminance = radiance_integrated_over_visible_range * 179
or
```

```
luminance = (RED*.263 + GREEN*.655 + BLUE*.082) * 179
  with the capability to model colour (where
radiance_integrated_over_visible_range == (RED + GREEN +
BLUE)/3).
```

From gensky , if the hour is preceded by a plus sign ('+'), then it is interpreted as local solar time instead of standard time. The second form gives the solar angles explicitly. The altitude is measured in degrees above the horizon, and the azimuth is measured in degrees west of South.

The x axis points east, the y axis points north, and the z axis corresponds to the zenith. The actual material and surface(s) used for the sky is left up to the user.

In addition to the specification of a sky distribution function, gendaylit suggests an ambient value in a comment at the beginning of the description to use with the -av option of the RADIANCE rendering programs. (See rvview(1) and rpict(1).) This value is the cosine-weighted radiance of the sky in W/sr/m².

Gendaylit can be used with the following input parameters. They offer three possibilities to run it: with the Perez parametrization, with the irradiance values and with the illuminance values.

- P epsilon delta (these are the Perez parameters)
- W direct-normal-irradiance (W/m²), diffuse-horizontal-irradiance (W/m²)
- L direct-normal-illuminance (lm/m²), diffuse-horizontal-illuminance (lm/m²)
- G direct-horizontal-irradiance (W/m²), diffuse-horizontal-irradiance (W/m²)

The output can be set to either the radiance of the visible radiation (default), the solar radiance (full spectrum) or the luminance.

- O[0|1|2] (0=output in W/m²/sr visible radiation, 0=output in W/m²/sr solar radiation, 2=output in lm/m²/sr luminance)

Gendaylit supports the following options.

- s The source description of the sun is not generated.
- g rfl Average ground reflectance is rfl. This value is used to compute skyfunc when Dz is negative.

The following options do not apply when the solar altitude and azimuth are given explicitly.

- a lat The site latitude is lat degrees north. (Use negative angle for south latitude.) This is used in the calculation of sun angle.
- o lon The site longitude is lon degrees west. (Use negative angle for east longitude.) This is used in the calculation of solar time and sun angle. Be sure to give the corresponding standard meridian also! If solar time is given directly, then this option has no effect.
- m mer The site standard meridian is mer degrees west of Greenwich. (Use negative angle for east.) This is used in the calculation of solar time. Be sure to give the correct longitude also! If solar time is given directly, then this option has no effect.

EXAMPLES

A clear non-turbid sky for a solar altitude of 60 degrees and an azimuth of 0 degree might be defined by:

```
gendaylit -ang 60 0 -P 6.3 0.12 or gendaylit -ang 60 0 -W
840 135 This sky description corresponds to the clear sky
standard of the CIE.
```

The corresponding sky with a high turbidity is:

```
gendaylit -ang 60 0 -P 3.2 0.24 or gendaylit -ang 60 0 -W
720 280
```

The dark overcast sky (corresponding to the CIE overcast standard, see CIE draft standard, Pub. No. CIE DS 003, 1st Edition, 1994) is obtained by:

```
gendaylit -ang 60 0 -P 1 0.08
```

A bright overcast sky is modelled with a larger value of delta, for example:

```
gendaylit -ang 60 0 -P 1 0.35
```

To generate the same bright overcast sky for March 2nd at 3:15pm standard time at a site latitude of 42 degrees, 108 degrees west longitude, and a 110 degrees standard meridian:

```
gendaylit 3 2 15.25 -a 42 -o 108 -m 110 -P 1 0.35
```

FILES

```
/usr/local/lib/ray/perezlum.cal
```

AUTHOR

Jean-Jacques Delaunay, FhG-ISE Freiburg, (jean@ise.fhg.de)

ACKNOWLEDGMENTS

The work on this program was supported by the German Federal Ministry for Research and Technology (BMFT) under contract No. 0329294A, and a scholarship from the French Environment and Energy Agency (ADEME) which was co-funded by Bouygues. Many thanks to Peter Apian-Bennewitz, Arndt Berger, Ann Kovach, R. Perez, C. Gueymard and G. Ward for their help.

SEE ALSO

```
gensky(1), rpict(1), rview(1), xform(1)
```

RADIANCE ISE/ADEME EXTENSIONS4/12/94

RAD(1)

RAD(1)

NAME

rad - render a RADIANCE scene

SYNOPSIS

```
rad [ -s ][ -n ][ -e ][ -V ][ -w ][ -v view ][ -o device ]  
rfile [ VAR=value .. ]
```

DESCRIPTION

Rad is an executive program that reads the given rfile and makes appropriate calls to oconv(1), mkillum(1), rpict(1), pfilt(1), and/or rview(1) to render a specific scene. Variables in rfile give input files and qualitative information about the rendering(s) desired that together enable rad to intelligently set parameter values and control the simulation.

Normally, commands are echoed to the standard output as they are executed. The -s option tells rad to do its work silently. The -n option tells rad not to take any action (ie. not to actually execute any commands). The -e option tells rad to explicate all variables used for the simulation, including default values not specified in the input file, and print them on the standard output.

Normally, rad will produce one picture for each view given in rfile. The -v option may be used to specify a single desired view. The view argument may either be a complete view specification (enclosed in quotes and beginning with an optional identifier) or a number or single-word identifier to match a view defined in rfile. If the argument is one of the standard view identifiers, it may or may not be further elaborated in rfile. (See "view" variable description, below.) If the argument does not match any views in rfile and is not one of the standard views, no rendering will take place. This may be convenient when the only action desired of rad is the rebuilding of the octree. In particular, the argument "0" will never match a view.

If the -V option is given, each view will be printed on the standard output before being applied, in a form suitable for use in a view file or rpict rendering sequence. This is helpful as feedback or for accessing the rad view assignments without necessarily starting a rendering.

By default, rad will run rpict and pfilt to produce a picture for each view. The -o option specifies an output device for rview (usually "x11") and runs this interactive program instead, using the first view in rfile or the view given with the -v option as the starting point.

Additional variable settings may be added or overridden on the command line following rfile. Upper case variables specified more than once will result in a warning message (unless the -w option is present), and the last value given will be the one used.

The -w option turns off warnings about multiply and misassigned variables.

Rendering variable assignments appear one per line in rfile. The name of the variable is followed by an optional equals sign ('=') and its value(s). The end of line may be escaped with a backslash ('\'), though it is not usually necessary since additional variable values may be given in multiple assignments. Variables that should have only one value are given in upper case. Variables that may have multiple values are given in lower case. Variables may be abbreviated by their first three letters. Comments in rfile start with a pound sign ('#') and proceed to the end of line.

The rendering variables, their interpretations and default values are given below.

OCTREE The name of the octree file. The default name is the same as rfile but with any suffix replaced by ".oct".

ZONE This variable specifies the volume of interest for this simulation. The first word is either "Interior" or "Exterior", depending on whether the zone is to be observed from the inside or the outside, respectively. (A single letter may be given, and case does not matter.) The following six numbers are the minimum and maximum X coordinates, minimum and maximum Y, and minimum and maximum Z for the zone perimeter. It is important to give the zone as it is used to determine many of the rendering parameters. The default exterior zone is the bounding cube for the scene as computed by oconv.

EXPOSURE This variable tells rad how to adjust the exposure for display. It is important to set this variable properly as it is used to determine the ambient value. An appropriate setting may be discovered by running rview and noting the exposure given by the "exposure =" command. As in rview and pfilt, the exposure setting may be given either as a multiplier or as a number of f-stop adjustments (eg. +2 or -1.5). There is no default value for this variable. If it is not given, an average level will be computed by pfilt and the ambient value will be set to 10 for exterior zones and 0.01 for interior zones.

scene This variable is used to specify one or more scene input files. These files will be given together with the materials file(s) and any options specified by the "oconv" variable to oconv to produce the octree given by the "OCTREE" variable. If the "scene" variable is not present, then the octree must already exist in order for rad to work. Even if this variable is given, oconv will not be run unless the octree is out of date with respect to the input files. Note that the order of files in this variable is important for oconv to work properly, and files given in later variable assignments will appear after previous ones on the oconv command line.

materials This variable is used to specify files that, although they must appear on the oconv command line, do not affect the actual octree itself. Keeping the materials in separate files allows them to be modified without requiring the octree to be rebuilt

(a sometimes costly procedure). These files should not contain any geometry, and the -f option must not be given in the "oconv" variable for this to work.

illum This variable is used to specify files with surfaces to be converted into illum sources by mkillum(1). When this variable is given, additional octree files will be created to contain the scene before and after illum source conversion. These files will be named according to the (default) value of the OCTREE variable, with either a '0' or a '1' appearing just before the file type suffix (usually ".oct").

objects This variable is used for files that, although they do not appear on the oconv command line, contain geometric information that is referenced indirectly by the scene files. If any of these files is changed, the octree will be rebuilt. (The raddepend(1) command may be used to find these dependencies automatically.)

view This variable is used to specify a desired view for this zone. Any number of "view" lines may be given, and each will result in a rendered picture (unless the -v or -o option is specified). The value for this variable is an optional identifier followed by any number of view options (see rpict(1) for a complete listing). The identifier is used in file naming and associating a desired view with the -v command line option. Also, there are several standard view identifiers defined by rad. These standard views are specified by strings of the form "[Xx]?[Yy]?[Zz]?[vlah]?". (That is, an optional upper or lower case X followed by an optional upper or lower case Y followed by an optional upper or lower case Z followed by an optional lower case V, L, A or H.) The letters indicate the desired view position, where upper case X means maximum X, lower case means minimum and so on. The final letter is the view type, where 'v' is perspective (the default), 'l' is parallel, 'a' is angular fisheye, and 'h' is hemispherical fisheye. A perspective view from maximum X, minimum Y would be "Xy" or "Xyv". A parallel view from maximum Z would be "Zl". If "ZONE" is an interior zone, the standard views will be inside the perimeter. If it is an exterior zone, the standard views will be outside. Note that the standard views are best used as starting points, and additional arguments may be given after the identifier to modify a standard view to suit a particular model. The default view is "X" if no views are specified.

UP The vertical axis for this scene. A negative axis may be specified with a minus sign (eg. "-Y"). There is no default value for this variable, although the standard views assume Z is up if no other axis is specified.

RESOLUTION This variable specifies the desired final picture resolution. If only a single number is given, this value will be used for both the horizontal and

vertical picture dimensions. If two numbers are given, the first is the horizontal resolution and the second is the vertical resolution. If three numbers are given, the third is taken as the pixel aspect ratio for the final picture (a real value). If the pixel aspect ratio is zero, the exact dimensions given will be those produced. Otherwise, they will be used as a frame in which the final image must fit. The default value for this variable is 512.

QUALITY This variable sets the overall rendering quality desired. It can have one of three values, "LOW", "MEDIUM" or "HIGH". These may be abbreviated by their first letter, and may be in upper or lower case. Most of the rendering options will be affected by this setting. The default value is "L".

PENUMBRAS This is a boolean variable indicating whether or not penumbras are desired. A value of "TRUE" will result in penumbras (soft shadows), and a value of "FALSE" will result in no penumbras (sharp shadows). True and false may be written in upper or lower case, and may be abbreviated by a single letter. Renderings generally proceed much faster without penumbras. The default value is "F".

INDIRECT This variable indicates how many diffuse reflections are important in the general lighting of this zone. A direct lighting system (eg. fluorescent troffers recessed in the ceiling) corresponds to an indirect level of 0. An indirect lighting system (eg. hanging fluorescents directed at a reflective ceiling) corresponds to an indirect level of 1. A diffuse light shelf reflecting sunlight onto the ceiling would correspond to an indirect level of 2. The setting of this variable partially determines how many interreflections will be calculated. The default value is 0.

PICTURE This is the root name of the output picture file(s). This name will have appended the view identifier (or a number if no id was used) and a ".pic" suffix. If a picture corresponding to a specific view exists and is not out of date with respect to the given octree, it will not be rerendered. The default value for this variable is the root portion of rfile.

AMBFILE This is the name of the file where "ambient" or diffuse interreflection values will be stored by rpict or rview. Although it is not required, an ambient file should be given whenever an interreflection calculation is expected. This will optimize successive runs and minimize artifacts. An interreflection calculation will take place when the "QUALITY" variable is set to HIGH, or when the "QUALITY" variable is set to MEDIUM and "INDIRECT" is positive. There is no default value for this variable.

DETAIL This variable specifies the level of visual detail in this zone, and is used to determine image sampling rate, among other things. If there are few surfaces

and simple shading, then this should be set to LOW. For a zone with some furniture it might be set to MEDIUM. If the space is very cluttered or contains a lot of geometric detail and textures, then it should be set to HIGH. The default value is "M".

VARIABILITY This variable tells rad how much light varies over the surfaces of this zone, and is used to determine what level of sampling is necessary in the indirect calculation. For an electric lighting system with uniform coverage, the value should be set to LOW. For a space with spot lighting or a window with sky illumination only, it might be set to MEDIUM. For a space with penetrating sunlight casting bright patches in a few places, it should be set to HIGH. The default value is "L".

OPTFILE This is the name of a file in which rad will place the appropriate rendering options. This file can later be accessed by rpict or rview in subsequent manual runs using the at-sign ('@') file insert option. (Using an "OPTFILE" also reduces the length of the rendering command, which improves appearance and may even be necessary on some systems.) There is no default value for this variable.

REPORT This variable may be used to specify a reporting interval for batch rendering. Given in minutes, this value is multiplied by 60 and passed to rpict with the -t option. If a filename is given after the interval, it will be used as the error file for reports and error messages instead of the standard error. (See the -e option of rpict(1). There is no default value for this variable.

oconv This variable may be used to specify special options to oconv. See the oconv(1) manual page for a list of valid options.

mkillum This variable may be used to specify additional options to mkillum. See the rtrace(1) manual page for a list of valid options.

render This variable may be used to specify additional options to rpict or rview. These options will appear after the options set automatically by rad, and thus will override the default values.

pfilt This variable may be used to specify additional options to pfilt. See the pfilt(1) manual page for details.

EXAMPLES

A minimal input file for rad might look like this:

```
::::::::::::
sample.rif
::::::::::::
# The octree we want to use:
OCTREE= tutor.oct # w/o this line, name would be
"sample.oct"
# Our scene input files:
scene= sky.rad outside.rad room.rad srcwindow.rad
```

```
# The interior zone cavity:
ZONE= I 0 3 0 2 0 1.75 # default would be scene bounding
cube
# The z-axis is up:
UP= Z # no default - would use view spec.
# Our exposure needs one f-stop boost:
EXPOSURE= +1 # default is computed ex post facto
```

Note that we have not specified any views in the file above. The standard default view "X" would be used if we were to run rad on this file. If we only want to see what default values rad would use without actually executing anything, we can invoke it thusly:

```
rad -n -e sample.rif
```

This will print the variables we have given as well as default values rad has assigned for us. Also, we will see the list of commands that rad would have executed had the -n option not been present. (Note if the octree, "tutor.oct", is not present, an error will result as it is needed to determine some of the option settings.)

Different option combinations have specific uses, ie:

```
rad -v 0 sample.rif OPT=samp.opt # build octree, put
options in "sample.opt"
rad -n -e -s sample.rif > full.rif # make a complete rad
file
rad -n sample.rif > script.sh # make a script of commands
rad -V -v Zl -n -s sample.rif > plan.vf # make a plan view
file
rad -s sample.rif & # execute silently in the background
```

If we decide that the default values rad has chosen for our variables are not all appropriate, we can add some more assignments to the file:

```
QUAL= MED # default was low
DET= low # default was medium - our space is almost empty
PEN= True # we want to see soft shadows from our window
VAR= hi # daylight can result in fairly harsh lighting
view= XYa -vv 120 # let's try a fisheye view
PICT= tutor # our picture name will be "tutorXYa.pic"
```

Note the use of abbreviations, and the modification of a standard view. Now we can invoke rad to take a look at our scene interactively with rview:

```
rad -o x11 sample.rif
```

Rad will run oconv first to create the octree (assuming it doesn't already exist), then rview with a long list of options. Let's say that from within rview, we wrote out the view files "view1.vp" and "view2.vp". We could add these to "sample.rif" like so:

```
view= vw1 -vf view1.vp # Our first view
view= vw2 -vf view2.vp # Our second view
RESOLUTION= 1024 # Let's go for a higher resolution result
```

To start rview again using vw2 instead of the default, we use:

```
rad -o x11 -v vw2 sample.rif
```

Once we are happy with the variable settings in our file, we can run rad in the background to produce one image for each view:

```
rad sample.rif REP=5 >& errfile &
```

This will report progress every five minutes to "errfile".

FILES

\$(PICTURE)\$(view).raw Unfiltered output of rpict

AUTHOR

Greg Ward

BUGS

Incremental building of octrees is not supported as it would add considerable complexity to rad. Complicated scene builds should still be left to make(1), which has a robust mechanism for handling hierarchical dependencies. If make is used in this fashion, then only the "OCTREE" variable of rad is needed.

The use of some pfilter options is awkward, since the "EXPOSURE" variable results in a single pass invocation (the -1 option of pfilter and two passes are necessary for certain effects, such as star patterns. The way around this problem is to specify -2 in the "pfilter" variable, followed by an exposure that undoes the "EXPOSURE" variable setting. This is preferable to leaving out the "EXPOSURE" variable, since the exposure level is needed to accurately determine the ambient value for rpict.

The use of upper and lower case naming for the standard views may be problematic on systems that don't distinguish case in filenames.

WARNING

Rad will automatically remove an ambient file that is out of date, overwrite an octree and clobber pictures, so it is good to get in the habit of running first with the -n option to see what it will do before it does it. If you think a file rad wants to remove or clobber is just fine, use the touch(1) command to bring it up to date manually before running rad. Note that you may have to let rad create the octree, then do the touch, to really fool it.

SEE ALSO

make(1), mkillum(1), oconv(1), pfilter(1), raddpend(1), rpict(1), rtrace(1), rview(1), touch(1), vgaimage(1), ximage(1)

RADIANCE 4/15/94

APPENDIX D

This appendix contains the rayinit.cal file that has been modified to allow for automatic rotations of material mappings on surface geometry.

```
{
    { SCCSid "@(#)rayinit.cal 2.10 2/29/96 LBL" }
    Initialisation file for Radiance.

    The following are predefined:

    Dx, Dy, Dz      - ray direction
    Nx, Ny, Nz      - surface normal
    Px, Py, Pz      - intersection point
    T               - distance from start
    Ts              - single ray (shadow) distance
    Rdot            - ray dot product
    S               - world scale
    Tx, Ty, Tz      - world origin
    Ix, Iy, Iz      - world i unit vector
    Jx, Jy, Jz      - world j unit vector
    Kx, Ky, Kz      - world k unit vector
    arg(n)          - real arguments, arg(0) is count

    For brdf functions, the following are also available:

    NxP, NyP, NzP   - perturbed surface normal
    RdotP           - perturbed ray dot product
    CrP, CgP, CbP   - perturbed material color

    For prisms1 and prisms2 types, the following are available:

    DxA, DyA, DzA   - direction to target light source

    Library functions:

    if(a, b, c)      - if a positive, return b, else c
    select(N, a1, a2, ..) - return aN
    sqrt(x)         - square root function

    sin(x), cos(x), tan(x),
    asin(x), acos(x),
    atan(x), atan2(y,x) - standard trig functions

    floor(x), ceil(x) - g.l.b. & l.u.b.
    exp(x), log(x), log10(x) - exponent and log functions

    erf(z), erfc(z) - error functions

    rand(x)          - pseudo-random function (0 to 1)

    hermite(p0,p1,r0,r1,t) - 1-dimensional hermite polynomial

    noise3(x,y,z), noise3a(x,y,z),
    noise3b(x,y,z), noise3c(x,y,z) - noise function with
    gradient (-1 to 1)

    fnoise3(x,y,z) - fractal noise function (-1 to 1)
}
```

```

    { Backward compatibility }
AC = arg(0);
A1 = arg(1); A2 = arg(2); A3 = arg(3); A4 = arg(4); A5 = arg(5);
A6 = arg(6); A7 = arg(7); A8 = arg(8); A9 = arg(9); A10 = arg(10);

    { Forward compatibility (?) }
D(i) = select(i, Dx, Dy, Dz);
N(i) = select(i, Nx, Ny, Nz);
P(i) = select(i, Px, Py, Pz);
noise3d(i,x,y,z) = select(i, noise3a(x,y,z), noise3b(x,y,z),
noise3c(x,y,z));

    { More robust versions of library functions }
bound(a,x,b) : if(a-x, a, if(x-b, b, x));
Acos(x) : acos(bound(-1,x,1));
Asin(x) : asin(bound(-1,x,1));
Atan2(y,x) : if(x*x+y*y, atan2(y,x), 0);
Exp(x) : if(-x-100, 0, exp(x));
Sqrt(x) : if(x, sqrt(x), 0);

    { Useful constants }
PI : 3.14159265358979323846;
DEGREE : PI/180;
FTINY : 1e-7;

    { Useful functions }
and(a,b) : if( a, b, a );
or(a,b) : if( a, a, b );
not(a) : if( a, -1, 1 );
abs(x) : if( x, x, -x );
sgn(x) : if( x, 1, if(-x, -1, 0) );
sq(x) : x*x;
max(a,b) : if( a-b, a, b );
min(a,b) : if( a-b, b, a );
inside(a,x,b) : and(x-a,b-x);
frac(x) : x - floor(x);
mod(n,d) : n - floor(n/d)*d;
tri(n,d) : abs( d - mod(n-d,2*d) );
linterp(t,p0,p1) : (1-t)*p0 + t*p1;
noop(v) = v;
clip(v) = bound(0,v,1);
noneg(v) = if(v,v,0);
red(r,g,b) = if(r,r,0);
green(r,g,b) = if(g,g,0);
blue(r,g,b) = if(b,b,0);
grey(r,g,b) = noneg(.265074126*r + .670114631*g + .064811243*b);
clip_r(r,g,b) = bound(0,r,1);
clip_g(r,g,b) = bound(0,g,1);
clip_b(r,g,b) = bound(0,b,1);
clipgrey(r,g,b) = min(grey(r,g,b),1);

dot(v1,v2) : v1(1)*v2(1) + v1(2)*v2(2) + v1(3)*v2(3);
cross(i,v1,v2) : select(i, v1(2)*v2(3) - v1(3)*v2(2),
v1(3)*v2(1) - v1(1)*v2(3),
v1(1)*v2(2) - v1(2)*v2(1));
fade(near_val,far_val,dist) = far_val +
if (16-dist, (near_val-far_val)/(1+dist*dist), 0);

bezier(p1, p2, p3, p4, t) = p1 * (1+t*(-3+t*(3-t))) +
p2 * 3*t*(1+t*(-2+t)) +
p3 * 3*t*t*(1-t) +
p4 * t*t*t ;

```

```

bspline(pp, p0, p1, pn, t) = pp * (1/6+t*(-.5+t*(.5-1/6*t))) +
    p0 * (2/3+t*t*(-1+.5*t)) +
    p1 * (1/6+t*(.5+t*(.5-.5*t))) +
    pn * (1/6*t*t*t) ;

turbulence(x,y,z,s) = if( s-1.01, 0, abs(noise3(x/s,y/s,z/s)*s) +
    turbulence(x,y,z,2*s) );
turbulencia(x,y,z,s) = if( s-1.01, 0,
    sgn(noise3(x/s,y/s,z/s))*noise3a(x/s,y/s,z/s) +
    turbulencia(x,y,z,2*s) );
turbulencb(x,y,z,s) = if( s-1.01, 0,
    sgn(noise3(x/s,y/s,z/s))*noise3b(x/s,y/s,z/s) +
    turbulencb(x,y,z,2*s) );
turbulencc(x,y,z,s) = if( s-1.01, 0,
    sgn(noise3(x/s,y/s,z/s))*noise3c(x/s,y/s,z/s) +
    turbulencc(x,y,z,2*s) );

    { Normal distribution from uniform range (0,1) }

un2`private(t) : t - (2.515517+t*(.802853+t*.010328))/
    (1+t*(1.432788+t*(.189269+t*.001308))) ;
un1`private(p) : un2`private(sqrt(-2*log(p))) ;

unif2norm(p) : if( .5-p, -un1`private(p), un1`private(1-p) ) ;

nrand(x) = unif2norm(rand(x));

    { Local (u,v) coordinates for planar surfaces }
crosslen`private = Nx*Nx + Ny*Ny;
    { U is distance from projected Z-axis }
U = if( crosslen`private - FTINY,
    (Py*Nx - Px*Ny)/crosslen`private,
    Px);
    { V is defined so that N = U x V }
V = if( crosslen`private - FTINY,
    Pz - Nz*(Px*Nx + Py*Ny)/crosslen`private,
    Py);

{
    *****

    Rayinit Additions made by S. Crone. 1997
    Calculation for automatic transforming flat bumpmap
    onto any flat surface.

    *****
}

{
    Tranformation for flat surface mapping
}

{
    Define Z axis
}

Zx = 0;
Zy = 0;
Zz = 1;

{
    Find temp Y axis and normalise its vector
}

res0x = Zy*Nz - Zz*Ny;
res0y = Zz*Nx - Zx*Nz;
res0z = Zx*Ny - Zy*Nx;

```

```

mag = sqrt( (res0x*res0x) + (res0y*res0y) + (res0z*res0z) );
resx = res0x / mag;
resy = res0y / mag;
resz = res0z / mag;

res2x = resy*Nz - resz*Ny;
res2y = resz*Nx - resx*Nz;
res2z = resx*Ny - resy*Nx;

{      Setup Matrix      }

mat_0_0 = resx;
mat_1_0 = resy;
mat_2_0 = resz;
mat_0_1 = Nx;
mat_1_1 = Ny;
mat_2_1 = Nz;
mat_0_2 = res2x;
mat_1_2 = res2y;
mat_2_2 = res2z;

mat_0_3 = 0.0;
mat_1_3 = 0.0;
mat_2_3 = 0.0;
mat_3_3 = 1.0;
mat_3_2 = 0.0;
mat_3_1 = 0.0;
mat_3_0 = 0.0;

{      Tranformation matrix      }

newPx = mat_0_0*Px + mat_1_0*Py + mat_2_0*Pz + mat_3_0;
newPy = mat_0_1*Px + mat_1_1*Py + mat_2_1*Pz + mat_3_1;
newPz = mat_0_2*Px + mat_1_2*Py + mat_2_2*Pz + mat_3_2;

isz = FTINY + abs( Nz ) - 1;

{      Use these values      }

tPx = if( isz, Px, newPx);
tPy = if( isz, Py, - newPz);
tPz = if( isz, Pz, newPy);

{      Calculation of 2d bump map perterbations.}

{

absx = abs( Nx);
absy = abs( Ny);
absz = abs( Nz);

Xp = if( absx - absy,
        if( absx - absz,
            0,
            if( Nz, Xpert, Xpert)
        ),
        if( absy - absz,
            if( Ny, -Xpert, Xpert),
            if( Nz, Xpert, Xpert)
        )
    );
};

```

```

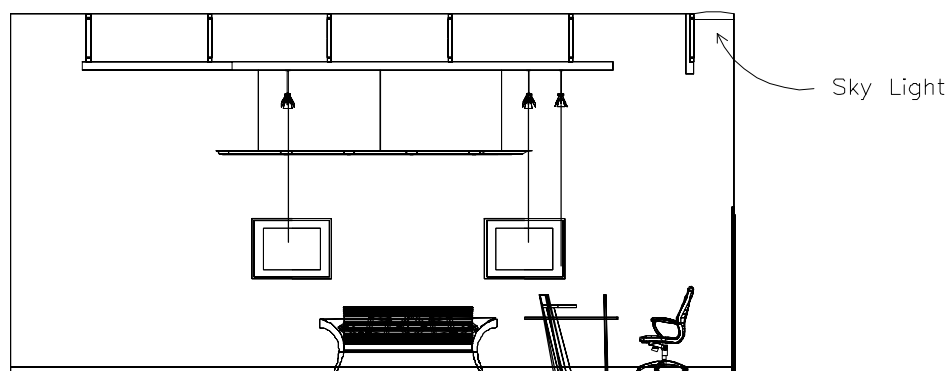
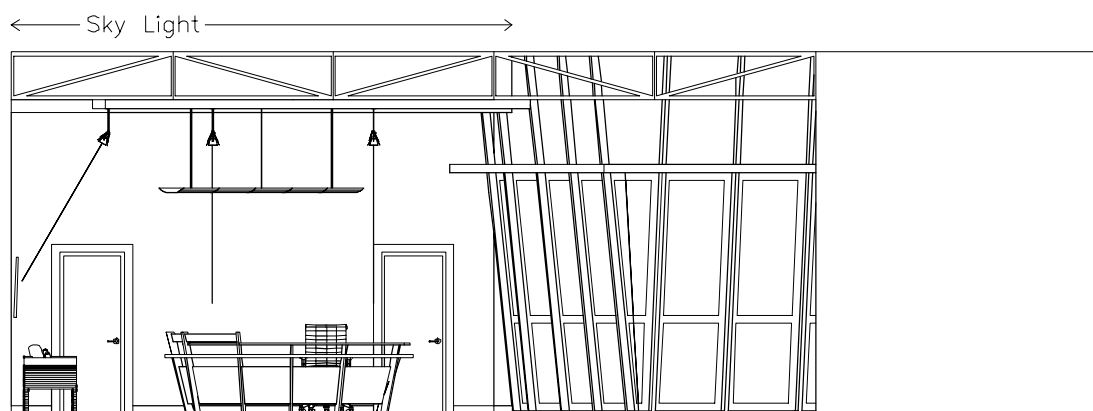
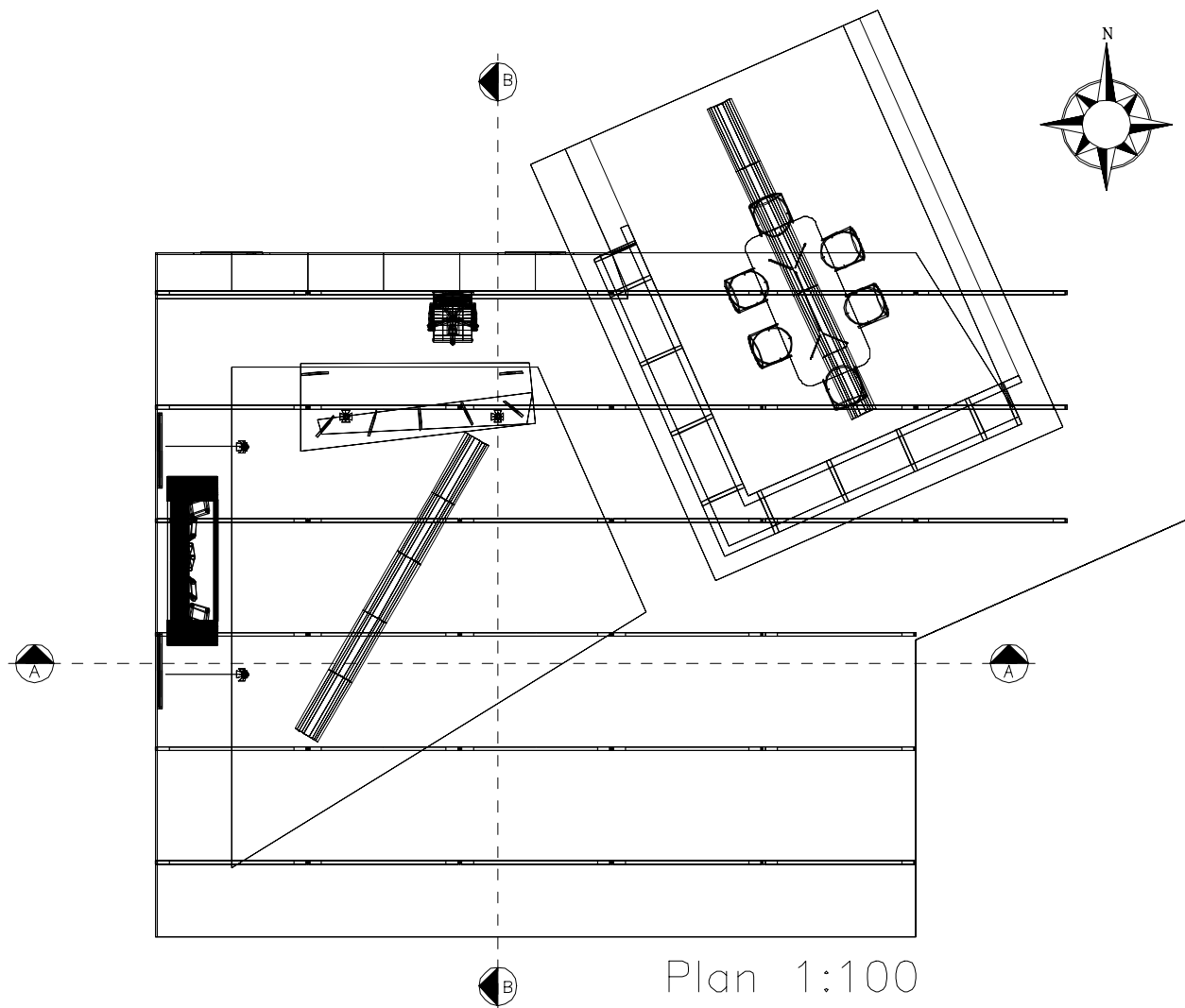
Yp = if( absx - absy,
        if( absx - absz,
            if( Nx, Xpert, -Xpert),
            if( Nz, Ypert, Ypert)
        ),
        if( absy - absz,
            0,
            if( Nz, Ypert, Ypert)
        )
    );

Zp = if( absx - absy,
        if( absx - absz,
            if( Nx, Ypert, Ypert),
            0
        ),
        if( absy - absz,
            if( Ny, Ypert, Ypert),
            0
        )
    );
}

```

APPENDIX E

This appendix shows sections and plans of the test room along with full colour quality prints of this room. It also contains some other higher quality prints of the images that are contained in the main body text of the thesis.



Plan and Section of Example Room