NOTE TO USERS

This reproduction is the best copy available.

UMI[®]

ALTERNATE LIGHTING DESIGN SOFTWARE:
A Tool for Cognitive Architectural Modeling

By

Ross Nicholson, B.I.D.

A thesis submitted to the
Faculty of Graduate Studies and Research
in the partial fulfillment of the requirements
for the degree of
Master of Architecture

Carleton University School of Architecture

Ottawa, Ontario, Canada

July 2003.

©2003, Ross Nicholson

Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-494-13427-5

Our file Notre référence

ISBN: 0-494-13427-5

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

ABSTRACT

Existing architectural and animation software capable of emulating constructed environments use the identical approach for placing artificial light sources into virtual models: import or define a light, install it into the model, render, *then* see what you get. An ‘experienced’ individual must perform a great deal of work before there is any visual confirmation of results. This prevents non-lighting professionals from using these tools. This thesis suggests a more intuitive methodology for the design of lighting software. It advances the idea that tools should reflect the design process: analyze, then decide what you want to see using the symbiosis of human visual feedback. The proposed method entails choosing the desired light effects visually from a selection matrix before they are added to the model, then viewing the result immediately via a fast-imaging, low-resolution proxy window. The software then places the matching hardware type and location corresponding to the light effect into the model and output data. Anyone can see what they want, then get what they see.

ACKNOWLEDGMENTS

Primarily, I would like to thank my advisor *Greg Andonian*. Without his trust, direction and support throughout this process the road would have been much more arduous, if not impossible. *Mike Getz* and his numerous bits of technical advice and help paved the way for a productive journey.

On the reasearch side, *Guy Newsham* and *Jennifer Veitch* from the National Research Council were of great help in obtaining background information, and I would also like to thank *Anca Galasiu* and *Dan Sander* for their help in studying Radiance and Lightscape.

Ivaldo Pasini and *Karen Pero* from Public Works and Government Services Canada made much of this effort possible through the use of their copy of Lightscape, which this study used as a considerable resource. Lightscape was used to generate most of the light-pattern matrices and renderings. Lightscape was also used as a model for part of the tool-set, as well as the graphical numerical output.

This thesis is dedicated to my mother, *Naomi Nicholson* (1929-2003). Without her support, none of this would have been possible.

LIST OF ILLUSTRATIONS

Figure 2.1- Use cycles for existing tools	15
Figure 2.2- Load a luminaire	21
Figure 2.1- Select a luminaire block	21
Figure 2.4- Choose the desired fixture characteristics	22
Figure 2.5- Verify the photometric file	22
Figure 2.6- Define the light characteristics	23
Figure 2.7- Create a luminaire instance in the model	23
Figure 2.8- Move the luminaire to the desired location	24
Figure 2.9- Initiate/reset the radiosity solution	24
Figure 2.10- Save the preparation file	25
Figure 2.11- Start the radiosity calculations	25
Figure 2.12- Stop the radiosity calculations	26
Figure 2.13- Change to elevation view	26
Figure 2.14- Adjust brightness to desired level	27
Figure 2.15- Check the perspective view	27
Figure 2.16- Create a room	32
Figure 2.17- The room created	32
Figure 2.18- Add a fixture to the schedule	33
Figure 2.19- Choose a photometric file	33
Figure 2.20- File info is displayed	34
Figure 2.21- Name the fixture type	34

Figure 2.22- The fixture is successfully added to schedule	35
Figure 2.23- Add a fixture to the model	35
Figure 2.24- Fixture successfully added to model	36
Figure 2.25- Select the area that you wish to analyze or render	36
Figure 2.26- Define the analysis grid parameters	37
Figure 2.27- Define desired output types	37
Figure 2.28- Calculations performed	38
Figure 2.29- The numerical output	38
Figure 2.30- Choose the perspective view	39
Figure 2.31- The rendered output	39
Figure 2.32- Radiance performs most actions through line commands	40
Figure 2.33- Examples of Radiance generated images	41
Figure 2.34- A screen capture showing a light's sphere of influence	44
Figure 2.35- The 3D Max interface showing the Create>Lights menu	45
Figure 2.36- The Form Z primary light specification menu	47
Figure 2.37- A secondary menu accessible under 'Options'	48
Figure 2.38- Form Z Radiosity parameters	48
Figure 2.39- The CosmoWorlds' light source adjustment handles	50
Figure 2.40- MAYA lighting interface	51
Figure 3.1- Ward's global illumination interface	56
Figure 3.2- LDV lighting options	59
Figure 3.3- More LDV lighting options	60
Figure 3.4- Design; best approximation; ray tracing	61

Figure 3.5- The 3D MAX viewport configuration menu	65
Figure 3.6- Open the file & select a view	68
Figure 3.7- Determine the next step	69
Figure 3.8- Jump to working view	70
Figure 3.9- Select a surface	71
Figure 3.10- Activate the wall light selection matrix	72
Figure 3.11- Select an effect type	73
Figure 3.12- Drag the desired effect to the surface	74
Figure 3.13- Observe the effect in the room	75
Figure 3.14- Drag the desired number of lights to the wall	76
Figure 3.15- The entire row of lights can be moved	77
Figure 3.16- The row can be moved away from the wall	78
Figure 3.17- Individual lights can be moved	79
Figure 3.18- Save a desired result as a solution	80
Figure 3.19- A new solution can include the one just created	81
Figure 3.20- Save the new composite solution to a new file	82
Figure 3.21- The new file shows included sub-solutions	83
Figure 3.22- Select a new working surface	84
Figure 3.23- Select the wall array selection matrix	85
Figure 3.24- Drag an entire row of lights to the surface	86
Figure 3.25- Observe the effect of the combined solutions	87
Figure 3.26- Render the achieved result in full resolution	88
Figure 3.27- Illuminance readings show the light levels	89

Figure 3.28- Any surface can be checked	90
Figure 3.29- Check the ceiling illuminance if desired	91
Figure 3.30- Increase the light levels	92
Figure 3.31- Decrease the light levels	93
Figure 3.32- Printed output from the solution file	94

TABLE OF CONTENTS

Title Page	i
Abstract	iii
Acknowledgments	iv
List of Illustrations	vi
Table of Contents	x
Chapter 1: INTRODUCTION	1
1.1 RATIONALE	2
1.2 FUNCTIONALITY & USABILITY	7
1.2.1 Mathematical method and algorithm used for calculation	8
1.2.2 Methodology and accuracy of texture mapping	9
1.2.3 Construction or modification of architectural forms vs. importing completed structures.	10
1.2.4 Use of photometric libraries of existing luminaires vs. created, idealized sources where the light characteristics are controlled by the user.	10
Chapter 2: THE EXISTING PARADIGMS	12
2.1 THE ARCHITECTURAL TOOLS	16
2.1.1 Lightscape	16
2.1.1.1 General Description	16
2.1.1.2 Usability/Interface/Use-cycle transparency	17
2.1.1.3 Toolset	17

2.1.1.4 Importing	18
2.1.1.5 Construction	18
2.1.1.6 Surface and texture mapping	18
2.1.1.7 Luminaire specification & placement	18
2.1.1.8 Light and Light Object Behaviour	19
2.1.1.9 Output	20
2.1.1.10 Overall	20
2.1.1.11 Typical use scenario	20
2.1.2 Lumen Micro And Genesys II	28
2.1.2.1 General Description	28
2.1.2.2 Use-cycle transparency	28
2.1.2.3 Toolset	28
2.1.2.4 Importing	29
2.1.2.5 Construction	29
2.1.2.6 Surface and texture mapping	30
2.1.2.7 Luminaire specification & placement	30
2.1.2.8 Light and Light Object Behaviour	30
2.1.2.9 Output	31
2.1.2.10 Overall	31
2.1.2.11 Typical use scenario	31
2.1.3 Radiance	40
2.1.4 Lithonia Visual	41
2.1.5 AGI 32	42

2.2 THE 3D ANIMATION TOOLS	42
2.2.1 3D Studio Max	43
2.2.1.1 General Description	43
2.2.1.2 Use-cycle transparency	43
2.2.1.3 Luminaire specification & placement	44
2.2.1.4 Light and Light Object Behaviour	45
2.2.2 Form Z	46
2.2.2.1 General Description	46
2.2.2.2 Use-cycle transparency	47
2.2.2.3 Luminaire specification & placement	47
2.2.2.4 Light and Light Object Behaviour	49
2.2.2.5 Overall	49
2.3.3 CosmoWorlds	49
2.2.4 MAYA	51
Chapter 3: CRITICAL DISCUSSION	52
3.1 ARCHITECTURAL LIGHTING SOFTWARE INTERFACES	55
3.2 PARALLEL STUDIES	58
3.2.1 Lighting Design Viewer	58
3.2.2 Painting with Light	60
3.3 PROPOSED SOFTWARE STRUCTURE	62
3.3.1 Assumptions & Caveats	62
3.3.2 Methodology- Key Concepts	62
3.3.3 The Tool Set	64

3.3.4 Sample use-cycle:	68
Chapter 4: ISSUES TO BE FURTHER EXAMINED	95
4.1 Lighting of irregular surfaces and 3D objects	95
4.2 Defining surface limits	95
4.3 Ceiling and floor applications	95
4.4 Complications from textures	96
4.5 Light object behaviour- visible sources	97
Chapter 5: CONCLUSION	98
5.1 Plug-in or program?	98
5.2 The Next Steps	99
5.2.1 Who Would Develop SeeScape?	100
5.2.2 Proposed Testing Methodology	101
5.2.3 The Challenges, Immediate and Future	102
5.3 Final Reflection	102
REFERENCES	104
BIBLIOGRAPHY	105
ON-LINE BIBLIOGRAPHY	106
SOFTWARE BIBLIOGRAPHY	107
PROFESSIONAL and RESEARCH CONTACTS	109
KEY DEFINITIONS	110
ADDENDUM	111

Chapter 1: INTRODUCTION

The use of computers to emulate envisioned architectural projects has become a de-facto standard industry-wide. There are two key components of realistic architectural depiction, a) the mapping of material behaviours onto surfaces in the model, and b), the representation of light sources and their inherent properties and observable interactions with those mapped materials.

The use of these architectural lighting tools has fallen into the domain of engineers and technicians, by default it seems, as virtually all software packages rely on specific knowledge regarding the behaviour of light. Most commercial applications also require a considerable background in lighting hardware (called luminaires in the industry). As a result, the manifestation of lighting designs, as well as the appearance and functions of the interfaces, have been developed around technical principals rather than intuitive application algorithms.

In order to determine the significance of what is being proposed here, it is necessary to compare and contrast the software tools presently used by design professionals in order to understand the variety (if any) of interface methodologies, and how those algorithms relate to intuitive and architectural mind-sets. As most of the use of lighting software involves the placing of artificial light sources (and this is virtually the only area that engineers and lighting professionals have as their domain), this study is limited to artificial lighting. It does not address sunlight and animation.

1.1 RATIONALE

Overall, there are two types of software used by different groups of practitioners. The first type is the architectural lighting tools, the second is the animation tools.

In the architectural group, users tend to be engineering technologists, the occasional P.Eng., and lighting equipment sales agents, who in the past, ironically, tended to do a great deal of lighting design. Motives for this group vary, as the technologists and engineers are generally concerned with numerical compliance with set Illumination Engineering Society standards, and the sales agents are concerned with providing a service to luminaire specifiers in order to obtain specifications for the products that they represent. Numerical compliance is also a driver, but agents have some motivation for producing believable visual results for use as a sales tool. Interestingly, luminaire manufacturers are the ones motivated to produce software usable tools, as it aids in the specification of their products.

Architects, interior designers and developers also use some of these tools, but are motivated by visual results, so likely to use one of the animation tools available because they require less proprietary and product-specific knowledge.

There is a small group of practitioners who call themselves lighting designers. These individuals often have both an engineering and architectural base to their practice. These people understand artificial lighting and its perceptual

consequences, but do not always use the software tools, often relying on back-of-the-envelope calculations and experience to derive their designs. Facility owners who have a vested interest in the project outcome most often employ these practitioners because they tend to look out for the owner's interests. The lighting designers fees are based on an hourly rate with a maximum upset and are therefore separated from the politics (read \$) of the tender process.

There are several important factors in the existing design/specify/build process that affects the lighting design and its effect on the architecture.

- Lighting practitioners are often not design-literate: as previously stated, much of the design is done by engineering technologists (a 2 or 3 year technically oriented college diploma) or sales agents with no particular design schooling.
- The financial framework in which engineers/technologists work: engineering firms, generally work off a percentage of building cost, thus there is a dollar-based disincentive to spending time on expensive virtual models or innovative designs. Iteration of previously used compliant systems is most profitable.
- The sales agent's goals: agents are also driven by the need to maximize profits. This is a factor because of the tender bidding process, and the electrical contractor's pressure on the agents to deliver less expensive (often inferior) products that will maximize contractor profits.

- Architects often do not grasp the importance of lighting as an effective means of strengthening design: architectural design education does not include rigorous focus on lighting as a design tool.
- Architects are seldom familiar with the hardware and technical aspects of lighting, as well as the virtual support tools: in today's era of increasing specialization, few architects are lighting hardware and software literate.

So what is the alternative? One would think that lighting tools should work along the lines of the lighting design process, and support that process. Well then, what is the lighting design process? Arguably, the process steps would be:

1. Analyze the architecture;
2. Analyze the client's programmatic needs;
3. Choose the surfaces that you wish to enhance to support the program and aesthetics;
4. Decide on the form of that expression;
5. Determine the location that equipment can be located in order to best achieve that expression;
6. Pick the hardware that will best suit the vision.

Subsequent iterations are likely, as the process is perceived as cyclical. The designer approach distills down to:

1. What do you want to see?

2. What will do that, and from where?
3. Adjust output to comply with programmatic parameters where necessary.

How does this approach compare to that used by the engineering mindset? The engineering approach could be summarized as:

1. Analyze the client's programmatic needs (maintenance often figures as a bigger driver than aesthetics here);
2. Find a fixture that will supply the right amount of foot-candles;
3. Install the correct number of fixtures to cover the space;
4. Verify that the calculated average illumination is within the programmatic parameters (best err on the high side).

This usually occurs without much or any iteration.

So, how could a 'What do you want to see?' lighting methodology work? This depends on what you want to achieve as a result. One could just 'spray' light on a virtual canvass (a kind of 3D Photoshop) so that the composition would provide a visual spec for a technical lighting practitioner to emulate. This requires two people to do the job, and would only really work if the 'painter' would have the specialized knowledge to only produce renditions that are achievable in practice. There is a study from Cornell University¹ in which software performs the implementation of the painter's 3D image, but that process has serious limitations (see 3.2.1 Parallel Studies).

A more useable system could rely on a predetermined library of effects produced by existing luminaires. In a fashion that would more closely emulate the design process, the procedure could be:

1. Pick a surface.
2. Pick a type of light effect or beam shape from an image-based selection matrix. The matrix would contain actual light distributions from a library of fixtures in specific typical mounting configurations
3. Drag it to the orthogonally viewed surface and place it as you see fit
4. Iterate, adjust, align, etc.
5. Hardware mounting positions are determined by manipulation of the effect.
6. The software provides a fixture type and location specification.

One could try to argue that this leaves no room for innovation. But innovation is generally the result of the combination and effective application of existing hardware anyway. Every composition would still be unique. It could be argued that at present, real lighting innovation could only really come from individuals that understand the hardware and behaviour of light. A truly useful tool for architects and other designers would be one that does NOT require this proprietary knowledge.

Why create such a tool, what would be gained by using this new method? For one, it would greatly simplify the creation-to-reality process. Architects would be able to directly oversee the interior and night-time manifestation of their spaces. Users would not require proprietary lighting knowledge. At worst, architects would end up with an unambiguous visual specification of their design intent to give the practitioners. In the best-case scenario, specifications and locations could be generated by the software. Architects, engineers and interior designers would benefit through greater understanding of what they are creating, as would the people who end up benefiting from a better-lit environment. Research has revealed that this approach has not been tried before, and anecdotal evidence suggests that this type of approach would be very positively received by lighting practitioners and architects alike.

1.2 FUNCTIONALITY & USABILITY

While the focus of this thesis regards issues of usability, it is impossible to do this without touching on many of the functional issues. This will become apparent as the various software tools are described. As most, if not all of lighting software research is done by engineering or computer science academics and professionals, the focus of this work has been on enhancing numerical accuracy, as well as 'believability'. Investigation has shown that interface issues are quite often only addressed after the software designers (engineers or computer scientists) get complaints from *other engineers and computer scientists*. A few relevant functional issues that affect usability are listed below.

1.2.1 Mathematical method and algorithm used for calculation

There are two basic methods of performing light calculations, ray tracing and radiosity. Ray tracing follows a 'reverse reflection', and follows a line from the observer eye into the virtual model, to the light source(s). In an animation, this means that each calculation is viewpoint-dependent. Some tools have a fudge-factor, which makes up for ambient inter-reflected light that usually assumes all surfaces are Lambertian (incident light is reflected off the surfaces in a diffuse, omni-directional manner). In spite of the repeated calculations for each observer point, this has been used in the past because it was not particularly computer-intensive for single views. It is not a particularly numerically accurate method, and shadows produced are more defined than in reality because there is no real inter-reflected light calculation. It does, however, produce the most realistic highlights off of specular or near-specular materials. When used in conjunction with radiosity, the results can be very convincing. Radiosity, however, accounts for inter-reflection and the light generated by objects that are luminous by virtue of the incident illumination hitting them from the primary light sources, as well as colour bleeding (see 2.1.1.1). Shadows have a softer, more realistic edge. There are various algorithms for this calculation type, but all are quite computer intensive. This every-day use of this type of calculation has only been made possible because of advances in computer power. The advantage is that calculations are independent of viewer position (as in real life), so an entire animation can use one set of radiosity calculations. Radiosity is more accurate,

but is only as good as the material and mapping information that is input. It therefore requires more accurate environmental information (surface textures and reflectances). This complicates the procedure required to produce a solution. Generally, the more realistic the desired result, the more input and layers of interface required to generate the images. Global illumination systems are capable of emulating both direct and indirect lighting by using radiosity and ray tracing. Photon mapping, an extension of ray tracing, accounts for direct and indirect lighting and produces great results, but is slow and too resource intensive for most applications.

So we see that much of computer based lighting research focuses on either numerically accurate calculation or visual believability through more detail. Given: one can quickly achieve results that are numerically accurate within 10% (which is good enough for most instances); and an artist can render a wholly believable, graspable image in terms of mood and ambiance showing very little detail. Then perhaps to create a more usable tool, we should be looking at *'What is required for an image be to be good enough?'* Perhaps a lesser degree of detail would be more usable, because quick imaging would allow quick, usable visual information of sufficient detail to make visually-based decisions.

1.2.2 Methodology and accuracy of texture mapping

When calculations of inter-reflections are performed, they are only as good as the information input regarding the surfaces that the light is reflecting off. So, not

only must the reflective and chromatic information be accurate with regards to the virtual material, it must also match the material used in construction *exactly* in order to be accurate. Of course, this too adds complexity to the interface.

1.2.3 Construction or modification of architectural forms vs. importing completed structures

The ability of software to deal with virtual construction varies greatly. Generally, the more complex and accurate the mapping and light calculation ability, the less flexible it is in terms of construction. The most believable lighting tools (i.e. Lightscape), have no construction ability at all.

1.2.4 Use of photometric libraries of existing luminaires vs. created, idealized sources where the light characteristics are controlled by the user

As previously mentioned, architectural systems employ actual photometric files of existing luminaires, while animation systems create idealized sources. The animation sources have both advantages and disadvantages when compared to actual source files. When constructing scenes, animation program light sources can exhibit non-real-world properties such as complete polarization, no reflected component or shadow creation, no inverse-square drop-off and the like. This can be quite useful when trying to create a composition (or 'scene') with a particular mood or effect with a predetermined look, especially if surrealism or theatricality is the goal. What these tools cannot do is create asymmetric distributions that emulate real architectural luminaires (i.e. wall-washers and the like), unless one

uses complex masks or bitmaps as projections. As yet, the animation tools are generally still not able to import IES format photometric files, though developers inform me that it is a goal. At the time of publishing this document, information indicates that the new release of Autodesk's VIZ 4 can handle IES format photometrics. This would be a positive development in terms of getting an all-purpose lighting tool, as the animation programs have enormous potential for producing believable images, and have intuitive construction methods built in.

The model proposed in this study does not address construction, material mapping or idealized sources, but focuses on a lighting design methodology relating to a use cycle that more closely emulates the design process.

Chapter 2: THE EXISTING PARADIGMS

In order to best demonstrate the issues at hand, this paper will briefly touch of typical use scenarios of the most widely used tools. All these tools were reviewed, with a focus being given to usability and use-cycle.

The Architectural/Engineering Model:

- Lumen Micro
- Genesys II
- Lightscape
- Radiance
- Lithonia Visual
- AGI-32

The Animation Model:

- 3D Studio Max
- Maya
- Form Z
- Cosmo Worlds (VRML)

As noted, there are two main models for creating realistic architectural lighting renderings. The first employs what is a basically an electrical engineering approach, which focuses on implementation of a known photometric behaviour from file. This type of software is commonly called (somewhat inappropriately),

and is more aimed at engineers and application specialists, as opposed to architects and interior designers. A good deal of the output is in the form of numerical or graphed information, and the focus tends toward numerical accuracy and not usability. This type of tool will be referred to as the 'Architectural Model'.

The use cycle involves importing a CAD .dxf or .dwg file (one can create the spaces in some of these programs, but they are usually crude and limited), then picking a fixture type from a known library or import. The fixture file is usually just the IES (Illuminating Engineering Society) photometric behaviour file, actual fixture/source test results in a standardized format. The fixtures are then placed in the ceiling (or on the wall, etc.), then calculations and rendering run and output. Generally, the more commercial the application, the more ordered and simple the use cycle. Many luminaire manufacturers author these tools and include their own proprietary databases in order to facilitate the selling of their fixtures. The logic and motivation being that the tool easiest to use will get the most use. However, this usually means that the program is quite limited, or unable to produce realistic renderings, even though the calculations may be nearly correct. Colour and texture mapping is not an option until you get into expensive packages like Lightscape. This, of course, makes the interfaces more involved. However, the architectural tools have lights that behave fairly realistically, as they consider interior reflections and contributions from other sources.

The second approach comes from the animation software packages (the 'Animation Model'). The focus of these programs is solid and environmental modeling, but the lighting components allow a lot of design freedom by virtue of the fact that the user can construct a source from scratch with the characteristics that they desire. Several idealized sources are available from a library with default parameters, but one must understand what those parameters are in order to successfully manipulate them. This is difficult for the lighting novice. As previously stated, one cannot import real-world IES files either, and asymmetrical distributions are not possible to construct, even though they are quite common in real-world applications. The animation tools allow the user to tweak a view so that it can look extremely realistic, but only through intuition, trial and error. Further, the parameter values are often relative, with arbitrary numeric parameters. Animation interpolation can also lead to nonsensical things like negative intensity values. Chromaticity is usually not addressed other than in RGB terms. There is usually no way to emulate specific spectral content.

Another, more troubling concern, is the fact that the spaces do not appear as they would in a real life situation. Inter-reflections and far-source contributions, which are pronounced in real life, are generally not shown as they would be perceived. For instance, a lone narrow spotlight in a small space will not provide light to the far side of the room, even though in real life this would actually be the case. The user has to add other sources with different characteristics in order to emulate what would be the real effect.

Figure 2.1- Use cycles for existing tools

2.1 THE ARCHITECTURAL TOOLS

The goal of these analyses is to determine which underlying processes and methods best facilitate the process of adding lighting to an architectural model in an intuitive and easily understood manner. For each piece of software examined, the identical process was performed: import a DWG/DXF file of a bare room; install a single light fixture; observe the visual results.

2.1.1 LIGHTSCAPE (PC only)

2.1.1.1 General Description

Lightscape had overtaken Lumen Micro in past years as the tool of choice among lighting professionals, primarily because of its architectural rendering capabilities. Only recently has it had any competition from AGI-32 and its companion rendering system, Helios. It produces very believable images by implementing both radiosity and ray tracing on top of surface and texture mapped materials that have specific reflective or refractive properties. Lightscape's modelling of colour bleeding is very realistic. Colour bleeding is the phenomenon that occurs when the reflected light from an incident beam is chromatically altered by the selective absorption of the material that it has hit. For instance, if white light hits a red wall or floor, the green part of the spectrum is absorbed. The light that is reflected is now minus the green, so it appears red. This red light in turn would give a red hue to white objects in the room, particularly those close to the red surface. While the visual results are unquestionably impressive, the software is not easy to use for the novice. A typical use cycle is shown later in this section.

2.1.1.2 Usability/Interface/Use-cycle transparency

There are no real clues from the interface as to how to go about the business of creating a realistic model. Most steps are highly procedural, with no overt indication of what to do next. Even a lighting professional with a computer-use background cannot get results without paying close attention to demo examples and a users' manual.

One initially confusing (and vexing) aspect of the use cycle is the fact that the room and fixtures must be manipulated in a preparation file, while actual visual solutions are rendered in a solution file. This is problematic because luminaires and their locations cannot be edited in the solution file. This means that if you don't like a rendering result, you must go back to the preparation file to edit, then create a new solution file.

2.1.1.3 Toolset

Lightscape implements a host of cryptic icons that will yield a text description to the patient. There are also redundant pull-down menus for some options. The biggest use difficulty is in knowing what order to proceed in, coupled with fact that each icon often reveals a phalanx of other parameter windows that lead down many levels.

2.1.1.4 Importing

Lightscape accepts DFX, DWG, Lightwave and 3D Studio formats. Many, if not most, of other non-mapping or non-rendering lighting programs export in Lightscape format, as it has become an industry standard.

2.1.1.5 Construction

The complete lack of construction tools is considered by some to be the biggest drawback to Lightscape. If one were to prepare a AutoCAD drawn DFX in Lightscape, map all the materials, add the lights etc., then desire to change an architectural feature, the process would indeed be laborious. For instance, if you wanted to go from a pitched to vaulted ceiling, you would have to delete the old ceiling in Lightscape, then construct the vault in AutoCAD, import it into Lightscape, then re-map every surface on it. Then you would have to re-install every light fixture that was in the ceiling before proceeding to a radiosity solution and/or ray tracing.

2.1.1.6 Surface and texture mapping

Good results are possible, but only through a laborious, non-intuitive process. Lightscape is more understandable than 3D Max, but much harder to grasp than Form Z.

2.1.1.7 Luminaire specification & placement

The required procedural steps are *not* transparent or intuitive and are therefore a potential source of great vexation for the uninitiated. Many of my difficulties involve assembling data together in 'Blocks' that can be used in Lightscape. A luminaire *block* would consist of its physical shape and colour etc. The luminaire's photometric behaviour (how it distributes light), and its observed photometric behaviour (how it looks in terms of brightness etc.), are added as separate parameters to the block to create a luminaire. The photometric behaviours are easily changed, the block is not. Arrays of fixtures can be created when a large field of luminaires is required. The arrays are created through a numerical input with no visual confirmation of the outcome until it is placed, although the array can be edited.

2.1.8 Light and Light Object Behaviour

The observable light effect is very realistic, but only if the overall brightness and contrast settings are adjusted properly. These properties are manipulated under the file menu, but one cannot set a default value other than the factory setting, which is too dark for most applications. To repeatedly use settings other than the default values, one must save and then consistently re-load the desired parameters. The numerical values of illumination may be accurate, but a knowing eye is still required to produce a believable image. The representation of luminaire behaviour (fixture brightness) is only an approximation of the average brightness, but this fault exists throughout most tools of this type. The best visual

results are achieved through using both the radiosity solution and ray tracing together.

2.1.1.9 Output

Statistical data, iso-lux maps, luminance values (actually one of the more useful bits of data), VRML, panoramic views and the usual host of pixel based formats for rendered images.

2.1.1.10 Overall

Lightscape produces believable results, but is initially hard to use. It has no construction ability and the two-stage solution process mean that use is time-consuming and highly procedural.

2.1.1.11 Typical use scenario

The following pages (Figures 2.2-2.15) describe a simple use scenario that depicts the procedures required to place a light into an existing model of a room using Lightscape. These figures can be viewed on the accompanying ADDENDUM CD ROM by double clicking the file LIGHTSCAPE.PPS.

Figure 2.4- Choose the desired fixture characteristics

This list is from a manufacturer's database. One must be familiar with the characteristics of each luminaire in order to choose what effect you desire.

Figure 2.5- Verify the photometric file

Each pre-constructed block may come with a file that represents the photometric behaviour of a particular fixture, but the same block can be used with many behaviour files. One should check which file comes with the block.

Figure 2.6- Define the light characteristics
Apart from choosing an IES format photometric file, colour and intensity can be adjusted.

Figure 2.7- Create a luminaire instance in the model

Figure 2.8- Move the luminaire to the desired location

Any move must be accomplished with a right-click or menu command.
The fixture appears at the XYZ origin and must be translated along all axes to be placed.

Figure 2.9- Initiate/reset the radiosity solution

Once ready the radiosity solution must be started. Parameters can be accessed through a menu driven window.

Figure 2.10- Save the preparation file

This step is confusing at first. A new 'solution' file must be started to calculate the radiosity solution, and the preparation file saved if you ever want to edit the model.

Figure 2.11- Start the radiosity calculations

The new model showing the radiosity divisions appears.
The model is no longer editable.

Figure 2.12- Stop the radiosity calculations

You can stop the calculations at any point before 100% of the photonic energy is expended. No further visual changes are usually apparent after ~95%. The green outline denotes which sector is undergoing calculation.

Figure 2.13- Change to elevation view, then adjust properties

You need a view that will show your results. Images are generally too dark to see using only a few fixtures at the default setting of 50% brightness.

Figure 2.14- Adjust brightness to desired level

A setting of 100 achieves a believable brightness. Contrast and ambient fill-light can also be adjusted to achieve 'believability'.

Figure 2.15- Check the perspective view

The results achieve a high degree of realism with no texture mapping, but Only now do you see what you get. However, one must return to the preparation file to add or move fixtures.

2.1.2 LUMEN MICRO AND GENESYS II (PC only)

2.1.2.1 General Description

These two programs are virtually identical. Genesys is a licensed version of Lumen Micro and supposedly is more stable. These are clearly tools for engineers and lighting application professionals. Results are centred on numerical output, although single image visual modelling with basic colours is easily delivered. Lumen Micro used to be *the* standard tool for lighting professionals, though that honour has fallen to Lightscape in recent years because of its more believable output as well as importable architectural files.

2.1.2.2 Use-cycle transparency

This software has one feature that definitely gives it an advantage in terms of usability. A procedural checklist shows what order to proceed in. Further, when you click on a function a linear series of windows leads you through the operation. If that operation is completed successfully, a check mark appears beside the function and you can continue. Critical and optional functions are differentiated through visual text hierarchy. Unfortunately, there are still a couple of glitches in the procedure, but over all a quite usable package.

2.1.2.3 Toolset

Undistinguished generic-looking PC fare for construction and light installation. The Genesys and Lumen Micro tool sets may appear slightly different at first blush, but their over functionality is the same.

2.1.2.4 Importing

Lumen Micro accepts DXF and DWG imports with few problems, except that even though these solid are visible in the wire-frame model, they do not show up in the renderings or calculations. Apparently the import function is solely for the purpose of providing a guide for construction within the program. This is a major flaw, as there is a specific DFX import function, yet no indication at all that the DFX is unusable. The refusal of the checklist to acknowledge the dxf import appears inexplicable to the novice. Constructing a dummy room around the imported one at first appears to solve the problem. There are no further indications of trouble until you try and render, and done of the imports show up. The only way to determine if there are scaling issues is through observation of the X and Y coordinates at the bottom of the screen.

2.1.2.5 Construction

Construction capability is one of the major drawbacks of this software. It is only capable of generating the most basic rectilinear rooms and solids. Even diagonals are beyond its scope, not to mention curvilinear elements (a zig-zag line of columns is required to emulate a diagonal wall). Rooms are created in entirety in one dialog box with properties assigned in the same box. Basic window elements, doors, furniture etc. are in a library, and easily accessed. However, form is very limited and you cannot import usable furniture elements.

2.1.2.6 Surface and texture mapping

The other great drawback of these tools is the inability to assign material properties to surfaces and objects. Surface colours and reflectances are assignable.

2.1.2.7 Luminaire specification & placement

Creation and placement of light sources is a fairly simple operation, as is their modification. The only problematic issues arise when building exterior pole assemblies with single or multiple luminaires. IES format photometric files of existing fixtures must be imported and added to the fixture schedule. This process of creating a schedule is a common aspect of lighting design. The procedure and window used for array creation allows visual control of spacing, orientation and placement within the room.

2.1.2.8 Light and Light Object Behaviour

Lumen Micro and Genesys II produce a fairly believable rendering, in spite of the lack of textures. Shadows are fairly realistic, with softness and hardness delivered in the appropriate circumstances. One drawback common to any of the programs is that the grid size for shadows and light is often visible as a checkerboard edge when the program is set up for quick solutions. The grid size is adjustable, but calculation time goes up with increased resolution. The output intensities are real world, but spectral content is confined to RGB white. The luminaires themselves appear as simple boxes when suspended, but black voids

when placed in the ceiling. The issue of fixture luminance its and effect on perception of the space is not addressed at all.

2.1.2.9 Output

These programs generate bit-mapped renderings, but their focus is on numerical analysis. The bulk of output centers on printable reports detailing averages, min/max conditions etc. There is an entire subsection of the software that deals with lighting economics and power management. However, even to render, the user has to specify a calculation grid and plane before running the calculations.

2.1.2.10 Overall

An excellent tool for quickly assessing lighting options and numerical results in simple structures. The positive features are things like the procedural checklist, which is an excellent use guide. Creation of form and lighting sources is obvious and intuitive, and the addition of basic room elements is very easy. Unfortunately, texture mapping is not possible, solid construction is extremely limited, and .dwg imports are invisible to the calculation process and are therefore not usable.

2.1.2.11 Typical use scenario

The following pages (Figures 2.16-2.31) describe a simple use scenario depicting the procedures required to place a light in a model using Genesys. These figures can be viewed on the ADDENDUM CD ROM by double clicking GENESYS.PPS.

Figure 2.16-Create a room

Creation of rectangular spaces is quite easy, but is done numerically.

Figure 2.17- The room created

The check-list shows that you have successfully created a room, and that the next step is to add a fixture to the schedule or create a calculation grid.

Figure 2.18- Add a fixture to the schedule from a photometric database
 One must know ahead of time which photometric files represent which fixture/lamp combinations and their associated characteristics.

Figure 2.19- Choose a photometric file
 There are no graphic clues as to what you are choosing.

Figure 2.20- File info is displayed

Only here do you get a representation of what you are installing.
It requires some experience to understand what you are seeing.

Figure 2.21- Name the fixture type

Figure 2.22- The fixture is successfully added to schedule
Note the check-list.

Figure 2.23- Add a fixture to the model

Figure 2.24- Fixture successfully added to model

The fixture can be dragged to the desired location,
the Z-axis default is the ceiling plane.

Figure 2.25- Select the area type that you wish to analyze

The limits of the space that is to be analyzed must be defined.

Figure 2.26- Define the analysis grid parameters
 The default settings are acceptable for most applications.
 The denser the grid, the longer the calculations take.

Figure 2.27- Define desired output types
 The limits of the particular calculation must be set.
 One can also set the output to simply produce numerical results,
 or render if you have the time.

Figure 2.28- Calculations performed

Notice how the check-list lets you know the system status. The coloured lines show areas of equal illumination. The line thresholds can be adjusted to your needs.

Figure 2.29- The numerical output

Info such as illumination average, maximums and minimums are displayed first when results are queried.

Figure 2.30- Choose the perspective view

Figure 2.31- The rendered output
 Note the saw-tooth edge to the light pattern due to a default low resolution setting for the radiosity solution.

2.1.3 RADIANCE (PC only)

Radiance is a program for scientists, developed and implemented by scientists. There is absolutely nothing about Radiance that could be considered user-friendly. The sole purpose is to deliver *accurate* data and images that emulate reality as closely as possible. This software is strictly a texture/material mapping tool that uses a complex radiosity algorithm and ray-tracing. Its functionality is impenetrable to the uninitiated. Radiance performs most actions through line commands, and reportedly takes months of daily use to master.

Regardless of its superior numerical results and image believability, it's a good example of how not to design an interface. Few aspects of input are GUI related. For instance, to move a chair in a room, one would have to find the data file that contains the location and orientation parameters (amongst a host of other un-referenced numbers), and alter those numerical values.

Figure 2.32- Radiance performs most actions through line commands

Radiance imports photometric files from existing luminaires, as well as a physical appearance file and a photometric file which shows how the luminaire itself appears in terms of brightness and glare. This is a somewhat unique capability. Visual results may be good, but this is a program that all but the most dedicated and stout of heart should avoid.

Figure 2.33- Examples of Radiance generated images

2.1.4 LITHONIA VISUAL (PC only)

This software is fairly generic in its approach, and lies clearly in the engineering tool category. As a result, it doesn't warrant much discussion. Only a few points are worth mentioning:

- construction of many (but not all) complex forms and extrusion are possible;
- it has no surface and texture mapping capability;
- it has no visual component to output other than isolux graphing and wire-frame — ironically 'Visual' is strictly a numerical analysis tool;
- it exports to Lightscape, so it can work as a build-the-structure-and-luminaire-file preparation tool.

2.1.5 AGI 32 (PC only)

Although touted by the engineering set as the next great thing, AGI 32 could well be taken as yet another unremarkable program of the engineering variety. Of note, there is a glare rating section for the exterior lighting, which might prove useful to someone with experience, but a feature more relevant to this study is the AGI 32 approach to tool bars. *Every* function has an icon as well as redundant pull-down menu access. Tools are grouped according to function typology. This actually makes for a very busy interface that is initially incomprehensible to the novice. However, an experienced user could tidy up the screen and only have the tool group being used visible. These tool groups are quite portable and can be re-oriented to suit the user. A feature that has enhanced the popularity of this tool is its versatile ability to handle reports with graphic and numeric content. It will summarize photometric and run-cost analyses in reports that can be structured to one's liking. Helios, the sister rendering program, delivers highly believable results that compare favourably with Lightscape.

2.2 THE 3D ANIMATION TOOLS

The tools studied in this section perform in a similar fashion to the architectural tools with respects to most issues, the exception being lighting, so there will be no detailed observation of characteristics other than those lighting related. As a general note, however it was observed that the animation tools were generally superior to the architectural programs in their ability to construct and manipulate

simple or complex solids with unambiguous faces (no one-sided surfaces). As discussed, these tools utilize generic light sources with assignable parameters. This section will briefly summarize the the light generation capacities of the programs of note.

2.2.1 3D STUDIO MAX

2.2.1.1 General Description

3D Studio Max (and the architectural version, Studio Viz) fall into the category of animation tools primarily because the object of the software is to produce animation with complex, non-rectilinear surfaces. Construction of regular solids is extremely simple, but more complex things such as 'loft objects' and surfaces constructed from the combination of planar and linear elements requires much more study and practice to use. Lighting elements are simple to create and modify, but their behaviour is not completely realistic. Realism can be achieved in scenes if the user has a good eye and is familiar with the software. The lighting is achieved through ray tracing, although less sophisticated shadow maps can be used to minimize rendering time on lengthy, complex animation sequence rendering.

2.2.1.2 Use-cycle transparency

Basic solid construction is very simple and obvious affair, as is modification. Creation and aiming of light sources is also quite understandable without instruction. Most other aspects are more problematic and require diligent

attention to user manuals. However, rendering images is quite a straightforward procedure because only ray tracing is used.

2.2.1.3 Luminaire specification & placement

Creation and placement of light sources is a fairly simple operation, as is their modification. Create>Lights> then specify the desired photometric character: omni-directional, spotlight (with or without targets) and directional (polarized). Application of arrays is a bit problematic, as there are no visual clues as to orientation and offset. Data input is completely numerical. As is usual with animation software, one cannot import real-world photometric data.

Figure 2.34- A screen capture showing how 3D Max displays a light's sphere of influence

Figure 2.35- The 3D Max interface showing the Create>Lights menu (right).
 There are a lot of parameter adjustments if you push the menu up or down, but the only indication is the thin black strip on the right of the menu.

2.2.1.4 Light and Light Object Behaviour

The most difficult aspect of creating realistic looking renderings is that the sources do *not* behave in a realistic manner. The ray tracing method does not account for inter-reflections in a manner that is believable. Shadows are hard and pronounced, and scenes generally appear much darker than would be the actual case. Obliquely illuminated surfaces in particular appear much darker than expected. The scope of affectation of a light source is defined by numerically input values, so scaling issues do crop up. Intensity units, however, are arbitrary scalars that have no real-world counterparts, so trial-and-error is the only method of implementation. Colour is controlled by RGB values only.

When animating scenes with dynamic lighting conditions, the interpolation algorithm assigns values that are often irrational (i.e. negative illumination values), so familiarity with the software is required in order to get around several default functions.

As with many similar programs, the placed light sources are invisible, and any desire to reproduce the visual appearance of sources themselves can only be achieved through an entirely separate modelling process. Emulation of glare and veiling reflections would require intimate knowledge of the tool, and probably some post-production effects.

2.2.2 FORM Z (Mac and PC)

2.2.2.1 General Description

Form Z is the only Mac based tool of any significant note, and is a sort of hybrid, as it functions much like the other animation tools except that it is capable of radiosity solutions as well as ray-tracing. Biases aside, as a Mac tool it is more intuitive to operate than many of the PC tools. Complex forms can be developed fairly easily, and quick ray-trace renderings are easy to achieve. Radiosity solutions are slightly more involved, but achieve surprisingly good results using a model with almost no mapping.

2.2.2.2 Use-cycle transparency

Most functions can be achieved without referencing a users manual, an uncommon feature of clarity in design. Less obvious is the setting up the radiosity solution, because there are some non-intuitive steps involved.

2.2.2.3 Luminaire specification & placement

Form Z uses sun and generic light sources, but the specification of artificial sources allows a good deal of control over the intensity (watts, candelas or lumens), area of affectation, and distribution as well as colour temperature. Sources can be grabbed and moved easily, or turned off. Lamentably, light source data cannot be imported.

Figure 2.36- The Form Z primary light specification menu

Accurate Intensity : Point Light

☒ **Radiometric**

Input Power	100	Watts
Output Efficiency	50	%

☒ **Photometric**

☒ Color Temperature	3500	Kelvin
Brightness	400	Candelas ▼

Cancel
OK

Figure 2.37- A secondary menu accessible under 'Options'

Radiosity Options

Initialization

Sources
▬
▬
 60 %

Patches
▬
▬
 60 %

Polygons : 3 Times # Of Patches

Estimate Mesh Density...

Light Intensity All Accurate ▼

☒ Generate Shadows

☐ Render Direct Illumination

☐ Include Area/Line Lights

☐ View Dependent Solution

☐ Resolve Intersections

☐ Ignore Object/Face Meshing Attribute

☒ Smooth Shading Options...

Parameters: Always Smooth ▼

Sampling

Minimum # 3 ☒ At Vertices

Maximum # 6 ☐ At Center

Quality Better ▼

Preset Parameters 6 ▼

Generation

☒ Adaptive Meshing

Density
▬
▬
 65 %

Threshold
▬
▬
 25 %

Cutoff
▬
▬
 0 %

☐ Continuous Meshing

☐ Decrease Meshing

☒ No Meshing After Processing Lights

☐ No Meshing After 100 Iterations

Terminate After

☒ 80 % Of Light Absorbed

☐ 500 Iterations

Current Statistics

Absorption 0 % Sources 0

Iterations 0 Patches 0

Render Time 00:00:00 Polygons 0

Process Time 00:00:00

☒ Apply Reflected Light As Ambient

Preview Options...
Cancel
OK

Figure 2.38- Form Z Radiosity parameters

2.2.2.4 Light and Light Object Behaviour

Form Z has numerous rendering modes. Scenes can be viewed as wire-frames, simple colours, shaded, ray-traced and as radiosity solutions. Shadows and light are very realistic when radiosity is used, even without surface enhancement. Softness and hardness can be delivered when stipulated. The light sources themselves do not appear in the renderings, but can be emulated without too much difficulty. The only real drawbacks are the inability to import photometric behaviours and the fact that there is no library linear (i.e. fluorescent) source. Linear sources have a diffuse shadowing effect, unlike point sources. With a bit practice (particularly with anti-aliasing parameters), believable images are within grasp.

2.2.2.5 Overall

This program has perhaps the best combination of usability and functionality of all tools reviewed. Creation of form and lighting sources is reasonably obvious and intuitive, and light behaviour can be *very* realistic when radiosity is used. The drawbacks include the fact that photometric files cannot be imported and there are no fluorescent (linear) sources in the idealized source lighting library.

2.3.3 COSMOWORLDS (PC only)

CosmoWorlds is a web-centred VRML construction tool that was briefly examined to see if it had anything to offer in terms of interface and usability. Shapes are easily created (only VRML files can be imported) and manipulated,

as are light sources. Manipulation handles allow for the ability to almost instantly pick and arbitrarily re-orient objects and light sources.

Realism though, is not a strong point as far as lighting goes. The default background source cannot be turned off, so highly contrasting scenes are not possible. There is the usual supply of omni, cone and polarized sources.

Figure 2.39- The CosmoWorlds' light source adjustment handles

2.2.4 MAYA

Similar in configuration to 3D MAX, MAYA is at the cutting edge of animation software. Incredible degrees of realism can be achieved through painstaking multi-layered texture mapping and light configuration. However, even the version about to be released (5.1) cannot utilize actual IES photometric files. Creation, placement or aiming of light sources is intuitive and simple, yet they are constrained to being symmetrical.

Figure 2.40- MAYA lighting interface

Chapter 3: CRITICAL DISCUSSION

From a usability standpoint, it is fairly obvious that architectural lighting tools are not as advanced as some of the highly intuitive visual-design tools. For instance, one could argue that the tasks in creating a video in Adobe Premier are as complicated, or more so, than what has been studied here, yet that process is not nearly as onerous due to the intuitive nature of the interface. The same can be said about Macromedia Director usage. Yet use of these tools can be learned quickly and intuitively, without much proprietary knowledge.

Why do we not see this type of usability seeping through to a field such as lighting design? It's all in the numbers, industry sources say. There just isn't the size of market niche required to support and focus the required resources on studying and improving these particular types of products. One could suggest that market size is only part of the problem. The other side of the issue probably relates to the types of people who use lighting products. For instance, graphically literate professionals *insist* on graphically intuitive tools, because that is how they think. The tool reflects the mindset. The groups of people who create graphic tools would also have to have such a background in order to know what tools to make. They would ask the question, 'as a visual design person, what do you want to do?' Lighting practitioners, on the other hand, are for the most part engineering and science based. They might be more inclined to think in a more linear, procedural way, and have less issue with reading a manual, following detailed instructions etc. I would hazard a guess that the software development

would be directed by people who are even more linear and procedural in their thinking. They might well be more concerned with the results and numerical accuracy than usability. What about the individuals that want to explore lighting that are not of the linear, procedural mindset, say architects and interior designers? The tools must be re-thought for these people.

Almost all of the software studied violated the benchmarks for usability proposed by Tognazzini² and Neilson³ in some capacity. The most flagrant infractions usually involve not keeping the process transparent, and lack of system status indicators. The user could easily be faced with a blank screen and no sense of place in the model or process.

The most problematic of the tools studied come from the realm of Architectural Lighting software. As mentioned earlier, this is really a misnomer, as it should really be called Illumination Engineering software. Ironically, for all the focus and effort put into accuracy and endless parameter variables in the high-end programs like Lightscape, the visual results derived from system default values are often not believable from an experiential standpoint. Individuals with lighting experience and a fairly keen reckoning of 'how-much-light-looks-like-what', often take issue with the software renditions. Complex interfaces with countless inter-dependent variables usually delivered the least believable images in the hands of the uninitiated user. Ironically, the less complex programs like Genesys give a much better quick rendition of how luminous the environment might appear with

default, low resolution settings. This is achieved with results that ignore textures but not reflectances.

Some of the believability issues are driven by technological limitations. The dynamic range of brightnesses that are visible simultaneously to the human eye are not reproducible by conventional VDT or LCD screens, the contrast is too high. This also affects issues regarding emulation of human eye/brain responses to glare. Temporal adaptation (or lack thereof) to fixture luminance and source chromaticity at various lighting levels may be hard to universally quantify and reproduce, but perhaps these phenomena could be believably fudged. Any step that lessens the reliance on specific technical information has the potential to increase the use of lighting software by architects and those with non-lighting backgrounds. Arguably, anything that increases the awareness of lighting issues could be considered beneficial the field.

How do we widen the scope of understanding? Do we group tools by what they do, or by what they're doing it to? For example, is it easier to grasp tool or command use by dividing them into things that move, modify and create, or should one separate architectural, lighting and image production functions? How do people recognize and break down tasks naturally? It probably depends on the type of person you are, and the type of task. Are there any mental mechanisms that are employed by a large cross-section of people for specific tasks? Are we looking at the problem the right way?

Usually the best way to deal with the difficulties of these operations is to step back. What is it that we want to accomplish with these tools? If the scientific and engineering communities desire a high degree detail, then they will focus in that direction. Perhaps those with a non-technical background could use is a tool more tuned to their needs. A more goal-oriented approach towards specific needs seems warranted.

The architectural *and* animation programs pretty much function along the lines of, 'Install this, see what we get.' Again, perhaps we should be looking at the question, 'What do we want to see?' and proceed from there. One could narrow the choices to things that can be achieved using known equipment, and visually choose the desired effect from a finite array of results for the relevant application. This type of mental shift could mean a different sort of interface with regards to lighting, with visual examples for narrowing choices and procedures. This approach might open a great can of worms with regards to conflicting sub-parameters, and conceivably, a great deal of confounds could arise from the behind-the-scenes calculations and coding. But tools can be simple on the outside, regardless of the complexity of the invisible inside.

3.1 ARCHITECTURAL LIGHTING SOFTWARE INTERFACES

There is almost no existing research on lighting software interfaces or usability issues to be found. Sources indicate that there are 1,893 papers, articles, books, technical reports, and theses available on radiosity and global illumination. Of

these publications, the *only* uncovered research that addresses interface design at all is Greg Ward's 1995 paper, 'Making Global Illumination User Friendly'⁴. The author of that work, makes reference to the Radiance web-site, where there was exactly one element relating to interface graphic content and cognition (see Figure 3.1).

Figure 3.1- Ward's global illumination interface

The preceding figure constitutes the entire graphic content of the paper. In fact the paper content focuses entirely on one small aspect of a particular mathematical parameter issue relating to Radiance. Ward's paper does indeed illuminate (?) some of the issues faced in trying to create believable, accurate results with a process that becomes exponentially complex as the solution method converges on emulation of reality. Ward points out the futility of addressing non-intuitive parameters for a non-scientist or expert, especially when they are presented in a non-intuitive form. His discussion points out these more global issues, and general ideas on approaching them, but his proposed solution

is for a small specific problem and is not an over-reaching, holistic solution in terms of mental models.

The apparent absence of research relating to architectural lighting interfaces seems surprising. Industry people, however, seem to be of the opinion that the target market is too small to warrant expensive research. The marketing numbers do not seem to be in the public domain, but it seems that a lot of attention is being paid to numerical correctness and the interfaces are all designed in-house by technicians.

There are, of course, countless research papers on web interface usability, and many of the principles remain the same. In particular, Tognazzini's² and Neilson's³ guidelines for interface design seems the most relevant. The most salient of usability issues that apply to this study are:

**Recognition rather than recall- what do these numbers mean?*

Lighting software relies on the import and use of manufacturer supplied files represented by a number only. There is no visual basis by which to make design decisions regarding luminaires.

**Transparent and intuitive procedural steps- what do I do next?*

One of the failings of much of existing goal-oriented software is that it fails to keep the user aware of what they might do next. With a potentially purely creative tool like Photoshop or Illustrator, this is not relevant. But with lighting, there is

certainly a degree of procedure that must be followed in order to achieve defined results. If the process cannot appear intuitive (which is possible with a user that is not familiar with lighting), there should at least be an instructive element to the interface. Some of these issues will become more obvious in the step-by-step demonstrations illustrated later on.

**Awareness of system status- blank screens, navigation within virtual environments.*

One of the most vexing of usability issues with 3D modelling is navigation within the model. Whether in wire-frame or solid mode, a zoom or pan can leave you in limbo-land with a blank, black screen and no clue as to where you are. Auto zooming to selected surface extents in the other (multiple) views would solve some of this problem. A visual or text indicator that points the user to the origin and lets them know what the scale and field of view is would be of great help.

What is not overtly mentioned in the many sources on interface design – maybe it's too obvious – is that there is perhaps an efficiency and clarity to having a goal-oriented process reflect the mental process that drives the design.

3.2 PARALLEL STUDIES

3.2.1 Lighting Design Viewer

The Indoor Environment Research Program at the National Research Council of Canada has developed a program called the Lighting Design Viewer⁵ which

allows individuals to make comparative judgments regarding preferences for indoor lighting conditions.

Figure 3.2- LDV lighting options

The LDV was constructed as a demonstration to show how a particular goal (a set illumination level at the work surface), can be achieved through many aesthetic means. Every example in the comparative matrix had the same amount of light at the work surface, but the expression and feeling (hence desirability) of the lighting solutions vary greatly. This study serves as a model for its ability to show how simply and directly someone can choose the lighting that they desire for a given space.

Figure 3.3- More LDV lighting options

9.2 Painting with Light

Another study of some interest is the paper *Painting with Light*¹ from Cornell University referred to earlier. The authors undertake solving an 'inverse problem', that is, how to get software to determine the actual light aiming and intensity that will most closely match a 'painted' target image. This approach relies on starting with a fixed number of lights in fixed locations and a model with light 'painted on it' through use of a specialized 'light tool' that is part of the software. The assumption is that in lighting design, geometry of the room often dictates where the fixtures must go, so what remains to be determined is how the lights combine and what are the appropriate intensities.

In Figure 3.4, the left image shows the 'painted' design. The center image has had the least squares algorithm applied to it using a fixed luminaire quantity and locations. The right image shows the result with additional ray tracing. The authors themselves admit that to be truly useful, the software would have to automatically place the sources. The study deals with actual computational problems that are involved with using the 'constrained least squares' approach. A fully developed, 'source locating' version of this tool would indeed be useful, but would only show the closest match to the desired image, and not readily demonstrate what is actually achievable by existing luminaires until a full calculation is run.

Figure 3.4- Design (left); best approximation (center); ray tracing (right)

3.3 PROPOSED SOFTWARE STRUCTURE

3.3.1 Assumptions and Caveats

This purpose of this thesis is to provide a schematic outline for a software package that would employ an intuitive methodology for visually choosing light sources in an architectural context. The academic premise is that one should be able to directly choose overt examples of light effects for creation of a visual environment in a method that reflects the design process. To that end, this study does not directly address issues of a computational nature, or matters relating to mathematical lighting algorithms.

There are several key assumptions to be factored in as well. Some aspects of the described system are speculative, keeping in mind that the increasing power and speed of personal computers will accommodate the concepts described. For reasons of simplicity, this schematic does not address issues regarding solid construction, and relies on the importing of files from other software tools. Matters relating to texture mapping are not addressed, as the addition of textures is assumed to be a confounding factor with regards to rapid or real-time imaging. More on this in section 4.5.

3.3.2 Methodology — Key Concepts

The goal is to see what you are adding to a space. Previous systems build a model based on specifications, then do time-consuming full-detail calculations to produce a rendering. What is being suggested here is that the tool employs an

approximate mode of calculation in a low-resolution proxy working-window. This would allow the user to get a good idea of how the prepared solution is shaping up, before resources are committed to a full-blown calculation. This proxy window system has been employed successfully in video editing tools such as Final Cut Pro. Multiple dynamic effects are produced in real-time at low resolution for the user to see what can be achieved, then, if the result looks promising, the system can be committed to a full rendering process which takes much longer to do.

To this end, this proposed system (tentatively called SeeScape) utilizes a set of proxy working views to choose and prepare visual compositions. Once a 'solution' is composed, it can be saved and rendered fully. A solution can be one saved component, or a set of components that will create a larger vision. In this fashion, users can work on various permutations of one aspect of a model at their leisure, then pick and combine the most successful components from various locations throughout the model to create holistic views.

The best way to show this is through example, which follows shortly. First, however, a quick run-through of the tool set is warranted to provide some additional clarity.

3.3.3 The Tool Set

- First Tier: The Usual:

File Edit Add Modify View Tools Window Help

Pull-down menus: This section would include all the usual file management commands, as well as command classes that one would normally find in a 3D environment and those necessary to add and edit lights. Use of these items is not addressed in this study.

Buttons: File/Folder/Save/Print/Undo: The usual.

- Selection controls:

Direct surface/object select, plus Render view and Working view select:

- Direct surface/object select allows the user to choose working surfaces, as well as manipulate light effects as physical entities.
- Render view allows full-screen, full calculation views of the model.
- Working view is the low-resolution proxy screen, as well as convenient section and elevation views. The Configure View command under the View pull-down menu would allow users to control viewport configuration so that the aspects of working screens best suit the type of surface being worked on.

A good example is the 3D MAX control viewport configuration window, shown in Figure 3.5. Section and elevation views would have scaled-to-zoom rulers indicating surface scale and mounting surface heights.

Figure 3.5- The 3D MAX viewport configuration menu

- View manipulation:

Pan (rotate) view / field of view angle / zoom / dolly camera / orbit / observer

left, right, up & down: These are simply the usual buttons required to

navigate. The commands would affect whichever window is active at the time.

- Wireframe/Solid & Surface value

Not addressed in this study. Textures are not to be used, but assignment of surface reflectance values and colours is assumed to be a required feature.

- Light addition:

Add light to wall / add array of lights to wall /

add light to floor / add array of lights to floor /

add light to ceiling / add array of lights to ceiling:

These commands refer to the principal area of affectation, not where the fixture is mounted. There would obviously be some overlap in effect between say, a wall-wash light and a floor aimed light if they are of the same type and mounted similarly near a wall. The differentiation in categorization is for ease of reference and use, depending on the application.

- Luminaire adjustment:

Move lights farther apart / move lights closer together (for multiple selections)

Move lights up / move lights down / space evenly (for selections > 3)

- Light measurement and adjustment:

-*E* will display surface illumination levels, *L* the luminance values. Numerical data will be displayed on the selected surface and in a pop-up data chart.

-*Brighter/Dimmer* will increment or decrement light levels to the next real-world value for the given lamp type.

- Procedural options:

This is intended as a 'What can I do next?' guide. If more than one option is possible at any given time, roll-down options are listed, but the most likely can be displayed. If an option is selected through the roll-down menu, the query button can provide further info through pop-up dialog windows.

- Solution file management:

Active solution: indicates which named or edited file is being worked on. A from-scratch, unsaved file is indicated here.

Load / New / Edit / Save / Save as: file functions for managing solutions

(*.*sss*, or SeeScape solution) which are internal components of the overall

*.*see* (SeeScape file). These functions are shown in the sample use-cycle.

3.3.4 Sample use-cycle

The following pages (Figures 3.6-3.32) describe a typical application process for the proposed methodology. The cycle includes taking an imported model through the necessary stages to apply light to two of the walls in the view through two methods, singly and in an array. These figures can be viewed on the accompanying ADDENDUM CD ROM by double clicking the file SEESCAPE.PPS.

Figure 3.6- Open the file & select a view

The first step is import a model constructed in AutoCAD or similar tool. This view shows a selected view with solid surface rendering in the full-screen render mode, but no lighting effects. The navigation tool-set allows for fairly typical 3D-modelling screen navigation.

Lighting related tools are not available at this time because no effects have been added. To add lighting effects, one must move to the 'Working View'.

Figure 3.7- Determine the next step

The 'Procedural Options' pull-down allows the uninitiated to find their way through the software use cycle. If an option is selected, the query (?) button will pop up a more detailed explanation of the option selected. In this case, the user wishes to move to the working view because there is nothing to view in render mode.

Figure 3.8- Jump to working view

The working view provides up to four windows, the content of which depends on the orientation and nature of the surface being worked on, as well as user preferences. As discussed in the tools description, the view combinations are accessed through the pull-down view menu.

Perspective, section and elevation are shown here, as they are arguably the most useful views for observing the application of lighting to the walls.

The section and elevation views are not active because no surface has been selected to work on. These views will become active when the surface is selected, and zoom to the extents of the active surface.

Figure 3.9- Select a surface

The user must select the surface to which they wish to add light. To do this, the select tool must be engaged. Once this is done and a surface picked, all working instances of the surface (in whichever view) can be selected to be active. In the active mode the surface outline will be red.

The system chooses the corresponding 'zoomed-extent' views using the selected surface from the perspective view and corresponding camera position as references. In this case, the elevation view window is highlighted because that is the optimum (hence default) working orientation for laying out a wall wash. Single lights could be added via the section view, but addition of multiple fixture/light-effect placements of similar types would lead to stacking orthogonal to the view orientation and be invisible.

The procedural options indicate that the next step in the process would be to pick a lighting option.

Figure 3.10- Activate the wall light selection matrix

The single-source wall light matrix, once activated, displays a number of factory and user selected preset effects. Each effect is the rendered result of the photometric distribution from an actual luminaire at a specific distance from the wall. The distance from the wall would depend on the function and type of the fixture, and would be specific to that fixture only. The default mounting height would be at the upper surface limit in the case of a down-light (8'-6" in this case), or at about 6' for a wall mounted sconce, and 0'-0" for a floor-light. The actual specifics cannot be known until a type is selected.

Parameters for each type could be accessed and modified from a right-click-activated menu. Lumen output (light intensity), loss-factors, source spectral content and other variables would also be modifiable using the right click or the Modify pull-down menu.

Figure 3.11- Select an effect type

Once a specific source has been selected, the parameters particular to that installation would appear. A highlighted type is now ready to be grabbed and dragged to the working view. To do this, the selection matrix must be placed or collapsed to a convenient size and location on the screen.

Figure 3.12- Drag the desired effect to the surface

A second click-and-hold on the selected effect type will allow it to be dragged to the working surface and released.

Figure 3.13- Observe the effect in the room

The released light immediately shows its effect in all view screens. The orthogonal views would show an RGB overlay of the light effect with values that would be added to the images. The perspective view would show a quickly calculated low-resolution radiosity solution. This low-res proxy working-window could employ whatever shortcuts deemed necessary to produce a believable, but not necessarily totally accurate representation.

Figure 3.14- Drag the desired number of lights to the working wall

As multiple effects are applied, the orthogonal view layers and adds the RGB values to create an overall effect. The perspective view re-calculates each time in low-res proxy mode.

Figure 3.15- The entire row of lights can be moved closer to the wall via the sectional view

The section view can be used to select and move entire rows of lights. A light can be de- or re-selected in the row using the shift-click convention

Figure 3.16- The lights can be moved away from the wall

As the row is moved away from the wall, notice that the section view beam shape and location is unchanged in the vertical plane. This is because the distance from the intersecting back wall is unchanged.

Figure 3.17- Individual lights can be moved sideways in the elevation view

Individual lights must be selected and moved in the elevation view. A snap can be set to whatever increments are convenient, both vertically and horizontally. Notice the effect of the RGB value addition in the beam overlap. It is not completely realistic, but is still very useful when used for balancing the wall composition.

Figure 3.18- Save a desired result as a solution

Once a desired result is achieved, the solution can be named and saved. This is achieved by using the 'Save solution' button from the solution menu area. The idea is that one should be able to save and combine various and numerous desirable components in order to construct a series of solution options. In this case the solution is named 'Right Wall'.

Figure 3.19- A new solution can include the one just created

By activating the 'New solution' button, fresh or composite solutions can be constructed. In this case, we desire to work on the other wall in the perspective view, but wish to employ what was done on the other wall, and use that previous solution as a guideline for further construction. Here the 'Right Wall' is selected for inclusion in the new file, 'Room'.

Figure 3.20- Save the new composite solution to a new file

Multiple components could be included, but in this case there is just one previously saved solution for use. Included solution components would provide reference guides for snap and spacing to achieve things such as wall wash symmetry when using arrays. End-condition redundancies (overlapping fixture locations) and interference-fit issues would have pop-up warnings about such areas of potential confusion, and/or defaults to remove doubled-up fixtures.

Figure 3.21- The new file shows the effects of included sub-solutions

The elevation and section view screens are now blank because a surface has not been selected.

Figure 3.22- Select a new working surface

The elevation view screen zooms to the extents of the selected surface while the section screen shows the most useful view, with the surface at one side. Again, the active working screen is the elevation view.

Figure 3.23- Select the wall array selection matrix

The wall array matrix is selected. The application methodology for an array matrix is the same as single fixtures: drag and drop. Fixture spacing and mounting height in each array would come from preset factory values, as well as any added user favourites that could be added through right-click menu controls.

The array's extents are determined by the surface extents. Easy (re)definition of surface extents would be a key aspect in rendering this process manageable.

Figure 3.24- Drag an entire row of lights to the surface

As with the single source matrix, arrays are applied through drag-and-drop.

Figure 3.25- Observe the effect of the combined solution on the room

The room now shows the effect of both the previously saved solution plus the new array. The system would allow application a centered array that would truncate at the surface ends and dove-tail with existing lines of fixtures. This could be achieved by defining conditions such as:

- minimum proximity to a surface end-condition before the array is truncated- in most cases, one would not want a fixture location to be less than a foot from any given wall surface.
- using the last fixture from an intersecting (perpendicular) row of lights as the first of the new row- this would remove redundancies at row intersections.
- using that fixture distance from the end of the finished wall as the spacing from the wall at the other end of the array- symmetrical application of arrays often requires that the on-center spacing be adjusted to provide symmetry. The fixture location that showed up in the section when the file was started (from the previously saved 'Right wall' solution) defines the fixture location at the other end of the new array. Fixtures between the end conditions are then evenly spaced.

Figure 3.26- Render the achieved result in full resolution

When 'Render View' is activated, the solution is calculated according to the default or preset parameters using conventional radiosity and ray-tracing algorithms. In this manner, resource-intensive calculation time is minimized, and only spent on outcomes that appear promising visually in the proxy-working window.

Output would have three main components:

- rendered images
- numerical analysis
- a plan showing the fixture locations and specifications- though not overtly shown, every application from a selection matrix would include the specific data that belongs to that fixture type (catalog #, lamp source, lamp output etc.). This would allow the designer to generate sufficient data for contract documents.

Figure 3.27- Illuminance readings show the light levels

Once the user has a solution that is visually acceptable, a quick check of the illumination levels will give the user easy verification if numerical compliance is required to satisfy program needs or regulatory guidelines. This is easily achieved by clicking the E (illumination) button and then the surface you wish to query. The output type shown here is a typical Lightscape feature.

Figure 3.28- Any surface can be checked

Most numeric guidelines are centered on horizontal illuminance. The designer might actually be more concerned with the luminance, or brightness, of the wall surface. Illuminance is shown here.

Figure 3.29- Check the ceiling illuminance if desired

Though not usually considered for down-lighting applications, the ceiling illuminance can give user a sense of how much inter-reflected light is in the room. This will vary greatly depending on the colour and reflectance values of the walls and floor.

Figure 3.30- Increase the light levels

Should the illumination levels not be satisfactory, adjustments can be easily made. The illumination level can be increased by pressing the button with the 'slider up' icon. Output would increase to the next real-world value for that lamp type. For example, if the A-19 lamp in the original matrix selection were 60W, then the next jump up would be to 100W. Actual values for the output or any other bit of luminaire information could be queried at any time through use of the right-click mouse button and selecting 'Properties'.

Figure 3.31- Decrease the light levels

Conversely, the 'slider down' button would change the output back down to the original 60W, then to 40W, or down to 25W with further clicks. The changes would be effected until the actual range of output possibilities for that lamp type is exhausted.

Figure 3.32- Printed output from the solution file

This is an example of the type of information delivered by the output file. Fixture locations and types are noted. The file would be in DWG vector or bitmap format for inclusion in specifications and AutoCAD drawings.

Chapter 4: ISSUES TO BE FURTHER EXAMINED

What factors are presently perceived to be possible confounds or complications to this application methodology?

4.1 Lighting of irregular surfaces and 3D objects

A suitable methodology would have to be developed for problematic issue of dealing with articulated or curved surfaces. With respect to the elevation of curved surfaces, should the view be 'straightened' such that the curvature is not shown, or should the light effects be compressed as a surface approaches perpendicularity to the view plane? In the case of sectional viewing of articulated walls or ceiling planes, hidden lines would be desirable in indicating hidden tectonic shifts.

4.2 Defining surface limits

Auto-selection of the elevation and section views relies heavily on clearly demarcated surface limits. The demonstration model consists of the simplest of planar surfaces. Large surfaces, particularly very long walls or a ceiling field, may occasionally require being broken down and defined as more manageable units. A surface definition methodology would need to be developed.

4.3 Ceiling and floor applications

The methods for applying light to ceilings and floors would be similar to the wall application algorithm, but could require different features because the application

criteria can be different. In the case of floors, a dotted line indicating the angle of 50% output would be helpful in spacing light distribution patterns, particularly in the sectional views. As mentioned in the tool-set descriptions, flexible working screen configurations would undoubtedly be a requirement. Since many fixtures can have multiple applications, it stands to reason that any given fixture could be included in more than one fixture selection matrix. For example, a down-light pot could be used as a wall-washer, or a wall mounted linear up/down light could be employed as a wall-washer, ceiling lighter or down-light. This redundancy actually makes application simpler.

4.4 Complications from textures

Texture and surface mapping complicates (and greatly slows) the calculation process, and even the low-res proxy window might require more calculation time to render an image than is desirable if textures are included. If the SeeScape model was to be implemented as a plug-in to Lightscape or another software package that implements texture mapping, the textures might have to be dropped or 'dumbed-down' to simplify and speed up the calculation process. The same would apply if SeeScape as a stand-alone were to open or import a textured model. Colour and reflectance values alone would deliver enough realism to make design decisions.

4.5 Light object behaviour- visible sources

Though not addressed in this example, many luminaires exist outside the surfaces planes to which they are mounted. Provision would have to be made for dealing with the physical and perceptual realities of the objects themselves. For example, an indirect linear fluorescent fixture has both physical shape and luminance characteristics (which vary according to the angle viewed). A key factor to be reckoned with here is the realistic emulation of glare.

Chapter 5: CONCLUSION

5.1 Plug-in or program?

If actual software that uses this lighting design methodology were to be constructed, how would it be best applied, as a stand-alone piece of software or a plug-in to existing tools such as Lightscape? Perhaps even as an adjunct to a modelling tool such as Form Z?

As a stand-alone tool, SeeScape would have to deal with using imported multiple surface/mapping formats, and include its own radiosity and/or ray tracing engines. This would make it more costly to create and implement. Marriage to an existing lighting tool such as Lightscape would lessen the complexity required in terms of dealing with file format flexibility, and the tool would be able to utilize the existing radiosity/ray-tracing engines in order to generate the proxy images. The drawback would be that the user might be stuck with using the vexing two-stage, preparation/solution file process inherent in Lightscape use.

The lighting application industry at present considers no modelling/construction tool (i.e. AutoCAD or Form Z) to be suitable for accurate lighting representation. As stated, this is mainly due to the fact that one can't use real-world photometry. So, as an adjunct to a construction tool, SeeScape might have to use its own light-modelling engines. Whichever avenue that might be developed, there are several key concepts to be retained:

- The visual effect-selection matrix: pick the effect that you want to see.
- Drag-and-drop light effects: immediately visually verify your individual actions
- The proxy window: see what you are getting as a whole or in part before committing to lengthy calculations.
- The save and combine solutions feature: utilize a single holistic file structure that incorporates the lighting sub-solutions that remain editable.

It has been suggested that the selection matrix on its own would be a useful tool.

A Visual Photometric Viewer could employ the matrix concept for sorting of IES format files intended for use in another piece of lighting software. Conceivably, one could have a series of pre-set typical application location types set up, and then be able to just drag multiple IES files into the matrix input and see what they look like. This seems to just be going part of the way to an effective lighting tool, but it would be an enormous improvement to the existing application methods.

5.2 The Next Steps

The research work from this thesis supported by anecdotal evidence suggests that there is an opportunity to take this thesis further. Therefore, some form of market study is warranted to determine if there is a basis for development of SeeScape as a working tool. Potential user groups must be polled for the type of functional requirements that they see as more useful than the tools already available, or to determine if SeeScape should be developed as an adjunct program or plug-in to existing tools.

It appears that in order to function in the manner envisioned, that is, quickly but without a great deal of detail, SeeScape would provide the highest utility as a 'sketch' tool. Users could quickly decide on design direction, then they could export the file to a high-end mapping tool like Lightscape to refine the images if required, or simply export the relevant fixture data in DWG form for specification if no further visual study is required.

To provide the highest utility, perhaps SeeScape should have a construction component. This would reduce the number of tools that a user would need in a 'suite' from three to two. Instead of AutoCAD, SeeScape and Lightscape, only SeeScape and Lightscape would be required. In order for this to be practical, the construction elements would need to be at least as good as the basic 3D software tools so that regular solid geometry could be simply achieved. Features such as loft generation would not really be necessary.

5.2.1 Who Would Develop SeeScape?

Apart from soliciting private seed money from venture capitalists, there are two avenues of approach for taking SeeScape to the next level. One would be to approach existing lighting software developers such as those behind Lightscape and Form Z. They would have the lighting expertise at hand, as well as the infrastructure and distribution required. Their motivation may not be that high in the case of Lightscape's parent, Autodesk, as they potentially would be creating competition for their own existing product. A group more likely to be motivated

would be the larger manufacturers of lighting equipment. They all have their own proprietary tools to enhance sales efforts. As mentioned earlier, they are also motivated to produce highly 'usable' tools as a sales aid. Further, they have no problem with capital.

5.2.2 Proposed Testing Methodology

The following areas of investigation are intended as an outline for determining the desirability, utility and market direction for any future development of SeeScape as a viable lighting tool. A questionnaire should be given after a SeeScape demonstration such as the one produced for this thesis.

- What is the area of expertise the user is involved in? Engineer, architect, interior designer, lighting designer, sales agent, educator...?
- What are the software tools that they presently use for construction and modelling?
- Would they find a tool such as SeeScape useful?
- How do they see it being used (sketch tool, full lighting package, plug-in)?
- What imports and exports would they find useful?
- Would construction capability improve SeeScape's utility, or would they still use their present construction tools?
- What would they see as possible improvements to SeeScape?

Analysis of data from this type of questioning should lead to greater understanding of the potential user groups and their needs.

5.2.3 The Challenges: Immediate and Future

The short-term challenge will be to ensure that the SeeScape idea is properly protected when potential developers are approached. Beyond the coverage that academic publication provides, some form of intellectual property registration may be required. The most promising avenue for development seems to lie with the larger lighting manufacturers. Preliminary contacts in this area have already been made. If there suitable protection exists, a demonstration at one of the larger lighting industry gatherings could produce some interest. LightFair and the annual IES conference would be the best venues.

In the longer term, as suggested in the previous section, some form of market research is warranted prior to any development and prototype testing on users. How all this occurs depends largely on where the interest comes from. If these hurdles are crossed, the technical challenges must also be addressed. There may be limiting factors that do not allow the process to perform as envisioned. Time, resources and determination are all required to move ahead.

5.3 Final Reflection

Overall, a problem exists with lighting tools, they are hard to use and you do not get to see what it looks like until you're done. The SeeScape process would allow freedom from technical constraints for designers and architects, and allow engineers and technologists to learn to see what the implications of their actions are. Lighting decisions that affect the architectural spaces could be visualized

early in the design cycle. In this manner, potential opportunities can be effected and implemented before the electrical design, and therefore be more cost-effectively achieved.

The SeeScape process has further potential as an educational or demonstration tool. Use of SeeScape in spatial visualization would greatly facilitate the teaching of lighting as an applied art form. Students could finish their education with a real-world grasp of lighting fundamentals and the inherent equipment. Further, in a professional context, SeeScape would allow consultants to educate and inform their clients, whether they are architects, developers or facility owners. In turn, the 'trickle-down' effect of this tool could allow designers and developers to easily give new home-buyers the freedom of seeing the choices available regarding their lit environment.

SeeScape could be a viable tool for both education and application. All would benefit from a more graphic dialog regarding the planning and realization of the constructed environment.

Postscript: At the time of printing for this thesis it has come to light that Autodesk has released VIZ 4 (formerly 3D Studio Viz). This 3D modeling tool is reportedly capable of importing and handling IES format files. This is a large step forward for architectural design, as it is the first incidence of a single, complete construction / mapping / lighting tool.

REFERENCES

1. Painting with Light, *Chris Schoeneman, Julie Dorsey, Brian Smits, James Arvo, and Donald Greenberg*, Program of Computer Graphics, Cornell University, Ithica, NY, 1993.
2. Ask Tog: Human Interface Evangelism & Practical Design, *Bruce Tognazzini*, Alert Box on-line, at <http://www.asktog.com/basics/firstPrinciples.html>
3. Top Ten Usability Heuristics, *Jakob Nielsen*, http://www.useit.com/papers/heuristic/heuristic_list.html
4. Making Global Illumination User-Friendly, *Greg Ward*, Lawrence Berkeley Lab., at <http://radsite.lbl.gov/radiance/papers/erw95.1/paper.html>
5. Lighting Design Viewer, Light Group, Institute for Research and Construction, National Research Council, Ottawa, Canada.

BIBLIOGRAPHY

Lighting Handbook: Reference and Application, 8th Edition, Illumination Engineering Society of North America, IESNA, 1995.

Photometry and Radiometry: A Tour Guide for Computer Graphics Enthusiasts, *Ian Ashdown*, P. Eng. John Wiley & Sons, 1994.

Virtual Photometry: New Directions in Photometry
Ian Ashdown, P.Eng. September, 1993.

Visual Reality: Computer Techniques for Lighting Design
Ian Ashdown, P. Eng. 1995

ON-LINE BIBLIOGRAPHY

Caustics by *Henrik Wann Jensen*

<http://graphics.ucsd.edu/~henrik/images/caustics.html>

Global Illumination by *Henrik Wann Jensen*

<http://graphics.ucsd.edu/~henrik/images/global.html>

Overview of Photon Mapping by *Yves Poissant*, 2003

<http://www.ypoint.com/tutorials/Photon-Overview.htm>

Ray Tracing vs Radiosity by *Florian Brunner & Martin Zürcher*

<http://www.fortunecity.de/lindenpark/schwitters/336/Compi/3D/8.Serie/>

Radiosity vs. Ray Tracing by *Ray Yep II*, 1995

graphics.lcs.mit.edu/classes/6.837/F00/Lecture23/lecture23.pdf

A Summary of Principles for User-Interface Design. by *Talin*

http://www.sylvantech.com/~talin/projects/ui_design.html

User Interface Design by *The Common Front Group*

<http://cfg.cit.cornell.edu/cfg/design/contents.html>

SOFTWARE BIBLIOGRAPHY

- AGI-32: Lighting Analysts Inc.

 10440 West Bradford Road, Unit A

 Littleton, Colorado 80127

- Cosmo Worlds: Silicon Graphics Inc.

 1600 Amphitheater Parkway

 Mountainview, California 94043

- Form Z: AutoDesSys Inc.

 2011 Riverside Drive

 Columbus, Ohio 43221

- Genesys II: Genlyte Thomas Group LLC

 10350 Ormsby Park Place, Suite 601

 Louisville, Kentucky 40223

- Lightscape: Autodesk Inc.

 111 McInnis Parkway

 San Raphael, California 94720

- Lithonia Visual: Lithonia Lighting Inc.
Conyers, Georgia 30012
- Lumen Micro: Lighting Technologies Inc.
1630 Welton Street, Suite 400
Denver, Colorado 80202
- Maya: Alias ®
210 King Street East
Toronto, Ontario M5A 1J7
- Radiance: Berkely Laboratories
University of California at Berkely
1 Cyclotron Road
Berkely, California 94720
- 3D Studio Max: Discreet Technologies
10 Duke Street
Montréal, Québec H3C 2L7

PROFESSIONAL AND RESEARCH CONTACTS

- Dr. Guy Newsham & Dr. Jennifer Veitch, Research Officers, Indoor Environment Research Program, National Research Council, M-24 Montreal Road, Ottawa.
- Dr. Anca Galasiu & Dr. Dan Sander, Research Officers, Indoor Environment Research Program, National Research Council, M-24 Montreal Road, Ottawa.
- Ivaldo Pasini, M. Eng., P. Eng., Senior Illumination Engineering Specialist, Technology Architectural and Engineering Services, PWGSC, Ottawa.
- Karen Pero, Lighting specialist, Technology Architectural and Engineering Services, PWGSC, Ottawa.
- Ian Ashdowne, P. Eng., Ledalite Architectural Lighting Products, Langley B.C.
- Dr. Mark Rea, Professor of Architecture, Psychology, Director, Lighting Research Center, Rensselaer Polytechnic Institute, Troy, NY.
- Peter Ngai, Application Specialist, Peerless Lighting, Berkely, California.
- David DiLaura, Senior Instructor, University of Colorado, Dept. of Civil Engineering-Civil, Environmental & Architectural Engineering.
- Chris King, Lighting Technologies Inc.
- George Mischler, simulations developer, schorsch.com, Lighting Design Tools.
- Tony Picco, Lighting Design + Application magazine, Official Publication of the Illumination Engineering Society of North America, 120 Wall St., NY, NY.
- David Speer, Vice President Sales & Marketing, Lighting Analysts Inc.
- Greg Ward, Lawrence Berkely Labs, California.

KEY DEFINITIONS

- *Chromaticity:*

The spectral content of light sources, which frequencies are present or predominant.

- *Illuminance:*

The amount of light a surface is receiving (units of lux or footcandles), varies with the inverse square of the distance from the light source.

- *Global Illumination:*

A rendering system that utilizes both radiosity and ray tracing to deliver emulation of direct and indirect lighting.

- *Isolux:*

Graphic representation showing lines of equal illumination.

- *Lambertian surface:*

A hypothetical surface which reflects the light it receives in an omnidirectional manner.

- *Luminaire:*

A light fixture, in particular the optical components and enclosure, independent of support structure.

- *Luminance:*

The brightness of an object or light source (units of candellas per sq. m. or foot-lamberts), it does not vary with distance. Object luminance is dependent on both the amount of incident light and the reflectance of the surface in question.

- *Photometrics:*

The light emitting behaviour of a luminaire, how light is distributed.

- *Radiosity*

An algorithm based on radiative heat transfer theory for calculating light distribution for a given combination of surfaces. One calculation delivers a solution that applies to all views.

- *Photon Mapping:*

An extension of ray tracing that can produce caustics, the effects produced by specular reflection and refraction through prismatic materials.

- *Ray tracing:*

A direct line-of-site tracing from the viewers perspective to all visible points within in a model. Ray tracing is the best method for accurate depiction of specular reflections. A calculation is required for every viewer position.

- *Specular:*

Shiny, the capacity to reflect light in a non-diffuse manner. A mirror is specular.

ADDENDUM

Accompanying this document is a CD ROM which contains the following image sequence files. These are identical images to the printed figures, and are intended to allow more realistic viewing of the software screen images.

- For the Lightscape use-cycle, Figures 2.2-2.15,
double-click **LIGHTSCAPE.PPS**
- For the Genesys use-cycle, Figures 2.16-2.31,
double-click **GENESYS.PPS**
- For the SeeScape sample use-cycle, Figures 3.6-3.32,
double-click **SEESCAPE.PPS**