The Pennsylvania State University
The Graduate School
College of Information Sciences and Technology

SYSTEM FOR AUTOMATED INTERACTIVE LIGHTING

A Dissertation in
Information Sciences and Technology
by
Joseph A. Zupko

© 2009 Joseph A. Zupko

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2009
The dissertation of Joseph A. Zupko was reviewed and approved* by the following:

Brian K. Smith  
Professor of Information Sciences and Technology  
Dissertation Adviser  
Co-Chair of Committee

Magy Seif El-Nasr  
Assistant Professor of Interactive Arts and Technology  
Simon Fraser University  
Co-Chair of Committee  
Special Member

David L. Hall  
Professor of Information Sciences and Technology

Mary Beth Rosson  
Professor of Information Sciences and Technology

John Yen  
Associate Dean for Research and Graduate Programs  
University Professor of Information Sciences and Technology

*Signatures are on file in the Graduate School.
ABSTRACT

Video games have grown to a billion dollar entertainment industry. Beyond entertainment, games are increasingly the target of serious research, from learning to “serious play”, training, therapy, and communication. Alongside the expansion of game applications has come the expansion of game media. Games are now incredibly complex and expressive multimedia platforms that rival the highest fidelity tools of film or television. Independent of their applications, most games are visual works and any researcher or entertainer interested in games must first purposefully control and manipulate the visual substance of a game to achieve a goal. Therefore, this dissertation presents a focus on one factor that is inseparable from the visual experience of a game: lighting.

Our real world visual experience is shaped by light. The very structure of modern life is possible because we have transcended the constraints of natural day and night cycles and can effectively operate 24 hours a day. Games too are now shaped by light. Modern computer visualization, the process of generating an image of a virtual thing, now offers a physically realistic illumination model. Virtual illumination can produce nearly photorealistic images (literally, images that look like a photograph).

However, successful lighting in video games requires more than a physically realistic illumination model. Aesthetics and function are of equal or greater importance. Lighting can control visual focus, directing a player’s gaze to important objects. It can also control mood and perceived emotion, communicating a narrative and engaging a player in a game world. The role of a lighting designer in games is therefore crucial.

Lighting designers control and purposefully manipulate light. A lighting designer in film will carefully control lighting, scene by scene and even frame by frame, to precisely convey emotion, control visual perception, and communicate intent. Game lighting designers face a challenge however, in that omniscient knowledge of a game’s state at any time is not available to them in advance. A player can make many decisions that change the game world. For example, a film lighting designer might carefully arrange lighting to strike a specific visual cue when the lead actor crosses the threshold of a front door. However, a game lighting designer might also need to predict and accommodate a player coming in through a bedroom window, a player destroying the front door, a player never coming to the house at all, or a player coming to the house at a point in the story where the visual cue is no longer appropriate.

Under an assumption that game lighting designers require tools to design light that assist with the challenge of adapting to player interaction at runtime, this dissertation presents the design, development, and final product of research into a lighting design tool, the System for Automated Interactive Lighting (SAIL). The goal of this research was to develop an adaptive system that maintains lighting design goals (aesthetic and functional) in the context of unpredictable, interactive experiences.
This dissertation makes several contributions. As an artifact, SAIL offers functionality not available in existing automated lighting systems. As a design tool, SAIL may be useful as a pre-visualization or storyboarding tool. As an experimental instrument, SAIL offers the possibility of simplifying existing experiments into automated lighting and facilitating new types of experiments. SAIL’s contributions are discussed in depth in Chapter 8. These contributions are supported by a qualitative study discussed in Chapters 6 and 7. These chapters also present several other interesting observations that may be useful in general to the field of automated lighting.
# TABLE OF CONTENTS

List of Figures .......................................................................................................................... viii  
List of Terms .......................................................................................................................... xiv  
List of Equations .................................................................................................................... xvii  

Chapter 1: Introduction ........................................................................................................... 1  
1.1 Automated Lighting ........................................................................................................ 5  
  1.1.1 Overview .................................................................................................................. 6  
  1.1.2 Theory .................................................................................................................... 7  
  1.1.3 Contributions ......................................................................................................... 7  
1.2 Organization ...................................................................................................................... 8  

Chapter 2: History and Theory of Light and Lighting Design .............................................. 10  
2.1 The Art of Rendered Light ............................................................................................. 10  
  2.1.1 Photographic Film and Contrast ............................................................................ 12  
  2.1.2 Lighting Direction ................................................................................................... 14  
  2.1.3 Usage of Contrast and Direction .......................................................................... 15  
  2.1.4 Three-point Lighting ............................................................................................. 21  
2.2 Color Constancy ............................................................................................................. 21  

Chapter 3: History and Theory of Light in Computer Rendering ........................................ 23  
3.1 Related Work .................................................................................................................. 23  
  3.1.1 Architectural Inverse Lighting ............................................................................... 23  
  3.1.2 Perceptual Inverse Lighting .................................................................................... 25  
  3.1.3 Interactive Inverse Lighting .................................................................................... 28  
3.2 Physics of Light in Computer Rendering ....................................................................... 30  
  3.2.1 Physically Accurate Rendering .............................................................................. 34  
  3.2.2 Psychophysical Rendering ..................................................................................... 38  
  3.2.3 Summary ................................................................................................................ 43  

Chapter 4: System for Automated Interactive Lighting (SAIL) ........................................... 44  
4.1 Considerations of Theory and Lighting Practice .......................................................... 44  
4.2 Components of SAIL ...................................................................................................... 45  
4.3 SAIL’s Relationship to Related Work ......................................................................... 46  
4.4 Current Implementation of SAIL .................................................................................. 48  
  4.4.1 Lighting, Shading, and Shadows .......................................................................... 49  
  4.4.2 COLLADA Materials ............................................................................................. 50  
  4.4.3 Occlusion ................................................................................................................ 50  

Chapter 5: Algorithms of SAIL ............................................................................................ 51  
5.1 Image Analysis ............................................................................................................... 51  
  5.1.1 Direction ................................................................................................................ 51  
  5.1.2 Contrast .................................................................................................................. 53  
5.2 Modeling an Object’s Response to Light ...................................................................... 54  
5.3 Runtime Lighting .......................................................................................................... 54  

Chapter 6: Evaluation of SAIL – Study Design ................................................................... 56  
6.1 Development of SAIL’s Implementation ...................................................................... 56
List of Figures

Fig. 01. This figure of Gordon Freeman from the video game Half-Life 2 (released in 2004 by Valve) illustrates the use of lighting to control visual appearance. A rim of white light surrounds the character, helping to isolate him from the background and draw the viewer’s attention to the character. ............... 2

Fig. 02. This bowl of fruit from (Birn 2006) illustrates light “cheating”. The bowl of fruit pictured in the top left image is illuminated by a window, represented by the red box in the bottom image. The top right image is the same bowl of fruit illuminated by a “cheated” light source, meant to improve the aesthetics of the scene while also appearing to still be motivated by the window. The cheated light source is represented by the green box in the bottom image................................. 3

Fig. 03. This figure from the video game Guild Wars (released in 2005 by ArenaNet) illustrates the usage of particle effects to indicate important objects. . 5

Fig. 04. This figure illustrates how an image of a shaded sphere provides the illumination example that SAIL uses to light a 3D model of a young girl (Milam 2008). SAIL chooses light sources to recreate the intensity contrast and apparent light direction as indicated by shading present in the example image....................... 6

Fig. 05. This figure from (Birn 2006) illustrates film three-point lighting. Three-point lighting divides lighting an object into three lights for directly controlling essential components of the visual presentation of an object............................... 7

Fig. 06. This figure from http://erich.realtimerendering.com/ shows an image created with a rendering system designed by Eric Haines in 1991...................... 10

Fig. 07. This image from the movie Monsters, Inc. (released in 2001 by Disney/Pixar) illustrates the advancement of computer generated art.................... 11

Fig. 08. This figure shows the somewhat odd usage of lens flare in a computer rendered image. Lens flare is an artifact of light interacting with a camera that is generally deemed a mistake in photography and cinema. Nevertheless, in computer rendering it is sometimes used to help add realism to a piece.............. 12

Fig. 09. This image from http://en.wikipedia.org/wiki/Exposure_(photography) shows the effects of exposure on film......................................................... 13

Fig. 10. This figure displays a Hurter and Driffield curve illustrating the relationship between a change in light exposure and a change in film density, which determines the contrast in an image. ...................................................... 13

Fig. 11. This image from (Lee, Hao et al. 2006) illustrates the effect of light direction on the perception of detail. The red arrows indicate the direction of lights controlled by the system in (Lee, Hao et al. 2006). The illuminated object is a 3D scanned manuscript with difficult to perceive lettering......................... 14

Fig. 12. This figure from (Birn 2006) illustrates the effect of lighting on the silhouette of an object and on occlusion. A back light is added to the right picture that helps the edge of the apple stand out and improves the perception that the apple is a separate object from the background........................................... 15

Fig. 13. This image from the movie Out of the Past (released in 1947 by RKO Pictures) demonstrates classical film noir in black and white; a dark, high-contrast style of lighting................................................................. 15
Fig. 14. This scene from the movie *The Dark Knight* (released in 2008 by Warner Bros.) is a modern day example of film noir using color, illustrating the same essential usage of extreme light contrast to create evocative imagery.

Fig. 15. This photograph titled *Nomad musician* by an unknown artist demonstrates silhouette lighting style.

Fig. 16. This figure shows an image of Boris Karloff as the monster in the 1931 film *Frankenstein*. The usage of under lighting (lighting from below) increases the unease and unnaturalness of the monster, helping to create the illusion of something not quite human.

Fig. 17. This figure from http://www.chromasia.com/ demonstrates soft or high-key lighting.

Fig. 18. This screenshot from the video game *Guild Wars* (released in 2005 by ArenaNet) illustrates the effective use of soft lighting to create a fantastical, storybook world. Bright colors and soft shadows create a warm yet otherworldly atmosphere.

Fig. 19. This image from the video game *Half-Life 2: Episode 2* (released in 2007 by Valve) demonstrates the effectiveness of light direction in implying larger context. The shafts of light behind the character indicate a window or other opening above the frame of the image.

Fig. 20. This image of *St. Peter in Prison*, Rembrandt van Rijn, demonstrates chiaroscuro style lighting.

Fig. 21. This image from a production of *Cabaret* illustrates cameo style lighting (see http://www.michaelskinnerld.com/cabaret). It is an extreme form of chiaroscuro lighting, hiding much of a character in darkness and creating sharp, clearly defined cast shadow.

Fig. 22. This image from the movie *Barry Lyndon* (released in 1975 by Warner Bros.) illustrates an extreme light condition. This scene was shot in natural candlelight using specially made, ultra-fast lenses. Typically, a scene like this would be shot by using small amounts of fill light to increase the intensity of ambient lighting to make the scene easier to capture with traditional lenses.

Fig. 23. This painting from http://tinyurl.com/bnvc7u illustrates Notan style lighting.

Fig. 24. This image is of an automatically illuminated virtual space from (Kawai, Painter et al. 1993). Kawai’s system found light intensities and angles given light positions and a set of metrics meant to represent subjective lighting goals such as “spaciousness” derived from (Flynn 1977).

Fig. 25. This figure shows a virtual scene of an office illuminated by the system in (Costa, Sousa et al. 1999). The system in this case was given goals for the intensity of lighting on different surfaces in the environment and was restricted to placing lights along the ceiling. It was otherwise free to choose light source positions and intensities.

Fig. 26. The 3D model of a ship in this figure is illuminated by the system of (Shacked 2001). This system’s intent was to maximize perceptual features for shape, detail, and the perception of depth automatically given a camera point of view specified by a user.
Fig. 27. This image from (Lee, Hao et al. 2006) illustrates that system’s consideration of cast shadow on an object to help identify depth in a figure. The red box highlights a shadow that the system explicitly manipulated to help identify the structure of this 3D object. .................................27

Fig. 28. This figure shows the Expressive Lighting Engine (Seif El-Nasr 2003). This prior work in automated interactive lighting automatically lit objects and scenes to “expressive” lighting goals such as “visual tension”. .................................28

Fig. 29. This figure shows an object lit by the system of (Halle and Meng 2003). Halle’s system simplified the process of lighting 3D medical models by providing users with a constrained light setup based on three-point lighting. ..... 29

Fig. 30. This figure illustrates a light photon interacting with a surface. An incoming photon stream represented as ray $\omega$ to a surface point $x$ is determined by two differential angles $\theta$ and $\lambda$. .................................................................31

Fig. 31. This figure illustrates the difference between a diffuse and a specular or shiny surface. The diffuse surface represented on the left reflects incoming light equally in all directions. In contrast, a specular surface only reflects light in a small wedge of directions and a perfectly specular surface reflects light in only one direction. ...................................................................................................32

Fig. 32. An illustration of why light intensity attenuates with distance. As photon streams diverge from the light source, less of the streams intersect a surface patch, resulting in fewer photons intersecting the surface per unit time and lower resulting light intensity.................................................................33

Fig. 33. This figure is an example of non-photorealistic rendering from the video game Ōkami (released in 2006 by CloverStudio). Non-photorealistic rendering is outside the scope of this work but it illustrates some of the fascinating interpretations of illumination that exist in computer rendering. ....................... 34

Fig. 34. This figure shows how radiosity transfers light throughout an environment. An initial source of light emits photon streams into an environment. Surfaces in the environment reflect these streams back and onto other surfaces. Each surface is assumed to be Lambertian. ..................35

Fig. 35. This image of the Cornell Box from http://www.graphics.cornell.edu/online/box/ illustrates the lighting phenomenon of light “bleed”. The indicated region shows that indirect reflections off of a colored surface cause some of the hue and saturation of that surface to interact and reflect off another surface, giving that surface a similar hue and saturation. 36

Fig. 36. This figure shows how ray tracing considers light in reverse, shooting rays from the eye’s point of view into an environment and tracing rays to any sources of illumination.................................................................37

Fig. 37. Although all cast from the same off screen light source (the Sun), three clearly distinct shadow outlines are visible within the red box shown in this figure. This is an artifact of the form of projected shadowing as used in this image from the video game Guild Wars (released in 2005 by ArenaNet). ............38

Fig. 38. This figure illustrates rasterization. The corners of a triangle are projected onto a near plane, which represents the screen of a computer display, and then the pixels that are between the projected corners are filled in. .................................39
Fig. 39. This image of a street lamp illustrates the emission typical in rasterization. The lamp bulb has an emission term giving it a white color. It does not however affect the surrounding environment. The lamp post therefore remains completely dark.

Fig. 40. This figure illustrates the three types of light sources typical of rasterization. The light on the left is a point light which emits light in all directions equally. The light in the middle is a directional light which emits parallel rays of light from an infinitesimal planar surface. The light on the right is a spot light which is similar to a point light but is restricted to a cone of influence.

Fig. 41. This figure shows a scene from the video game *Half-Life 2* (released in 2004 by Valve). *Half-Life 2* uses radiosity lighting to pre-compute static lighting with indirect illumination. The left image is the same scene as the right, except the right image includes indirect or “bounced” illumination while the left image does not. This illustrates that indirect illumination can contribute greatly to the overall lighting of a scene.

Fig. 42. This figure illustrates the computer rendering technique of texture mapping. An image shown on the left is mapped to the surface of a geometrical sphere providing the sphere with the texture and detail of the texture map.

Fig. 43. This figure from http://www.punkuser.nte/vsm/ illustrates a real-time shadowing technique which allows shadows to have soft edges as indicated by the red box.

Fig. 44. This image from the video game *Gears of War* (released in 2006 by Epic Games) illustrates an object casting shadow on itself as identified by the red box. This case is a difficult rendering effect to achieve in rasterization and is usually handled as a special case or not at all.

Fig. 45. This is an image of a mirror ball from http://www.dbki.de/, a real world spherical mirror surface that represents a 360° illumination capture of the environment around the ball. It is often used for a technique called image based lighting where the image of the ball is turned “inside out” and mapped to the inside of a spherical surface which is then used to apply lighting to an environment as if it originated from the sphere.

Fig. 46. This figure illustrates the high-level process of SAIL.

Fig. 47. This figure illustrates SAIL. The shaded circle is the goal image. The left image is the man naturally lit. The center image is the man lit to achieve the goal image. The right image is the man lit as a compromise between the two.

Fig. 48. This figure shows an interface of the Expressive Lighting Engine (ELE). ELE’s interface consisted of a large number of interdependent numerical terms. Lighting experts found this interface difficult to use and were unable to effectively specify lighting designs with it.

Fig. 49. This screenshot is from the implementation of SAIL used during evaluation.

Fig. 50. This image illustrates the definition of image direction metrics $\theta$ and $\beta$.

Fig. 51. This figure demonstrates jitter sampling, a sampling scheme that divides a space into an even grid and randomly samples an equal number of points from
each grid. This approach requires fewer samples than a straight random sampling approach. ................................................................. 54

Fig. 52.  This figure illustrates the method of stochastic gradient descent. The solution for a function begins at some initial value \( x_0 \). This is the appearance of an object under the illumination of its environment. The value is then stepped towards the best value \( x_5 \) based on the slope of the error between each point \( x_0, x_1, \ldots x_5 \). The best value is the desired appearance. Because gradient descent is a hill climbing algorithm, it effectively walks along an error surface representing a gradual decrease in error. Because of this, the interim solutions \( x_0, x_1, \ldots x_4 \) are still reasonable solutions which allows SAIL to maintain visual continuity as it approaches the best solution................................................................. 55

Fig. 53.  This screenshot is from version 1 of SAIL which was used for demonstration at this dissertation’s proposal. The character is from the video game Max Payne (released in 2001 by Rockstar Games)......................... 56

Fig. 54.  This screenshot is from version 2 of SAIL created to address criticisms of version 1 ................................................................. 57

Fig. 55.  This screenshot is from the implementation of SAIL used during evaluation ................................................................. 58

Fig. 56.  This figure shows age and sex distribution of participants. 1 participant reported “greater than 21” for this question and is not shown in the chart......... 59

Fig. 57.  These figures show participants’ experience in years and specific areas of expertise. Participants were allowed to give multiple responses to the question regarding their area of expertise, so the individual counts do not sum to 19. Note that “Photography” replaces “Other” in the chart (“Other” was a fill in choice on the survey). In every case where a participant wrote in an answer for their area of expertise they wrote “Photography” ................................................................. 60

Fig. 58.  This figure shows the activity level of participants. Participants were allowed to give multiple responses to this question, so the individual counts do not sum to 19. Two participants responded with “professional artist” and “installation art” for this question. These responses were included in the “Production” count. The “Amateur” choice was possibly misinterpreted as a measure of experience. It was intended to indicate practice without pay or without formal training. ................................................................. 61

Fig. 59.  This figure illustrates the three images available to participants during evaluation. The leftmost image is the image attached to key “Q”. The center image is the image attached to key “W”. The right image is the image attached to key “E” ................................................................. 64

Fig. 60.  Many participants expected the windows in the back of the room shown in this figure to be sources of illumination even though they were not............. 71

Fig. 61.  This image illustrates a billboard. The tree shown in the left image is not composed of geometry. It is only a single image whose orientation is synced with the orientation of the camera to maintain an illusion of an object with 3D depth. The right image shows this illusion failing as the tree is viewed from above. ... 84

Fig. 62.  This image from the video game Guild Wars (released in 2005 by ArenaNet) illustrates a situation where SAIL would probably be ineffective. A
wide landscape style portrait such as this figure requires lighting that considers individual objects (such as the trees or mountains in this figure) as collections of objects to be lit to the same goal. SAIL would consider all the objects in this scene as individual pieces. ................................................................. 90

Fig. 63. This image from http://www.rosco.com/us/gobocatalog/fire.html is a gobo, a template that is placed over a physical light to cast patterns into an environment. Various rendering techniques are used to create similar effects in computer rendering. ........................................................................................................ 91

Fig. 64. This figure illustrates the rough symmetry of a human model that can be exploited to reduce the memory storage requirements of the mathematical model used by SAIL to understand how that object responds to light. ......................... 102

Fig. 65. This graph demonstrates the roughly cubic relationship between a change in key yaw direction and the β image metric across 0.1 coarse samplings of fill intensity. ...................................................................................................................... 103
List of Terms

Ambient Light – Ambient light is any form of lighting meant to simulate or account for indirect lighting in a scene. With regards to design, ambient light is often used to describe light that is added to give a scene a minimum brightness. With regards to computer rendering, ambient light is any one of many techniques meant to substitute for a lack of indirect lighting in a visualization algorithm. Rasterization for example typically includes a form of ambient light.

Back Light – A back light is a light source placed behind an object relative to the point of view of the viewer. It is used to create a rim of light around the edge of the object to accentuate its silhouette and isolate it visually from the background.

Bidirectional Reflectance Distribution Function (BRDF) – A bidirectional reflectance distribution function (BRDF) is a mathematical function that describes how a surface reflects illumination. It maps incoming illumination to outgoing illumination. A Lambertian reflectance function is an example of a BRDF.

Cameo Lighting – Cameo lighting is an extreme form of chiaroscuro lighting that creates areas of almost complete darkness and intense brightness.

Chiaroscuro Lighting – Chiaroscuro lighting is a style of lighting most readily identified with the painter Rembrandt. This style of lighting creates shading and light contrast that gives objects a realistic or natural appearance.

Computational Cost – Computational cost refers to the computing resources needed to evaluate an algorithm. Often, it is synonymous with “time needed to calculate” but it can refer to costs other than time, such as storage space.

Diffuse Light – Diffuse light is light that is reflected equally in all directions by a surface. With regards to a light source, the term diffuse refers to an unfocused illuminator that does not create hard, easily identified areas of shadow.

Direct Light – Direct light is light emitted from primary sources of illumination. A primary source of illumination can be a light bulb for example. In the context of computer rendering, direct light can also be used to refer to non-global illumination algorithms, where lighting is not calculated inherently during the process of generating a rendered image.

Entropy – The entropy referred to in this dissertation is information entropy. Entropy is a quantification of the information potential of a variable. Maximizing entropy essentially means to maximize the information communicated by a variable.

Exposure – In film development, exposure is the duration of time that film is allowed to interact with a source of light. Exposure fundamentally controls the contrast of a photographic image. In computer rendering, the effects of exposure are often simulating to produce a more photorealistic image.

Fill Light – A fill light often compliments a key light. It is a secondary, usually diffused source of illumination that is used to control overall light contrast on an object.
Flat Lighting – Flat lighting is a lighting technique that produces very little shading gradation. Flat lighting is generally used to create a happy or comical lighting appearance and it usually appears less realistic or natural than chiaroscuro lighting.

Global Illumination – Global illumination refers to a class of computer rendering algorithms that calculate illumination inherently during the process of generating a rendered image.

Hard Light – Hard light is highly directional light that produces shadows with clearly defined shapes and edges. Lighting with a great deal of contrast between light and dark areas is also generally referred to as hard light.

Indirect Light – Indirect light is also called “bounce” light. Indirect light is any light that is emitted into an environment from surfaces that reflect the primary sources of illumination. For example, a light bulb is typically a form of direct light while a tabletop near a light bulb is a form of indirect illumination.

Irradiance – Irradiance is a unit of measurement describing the amount of light incoming or incident to a particular area.

Key Light – A key light is the primary light affecting an illuminated object. The key light establishes dominant light direction cues.

Lambertian Reflectance – Lambertian reflectance is the most common algorithmic model for diffuse reflectance. Diffuse reflectance describes a surface that reflects light in all directions equally.

Lightmap – Lightmaps or “baked” lighting refer to a computer visualization technique. Light is pre-calculated and stored and then later applied at runtime in real-time rendering. Lightmaps have recently been supplanted by spherical harmonics because while lightmaps can only capture radiance, spherical harmonics can capture irradiance.

Notan Lighting – Notan lighting is a style of lighting that focuses on contrasting roughly homogenous areas of tone. Notan lighting is less naturalistic than chiaroscuro and may appear “cartoony” but work identified as Notan is not necessarily a cartoon.

Photorealism – Photorealism literally means “looks like a photo”. In the context of painting, it refers to painting styles that produce images that look like photographs. In the context of computer rendering, it refers to the pursuit of rendering algorithms that produce generated images that are indistinguishable from photographs of the real world.

Psychophysics – Psychophysics is an area of study focusing on the relationship between physical stimulus and subjective perception. For example, the relationship between a physical measurement of a sound’s intensity in decibels and the perceived loudness of the sound by a listener is a psychophysical question.

Radiance – Radiance is a unit of measurement describing the amount of light emitted or outgoing from a particular area.

Radiosity – Radiosity is an algorithm used to generate images of illuminated virtual objects. It is similar to ray tracing. Like ray tracing, it is a global illumination algorithm such that light is calculated in the process of generating an image of an
object. Radiosity uses a different technique than ray tracing and generally produces soft lighting while ray tracing generally produces hard lighting.

**Rasterization** – Rasterization is an algorithm used in computer visualization to generate images of virtual objects. It refers to the process of taking data of three-dimensional objects and converting it into a two-dimensional image that can be shown on computer displays. Rasterization is the most common technique used in real-time rendering due to its computational efficiency and due to the common availability of accelerated rasterization using dedicated computer hardware.

**Ray Tracing** – Ray tracing is an algorithm used to generate images of illuminated virtual objects. Ray tracing is a global illumination algorithm such that the calculation of light is inherent to the generation of an image of an object. This is in contrast to rasterization, where lighting is a special case calculation.

**Silhouette Lighting** – Silhouette lighting is a style of lighting where the dominant or possibly only source of light affecting an object is behind the object with respect to the point of view of the viewer. Only the outside edge or silhouette of the object is typically visible.

**Soft Light** – Soft light is light that produces shadows with poorly defined edges and shapes. Lighting with very low contrast, containing very little or no completely dark areas, is also called soft lighting.

**Specular Light** – Specular light is light that is reflected in only one or a very small cone of direction. A specular surface appears varying degrees of shiny or glossy.

**Spherical Harmonics** – Spherical harmonics are a set of periodic functions. In the context of computer graphics, spherical harmonics have recently become popular for encoding irradiance relative to a surface point. Spherical harmonics can in particular be used to pre-calculate and store illumination that is later applied at runtime in real-time rendering.

**Texture** – Texture refers to the perception of small details on an object. With regards to computer rendering, a texture is an image that is mapped onto the surface of an object to give it visual surface detail.

**Three-point Lighting** – Three-point lighting is a lighting technique that divides lighting usage into three goals: key lighting to control direction cues, fill lighting to control light contrast, and back lighting to control an object’s silhouette and separation from the background.

**Under Lighting** – Under lighting is a lighting technique where sources of light are placed low, generally below the point of view of a viewer. This form of lighting creates an uneasy and unnatural appearance and is often used in horror or similar genres.
List of Equations

Eq. 01. The computer rendering equation. ............................................................ 30
Eq. 02. Single-scale retinex. ................................................................................. 51
Eq. 03. Relationship of an illumination image to an input lighting design image. 51
Eq. 04. The Y component of the XYZ color space from the sRGB color space. 52
Eq. 05. A component C’ of the sRGB color space from the RGB color space. 52
Eq. 06. Term β representing one of two image metrics for light direction. .......... 53
Eq. 07. Calculation of light entropy from (Gumhold 2002). .............................. 53
Eq. 08. Probability of a pixel falling into bin i. .................................................. 53
Eq. 09. Selection of the bin for a pixel during the calculation of light entropy. ... 53
Eq. 10. Form of a cubic polynomial used for encoding the response surface. .... 103
Eq. 11. Mean square error used for linear regression. ...................................... 103
Chapter 1: Introduction

Video games have grown to a billion dollar entertainment industry with a high profile title like Grand Theft Auto 4 earning $500 million dollars in its first week of availability (Richtel 2008). But games extend well beyond entertainment. Many researchers (White 1984; Glifford 1991; Jasinski and Thiagarajan 2000; Aldrich 2003; Gee 2003; Prensky 2003; Yucel, Zupko et al. 2006) have argued for the importance of video games for learning and “serious play”. “Serious Games” (Initiative 2008) incorporate research into everything from military training, to psychotherapy, and communication. Although the interests into video games are diverse, certain inherent factors of game media remain true across topics and applications. Most games are visual and anyone intent on creating a game product must consider the design and implementation of a game’s visual function and aesthetic. Therefore, the focus of this dissertation is on one factor that is inseparable from all visual experience in games: lighting.

The real world is revealed by light. Visual perception and visual artistic expression are fundamentally dependent on light. The great works of Rembrandt and other artists depend on contrast and the visual intensity created by mimicking the behavior of light. Scientists and engineers study and manipulate light for practical and theoretical goals. The very structure of modern life is possible because we have transcended the constraints of natural day and night cycles and can effectively operate 24 hours a day.

Much as it shapes the real world, light also shapes the experience of a video game. A player’s visual focus is inherently guided by light. Intense light can be used to isolate an important object or subdue unimportant background information (see Fig. 01). Color can convey “warmth” or isolation, anger, sadness, joy, or a sense of foreboding (Millerson 1991). Complex interactions of light direction, contrast, shadow, and color combine to create a final visual narrative.
Fig. 01. This figure of Gordon Freeman from the video game Half-Life 2 (released in 2004 by Valve) illustrates the use of lighting to control visual appearance. A rim of white light surrounds the character, helping to isolate him from the background and draw the viewer’s attention to the character.

Modern computer visualization, the process of generating an image of a virtual thing, now affords great control over light. Virtual illumination can produce photorealistic images (literally images that look like a photograph) such as the bowl of fruit shown in Fig. 02. It can also produce entirely non-photorealistic yet stylistic images such as the video game shown in Fig. 33. Independent of a video game’s purpose, whether to entertain, inform, or educate, lighting controls our perception of a game. Virtual lighting is sophisticated enough that the visual quality of a virtual piece is more determined by the skill and competency of the designer and how she can effectively control light and less determined by the quality or physical accuracy of computer visualization algorithms. Thus, the role of a lighting designer is crucial.
Fig. 02. This bowl of fruit from (Birn 2006) illustrates light “cheating”. The bowl of fruit pictured in the top left image is illuminated by a window, represented by the red box in the bottom image. The top right image is the same bowl of fruit illuminated by a “cheated” light source, meant to improve the aesthetics of the scene while also appearing to still be motivated by the window. The cheated light source is represented by the green box in the bottom image.

Lighting design is the purposeful control of illumination. In cinematography and theatre, the purpose might be to entertain or inform (Shelley 1999; Zettl 1999; Ablan 2003). In architecture, the purpose might be to create mood (Flynn 1977) or to maintain illumination for reading and work. In engineering or medicine, the purpose might be to reveal surface orientation, detail, and edges (Gooch, Gooch et al. 1998). In video games, the purpose is often a combination of all of these.

Video games must create mood while remaining clear with their visual cues and entertain while delivering pertinent information. Lighting design in video games is a complex balancing of physical accuracy with desired appearance. For example, in Fig. 02 the top left image is a physically accurate rendering of a bowl of fruit. The fruit bowl is illuminated as it would naturally be from light coming through a window. The top right image is “cheated”. The light source was altered to produce an aesthetically “better” image (Birn 2006), one that did not “flatten out the round forms” (Birn 2006). Effective lighting is therefore not just strict placement of lights to make an object visible based on the physical nature of light but a complex, subjective, functionally and aesthetically motivated design and “cheating” of lighting.

Depending on the domain, lighting designers will cheat lighting in various ways. In a film for example, lights can be placed in unrealistic places if they are hidden outside the frame of the camera. Digital media, such as computer generated movies or video games, offers even more flexibility. Lighting can be placed anywhere and even violate physical law, using mathematical models that produce whatever results an artist desires. A designer may even “paint” desired shading directly on a surface (see
interview 10_29_01 in Appendix D) similar to how a painter shades a canvas. Independent of the tools available to a designer, lighting design is invariably an iterative “tweaking” process. A designer applies lighting, evaluates the result, and then adjusts the lighting until she is satisfied.

This tweaking process can be problematic in video games however. In Fig. 02 for example, the final light source position and orientation were chosen based on the position of the bowl of fruit, the table, the window, and the appearance and physical properties of the bowl of fruit itself (Birn 2006). In a video game however, the user has control. A user might choose to move the bowl of fruit, knock it off the table entirely, or eat an apple, changing the appearance and properties of the bowl. If this happens, the precisely chosen settings for light sources would no longer achieve the goals originally desired by a lighting designer.

One technique that video game lighting designers use to address this problem is to impose constraints on the virtual environment. For example, a designer might simply prevent a player from altering the bowl in Fig. 02. The drawback of this technique is that it fundamentally limits the unique strength of video games: the ability to interact.

Another technique is to focus on lighting an object’s environment in such a way that the object will be lit “reasonably” well in all cases. For example, the light that represents the window in Fig. 02 could be tweaked so that no matter how the bowl was moved or manipulated, the “round forms” (Birn 2006) of the fruit would still be roughly maintained. The drawback of this approach is that it often requires sacrificing striking visual imagery, such as using highly contrasted areas of light and dark, in exchange for a lighting design that works as often as possible. This technique is also not always possible without resorting to restricting an object’s changeability as described previously.

A solution sometimes used to combat this problem is to attach a “rig” of lights to an object (see interview 10_29_01 under Appendix D). This rig is composed of several lights that follow the object, maintaining a desired lighting configuration. The lighting created by this rig is then blended with the natural lighting state. The difficulty of this approach is that it is labor intensive. It requires a lighting designer to manually specify all the different states for the attached rig and to program the conditions in which these states will change at runtime. It is also rigid. If a designer fails to predict a possible context and does not code a state for that context, the results in that context may be undesirable.

As a side note, video games can also use a variety of special case effects to control visual focus or to achieve other lighting goals. For example, to indicate important objects, particle effects can be used (see Fig. 03). These types of effects, while effective, are usually overt and unrealistic. They are typically not appropriate when a natural visual appearance is desired. They are also completely tied to a specific video game and art style.
This dissertation proposes an alternative, programmatic solution to accommodating user interaction. Instead of fixing the bowl of fruit in place in Fig. 02, a user could still manipulate the bowl. When moved, a software system would attempt to adjust the light represented by the green region in Fig. 02 to maintain the “round forms” (Birn 2006) while still maintaining the illusion that the light is coming from the window. This is essentially the attached rig solution previously described with the addition of software algorithms to make decisions about when and how to change the rig based on context. This “automated lighting” solution has the strengths of attaching a rig of lights to an object. Lighting goals can be maintained across changes in context. It also has the advantage of a broad or general purpose lighting design. An object’s environment can be lit naturally and the system will attempt to reflect that natural lighting state at all times.

1.1 Automated Lighting

Previous research into automated lighting includes the Expressive Lighting Engine (ELE) (Seif El-Nasr 2003). ELE used an attached rig of lights as described in the previous section. It based its lighting decisions on theories derived from theatre and film. For example, ELE had a metric that quantified the subjective notion of “visual tension”. A designer could then instruct ELE to increase or decrease visual tension and ELE would react by moving lights and changing their color to achieve desired visual tension while also considering the natural lighting of a space. ELE would attempt to find a balance between the positions and colors of lights necessary to achieve desired visual tension and where the lights were naturally located in a scene.

A shortcoming of ELE was that a designer needed to map her own personal notions (such as visual tension) to ELE’s numerical metrics. Further, ELE did not consider the
final appearance of an object when finding a final lighting arrangement. It only based its decisions on the positions and colors of lights and the rough shape of an object. Therefore, ELE would illuminate a girl in a white dress the same as a tree stump if they both had roughly the same height and girth, even though the way each responds to light is very different.

This dissertation presents a new automated lighting system, the System for Automated Interactive Lighting (SAIL). SAIL’s design was intended to address many of the shortcomings of ELE.

1.1.1 Overview
While a user plays a video game, SAIL runs in the background attempting to control illumination on an object to maintain goals specified by a lighting designer. These goals are specified using an image (see Fig. 04). This image represents the shading direction and contrast that a designer wants maintained on an object. The image does not necessarily need to be of the object that SAIL is lighting at runtime. For example in Fig. 04, an image of a shaded sphere is used to represent desired lighting for a model of a young girl that is illuminated at runtime.

Fig. 04. This figure illustrates how an image of a shaded sphere provides the illumination example that SAIL uses to light a 3D model of a young girl (Milam 2008). SAIL chooses light sources to recreate the intensity contrast and apparent light direction as indicated by shading present in the example image.

Unlike simply attaching lights to an object as described previously, SAIL considers the context of an object while trying to maintain a lighting goal. For example, if a designer specifies a goal indicating light coming from above but all the lights in an environment are from below, SAIL will adjust the positions of lights affecting an object to achieve a compromise between lighting from below and lighting from above. SAIL does not simply classify lighting as above and below based on the angle of the light sources however. Rather, it classifies lighting as above or below (or other
features, such as high and low contrast) based on the appearance of the object it is lighting. For example, a cat, a tree, and a human will each have different shading and contrast cues if a light source is placed in the same position relative to each. SAIL will choose a light source position that achieves similar shading gradients and contrast on all three objects, attempting to match the contrast and shading gradients in the example image.

1.1.2 Theory
SAIL’s behavior is determined by two key theories. The first described in Chapter 2 is the psychophysical theory of color constancy. Color constancy is the human visual system’s ability to separate illumination from reflectance. This is a heuristic process that solves an ill-posed mathematical problem. SAIL uses a specific formulation of color constancy called retinex to isolate illumination information in an image. This illumination image is then analyzed to quantify the lighting goals specified by an image.

The second theory used by SAIL is the cinematography technique of three-point lighting (see Fig. 05), described further in Chapter 2. Three-point lighting is a constrained approach to lighting an object which reduces the necessary lights and their configurations to essential components for controlling various visual effects, particularly, contrast and direction. This reduction is necessary to allow SAIL to be computationally efficient enough to operate in a real-time rendering environment such as a video game.

Fig. 05. This figure from (Birn 2006) illustrates film three-point lighting. Three-point lighting divides lighting an object into three lights for directly controlling essential components of the visual presentation of an object.

SAIL is also constrained by related work in automated lighting and the mathematics behind computer rendering. These constraints are discussed in detail in Chapter 3.

1.1.3 Contributions
SAIL contributes to the field of automated light as an artifact, as a design tool, and as an experimental instrument.
As an artifact, SAIL contributes two functions that differ from related work. First, SAIL includes an image analysis component to understand light features that does not require correlated data. In contrast, related work that includes an image analysis component requires correlated geometry or other correlated data. Second, SAIL applies this image analysis to real-time lighting, allowing it to adapt lighting based on an object’s appearance. Existing work such as ELE uses a rough approximation of an object (such as an object’s shape and position) and does not consider the object’s appearance.

The study presented in Chapters 6 and 7 indicates that as a design tool, SAIL may be useful as a pre-visualization or storyboarding tool. Chapter 8 also discusses how SAIL might be useful in supplementing an existing video game pipeline or to reduce the complexity of creating a virtual environment for individuals who are not experts in lighting design.

SAIL also offers the possibility of simplifying existing experimental work and the possibility of facilitating new experimental work. Construction of previous experiments into lighting required time consuming creation of virtual environments. Lighting designs for these environments were specific to the environment and the experiment. Changes to the experiment required changes to the lighting designs. SAIL’s ability to adapt lighting designs to context changes might allow a single design to be created and then applied to multiple environments and experiments. This is explored further in Chapter 8.

1.2 Organization

The rest of this dissertation is organized as follows:

- **Chapter 2: History and Theory of Light and Lighting Design** – this chapter discusses the art of lighting design and the psychophysics of light perception, specifically the psychophysical theory of color constancy. The art of lighting design shapes our subjective interpretation of light while psychophysics describes our objective perception of light. Consideration of both was important to the design of SAIL.

- **Chapter 3: History and Theory of Light in Computer Rendering** – this chapter discusses related work in the field of inverse lighting and the algorithms used for real-time computer rendering. Both of these topics constrained the design of SAIL. SAIL was built in consideration of existing work, to extend and contribute to the field, and was built to work with existing real-time rendering technologies to increase its practicality.

- **Chapter 4: System for Automated Interactive Lighting (SAIL)** – this chapter discuss the specifics of SAIL at a high-level, how SAIL was designed in consideration of existing theory and practice, how SAIL differs from or complements related automated lighting work, and the details of the current implementation of SAIL.
• Chapter 5: Algorithms of SAIL – this chapter discusses the mathematical and technical details of the algorithms that SAIL uses for image analysis and adaptation of lighting at runtime.

• Chapter 6: Evaluation of SAIL - Study Design – this chapter describes a qualitative study designed to evaluate the contributions of SAIL as an artifact. Specifically, it discusses considerations behind the implementation of SAIL used in study sessions, interview questions, and the protocol used to conduct sessions.

• Chapter 7: Evaluation of SAIL - Study Results – this chapter details the results of the study introduced in Chapter 6. Specifically, interviews are presented in support of SAIL’s contributions as an artifact and to describe several observations regarding what lighting experts think about the design of light. These observations support SAIL’s contributions as a design tool and may also be of use to the field of automated lighting in general.

• Chapter 8: Conclusions – this chapter presents final conclusions of this dissertation. Specifically, SAIL’s contributions as an artifact, as a design tool, and as an experimental instrument are discussed. This chapter ends with an overview of future work left for SAIL and closing thoughts.
Chapter 2: History and Theory of Light and Lighting Design

SAIL’s design was impacted by both the art of lighting design, which shapes our subjective impressions of light, and the psychophysics of light, which describes our objective perception of light. Section 1 of this chapter discusses the nature of film media and influential lighting styles that have greatly impacted lighting design sensibilities. Section 2 discusses color constancy, a fundamental visual process that shapes our perception of light.

2.1 The Art of Rendered Light

Fig. 06. This figure from http://erich.realtimerendering.com/ shows an image created with a rendering system designed by Eric Haines in 1991.

The art of rendered light is the practice of manipulating light for aesthetic goals while maintaining functional needs in a virtual domain. It is an explicit manipulation of visual information to convey meaning and form. Rendered light has progressed from the explorations of graphics researchers (see Fig. 06) to the emotional and gripping computer generated movies of Pixar (see Fig. 07). The canvas of the computer has stretched and expanded but the underpinnings have stayed grounded in long standing practice.
Fig. 07. This image from the movie *Monsters, Inc.* (released in 2001 by Disney/Pixar) illustrates the advancement of computer generated art.

Photography and cinematography are two artistic domains that contribute greatly to the sensibilities of the rendered art community. Much time and effort is spent achieving photorealism or work that looks like a photograph. While some of this is dependent on the physics of rendered light much is determined by artistic practice. In particular, capturing a photorealistic image requires recreating many of the quirks of cinematography and photography that are a result of the mechanics of those media.

An interesting example is lens flare (see Fig. 08). This color artifact is a result of light interacting with the structure of a camera, reflecting and scattering in the lens. As discussed in (Millerson 1991), it is usually considered an error in cinematography and photography and great effort is expended to avoid it. Ironically, lens flare has appeared extensively in virtual art. As mentioned in (Bim 2006), lens flare is often used in computer generated imagery to help create an impression of an extremely bright light.
This figure shows the somewhat odd usage of lens flare in a computer rendered image. Lens flare is an artifact of light interacting with a camera that is generally deemed a mistake in photography and cinema. Nevertheless, in computer rendering it is sometimes used to help add realism to a piece.

More pertinent for the work of this dissertation is how the general mechanics of photography and cinematography determine virtual art practice. In particular, the control of contrast and the direction of shading are crucial to the production of a good photorealistic image. The next two sections describe how the mechanical properties of photographic film may encourage photographers to focus on the control of light contrast and light direction to control visual elements of their final work.

2.1.1 Photographic Film and Contrast
Photographic film is used in both photography and cinematography to capture light. Film is plastic that is light sensitive. The material of film responds when exposed to light, forming an invisible image (Gallardo 2001). After exposure, the film is then developed using chemical processes to reveal the invisible image. Various types of film use various forms of development but cinema and photography generally use types that can be precisely controlled.

The process of controlling a film’s exposure and development fundamentally controls the light contrast in a resulting image. Film has a limited dynamic range like any image detector. Overexposure of film to light that is too bright will result in an image that is washed out with a loss of detail in the bright areas. An underexposed film will result in an image that leaves out shadow details, form, and texture (see Fig. 09 for a comparison of underexposure and overexposure in photography).
Control of light contrast is not an automatic process and as such, it contributes much to the art of photography. Information is present in different forms in different ranges of light intensity. The right amount of tones must be present in the right intensity ranges to create the correct aesthetic impression. Due to the mechanical nature of photographic film, this need often results in a focus on bright and dark tones.

Fig. 10 shows a typical Hurter and Driffeld curve from (Gallardo 2001) for photographic film. This curve relates light exposure to film density. Film density controls the contrast of photographic film. Notice the linear center and nonlinear ends of the curve. Mid tones are relatively easy to control. Bright and dark tones are more difficult. Changes to exposure create very large or very small changes in resulting contrast in the very dark and very bright regions of the curve. This requires photographers to focus more on the control of very bright and very dark tones in their work.

Although the mechanics of photographic film do not apply to the mechanics of virtual rendering, the sensibilities afforded by these mechanics have translated into the virtual domain. A quick glance at a modern book (Birn 2006) on lighting in computer generated imagery shows an entire chapter (Chapter 3) devoted to the control of shadow and another (Chapter 6) devoted to simulating the effects of film exposure. In short, control of light contrast is important for the creation of a photorealistic image.
2.1.2 Lighting Direction

Light direction is as important to photography and cinematography as light contrast. A possible explanation for this is the lack of binocular depth perception cues in these media. Our perception of depth comes from many visual cues. In common viewing of the real world, a great deal of this information comes from the relative information of our two eyes. When viewing a photograph or film however, we lack two discrete sources of information. As a result, a great deal of pressure is placed on these media to overemphasize or deemphasize monocular depth cues to control the impact of depth.

For example, texture is an important visual cue for depth. The clarity of texture on a surface gradually diminishes as the surface is farther away. The perception of texture is a fundamental property of the direction of illumination (see Fig. 11). Humans also use occlusion\(^1\) to determine the distance of an object. Occlusion can be controlled by the strength of an object’s silhouette which is fundamentally a property of light direction (see Fig. 12).

---

\(^1\) Occlusion occurs when one object blocks another object from view.
2.1.3 Usage of Contrast and Direction
A classic usage of light contrast is the film noir style of lighting. This is an absolute form of contrast. Low-key lighting creates abrupt changes between areas of extreme dark and extreme light. Mid tones are used sparingly. Much of this style is identified with the black and white era of film, which adds to the look (see Fig. 13).

Fig. 13. This image from the movie Out of the Past (released in 1947 by RKO Pictures) demonstrates classical film noir in black and white; a dark, high-contrast style of lighting.
Modern film noir may use color or be filmed in black and white to capture a traditional look. Although color adds a new dimension to the aesthetic of a film, the essentials remain the same: high-contrast, extreme light and dark, and bold imagery (see Fig. 14).

Fig. 14. This scene from the movie The Dark Knight (released in 2008 by Warner Bros.) is a modern day example of film noir using color, illustrating the same essential usage of extreme light contrast to create evocative imagery.

The silhouette style is common in film noir but it is also a style in its own right (Fig. 15). Silhouette lighting accentuates form over detail. Texture is all but eliminated. This style creates a sense of mystery and unsettledness. The horror genre of cinematography often uses this technique to make the “monster” more ominous.

Fig. 15. This photograph titled Nomad musician by an unknown artist demonstrates silhouette lighting style.
Even more than silhouette, horror commonly uses the lighting technique of under lighting or lighting from below. Lighting from below is used to increase shock and unease (see Fig. 16). Psychophysical work (Ramachandran 1988) has documented a preference by humans for lighting from above which may explain the power of this technique. Ramachandran reported that when shown images of shaded spheres that can be interpreted as either concave regions with light from below or convex regions with light from above, participants were more likely to identify the spheres as convex regions with light from above.

At the opposite extremes of silhouette and under lighting are soft lighting styles. Bright and usually cheery, these styles create a warm, soothing atmosphere (see Fig. 17). There is very little contrast between shadow and highlight. Extreme dark is kept to a minimum or eliminated completely.

Fig. 16. This figure shows an image of Boris Karloff as the monster in the 1931 film Frankenstein. The usage of under lighting (lighting from below) increases the unease and unnaturalness of the monster, helping to create the illusion of something not quite human.

Fig. 17. This figure from http://www.chromasia.com/ demonstrates soft or high-key lighting.
Soft lighting styles are also very common in fantasy or other storybook worlds (see Fig. 18). This style of lighting is very effective at creating an inviting yet otherworldly atmosphere, instilling a sense of mystery and wonder. Soft lighting is often coupled with bright, saturated colors, increasing the power of this effect.

![Fig. 18. This screenshot from the video game *Guild Wars* (released in 2005 by ArenaNet) illustrates the effective use of soft lighting to create a fantastical, storybook world. Bright colors and soft shadows create a warm yet otherworldly atmosphere.](image)

Light direction can be used to overcome the frame of photographic media. There is no peripheral vision in an image. We see what the artist permits us. The direction of light serves as a grounding visual cue, placing the frame in a larger context (see Fig. 19).

![Fig. 19. This image from the video game *Half-Life 2: Episode 2* (released in 2007 by Valve) demonstrates the effectiveness of light direction in implying larger context. The shafts of light behind the character indicate a window or other opening above the frame of the image.](image)
There are several general styles to lighting that combine the control of contrast and direction into a holistic approach. Chiaroscuro style is readily identified in the work of Rembrandt (see Fig. 20). Chiaroscuro is the pursuit of naturalism and modeling. It is focused on the creation of gradual shading and poignant light contrast. Rembrandt carefully controlled lighting in his paintings. Important areas are illuminated while unimportant areas are left completely dark. Shadows create contrast but are not excessively dark (most of the visible cast shadows shown in Fig. 20 are dark but also quite soft).

![Fig. 20. This image of St. Peter in Prison, Rembrandt van Rijn, demonstrates chiaroscuro style lighting.](image)

While the works of Rembrandt are quickly identified as chiaroscuro style lighting, this is not the only chiaroscuro style. Cameo lighting is “chiaroscuro lighting pushed to its extreme” (Zettl 1999). Camera lighting is an extreme form of chiaroscuro (see Fig. 21). Much of a character is shrouded in complete darkness. Sharp, clearly defined cast shadows are created. Cameo lighting is such an extreme lighting style that it is difficult to use in television or film (Zettl 1999). The contrast created is hard to capture on film and it is difficult to maintain the aim of such highly focused light on a moving actor.
Fig. 21. This image from a production of Cabaret illustrates cameo style lighting (see http://www.michaelskinnerld.com/cabaret). It is an extreme form of chiaroscuro lighting, hiding much of a character in darkness and creating sharp, clearly defined cast shadow.

A famous example of the difficulty of capturing extreme light contrast is the work of Stanley Kubrick for the movie Barry Lyndon. Kubrick wanted to capture scenes in difficult to capture natural lighting conditions, such as a scene shot in candlelight shown in Fig. 22. Doing so required multiple takes and the creation of special camera lenses (DiGiulio), derived from lenses originally created for use by NASA in the Apollo Moon Landing program. The effort resulted in a unique and fascinating visual style, producing imagery of live actors strikingly similar to the look of an oil painting.

Fig. 22. This image from the movie Barry Lyndon (released in 1975 by Warner Bros.) illustrates an extreme light condition. This scene was shot in natural candlelight using specially made, ultra-fast lenses. Typically, a scene like this would be shot by using small amounts of fill light to increase the intensity of ambient lighting to make the scene easier to capture with traditional lenses.
Notan style is a concept of Japanese origin (Millerson 1991) originally identifying a style of illustration but the term has grown to classify a wide variety of imagery (see Fig. 23). Notan is in contrast with chiaroscuro. It emphasizes form and “important” tones over modeling and shading. Objects are generally rendered using flat contrasting areas of tone. This style does not necessarily create a cartoon but it can be described as less naturalistic than chiaroscuro.

Fig. 23. This painting from http://tinyurl.com/bnvc7u illustrates Notan style lighting.

2.1.4 Three-point Lighting
Many of the usages of light and contrast described in Section 2.1.3 can be accomplished either entirely or partially using a lighting technique known as three-point lighting (Birn 2006). Three-point lighting consists of three lights: a key light, a fill light, and a back light. The key light provides a dominant sense of direction and base illumination, the fill light reduces overall tonal contrast but contributes minimal direction information, and the back light rims the object. As an example of three-point lighting, Fig. 16 of the monster from Frankenstein could be created with a three-point lighting arrangement with the key light down and to the right, the fill light up and to the left with a low intensity, and the back light behind, above, and slightly to the right of the monster.

2.2 Color Constancy
In addition to aesthetic choices made by a lighting designer, our perception of light is also a psychophysical process. Psychophysics is a subfield of psychology that studies the relationship between a measurable, physical stimulus and the subjective, perceived response. For example, the relationship between a measured intensity of light and the perceived brightness of that light by a human observer is a psychophysical relationship. One such psychophysical process that is essential to visual perception is
color constancy. Color constancy is the human visual system’s ability to adapt vision to achieve constant color under varying illumination conditions. It is our visual system’s ability to essentially isolate an object’s surface color from illumination and then remove or normalize that illumination. This is the reason that a gray surface still appears gray in varying illumination conditions (Adelson 2000) or a green apple will still appear green when viewed in either sunlight or under fluorescent lights.

A survey and comparison of the many theoretical models of color constancy is presented in (Barnard, Cardei et al. 2002). Although there are many different models for how exactly color constancy works, several threads exist between them. Many assume that the illumination affecting an image is diffuse only. It is also commonly assumed that light is “sufficiently distant” (Barnard, Cardei et al. 2002) from the object such that the effects of diffuse only illumination are low frequency. This is explained further in (Ramamoorthi and Hanrahan 2001; Ramamoorthi and Hanrahan 2001) but briefly, sufficiently distant diffuse illumination cannot create abrupt intensity changes on the surface of a convex object. Therefore, any abrupt changes must be due to the surface reflectance properties of the object. By exploiting this fact, information due to surface properties and information due to lighting can be isolated. Another assumption of many color constancy models is the independence between the cones of the human eyes.

The human eyes contain cells known as cones and rods. The cones are responsible for color vision while the rods are responsible primarily for night (low intensity) vision. A cone in the human eye is tuned to a particular subset of wavelengths of the entire spectrum of light. The human eye contains three types of cones, each tuned to a separate wavelength range, which results in our trichromatic color vision. In many color constancy models, it is assumed that color constancy processing is the same for each cell in the eye and it is also assumed that the processing for one cell is independent from the processing of the other cells. The Purkinje effect partially supports this hypothesis (Frisby 1980). The Purkinje effect is the tendency for our vision to shift towards the blue portion of the light spectrum during low intensity light conditions (due to the rod cells responding better to light in the blue ranges of the spectrum and the rods becoming dominant in low intensity light conditions). This effect can result in mismatches in our color constancy processing, where the relationship between objects will change depending on the intensity of lighting conditions. For example, the relationship between a red flower and its green leaves may change depending on whether they are viewed at night or in daylight. This mismatch is believable if the processing of each cell remains independent and does not account for the blue shift that occurs during low light conditions.

---

2 Diffuse illumination is light that is reflected equally in all directions by a surface. See Chapter 3 for more information about diffuse and other forms of illumination.
Chapter 3: History and Theory of Light in Computer Rendering

SAIL’s development was constrained by two domain considerations. As will be described in Section 1 of this chapter, SAIL was constrained by related work into automated software lighting systems. SAIL was also constrained by the underlying mathematics used in real-time computer rendering. This is discussed in Section 2 of this chapter.

3.1 Related Work

The field of this dissertation is inverse lighting, which is a subfield of the area of inverse rendering. The broader topic of inverse rendering is outside the scope of this dissertation but an excellent survey is presented in (Patow and Pueyo 2003). An example of an inverse lighting problem is, given an image of a scene, determine the positions, orientations, colors, and other parameters of lights that were affecting that scene. The information available to an application other than an image differs between different problems. For example, in addition to an image, 3D data of the structure of the scene pictured in the image might also be available. Or reflectance functions describing how the surfaces pictured in the image respond to light might be provided. Inverse lighting work can be roughly categorized into three groups: architectural inverse lighting, perceptual inverse lighting, and interactive inverse lighting.

3.1.1 Architectural Inverse Lighting

Architectural inverse lighting is a subfield of inverse lighting. This body of work is focused on removing iteration from the task of designing light for architectural spaces.

Traditionally when designing light for an architectural space, a lighting designer places lights in a 3D virtual representation of that space using a tool such as Autodesk AutoCAD. After placing these lights and rendering the results, the designer evaluates the results to determine if they meet required criteria. These criteria might include minimum intensity in portions of the space, energy usage, and the “feel” of the space, such as whether the space feels spacious or warm.

This traditional process is iterative and time consuming. A designer places lights, renders the result, evaluates the results, and then repeats this process until satisfied. A much more efficient process would be possible if a designer could directly code her required criteria and then instruct a system to find a light source configuration that satisfies that criteria. For example, given the criteria of a “warm” space with sufficient visibility for reading, a software system would place lights in valid places to achieve this goal. This improved workflow is the underlying problem that all architectural inverse lighting work has attempted to solve.
(Kawai, Painter et al. 1993) was one of the first software systems designed to remove the need for iteration in architectural lighting design (see Fig. 24). This system found angles and intensities for lights given: 1) light positions manually specified by a user, and 2) numerically defined goals for the lighting design. Some of these goals were literal in nature, such as desired intensity at a point within the scene being illuminated. Other goals were subjective, such as how “spacious” a room felt. The subjective terms were derived from psychophysical work conducted in (Flynn 1977). Flynn explored participants’ subjective experiences to lighting using self-reported terms. For example, Flynn identified correlation between lighting positioned overhead vs. lighting positioned on walls and self-reported descriptions of a space as “unpleasant” (overhead) vs. “pleasant” (walls).

![Fig. 24. This image is of an automatically illuminated virtual space from (Kawai, Painter et al. 1993). Kawai’s system found light intensities and angles given light positions and a set of metrics meant to represent subjective lighting goals such as “spaciousness” derived from (Flynn 1977).](image)

(Costa, Sousa et al. 1999) used a similar approach to Kawai. Costa’s system is the most recent in the field of architectural inverse lighting and is the most complete solution (see Fig. 25). Costa’s approach made only three assumptions about an environment: 1) surface reflectance can be described with symmetrical bidirectional reflectance distribution functions (SBRDFs), 2) the environment has no participating media, and 3) the light model used in the rendering environment obeys the photon nature of light.

---

3 A symmetrical bidirectional reflectance distribution function (SBRDF) is a reflectance function for which an inverse can be calculated. A reflectance function describes how a surface reflects light, mapping incoming light to outgoing light.

4 Participating media are small particles that absorb, emit, or scatter light. An example of participating media is fog.
Fig. 25. This figure shows a virtual scene of an office illuminated by the system in (Costa, Sousa et al. 1999). The system in this case was given goals for the intensity of lighting on different surfaces in the environment and was restricted to placing lights along the ceiling. It was otherwise free to choose light source positions and intensities.

Within these assumptions, Costa allowed designers to specify almost any illumination goal using scripting. For example, a designer might code a rule that constrained intensity on a surface within a specific range and code a rule that constrained light locations to the ceiling of a space. Costa’s system would then find light sources that fulfilled this and all other rules for a given space.

3.1.2 Perceptual Inverse Lighting
Perceptual inverse lighting is the second subfield of inverse lighting. This field is focused on completely eliminating the need to design lighting for domains where lighting design is a peripheral need rather than a central focus.

For example, in the domain of medical visualization, medical experts want to visualize parts of the human body (see Fig. 27). Applying lighting to the 3D representations of these parts is a necessary step for effective visualization. However, this step is viewed as an unfortunate annoyance in this domain, particularly considering that the goals are nearly always the same: apply lighting to maximize the perception of shape, detail, and depth from a desired point of view.

A more effective workflow for this domain would allow an expert to select a 3D object to view, select a point of view to view the object from, and then instruct a software system to apply lighting to the object to maximize shape, detail, and depth perception. This is the premise of perceptual inverse lighting research.

Perceptual inverse lighting includes the largest body of literature in inverse lighting (Poulin, Ratib et al. 1997; Shacked 2001; Gumhold 2002; Vázquez and Sbert 2002; Shesh and Chen 2004; Lee, Hao et al. 2006). This work has in common that it operates primarily in the image domain. Rendered image data for a model is analyzed and features are extracted. The 3D geometry data used to generate the image data may
also be analyzed to provide correlating information. This data is then evaluated by a
cost function that is designed to represent how effective a particular image is at
presenting shape, detail, depth, and perhaps other essential visual features. Once
quantified, a new image is generated under new lighting settings and evaluated. This
process is repeated until the best settings for a particular object and user point of view
are found.

(Shacked 2001) was one of the first software systems to automatically light objects
using a perceptual cost function (see Fig. 26). Shacked’s cost function for evaluating
an image consisted of a histogram normalization term and an average image gradient.
The average image gradient represented the mean change of pixel intensity from one
pixel to the next and was meant to be indicative of visible surface detail. The
histogram normalization term was meant to represent how uniform the overall color
distribution in an image was and was also intended to represent the overall visibility
of an object.

![Fig. 26. The 3D model of a ship in this figure is illuminated by the system of (Shacked 2001). This
system’s intent was to maximize perceptual features for shape, detail, and the perception of depth
automatically given a camera point of view specified by a user.](image)

Shacked used the geometry of an object to derive an “edge strength” term. Using
the geometry of an object, Shacked’s system could derive precisely where silhouette
edges should be in an image for a given point of view. A Canny image edge detector
(Canny 1986) was then used to determine where edges visually appeared to be. The
ratio between actual edges and visible edges was used to quantify the quality of shape
depiction for a given light configuration.

(Gumhold 2002; Vázquez and Sbert 2002) used a simpler cost function based solely
on Shannon’s entropy metric (Shannon 1948). Gumhold’s “light entropy” metric was
a spatially independent term that quantified the illumination entropy of an image. In
practice, it provided data similar to Shacked’s histogram normalization term.
Gumhold also conducted a small psychophysical study with 15 subjects. Gumhold
asked each subject to adjust a single light such that they could “perceive the three
dimensional shape of the scene best” (Gumhold 2002) for 5 test scenes. Gumhold then
used the data from this experiment to bias a light entropy metric, although Gumhold reported that participants had different preferences and had difficulty picking a “best” position for the light.

(Vázquez and Sbert 2002) extended Gumhold’s metric by adding a spatial consideration to the function and by operating in the color domain. Both Shacked and Gumhold’s work only analyzed an image after it had been converted to grayscale. In comparison, Vázquez and Sbert used a distance metric on the CIE 1976 color space (also known as the CIELUV color space) (Vázquez and Sbert 2002). Vázquez and Sbert also considered the spatial distribution of pixels when calculating entropy. Instead of calculating entropy as a function of pixel intensities in a spatially independent fashion, Vázquez and Sbert calculated entropy as a function of contiguous areas of color across the image. A contiguous color area was a spatially local region in the image with approximately the same LUV color value.

Fig. 27. This image from (Lee, Hao et al. 2006) illustrates that system’s consideration of cast shadow on an object to help identify depth in a figure. The red box highlights a shadow that the system explicitly manipulated to help identify the structure of this 3D object.

The work of (Shacked 2001; Gumhold 2002; Vázquez and Sbert 2002) all reliably used only a single light source to illuminate an object (Shacked’s system could apparently function with multiple light sources, but this was described as generating potentially confusing results for a user given the assumptions of Shacked’s algorithms). (Lee, Hao et al. 2006) differed in this assumption and in several other ways. In Lee’s work, an object was separated into areas of local curvature based on the geometry of the object. Each area was then illuminated with a separate light source configuration to maximize metrics similar to related work. These metrics were maximized locally and were globally discontinuous such that an object was not illuminated with necessarily similar lighting across its surface. Lee’s work also considered cast shadow (see Fig. 27) and explicitly adjusted lighting to use cast shadow to maximize the depth perception between portions of an object, the first work to do so in this field.
3.1.3 Interactive Inverse Lighting
The field of interactive inverse lighting is the third subfield of inverse lighting. Unlike architectural inverse lighting and perceptual inverse lighting, there is not a single common problem being pursued in this field. Rather, this field is linked by a requirement to produce solutions that operate in real-time for interactive real-time applications such as video games.

Only two prior known systems exist in the field of inverse interactive lighting, The Expressive Lighting Engine (ELE) (Seif El-Nasr 2003; Seif El-Nasr and Horswill 2004) and LightKit (Halle and Meng 2003). ELE was an automated software system designed for adapting lighting to changing context in a virtual interactive environment. LightKit was a software system designed for lighting medical images.

The goal of ELE (see Fig. 28) was automated lighting for unpredictable interactive environments. This goal is similar to the goal of this dissertation. ELE adapted lighting using algorithmic models of theatre and cinematography principles. Lighting for an object was based on its relationship to the environment and on subjective metrics such as visual tension. The object itself was represented as a cylinder. Although ELE maintained appropriate lighting based on changes to an object’s context in an environment, it did not do so in consideration of the object itself. For example, ELE would light a character in a bright white dress the same as a tree stump if she fit within a cylinder of roughly the same size and shape.

![Expressive Lighting Engine](image)

Fig. 28. This figure shows the Expressive Lighting Engine (Seif El-Nasr 2003). This prior work in automated interactive lighting automatically lit objects and scenes to “expressive” lighting goals such as “visual tension”.

In addition to lighting individual objects, ELE could also light the scene surrounding those objects. ELE divided a scene into different areas and then categorized those areas as focus, non-focus, or background. Based on goals specified by a designer, such as directing attention, increasing depth, or evoking mood, ELE would compute a light layout, placing lights in each area to find an optimal solution to the (possibly conflicting) goals.
Finally, ELE considered the colors of lights and their change over time. ELE used a formula to constrain the transition of light colors that allowed a designer to emphasize certain types of continuity. For example, a designer could specify that maintaining the warmth of a color was of highest priority. As a result, whenever ELE needed to change the color of a light (for example, to increase visual tension), ELE would choose intermediary colors that represented the minimal change of warmth possible with respect to the start and end colors.

The goal of LightKit (see Fig. 29) was to simplify the interface for users interactively visualizing a medical model. Essentially, the goal of LightKit was the same as that of perceptual inverse lighting research except LightKit was targeted at applications that required real-time changes to the point of view of a model (all perceptual inverse lighting work is too computationally expensive to operate interactively).

![Fig. 29. This figure shows an object lit by the system of (Halle and Meng 2003). Halle’s system simplified the process of lighting 3D medical models by providing users with a constrained light setup based on three-point lighting.](image)

LightKit used three-point lighting as a constraining lighting technique. As described in Chapter 2, three-point lighting is a lighting technique that divides lighting into three lights: a key, a fill, and a back light. The key light provides a dominant sense of direction, the fill reduces tonal contrast, and the back isolates an object from its surrounding background. In LightKit, the key light was a light that provided a dominant sense of direction and it was always located at 45° relative to the orientation of the camera. The fill light was fixed at 90° relative to the key and was maintained at an intensity ratio of 2:1 to the key. LightKit also used two back lights that illuminated an object from behind and helped to isolate it from the background. In addition to adjusting the point of view, a user could adjust the lighting configuration of LightKit by specifying the intensity of the key light or the color of the key light using a warmth scale derived in (Halle and Meng 2003).
3.2 Physics of Light in Computer Rendering

The physics of light in computer rendering is important to this dissertation as it fundamentally constrains how light is calculated and applied in real-time rendering environments. This section discusses two types of computer visualization: physically accurate rendering as achieved through global illumination and psychophysical rendering using rasterization.

Although the modern physics of light describes light as having a dual nature as particles and waves, computer rendering models consider light as only consisting of particles. Therefore, light phenomenon that is dependent on the wave nature of light cannot be replicated in most computer rendering environments. However, work exists to implement limited wave phenomenon such as diffraction as special case effects (Stam 2004). Because it is limited to special case effects, this chapter and this dissertation assume only the particle nature of light as formalized and expressed through the rendering equation (Green 2003):

\[
\begin{align*}
L(x, \omega_o) &= L_e(x, \omega_o) + \int \frac{f_r(x, \omega_i \rightarrow \omega_o) G(x, x')}{\omega_i} L(x', \omega_i) V(x, x') d\omega_i \\
\end{align*}
\]

Eq. 01. The computer rendering equation.

where:
- \( L(x, \omega_o) \) = the total light intensity from an object at a surface point \( x \) in direction \( \omega_o \).
- \( L_e(x, \omega_o) \) = the emission or light emitted by the object itself at surface point \( x \) in direction \( \omega_o \).
- \( f_r(x, \omega_i \rightarrow \omega_o) \) = the reflectance function at surface point \( x \), which transforms incoming light \( \omega_i \) into outgoing light \( \omega_o \).
- \( L(x', \omega_i) \) = light from \( x' \), a surface point on another object, arriving along \( \omega_i \).
- \( G(x, x') \) = the geometric relationship between \( x \) and a second point \( x' \).
- \( V(x, x') \) = a visibility test; equals 1 if \( x \) can see \( x' \), 0 otherwise.

In theory, the surface point \( x \) is an arbitrary infinitesimally small point chosen at which to evaluate the integral. In practice, it is one of several points that are visible from a camera and are displayed on a computer display to produce a final rendered image. The integral is evaluated across all other surface points \( x' \) that exist in an environment in the half hemisphere space above a surface point as illustrated in Fig. 30. The terms \( \omega_o \) and \( \omega_i \) represent differential angles, usually defined in spherical coordinates on \( \theta \) and \( \lambda \) as shown in Fig. 30. \( \omega_o \) is the angle of radiance\(^5\) from a surface point \( x \). This direction is in practice the direction to a camera that is used to frame a

\(^5\) Radiance is light outgoing from a surface point.
scene. $\omega_i$ is a single infinitesimal irradiance\(^6\) direction formed between the surface point $x$ and a point of evaluation $x'$.

![Diagram](image.png)

Fig. 30. This figure illustrates a light photon interacting with a surface. An incoming photon stream represented as ray $\omega$ to a surface point $x$ is determined by two differential angles $\theta$ and $\lambda$.

The emission term of the rendering equation describes to what extent the current point on the current surface is a light source. It is the amount of radiance a surface point contributes in a given outgoing direction independent of any incoming irradiance. For example, when rendering a light bulb, the emission term would describe the amount of light emitted by the bulb.

The reflectance function or bidirectional reflectance distribution function (BRDF) maps incoming light to outgoing light. It describes how a surface responds to illumination. For example, a diffuse or Lambertian surface (Akenine-Möller and Haines 2002) reflects light equally in all directions and a surface of this type will have the same visual appearance viewed from any point of view. In comparison, a specular surface changes appearance with a change in point of view (see Fig. 31). The outgoing radiance of a specular or semi-specular surface changes as the angle $\omega_o$ changes. In reality, most surfaces fall in between a fully specular and a fully diffuse surface. The range of different surfaces and how they map incoming light to outgoing light is described in the computer rendering equation by a BRDF function.

---

\(^6\) Irradiance is light incident to a surface point.
Fig. 31. This figure illustrates the difference between a diffuse and a specular or shiny surface. The diffuse surface represented on the left reflects incoming light equally in all directions. In contrast, a specular surface only reflects light in a small wedge of directions and a perfectly specular surface reflects light in only one direction.

The geometric term represents the spatial relationship between the two surface points \( x' \) and \( x \). These two points are any arbitrary surface points in a rendering environment under consideration at any given time. Theoretically these points are infinitesimally small but because the rendering equation cannot be computed analytically, they are often represented by a surface patch with an area that is defined as needed by a particular implementation of the rendering equation. The geometric term encodes the effects of participating media\(^7\). Participating media can include smoke, fog, or semi-transparent surfaces between \( x' \) and \( x \). The geometric term also encodes attenuation of light due to distance. In the photon model of light, light attenuates (its intensity decreases) as fewer photons reach a point. This occurs as the distance from a source of light increases because the photon streams generally diverge from their source (Fig. 32). Although this is the physical description of the phenomenon, the implementation of the effect of attenuation is often approximated. In many cases, it is described using a simple inverse square rule, which is physically accurate when a surface receiving light is sufficiently distant from the source of illumination.

\(^7\) Participating media consists of small particles that absorb, emit, or scatter light. An example of participating media is fog.
Finally, the visibility term encodes occlusion in the environment. Occlusion occurs when light photons cannot reach the surface point $x$ from a surface point $x'$ due to a blocker of some sort positioned between the two. This prevents any light contribution to the surface point $x$ from the surface point $x'$. Visibility is generally defined as taking either the value 0 or 1 with no “partial” visibility (translucency as is caused by partial visibility is generally encoded in the geometric term as participating media).

There is no known method with which to evaluate the rendering equation analytically. Therefore, it must be solved numerically using various approximations. Three schools of thought categorize the various approximations for solving the computer rendering equation. Some systems aim to be as physically accurate as possible. These systems focus completely on exact calculation of the computer rendering equation. Other systems consider the psychophysics of light and how the human visual system perceives illumination. These systems make approximations, some physically inaccurate, based on how we interpret illumination. In exchange, these systems are usually less computationally expensive than physically accurate solutions or are more controllable. For example, a psychophysical solution may represent cast shadows as explicit first-order objects that can be manipulated independently from the objects that cast them. While this will produce physically inaccurate results, it is also a desirable level of control for lighting designers.

![Fig. 32. An illustration of why light intensity attenuates with distance. As photon streams diverge from the light source, less of the streams intersect a surface patch, resulting in fewer photons intersecting the surface per unit time and lower resulting light intensity.](image-url)
Finally, some systems are completely based in an artistic interpretation of light, such as non-photorealistic rendering (see Fig. 33). While this dissertation considers physically accurate and psychophysical solutions to the computer rendering equation, non-photorealistic rendering is considered outside the domain of this work. A non-photorealistic rendering solution is completely designed based on artistic motivations for a specific application and very little can be assumed about the constraints of this type of solution. The goal of this dissertation is to develop algorithms that are as general as possible so that they can be potentially used in a variety of applications. Consideration of non-photorealistic rendering would require picking a specific application and a specific non-photorealistic rendering solution. Therefore, the rest of this dissertation discusses physically accurate rendering and psychophysical rendering only.

3.2.1 Physically Accurate Rendering
Physically accurate rendering is generally known as global illumination. Individual global illumination algorithms can differ greatly but they do share common properties. All global illumination algorithms mean to solve the computer rendering equation with exact physical accuracy within the limits of the algorithm. In practice this means that they may specifically exclude portions of the rendering equation (such as participating media) or constrain it (for example, by limiting the BRDF function to a subset of all possible BRDF functions) but they otherwise produce physically accurate results. Further, global illumination algorithms do not have “special cases”. For example, the visibility term of the computer rendering equation will be calculated the same way for all objects under all circumstances in a global illumination algorithm. This usually results in global illumination algorithms having much simpler
mathematical forms and implementations than the psychophysical approaches described in the next section because one algorithm or set of algorithms is evaluated holistically in all situations.

The two key global illumination techniques are radiosity and ray tracing. These techniques differ in how they simplify and constrain the rendering equation and they have very different strengths and weaknesses.

Radiosity constrains the computer rendering equation by assuming that all surfaces have a Lambertian BRDF function (all surfaces reflect light equally across a half hemisphere above the surface as illustrated in Fig. 31). This constraint is usually reasonable as most objects are diffuse to some extent but it means that notably specular surfaces (such as glass, water, marble, and polished metals) are incompatible with radiosity. Participating media is also not considered, so the geometric term is only considered with regards to light attenuation. These assumptions allow radiosity to calculate lighting that is independent from the eye position of a viewer because the radiance of a surface is the same for any outgoing direction $\omega_o$.

Radiosity segments an environment into patches. These patches are radiosity’s approximation to the infinitesimally small surface points used in the rendering equation $x$ and $x'$. The final appearance of each patch is found by calculating the reflection of light between all patches in an environment (see Fig. 34).

![Fig. 34. This figure shows how radiosity transfers light throughout an environment. An initial source of light emits photon streams into an environment. Surfaces in the environment reflect these streams back and onto other surfaces. Each surface is assumed to be Lambertian.](image)

Radiosity has several advantages. Illumination in radiosity is view independent which means it can be calculated and stored. Many implementations of “baked” lighting for real-time rendering as described in Chapter 1 use radiosity because of this trait. Radiosity also captures indirect reflections and any visual phenomena that results from indirect reflections, such as light “bleed” (see Fig. 35). The key disadvantage of radiosity is its inability to handle specifically view dependent effects such as specular highlights. Further, because radiosity “patches” an environment it is
often impractical to represent high frequency lighting effects (such as sharp edges of cast shadow). While these effects are theoretically possible, the patch size must be very small and this is often beyond the computational limits of a radiosity implementation or it requires more complex, adaptive patching that adjusts the patch size to a smaller size when high frequency effects need to be accommodated. The limitations of radiosity are essentially the strengths of ray tracing.

![Image](http://www.graphics.cornell.edu/online/box/)

**Fig. 35.** This image of the Cornell Box from http://www.graphics.cornell.edu/online/box/ illustrates the lighting phenomenon of light “bleed”. The indicated region shows that indirect reflections off of a colored surface cause some of the hue and saturation of that surface to interact and reflect off another surface, giving that surface a similar hue and saturation.

Ray tracing focuses on the illumination space from the eye’s point of view. Photon streams are approximated as mathematical rays and usually traced backwards, from the eye to the light source instead of the natural flow of photons, which is from the light source to the eye. Each ray’s interaction with surfaces and light sources is then computed to determine the scene appearance from the eye’s point of view (see Fig. 36).
This approach effectively allows ray tracing to “patch” an environment with respect to the resolution of the final rendered image. As a result, ray tracing does not have a problem with high frequency effects like radiosity does and can create sharp shadows. Further, ray tracing can easily handle specular highlights, participating media, and any other view dependent effects. In fact, a common selling point of ray tracing in the rendering industry is its effectiveness in creating refraction, reflection, and view dependent effects.

Conversely, ray tracing has a difficult time with indirect illumination. Unlike radiosity, which inherently calculates indirect illumination by calculating all light transport in an environment, ray tracing must emit an ever increasing number of rays to calculate indirect illumination. This number grows exponentially and it quickly becomes too expensive to compute. Further, the results of ray tracing are inherently invalidated whenever the camera moves and cannot be stored like radiosity, although various forms of caching intermediate values are possible (Dietrich, Colditz et al. 2005). Finally, ray tracing limits the BRDF function of the computer rendering equation to symmetric functions which are BRDFs that have an inverse. This is required to trace rays in the reverse direction of actual light transport.

Often, professional rendering software will use both the techniques of ray tracing and radiosity simultaneously to produce the final rendered image or will use more advanced versions of the basic algorithms of these techniques. For example, see (Kajiya 1986) for an enhancement on ray tracing known as path tracing. The choice between an advanced algorithm and combining simpler forms of multiple algorithms is usually determined by computational cost. Combining radiosity and ray tracing is usually less expensive than using path tracing, for example.

The fundamental limitation of all global illumination techniques is speed. A naive implementation of these techniques can grow with $O(n^2)$ or even $O(C^n)$ complexity, where $n$ is the number of objects to render. Typically, even highly optimized implementations that take advantage of acceleration hardware cannot operate in real-time (Gebhardt 2006). In the target applications of this dissertation (real-time rendering), global illumination is typically used for “baked” lighting. Baked lighting pre-calculates lighting between static objects and then stores the final results of that
lighting, which are then applied at runtime. However, most dynamic lighting in real-time rendering is done using psychophysical techniques.

### 3.2.2 Psychophysical Rendering

Psychophysical rendering techniques exist primarily for computational efficiency. They are psychophysical in the sense that they use approximations that are not necessarily physically accurate but are motivated by what is computationally efficient yet visually acceptable. For example, Fig. 37 shows three distinct shadow regions cast from three characters by a single light source, which is physically incorrect but generally not noticeable to a human observer (the physically correct result would be a single shadow region with no noticeable interior edges that is the union of the individual shadows cast from each character).

![Fig. 37](image)

Fig. 37. Although all cast from the same off screen light source (the Sun), three clearly distinct shadow outlines are visible within the red box shown in this figure. This is an artifact of the form of projected shadowing as used in this image from the video game *Guild Wars* (released in 2005 by ArenaNet).

Most psychophysical rendering uses rasterization. In rasterization, objects are typically composed of triangles. Each triangle is rendered by projecting the corners of the triangle onto the camera near plane (the computer display, see Fig. 38) and then filling in the pixels between the corners to produce a rendered image. While global illumination rendering techniques such as ray tracing inherently calculate lighting in the process of drawing an object, lighting in rasterization is a special case. It is evaluated while the triangle is being drawn. For example, a pixel may calculate the distance between itself and a light source and then calculate the illumination contribution from that light source based on a simple model of distance attenuation. This process of calculating the lighting while drawing the final pixel is referred to as local lighting.
While global illumination is holistic, local lighting can and usually does consist of many special cases. For example, while some pixels may include a visibility term to account for shadows cast between multiple objects, others may not. Or, while some pixels may be shadowed by not applying lighting in the first place, other pixels may be lit and then darkened later. This later form of shadowing is called projected shadowing and can result in the artifacts shown in Fig. 37.

In the real world, any object is a source of illumination. In rasterization, light is typically only emitted by special light source objects although regular objects can have “emission”. Unlike the emission term of the rendering equation, emission in rasterization serves only to make an object visible in the absence of light sources. Emitted light does not affect other objects (see Fig. 39).

The specialized light sources of rasterization include point lights, directional lights, and spot lights. Each is illustrated in Fig. 40. Although newer technology allows for a wider variety of light source types in rasterization (Akenine-Möller and Haines 2002) such as a hemispherical light, all light sources in rasterization are chosen to have
simple mathematical properties. This makes their contribution to a scene less computationally expensive to calculate than if lights with arbitrary shapes were used.

![Fig. 40](image)

**Fig. 40.** This figure illustrates the three types of light sources typical of rasterization. The light on the left is a point light which emits light in all directions equally. The light in the middle is a directional light which emits parallel rays of light from an infinitesimal planar surface. The light on the right is a spot light which is similar to a point light but is restricted to a cone of influence.

Rasterization is direct illumination only. There is no indirect reflection (see Fig. 41) in direct illumination. The effects of indirect reflection are often approximated in rasterization with a wide variety of approximations of ambient illumination. This can be a simple constant light intensity applied to all surfaces equally, a hemispherical light (Snyder 1996), an environment map (Chen and Liu 2008), or a complex encoding of the irradiance environment using spherical harmonics (Green 2003). While the techniques for accounting for indirect illumination are complex and diverse, most are static (dynamic objects do not contribute to indirectly reflected light).

![Fig. 41](image)

**Fig. 41.** This figure shows a scene from the video game *Half-Life 2* (released in 2004 by Valve). *Half-Life 2* uses radiosity lighting to pre-compute static lighting with indirect illumination. The left image is the same scene as the right, except the right image includes indirect or “bounced” illumination while the left image does not. This illustrates that indirect illumination can contribute greatly to the overall lighting of a scene.

The BRDF of the computer rendering equation has many diverse approximations in rasterization. The simplest and most common is the texture map. A texture map is an image that is mapped to an object’s surface (see Fig. 42). It effectively gives an object the illusion of surface texture and detail. While rasterizing a triangle, the texture image is interpolated across the surface of the triangle and a color from the image is found for each rendered pixel. This color is then either output directly to the screen or is further used by the variety of lighting calculations that may occur for each rendered pixel.
Fig. 42. This figure illustrates the computer rendering technique of texture mapping. An image shown on the left is mapped to the surface of a geometrical sphere providing the sphere with the texture and detail of the texture map.

Fig. 43. This figure from http://www.punkuser.nte/vsm/ illustrates a real-time shadowing technique which allows shadows to have soft edges as indicated by the red box.

The visibility term of the computer rendering equation is handled as a first order object in rasterization. It is not an inherent part of lighting calculation as it is in global illumination. In some cases, a shadowing term is determined for each rendered pixel and then used to determine whether a shadow casting light source affects each pixel or not. In other cases, screen pixels are rendered normally and a darkening term is applied afterwards, as shown in Fig. 37. Shadowing is also divided into several
special cases: soft shadow edges (Fig. 43), cast shadow\(^8\), and self-shadowing\(^9\) (Fig. 44). For example, soft shadowing may be achieved by blurring the shadow in a completely artificial but visually believable way (the edges of the shadows shown in Fig. 43 are made to appear soft using this technique).

![Fig. 44](image.png)

Fig. 44. This image from the video game *Gears of War* (released in 2006 by Epic Games) illustrates an object casting shadow on itself as identified by the red box. This case is a difficult rendering effect to achieve in rasterization and is usually handled as a special case or not at all.

Finally, the geometric term of the rendering equation is approximated in rasterization as specific special case effects. For example, fog can be approximated by altering the final intensity of a point on a rendered surface based on its distance from the viewer (Akenine-Möller and Haines 2002).

Rasterization also uses a number of special case approximations that recreate several parts of the rendering equation at once. Ambient occlusion approximates the self-shadowing in cavities of an object. It calculates a darkening term based on the relationship of a point on an object’s surface to other nearby surfaces that occlude the point from illumination (Filion and McNaughton 2008). Ambient occlusion is essentially a rough approximation of the visibility term of the rendering equation under conditions where light can be assumed to be arriving at a point from roughly all directions at once. This assumption is true when occluding surfaces are nearby to a point for which ambient occlusion is calculated and for lighting environments that have an approximation of indirect illumination. For example, when indirect illumination is approximated as a constant color applied to every point in an

---

\(^8\) Cast shadow is a shadow cast from one object to another.

\(^9\) Self-shadowing is shadow cast from an object onto itself. Physically, there is no difference between cast shadow and self-shadowing and there is also usually no distinction between the two in global illumination algorithms. However, in rasterization, a distinction is usually made between the two due to implementation challenges. Roughly, because self-shadowing surfaces are usually much closer spatially than cast shadowing surfaces, they must be handled more carefully to avoid visual artifacts.
environment, ambient occlusion can be effective at modulating this color to create a more realistic appearance.

Image based lighting is another special case approximation. Image based lighting applies lighting to a point in space as captured from an environment. The captured lighting is often captured from the real world using a mirror ball (Fig. 45). Image based lighting stores the irradiance arriving from all directions at a point in a space and then applies it to a point in a virtual environment. It is effectively accurate when the sources of light are all distant from the point of capture.

Fig. 45. This is an image of a mirror ball from http://www.dbki.de/, a real world spherical mirror surface that represents a 360° illumination capture of the environment around the ball. It is often used for a technique called image based lighting where the image of the ball is turned “inside out” and mapped to the inside of a spherical surface which is then used to apply lighting to an environment as if it originated from the sphere.

3.2.3 Summary
Psychophysical rendering is generally implemented using rasterization and consists of many special case solutions to the rendering equation vs. a single holistic solution as in global illumination techniques. The benefit of psychophysical rendering techniques in comparison to global illumination algorithms is computational cost. They take less time to calculate and are therefore more compatible with real-time rendering. Special hardware that supports rasterization is also widely available, increasing the speed benefits of rasterization even further.

The drawback of most psychophysical approaches to rendering is algorithmic complexity. While a ray tracer can be implemented in a hundred lines of C++ code (Consultancy 2007), modern real-time rasterization engines may include thousands of lines of code developed with a variety of software engineering paradigms (for example, functional programming and concurrent programming).
Chapter 4: System for Automated Interactive Lighting (SAIL)

This chapter describes SAIL, the automated lighting system of this dissertation. This chapter discusses what SAIL does at a high-level, considerations of theory and related work that went into SAIL’s design, and the current implementation of SAIL. Details of SAIL’s algorithms are presented in Chapter 5. An overview of SAIL’s process at a high-level is shown in Fig. 46.

4.1 Considerations of Theory and Lighting Practice

Based on the importance of light contrast in lighting design (see Chapter 2), one of the two priorities for SAIL is the control of light contrast. SAIL controls overall contrast
by controlling how lights affect objects in a scene. Simulating the process of film exposure, which is a fundamental property of light contrast in photographic film, is outside the scope of this dissertation and is already addressed in research dealing with a technique called tone mapping as part of high dynamic range (HDR) rendering (Reinhard, Stark et al. 2002). Integrating SAIL with an HDR rendering environment is left as future work and SAIL does not explicitly consider the presence of HDR rendering in its current implementation.

SAIL’s second priority is the control of light direction. This is based on the importance of light direction for controlling visual cues such as depth (see Chapter 2). To control direction and also light contrast, SAIL uses the lighting technique of three-point lighting in a manner similar to the Expressive Lighting Engine (ELE) (Seif El-Nasr 2003). As described in Chapter 2, three-point lighting consists of a key light to control light direction, a fill light to control light contrast, and a back light to control the perception of an object’s silhouette. SAIL’s version of three-point lighting consists of a key light and a fill light; it does not use a back light. This is explained further in Chapter 5.

SAIL also considers color constancy (see Chapter 2). The specific mathematical model of color constancy used by SAIL is retinex (Land and McCann 1971). SAIL uses single-scale retinex (SSR) (Jobson and Woodell 1995) in its image processing to isolate light information in an image. Retinex was chosen because it is the oldest color constancy model and well vetted. Its assumptions are also mathematically supported by recent work (Ramamoorthi 2002). SSR was chosen in comparison to alternatives (such as multi-scale retinex) (Jobson and Woodell 1995) because SSR is computationally inexpensive to apply and the features of more developed approaches such as multi-scale retinex are not necessary in the context of this dissertation. Improvements to SSR were intended to improve the accuracy of image segmentation. This was motivated by the typical usage of retinex which is to remove lighting information from an image. In the context of SAIL however, lighting information is extracted, not removed, and this lighting information is summarized into a small set of terms for the entire image. As a result, precise segmentation of regions in an image is unnecessary as SAIL considers the image as one global source of information.

SAIL uses rasterization to illuminate an object at runtime because real-time rendering is currently not practical with global illumination algorithms such as radiosity or ray tracing (see Chapter 3). SAIL uses the typical light sources of rasterization (point, directional, and spot lights as shown in Fig. 40). It does not explicitly consider the visibility term of the rendering equation (see Eq. 01) because the availability of shadowing in implementations of rasterization is inconsistent and composed of many special case effects. However, if a particular rasterization engine includes shadowing, SAIL’s image analysis will implicitly consider this shadowing.

4.2 Components of SAIL

SAIL consists of two main components. The first component is image analysis which SAIL uses to understand light features in an image. The image shown in Fig. 46 is of a shaded circle but an image can be of anything with shading and color contrast
information. This component is similar to perceptual inverse lighting work described in Chapter 3. Unlike this related work, SAIL must operate in real-time to support interactive applications such as video games. To accomplish this, SAIL samples the space of possible light configurations for an object in a preprocess step and then stores the result of this sampling as a surface of points. This surface is then used at runtime to make decisions when adapting lighting, avoiding the computationally prohibitive step of rendering and analyzing images at runtime.

The second component of SAIL is object understanding used to achieve adaptive runtime lighting. This component is very similar to the object lighting component of ELE discussed in Chapter 3 with the addition of image analysis to understand how an object responds to changes in illumination. This component uses metrics extracted from a goal image and the position of light sources in the environment. It then lights an object, using the three-point lighting technique, to achieve an appearance between how the object “should” look based on the image and how it would look naturally based on the light sources. This process is illustrated in Fig. 47.

![Fig. 47. This figure illustrates SAIL. The shaded circle is the goal image. The left image is the man naturally lit. The center image is the man lit to achieve the goal image. The right image is the man lit as a compromise between the two.](image)

### 4.3 SAIL’s Relationship to Related Work

SAIL’s design was motivated by related work discussed in Chapter 3. Related work in architectural inverse lighting is effectively separate from the work of SAIL. Architectural inverse lighting solutions are designed to eliminate the need for iteration when designing lighting for architectural spaces. These solutions are meant to be used in visualization tools that model a real-world architectural space accurately and do not typically render in real-time. As a result, architectural inverse lighting solutions are too computationally expensive to be applied to a real-time problem. Also, the
constraints of interactive real-time environments are very different from the constraints of real world spaces. Lighting in real world spaces is constrained by real world physics, financial cost, functionality, and aesthetics. Lighting in virtual environments does not necessarily obey real world physics and financial cost of lighting is not a concern, but computational efficiency of lighting algorithms is of paramount concern.

The field of perceptual inverse lighting also consists of solutions that are too computationally expensive for real-time rendering. The goals of perceptual inverse lighting are also very different from the goals of SAIL. Perceptual inverse lighting solutions are designed to automatically illuminate a virtual object to a perceptual ideal using a cost function. SAIL is designed to automatically light an object based on a designer’s intent and the context of the object as it changes over time.

Although the solutions of perceptual inverse lighting are intended for different goals and are too expensive for real-time rendering, perceptual inverse lighting software operates on images in a manner similar to SAIL. As a result, some of the algorithms of perceptual inverse lighting are similar to those used by SAIL. In particular, SAIL uses the light entropy metric of (Gumhold 2002) which is discussed further in Chapter 5. SAIL’s consideration of light direction is also based on the usage of shading gradients in (Shacked 2001) to indicate direction of illumination.

The field of interactive inverse lighting is the direct subfield of SAIL. This field consists of two prior lighting systems, LightKit (Halle and Meng 2003) and ELE. LightKit was designed to achieve many of the same goals as perceptual inverse lighting research but it also needed to operate in real-time. LightKit was designed for non-experts that must design light as an undesirable necessity rather than an integral part of their process (for example, a physician might need to illuminate a 3D medical image while interactively visualizing it). As a result, LightKit was intended for contexts in which light is actively and deliberately controlled by a user rather than in contexts where light responds to a user interacting with other elements of an environment. The different assumptions and goals of LightKit make this work effectively separate from the work of SAIL.

SAIL’s design was most impacted by ELE. ELE was a software lighting system with goals similar to SAIL. For example, lighting on objects was controlled using three-point lighting. ELE also adapted lighting to changes in context while attempting to maintain a goal specified by a lighting designer. However, ELE had several shortcomings that SAIL is designed to address. In particular, ELE did not consider the appearance of an object. ELE would light an object of the same size and approximate shape, whether it was a tree stump or a woman in a bright white dress. ELE also had an interface that was difficult for lighting experts to use (see Fig. 48). SAIL’s usage of images to specify a lighting goal is based on research (Zupko 2008) intended to find a better interface for ELE.
4.4 Current Implementation of SAIL

To implement SAIL, a custom real-time rasterization engine was built on top of the Microsoft XNA Framework 2.0 (Microsoft 2008). This engine supports:

1. Dynamic, per-pixel lighting and dynamic shadow map shadows cast from spot lights.
2. Lighting, environments, characters (including animation), and textures imported using the COLLADA 3D content exchange format (Group 2008).
3. Occlusion culling to avoid rasterizing invisible objects to allow for more complex environments.
4. SAIL implemented as a software library and integrated to light a target object, such as a character.

Feature 1 allows the engine to apply a consistent lighting model to all objects. Even though the environment surrounding an automatically lit object is illuminated manually, all objects are lit using the same light primitives and equations. Any visual difference between an object lit by SAIL and the environment should therefore be a factor of how SAIL is controlling the light and not a difference in lighting models.

Feature 2 allows for the import and display of complex environments and characters. COLLADA is a standardized 3D content exchange format and is now supported by several 3D modeling packages, such as Autodesk Maya or Autodesk 3ds Max.

Feature 3 allows for complex environments. An environment with geometric complexity similar to 3-5 year old video games was created in Autodesk Maya by
David Milam (Milam 2008) and used for the study described in Chapter 6. A screenshot of the current implementation of SAIL is shown in Fig. 49.

Feature 4 allows SAIL to be used in engines other than the rasterization engine described here. This is a practical consideration that should help facilitate future research and integration of SAIL.

4.4.1 Lighting, Shading, and Shadows
The lighting and shading of the rasterization engine is implemented using programmable pixel and vertex shaders. A shader is a relatively new feature of rasterization hardware which allows for portions of the rasterization pipeline to be programmed and new effects to be created, such as new lighting, shading, and material models. The manner in which a pixel is drawn as described in Chapter 3 can be controlled explicitly by a programmer.

The rasterizer implements three types of lights in shaders: point, directional, and spot lights. These follow their definitions as described in Chapter 3 (see Fig. 40). A point light is an infinitesimally small light source that emits light equally in all directions. A directional light is an infinitesimal plane with no position that emits light in parallel rays in a single direction. A spot light is effectively a point light that is limited to a cone of emission instead of emitting light in all directions. The shader code in High Level Shader Language (Microsoft 2008) for the materials and lights of the rasterizer is shown in Appendix B.

Shadows in the rasterizer are implemented using shadow maps. Shadow maps are textures that contain the visibility information for a shadow casting light source. These textures are created by rendering the scene from the light’s point of view and storing the depth of rendered pixels from this point of view. While rendering an object affected by a light source, the pixels of the object are transformed into the coordinate frame of the light and then compared against the pixels in the light’s shadow map. If a
pixel in the shadow map has a depth value that places it in front of the pixel of the rendered object, then the corresponding pixel of the object is in shadow and does not receive illumination from the light. Otherwise, it is illuminated by the light.

4.4.2 COLLADA Materials
Materials describe how a surface of an object responds to lighting in the rasterizer. Similarly to the lighting itself, these are described using programmable shaders. Materials in the rasterizer completely implement the <constant>, <lambert>, <phong>, and <blinn> materials as described in the COLLADA 1.4.1 standard (Group 2008).

Briefly, <constant> materials contain only emission and ambient colors. Emission colors represent color that is emitted by the surface independent of lighting and ambient color is color that is emitted by the surface but scaled by a global ambient factor. <constant> material surfaces do not respond to lighting at all. <lambert> materials respond to diffuse lighting but are otherwise identical to constant materials. <phong> materials respond to specular lighting using the Phong (Phong 1973) lighting model but are otherwise identical to <lambert> materials. Similarly, <blinn> materials respond to specular lighting using the Blinn-Phong (Blinn 1977) lighting model but are otherwise identical to <lambert> materials. Each channel of a material (emission, diffuse, specular, ambient, etc.) is defined as either a constant color or as a texture map in the rasterizer. See Appendix B for the shader code that defines materials in the engine.

4.4.3 Occlusion
To handle complex environments, the rasterizer uses three levels of visible surface determination to detect occlusion. First, objects outside the view frustum are removed. The view frustum is a six sided region formed by the camera that determines the area within which objects are visible. Second, portal culling is used. Briefly, portal culling divides an environment into convex regions separated by portals, which are planar windows. The assumption of portal culling is that each convex region can only see another convex region through a portal. With this assumption, entire convex regions can be quickly identified as not visible if no portal into that region is visible. See (Lengyel 2004) for a more in depth treatment of portal culling.

Finally, hardware occlusion queries (Wimmer and Bittner 2005) are used to further remove objects that are visible through a portal but are occluded by other objects. Hardware occlusion queries are a feature of modern accelerated rasterization hardware. By rasterizing simplified versions of an object with hardware occlusion queries enabled, the hardware can be queried to determine if the simplified object was visible or not. If this simplified version was not visible, the more expensive version does not need to be rendered.
Chapter 5: Algorithms of SAIL

This chapter presents the algorithms and technical details of SAIL. SAIL’s algorithms make up two components. The first component is image analysis used to understand light features present in an image. The second component is runtime adaptation, used to apply lighting at runtime in consideration of context.

5.1 Image Analysis

SAIL extracts two types of information from an image: 1) light direction, and 2) light contrast. Light direction roughly correlates to the direction of key light and light contrast roughly correlates to the intensity of fill light, but both are a combination of light direction and intensity and their effect on the object present in an image.

5.1.1 Direction

An image is first processed using the single-scale retinex (SSR) algorithm of (Jobson and Woodell 1995) to produce an illumination image based on the retinex model (Land and McCann 1971) of color constancy. An illumination image is a filtered version of an image that does not contain high frequency pixel intensity changes present in the original image. This is based on the assumption that lighting (without consideration of visibility) contributes only low frequency information to an image while high frequency information is due to surface reflectance. The original form of SSR is arranged to produce an illumination independent reflectance image:

\[
R(x, y) = \log L_i(x, y) - \log[F(x, y) * L_i(x, y)]
\]

Eq. 02. Single-scale retinex.

where \( i \) denotes the color channel, \( L_i(x, y) \) is the \( i \)th channel of the source image, \(*\) is the convolution operator, and \( F(x, y) \) is a Gaussian kernel used to filter the source image. For the purposes of this dissertation the illumination image is the goal term, which is:

\[
I_i(x, y) = F(x, y) * L_i(x, y)
\]

Eq. 03. Relationship of an illumination image to an input lighting design image.

where \( I_i(x, y) \) denotes the \( i \)th channel of the illumination image.

The filtered light image is then transformed into a grayscale image using the \( Y \) value from the sRGB derived XYZ color space as in (Gumhold 2002):
Eq. 04. The Y component of the XYZ color space from the sRGB color space.

\[ Y = 0.2126R' + 0.7152G' + 0.722B' \]

where Y is the Y term of the XYZ color space conversion from the sRGB color space and R', G', B' are defined by the function:

\[ C' = \begin{cases} 
\left( \frac{C}{1.055} \right)^{2.4} & \text{if } C > 0.04045 \\
\frac{C}{12.92} & \text{else}
\end{cases} \]

Eq. 05. A component C' of the sRGB color space from the RGB color space.

with C substituted for either R, G, or B.

Once the grayscale light image has been generated, two metrics for light direction are calculated. These roughly correspond to angles defined as shown in Fig. 50 but they are actually ratios of pixel intensity calculated in local regions and averaged across the image.

Fig. 50. This image illustrates the definition of image direction metrics \( \theta \) and \( \beta \).

To find \( \theta \) as indicated in Fig. 50, local 3x3 windows of pixels are analyzed across the image, similarly to the application of a convolution kernel. For each window, the center of mass of pixel intensity is found relative to the window center. This relative position is then converted to a normalized 2D vector. The average 2D vector for the entire image is converted to an angle to find \( \theta \).

Once \( \theta \) has been found, the term \( \beta \) as indicated in Fig. 50 is calculated by finding the average center of mass of pixel intensities along the axis \( x' \) as indicated in Fig. 50, specifically:
where $i$ is a line out of $n$ adjacent lines spread across all pixels in an image parallel with $x'$ and perpendicular with $y'$, $m_i$ is the pixel count of line $i$, and $Y_{ij}$ is the $Y$ value of the $j$th pixel of line $i$.

5.1.2 Contrast

Light entropy is used along with the term $\beta$ to represent light contrast in an image. Two light entropy metrics have been defined in related work (see Chapter 3) in (Gumhold 2002) and later in (Vázquez and Sbert 2002). SAIL uses the light entropy metric of (Gumhold 2002) as: 1) it is defined in the grayscale domain, and 2) it is spatially independent. Spatial independence is important as the light entropy metric should contribute information that is roughly independent from the direction metrics calculated in the previous section. The light entropy of (Gumhold 2002) is:

$$H = \sum_{i=1}^{m} p_i \log \frac{1}{p_i}$$

Eq. 07. Calculation of light entropy from (Gumhold 2002).

where $m$ is a number of bins used to segment the $Y$ value of a pixel, and $p_i$ is the probability of a pixel falling into the bin $i$. $p_i$ is defined as:

$$p_i = \frac{c_i}{n}$$

Eq. 08. Probability of a pixel falling into bin $i$.

where $c_i$ is the number of pixels falling into a bin $i$ and $n$ is the total number of pixels in the image. The $i$ of a pixel is defined as:

$$i = \left\lfloor m \left(Y' + \frac{1}{2}\right) \right\rfloor$$

Eq. 09. Selection of the bin for a pixel during the calculation of light entropy.

where $Y'$ is the $Y$ value for a given pixel normalized to the range $[0, 1]$ and $m$ is a number of bins used to segment the $Y'$ value of a pixel. The bin count is set to $m = 30$ bins as in (Gumhold 2002).
5.2 Modeling an Object’s Response to Light

To adapt lighting at runtime, SAIL needs to understand how an object appears under different lighting conditions. Calculating this information directly from images of an object would be too slow to perform at runtime, so SAIL generates a model of how an object responds to light in a preprocess step.

The model is generated by jitter sampling (see Fig. 51) the space of control parameters of SAIL’s runtime lighting model. This model consists of 5 control parameters: camera direction (specified as pitch and yaw), key direction (specified as yaw and roll), and key-to-fill ratio. For each randomly sampled control parameter set, the object is rendered. This rendered image is analyzed by the image analysis component and a set of image metrics (consisting of shading direction and contrast indicators as described in Sections 5.1.1 and 5.1.2) is produced. The surface of all randomly generated image metric sets is stored and used at runtime as a mathematical model of how an object’s appearance changes in response to a change in illumination.

Fig. 51. This figure demonstrates jitter sampling, a sampling scheme that divides a space into an even grid and randomly samples an equal number of points from each grid. This approach requires fewer samples than a straight random sampling approach.

Naïve storage of the surface of randomly sampled points requires 824 KB (kilobytes) of memory. This is probably an unreasonable size for most real-time visualization applications such as video games. Details of this calculation and solutions to reduce storage requirements are described in Appendix A.

5.3 Runtime Lighting

Lighting is applied at runtime using a constrained set of lights based on the three-point lighting technique discussed in Chapter 2. Three-point lighting consists of a key, a fill, and a back light. The key light provides dominant direction cues, the fill light
reduces tonal contrast, and the back light rims the object, isolating it from the background. SAIL only uses a key and a fill light, as the effects of a backlight are high frequency and effectively removed in the Gaussian filtering step of image analysis. Capturing and applying the effect of a back light is left for future work.

The key light is a point light that is limited to only affecting the target object. The fill light is a directional light that is similarly limited. The control parameters for this configuration are a yaw and roll for the key direction and a key-to-fill ratio for the fill intensity. The fill direction is fixed at 90° relative to the key light. The desired key intensity is always 1.0. The actual key intensity at runtime is an average between the intensity as sampled from an object’s environment and a value of 1.0. SAIL understands the appearance of an object based on the jitter sampled surface derived as described in Section 5.2. It understands a goal image through image analysis. It derives the appearance of an object under natural lighting by evaluating the jitter sampled surface at the coordinates of the effective key and fill settings as derived from natural light source positions.

Lighting adaptation at runtime is achieved by applying stochastic gradient descent (see Fig. 52) to move towards the point on the jitter sampled surface that is closest to the goal image and to the point that is closest to the natural lighting state. Closeness is measured using a least squares error metric between appearance metrics. The behavior of stochastic gradient descent leads to temporally consistent transitions, where SAIL approaches the best solution by proceeding through locally good solutions.

Fig. 52. This figure illustrates the method of stochastic gradient descent. The solution for a function begins at some initial value \( x_0 \). This is the appearance of an object under the illumination of its environment. The value is then stepped towards the best value \( x_5 \) based on the slope of the error between each point \( x_0, x_1, \ldots, x_5 \). The best value is the desired appearance. Because gradient descent is a hill climbing algorithm, it effectively walks along an error surface representing a gradual decrease in error. Because of this, the interim solutions \( x_0, x_1, \ldots, x_4 \) are still reasonable solutions which allows SAIL to maintain visual continuity as it approaches the best solution.
Chapter 6: Evaluation of SAIL - Study Design

This chapter describes the design of a qualitative study that was conducted to evaluate SAIL’s contributions as an artifact. Specifically, SAIL’s usage of images as interface for lighting goal specification and its reflection of a natural lighting environment while applying a lighting goal at runtime are discussed. Participants in this study interacted with an implementation of SAIL and were then interviewed using a semi-structured approach. After interviews, participants were also given an opportunity to ask questions and engage in conversation with an investigator in a freeform manner. The results of this study are presented in Chapter 7.

6.1 Development of SAIL’s Implementation

Several versions of SAIL were developed leading up to the current version used for the study. This section overviews these previous versions to explain the motivation for some of the design decisions of the final version.

Version 1 was used for presentation at this dissertation’s proposal and is shown in Fig. 53. This version was criticized by the committee of this dissertation for being too dark and abstract. It was difficult to understand the context without clear logical sources of illumination.

![Fig. 53. This screenshot is from version 1 of SAIL which was used for demonstration at this dissertation’s proposal. The character is from the video game Max Payne (released in 2001 by Rockstar Games).](image)

Version 2 shown in Fig. 54 was created to address this. A physical light source was added and the overall intensity of lighting increased. The environment was also simplified and designed to imply a black box theatre or other similar setting. This
version was never presented but was casually evaluated by lighting design acquaintances of the author. Feedback from this evaluation was used to motivate the design of the final implementation version of SAIL. Lighting designers commented that the environment of version 2 was too simple. Without a sufficiently complex environment, it was said that it would be difficult to evaluate lighting since lighting design is motivated by the environment and the overall visual context. The final implementation version of SAIL, used for evaluation, was designed to address this comment and is described in the next section.

Fig. 54. This screenshot is from version 2 of SAIL created to address criticisms of version 1.
6.2 Current Implementation of SAIL

The screenshot in Fig. 55 shows the implementation of SAIL used for the study. The technical details of this implementation were explained in Chapter 4. This implementation was designed with several goals in mind:

1. The focus of SAIL is object lighting and a human character is often the subject of object lighting, so participants should be given a human character to control. This character should also be present in a complex environment, where complex was subjectively defined. Roughly, the complexity goal was an environment that had many visual elements, several connected but different sections, and a number of practical motivations for lighting.

2. Participants should be able to affect both the environment lighting and the character lighting. Their options should be limited however. Since the goal of the study was to generate data regarding the contributions of SAIL as an artifact, a participant becoming “bogged down” in designing the environment was undesirable.

3. Participants should be able to compare between unautomated lighting, automated lighting affected by the environment and displayed image, and automated lighting affected by only the displayed image. These three modes were presented as “natural” (unautomated lighting), “auto” (automated lighting considering both the environment and displayed image), and “attached” (automated lighting considering only the displayed image).

6.3 Sampling

19 participants took part in the study. Participants were lighting “experts”. An expert was defined as someone who had a personal or professional interest in lighting. Some
participants described themselves as lighting designers and had a working vocabulary for the tools of light and a perspective that allowed them to purposefully control illumination to achieve a goal. Other participants were art directors or similar roles. These participants had an investment in light as a significant piece of their final visual work, but they might not have the vocabulary or perspective to purposefully control light to achieve a goal.

Participants were recruited through personal contacts of the author. Each was given a survey to gather basic demographic and expertise identity data (this survey is displayed in Appendix C). Age and sex distribution of participants are shown in Fig. 56. Most participants were male. All participants were 20 years or older. Age was fairly evenly distributed between 20 and 49 years of age inclusive.

![Age Distribution](chart1.png)

**Age (18 responses out of 19 participants)**

![Sex Distribution](chart2.png)

**Sex (19 out of 19 participants)**

Fig. 56. This figure shows age and sex distribution of participants. 1 participant reported “greater than 21” for this question and is not shown in the chart.
Fig. 57 shows the years of experience and specific areas of expertise of participants. Most participants had between 2 and 10 years of experience and a background in film and/or video games.

Fig. 57. These figures show participants’ experience in years and specific areas of expertise. Participants were allowed to give multiple responses to the question regarding their area of expertise, so the individual counts do not sum to 19. Note that “Photography” replaces “Other” in the chart (“Other” was a fill in choice on the survey). In every case where a participant wrote in an answer for their area of expertise they wrote “Photography”.

Fig. 58 displays the activity level of participants. It is possible that this question was misinterpreted by participants. The “Amateur” choice was intended to reflect a background without professional training or pay (not an indication of experience or expertise). “Resident” referred specifically to salaried employees of stage theatre companies. This choice was probably too domain specific. Also, the choices of
“Production” and “Resident” should have been merged into a single “Professional” choice.

![Activity Level](image)

**Fig. 58.** This figure shows the activity level of participants. Participants were allowed to give multiple responses to this question, so the individual counts do not sum to 19. Two participants responded with “professional artist” and “installation art” for this question. These responses were included in the “Production” count. The “Amateur” choice was possibly misinterpreted as a measure of experience. It was intended to indicate practice without pay or without formal training.

The goal of this study was to generate evidential data regarding the two contributions of SAIL as an artifact (images as interface and consideration of natural lighting at runtime). Sampling was not performed to achieve a representative sample of a population but rather to collect a diverse sample of the population of lighting experts to capture different perspectives.

### 6.4 Session Design

Participant sessions were 30-45 minutes each and were typically conducted by a single investigator. Each participant took part in one session. Each session consisted of three parts.

#### 6.4.1 Part 1 - Interaction with SAIL

Participants interacted with the implementation of SAIL shown in Fig. 55. They could move the character, select between three goal images (one is displayed in the upper right of Fig. 55), select between three preset light states for the environment, and switch between “auto”, “natural”, and “attached” mode. “Auto” mode enabled SAIL, “natural” mode disabled SAIL and used only environment lighting, and “attached” mode enabled a “hacked” version of SAIL that completely disabled context considerations (SAIL applied lighting only in consideration of the goal image).
Participants were told the following about the system:

1. This is a virtual environment with an automatic lighting system.
2. Specifically, the system controls lighting on the character. The lighting on
the environment is fixed and was preset by a designer.
3. The system uses two inputs when deciding how to control lighting on the
character. It uses the image displayed in the upper right corner and it uses
the lighting in the environment.
4. The lights that actually affect the character are indicated by the two red
arrows visible in the environment.
5. “Auto” mode is a fully automatic mode where the system considers both the
environment and the image when deciding how to light the character.
6. “Natural” mode is completely unautomated. The character is illuminated by
the same lights that illuminate the environment.
7. “Attached” mode uses the same lights as automatic mode, but the system
now only considers the image and does not consider the environment when
deciding how to light the character.
8. While in “auto” mode, you may need to move the character to see the
impact of changing a setting. For example, move the character 10 paces
after changing the image.

Participants were also told about the controls available to manipulate the camera
and move the character around the environment. After this information was presented,
participants were asked to take 5 to 10 minutes to interact with the environment, until
they felt that they could discuss and talk about what was going on in the environment
with regards to the function and aesthetics of light and particularly the behavior of
light on the character.

6.4.2 Part 2 - Semi-structured Interview
Participants were interviewed using a semi-structured approach. Participants were
asked six questions. For most sessions, the order of questions listed below was
followed. In some cases, they were slightly rearranged to accommodate conversation
flow. Some questions were skipped if the investigator felt that a question was
effectively answered by a response already given by a participant. The results of each
participant interview can be seen in transcripts included in Appendix D.

All interviews started with question 1 and ended with question 6. Interviews
generally lasted between 7 and 10 minutes. Questions included:
1. Could you describe your typical process as a lighting designer?
2. What are your thoughts in general regarding the behavior of the automated
lighting system?
3. Would you describe the lighting in this environment as satisfactory?
4. What is your interpretation of the image in the upper right corner?
5. What would you have done differently with the lighting?
6. Any other thoughts or comments that haven’t come up?

Question 5 was often unnecessary because participants would answer question 3 by
describing how they would have approached lighting in the environment differently.
The investigator remained vague about the purpose and algorithms of the system
during the interview. If a participant asked a question such as, “What is the system doing here?” or “I’m interested to know what you’re trying to do here” the investigator would typically respond with, “Let’s get back to that later” or “Do you have any other impressions before I answer that question?”

6.4.3 Part 3 - Offline Discussion
After the interview was completed, an “offline” discussion was conducted. The offline discussion was the first part during which a participant was allowed unhindered access to information about the system. If a participant had asked a specific question regarding the behavior of the system earlier during the interview (for example, “What lighting model are you using?” or “What is your intended application for this system?”), that question was answered at this time. The participant and investigator then discussed the system in a freeform manner.

This part of the session was not transcribed because the flow of the conversation was not important. Rather, this portion of a session generally led to many interesting suggestions and comments about what participants felt were important research directions for SAIL in the future and for automated lighting tools in general. The goal of this part was to gather suggestions and insight from participants with knowledge of the theoretical – what the system could become in the future and what it might be able to do given a different environment or changes to the underlying algorithms.

6.4.4 Summary of Design
The design of the study sessions was approached from a deductive, qualitative mindset. The goal was to generate data to argue evidentially about SAIL’s use of images as input and SAIL’s ability to reflect natural lighting at runtime. It was the views and interpretations of lighting experts that were important, because lighting experts are the target of this work and it is their expert perspective that determines whether SAIL is a contribution as an artifact or not.

Giving participants incomplete knowledge during parts 1 and 2 was intended to encourage them to make comments that would be specific but not so specific as to have little meaning outside the very particular constraints of the demo. Further, it was hoped that it would encourage them to discuss their process. This appears to have been successful. For example, when asked what their interpretation of the images was, many participants explained their interpretation and also explained why they interpreted them as such.

6.5 Analysis
Passages from interview transcripts were categorized as evidence for either SAIL’s usage of images as interface or as evidence for SAIL’s consideration of natural lighting while lighting an object at runtime. This data was then further coded into common themes as they were identified. Discussions that occurred during the freeform conversations of part 3 were included as data to support or enhance arguments made based on the data categorized from part 2.
Chapter 7: Evaluation of SAIL - Study Results

SAIL’s contributions as an artifact are twofold: 1) using images as an interface to specify lighting goals, and 2) reflecting the natural lighting of an environment at runtime while applying a lighting goal specified by an image. Note that the numerical codes used to identify participants throughout this chapter can be used to find the complete transcription of a participant’s session in Appendix D.

7.1 Images as Interface

Evaluation indicates that the semantics of the images used in the demo were understandable to participants. The three possible images a participant could select from are shown in Fig. 59. Images represented a “lit sphere”, a technique used to study shading in illustration (Sloan, Martin et al. 2000). Each image represented lighting direction and light contrast. The meaning of the image as light direction was clearly understood by participants although some were confused about its indication of light contrast. Other participants were confused by deviation between what the image contained and what they saw on the character while SAIL was considering context. Overall, evaluation indicates that images can be a reasonable interface for specifying lighting goals but care must be taken with regards to how the image is matched with SAIL’s behavior at runtime. Specifically, the presence of a fill light might imply the elimination of absolute darkness, so the presence of absolute darkness in an example image can be confusing. Further, deviation between what a displayed image indicates (the ideal goal) and what SAIL is doing at runtime (a compromise between the ideal goal and the natural state) can be confusing. Essentially, as described by one participant, if the lighting in the scene can deviate from what is indicated by the image, then the logic of what the image represents is violated.

![Fig. 59. This figure illustrates the three images available to participants during evaluation. The leftmost image is the image attached to key “Q”. The center image is the image attached to key “W”. The right image is the image attached to key “E”.

SAIL interprets an image as visual cues indicating light direction and contrast. Participant 10_27_01 appeared to interpret the image in the same manner:
INVESTIGATOR:
So, is there any sense what it’s doing specifically right now?

10_27_01:
Well, when I look at the image and I look at the character I can see it does seem to be the direction of the lighting and the overall shading. I’d be curious to see how much I could mess with that image.

As did participant 10_28_04:
INVESTIGATOR:
What do you think the role of the image is? The image in the upper right.

10_28_04:
The lighting on the character. Or, if that was her face, if her face was a sphere, that’s how it would look.

And participant 10_28_02:
INVESTIGATOR:
Go into a little bit [pause] What’s your interpretation of what those images mean? In terms of what the system’s doing with them?

10_28_02:
My interpretation is I’m looking at the source of the lighting. When I look at this [referring to image attached to the key Q] I see there’s a top 45º angle [pause] It’s almost a rim light. It’s fairly far back, so it’s not lighting the front. It’s not all the way back ‘cause I can still see, it’s a highlight. This [referring to the image attached to key W] I’m seeing a light with more spread, lighting from the bottom right quadrant, but a fairly hard light, in the sense that this is in shadow [10_28_02 points out the upper left corner of the image]. It’s not diffused enough to get me bounce back here. This [10_28_02 points to image attached to key E] I’m looking at the same idea but from more [pause] more top-ie. It seems a little more top-ie even than if you were to switch this one that I’m looking at [10_28_02 indicates the image attached to key Q]. [pause]. [It shows] Where’s the light coming from. If this were the key light, [it shows] where’s the source and what’s the nature of the source.

Participant 10_30_02 explained why an image of a shaded circle might make sense to a lighting expert, comparing it to an “environment map”:
INVESTIGATOR:
What’s your interpretation of the image?

10_30_02:
What I get from the image is like in an environment map, it’s maybe the translation I would do. It’s like a map that you would multiply on top of the whole character. For example, if you [pause] What I was looking for was if this [10_30_02 is pointing on image #2] has a dark area on the top, I will get that always on the top of the character and it’s going to be brighter going down. That’s going to be multiplied to the frame buffer on the character area. That’s what I interpreted.

And participant 10_30_03 made a similar comparison to “hemispherical lighting”:

INVESTIGATOR:
What’s your interpretation of the image?

10_30_03:
Of the…

INVESTIGATOR:
Ya, the thing that’s surrounded in the magenta color.

10_30_03:
I would expect that would be a spherical representation of what the lighting’s doing. So a hemisphere lighting solution [pause] Not lighting solution but if I was viewing hemisphere lighting it would be in a similar format. So it’s basically where the lighting is coming from if you were looking at it on a globe.

All other participants described the image as indicating direction. Some participants gave responses that were unclear with regards to whether the image indicated light contrast or not:

INVESTIGATOR:
What do you think is the role of the image? The thing in the upper right corner.

10_28_03:
Oh, I have to admit [pause] I only read it as the source of the primary light. Might have meant something different but I have to admit I didn’t…[pause]

And participant 10_26_01 did not think the image represented a fill light, which controls contrast:
INVESTIGATOR:

On that note, what do you think the role of the image was then?

10_26_01:

The role of the image seemed like a quick, top down sort of chart of kind of where the key light was coming from. I didn’t get much feel for the fill light in this. […]

Confusion regarding a fill light is potentially described by a similar confusion and comment from participant 10_29_02…

INVESTIGATOR:

Ok. What do you think the role of the image is? In the upper right?

10_29_02:

It’s a model of the light setting parameters that you’re using [pause] Under that setting. So it’s just a representation, I think, of the direction of light and the quality of light. So that’s a smaller light source [10_29_02 has the image attached to Q selected] and it models the direction it’s coming from. Possibly without any fill light [pause] [10_29_02 interacts with the software system].

…and additional non-transcribed conversation. Participant 10_29_02 further explained the comment “Possibly without any fill light”. To participant 10_29_02, a fill light in three-point lighting is applied to set a minimum contrast and eliminate any absolute darkness. However, all of the images that participants could select from had an absolutely dark portion (see Fig. 59). SAIL was trying to recreate this feature and left part of the character completely dark. This confused participant 10_29_02 and resulted in doubt about whether there was a fill light at all.

Participant 10_26_02 was confused about the role of the image in general but still described its semantics as both indicating direction and contrast:

INVESTIGATOR:

You mentioned the image in the upper right corner, could you go into a little bit more detail about what you think the role of the image is?

10_26_02:

I actually don’t really know…

INVESTIGATOR:

Do you have any sort of inkling or would you venture a guess?

10_26_02:

I assume it’s where the lighting is coming from so, this [indicating the image in the software system] would be where the light is coming from the top or directionality as
opposed to [long pause, while changing the image in the demo]. Ya, I’m not quite sure. If I had to venture a guess I would probably say the intensity and the directionality of the character lighting. But I didn’t see it on her so I wasn’t sure if I was right or not.

Participant 10_29_01 had similar confusion:

INVESTIGATOR:
Related to that, what do you think the role of the image is, in the upper right?

10_29_01:
I would imagine that it’s sort of the lighting cast on the character. That’s what I’m guessing. [pause] By the individual light sources. I’m not entirely sure, actually, to be honest, now.

INVESTIGATOR:
You’re in auto mode, correct?

10_29_01:
Let me put it in auto mode…

INVESTIGATOR:
Try moving her a little bit after you change the image.

10_29_01:
To be honest I still don’t understand it.

This confusion is possibly explained by participant 10_30_05:

INVESTIGATOR:
What is you interpretation of the image in the upper right?

10_30_05:
I’m not really sure what you’re trying to indicate there [pause] If it’s a direction of light source because it can be connected and can be disconnected from what the lighting in your scene is, then it breaks the logic for what it’s trying to tell me. I think. I got confused when I was switching back and forth between the different modes as to what this was actually trying to do. I guess [pause] I’m familiar with lighting things in applications that have just a sphere to tell you where the light source is [pause] I’m kind of used to seeing these kinds of things and I would expect that to be an uplight, kind of very broad. Key light, rim light.
Similar confusion from participant 10_27_01 was alleviated when SAIL’s “auto” mode (the mode in which SAIL considered environment lighting) was disabled and SAIL was put into “attached” mode (where SAIL only recreated the lighting of the image and did not consider environment lighting):

10_27_01:

So, looking at the image is my guess is it’s a simplified lighting model. That allows you to use an image to basically light the scene or, like you said, just modify what’s going on in there, hopefully there’s some influence from something else. I could see something like that being used possibly on the character but certainly not on the backgrounds at all, that wouldn’t work very well. If I was going to use a lighting model like that I would probably restrict it to the character and do the backgrounds a different way.

INVESTIGATOR:

So, is there any sense what it’s doing specifically right now?

10_27_01:

Well, when I look at the image and I look at the character I can see it does seem to be the direction of the lighting and the overall shading. I’d be curious to see how much I could mess with that image.

INVESTIGATOR:

So actually manipulate the image, like in Photoshop or something?

10_27_01:

Ya, to see if it’s strictly that image that’s driving the lighting or if it’s a multiplier of it or whatever. It’d be interesting to see if I could get something a little more realistic out of it. Overall, [pause]. So this one here doesn’t respond like I’d think though. I’ve got this image here that’s just really bright in the upper left corner and the rest of it’s dark. But I don’t seem to be seeing that on the character. So maybe I’m interpreting that wrong.

INVESTIGATOR:

Well, this system does take into account both the system and the environment. So, if you [pause] The attached mode. That is purely the image.

10_27_01:

Ok ya, that’s more what I’d expect to see.
Overall, an image of a shaded sphere appears to be an understandable interface to lighting experts. Confusion seemed to stem from how SAIL was applying this image in “auto” mode. An image for specifying lighting goals appears to be a reasonable approach for future work, but care must be taken with regards to how this image is presented. Specifically, a fill light has inherent semantics that should be considered (a fill light potentially implies removal of absolute darkness). Also, it can be confusing if the image only reflects the desired lighting goal. There was an expectation that the images used in the demo reflected the appearance of the character but SAIL in “auto” mode configured lighting to achieve a compromise between the natural environment lighting and the desired lighting appearance as indicated by an image. As a result, an image almost never matched the appearance of the character.

7.2 Reflection of Natural Lighting

SAIL was successful at reflecting natural lighting while applying a lighting goal at runtime. However, many participants felt that the natural lighting state itself was bad. SAIL’s successful reflection of natural lighting was indicated mostly by comments where participants identified bad lighting that was present in both “natural” and “auto” modes.

This section includes discussion of what experts found unsatisfactory about the environment in addition to evidence for SAIL’s success at reflecting natural lighting. This discussion is included to motivate future work to create a tool that is not only predictable but also satisfying in its results.

This section is divided into three parts: 1) lighting was not sufficiently motivated by logical light sources in the environment, 2) intensity contrast was at times too great, and 3) exceptions to the rule are necessary tools in the toolbox of lighting experts.
7.2.1 Logically Motivated Light Sources

Fig. 60. Many participants expected the windows in the back of the room shown in this figure to be sources of illumination even though they were not.

Lighting direction was not sufficiently motivated by logical light sources. The first room (see Fig. 60) contained two large windows, picturing an outside spring day. Logically, these should be large sources of illumination. But they were not, as noticed by participant 10_29_02:

INVESTIGATOR:

What are your general impressions regarding this system?

10_29_02:

It’s a simple system. There’s a few things I have questions about. For example, there are windows over here. I’m just wondering why there are no lights that represent the windows, unless [pause] ‘Cause I don’t see a change going from natural to… [pause]

INVESTIGATOR:

Ya, there’s no lights representing the windows.

10_29_02:

So, I would say your ambient lighting, at the very least, should account for actual sources of light, and not just aesthetic sources. When I switch to setting 2 [referring to the environment preset attached to key 2], there seems to be a light source here [10_29_02 indicates the light source in the ceiling about the dining room table in the first room] but it doesn’t seem to cast on the table. That should be there. Also, it
doesn’t affect the character in an expected way for me. This [10_29_02 indicates the lighting in the room with the dining room table] looks like it has direction like a directional light but [pause] As this [indicating the point of light in the ceiling above the dining room table] is the only point source that I can see in the room, yet it’s [referring to the lighting of the character] coming from almost directly above. Just wondering what that choice means [note: 10_29_02 has image Q selected during this dialogue].

Participant 10_26_01 found the lighting in general to be “non-diegetic”:

INVESTIGATOR:
Would you describe the lighting as satisfactory in the environment?
10_26_01:
As I said, the fact that the lighting seemed non-diegetic to me [referring to a comment made by 10_26_01 while 10_26_01 was interacting with the system before the interview portion of the session] was a little bit confusing.

INVESTIGATOR:
Could you explain what you mean by “non-diegetic”?
10_26_01:
Sure. And diegetic might not even be used frequently with lighting it’s used with sound all the time and film and that means that there’s a difference between when you have the soundtrack playing in the background and when you have the sound seem to be coming from the radio in the car that the person is sitting in. That’s a diegetic sound, a sound that is actually emanating from the environment in some way rather than sound. So, I guess the real term, the stage term would be a practical light. You can have a lamp sitting on stage that is actually casting light or you can have a lamp sitting on stage and have a light that is up in the rafters that is trying to simulate what that light would look like coming from the lamp.

And so the one thing that I found not satisfying about the lighting in the scene as I mentioned [referring to a comment made by 10_26_01 while 10_26_01 was interacting with the system before the interview portion of the session] was the two lamps didn’t seem to be casting the light. They seemed to be representing light sources but they were representing lighting sources that were outside the room. That was a little by confusing to me.
INVESTIGATOR:
By the two lamps, you mean the lights indicated by the arrows?
10_26_01:
No, actually what I mean is the physical lamps sitting on the chest of drawers here [10_26_01 points to a lamp in the 3D scene that is located on a chest of drawers in the center room of the environment] and the one sitting on the table in the third room.
INVESTIGATOR:
So would you describe this more as a problem with the preset environment lighting?
10_26_01:
Yes, absolutely. […]
The lack of motivation for lighting from logical light sources resulted in an environment that overall was considered too “flat”:
INVESTIGATOR:
So on that note, what are your general thoughts regarding the behavior of the system?
10_27_01:
My general thoughts are that it doesn’t look like much has been put into the background at this point. I like to see a lighting environment that really looks like it all works together. I think part of that is I’m just really only evaluating the lighting on the character but it’s kind of hard to tell [pause] The character doesn’t fit in with the environment so there’s no relationship…
INVESTIGATOR:
Could you go a bit more into [pause] How specifically? [referring to the 10_27_01’s statement that the character does not fit into the environment]
10_27_01:
Specifically the environment looks like your typical evenly lit no lighting environment. I can’t see any distinguishing shading other than a very kind of high-contrast from the key light. There’s not much fill there, there’s no shadows. There’s no ambient occlusion in the corners or anything like that it’s just basically really flat.
SAIL appeared to successfully reflect the flatness of the environment when lighting the character. Participant 10_28_02 described SAIL’s “auto” mode as reflecting the “kick” off the environment while the “attached” mode (where SAIL completely ignored the environment) was described as only reflecting the “global source”, where
“global source” was participant 10_28_02’s term for light direction and contrast as indicated by the images in the demo:

INVESTIGATOR:

Go into a little bit [pause] What’s your interpretation of what those images mean? In terms of what the system’s doing with them?

10_28_02:

My interpretation is I’m looking at the source of the lighting. When I look at this [referring to image attached to the key Q] I see there’s a top 45° [pause] It’s almost a rim light. It’s fairly far back, o it’s not lighting the front. It’s not all the way back ‘cause I can still see, it’s a highlight. This [referring to the image attached to key W] I’m seeing a light with more spread, lighting from the bottom right quadrant, but a fairly hard light, in the sense that this is in shadow [10_28_02 points out the upper left corner of the image]. It’s not diffused enough to get me bounce back here. This [10_28_02 points to image attached to key E] I’m looking at the same idea but from more [pause] more top-ie. It seems a little more top-ie even than if you were to switch this one that I’m looking at [10_28_02 indicates the image attached to key Q]. [pause]. [It shows] Where’s the light coming from. If this were the key light, [it shows] where’s the source and what’s the nature of the source.

INVESTIGATOR:

Would you say based on your interpretation that what the system is doing makes sense?

10_28_02:

Yes. I’d want to spend a bit more time with it to see how wandering around in this environment, what the implications were? ‘Cause this room for example [indicating the first room with the dining room table], there’s more kick coming off the walls. So this [referring to the auto mode] gives me sort of my ambient idea and then the attached [referring to attached mode] looks like it’s just being lit from the global source.

INVESTIGATOR:

By global source, you mean the image right?
Ya, these guys [10_28_02 indicates the images displayed in the system]. So it’s independent of...[pause]

Participant 10_27_01 described the lighting in “natural” and “auto” modes as comparably poor:

INVESTIGATOR:
So the environment seems very flat, but she [referring to the character in the software system] doesn’t, or...?

10_27_01:
She seems too contrasty. It looks like there’s a bit of a fill light there, in some cases I can see a bit of a rim light. In one of those modes, little bit. But overall it’s just way too harsh. It’s not adapting well...what I want to see here is certainly more of a fill light so that I can at least see her features [note: during this sentence, the 10_27_01 is pointing out the character being lit under modes that result in significantly portions of her body being completely black or nearly completely black]. Usually what’ll happen in an animated series and such is that they’ll have a separate pass. The background’s gotta look good and the character’s gotta look good. And they’ll often have a rig that even travels along with the character, so they’ll always be a rim light to a certain amount. And they’ll just kind of cover those basic three-point lighting principles.

INVESTIGATOR:
So is that notably about the same in natural mode, ‘cause currently you’re in auto mode so if you switch it to natural mode?

10_27_01:
The natural mode is better. But it’s still [pause]. Some of that may just be the gamma of this screen too, ‘cause it looks a little bit washed out. I guess the thing that I noticed the most is that there doesn’t seem to be much interaction with the environment. And, well here I can see that I’m passing underneath a light source. A lot of this could be not necessarily the technology but just how the lights were placed. I think you can work with whatever you got, and it’s just a matter of placing the lights a little better.

Participant 10_30_01 noted an unmotivated light source in both “natural” and “auto” modes:
INVESTIGATOR:
So you would say that in natural mode, the lighting between the character and the environment doesn’t match? Or what do you mean by that?

10_30_01:
I don’t know what kind of technique you use but sometimes when you walk through the rooms of the environment you don’t really see the lighting change that much. Just the interaction between the lighting on the character to the environment, it’s not that noticeable. For instance, you can see here…

INVESTIGATOR:
So the lamp in the center room?

10_30_01:
Ya. You see here that you obviously have a dominant light source coming from this side but your character is not lit from behind [behind being the direction towards the light source in this case]. [pause] [participant interacts with demo] It would be interesting, since now you have the ability to mix between your preset lighting and your environment lighting, it would be interesting if you could set the scene, to set multiple lighting sources, if you can. Maybe one is from in this area maybe if you want to say the light is on you can have your light source put in there to light a character and then another in here, maybe you can set another light source, just another coming from in there or have some lights from this angle to simulate that lighting. And you can just lerp [linear interpolate] between them to get more interactive [pause] You’ll pick up the environment better.

INVESTIGATOR:
Is that all specifically the natural mode or is…?

10_30_01:
No, it’s not just natural mode, it’s when you have the auto mode which is kind of a blend between natural and your attached lighting. I’m just saying, for your attached to set up multiple light sources, you can use your attached lighting to do that. I don’t know exactly how you do this. But I would say just in general since you have this ability to blend [pause] You have a natural and an attached, you can use an attached
lighting mode to set up some multiple light sources and have your character interact, it would be more interesting.

And participant 10_26_02 noted a general mismatch between lighting on the character and lighting on the environment that was comparable in both “auto” and “natural” modes:

10_26_02:
And I think that might be an “attached” lighting issue. I always feel like players feel pretty flat when they don’t have shadows. And most of these rooms feel really, really over lit compared to her. ‘Cause she’s got …

INVESTIGATOR:
So you would say the rooms seem brighter than she is?

10_26_02:
Ya. She’s reacting really nicely to the light but I don’t think that it reflects it in the world very well.

INVESTIGATOR:
Is that specifically in a specific mode?

10_26_02:
It’s more the mismatch between the way she’s lit [pause]. When I expect her to have black on the back and the lighting kind of striking her like this I expect it to be darker on the back end. I’m assuming that’s because it’s vertex lighting and not many…

INVESTIGATOR:
But specifically, that seems to be the case in any mode?

10_26_02:
Mm-hmm. She tends to be lit pretty well. But the world isn’t really matching the way she’s being lit. But I was looking pretty closely at just her lighting rather than the world so I didn’t really notice until now.

SAIL was apparently successful at reflecting the natural lighting, but this lighting itself was considered poor by most participants. This specifically may not be a problem to solve as SAIL would be expected to be given better data in a context where it was used. However, for applications where better data is not an option (i.e. SAIL integrated into a learning environment and used as a tool to make lighting design easier), SAIL could possibly be integrated with a system like the Expressive Lighting Engine (Seif El-Nasr 2003) to automatically illuminate the environment.
7.2.2 Extreme Light Contrast
The environment of the demo had locations from which both SAIL and the natural lighting would create extreme light contrast on the character, an effect considered undesirable:

INVESTIGATOR:

Could you go into that a bit more specifically? As in, what are you specifically noticing that’s different [pause] That you said you’re not going about it the same kind of way?

10_30_04:

Well, because of memory constraints and stuff we have certain lights that are dynamic and other ones that are baked. So, using spherical harmonics or whatever we bake in the fill or we bake in two lights and we have one light [pause] Our key light we’ll try to get dynamic or our rim light we’ll always keep dynamic. And then other things will be baked per environment. So that we get a nice mix of shadow plus occlusion [occlusion in this case refers to ambient occlusion. Several participants appeared to use “occlusion” to refer to ambient occlusion and to infer the general global illumination effect of ambient occlusion, not the literal definition of occlusion, which results in a loss of light] and everything.

That’s one thing that I found that the darks are really dark. Like you’d never get blacks that black on the character, that kind of thing. So you need some sort of occlusion in there.

Echoed by participant 10_28_04:

INVESTIGATOR:

Let’s say I gave you this room, completely unlit. Would you light it the same way? What would you do differently?

10_28_04:

Like this current look right now? I don’t really like these completely dark spots on her [participant indicates areas of complete black on the character]. And sometimes how her lighting doesn’t match with the background. If you just talk about these three settings on the background [referring to the room presets], it is. It could have used shadows I guess, but. Shadows and [pause] Not so even. Not so completely even. Darker corners maybe. Give some more depth to…[pause]
Why this effect is undesirable was explained in a non-transcribed conversation with participant 10_27_01. Simply put, it is physically implausible. Although lighting design is to a great extent about cheating physical reality, it is also about respecting physical reality to make lighting plausible (Birn 2006). Complete darkness is typically outside this respect of physical reality. Any light source in a room, due to indirect reflection, will invariably cause some illumination to illuminate everywhere and allow our eyes to make out some detail. A completely dark environment “simply can’t happen”, as participant 10_27_01 put it.

It seems important to consider this in future iterations of SAIL. SAIL should probably avoid extreme light contrast by default and not allow an object to become completely dark.

7.2.3 Exceptions to the Rule
The previous two sections have argued that lighting motivated by logical sources of light is important and that extreme light contrast is usually undesirable. These can be identified as rules that an automated system such as SAIL should follow. However, an equally important “rule” is to allow a lighting expert to break rules if desired. Both participant 10_29_03 and participant 10_28_04 found “attached” mode to be interesting for the ability to detach the character lighting completely from the environment and therefore completely ignore the logical sources of light:

10_29_03:
What I really am most fond of is this ability to detach [pause] I like that ability to detach the figure from the environment. Just because it's a kind of shadow like effect where I can make things that I regard as less important [pause] But I sure would like to be able to light something in the background. Oh, I don’t know why because in the context of a game maybe you don’t want to do that, there’s certainly the game isn’t going to be about designing the lighting.

And participant 10_28_04:
INVESTIGATOR:
What are your general reactions to the behavior of this system?

10_28_04:
I thought it was neat how you could detach the lights [pause] Like, this is a scary look [participant has selected image attached to letter W] with this, but if you put this it doesn’t quite match.

INVESTIGATOR:
So you’re talking about attached mode, specifically?
10_28_04:

Ya, attached mode. It gives a more cinematic look I guess. If this were a game you could create more cinematic looks on [pause] Really highlight your character how you want. Independent of the environment or [pause] You just have more control I guess. I haven’t done any lighting in games but [pause] Ya, it sounds like you’d have more control.

This was echoed by participant 10_28_03 who described it as “my favorite mode”:

INVESTIGATOR:

So, overall would you call the lighting satisfactory?

10_28_03:

In some modes in some locations. The night location was in some ways the most interesting but only if there was a source of strong light. If she was on the other side of the room or further down this way [participant indicating the corner of the room with the dining room table near the entrance to the adjacent room] there just wasn’t any light coming in on her, and the night mode [pause] Or if she gets over here [participant puts the character completely in the corner between the room with the dining room table and the room with the radio] she’s just completely dark.

INVESTIGATOR:

Well, actually that could be attached mode [pause] I think you’re in attached mode. Attached is meant to be intentionally sort of very, very disconnected from… [pause]

10_28_03:

If I’m in auto there [pause] The night in auto there, you can’t really do very much [pause] And then in attached mode it becomes quite interesting in fact that may be my favorite mode, in truth.

Also, despite the general dislike of complete darkness, participant 10_29_06 described a possible usage for this effect:

INVESTIGATOR:

What are your general reactions to this?

10_29_06:

I’m not sure what’s going on [pause] Why the face is so dark while I’m in this room. ‘Cause when I went to the other rooms and hid behind a wall, the same reaction did not happen. It seemed that it did not take the situation of blocking. I also wanted to
see if it’s going to make me feel different when I was changing different light modes. Yes and no. The person didn’t really react [pause] I couldn’t really tell between E and W [referring to lighting images attached to E and W] but in this mode, which one is it? In the auto one, it seemed to be the better looking one, although again you have these shadows. In advertising you would never want to do that unless [pause] You don’t want them to become so dark unless there’s a reason for it. I was trying to figure out what the reason was she had a shadow cast on her in this spot when [pause] I understand that there’s a light up here but it just seems strange that when you’re closest that you wouldn’t have a corrective light to shine on her face. The one that I liked the most was the auto mode, that wasn’t too dark.

[...]

INVESTIGATOR:
Specifically, would you describe the lighting as satisfactory? You talked a little bit about it but…

10_29_06:
It depends on what exactly you want to do. If she was trying to move [pause] I’m a sensitive person and I come here and now I actually realize that I’m guilty [10_29_06 has moved the character to a spot where much of her face is complete blackness] where she slowly transitions to dark, then I get it. But to me it’s a bit dramatic in this case for no reason. I don’t get the reason, I don’t see the context. This one seems too plain to me, the S, the natural one. She seems very flat. [pause] But I’m not getting the same effect. Like, I don’t understand why this effect happens here. It’s funny how you’re trying to see the details of her face and you can’t. I think that kind of bothers you from the beginning because I think people have a tendency to really focus on facial features.

Ya, it becomes too dark too soon, I guess.

The effect of turning off the light it feels to me like the end of Act I or something like that [10_29_06 has transitioned room lighting cue to cue 2, the “nighttime” cue].

The conclusion is that experts may want to completely violate any rule that is inherent to an automated lighting system. Although participants were able to detach lighting using the “attached” mode of the demo, SAIL was never designed with this mode in mind. It was effectively a “hack” for comparison purposes. SAIL has no manner in which to react or accommodate an exception such as completely detaching lighting but evaluation indicates that these exceptions are desired and will be used in a
practical application of SAIL. Therefore, future work on SAIL should incorporate exceptions in an integrated fashion.
Chapter 8: Conclusions

This chapter presents conclusions regarding SAIL divided into 5 sections. Section 1 summarizes SAIL’s contributions as an artifact. Section 2 discusses SAIL as a design tool and overviews what was learned by interacting with lighting experts. Section 3 discusses SAIL as an experimental instrument and suggests how SAIL might be useful for future experimental work. Section 4 discusses future work and research directions for SAIL and automated lighting work in general. Section 5 concludes the dissertation with closing thoughts.

8.1 SAIL as an Artifact

SAIL contributes two functions that differ from related work. First, SAIL includes an image analysis component to understand light features that does not require correlated geometry. This is in contrast to similar functionality in related perceptual inverse lighting work (see Chapter 3). Second, SAIL applies this image analysis to real-time lighting, allowing it to adapt lighting based on an object’s appearance. This is in contrast to related work such as the Expressive Lighting Engine (ELE) (Seif El-Nasr 2003) which uses only a rough approximation of an object’s shape and its position to adapt lighting. This section discusses potential uses of these contributions.

8.1.1 Reverse Engineering Lighting

In non-transcribed conversation, participant 10_30_05 suggested an interesting potential application for the image analysis algorithms of SAIL (see Appendix D for the transcribed interview with participant 10_30_05). Billboards are sometimes used in video games (see Fig. 61). Billboards refer to, effectively, “cardboard cutouts” of rendered or photographed objects. These cardboard cutouts are used because they are less computationally expensive to render than full 3D geometry in large numbers and within limits can be visually imperceptible from real 3D geometry.
A challenge with billboards however is that the image on the billboard can be generated in such a way that it has inherent lighting that is not the same as the lighting environment the billboard is used in. If this occurs, it is not always possible to regenerate the billboard with desired lighting. This results in an object with an unknown lighting state being illuminated to fit into a different lighting state.

Participant 10_30_05 suggested that an algorithm which could reverse engineer the lighting contained within a billboard would be useful. The algorithms of SAIL could be intuitively applied to this problem. The desired lighting could be specified by an image and SAIL could then analyze the billboard under different lighting conditions in the same way as would be done with a regular 3D model. SAIL would then apply lighting that would achieve the lighting as specified by an image and achieve a desired appearance.

8.1.2 Lighting Design Vocabulary
SAIL does not require correlated geometry to understand lighting. As a result, a specialized “visual language” could be explored. Lighting could be indicated by a user drawing swathes of color or other abstract shapes that have no physical representation. This ability to paint an image seems like a natural interface and was suggested by participant 10_27_01:

10_27_01:
Well, when I look at the image and I look at the character I can see it does seem to be the direction of the lighting and the overall shading. I’d be curious to see how much I could mess with that image.

INVESTIGATOR:
So actually manipulate the image, like in Photoshop or something?
8.1.3 New Styles of Interactive Lighting
SAIL’s adaptive functionality can potentially facilitate lighting in interactive environment that is not currently possible. For example, the video game Doom 3 (released in 2004 by id Software) used extremely low-key lighting that could sometimes result in total darkness. The player could compensate for this total darkness by using a flashlight. However, designing an extremely low-key lighting environment similar to Doom 3 that does not require a flashlight but rather relies only on natural lighting is very difficult. Inevitably, the player will encounter points of view from which the visible environment is completely dark, an undesirable situation. SAIL’s ability to adapt to changes in context could be applied to this problem to create a solution that does not require an explicit light object controlled by the player.

8.1.4 Interactive Storytelling
The goal of interactive storytelling research (Glassner 2004; Crawford 2005) is very dynamic storytelling. In these environments, very little can be scripted because very little can be assumed about the state of the player at any given time. Traditionally, as discussed in Chapter 1, if a lighting designer wanted to create an evocative moment in an interactive environment with light, she would find a way to constrain the player (for example, locking him in a room for 5 seconds) and would then script the lighting as desired. This approach is not an option in an interactive storytelling environment but SAIL’s ability to adapt lighting provides a potential solution.

Emotional moments could be described with images that SAIL’s image analysis functionality would encode. When these moments trigger in the interactive story, SAIL’s adaptive functionality could then apply the encoded lighting to the context of a player.

8.2 SAIL as a Design Tool
The study presented in Chapters 6 and 7 indicates that for SAIL to be useful as a design tool, several issues need to be addressed. Flat lighting and complete darkness are typically undesirable (see Chapter 7) unless these features are needed for a very specific visual effect. This indicates that SAIL needs to incorporate limits on the designs that it produces. However, SAIL also needs to include exceptions to its rules. If a designer wants complete darkness or very flat lighting in a specific instance, SAIL should be able to achieve these effects in a considerate manner.

Similarly, many participants of the study indicated that lighting needs to be motivated by objects in an environment that appear to be sources of light (see Chapter 7). This indicates that SAIL is ultimately dependent on the quality of the input data that specifies the lighting of an environment. This dependency either necessitates better data or it demands additional mechanisms to generate the environmental
lighting and therefore guarantee the quality of the data. For example, ELE could be integrated with SAIL in order to light the environments that provide the lighting context for an object that SAIL illuminates.

With these issues addressed, several scenarios are possible in which SAIL could be used as a design tool.

8.2.1 Pre-visualization or Storyboarding

During the study described in Chapters 6 and 7, several participants suggested that SAIL could be useful as a storyboarding or pre-visualization tool. In particular, it was suggested that using an image to display desired shading is a more understandable concept to experts who may not be experts in lighting but who are still invested in the visual results of a project. For example, the director of a film might know that she is unhappy with the lighting of a scene but may not have the vocabulary to describe specifically what is wrong and how it should be changed. It was suggested that using an image would bridge this vocabulary gap. For example, participant 10_28_02 who has a background in film:

INVESTIGATOR:

Anything else that you have to say about this that I haven’t mentioned?

10_28_02:

This as a tool [pause] With more options it could be a really fantastic and quick way to pre-visualize lighting in a game. Or even in an animated storyboard for a film. It’s not detailed or refined enough to actually duplicate a full fledged lighting setup. You’d have to add lots more lights, there’d be a lot more going on. But as a way to pre-visualize and strategize about lighting, it’s very cool. It could be a really neat [pause] It’s a great alignment tool.

INVESTIGATOR:

Alignment?

10_28_02:

You could sit down with a designer or a DOP or whatever and you’re looking at the image and you can change it and so you’re getting [pause] You’re on the same page. You start to align everyone on the team because they can see what some of the options are and what the choices are. And that’s really valuable.

Further, it was suggested that the constrained setup of the demo used in the study could be useful for storyboarding, where the goal is to lay out the major shots of an entire visual piece as quickly as possible. Using an image to specify general shading of a scene could result in better results than time would allow to be done by hand.
8.2.2 Simplifying Iteration in a Video Game Pipeline
A possible process to video game lighting is as follows:
1) Scene content is created and then illuminated by an artist.
2) Example images of the illuminated scene from important points of interest are collected and then given to a lead artist.
3) The lead artist uses 2D image manipulation software (such as Adobe Photoshop) to modify the example images to what she desires them to look like by manipulating shading, color, intensity, and other elements in the images.
4) The modified images are returned to an artist who then makes the necessary changes to the lighting in the 3D environment to match the annotated images.

In this scenario, SAIL could take the place of step 4. Given annotated images and an environment, SAIL could apply lighting to objects in the environment to achieve the desired illumination features. It could further simplify step 1 by requiring an artist to only create the unlit environment. SAIL could then apply a default illumination to the environment by using target images that provide a default illumination appearance.

8.2.3 Reducing Cognitive Load in Learning Environments
SAIL could also be used as a tool in educational environments. An example case is IST Gaming for Girls (Hopkins 2006). Gaming for Girls was a biyearly class held for middle school and high school girls. It aimed to teach basic IT skills to girls through video game “mod” development. A “mod” is a modification of an existing game to create a new game or experience. Students would learn how to interact with a tool and then manipulate existing art content (or in some cases, create their own content) in order to tell new stories and create new game play scenarios.

One difficulty of Gaming for Girls was using modern visualization environments. Many students verbally expressed a desire to create complex 3D objects or at the very least, create games that looked like the games they purchased and played. Unfortunately, modern game software is complex and teaching all the fundamentals necessary to navigate and create a full 3D environment is a daunting task in the short timeframe of Gaming for Girls (usually once a week for four to six weeks). Simplification of the number of concepts a student needed to consider was necessary and this was often achieved by using old game engines or constrained editing environments that produced results which appeared outdated to students.

In this scenario, SAIL offers a possible way to simplify the complexity of a content creation environment while still allowing modern fidelity. Other developments would be necessary (the creation of objects that respond to light is still a complex task), but a student who could apply lighting by providing a picture (or other possible interfaces, such as selecting a picture) would be freed from the manual and complex iterative process normally associated with finding and specifying light source parameters to illuminate an environment.
8.3 SAIL as an Experimental Instrument

Because SAIL can adapt to the effects of light on an object’s appearance, it offers the possibility of simplifying experimental work dealing with lighting in interactive environments and the possibility of broadening the scope of experimental work.

For example, the study described in (Seif El-Nasr and Rao 2004) evaluated the effectiveness of ELE to improve a player’s ability to recognize enemies in a first-person shooter (FPS) video game. This study used the video game Unreal Tournament 2004 (released in 2004 by Epic Games) and a custom environment hand crafted by its investigators. As mentioned in Chapter 4, ELE does not consider the appearance of an object. ELE was tuned specifically for the environment of this study and the appearance of the characters in this environment. As a result, broadening the experiment of this study to a large number of environments or characters would be very difficult. Also, this experiment could not be conducted with ELE if its environment could change dynamically (for example, if a player could destroy parts of the environment or if a player could rearrange the environment to make new structures). Neither of these cases would be a problem for the algorithms of SAIL. Therefore, SAIL offers the possibility for a wider variety of experiments into visibility, object recognition, and the effects of lighting on visibility and object recognition in interactive virtual environments such as video games.

The experiment described in (Seif El-Nasr, Zupko et al. 2006) offers another situation in which SAIL’s ability to adapt to lighting appearance could be useful. This experiment attempted to identify relationships between patterns of light (for example, highly saturated light) and a player’s tension level. The experiment was conducted by applying light changes over time in response to a player’s actions (for example, by varying the saturation of room lighting in response to the number of enemies nearby to the player). The precise light settings used were specific to the environment used in the experiment and would need to be redesigned if a new environment was used. SAIL’s ability to apply a lighting design in consideration of an environment offers the potential for a single design to be created to test a factor (such as the effects of saturated lighting). This design could then be applied to multiple environments.

SAIL may also be useful for experimental work into interactive virtual environments that is not focused on lighting. Content creation for a virtual space is a nontrivial task. As was discussed in Chapter 1, lighting designers are often needed to design lighting due to the complexity and difficulty of creating effective lighting. By applying SAIL’s ability to adapt a lighting design to multiple contexts, a lighting designer could provide several designs for a character (for example, a design that makes the character appear “warm” or a design that makes the character appear ominous). These designs could then be used across multiple characters, environments, and even experiments.
8.4 Future Work

This section presents suggestions for future work on SAIL based on the results of the qualitative study presented in Chapters 6 and 7 and on missing features explicitly identified in Chapters 4 and 5.

8.4.1 High Dynamic Range (HDR) Lighting
High dynamic range (HDR) lighting (Reinhard, Stark et al. 2002) with tone mapping is a technique used to mimic the effects of exposure in real-time rendering that are present in photography and film. Exposure greatly controls overall light contrast of these media and it is therefore tightly coupled with SAIL’s goal of controlling light contrast. SAIL does not consider and has not been integrated with HDR. Consideration of HDR could take two forms: 1) SAIL would control tone mapping as part of its runtime adaptation of lighting, or 2) SAIL would consider tone mapping as a contextual constraint and respond to changes in tone mapping similarly to how it responds to changes in natural light direction or contrast. It is also possible that SAIL would do some combination of both.

8.4.2 Back Light
As mentioned in Chapter 5, SAIL does not currently use a back light in its version of the three-point lighting technique. In three-point lighting, a key light is used to control dominant direction, a fill light controls tonal contrast, and a back light isolates an object from its environment. SAIL’s understanding of an image is low frequency (see Chapter 5) and the effects of a back light are high frequency. Extending SAIL’s image analysis to include object recognition and image segmentation will probably be necessary to effectively quantify the effects of a back light.

8.4.3 Domain Specificity
An important topic for future work is considering illumination needs that are specific to certain applications. For example, a video game using a theateric style vs. a cinematic style will have additional considerations for SAIL. If SAIL was integrated into a pipeline as a storyboarding or pre-visualization tool, this would have very different considerations as well. For example, the focus would probably shift from believable adaptation to usefulness and usability.

These domain specific considerations could also be used to address SAIL’s adaptation to a bad environment. For example, a video game with a theateric lighting style might allow for significant deviation from the logical direction of lighting. In this case, algorithms from ELE could be incorporated to produce aesthetically more pleasing lighting.

8.4.4 Scene Lighting
Future work should also explore the illumination of entire scenes (see Fig. 62). While SAIL can currently light a scene as a collection of individual objects, it does so by considering each object as a separate piece. Lighting multiple pieces as holistic swathes, much as a landscape painter might shade a painting, is a different conceptual
problem that might be useful for computational and aesthetic reasons. Computationally, SAIL would do less work by deciding to use a single light to light the trees shown in Fig. 62 than by using two lights to illuminate each tree. Aesthetically, SAIL lights each object to a goal without consideration of other objects. By considering objects together, SAIL could create contrast between the mountains in the background of Fig. 62 and the foreground, drawing visual attention and guiding the player to these mountains.

Fig. 62. This image from the video game *Guild Wars* (released in 2005 by ArenaNet) illustrates a situation where SAIL would probably be ineffective. A wide landscape style portrait such as this figure requires lighting that considers individual objects (such as the trees or mountains in this figure) as collections of objects to be lit to the same goal. SAIL would consider all the objects in this scene as individual pieces.

The interface for lighting a scene is also an interesting question. One possibility is to extend the image as interface concept of SAIL to interpret an image for an entire scene instead of a single object, as inspired by comments such as this comment from participant 10_29_04:

10_29_04:

I set up the stage and first develop what I actually want to light for. So I usually do research. So I can support whatever I’m wanting to light. So that’s in the background of what you’re showing me. And…

INVESTIGATOR:

Could you go into the research part a bit, specifically?
If I was doing a video piece where I would require quite a bit of lighting to set the stage, I would look into what my concept is about or what it’s addressing. I would kind of mimic off some of the lighting that was done in the films I’m being inspired by. That I’m using for the piece. I just do a whole bunch of research by just looking at different images and different angle shots and the consistency.

8.4.5 Shadows
Additional exploration of the relationship between multiple objects and the effects of cast and attached shadow are also needed. (Lee, Hao et al. 2006) is the only automated lighting work to explicitly consider cast shadow and its algorithms are too expensive for an interactive real-time environment. Considering cast shadow is a difficult problem because the response of shadowed areas is very discontinuous with a change in light parameters. Shadow is a combination of the parameters of the occluder, the occludee, and the relationship between the two and any light sources. This relationship is challenging to unpack. Similar challenges face the use of gobos, which are cutouts placed on lights to create patterns (see Fig. 63).

![Gobo Image](http://www.rosco.com/us/gobocatalog/fire.html)

**Fig. 63.** This image from http://www.rosco.com/us/gobocatalog/fire.html is a gobo, a template that is placed over a physical light to cast patterns into an environment. Various rendering techniques are used to create similar effects in computer rendering.

It should be emphasized that consideration of shadow in future work is important. The lack or inappropriateness of shadow was mentioned by many participants. For example, participant 10_29_07:

**INVESTIGATOR:**

In general would you describe this lighting as satisfactory?

[...]

10_29_04:
The first thing that I find is that the shadow is not quite good. I don’t know whether it’s very closely related to your research, but I didn’t find the shadow for some of the objects. […]

Specifically the environment looks like your typical evenly lit no lighting environment. I can’t see any distinguishing shading other than a very kind of high-contrast from the key light. There’s not much fill there, there’s no shadows. There’s no ambient occlusion in the corners or anything like that it’s just basically really flat.

Like this current look right now? I don’t really like these completely dark spots on her [participant indicates areas of complete black on the character]. And sometimes how her lighting doesn’t match with the background. If you just talk about these three settings on the background [referring to the room presets], it is. It could have used shadows I guess, but. Shadows and [pause] Not so even. Not so completely even. Darker corners maybe. Give some more depth to…[pause]

So you mean the unevenness of the shadows is a problem or they should be more uneven?

More uneven. The furniture doesn’t even really have shadows. Ya, you put a light there, put a light there.

And I think I talked about the windows. I guess you’re not really considering shadows or anything in here. I would expect some sort of shadow here. Some ambient…

Around where the chair is?
10_29_01:
Coming from the windows. So there should be some lights coming through the windows casting shadows…

INVESTIGATOR:
You mean the actual…[pause].

10_29_01:
The mulleins or whatever.

INVESTIGATOR:
The mulleins.

10_29_01:
Or whatever they’re called should be casting. It looks like from the image I see in the background [10_29_01 is referring to the images placed behind the windows to create an illusion of outdoor space] that the light is sort of that way. I would expect the Sun or the outdoor contribution to be in this general direction. So the window shadows should sort of splay out in this direction. There should be some lighting coming through the windows. I don’t see any contributions of the outdoor lighting on the interior environment. Nothing significant.

Although it is difficult to derive specifics from these interviews with confidence, it is safe to assume that the involvement of shadow is necessary and complex. Self-shadowing, soft shadowing, and even forms of shading that are strictly not the effects of occlusion all seem to be involved in what participants discussed when referring to shadow during interviews.

8.4.6 Outdoor Lighting
Several participants indicated that outdoor scenes are lit differently than indoor scenes, such as participant 10_29_05:

10_29_05:
Possibly a bigger environment. It’d be nice to see how this lighting would deal in outside environments possibly vs. inside. Possibly different characters. Just to see the effects on different models and different shapes.

Participant 10_30_01:

10_30_01:
[...]
For characters, usually we light a character just inside a game. So all the lighting is built into shaders. In terms of what kind of lighting model we use, usually it depends on the project need. Some projects are more focused on indoor lighting then you really don’t need that kind of dominant light source like outdoor would have. For those types of games, usually they use image based lighting more than the other. For outdoor, you will have a dominant light as your key light source. We also use image based lighting or any other type of technique like spherical harmonics or just a hemispherical light to simulate the environment terrain. To pick up all those environment lights field into your characters.

And participant 10_28_02:

10_28_02:

[…] You block out the areas where stuff is happening based on the narrative then control the light to create the mood to focus on your character or pick out things in the environment that you want your audience to see it's a very [pause] Or you go outside and everything’s bright and that’s a different [pause] As an interior, this is the most challenging lighting setup, actually.

What specific considerations would be needed in an outdoor environment is a completely open question with regards to SAIL.

8.5 Closing

This dissertation described the System for Automated Interactive Lighting (SAIL). SAIL is an automated lighting system for video games or other interactive virtual environments that adapts lighting to changes in context at runtime. It attempts to maintain lighting design goals stipulated by a designer to the unpredictable environments of video games by balancing the ideal goal of a designer with the natural lighting state of an environment at any given time. A qualitative study conducted to evaluate SAIL shows that it successfully makes two contributions.

First, SAIL uses images to specify lighting goals. Images can be an understandable interface for lighting experts to express lighting goals and SAIL’s interpretation of an image can be in line with the interpretation of these experts. However, SAIL’s adaptive behavior, during which it deviates from the goal specified in an image to accommodate context, can be confusing. A focus for SAIL or other automated lighting work in the future should be on this accommodation and how lighting experts expect an automated system to behave when balancing an ideal state with a natural state.
Second, SAIL successfully reflects the natural environment of an object at runtime. However, an interesting observation is that the quality of the lighting of an environment will usually trump SAIL’s behavior. During the study when SAIL accurately reflected the “badly” lit environment of the study, the results were considered undesirable. In fact, several participants preferred lighting on the character used in the study that was completely detached but looked “good” in comparison to the character’s surroundings. This observation implies that future versions of SAIL should include limits on its reflection of an environment, preventing certain undesirable extremes such as allowing lighting to become completely dark. It also implies that SAIL should account for exceptions to its rules. For example, in cases where a lighting designer wants completely detached lighting, SAIL should be able to achieve this in a controlled fashion that takes the detached lighting into account.

With these considerations, there are several situations in which SAIL may be used, including: reverse engineering lighting, as a pre-visualization or storyboarding tool, as a tool in education for simplifying content creation, or as an experimental tool, broadening or simplifying experimental work conducted with interactive environments.
References


DiGiulio, E. "Two Special Lenses for "Barry Lyndon" by Ed DiGiulio." American Cinematographer


Appendix A: Reducing Memory Requirements of SAIL

The response surface of an object is a mathematical model that is fitted to a particular object and stores how it responds to changes in lighting. At runtime, this response surface is used to determine light placement to achieve goals. Each point on the response surface consists of 4 32-bit floating point variables: 1) max intensity (the maximum pixel value in the image), 2) entropy as defined in Eq. 07, 3) θ as defined in Chapter 5 in Section 5.1.1, and 4) β as defined in Eq. 06.

Images are generated by jitter sampling the five independent variables, fill intensity, key roll, key yaw, camera pitch, and camera yaw as defined in Chapter 5 in Section 5.2. In the current implementation, 12 samples are taken along key roll and 13 samples along fill intensity and key yaw. 8 samples of camera yaw are taken for camera pitch values of 0°, -45°, and 45°. One sample is taken at camera pitch values of -90° and 90°. This requires $(12 \times 13 \times 13 \times (8 \times 3 + 2)) = 52,728$ samples and $(52,728 \times (4 \times 4)) = 843,648$ bytes of total storage space for an object.

The rest of this appendix provides suggestions for how this storage requirement can be reduced.

A.1 Compression

Although not tested, it seems that common compression algorithms used in real-time visualization applications such as zlib (Gailly and Adler 2005) will be effective at reducing the storage requirements for this data. However, this will only reduce permanent storage requirements; volatile memory requirements at runtime will still be a problem.

A.2 Reduce Sample Frequency

Sampling key roll at 12 samples and key yaw and fill intensity at 13 samples was a choice based on empirical evaluation of the behavior of stochastic gradient descent on the character used in the demo environment used during the study as described in Chapter 6. It is entirely possible that different characters in different environments will require a lower sampling frequency, particularly across key yaw and fill intensity.

A.3 Limit the Points of View

For an object that will be known to be primarily viewed from a limited number of angles, a very simple solution is to reduce the number of points of view that the model is sampled from. By default, the response surface generator samples an object from 26 camera positions. However, in the case of a bipedal human that will be mostly viewed from the front, it may be sufficient to only sample from the front of the character, the front and rear of the character, or the front, rear, and one or both sides of the
character. The implementation of SAIL used during the study described in Chapter 6 only samples the character from the front, for example.

A.4 Symmetry

A similar solution to A.3 is to take advantage of symmetry of an object. Not all objects vary asymmetrically across their surface. Any object that exhibits symmetry across its surface can be exploited to reduce the number of samples necessary to represent the response surface of that object. For example, in the case of an assumed symmetrical human body (see Fig. 64) where symmetry between the left half and right half of the body is used, only samples from one half need to be stored. An object that is very close to a sphere in shape may only require samples from one direction to accurately describe it from all angles.

Fig. 64. This figure illustrates the rough symmetry of a human model that can be exploited to reduce the memory storage requirements of the mathematical model used by SAIL to understand how that object responds to light.

A.5 Fitting Samples to a Function

By fitting the samples to a function, only the coefficients to this function need to be stored and the original samples can be derived at runtime. Experimental data indicates that given intensity for the fill, one dimension of the key direction is approximately cubic with relation to the illumination metrics used for shading direction and contrast. An average sample plot from several objects is shown in Fig. 65.
Fig. 65. This graph demonstrates the roughly cubic relationship between a change in key yaw direction and the $\beta$ image metric across 0.1 coarse samplings of fill intensity.

To take advantage of this relationship, a cubic polynomial can be used of the form:

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

Eq. 10. Form of a cubic polynomial used for encoding the response surface.

Two polynomials of this form would be used. One relates changes in key direction to changes in shading direction. A second relates changes in key direction to changes in intensity contrast. To find the coefficients of this polynomial, linear regression of a cubic equation is used. Linear regression finds the coefficients for a polynomial that minimize the error between the output of the equation and a set of sample points. The error is defined as the mean square error which is:

$$\text{MSE}(f) = \frac{1}{n} \sum_{j=1}^{n} (f(x_j) - \overline{f}(x_j))^2$$

Eq. 11. Mean square error used for linear regression.

Where $n$ is the number of samples, $f(x_j)$ is the jth sampled independent variable for the jth dependent variable, and $\overline{f}(x_j)$ is the output of a function at the jth dependent variable.

Once the cubic functions are found, the response surface can be stored as samples for the camera direction and one dimension of the key direction. For each sample of fill intensity a pair of cubic polynomials is found that relates changes in the second key direction to changes in illumination features.
Appendix B: Shader Code for Current Rasterizer of SAIL

This appendix lists the shader code used for materials and lighting of the rasterizer of the current implementation of SAIL. This is the HLSL code of a Microsoft Effect file (Microsoft 2008). In general, an \#if \<identifier>\ directive indicates a conditional compilation directive used when content is being imported. The shader used for a 3D model is custom built based on the requirements for that model. For example, an animated model has different requirements from a static model.

Note that the material definitions here are intended to recreate the Profile\_COMMON namespace of COLLADA. See for (Group 2008) more information:

```c
// helper macros – simplify some of the later code based on directives defined // by the shader build environment. //------------------------------------------------------------------------------
#if defined(DIFFUSE_COLOR) || defined(DIFFUSE_TEXTURE) || defined(DIFFUSE_VERTEX)
  #define DIFFUSE
#endif
#if defined(AMBIENT_COLOR) || defined(AMBIENT_TEXTURE)
  #define AMBIENT
#endif
#if defined(EMISSION_COLOR) || defined(EMISSION_TEXTURE)
  #define EMISSION
#endif
#if defined(TRANSPARENT_COLOR) || defined(TRANSPARENT_TEXTURE)
  #define TRANSPARENT
#endif
#if defined(REFLECTIVE_COLOR) || defined(REFLECTIVE_TEXTURE)
  #define REFLECTIVE
#endif
#if defined(SPECULAR_COLOR) || defined(SPECULAR_TEXTURE)
  #define SPECULAR
#endif
#if defined(BUMP_TEXTURE)
  #define BUMP
#endif

// static constants – compile time constants. //------------------------------------------------------------------------------
static const int kSkinningMatricesCount = 72;
static const int kSkinningMatricesSize = kSkinningMatricesCount * 3;
static const float kLooseTolerance = 1e-3;
static const int kShadowDimension = 512;
static const float kShadowDelta = 1.0f / (float)kShadowDimension;
static const float kShadowDepthBias = 3.81e-4;

// This is the alpha value that will be considered opaque. Any alpha equal to or // greater than this value will be rendered with zwrites enabled and alpha // blending turned off.
#define OPAQUE_OF_TRANSPARENCY 127
#define OPAQUE_OF_TRANSPARENCY_F (127.0 / 255.0)
```
#define BACK_FACE_CULLING Ccw
#define FRONT_FACE_CULLING Cw

// This indicates that objects with transparent textures defined
// should be rendered as "1-bit alpha" – the alpha is either considered
// as fully transparent or fully opaque, the transparent regions are
// not drawn at all and the opaque regions are drawn as normal opaque
// objects.
#define TRANSPARENT_TEXTURE_1_BIT 1

//-----------------------------------------------------------------------------
// constants – set by the engine at runtime.
//-----------------------------------------------------------------------------
#if defined(ANIMATED)
float4 SkinningTransforms[kSkinningMatricesSize] : sail_SkinningTransforms;
#endif

float Gamma : sail_Gamma;
float4x4 InverseViewTransform : sail_InverseViewTransform;
float4x4 InverseTransposeWorldTransform : sail_InverseTransposeWorldTransform;
float3 LightAttenuation : sail_LightAttenuation;
float3 LightDiffuse : sail_LightDiffuse;
float3 LightPositionOrDirection : sail_LightPositionOrDirection;
float3 LightSpecular : sail_LightSpecular;
float4 PickingColor : sail_PickingColor;
float ShadowFarDepth : sail_ShadowRange;
texture ShadowTexture : sail_ShadowTexture;
float4x4 ShadowTransform : sail_ShadowTransform;
float3 SpotDirection : sail_SpotDirection;
float SpotFalloffCosAngle : sail_SpotCutoffCosHalfAngle;
float SpotFalloffExponent : sail_SpotFalloffExponent;
float4x4 ViewTransform : sail_ViewTransform;
float4x4 ViewProjectionTransform : sail_ViewProjectionTransform;
float4x4 WorldTransform : sail_WorldTransform;
sampler ShadowSampler = sampler_state
{
  texture = <ShadowTexture>;
  AddressU = clamp;
  AddressV = clamp;
  MinFilter = POINT;
  MagFilter = POINT;
  MipFilter = NONE;
};
#if defined(EMISSION_COLOR)
float4 EmissionColor : EMISSION_COLOR;
#endif
#if defined(REFLECTIVE_COLOR)
float4 ReflectiveColor : REFLECTIVE_COLOR;
#endif

sampler EmissionSampler = sampler_state
{
  AddressU = EMISSION_ADDRESSU;
  AddressV = EMISSION_ADDRESSV;
  AddressW = EMISSION_ADDRESSW;
  MinFilter = EMISSION_MIN_FILTER;
  MagFilter = EMISSION_MAG_FILTER;
  MipFilter = EMISSION_MIP_FILTER;
  BorderColor = EMISSION_BORDER_COLOR;
  MaxMipLevel = EMISSION_MAX_MIP_LEVEL;
  MipMapLodBias = EMISSION_MIP_MAP_LOD_BIAS;
  Texture = (EmissionTexture);
};
texture ReflectiveTexture : REFLECTIVE_TEXTURE;
sampler ReflectiveSampler = sampler_state
{
    AddressU = REFLECTIVE_ADDRESSU;
    AddressV = REFLECTIVE_ADDRESSV;
    AddressW = REFLECTIVE_ADDRESSW;
    MinFilter = REFLECTIVE_MIN_FILTER;
    MagFilter = REFLECTIVE_MAG_FILTER;
    MipFilter = REFLECTIVE_MIP_FILTER;
    borderColor = REFLECTIVE_BORDER_COLOR;
    MaxMipLevel = REFLECTIVE_MAX_MIP_LEVEL;
    MipMapLodBias = REFLECTIVE_MIP_MAP_LOD_BIAS;
    Texture = (ReflectiveTexture);
};
#endif
#if defined(REFLECTIVE)
float Reflectivity : REFLECTIVITY;
#endif
#if defined(TRANSPARENT_COLOR)
float4 TransparentColor : TRANSPARENT_COLOR;
#elif defined(TRANSPARENT_TEXTURE)
texture TransparentTexture : TRANSPARENT_TEXTURE;
sampler TransparentSampler = sampler_state
{
    AddressU = TRANSPARENT_ADDRESSU;
    AddressV = TRANSPARENT_ADDRESSV;
    AddressW = TRANSPARENT_ADDRESSW;
    MinFilter = TRANSPARENT_MIN_FILTER;
    MagFilter = TRANSPARENT_MAG_FILTER;
    MipFilter = TRANSPARENT_MIP_FILTER;
    borderColor = TRANSPARENT_BORDER_COLOR;
    MaxMipLevel = TRANSPARENT_MAX_MIP_LEVEL;
    MipMapLodBias = TRANSPARENT_MIP_MAP_LOD_BIAS;
    Texture = (TransparentTexture);
};
#endif
#if defined(TRANSPARENT)
float Transparency : TRANSPARENCY;
#endif
#if defined(AMBIENT_COLOR)
float4 AmbientColor : AMBIENT_COLOR;
#elif defined(AMBIENT_TEXTURE)
texture AmbientTexture : AMBIENT_TEXTURE;
sampler AmbientSampler = sampler_state
{
    AddressU = AMBIENT_ADDRESSU;
    AddressV = AMBIENT_ADDRESSV;
    AddressW = AMBIENT_ADDRESSW;
    MinFilter = AMBIENT_MIN_FILTER;
    MagFilter = AMBIENT_MAG_FILTER;
    MipFilter = AMBIENT_MIP_FILTER;
    borderColor = AMBIENT_BORDER_COLOR;
    MaxMipLevel = AMBIENT_MAX_MIP_LEVEL;
    MipMapLodBias = AMBIENT_MIP_MAP_LOD_BIAS;
    Texture = (AmbientTexture);
};
#endif
#if defined(DIFFUSE_COLOR)
float4 DiffuseColor : DIFFUSE_COLOR;
#elif defined(DIFFUSE_TEXTURE)
texture DiffuseTexture : DIFFUSE_TEXTURE;
sampler DiffuseSampler = sampler_state
{
    AddressU = DIFFUSE_ADDRESSU;
    AddressV = DIFFUSE_ADDRESSV;
    AddressW = DIFFUSE_ADDRESSW;
    MinFilter = DIFFUSE_MIN_FILTER;
    MagFilter = DIFFUSE_MAG_FILTER;
    MipFilter = DIFFUSE_MIP_FILTER;
    BorderColor = DIFFUSE_BORDER_COLOR;
    MaxMipLevel = DIFFUSE_MAX_MIP_LEVEL;
    MipMapLodBias = DIFFUSE_MIP_MAP_LOD_BIAS;
    Texture = (DiffuseTexture);
};
#endif
#if defined(SPECULAR_COLOR)
    float4 SpecularColor : SPECULAR_COLOR;
#elif defined(SPECULAR_TEXTURE)
    texture SpecularTexture : SPECULAR_TEXTURE;
    sampler SpecularSampler = sampler_state
{
    AddressU = SPECULAR_ADDRESSU;
    AddressV = SPECULAR_ADDRESSV;
    AddressW = SPECULAR_ADDRESSW;
    MinFilter = SPECULAR_MIN_FILTER;
    MagFilter = SPECULAR_MAG_FILTER;
    MipFilter = SPECULAR_MIP_FILTER;
    BorderColor = SPECULAR_BORDER_COLOR;
    MaxMipLevel = SPECULAR_MAX_MIP_LEVEL;
    MipMapLodBias = SPECULAR_MIP_MAP_LOD_BIAS;
    Texture = (SpecularTexture);
};
#endif
#if defined(SPECULAR)
    float Shininess : SHININESS;
#endif
#if defined(INDEX_OF_REFRACTION)
    float IndexOfRefraction;
#endif
#if defined(BUMP_TEXTURE)
    texture BumpTexture : BUMP_TEXTURE;
    sampler BumpSampler = sampler_state
{
    AddressU = BUMP_ADDRESSU;
    AddressV = BUMP_ADDRESSV;
    AddressW = BUMP_ADDRESSW;
    MinFilter = BUMP_MIN_FILTER;
    MagFilter = BUMP_MAG_FILTER;
    MipFilter = BUMP_MIP_FILTER;
    BorderColor = BUMP_BORDER_COLOR;
    MaxMipLevel = BUMP_MAX_MIP_LEVEL;
    MipMapLodBias = BUMP_MIP_MAP_LOD_BIAS;
    Texture = (BumpTexture);
};
#endif
//------------------------------------------------------------------------------
// inputs outputs
//------------------------------------------------------------------------------
struct vsIn
{
    float3 Normal : NORMAL;
}
float4 Position : POSITION;

# if defined(DIFFUSE_VERTEX)
    float4 DiffuseColor : COLOR0;
# endif

# if defined(ANIMATED)
    float4 BoneIndices : BLENDINDICES;
    float4 BoneWeights : BLENDWEIGHT;
# endif

# if defined(BUMP)
    float3 Tangent : TANGENT;
# endif

# if (TEXCOORDS_COUNT > 0)
    float2 Texcoords0 : TEXCOORD0;
# endif

# if (TEXCOORDS_COUNT > 1)
    float2 Texcoords1 : TEXCOORD1;
# endif

# if (TEXCOORDS_COUNT > 2)
    float2 Texcoords2 : TEXCOORD2;
# endif

# if (TEXCOORDS_COUNT > 3)
    float2 Texcoords3 : TEXCOORD3;
# endif

# if (TEXCOORDS_COUNT > 4)
    float2 Texcoords4 : TEXCOORD4;
# endif

# if (TEXCOORDS_COUNT > 5)
    float2 Texcoords5 : TEXCOORD5;
# endif

# if (TEXCOORDS_COUNT > 6)
    float2 Texcoords6 : TEXCOORD6;
# endif

# if (TEXCOORDS_COUNT > 7)
    float2 Texcoords7 : TEXCOORD7;
# endif

};

struct vsOutBase {
    float4 Position : POSITION;

    # if defined(DIFFUSE_TEXTURE) || defined(AMBIENT_TEXTURE)
        float4 DiffuseAmbientTexCoords : TEXCOORD0;
    # endif

    # if defined(EMISSION_TEXTURE) || defined(TRANSPARENT_TEXTURE)
        float4 EmissionTransparentTexCoords : TEXCOORD1;
    # endif

    # if defined(REFLECTIVE_TEXTURE)
        float2 ReflectiveTexCoords : TEXCOORD2;
    # endif

};

struct vsOut {
    float4 Position : POSITION;
    float3 Eye : TEXCOORD0;
    float3 Light : TEXCOORD1;
    float3 Normal : TEXCOORD2;
    float4 ShadowTexCoords : TEXCOORD3;

    # if defined(DIFFUSE_TEXTURE) || defined(TRANSPARENT_TEXTURE)
        float4 DiffuseTransparentTexCoords : TEXCOORD4;
    # endif

    # if defined(REFLECTIVE_TEXTURE) || defined(SPECULAR_TEXTURE)
float4 ReflectiveSpecularTexCoords : TEXCOORD5;
#endif
#if defined(BUMP_TEXTURE)
float2 BumpTexCoords : TEXCOORD6;
#endif
#if defined(DIFFUSE_VERTEX)
float4 DiffuseColor : TEXCOORD7;
#endif
};

//-----------------------------------------------------------------------------
// functions
//-----------------------------------------------------------------------------
#if defined(ANIMATED)
// unpack skinning transforms from vectors-as-columns form into
// vectors-as-rows form.
float3x3 _UnpackTransform3x3(float aIndex, float4 m[kSkinningMatricesSize])
{
    float i = aIndex * 3.0f;
    return float3x3(m[i+0].x, m[i+1].x, m[i+2].x,
                    m[i+0].y, m[i+1].y, m[i+2].y,
                    m[i+0].z, m[i+1].z, m[i+2].z);
}
float3x3 _GetTransform3x3(vsIn aIn, float4 m[kSkinningMatricesSize])
{
    float3x3 ret = aIn.BoneWeights.x * _UnpackTransform3x3(aIn.BoneIndices.x, m);
    ret += aIn.BoneWeights.y * _UnpackTransform3x3(aIn.BoneIndices.y, m);
    ret += aIn.BoneWeights.z * _UnpackTransform3x3(aIn.BoneIndices.z, m);
    ret += aIn.BoneWeights.w * _UnpackTransform3x3(aIn.BoneIndices.w, m);
    return ret;
}
float4x4 _UnpackTransform4x4(float aIndex, float4 m[kSkinningMatricesSize])
{
    float i = aIndex * 3.0f;
    return float4x4(m[i+0].x, m[i+1].x, m[i+2].x, 0,
                    m[i+0].y, m[i+1].y, m[i+2].y, 0,
                    m[i+0].z, m[i+1].z, m[i+2].z, 0,
                    m[i+0].w, m[i+1].w, m[i+2].w, 1);
}
float4x4 _GetTransform4x4(vsIn aIn, float4 m[kSkinningMatricesSize])
{
    float4x4 ret = aIn.BoneWeights.x * _UnpackTransform4x4(aIn.BoneIndices.x, m);
    ret += aIn.BoneWeights.y * _UnpackTransform4x4(aIn.BoneIndices.y, m);
    ret += aIn.BoneWeights.z * _UnpackTransform4x4(aIn.BoneIndices.z, m);
    ret += aIn.BoneWeights.w * _UnpackTransform4x4(aIn.BoneIndices.w, m);
    return ret;
}
#endif

float3x3 GetInverseTransposeWorldTransform(vsIn aIn)
{
    #if defined(ANIMATED)
        return mul(_GetTransform3x3(aIn, SkinningTransforms),
                   (float3x3)InverseTransposeWorldTransform);
    #else
        return (float3x3)GetInverseTransposeWorldTransform;
    #endif
}

float4x4 GetWorldTransform(vsIn aIn)
{
    # if defined(ANIMATED)
return mul(_GetTransform4x4(aIn, SkinningTransforms), WorldTransform);
#endif
}

#if defined(BUMP)
float3x3 GetTangentToWorldTransform(float3 aNormal, float3 aTangent, float3x3 aITWorldTransform)
{
float3 binormal = cross(aNormal, aTangent);

float3 t = normalize(mul(aTangent, aITWorldTransform)); // row 0
float3 b = normalize(mul(binormal, aITWorldTransform)); // row 1
float3 n = normalize(mul(aNormal, aITWorldTransform)); // row 2

return float3x3(t.x, b.x, n.x,
                t.y, b.y, n.y,
                t.z, b.z, n.z);
}

float3x3 GetWorldToTangentTransform(float3 aNormal, float3 aTangent, float3x3 aITWorldTransform)
{
float3x3 ret;
float3 binormal = cross(aNormal, aTangent);

ret[0] = normalize(mul(aTangent, aITWorldTransform)); // row 0
ret[1] = normalize(mul(binormal, aITWorldTransform)); // row 1
ret[2] = normalize(mul(aNormal, aITWorldTransform)); // row 2

return ret;
}
#endif

void LightTerms(vsIn aIn, float4 aWorld,
uniform bool abDirectional,
uniform bool abPoint,
uniform bool abSpot,
uniform bool abShadow,
out float3 arEye,
out float3 arLight,
out float3 arNormal,
out float4 arShadowTexCoords)
{
if (abShadow) { arShadowTexCoords = mul(aWorld, ShadowTransform); }
else { arShadowTexCoords = float4(0, 0, 0, 0); }

float3 eyePos = float3(InverseViewTransform._41,
InverseViewTransform._42,
InverseViewTransform._43);

if (abSpot || abPoint) { arLight = LightPositionOrDirection - aWorld.xyz; }
else { arLight = -LightPositionOrDirection; }

float3x3 itWorld = GetInverseTransposeWorldTransform(aIn);
arEye = eyePos - aWorld.xyz;

# if defined(BUMP)
float3x3 worldToTangent = GetWorldToTangentTransform(
    aIn.Normal, aIn.Tangent, itWorld);
arEye = mul(arEye, worldToTangent);
arLight = mul(arLight, worldToTangent);
arNormal = float3(0, 0, 0); // remove the output normal if there is a bump-map.
#else
arNormal = mul(aIn.Normal, itWorld);
#endif
float Shadow(float4 aShadowTexCoords, float aPixelDepth, uniform bool abFiltered) {
    float ret = 0.0f;
    float offset = (kShadowDelta * aShadowTexCoords.w);
    float noffset = -offset;
    if (abFiltered) {
        float4 shadowDepths;
        shadowDepths.x = tex2Dproj(ShadowSampler, aShadowTexCoords + float4(noffset, noffset, 0, 0)).x;
        shadowDepths.y = tex2Dproj(ShadowSampler, aShadowTexCoords + float4(offset, noffset, 0, 0)).x;
        shadowDepths.z = tex2Dproj(ShadowSampler, aShadowTexCoords + float4(noffset, offset, 0, 0)).x;
        shadowDepths.w = tex2Dproj(ShadowSampler, aShadowTexCoords + float4(offset, offset, 0, 0)).x;
        float4 c = (aPixelDepth <= shadowDepths);
        ret = (c.x + c.y + c.z + c.w) * 0.25;
    } else {
        float shadowDepth = tex2Dproj(ShadowSampler, aShadowTexCoords).x;
        if (aPixelDepth <= shadowDepth) { ret = 1.0f; }
    }
    return ret;
}

float4 GammaColor(float4 aColor) {
    float4 ret;
    ret.r = (aColor.r >= kLooseTolerance) ? pow(aColor.r, Gamma) : 0.0;
    ret.g = (aColor.g >= kLooseTolerance) ? pow(aColor.g, Gamma) : 0.0;
    ret.b = (aColor.b >= kLooseTolerance) ? pow(aColor.b, Gamma) : 0.0;
    ret.a = aColor.a;
    return ret;
}

float4 GammaTextureRead(sampler aSampler, float2 aTexCoords) {
    float4 col = tex2D(aSampler, aTexCoords);
    return float4(pow(col.rgb, Gamma), col.a);
}

// vertex shaders

// base vertex shader, used during unlit base pass (affected by ambient, emission)
vsOutBase VertexBase(vsIn aIn) {
    vsOutBase output;
    float4 world = mul(aIn.Position, GetWorldTransform(aIn));
    output.Position = mul(world, ViewProjectionTransform);
    # if defined(DIFFUSE_TEXTURE)
        output.DiffuseAmbientTexCoords.xy = aIn.DIFFUSE_TEXCOORDS;
    # else if defined(AMBIENT_TEXTURE)
        output.DiffuseAmbientTexCoords.xy = float2(0, 0);
    # endif
    # if defined(AMBIENT_TEXTURE)
        output.DiffuseAmbientTexCoords.zw = aIn.AMBIENT_TEXCOORDS;
    # endif
}
// main vertex shading, used when lighting is applied.
vsOut Vertex(vsIn aIn, uniform bool abDirectional, uniform bool abPoint, uniform bool abSpot, uniform bool abShadow)
{
    float4 world = mul(aIn.Position, GetWorldTransform(aIn));
    vsOut output;
    LightTerms(aIn, world, abDirectional, abPoint, abSpot, abShadow, output.Eye, output.Light, output.Normal, output.ShadowTexCoords);
    output.Position = mul(world, ViewProjectionTransform);
    # if defined(DIFFUSE_TEXTURE)
        output.DiffuseTransparentTexCoords.xy = aIn.DIFFUSE_TEXCOORDS;
    # endif  
    # if defined(TRANSPARENT_TEXTURE)
        output.DiffuseTransparentTexCoords.xy = aIn.TRANSPARENT_TEXCOORDS;
    # endif  
    # if defined(TRANSPARENT_TEXTURE)
        output.DiffuseTransparentTexCoords.zw = aIn.TRANSPARENT_TEXCOORDS;
    # elif defined(DIFFUSE_TEXTURE)
        output.DiffuseTransparentTexCoords.zw = float2(0, 0);
    # endif
    # if defined(REFLECTIVE_TEXTURE)
        output.ReflectiveSpecularTexCoords.xy = aIn.REFLECTIVE_TEXCOORDS;
    # elif defined(SPECULAR_TEXTURE)
        output.ReflectiveSpecularTexCoords.xy = float2(0, 0);
    # endif
    # if defined(SPECULAR_TEXTURE)
        output.ReflectiveSpecularTexCoords.zw = aIn.SPECULAR_TEXCOORDS;
    # elif defined(REFLECTIVE_TEXTURE)
        output.ReflectiveSpecularTexCoords.zw = float2(0, 0);
    # endif
    # if defined(BUMP_TEXTURE)
        output.BumpTexCoords = aIn.BUMP_TEXCOORDS;
    # endif
    # if defined (DIFFUSE_VERTEX)
        output.DiffuseColor = aIn.DiffuseColor;
    # endif
    return output;
}

//-----------------------------------------------------------------------------
// fragment shaders
//-----------------------------------------------------------------------------
float4 FragmentBase(vsOutBase aIn) : COLOR
{
    float alpha = 1.0f;

    //----- Get diffuse color.
    #if defined(DIFFUSE_COLOR)
        float3 diffuse = GammaColor(DiffuseColor).rgb;
    #elif defined(DIFFUSE_TEXTURE)
        float3 diffuse = GammaTextureRead(DiffuseSampler, 
                        aIn.DiffuseAmbientTexCoords.xy).rgb;
    #endif

    //----- Get ambient color.
    #if defined(AMBIENT_COLOR)
        float3 ambient = GammaColor(AmbientColor).rgb;
    #elif defined(AMBIENT_TEXTURE)
        float3 ambient = GammaTextureRead(AmbientSampler, 
                        aIn.DiffuseAmbientTexCoords.zw).rgb;
    #endif

    //----- Get emission color.
    #if defined(EMISSION_COLOR)
        float3 emission = GammaColor(EmissionColor).rgb;
    #elif defined(EMISSION_TEXTURE)
        float3 emission = GammaTextureRead(EmissionSampler, 
                        aIn.EmissionTransparentTexCoords.xy).rgb;
    #endif

    //----- Get transparent color and calculate alpha.
    //----- See COLLADA spec for what all this means.
    #if defined(TRANSPARENT_COLOR)
        float4 transparent = TransparentColor;
    #elif defined(TRANSPARENT_TEXTURE)
        float4 transparent = tex2D(TransparentSampler, 
                        aIn.EmissionTransparentTexCoords.zw);
    #endif
    #if defined(TRANSPARENT)
        #if defined(ALPHA_ONE)
            alpha = transparent.a * Transparency;
        #elif defined(RGB_ZERO)
            alpha = 1.0f - (((0.212671 * transparent.r) + 
                            (0.715160 * transparent.g) + 
                            (0.072169 * transparent.b)) * Transparency);
        #endif
    #endif

    //----- Get reflective color and combine with diffuse.
    #if defined(REFLECTIVE_COLOR)
        float3 reflective = GammaColor(ReflectiveColor).rgb;
    #elif defined(REFLECTIVE_TEXTURE)
        float3 reflective = GammaTextureRead(ReflectiveSampler, 
                        aIn.ReflectiveTexCoords).rgb;
    #endif
    #if defined(REFLECTIVE_COLOR) || defined(REFLECTIVE_TEXTURE)
        diffuse = lerp(diffuse, reflective, Reflectivity);
    #else
        diffuse = reflective;
    #endif

    //----- Calculate output color.
    float3 ret = float3(0, 0, 0);
    #if (defined(DIFFUSE) || defined(REFLECTIVE)) && defined(AMBIENT)
        ret += (diffuse * ambient);
    #endif

    #if defined(EMISSION)
float4 Fragment(vsOut aIn,
    uniform bool abPoint,
    uniform bool abSpot,
    uniform bool abShadow,
    uniform bool abShadowFiltered) : COLOR
{
    float alpha = 1.0f;

    //---- Get diffuse color.
    # if defined(DIFFUSE_COLOR)
        float3 diffuse = GammaColor(DiffuseColor).rgb;
    # elif defined(DIFFUSE_TEXTURE)
        float3 diffuse = GammaTextureRead(DiffuseSampler,
            aIn.DiffuseTransparentTexCoords.xy);
    # elif defined(DIFFUSE_VERTEX)
        float3 diffuse = GammaColor(aIn.DiffuseColor).rgb;
    # endif

    //---- Get transparent color and calculate alpha.
    # if defined(TRANSPARENT_COLOR)
        float4 transparent = TransparentColor;
    # elif defined(TRANSPARENT_TEXTURE)
        float4 transparent = tex2D(TransparentSampler,
            aIn.DiffuseTransparentTexCoords.zw);
    # endif
    # if defined(TRANSPARENT)
        # if defined(ALPHA_ONE)
            alpha = transparent.a * Transparency;
        # elif defined(RGB_ZERO)
            alpha = 1.0f - (((0.212671 * transparent.r) +
                (0.715160 * transparent.g) +
                (0.072169 * transparent.b)) * Transparency);
        # endif
    # endif

    //---- Get reflective color and combine with diffuse.
    # if defined(REFLECTIVE_COLOR)
        float3 reflective = GammaColor(ReflectiveColor).rgb;
    # elif defined(REFLECTIVE_TEXTURE)
        float3 reflective = GammaTextureRead(ReflectiveSampler,
            aIn.ReflectiveSpecularTexCoords.xy);
    # endif
    # if defined(REFLECTIVE)
        # if defined(DIFFUSE)
            diffuse = lerp(diffuse, reflective, Reflectivity);
        # else
            diffuse = reflective;
        # endif
    # endif

    //---- Get specular color
    # if defined(SPECULAR_COLOR)
        float3 specular = GammaColor(SpecularColor).rgb;
    # elif defined(SPECULAR_TEXTURE)
        float3 specular = GammaTextureRead(SpecularSampler,
            aIn.ReflectiveSpecularTexCoords.zw);
    # endif
//---- Calculate light vectors if necessary.
#if defined(DIFFUSE) || defined(REFLECTIVE) || defined(SPECULAR)
    float3 lv = normalize(aIn.Light.xyz);
#else
    float4 bump = tex2D(BumpSampler, aIn.BumpTexCoords.xy);
    float3 nv = normalize((2.0 * bump.rgb) - 1.0);
#else
    float3 nv = normalize(aIn.Normal.xyz);
#endif
//---- Calculate eye vector if necessary and light component vector if necessary.
#if defined(SPECULAR)
    float3 ev = normalize(aIn.Eye.xyz);
    float ndotl = dot(nv, lv);
#else
    float4 l = float4(1, max(dot(nv, lv), 0.0f), 0, 1);
#endif
float3 ret = float3(0, 0, 0);
#if defined(DIFFUSE) || defined(REFLECTIVE) || defined(SPECULAR)
//---- Calculate attenuation if spot or point light.
    float distance = 0.0f;
    float att = 1.0f;
    if (abSpot || abPoint)
    {
        distance = length(aIn.Light.xyz);
        att = 1.0f / (LightAttenuation.x + (LightAttenuation.y * distance) + (LightAttenuation.z * distance * distance));
    }
#endif
//---- Calculate diffuse contribution.
#if defined(DIFFUSE)
    ret += LightDiffuse * diffuse * l.y;
#endif
//---- Calculate specular contribution.
#if defined(SPECULAR)
    ret += LightSpecular * specular * l.z;
#endif
#if defined(DIFFUSE) || defined(SPECULAR)
    ret *= att;
#endif
//---- "Premultiplied alpha - see http://home.comcast.net/~tom_forsyth/blog.wiki.html# enlistedalpha"
#if defined(TRANSPARENT)
    ret *= alpha;
#endif
# if defined(DIFFUSE) || defined(REFLECTIVE) || defined(SPECULAR)
//---- If a spot light, calculate spot contribution.
if (abSpot)
{
    float spotDot = -dot(lv, SpotDirection);
    float spot = pow(max(spotDot, 0.0f), max(SpotFalloffExponent, kLooseTolerance));
    if (spotDot < SpotFalloffCosAngle) { spot = 0.0f; }
    ret *= spot;
}

//---- If a shadow casting light, calculate shadow contribution.
if (abShadow)
{
    float pixelDepth = ((distance / ShadowFarDepth) - kShadowDepthBias);
    ret *= Shadow(aIn.ShadowTexCoords, pixelDepth, abShadowFiltered);
}
#endif
return float4(ret, alpha);
}

technique siat_RenderBase
{
// if defined(TRANSPARENT_TEXTURE)
pass Pass0
{
    AlphaBlendEnable = false;
    AlphaFunc = GreaterEqual;
    AlphaRef = OPAQUE_OF_TRANSPARENCY;
    AlphaTestEnable = true;
    ColorWriteEnable = RED|GREEN|BLUE|ALPHA;
    # if defined(TRANSPARENT_TEXTURE_1_BIT)
    CullMode = BACK_FACE_CULLING;
    # else
    CullMode = None;
    # endif
    FillMode = Solid;
    ZWriteEnable = true;
    VertexShader = compile vs_2_0 VertexBase();
    PixelShader = compile ps_2_0 FragmentBase();
}

// if !defined(TRANSPARENT_TEXTURE_1_BIT)
pass Pass1
{
    AlphaBlendEnable = true;
    AlphaFunc = Less;
    AlphaRef = OPAQUE_OF_TRANSPARENCY;
    AlphaTestEnable = true;
    ColorWriteEnable = RED|GREEN|BLUE|ALPHA;
    CullMode = FRONT_FACE_CULLING;
    DestBlend = InvSrcAlpha;
    FillMode = Solid;
    SrcBlend = One;
    ZWriteEnable = false;
    VertexShader = compile vs_2_0 VertexBase();
    PixelShader = compile ps_2_0 FragmentBase();
}

pass Pass2
{
    AlphaBlendEnable = true;
    AlphaFunc = Less;
    AlphaRef = OPAQUE_OF_TRANSPARENCY;
    AlphaTestEnable = true;
}
ColorWriteEnable = RED|GREEN|BLUE|ALPHA;
CullMode = BACK_FACE_CULLING;
DestBlend = InvSrcAlpha;
FillMode = Solid;
SrcBlend = One;
ZWriteEnable = false;

VertexShader = compile vs_2_0 VertexBase();
PixelShader = compile ps_2_0 FragmentBase();

# endif

#elif defined (TRANSPARENT)

pass Pass0
{
    AlphaBlendEnable = true;
    AlphaTestEnable = false;
    ColorWriteEnable = RED|GREEN|BLUE|ALPHA;
    CullMode = FRONT_FACE_CULLING;
    DestBlend = InvSrcAlpha;
    FillMode = Solid;
    SrcBlend = One;
    ZWriteEnable = false;

    VertexShader = compile vs_2_0 VertexBase();
    PixelShader = compile ps_2_0 FragmentBase();
}

pass Pass1
{
    AlphaBlendEnable = true;
    AlphaTestEnable = false;
    ColorWriteEnable = RED|GREEN|BLUE|ALPHA;
    CullMode = BACK_FACE_CULLING;
    DestBlend = InvSrcAlpha;
    FillMode = Solid;
    SrcBlend = One;
    ZWriteEnable = false;

    VertexShader = compile vs_2_0 VertexBase();
    PixelShader = compile ps_2_0 FragmentBase();
}

#else

pass Pass0
{
    AlphaBlendEnable = false;
    AlphaTestEnable = false;
    ColorWriteEnable = RED|GREEN|BLUE|ALPHA;
    CullMode = BACK_FACE_CULLING;
    FillMode = Solid;
    ZWriteEnable = true;

    VertexShader = compile vs_2_0 VertexBase();
    PixelShader = compile ps_2_0 FragmentBase();
}

#endif

technique sail_RenderDirectionalLight
{
    # if defined(TRANSPARENT_TEXTURE)
    pass Pass0
    {
        AlphaBlendEnable = true;
        AlphaFunc = GreaterEqual;
        AlphaRef = OPAQUE_OF_TRANSPARENCY;
        AlphaTestEnable = true;
        ColorWriteEnable = RED|GREEN|BLUE|ALPHA;
        # if defined(TRANSPARENT_TEXTURE_1_BIT)

        VertexShader = compile vs_2_0 VertexBase();
        PixelShader = compile ps_2_0 FragmentBase();
        # endif
        # endif

        VertexShader = compile vs_2_0 VertexBase();
        PixelShader = compile ps_2_0 FragmentBase();
    }

    pass Pass1
    {
        AlphaBlendEnable = true;
        AlphaTestEnable = false;
        ColorWriteEnable = RED|GREEN|BLUE|ALPHA;
        CullMode = BACK_FACE_CULLING;
        DestBlend = InvSrcAlpha;
        FillMode = Solid;
        SrcBlend = One;
        ZWriteEnable = false;

        VertexShader = compile vs_2_0 VertexBase();
        PixelShader = compile ps_2_0 FragmentBase();
    }

    # else

    pass Pass0
    {
        AlphaBlendEnable = false;
        AlphaTestEnable = false;
        ColorWriteEnable = RED|GREEN|BLUE|ALPHA;
        CullMode = BACK_FACE_CULLING;
        FillMode = Solid;
        ZWriteEnable = true;

        VertexShader = compile vs_2_0 VertexBase();
        PixelShader = compile ps_2_0 FragmentBase();
    }

    # endif

    }

CullMode = BACK_FACE_CULLING;
# else
CullMode = None;
# endif
DestBlend = One;
FillMode = Solid;
SrcBlend = One;
ZWriteEnable = true;

VertexShader = compile vs_2_0 VertexBase();
PixelShader = compile ps_2_0 FragmentBase();
}

# if !defined(TRANSPARENT_TEXTURE_1_BIT)
pass Pass1
{
    AlphaBlendEnable = true;
    AlphaFunc = Less;
    AlphaRef = OPAQUE_OF_TRANSPARENCY;
    AlphaTestEnable = true;
    ColorWriteEnable = RED|GREEN|BLUE|ALPHA;
    CullMode = FRONT_FACE_CULLING;
    DestBlend = One;
    FillMode = Solid;
    SrcBlend = One;
    ZWriteEnable = false;

    VertexShader = compile vs_2_0 VertexBase();
    PixelShader = compile ps_2_0 FragmentBase();
}

class Pass2
{
    AlphaBlendEnable = true;
    AlphaFunc = Less;
    AlphaRef = OPAQUE_OF_TRANSPARENCY;
    AlphaTestEnable = true;
    ColorWriteEnable = RED|GREEN|BLUE|ALPHA;
    CullMode = BACK_FACE_CULLING;
    DestBlend = One;
    FillMode = Solid;
    SrcBlend = One;
    ZWriteEnable = false;

    VertexShader = compile vs_2_0 VertexBase();
    PixelShader = compile ps_2_0 FragmentBase();
}
# endif

# endif defined (TRANSPARENT)
pass Pass0
{
    AlphaBlendEnable = true;
    AlphaTestEnable = false;
    ColorWriteEnable = RED|GREEN|BLUE|ALPHA;
    CullMode = FRONT_FACE_CULLING;
    DestBlend = One;
    FillMode = Solid;
    SrcBlend = One;
    ZWriteEnable = false;

    VertexShader = compile vs_2_0 VertexBase();
    PixelShader = compile ps_2_0 FragmentBase();
}

class Pass1
{
    AlphaBlendEnable = true;
    AlphaTestEnable = false;
ColorWriteEnable = RED|GREEN|BLUE|ALPHA;
CullMode = BACK_FACE_CULLING;
DestBlend = One;
FillMode = Solid;
SrcBlend = One;
ZWriteEnable = false;

VertexShader = compile vs_2_0 VertexBase();
PixelShader = compile ps_2_0 FragmentBase();
}

# else

pass Pass0
{
    AlphaBlendEnable = true;
    AlphaTestEnable = false;
    ColorWriteEnable = RED|GREEN|BLUE|ALPHA;
    CullMode = BACK_FACE_CULLING;
    DestBlend = One;
    FillMode = Solid;
    SrcBlend = One;
    ZWriteEnable = true;

    VertexShader = compile vs_2_0 Vertex(true, false, false, false);
    PixelShader = compile ps_2_0 Fragment(false, false, false, false);
}

# endif

sail_RenderPointLight, sail_RenderSpotLight, sail_RenderSpotLightShadow_Unfiltered, and sail_RenderSpotLightShadow_Filtered are also techniques. These have the exact same form as sail_RenderDirectionalLight with the appropriate changes made to the parameters of VertexShader = compile and PixelShader = compile. For example, sail_RenderPointLight is the following for these two lines:

VertexShader = compile vs_2_0 Vertex(false, true, false, false);
PixelShader = compile ps_2_0 Fragment(true, false, false, false);
Appendix C: Study Survey

1. What is your age?  
2. What is your sex?  
3. How long have you been a lighting expert?  
4. Which term best describes your area of expertise (check all that apply)?
   - Architectural
   - Concert
   - Film
   - Interactive Art
   - Multimedia
   - Rock and Roll
   - Stage
   - Television
   - Video game
   - Other  

5. Which term best describes your activity level (check all that apply)?
   - Academic
   - Amateur
   - Production
   - Resident
   - Other  

6. Anything else you wish to share about your lighting expertise?

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________
INVESTIGATOR:
Let’s get started. First question for you, before we get back to this [referring to the software system].

10_26_01:
Of course.

INVESTIGATOR:
In addition to what you put on the survey, describe briefly what your process would be as a lighting expert, in your most recent field or in a field that was particularly memorable for you.

10_26_01:
----

INVESTIGATOR:
In general, what did you think about the system and the way it was lighting the character?

10_26_01:
I felt like there were [pause]. I’ll tell you what it seemed like and you can tell me whether or not I’m right or wrong. It seemed like there were two things it was trying to do here.

One was trying to find a way to really make the character fit into the scene and really match the lighting of the scene and follow along with how the scene was created.

And then the second was trying to sort of serve the two masters of trying to make the character fit ok into the scene but also have the lighting and the sort of mood lighting on the character that you want for the emotional impact that you’re trying to get.

I don’t know if that’s the case but that’s kind of what it appeared like to me.
INVESTIGATOR:
Go into a little bit more about what you mean by “mood lighting”.

10_26_01:
I mean that there are very different ways that you light people if you’re trying to get someone observing that person to see certain emotions in them. Very, very harsh down light makes a person look very small, a very harsh, focused up light can be a sort of horror kind of look. If you light someone from the side they’ll pop out. A lot of dance lighting is done from very low down lights on the side of the stage because it pops the dancers out from the background and gives them dimensionality and all these different things I think traditionally have been hard to do in video games and in computer lighting because the lights in the scene are pretty much the same as the lights that are used on the character for the most part. I don’t really know what you’re trying to do but it seems like some of what you’re trying to do is sort of do both of those things. Sort of pick how do you want the lights to look on the character, what do you want to read on the character as a viewer, and then try and do that in concert with or maybe even regardless of the lighting in the scene.

INVESTIGATOR:
On that note, what do you think the role of the image was then?

10_26_01:
The role of the image seemed like a quick, top down sort of chart of kind of where the key light was coming from. I didn’t get much feel for the fill light in this. And it would be interesting to see, if that’s right [referring to 10_26_01’s idea of the role of the image] and it is sort of a top down view of that it’d be interesting to see the fill light as a gradient and the key light as like some sort of a cone so that you can sort of see both of them. I’m not entirely sure that’s what it was but definitely when I was in the attached setting it was very obvious that I was changing directly those lights on the player [referring to the character in the software system] with Q, W, and E [referring to keyboard keys that were used to select between three images in the software system].

INVESTIGATOR:
Would you describe the lighting as satisfactory in the environment?
10_26_01:
As I said, the fact that the lighting seemed non-diagetic to me [referring to a comment made by the 10_26_01 while the 10_26_01 was interacting with the system before the interview portion of the session] was a little bit confusing.

INVESTIGATOR:
Could you explain what you mean by “non-diagetic”?

10_26_01:
Sure. And diagetic might not even be used frequently with lighting it’s used with sound all the time and film and that means that there’s a difference between when you have the soundtrack playing in the background and when you have the sound seem to be coming from the radio in the car that the person is sitting in. That’s a diagetic sound, a sound that is actually emanating from the environment in some way rather than sound. So, I guess the real term, the stage term would be a practical light. You can have a lamp sitting on stage that is actually casting light or you can have a lamp sitting on stage and have a light that is up in the rafters that is trying to simulate what that light would look like coming from the lamp.

And so the one thing that I found not satisfying about the lighting in the scene as I mentioned [referring to a comment made by the 10_26_01 while the 10_26_01 was interacting with the system before the interview portion of the session] was the two lamps didn’t seem to be casting the light. They seemed to be representing light sources but they were representing lighting sources that were outside the room. That was a little by confusing to me.

INVESTIGATOR:
By the two lamps, you mean the lights indicated by the arrows?

10_26_01:
No, actually what I mean is the physical lamps sitting on the chest of drawers here [10_26_01 points to a lamp in the 3D scene that is located on a chest of drawers in the center room of the environment] and the one sitting on the table in the third room.

INVESTIGATOR:
So would you describe this more as a problem with the preset environment lighting?
Yes, absolutely. That being said though, I feel like the lighting on the character relative to the lighting in the room, in particular with the “auto” and “natural” settings, was very nice. With the natural settings, the character fit into the room very, very well and with the auto settings even though watching where the key light and fill light were coming from they didn’t seem to make sense in the scene it still felt like good lighting on the character to me.

**INVESTIGATOR:**

As a designer, do you think you would have done anything differently? It seems like you probably would have motivated the lights differently but in addition to that?

**10_26_01:**

[Pause] I think I would have just explained what was going on more. I mentioned the sort of possibility for the lighting images up here [10_26_01 points to the image displayed in the upper right corner of the software system], to sort of clarify that. Right now it feels very obtuse to me.

**INVESTIGATOR:**

The lighting itself you mean?

**10_26_01:**

Just what it is…[Pause] I believe there is something there that you are doing that is different and new and interesting and I’m having trouble picking that out from the rest of what’s going on here. And understanding exactly what it was that I was affecting. So, I felt like if I had more of an understanding of the system going in or if I had more of the system sort of described to me as I was starting out that would help me understand that much better.

I think, for instance, 1, 2, 3 to select light cue here [10_26_01 points to the software system, indicating the help text that describes the ability to press keys 1, 2, or 3 to select a preset lighting configuration for the room around the character] as soon as I select the lighting cue I’m like, I get it, I know what I’m doing. I can see real demonstrable changes, right? [10_26_01 is referring to the fact that when the keys 1, 2, or 3 are pressed, the lighting in the room immediately cross fades from the current preset configuration to the preset configuration attached to the pressed key]. But,
“desired character lighting” or “auto”, “natural”, and “attached” lighting, I don’t really understand what those mean. I almost feel like I’m spending more time trying to figure out what’s the meaning of those different things and less time trying to figure actually how to use this as a tool. And I assume what you’re doing is you’re creating a tool to do something. I don’t really know exactly.

INVESTIGATOR:

Any other general comments that I haven’t covered?

10_26_01:

I felt like the character moved really slowly. And I wanted some way to be able to, I don’t know, hold down shift or something, to get her to move a little bit faster ‘cause if I was looking at how she moved through the room…

INVESTIGATOR:

Well, she’s a very elegant woman, so…

10_26_01:

Yes, of course. If I look at just how she moves through the room, this is a good speed. But, if what I want to do is, oh, I want to see myself over in the other room, I want her to move much more quickly because it’s just excruciating trying to get somewhere in particular.

I think that’s most of the comments that I have [pause].

You have a very simple room with simple texture maps, and the character as well, but it doesn’t really seem like there’s like, a lot of normal mapping, no normal mapping going on…

INVESTIGATOR:

There’s no bump mapping, no.

10_26_01:

I mean it’s hard to [pause] There’s something interesting going on here but it’s hard for me to put my finger on it. And I can’t say, “Oh my god, this looks fantastic”, because I’m used to seeing shaders and normal mapping and all that crazy kind of stuff. I would love to hear more about what’s actually going on here and what’s it doing in the scene. And I think once I have a better understanding I’ll be better to sort of react to it.
INVESTIGATOR:

Thank you.

10_26_02

INVESTIGATOR:

Before we actually start talking about this lighting system in addition to what you put on the survey could you describe a typical process for you as a lighting designer?

10_26_02:

----

INVESTIGATOR:

What are your thoughts in general about this system [referring to the automated lighting system of the experiment]?  

10_26_02:

Part of [pause]. I wasn’t super clear on the difference between Q and E…

INVESTIGATOR:

What are “Q” and “E”?

10_26_02:

“Q” being the desired character lighting. what comes here…

INVESTIGATOR:

The image?

10_26_02:

Ya, the image. And so I didn’t exactly see a big difference on the character herself. I wasn’t really sure if that was because I’m in a lit area, and so I thought maybe if I were in different room lighting that I would see different reactions. I didn’t get any feedback as to whether it was changing so I wasn’t sure if they were working.

INVESTIGATOR:

Did you try attached mode?

10_26_02:

Attached mode?
INVESTIGATOR:
Which is “D” [referring to a key on the keyboard which selects the attached lighting mode].

10_26_02:
Yes, I did try that. I’m going to assume that lighting is attached to the character themselves. That’s what I had assumed at least. Let me get into a different lighting… [10_26_02 changes lighting mode and environment cue in the software system]. 2 was really hard for me to see [referring to an environment lighting cue that represents nighttime] because it was so dark. I’m not sure if attached… [pause]. So sometimes if I came out of a different lighting cue, I don’t know if it’s just a bug, that the attached lighting, it stays with the character regardless of how the room changes. That was my impression of it.

INVESTIGATOR:
That pretty much describes attached mode, ya.

10_26_02:
Ok. I assumed auto was based on the lighting itself in the world and it uses point lights, I thought. To light the character themselves. And natural I assumed was based off the actual, environment lights and the lights pointed at the character but I was really sure. I wasn’t sure if it was just…[pause] It didn’t seem like there was ambient lighting right on the character, it seemed like it was based on what was hitting the character.

INVESTIGATOR:
Would you describe what you’re seeing in terms of the character lighting as satisfactory?

10_26_02:
I actually thought it depended on, depending on which room I was in and what lighting I was in, I liked “auto” vs. “natural” vs. “attached” lighting depending on what type of lighting it was.

INVESTIGATOR:
Could you give me an example?
10_26_02:
When I was walking around in the backroom…

INVESTIGATOR:
So this is the room with the fireplace?

10_26_02:
Yes. The room with the fireplace had kind of an orange tone to it and I liked the natural lighting in this one. The orange color is…[pause]. I don’t know why…[pause]. Maybe it was the wooden motif or something, but it felt like, like the light shouldn’t have bounced off of stuff. So she had more blacker, deeper details around.

In this room, the second room with the light [referring to the actual lamp located on a chest of drawers in the second room], I think I remember liking “auto” best because I felt like it was the right amount of lighting in her skin.

INVESTIGATOR:
Compared to natural?

10_26_02:
Compared to natural. Ya, for some reason natural felt like it was taking in too much of the lighting from the other room. I felt like the rooms needed to be more [pause]. The lighting was based on the small room rather than like what room is coming from the opposite rooms because they had different style of room altogether and so I felt like it should come off of this [indicating the second room in the demo] in general. When I came into this room I totally forgot about the lighting in the back room [referring to the room with the fireplace] so I expected the lighting to only be affected by the close proximity things. And then in the third, no, first lighting [to environment lighting cue 1], I really liked “auto” in the first lighting cue…

INVESTIGATOR:
In the center room?

10_26_02:
Yes, in the center room. I thought the shadows off of her face were about right.
INVESTIGATOR:
You mentioned the image in the upper right corner, could you go into a little bit more
detail about what you think the role of the image is?

10_26_02:
I actually don’t really know…

INVESTIGATOR:
Do you have any sort of inkling or would you venture a guess?

10_26_02:
I assume it’s where the lighting is coming from so, this [indicating the image in the
software system] would be where the light is coming from the top or directionality as
opposed to [long pause, while changing the image in the demo]. Ya, I’m not quite
sure. If I had to venture a guess I would probably say the intensity and the
directionality of the character lighting. But I didn’t see it on her so I wasn’t sure if I
was right or not.

INVESTIGATOR:
As a designer, what would you have done differently, specifically, compared to what
the system was doing?

10_26_02:
Design is all about authorship. It’s not really about the end result. Designers tend to
look at how easy is it for me to change the lighting direction on her, is it dynamic as
opposed to do I have to bake the lighting in?

INVESTIGATOR:
You mean by authorship, in terms of…?

10_26_02:
If I were to go into a level editor, like how easy is it for me to make changes to the
lighting.

INVESTIGATOR:
So you mean you’re always thinking about the constraints first?
10_26_02:
Yes. How the lighting looks is more of an art question, I’m not sure if I’m the best person to answer that. But, it reacts, the lighting is reacting pretty nicely to the point lights in the world, it feels like.

INVESTIGATOR:
You mean in auto mode, specifically?

10_26_02:
Ya. In terms of control, it would be pretty cool if, as a designer, I could author certain areas to be, like, this area you want it to be more spooky, so it’s like you can kind of put triggers in the plane and kind of say, ok, this area I want it to be this kind of lighting and so then the designer can come in and do some kind of change over time that says lighting switches from this lighting scheme to another lighting scheme. And really it’s going to depend on what you want to tell about the characters, the story or whatever.

But ya, it’s authorship at that point. I think it’d be nice, and this is going to sound really cheesy because people don’t really have shadows but, this world feels really, really lit and I’m not seeing her shadow very well…

INVESTIGATOR:
She only shadows in natural mode.

10_26_02:
Ok.

INVESTIGATOR:
And then only under specific lights.

10_26_02:
And I think that might be an “attached” lighting issue. I always feel like players feel pretty flat when they don’t have shadows. And most of these rooms feel really, really over lit compared to her. ‘Cause she’s got …

INVESTIGATOR:
So you would say the rooms seem brighter than she is?
Ya. She’s reacting really nicely to the light but I don’t think that it reflects it in the world very well.

INVESTIGATOR:
Is that specifically in a specific mode?

It’s more the mismatch between the way she’s lit [pause]. When I expect her to have black on the back and the lighting kind of striking her like this I expect it to be darker on the back end. I’m assuming that’s because it’s vertex lighting and not many…

INVESTIGATOR:
But specifically, that seems to be the case in any mode?

Mm-hmm. She tends to be lit pretty well. But the world isn’t really matching the way she’s being lit. But I was looking pretty closely at just her lighting rather than the world so I didn’t really notice until now.

INVESTIGATOR:
Any other thoughts I haven’t covered or asked you about?

Change your controls. I expected it to be World of Warcraft controls, that’s just ‘cause of standardization.

INVESTIGATOR:
So is that WASD?

It’s a, up is…

So invert the up axis?

Actually, up and down is right. It’s left and right that feels weird. Or maybe just give your testers the choice to switch. ‘Cause it actually distracted me from looking at stuff. ‘Cause I’m going the wrong way, oh, I’m going the wrong way. Ya, World of
Warcraft is actually backwards for left and right. Up is also backwards from World of Warcraft too. And the only reason I compare to World of Warcraft is because most games have made their camera systems based on that game these days.

INVESTIGATOR:
Any other comments about the lighting itself?

10_26_02:
Looks pretty good. It would be kind of cool if I could move the lights, ‘cause I remember the TAB thing which said I could see the attached lighting. I thought I could move them, I thought that would have been kind of cool to see how it reacted if I moved them around.

INVESTIGATOR:
Thank you.

10_27_01

INVESTIGATOR:
Briefly describe your typical process as a lighting designer.

10_27_01:
Work with the tools that are there. Try to develop better ones and better pipelines. And, the process of actually lighting on the games that I’ve been on, it’s usually either a dynamic lighting scenario or kind of a mixture of pre-lit backgrounds or environments and usually the characters are often dynamically lit. So it would typically be placing a bunch of lights in a 3D package such as Maya. Generating some baked lighting. Sometimes that’s done in the package, sometimes that’s done in a pipeline. That may or may not include lightmaps. And then there’s sort of a separate pass that would be for placing dynamic lights that hopefully map to where your light sources are in the environment.

INVESTIGATOR:
Just so I’m clear, you mean that the lights that would affect dynamic objects, like a character…are usually separate from…?
Ya, characters, props, anything that usually moves around in the world. So usually, and the only reason for that and, I haven’t done lighting on the current gen games I’ve only done lighting on PS2 and Xbox titles, so that model has changed now to be mainly dynamic lighting, so that everything is dynamically lit ‘cause the hardware can kind of keep up with it. There is still some pre-lighting that’s done. The shift in the last number of years has been from doing pre-lighting on the environments and dynamic lighting on the characters to trying to dynamically light everything and then move in to lighting sets or dynamic time of day stuff. So, to say, here is a set of lights and here’s one for the evening, etc. and trying to come up with the correct key frame times would be because you can’t really blend between pure day to pure night because in between there it’s going to look different, right? So that’s sort of the current issue.

One of the problems that came up with the current gen systems was that initially we thought we could use a lot of dynamic lights and have a lot of real-time rendering and of course everything starts to add up and the memory suddenly isn’t as much because you’ve got multiple texture layers and rendering passes and stuff and it kind of came back around full circle where there were just a few dynamic lights in the world and a lot of tricks being used to kind of makes things look good.

So, the problem with the process is really to use whatever tools are available and doing a lot of cheating and approximating which I think you’re probably doing a fair bit of here as well.

So on that note, what are your general thoughts regarding the behavior of the system?

My general thoughts are that it doesn’t look like much has been put into the background at this point. I like to see a lighting environment that really looks like it all works together. I think part of that is I’m just really only evaluating the lighting on the character but it’s kind of hard to tell [pause] The character doesn’t fit in with the environment so there’s no relationship…
INVESTIGATOR:

Could you go a bit more into [pause] How specifically? [referring to the 10_27_01’s statement that the character does not fit into the environment]

10_27_01:

Specifically the environment looks like your typical evenly lit no lighting environment. I can’t see any distinguishing shading other than a very kind of high-contrast from the key light. There’s not much fill there, there’s no shadows. There’s no ambient occlusion in the corners or anything like that it’s just basically really flat. I know that the technology is definitely there to do a more realistic job, so overall I’d say this looks like a kind of old school, virtual environment lighting. Which is still pretty common to see but I know the technology is there to do a more realistic job. There’s a lot of cheats you can use.

INVESTIGATOR:

So specifically…

10_27_01:

It seems very flat.

INVESTIGATOR:

So the environment seems very flat, but she [referring to the character in the software system] doesn’t, or…?

10_27_01:

She seems too contrasty. It looks like there’s a bit of a fill light there, in some cases I can see a bit of a rim light. In one of those modes, little bit. But overall it’s just way too harsh. It’s not adapting well…what I want to see here is certainly more of a fill light so that I can at least see her features [note: during this sentence, the 10_27_01 is pointing out the character being lit under modes that result in significantly portions of her body being completely black or nearly completely black]. Usually what’ll happen in an animated series and such is that they’ll have a separate pass. The background’s gotta look good and the character’s gotta look good. And they’ll often have a rig that even travels along with the character, so they’ll always be a rim light to a certain amount. And they’ll just kind of cover those basic three-point lighting principles.
INVESTIGATOR:
So is that notably about the same in natural mode, ‘cause currently you’re in auto mode so if you switch it to natural mode?

10_27_01:
The natural mode is better. But it’s still [pause]. Some of that may just be the gamma of this screen too, ‘cause it looks a little bit washed out. I guess the thing that I noticed the most is that there doesn’t seem to be much interaction with the environment. And, well here I can see that I’m passing underneath a light source. A lot of this could be not necessarily the technology but just how the lights were placed. I think you can work with whatever you got, and it’s just a matter of placing the lights a little better. But here you should never see that [10_27_01 is indicating the following situation – character is in the corner of the room with the fireplace, where there is no lighting from the direction, which results in parts of the character being completely black].

INVESTIGATOR:
You mean complete blackness?

10_27_01:
Complete blackness. Ya, it just never happens. So ya, I think what’s just really missing here is there’s no ambient light that I can see. There’s basically, in some places you’ve got a fill light, and there’s always that kind of key source, but [pause] There doesn’t seem to be any absolute, minimum level. It’s just black there.

INVESTIGATOR:
Ya, there’s no ambient, no.

10_27_01:
Ya, there’s no ambient, so. Whether that’s just a level or preferably it should be some kind of location based ambient light so you could say on here that there’s a little bit of glow from the fire or this room’s a little bluer than that one. Any kind of sense of ambient light will help here. So I think that’s the most noticeable thing that’s missing. Even just that in this case would help, ‘cause then it always looks like there’s something. Then the key lights really just for the accent.
INVESTIGATOR:
Would you say that that problem is worse or better with any of the different modes if you switch between auto or natural or attached?

10_27_01:
Here’s where I can kind of see a rim light, I don’t know if that’s kind of just designed…

INVESTIGATOR:
Well, if you’re in natural mode, that’s kind of just [pause]. She’s being lit by whatever’s there. So there’s nothing fancy going on in that case.

10_27_01:
So there’s no light rig that’s traveling with her or anything?

INVESTIGATOR:
No, not in this mode. So you’re basically just walking near a light in that case.

10_27_01:
It might just be an artifact or something actually on her arm it just looks like a rim light [pause] No, I’m definitely seeing something, at least a back light or something there. Ok, how I’m doing, next question?

INVESTIGATOR:
Is there anything else you would say about this [pause] Is the lighting satisfactory or not in any way that we haven’t discussed?

10_27_01:
I think what stands out to me the most is it’s very flat. The character isn’t tied to the environment. Like when I walk under this light here [pause] Well, ya, I guess I can see that there is something going on. There’s a little bit of a direction to the light coming from there. But it’s not that apparent.

INVESTIGATOR:
So is that a shadow problem or a light direction problem?

10_27_01:
The other thing is definitely shadows. Because there’s no cast shadows. Like if there was cast shadow here then I would definitely know where the light source was.
INVESTIGATOR:
Try cue 1. So pressing 1. See if you notice any difference with that.
10_27_01:
It doesn’t look like it’s getting quite as black. It just looks like [pause] There is a cast shadow on the character. So she’s self-shadowing. But I can only see it in some places. I think that looks a little better. I’m still getting this kind of blackness in this room [here the 10_27_01 is indicating the same problem with blackness in the same place in the environment as earlier in the interview]. It’s also that there’s just nothing really interesting going on in the lighting. So you’ve got the possibility of an interesting environment here. Like there’s candles on that table, there’s some kind of a lamp here that doesn’t look like it’s being used, I think there’s a fireplace there that could definitely be used. But in general it’s like, where’s the light source in this room [participant is indicating the center room of the demo]? I have no idea. It’s like there is no visible light source in the ceiling yet it looks like I’ve got a little bit of light coming from everywhere but no where in particular. So again, maybe it’s just the mood, the mood in the environment, that’s kind of missing.
INVESTIGATOR:
So what do you think the role of the image is then?
10_27_01:
I was just thinking about that. Well, that looks like an interesting lighting model. I assume that’s a tool for the lighting designer and probably just a way of, just a rendering model, maybe what this whole thing’s all about.
INVESTIGATOR:
What do you mean “an interesting lighting model”?
10_27_01:
Well, I’m guessing that that is, I don’t think it’s changing…
INVESTIGATOR:
Make sure you’re in auto mode, ‘cause in natural mode, she’s just completely environmentally lit.
10_27_01:
Should that image being changing as well?
INVESTIGATOR:

The image is sort of [pause]. You’re specifying your lighting goal in some sense, with that. So that is feeding the system. So it changes if you tell it to change. But otherwise it’s sort of just the input.

10_27_01:
And which keys change the image? [referring to keyboard keys]

INVESTIGATOR:
Q, W, and E.

10_27_01:
And you said I needed to move around a bit to see that change?

INVESTIGATOR:
Yes.

10_27_01:
So, looking at the image is my guess is it’s a simplified lighting model. That allows you to use an image to basically light the scene or, like you said, just modify what’s going on in there, hopefully there’s some influence from something else. I could see something like that being used possibly on the character but certainly not on the backgrounds at all, that wouldn’t work very well. If I was going to use a lighting model like that I would probably restrict it to the character and do the backgrounds a different way.

INVESTIGATOR:
So, is there any sense what it’s doing specifically right now?

10_27_01:
Well, when I look at the image and I look at the character I can see it does seem to be the direction of the lighting and the overall shading. I’d be curious to see how much I could mess with that image.

INVESTIGATOR:
So actually manipulate the image, like in Photoshop or something?
Ya, to see if it’s strictly that image that’s driving the lighting or if it’s a multiplier of it or whatever. It’d be interesting to see if I could get something a little more realistic out of it. Overall, [pause]. So this one here doesn’t respond like I’d think though. I’ve got this image here that’s just really bright in the upper left corner and the rest of it’s dark. But I don’t seem to be seeing that on the character. So maybe I’m interpreting that wrong.

INVESTIGATOR:

Well, this system does take into account both the system and the environment. So, if you [pause] The attached mode. That is purely the image.

Ok ya, that’s more what I’d expect to see.

INVESTIGATOR:

So, back in auto mode for a second. So, how do you feel about the conflict that? The mismatch between what the image is and what you actually see on the character?

Well, in the attached mode it makes sense. In the other, I’m not sure what it’s doing. It doesn’t seem to be doing much.

INVESTIGATOR:

Ok, so anything else that I haven’t brought up or talked about that you’d like to say?

It’s hard for me to say what else without knowing what I’m actually critiquing or talking about. Is it the end result, is it the interface for putting this together…?

INVESTIGATOR:

It’s definitely not the interface. But, it’s sort of everything else. In terms of the lighting.

So is it the aesthetic way that this is put together? Or it is just like..
INVESTIGATOR:

Ya, it could be that too. So, I mean, your comments about the environment being very flat, the practical sources not being used well, is, ya, that’s what I’m looking for too. But, definitely not the interface.

10_27_01:

Ok, well, what else ya got?

INVESTIGATOR:

Ok, thank you.

10_28_01

INVESTIGATOR:

You said that you’re not a lighting expert, but could you give me some background about your experience with Interactive Art?

10_28_01:

I’m a ---- and I have been working in Interactive Art and teaching people how to make it ----.

INVESTIGATOR:

Could you go into some specifics about some of the stuff that you teach?

10_28_01:

You mean like the type of things…

INVESTIGATOR:

Ya, specifically…[pause]. Ya, the type of things.

10_28_01:

So, I’ve thought everything from PhotoShop to animation and so [pause]. There’s so many things that lighting affects in those things. So that would be about it. I mean, I don’t know how you want me to…

INVESTIGATOR:

Well I’m just trying to sort of establish the frame of reference that you’re coming from.
Ah, ok. A 2D artist.

INVESTIGATOR:

Ok. So what are your general reactions to the behavior of the system?

One is that it’s relatively easy to use. It doesn’t make it difficult to work with. They’re all in sets of three [referring to the ability to choose between three environments, three images, and three lighting modes]. It’s A, S, W. Setting 1, 2, 3. All that. Everything’s relatively easy. You can make easy comparisons by just [pause]. You know when you go W vs. E you see immediate changes based on where the key light is. Is it a key light?

INVESTIGATOR:

Yes.

I also did a bunch of TV production so we would always have to know how to light people so that they didn’t look two-dimensional and they popped out, that kind of stuff.

INVESTIGATOR:

Specifically with regards to the appearance of the character what are your feelings?

It makes sense how it’s working. The appearance on the character when it’s on setting Q [this is the light image indicating light from a rim light from the upper left] [pause]

So if I select 1 or 2 [pause] Well 2 is kind of ridiculous as an option because you can’t see anything. It looks like [pause] You can do night vision a lot better. [pause] When you do 3 and it moves to a more natural kind of look, it looks very good. And the fact that the light follows the person is [pause] That’s one of the problems that the lighting has to [pause] Tracking the lighting with the person is always difficult.

INVESTIGATOR:

What do you think the role of the image is?
10_28_01:
Tell you the direction of the light.

INVESTIGATOR:
Ok. Could you be more specific? [pause] Direction of light in…

10_28_01:
In how far it casts. I’d just assumed by looking at the image that it’s coming from the upper left [10_28_01 has image Q selected].

INVESTIGATOR:
Would you call the lighting overall satisfactory?

10_28_01:
Yes.

INVESTIGATOR:
Is there anything specifically that you would have done differently? With this environment or the demo?

10_28_01:
No. You have variables, you have things to choose, you have options as to how to look at things, you can look at things close or far away. You can change your perspective on the camera and at all times. It all makes sense. I’m looking at it right now and her face is away from the light so of course it’s more shadowed so it seems to be very natural. Where as if I turn her around then the back has the right shadow. So it does that very nicely.

INVESTIGATOR:
Any other comments, is there anything else I haven’t talked about that you’d like to say?

10_28_01:
No, it works very nicely and it’s easy to understand. Which I think is fundamentally the best thing you can do [pause] It kind of has an intrinsic [pause] Like I can just sit down and use it right away, makes it a good tool.
INVESTIGATOR:
Describe your creative process a little bit for me.

10_28_02:
On a film that’s scripted and has dramatic elements you decide on a look that supports telling the story you’re telling and you usually bring sample images of the kind of look that you’re interested in when you sit down and talk with your cinematographer. And sometimes with a lighting person as well but often it’s just with the DOP [DOP means Director of Photography]. And then there’s another conversation often at a rental house with the head lighting guy, he’s usually a guy, to talk about what tools would be needed, about what kind of units would be needed to create the look that you’re after so that everyone’s on the same page with what you’re going for. Do you want really flat, hot colors, washed out, desaturated, kind of a desert, if your main character is emotionally vacant and you want to see that. Do you want Rembrandt lighting, deep shadows, warm colors, etc. So, that’s the discussion and then it’s sorting out the tools, the actual lighting units that would give that to you. And of course budget.

INVESTIGATOR:
What are your general reactions to this system?

10_28_02:
It would help [pause] Let me just say, the other territory of lighting is in documentaries where you have much less control. And if you even get the opportunity to go look at the locations then you would also have a discussion just about how to get enough exposure. In some case, you’d just want to see what is going on in this room, and you’d be, give me a big Chimera [Chimera refers to a brand of film lighting instrument] and get the ambient up, that’s all we’ll have time to do, that’s all you can do. A lot of stuff is done with practicals so any lamps, the fire, the things in the room become sources. That’s what you’re looking at and if you’re in [pause] They’re just a whole different set of considerations. You can’t fly big silks on century stands in somebody’s living room. You’re dealing with a different palette, a different set of tools.
This [pause] If there were more possible settings [pause] Is this helpful? Are we supposed to go with this?

INVESTIGATOR:
Yes, yes, very much.

10_28_02:
Ok…[this] would go some way towards a very broad outlines. Using this as a planning tool or as a pre-visual tool with your DOP, with your lighting person, or if it’s a game scenario then how’d you be lighting the figure to guide the player. It’s similar [pause] It’s a similar challenge in a way…

INVESTIGATOR:
Just to interrupt you for a second when you mean more possibilities, you mean the images?

10_28_02:
No, the actual images [pause] The lighting images.

INVESTIGATOR:
Right, the lighting images.

10_28_02:
Ya.

INVESTIGATOR:
Go into a little bit [pause] What’s your interpretation of what those images mean? In terms of what the system’s doing with them?

10_28_02:
My interpretation is I’m looking at the source of the lighting. When I look at this [referring to image attached to the key Q] I see there’s a top 45° angle [pause] It’s almost a rim light. It’s fairly far back, so it’s not lighting the front. It’s not all the way back ‘cause I can still see, it’s a highlight. This [referring to the image attached to key W] I’m seeing a light with more spread, lighting from the bottom right quadrant, but a fairly hard light, in the sense that this is in shadow [10_28_02 points out the upper left corner of the image]. It’s not diffused enough to get me bounce back here. This [10_28_02 points to image attached to key E] I’m looking at the same idea but from
more [pause] more top-ie. It seems a little more top-ie even than if you were to switch this one that I’m looking at [10_28_02 indicates the image attached to key Q]. [pause]. [It shows] Where’s the light coming from. If this were the key light, [it shows] where’s the source and what’s the nature of the source.

INVESTIGATOR:
Would you say based on your interpretation that what the system is doing makes sense?

10_28_02:
Yes. I’d want to spend a bit more time with it to see how wandering around in this environment, what the implications were? ‘Cause this room for example [indicating the first room with the dining room table], there’s more kick coming off the walls. So this [referring to the auto mode] gives me sort of my ambient idea and then the attached [referring to attached mode] looks like it’s just being lit from the global source.

INVESTIGATOR:
By global source, you mean the image right?

10_28_02:
Ya, these guys [10_28_02 indicates the images displayed in the system]. So it’s independent of …[pause]

INVESTIGATOR:
The attached mode, to put it a different way, might be like how a lot of games would do it right now. In that, all they really want is an attached rig. And then they let the designer figure out how to make that fit into what’s around her depending on [pause] Just guessing in advance, that kind of thing.

10_28_02:
So you’re modifying it?

INVESTIGATOR:
In auto mode…
10_28_02:
Right. So, clearly I look at it from a film lighting perspective. And I know it works differently in games. But I would say that the effect, ‘cause we’re so accustomed to looking at real world lighting, and lighting in film and television and other media, that they way we read it actually doesn’t [pause] It doesn’t matter how you got it. It doesn’t matter whether it’s spectral, reflective, whether you’ve programmed it into surfaces or whether you literally shine a light, it’s [pause] That’s my bias I guess.

INVESTIGATOR:
So, you would say that the final effect is what matters, it doesn’t really matter how you got there?

10_28_02:
Ya, that’s right. ‘Cause obviously you get there in a whole different way, like conceptually the way you bring light to a game is completely different.

INVESTIGATOR:
Overall, would you describe the light in this environment as satisfactory?

10_28_02:
No.

INVESTIGATOR:
In what ways?

10_28_02:
It’s too prude. The thing when you light is control and there’s very little control over the lights. You need all the tools that shape the light.

INVESTIGATOR:
Ok, so you’re talking about your ability to control what’s in front of you [10_28_02 nodes in agreement].

10_28_02:
This is very overall like the ambient [pause] With Q [referring to the image attached to key Q] there’s very little light so right away you’re into a certain mood with that choice of lighting. The W lighting stuff from below [referring to the image attached to key W] is a very specific use. It’s not a general purpose lighting scheme. It has
implications around horror. You’d never ordinarily ever light someone from below, it’s not flattering, it kind [pause] It makes people look odd. So E [referring to the image attached to key E] is the only one that in a way becomes [pause] Ya, I would put a light source there for general use. Except for very specific things, I wouldn’t use the other two, I wouldn’t use them to light a room necessarily. I might use them for a very specific thing. ‘Cause generally when you light I’d say it’s a different process but you [pause] You block out the areas where stuff is happening based on the narrative then control the light to create the mood to focus on your character or pick out things in the environment that you want your audience to see it’s a very [pause] Or you go outside and everything’s bright and that’s a different [pause] As an interior, this is the most challenging lighting setup, actually.

INVESTIGATOR:
Anything else that you have to say about this that I haven’t mentioned?

10_28_02:
This as a tool [pause] With more options it could be a really fantastic and quick way to pre-visualize lighting in a game. Or even in an animated storyboard for a film. It’s not detailed or refined enough to actually duplicate a full fledged lighting setup. You’d have to add lots more lights, there’d be a lot more going on. But as a way to pre-visualize and strategize about lighting, it’s very cool. It could be a really neat [pause] It’s a great alignment tool.

INVESTIGATOR:
Alignment?

10_28_02:
You could sit down with a designer or a DOP or whatever and you’re looking at the image and you can change it and so you’re getting [pause] You’re on the same page. You start to align everyone on the team because they can see what some of the options are and what the choices are. And that’s really valuable.

INVESTIGATOR:
Thank you.
INVESTIGATOR:
I’m sorry, what’s your background?

10_28_03:
I was in the film business for about ----. Mainly a film producer [pause] Did start off as an editor [pause] Occasionally a cameraman but mainly an editor, and mostly a producer.

INVESTIGATOR:
Could you describe a little bit of the process [pause] You can choose editing, or you can choose your camera work [pause] Could you describe a little bit of that process for me?

10_28_03:
Editing I'll talk about since that was my original craft experience. Some people say it’s just getting rid of all the bad bits which is really simply you shoot anywhere from 10:1 in drama, usually you’re averaging 6 or 10, meaning you shoot 6 to 10 minutes for every minute that you end up using. And it can go higher than that. Documentaries in the film world it used to be as high as 20, 25. In video, since there’s no cost, it can go up to the hundreds to 1 ratio. So getting rid of the bad bits, just cutting out all the bad bits is just very simplistic [pause] What you’re doing is you’re looking for the best bits of film you have. And then trying to structure story around them. In case of a script, you’re basing it on the script although surprisingly you still end up messing with the script a whole lot. In the case of a documentary you’re just going looking for the best bits you can find. Once you’ve found the best bits then you make up the story, essentially.

INVESTIGATOR:
What are your general reactions to this [INVESTIGATOR indicates the system]? 

10_28_03:
I didn’t know quite what it was. My first reaction was that I’d be able to play with the lighting, where as what I’m doing is playing with a set of presets. And depending on where I am in the room something like the night preset, 2 [referring to the environment preset attached to key 2] I guess, isn’t really all that interesting unless
you get in certain areas of the room where there’s some shadow. Where there’s actually some light and shadow, other views from other areas of the room. It’s kind of nice but it doesn’t have much active.

1 [referring to the environment preset attached to key 1] I generally find the most boring, 3 [referring to the environment preset attached to key 3] seems to be a little more interesting in terms of the contrast depending on where you are in the room. So basically what you’ve got is a series of presets relating the specific lighting on the character to the background in general and to their location in the room.

INVESTIGATOR:
What do you think is the role of the image? The thing in the upper right corner.

10_28_03:
Oh, I have to admit [pause] I only read it as the source of the primary light. Might have meant something different but I have to admit I didn’t…[pause]

INVESTIGATOR:
Do you notice anything specifically different between the three different modes? So, “auto”, “attached”, “natural”?

10_28_03:
I have to admit, that was one that escaped me. [pause] I can sort of see the differences it makes but in terms of functionality I suppose, in terms of how I would actually deal with it if I was a cameraman in here working on it. I can see that one of them was essentially the same lighting and another one was sort of unattached from the background a little bit. But I couldn’t change the background lighting so it didn’t really much matter to me functionally. Since all I could do was mess with the character [pause] And I couldn’t change what was going on in the background that was simply another preset. I guess I didn’t mess with it very much.

INVESTIGATOR:
So, overall would you call the lighting satisfactory?

10_28_03:
In some modes in some locations. The night location was in some ways the most interesting but only if there was a source of strong light. If she was on the other side of the room or further down this way [participant indicating the corner of the room
with the dining room table near the entrance to the adjacent room] there just wasn’t any light coming in on her, and the night mode [pause] Or if she gets over here [participant puts the character completely in the corner between the room with the dining room table and the room with the radio] she’s just completely dark.

INVESTIGATOR:
Well, actually that could be attached mode [pause] I think you’re in attached mode. Attached is meant to be intentionally sort of very, very disconnected from… [pause]

10_28_03:
If I’m in auto there [pause] The night in auto there, you can’t really do very much [pause] And then in attached mode it becomes quite interesting in fact that may be my favorite mode, in truth.

INVESTIGATOR:
Auto mode?

10_28_03:
Attached.

INVESTIGATOR:
Attached. Ok.

10_28_03:
Night in attached. Except it’s not very interesting there either, you have to go to [pause] [participant interacts with the system].

INVESTIGATOR:
How does natural mode look in this case?

10_28_03:
I tend to like the two extremes. Natural mode was never really doing very much for me. It was never of any interest. [10_28_03 interacts with system] [INVESTIGATOR explains some controls of the system] Ok, I can see a little bit more of what that’s doing [referring to the images]. I like the middle one best.

INVESTIGATOR:
The middle image?
10_28_03:
Ya. It produces the best [pause] [10_28_03 interacts with system]. It’s funny. When I came in, D [referring to the image attached to key D] gave me light but when I switched to something else I lost the light.

INVESTIGATOR:
That’s [pause] really more of a technical problem.

10_28_03:
Ya, ’cause sometimes I do like the night on D when the light’s there. But then attached only seems to work at night, it’s not very useful during either of the day time things.

INVESTIGATOR:
Any other comments I haven’t brought up or anything else you’d like to say?

10_28_03:
No, I’m still [pause] I wasn’t sure what you were bringing in. I thought it was going to be more of a lighting exercise but it seems to be more perceptual test in a sense. And if I look at it from a purely lighting point of view, I want to mess with it I want to move the lights. I want to change the background in relation to the foreground. So, seeing what happens to the character depending on whether it’s responding to the environment. If I can’t change the environment [pause]. So I can see what happens as a consequence so it does teach me something in a very preliminary sense so about what happens if you either have environmental factors feeding into the lighting situation or you don’t, you ignore it. But that’s all I can think of.

10_28_04

INVESTIGATOR:
Could you describe your process for me a little bit?

10_28_04:
That’s just [pause] If I built a model, I would render it with lighting in the environment.
INVESTIGATOR:
When you say render a model, what are you talking about specifically? What kind of modeling?

10_28_04:
A house, or in one of our projects we [pause] I made some characters for this game, so I placed some lights and made some renderings of it.

INVESTIGATOR:
So you’re talking about building levels for games?

10_28_04:
Ya. Or sometimes I do a model of my house, I’ll do some lighting for that.

INVESTIGATOR:
What are your general reactions to the behavior of this system?

10_28_04:
I thought it was neat how you could detach the lights [pause] Like, this is a scary look [participant has selected image attached to letter W] with this, but if you put this it doesn’t quite match.

INVESTIGATOR:
So you’re talking about attached mode, specifically?

10_28_04:
Ya, attached mode. It gives a more cinematic look I guess. If this were a game you could create more cinematic looks on [pause] Really highlight your character how you want. Independent of the environment or [pause] You just have more control I guess. I haven’t done any lighting in games but [pause] Ya, it sounds like you’d have more control.

INVESTIGATOR:
What do you think the role of the image is? The image in the upper right.

10_28_04:
The lighting on the character. Or, if that was her face, if her face was a sphere, that’s how it would look.
INVESTIGATOR:
Would you describe the lighting as satisfactory in this environment?

10_28_04:
In the current mode?

INVESTIGATOR:
Just overall [pause] In a specific mode, a specific place [pause] aesthetically, functionally.

10_28_04:
That reflects this [referring to the image] well?

INVESTIGATOR:

10_28_04:
I’m not sure what you mean?

INVESTIGATOR:
Let’s say I gave you this room, completely unlit. Would you light it the same way? What would you do differently?

10_28_04:
Like this current look right now? I don’t really like these completely dark spots on her [participant indicates areas of complete black on the character]. And sometimes how her lighting doesn’t match with the background. If you just talk about these three settings on the background [referring to the room presets], it is. It could have used shadows I guess, but. Shadows and [pause] Not so even. Not so completely even. Darker corners maybe. Give some more depth to…[pause]

INVESTIGATOR:
So you mean the unevenness of the shadows is a problem or they should be more uneven?

10_28_04:
More uneven. The furniture doesn’t even really have shadows. Ya, you put a light there, put a light there.
INVESTIGATOR:
Any comments about this that I haven’t brought up or haven’t asked you about?

10_28_04:
So, what kind of applications would this have?

INVESTIGATOR:
Well, any other thoughts or opinions before I go into this?

10_28_04:
It’s neat.

10_29_01

INVESTIGATOR:
Could you describe your typical process as a designer?

10_29_01:
I’ll describe what I do, or have done in the past. I have been an art director for a video game company. My typical process is to actually create concept art first [pause] Even lighting and so forth. You would have mood panels, mood sheets that you would create [pause] Our concept artists would create for us to determine specific kind of lighting modes within the game. Typically we would create maybe 5 mood sheets to determine just the general kind of modes we would have in each game. We would then expand that into separate mood sheets for a specific environment types and so forth. And that’s typically hand painted by very good Photoshop experts.

INVESTIGATOR:
Excuse me, but what would a mood sheet consist of, more specifically?

10_29_01:
It’ll be a concept art of a scene in the game. So you would create [pause] Say that van outside, somebody would paint it in this kind of lighting conditions, and then in nighttime, etc. That would not just include the physical lighting conditions, we would actually add aesthetic conditions to it. So it would be design choices, for example, you would create quite artificial moods just using lighting, similar to the way theatre and film do it. Blue means this, red means that.
On a high-level that’s what we do at first. And then it goes into a 3D phase, we would actually build an environment and create the actual 3D concept art. We go from a 2D concept art phase into a 3D phase, we would then build 3D concept art of a particular piece of environment. It’s usually a slice of the game, typically called a vertical slice, because we’re trying to create a fairly full featured slice of the game from top to bottom. Then we would come up with some lighting modes and models based on that. We would design it that way and then we would change and input different conditions into that vertical slice. So, for example, the stress level of the character, the health of the character, the state of the game [pause] His general state in the game and also, we would add visual narrative to it. That is a key in video games is to create some sort of visual narrative. A lot of that is based around lighting.

INVESTIGATOR:
Could you give me an example of what you mean by visual narrative?

10_29_01:
A generic example or fairly clichéd example would be blue means sort of cool [pause] It would invoke specifics kinds of sensations for the character. For example, it might, depending on the art direction that you’ve chosen, it might mean it’s calm and safe. But when the lighting turns red, yellow, orange, those kinds of things it might mean it’s just danger around the corner, that kind of thing.
Those are the kind of things that we do. And they’re usually aesthetic choices they’re not necessarily physical.

INVESTIGATOR:
So you mean physically motivated?

10_29_01:
Yes, exactly. The physical [pause] The optics of rendering. It’s very artificial, theatre lighting is that way as is film lighting as well. The conventions came from theatre. So, we would design a game with a visual narrative in mind right from the start. Even things like cameras, and the way that the lenses change. The way that the camera angle changes and the depth of field, etc. are all based around the kinds of inputs that we provide into the character and lighting system. So usually we would provide it with very high-level things like the status of the character and also where they’re at in the game, which particular level. Each level would usually have its own set of
lighting modes and also what the game designer decides that character should be feeling at the moment.

At a high-level we do that. And then we go from that vertical slice we would often design lighting systems based on the kinds of choices we made. So usually it’s very high-level choices, we would provide it with basic hooks. And then the game programmers would [pause] I guess you could basically refer to those hooks as knobs [pause] [the game programmers] would twiddle the knobs based on game states and so forth.

INVESTIGATOR:

What are your general impressions regarding this system?

10_29_01:

It’s a simple system. There’s a few things I have questions about. For example, there are windows over here. I’m just wondering why there are no lights that represent the windows, unless [pause] ‘Cause I don’t see a change going from natural to… [pause]

INVESTIGATOR:

Ya, there’s no lights representing the windows.

10_29_01:

So, I would say your ambient lighting, at the very least, should account for actual sources of light, and not just aesthetic sources. When I switch to setting 2 [referring to the environment preset attached to key 2], there seems to be a light source here [10_29_01 indicates the light source in the ceiling about the dining room table in the first room] but it doesn’t seem to cast on the table. That should be there. Also, it doesn’t affect the character in an expected way for me. This [10_29_01 indicates the lighting in the room with the dining room table] looks like it has direction like a directional light but [pause] As this [indicating the point of light in the ceiling above the dining room table] is the only point source that I can see in the room, yet it’s [referring to the lighting of the character] coming from almost directly above. Just wondering what that choice means [note: the 10_29_01 has image Q selected during this dialogue].

INVESTIGATOR:

Related to that, what do you think the role of the image is, in the upper right?
10_29_01:
I would imagine that it’s sort of the lighting cast on the character. That’s what I’m guessing. [pause] By the individual light sources. I’m not entirely sure, actually, to be honest, now.

INVESTIGATOR:
You’re in auto mode, correct?

10_29_01:
Let me put it in auto mode…

INVESTIGATOR:
Try moving her a little bit after you change the image.

10_29_01:
To be honest I still don’t understand it.

INVESTIGATOR:
Would you describe the overall environment as satisfactory or how else would describe the lighting in this environment?

10_29_01:
It’s not satisfactory. I would certainly [pause] It’s very flat, at least compared to the kinds of standards I’m used to. The actual surfaces, the walls, etc., should have more variation in lighting, just to give it overall more lighting texture as I often like to call it. It gives the environment more depth if you do it that way. Currently it’s a very simplistic model [pause] Just a general ambient light that’s applied to the [pause] Maybe a couple of directional sources or point sources that gets applied to the environment.
Typically we would often hand paint light maps, just to make it look better than if you’d just placed lights within a 3D context.

INVESTIGATOR:
Any other thoughts that I haven’t mentioned?

10_29_01:
Ya. Some of the props seem a little on the hot side.
INVESTIGATOR:
So you mean the lighting casting on them is too bright?
10_29_01:
The way that the props react to the lighting looks pretty unnatural to me. It looks more like a UV light or a black light. For example, this thing here [10_29_01 indicates a doily under the candlestick on the dining room table] and the newspaper is quite bright, unexpectedly so.
The light source seems to look green here [10_29_01 indicates the ceiling light in the room with the dining room table] but it’s casting almost a black light blue on everything else. It doesn’t match [pause] I would expect it to cast at least some of this light somewhere [10_29_01 is referring to the green hue of the light in the ceiling not casting on the rest of the environment]. On the table, up on the ceiling. That’s going to contribute to the general lighting in the room. It’s going to be reflected off of there and if you take into account things like bounces and so forth, some of this green needs to end up somewhere.
INVESTIGATOR:
Anything else?
10_29_01:
I think I talked about the windows. I guess you’re not really considering shadows or anything in here. I would expect some sort of shadow here. Some ambient…
INVESTIGATOR:
Around where the chair is?
10_29_01:
Coming from the windows. So there should be some lights coming through the windows casting shadows…
INVESTIGATOR:
You mean the actual…[pause].
10_29_01:
The mulleins or whatever.
INVESTIGATOR:
The mulleins.

10_29_01:
Or whatever they're called should be casting. It looks like from the image I see in the background [10_29_01 is referring to the images placed behind the windows to create an illusion of outdoor space] that the light is sort of that way. I would expect the Sun or the outdoor contribution to be in this general direction. So the window shadows should sort of splay out in this direction. There should be some lighting coming through the windows. I don’t see any contributions of the outdoor lighting on the interior environment. Nothing significant.

If you look at it now here [10_29_01 is indicating the physical room that the interview is taking place in] you’ve got tons of lighting just from the windows and most of it is coming from the windows. The actual electrical sources contribute a fairly insignificant amount and you’ve got fairly large windows here [10_29_01 is again referring to the windows in the software system demo] and there should be more of that in the environment.

INVESTIGATOR:
Anything else?

10_29_01:
I see some shadows here [10_29_01 is indicating the shadow cast from the radio located in the center room]. I’m wondering why there aren’t more shadows. There seems to be a shadow right there but I don’t see one here. I’m not sure where the point source is here. There seems to be a directional source almost directly above [10_29_01 is indicating the area above the radio in the center room]. There should definitely be pools of shadow in the corners and so forth just to also once again give it depth. Just once again in video games and in film, we would often paint artificially the shadow map just to get that [pause] more depth. Just to add more depth in the environment.

So, I would say generally it's very flat lighting, it’s typical to me of a VR environment more than a video game environment.
INVESTIGATOR 2:
Any idea about the lighting on the character?

10_29_01:
Lighting on the character. Yes. Still kind of the same. The character is kind of a like a prop to me in the environment. In video games as well we would often have a special light rig attached to the character, so you’ve got the right idea here. And the contributions of the environment lights add to the character lighting. I would expect a little bit more contribution from things like the windows and right now I don’t see any contribution. As a matter of fact there seems to be shadows where there shouldn’t be. ‘Cause right now she’s standing by the window in a fairly bright day yet there’s just shadow being cast, right there where the windows are.

10_29_02

INVESTIGATOR:
Could you briefly describe your process as an artist or designer?

10_29_02:
I create interactive projection designs for live performance. Sometimes they are more in the genre of visual music [pause] Projection imagery for live music performance. Other times they are much more interdisciplinary, incorporating dance. But, [I do] primarily live stage interactive performance. I also use software to control interactive lighting.

INVESTIGATOR:
Do you mean that the lighting responds to a dancer’s movements or something of that nature?

10_29_02:
Yes, exactly. Or to sound.

INVESTIGATOR:
Sound? Oh, interesting, ok [pause] What are your general reactions to this [INVESTIGATOR is indicating the software system of the experiment]?
10_29_02:

Well, I’m having fun. As a person whose had to create lighting on people most of my artist career, I was a photographer before being a film maker before being an interactive performance artist. So I’ve always had to light [pause] I have much more experience in those backgrounds then in gaming. However, as I’m walking through the one question [pause] I liked the set that you’ve come up with, I just [pause] I felt an incongruity room to room that the direction of the key light and the fill light didn’t seem to change no matter which room you were in. It was simply a function of the setting key. It didn’t seem to change enough in my mind. Especially from the first room which has windows [pause] Visible windows where in this middle room and the third room, there don’t appear to be any visible windows [pause] So it felt to me [pause] Although it was reasonably good, some of the directions of the key light did not feel quite spot on to me. As far as comparing it to a real world environment.

INVESTIGATOR:

Just so this is on the recording, specifically you’re referring to the key light following [pause] environment motivation? Or [pause] practical sources?

10_29_02:

Yes. I guess in the way that this is constructed it would be the environment lighting, which if I’m understanding correctly is connected, as long as you’re in auto mode, to the direction of the key light.

In this situation for instance there must be light coming through the windows as well as coming from lighting fixtures in the room. However, it feels like when I change between the different settings [pause] This is obviously later in the day and the key light is more internal in the room [10_29_02 has selected room lighting 1], I’m assuming [pause] This is nighttime [10_29_02 has now selected room lighting 2] [pause] And this one is a daytime one [pause] Ya, at no time does the key light seem to be affected by the exterior lighting in that room.

INVESTIGATOR:

So would you describe the lighting in general as satisfactory?
10_29_02:
Just with that one proviso, yes. It seems to be congruent. As she moves, the [pause] [participant interacts further with the software system] Satisfactory yes and no.

INVESTIGATOR:
Could you go into that a little bit?

10_29_02:
As a non-gamer…

INVESTIGATOR:
Well, don’t [pause] don’t worry too much about trying to empathize as a gamer. Think of this just as from your professional experience.

10_29_02:
I feel that the lighting should change more room to room. If I was just observing the model in one room under different settings, I would say it was totally satisfactory. But when changing room to room, it feels that the position of the key and fill light don’t change enough to reflect the change to the room environment.

INVESTIGATOR:
Ok. What do you think the role of the image is? In the upper right?

10_29_02:
It’s a model of the light setting parameters that you’re using [pause] Under that setting. So it’s just a representation, I think, of the direction of light and the quality of light. So that’s a smaller light source [10_29_02 has the image attached to Q selected] and it models the direction it’s coming from. Possibly without any fill light [pause] [10_29_02 interacts with the software system].

INVESTIGATOR:
After making the change, try moving the character.

10_29_02:
[10_29_02 interacts with system] Ya, it seems to reflect a harder and softer light source. A larger and smaller light source. With E [10_29_02 is referring to the image attached to E], softer again but from a different direction [pause] Top as opposed to bottom or side light [10_29_02 interacts with system]
INVESTIGATOR:
As a designer would you do anything differently in this environment? Other than what you’ve mentioned about the change from room to room…

10_29_02:
Only that it would be wonderful to have mouse control [pause] To change the position of the fill and key lighting and to change the intensities, to be able to just click and scroll the numbers. It would be nice. It’s not a deficiency it’s just an additional…

INVESTIGATOR:
That’s exactly what I’m looking for, ya. Anything else I haven’t mentioned or that you’d like to say about this?

10_29_02:
Well, I’ve noticed that you’ve got two fireplaces in these two rooms here. It would have been interesting to see the effect of the lighting coming from a fire that was lit, since they’re there. Because it would add another fill light from another angle, for instance. If I was to walk a character up to this fire the light from the fire would [pause] It would be interesting to see that. I guess that would also apply to the candles. [next two sentences indecipherable] [10_29_02 interacts with the system]. The fill light intensity [pause] The number doesn’t seem to ever go very high [10_29_02 is referring to the numbers displayed next to the fill and key lights]. I’m not sure if that’s because [pause] It does change. Obviously it couldn’t [pause] [indecipherable phrase] Is a value of 1 making it a key light? Or is it a value of 100 making it a key light? Is it a percentage or is it a…?

INVESTIGATOR:
Before I answer that question, any other thoughts?

10_29_02:
I don’t think so, that’s good.

10_29_03

INVESTIGATOR:
Could you briefly describe your process as a designer, for me?
10_29_03:

My process as a designer is to analyze mechanically and aesthetically the overall event I’m going to design. I really only do live events [pause] Plays, musicals, songs. What is it mean and what’s important to see [pause] What do we need to see, because I’m working in a live situation and presuming that it’s not outdoors, I’m in control of what we can look at. So, what’s important, because if I don’t light it, we won’t see it. Then what’s an overall thematic idea that’ll hold all of the various different looks together in some way. That’s step one, to kind of get a sense of that [pause] It may mean writing something down.

Step two is to break the event down into parts and identify preliminary lighting states for each segment of the part. They could be what stage lighting designers would call looks or cues or they could be blocks of looks or cues like might be in a scene. And then the third step is [pause] Meanwhile, I’m kind of negotiating this with people I’m working with. I may be showing it to them live, I may be talking to them about it, I may be making a sketch or two of it. But it’s mostly to get a sense of what the overall tone and kind of macro structure is and then once that seems fairly firm, to go and break it down into small manageable parts so I can assign lights and hang them.

And finally it’s to install it into a theatre or other venue and look at it, change it, fix it, or completely throw everything out and start over. And then finish, walk away because it’s time for it to be seen by the public.

INVESTIGATOR:

What are your general impressions of this?

10_29_03:

I like it, it’s easy to learn. I could learn what it does from these commands fairly quickly. There’s things that I want to be able to do, probably because lights are what I think about, and that is have more control over the lighting of the environment. I don’t know how easy that is o program and I don’t know how easy that is to [pause] Or how likely it is that somebody who isn’t a lighting designer would want that. But right away I don’t like that you’re only giving me three choices about what the environment looks like.

It doesn’t really bother me [pause] I don’t find it necessary and I don’t really [pause] Like, I really like this look [10_29_03 has configured the system to take advantage of
a bug, where the character is lit using the “warm” lighting motivated by environment
cue 3 but has environment cue 2 selected, so the environment is a dark, nighttime
appearance while the character is unrealistic warm and isolated] because I can see the
character better than anything else. So that’s telling me ok, the character is the most
important thing which is the thing that I’m accustomed to thinking. This one
[10_29_03 selected environment lighting cue 1 in auto mode so that everything,
including the character, is lit according to this cue] I can still see the character but I
have a lot of other information that I may or may not want and that’s why I want to be
able to control the lighting in the environment because maybe the character’s
significant and this console [10_29_03 points to the radio in the middle room] here is
also significant and nothing else is very significant. So the ambient light [pause] It’s
great to shoot a movie because overall it’s evenly lit, the camera would register but
for a stage designer I’m going “Ok, why do I have to look at the whole thing I only
want to look at the three important things.”

So, that subtle control over the environment but I don’t know how useful that is to
somebody other than me. And overall I really like it that’s kind of the only thing I
miss, I think this little interface with there’s a ball [pause] I mean, it’s good to learn
and then you go and get rid of it right away because you can just see it on the person
and start learning the different [pause] You’re using the W key and change the kind
of direction of light that’s on the character and that’s fine. It would be nice to be able
to roll those to different directions, like to orient them three dimensionally relative to
her rather than having to orient her.

INVESTIGATOR:
Can I interrupt you for a second?

10_29_03:
Ya, please do.

INVESTIGATOR:
Could you describe [pause] Just since you’re talking about it, could you describe
what your interpretation of the image is, specifically?

10_29_03:
Which image?
INVESTIGATOR:
This thing that you’re talking about, in the upper right.

10_29_03:
Light is [pause] My assumption is that light is coming from this direction on what’s a sphere.

INVESTIGATOR:
Ok, thank you.

10_29_03:
And she’s the equivalent of the sphere but, because the little red dot is, I’m assuming the light, that’s coming from the direction, I have to be oriented relative to it and to her here, if I’m looking at it this way then she doesn’t have a lot of light on her, although there does seem to be another one over here [pause] So you’re doing a key and a fill on each one, which I’m not reading so well on that [10_29_03 indicates the image] that there’s both a key and a fill. But, suppose I want to swap them? What I really want is to click on this little key and put it over here, click on this little fill, put it over here. But again, that’s a lighting designer talking [pause] Right away I want to mess with your decisions and make different ones and see what happens. Which I think if you’re trying to invent a lighting design program, it’s important to put that functionality in there. If you’re trying to do an add-on to a game where I’m just going to work with some parameters but really what I’m doing is playing the game rather than lighting the scene, then it’s great because it gives me some choice, not too much choice. I’ve not done any video games since Tetris so I don’t really know how this compares to other video game lighting options, like is it really sophisticated…

INVESTIGATOR:
In pure terms of physical realism, and the number of lights, this is probably 5 to 10 years old, in that regard. The current technology is quite a bit more advanced. They can easily have [pause] 10 times as many lights as are in this environment currently…

10_29_03:
And I can muck with them? Like I can change them, I can…
INVESTIGATOR:

Ya. Depending on [pause] There are limitations that may not be obvious because they’re sort of technical but if you want them to change while the player is playing, then they’re more limited, you could probably have [pause] Maybe 40 or so. If you want to set them and then they stay that way, and they really just want to change from one to the next, then you can have probably as many as you want.

10_29_03:

And are there little sliders and things [pause] Virtual ways of controlling ‘em?

INVESTIGATOR:

The interface’s are all over the map. But just in pure terms of how it looks, ya this is probably about 5 to 10. I would say.

10_29_03:

I would probably want, and I want this in a different context for live lighting, I want to be able to left or right click on something and change the light that’s on it that way or left or right click on a light and move it around from one place to another. I’ll probably also want to be able to govern the kind of light [pause] I’m not sure but I’m not seeing any cast shadows that might be because you just decided not to render ‘em.

INVESTIGATOR:

Yes.

10_29_03:

Because they take so much longer. But that makes it difficult [pause] Not difficult, it’s just nicer to have them because they give it a better sense of sort of depth [pause] –ness. Which I imagine is useful. But again, it’s useful aesthetically, maybe it’d be useful in a game story…

INVESTIGATOR:

I can say [pause] It tends to be very useful. Shadows are very popular and they’re very difficult to do, for technical reasons. Shadows are interesting in the sense that their limitations are different, purely on technical grounds, then lighting. So, the way you’d see them in a game is kind of [pause] interesting some times, in terms of where they show up and where they don’t.
10_29_03:
What I really am most fond of is this ability to detach [pause] I like that ability to detach the figure from the environment. Just because it’s a kind of shadow like effect where I can make things that I regard as less important [pause] But I sure would like to be able to light something in the background. Oh, I don’t know why because in the context of a game maybe you don’t want to do that, there’s certainly the game isn’t going to be about designing the lighting.

INVESTIGATOR:
I don’t know about any games that are about designing lighting but designing lighting in games is becoming very important, from what I understand.

10_29_03:
So, I like it.

INVESTIGATOR:
So, actually, you’ve managed to answer every single question I have her, so [pause] Any other thoughts?

10_29_03:
How hard is it to program and what’s it programmed in? [transcribed portion ends here]

10_29_04

INVESTIGATOR:
Could you briefly describe your process as a designer?

10_29_04:
For lighting?

INVESTIGATOR:
Ya.

10_29_04:
I set up the stage and first develop what I actually want to light for. So I usually do research. So I can support whatever I’m wanting to light. So that’s in the background of what you’re showing me. And…
INVESTIGATOR:
Could you go into the research part a bit, specifically?

10_29_04:
If I was doing a video piece where I would require quite a bit of lighting to set the stage, I would look into what my concept is about or what it’s addressing. I would kind of mimic off some of the lighting that was done in the films I’m being inspired by. That I’m using for the piece. I just do a whole bunch of research by just looking at different images and different angle shots and the consistency.

INVESTIGATOR:
What are your general responses to this [INVESTIGATOR is referring to the software of the experiment]?

10_29_04:
I like it, I just wish there was more opportunity to play with the lighting. Whether it was dim or…

INVESTIGATOR:
What part of the lighting specifically?

10_29_04:
There’s two parts to it. The lighting cue [pause] Being able to try different intensities with the lighting. And then also for the character lighting, the different angles. So instead of it more like a flat, straight on or like a spot light straight on [pause] Actually he spot light is kind of coming all the way down the middle of the body. But, being able to play with where you want to be able to position the lighting. It would probably be a little more difficult because it’s like, unless you’re setting up a specific scene, which could be interesting but I think in this scenario [pause] I’m not sure. Or allowing the light source to make it out from the windows or whatever you’re wanting it to come out from.

INVESTIGATOR:
So, if I say you would like the light to be more motivated by a practical source, is that accurate?
10_29_04:
I guess being able to move the light source wherever you feel it’s needed. Like being able to manipulate where it’s positioned in space. ‘Cause I’m not sure right now where it’s coming from.

INVESTIGATOR:
What do you think the role of that image is?

10_29_04:
Is that [pause] That’s probably what it is.

INVESTIGATOR:
Well, what do you think that represents?

10_29_04:
I think that’s what it represents. But I don’t find it as indicative. So I was just playing around with it. If I’m able to go into it and move it around, then I’d be able to see how it’s flowing when I’m looking at it at the same time that I’m looking at the picture, then it’d be cool ‘cause I’d be able to [pause] And I’d feel more in control of where the light source is coming from and how it’s being moved. But that’s just assuming it is.

INVESTIGATOR:
Other than control of the light source, would you call the lighting here satisfactory?

10_29_04:
Mmmm-hmmm. Ya, totally. I’m not too sure about #2 [10_29_04 is referring to scene lighting cue 2]. Is it for nighttime?

INVESTIGATOR:
Sure, you could describe it as that.

10_29_04:
Oh, ok. Like, I thought it was maybe [pause] This one seems it has a filter to it.

INVESTIGATOR:
That’s cue 3?

10_29_04:
Ya. No, cue…
INVESTIGATOR:

Or 1?

10_29_04:

This one’s the most practical [10_29_04 is referring to cue 1]. This one seems a bit too, too dark. Could you lighten it? No? Like I kept trying to think that I could press 2 and 3 and get the medium in between…

INVESTIGATOR:

No, they’re [pause] Think of them like a stage lighting cue. That’s pretty much what it is. So which one, I’m interesting in you said that one looked like it had a filter?

10_29_04:

Uh, ya, #1. It just looks like there’s a slight filter to it.

INVESTIGATOR:

Interesting. So what kind of filter?

10_29_04:

Like a blue filter. Or, magenta. Is it magenta?

INVESTIGATOR:

Any other comments?

10_29_04:

No, it’s pretty cool. I think it’s awesome. The only main thing I thought or you need is if you were able to position where you want the light to be placed. So maybe if you got miniature cues or something that would indicate like somewhat to where you have the red [10_29_04 is referring to the red arrows that indicate the light sources affecting the character] where it’s just visible to myself but not maybe the person who’s viewing it on the other end. And then it would be kind of neat because you could manipulate it. Or you could set up a scene, so if you’re doing a Machinima and if you wanted to build that film effect, you could use it to do that advantage.

INVESTIGATOR:

Anything else?

10_29_04:

No.
INVESTIGATOR:
Could you describe your process as a designer for me?

10_29_05:
In what I do now or what I’ve done in the past?

INVESTIGATOR:
Well, what are your areas?

10_29_05:
Film, mainly. And photography.

INVESTIGATOR:
So pick whichever one you would prefer to talk about.

10_29_05:
I come from mainly a journalistic photography background. So much of the stuff I have done deals with actually mostly ambient lighting but when I do use light it’s usually a combination of three or four flashes. Film wise, more static lighting than anything.

INVESTIGATOR:
What do you mean by static lighting?

10_29_05:
With photography I would use mostly synchronized flashes so they’d only strobe. With film it’d be just static lighting, meaning always on.

INVESTIGATOR:
What are your general thoughts about this?

10_29_05:
It was pretty cool. I like that you’re able to switch between lighting styles very fast. And it seemed pretty accurate from what I’ve done before in real life.

INVESTIGATOR:
Accurate with regards to…?
10_29_05:
Directional lighting, types of lighting that you’re able to do. Ya, definitely with natural and attached lighting.

INVESTIGATOR:
Would you describe the lighting as satisfactory?

10_29_05:
Ya. For this virtual world it seems pretty satisfactory. Do you guys have plans to put it into other…?

INVESTIGATOR:
Let me get back to that afterwards…

10_29_05:
Sure.

INVESTIGATOR:
So, with regards to what you have here you would describe it as satisfactory?

10_29_05:
Ya.

INVESTIGATOR:
What do you think the role of this image is, in the upper right corner?

10_29_05:
Changing light source’s direction, it seems.

INVESTIGATOR:
So you would describe that as representing direction?

10_29_05:
Ya.

INVESTIGATOR:
As a designer yourself, would you have done anything differently in this environment?

10_29_05:
As far as designing the whole thing?
INVESTIGATOR:
Lighting, or ya the whole thing.
10_29_05:
Possibly a bigger environment. It’d be nice to see how this lighting would deal in outside environments possibly vs. inside. Possibly different characters. Just to see the effects on different models and different shapes.
INVESTIGATOR:
Any other thoughts or anything I haven’t talked about?
10_29_05:
Not other than the one I was just gonna ask.

10_29_06

INVESTIGATOR:
Could you describe your process as a lighting designer for me? Or whatever [pause] I’m sorry, what are your background?
10_29_06:
Film.
INVESTIGATOR:
Film? So ya.
10_29_06:
What I did, you mean? Or what I did in this…?
INVESTIGATOR:
Projects that you’ve done before this.
10_29_06:
So I used to setup lighting for photography for TV ads.
INVESTIGATOR:
Could you go into that a little more specific?
10_29_06:
To specifically light someone you mean? What I did?
INVESTIGATOR:

Ya, I mean, run me through a typical shoot.

10_29_06:

Ok. You first measure the light on the person [pause] You put it there and make sure that [pause] Depending on the mood you want to give ‘em or what the advertising is all about, you have to understand the character of the actor and what you want to portray and then you decide what kind of lighting you wanna use. So depending on the mood [pause] There was shoot where we did for ---- where we wanted the character to be more perfect than real. Not reality. So we used a lighting system that shoot down on the ground and make the whole room white. The lighting ambience was really, really, sharp. So the person’s face was almost bleached [pause] Not bleached out but [pause] It popped out and kind of merge with the edges in the background. I’m not sure what you’re…

INVESTIGATOR:

No, that’s exactly what I’m looking for, thank you. What are your general reactions to this?

10_29_06:

I’m not sure what’s going on [pause] Why the face is so dark while I’m in this room. ‘Cause when I went to the other rooms and hid behind a wall, the same reaction did not happen. It seemed that it did not take the situation of blocking. I also wanted to see if it’s going to make me feel different when I was changing different light modes. Yes and no. The person didn’t really react [pause] I couldn’t really tell between E and W [referring to lighting images attached to E and W] but in this mode, which one is it? In the auto one, it seemed to be the better looking one, although again you have these shadows. In advertising you would never want to do that unless [pause] You don’t want them to become so dark unless there’s a reason for it. I was trying to figure out what the reason was she had a shadow cast on her in this spot when [pause] I understand that there’s a light up here but it just seems strange that when you’re closest that you wouldn’t have a corrective light to shine on her face. The one that I liked the most was the auto mode, that wasn’t too dark.
INVESTIGATOR:
So the auto mode in general?

10_29_06:
Ya, I think so, the auto mode. It looked a little bit better in general. And I liked how it slowly adapted to the light. When it comes too close [pause] I know it’s exaggerated but [pause] When I left the room I felt that there is nothing going on [pause] Like here, I see the transition here but once she goes [pause] Lighting wise she seems to be the same.

INVESTIGATOR:
Specifically, would you describe the lighting as satisfactory? You talked a little bit about it but…

10_29_06:
It depends on what exactly you want to do. If she was trying to move [pause] I’m a sensitive person and I come here and now I actually realize that I’m guilty [10_29_06 has moved the character to a spot where much of her face is complete blackness] where she slowly transitions to dark, then I get it. But to me it’s a bit dramatic in this case for no reason. I don’t get the reason, I don’t see the context. This one seems too plain to me, the S, the natural one. She seems very flat. [pause] But I’m not getting the same effect. Like, I don’t understand why this effect happens here. It’s funny how you’re trying to see the details of her face and you can’t. I think that kind of bothers you from the beginning because I think people have a tendency to really focus on facial features.

Ya, it becomes too dark too soon, I guess.
The effect of turning off the light it feels to me like the end of Act I or something like that [10_29_06 has transitioned room lighting cue to cue 2, the “nighttime” cue].

INVESTIGATOR:
What do you think the role of the image is, in the upper right corner?

10_29_06:
It’s to show where the light is coming from, right?

INVESTIGATOR:
That’s your interpretation of the image?
10_29_06:
Ya. It’s like a sphere that shows where there’s going to be more light on her.

INVESTIGATOR:
You’ve mentioned that it becomes too dark in some places in the environment, is there anything else you would have done differently as a designer in this environment?

10_29_06:
I would probably put lights on the candles. Not to work only with this ambient light but add a few light sources in the room and see as she goes next to them how she actually reacts. Also here I would probably dramatic the effect of the sunlight [10_29_06 has walked into the room with the dining room table] if there was one. If there was sunlight and whenever I get really close to here [10_29_06 is referring to the windows], I wish I could see from behind and see how the actual daylight actually affects her. Again, place some fire in the fireplace. I think it should be lit up.

INVESTIGATOR:
So the fireplace in the center room? So any other thoughts?

10_29_06:
I wouldn’t have both of them [fireplaces] though. I would only have one.

INVESTIGATOR:
So you would have one fireplace but not the other?

10_29_06:
Ya. I wouldn’t have this one [pause] There’s something wrong about having two fireplaces I guess.
I would probably, this is obviously considered a closed room, I would try to [pause] There’s no light in it [pause] I don’t know what the purpose of it is, if you want a light, I would put one there or maybe just have this one open.
I mean, if you want to make me actually walk then I would use light as a calling point in the sense that I am going to this room and I see this light in the background that’s brighter than the other one, if you don’t want to use the fireplace here, I want something to [pause] To call my attention to go where the light is instead of staying here.
Ya, I’m not sure what else you want me to comment on?

INVESTIGATOR:

That’s fine. Any other general thoughts or anything I haven’t asked about?

10_29_06:

I think it’s fine. I wish you’d have given me a [pause] Not a narrative but some sort of idea of what you were trying to express so I could understand more the [pause] Why the lighting was placed and the reasons it was placed.

Other than that, no it’s fine.

10_29_07

INVESTIGATOR:

Could you describe a typical process or you as a designer?

10_29_07:

In this environment?

INVESTIGATOR:

Well, what’s your background?

10_29_07:

Um, artist. 2D/3D.

INVESTIGATOR:

2D/3D art? So, pick an interesting project that you’ve done and walk me through it. So not this, but in your past.

10_29_07:

Last ---, we did a fishery, ----, which is an ocean species visualization. We built the models in Maya and import the model into Blender and all the lighting and animation we did in Blender.

INVESTIGATOR:

Go a little bit into the aesthetic decisions you might make doing that?

10_29_07:

Actually, I didn’t do a lot of work on the lighting part, which is kind of the weak part in my 3D works. Usually I just try in Max to put the skylight and use the advanced
lighting system which could mimic a kind of very realistic lighting reflection between objects.

INVESTIGATOR:

Would you say that for your [pause] Are you satisfied with the results of that? As in, using that physically accurate form of lighting?

10_29_07:

Ya, because there’s some way to mimic the realistic lighting which I saw some of in the tutorial which puts a lot of omni lights to fake a kind of environment which has a [pause] Final Gathering?

INVESTIGATOR:

Ya.

10_29_07:

Final gathering. [the process is] To use the regular lights to mimic the environment which saves a lot of system resources but it takes times to set up.

INVESTIGATOR:

What are your general thoughts or reactions to this [INVESTIGATOR is referring to the system of the experiment]?

10_29_07:

I find the 1, 2, 3 buttons which change the whole environment illumination [pause] The first one is kind of a cold color, like blue and a little bit green. The second one is the environment which is at night, very dark. This kind of feeling. And the third one is kind of warm color compared to the first one.

The Q, W, E buttons I think represent the lights from a different direction. First one from a top right and the second one is the bottom left and the third one is from the top.

A, S, W is a way to switch between different modes. I understand the last two [10_29_07 is referring to “natural” and “attached” modes]. Natural one is the one that won’t be influenced by the image. And the third one will be influenced by the image.

INVESTIGATOR:

So you don’t understand the difference with “auto” mode?
INVESTIGATOR:
In general would you describe this lighting as satisfactory?

10_29_07:
Satisfactory? [10_29_07 is asking for clarification of the word “satisfactory”]

INVESTIGATOR:
Ya. Pleasing, acceptable [pause] Maybe the other way to put it is, as a designer, would you do something different with the lighting in this? And that can be in a specific mode, that can be with a specific cue chosen [pause] How do you feel about the lighting?

10_29_07:
The first thing that I find is that the shadow is not quite good. I don’t know whether it’s very closely related to your research, but I didn’t find the shadow for some of the objects. I think the lighting is totally based on what kind of requirement you need, like what kind of feeling you want to show to your audience. Like in the night mode, I found that it’s [pause] The lighting comes from the bottom [10_29_07 is referring to the image that indicates lighting from below and is talking about this selection combined with the nighttime cue] could be used in some of the horror movies. This kind of lighting makes the audience kind of scared. That’s pretty much what I feel so far.

INVESTIGATOR:
Specifically, would you have done anything very differently from what you see in this? Is there a particular mode that you found was really, really off or something that you saw when you were walking around was particularly [pause] You didn’t like it and you would have done it differently?

10_29_07:
The one thing I want to mention is all these three lights is just a one main light to illuminate the character. Maybe it’s better to have another light to put in the other side which is much less intense than the main light as a kind of reflection from the
‘Cause usually the lights go on the floor and they are reflected to the character.

INVESTIGATOR:
So you mean like a fill light?

10_29_07:
Any other thoughts or anything I haven’t asked about?

INVESTIGATOR:
That pretty much cover everything.

10_30_01

INVESTIGATOR:
Could you describe your process as a designer briefly for me?

10_30_01:
For designing lighting?

INVESTIGATOR:
Ya.

10_30_01:
You mean in terms of how I light a scene or…?

INVESTIGATOR:
Ya, I mean you can pick a project or walk me through it however you want to do it.

10_30_01:
Ok, usually what we do [pause] Usually the way we light in here, we light the environment and characters separately. So the environment is usually [pause] We pre-light it. We use software render. And then transfer lighting into lightmaps that we use when lighting the game. We use a pre-baked lighting for lighting the environment. And for characters we use real-time shaders to do interactive lighting. So usually the tool parts are lights that are early.

For environments, the workflow will be [pause] It's pretty similar to when you would do lighting in a software package. You set up your key light first, and then [pause] Of
course you need to grab the ---- first [pause] What is your final target, what is the final look you want it to look like. And from there we analyze the image and start setting up your lighting to match that. And once you’ve done that we do a preview render and kind of pre-visualize it and during this process there are several ways of doing that, because we do compositing as well. So, you can kind of separate out your render into different passes and composite them together using compositing software or PhotoShop. Then once you are happy with the results, you transfer you lighting into a map and then do a final composite render.

For characters, usually we light a character just inside a game. So all the lighting is built into shaders. In terms of what kind of lighting model we use, usually it depends on the project need. Some projects are more focused on indoor ---- lighting then you really don’t need that kind of dominant light source like outdoor ---- would have. For those types of games, usually they use image based lighting more than the other ----. For outdoor ----, you will have a dominant light as your key light source. We also use image based lighting or any other type of technique like spherical harmonics or just a hemispherical light to simulate the environment terrain. To pick up all those environment lights field into your characters.

On top of that you have other like rig lights, which is a camera based light where you attached to the camera so you say the light is on a certain side of the character to give you the highlight. You have some other control to control how much you want to bias it to the edge and that sort of thing.

That’s pretty much how we do lighting here.

INVESTIGATOR:

What are your general reactions to this?

10_30_01:

I think it’s interesting. My first reaction was this would be in the natural light which you use to [pause] You use the same lighting as you use to light the environment, is that what it is?

INVESTIGATOR:

Yes.

10_30_01:

It doesn’t really match that well.
INVESTIGATOR:
So you would say that in natural mode, the lighting between the character and the environment doesn’t match? Or what do you mean by that?
10_30_01:
I don’t know what kind of technique you use but sometimes when you walk through the rooms of the environment you don’t really see the lighting change that much. Just the interaction between the lighting on the character to the environment, it’s not that noticeable. For instance, you can see here…
INVESTIGATOR:
So the lamp in the center room?
10_30_01:
Ya. You see here that you obviously have a dominant light source coming from this side but your character is not lit from behind [behind being the direction towards the light source in this case]. [pause] [participant interacts with demo] It would be interesting, since now you have the ability to mix between your preset lighting and your environment lighting, it would be interesting if you could set the scene, to set multiple lighting sources, if you can. Maybe one is from in this area maybe if you want to say the light is on you can have your light source put in there to light a character and then another in here, maybe you can set another light source, just another coming from in there or have some lights from this angle to simulate that lighting. And you can just lerp [linear interpolate] between them to get more interactive [pause] You’ll pick up the environment better.
INVESTIGATOR:
Is that all specifically the natural mode or is…?
10_30_01:
No, it’s not just natural mode, it’s when you have the auto mode which is kind of a blend between natural and your attached lighting. I’m just saying, for your attached to set up multiple light sources, you can use your attached lighting to do that. I don’t know exactly how you do this. But I would say just in general since you have this ability to blend [pause] You have a natural and an attached, you can use an attached
lighting mode to set up some multiple light sources and have your character interact, it would be more interesting.

INVESTIGATOR:

Would you describe this lighting as satisfactory? Just, overall.

10_30_01:

May I ask, what is the purpose of this problem? What is it that you want to demonstrate?

INVESTIGATOR:

The basic idea is the automatic control of lighting. This is meant to demonstrate it in a complex enough environment so you can sort of see what’s going on but there’s no sort of narrative to what you’re seeing here, there’s no story being told here. So, to whether lighting is satisfactory, it’s sort of in a general aesthetic sense. So, what I mean is the whole composition, the environment, the character, the whole thing.

10_30_01:

I like the idea but in terms of [pause] Just by looking at the lighting, I would definitely hope the character would tie into the environment more. In the sense that it was actually this environment would interact with the light from this environment. More than what it is now. [the 10_30_01 appears to be pointing out the lack of motivation to the environment lighting – light is not coming from the windows or the candles for example, it’s just a general sense of ambient lighting]

INVESTIGATOR:

What’s your interpretation of the image in the upper right corner?

10_30_01:

I assume it’s just an image that represents you’re attached lighting to show what that is doing. That’s my take on it.

INVESTIGATOR:

Anything else I haven’t brought up or you’d like to say?

10_30_01:

Yes, I think [pause] Do you mind if I ask how you do your environment light? I’m not quite getting…
INVESTIGATOR:
If you have any other impressions and then I’ll explain the whole thing to you.

10_30_01:
No. I would just say you can probably consider just [pause] No, we can talk about it. It’s just more about techniques that I would [pause] I can see that some of the techniques that we use in games can have actually [pause] You would benefit from those methods to get your character more tied in with your environment.

10_30_02
INVESTIGATOR:
Could you briefly describe your process as a designer for me?

10_30_02:
The process I just…
INVESTIGATOR:
Actually, just in general…

10_30_02:
What I do?
INVESTIGATOR:
Yes.

10_30_02:
Ok. I usually set up [pause] I usually look for reference first. So, what I want to achieve in the final target. Then I find what’s the lowest value [value refers to light intensity] that we’ll get in the scene. So I look for the shadow value. Then I go for the highest value. So I can have the contrast in the scene. And then knowing this I start thinking about the hues that I want in the image. And what’s my point of interest. And so on.
Then I split that in the tools. I will have an ambient light, and if I need a spot light, or if I need a specific light or fill light or bounce card or even a shadow. And then apply it.
INVESTIGATOR:

What are your general reactions to this?

10_30_02:

It looks it’s [pause] I mean I’ve seen one control, I think, so far. I’ve only experienced with this for a few minutes, but it looks like it’s all ambient lighting so it’s very hard to control where you get the accent on the character. For example, the first thing that I did was I went to the window to get some lighting from the outside. Because that is what I would do if I am taking a photograph of the character I will go right to the window and then get some bouncing coming from the indoor lighting. It’s going to give me some kind of shape on the character. So what I was missing here is, that’s why I was asking that the two lights that are attached to the character I would love to be able to move those to set up as I want. The best setup that you have, the room lighting cue 2, it’s better because you have some contrast. The characters are [pause] Even if it’s nighttime or very dark you get a nice shape of her face or what she’s wearing. #1 is too flat, like you can see that there is almost no shadows. It looks like an unlit environment. And #3, it gives you the warm cue that by nature we like because it looks classy and warm and comfortable but I think it’s still very, very flat. The one that gives you the most contrast is the second one which I like most.

INVESTIGATOR:

What’s your interpretation of the image?

10_30_02:

What I get from the image is like in an environment map, it’s maybe the translation I would do. It’s like a map that you would multiply on top of the whole character. For example, if you [pause] What I was looking for was if this [10_30_02 is pointing on image #2] has a dark area on the top, I will get that always on the top of the character and it’s going to be brighter going down. That’s going to be multiplied to the frame buffer on the character area. That’s what I interpreted.

INVESTIGATOR:

Any other thoughts on this that I haven’t mentioned specifically?
10_30_02:
The refresh is very important especially when you’re working with applications for lighting. Even if you know that you have to move the character to get the feedback…
INVESTIGATOR:
Oh, so you mean the feedback of…
10_30_02:
What you are doing, ya. ‘Cause lighting is very subtle, sometimes you move something a little bit and then if you have to move the character to get that feedback that actually make you lose a little bit where you were. So that makes the workflow bumpy. That’s one of your biggest problems when you’re doing lighting, you have to be [pause] The tool has to be very, very robust to get a good process.

10_30_03
INVESTIGATOR:
Could you describe briefly your process as a designer, in general?
10_30_03:
Can you be more specific?
INVESTIGATOR:
You could pick a project you worked on and walk me through it, like how you go about [pause] What would you call yourself?
10_30_03:
I’m a lead lighter.
INVESTIGATOR:
Lead lighter. You could pick a project and walk me through how you would go about lighting for that project.
10_30_03:
Ok, do you mean specifically lighting [pause] The reason I ask is because my job has to cover a lot of different areas. Do you mean lighting process for an environment, do you mean character lighting, do you mean shaders, do you mean workflow?
INVESTIGATOR:
What would be the most interesting for you to describe?

10_30_03:
Any of them, they’re all equally [pause] You pick one, this is for your research.

INVESTIGATOR:
Ok, so character lighting.

10_30_03:
For character lighting the way we would go about it is obviously designing the various features so you’d obviously want to pick out your most important parts, pick out the overall look that you’re going for and the type of mood, type of energy that you want the characters to portray. Pick out the main features that can then portray that, the contrast levels, the kind of detail, do you want cartooney, do you want realistic and then break that down from there as to how you can actually achieve that. So, for instance right now I’m on ----, so we’re going for a very realistic look, so then it’s really up to how realistic do we really want to go and how can we achieve that, ok then let’s break that down even further. We want really realistic skin, we want realistic hair, fabrics, blood, reactions, all those kinds of things. So then how can we get the texture to feed into that properly, how can we develop the shader to get all those pieces together. For instance, because we’re in video games, how can we actually get something realistic that still runs at 60 frames per second. And basically designing all of that, and then it, obviously, once we have the technology, then actually putting it together and then feeding back in all of that more artistic style of how we want it to feel vs. just how we want it to look.

INVESTIGATOR:
Thank you. What are your general thoughts about this?

10_30_03:
This is quite interesting. What application would this [pause] Are you thinking?

INVESTIGATOR:
Let me get to that a little bit later.
The reason I ask is I’m just trying to organize my thoughts and I guess give you the information...

INVESTIGATOR:

What I’m really for is what are you interpretations of what’s going on here, what are your gut reactions to what you’re seeing happening?

Gut reactions? It’s definitely very interesting. It’s [pause] The environment lighting and the character lighting, I like how they’re actually integrated but still separated, that works nicely. The one things I was kind of wondering about are, for instance, I noticed that the key light constantly follows the character vs. being a stationary point that the character could go in and out of or could interact with in some way.

What I see right now is actually [pause] We have a key and a fill, I don’t know what kind of lights they would be, but clearly the key light would be a directional light that’s why it’s following the character.

I’m wondering [pause] I wonder how one of the key missing features out of the total lighting model would be an ambient occlusion, both in the environment and on the character and I’m wondering how that could potentially be integrated, especially dynamically. The way the character moves around, casting shadows and not just the environment but sort of, giving light but also taking away light, which is a very key part of a very realistic lighting model. Or any lighting model for that matter.

I definitely can see the potential, so those are kind of my initial impressions.

INVESTIGATOR:

Would you describe the lighting as satisfactory?

You mean would [pause] I guess, define satisfactory. It wouldn’t be good enough to put into a game, no. It’s too simplistic, like I said it’s missing key features like ambient occlusion. This is secondary, it doesn’t have any specular. The environment lighting is a little too flat. Well, a lot too flat. The kind of lighting that we do, especially because we do baked in lighting, the level of fidelity that we can hit is very close to reality. Like often times you can actually put a photograph side by side with
our in game lighting and there won’t be too many differences. This is much flatter, especially in the environment. The character would need a lot more light sources. For instance, with our characters we use two methods to kind of get everything together. We have our environment or hemisphere lighting together with real-time lighting and then real-time shadowing and ambience, specular, all those kind of things, to give all those different features to the characters and to the environments and really push their reality.

INVESTIGATOR:

What’s your interpretation of the image?

10_30_03:

Of the…

INVESTIGATOR:

Ya, the thing that’s surrounded in the magenta color.

10_30_03:

I would expect that would be a spherical representation of what the lighting’s doing. So a hemisphere lighting solution [pause] Not lighting solution but if I was viewing hemisphere lighting it would be in a similar format. So it’s basically where the lighting is coming from if you were looking at it on a globe.

INVESTIGATOR:

Any other thoughts that I haven’t brought up or mentioned?

10_30_03:

Not so far. How would this integrate because obviously this is directly through special software to be able to run it, how would this actually extract out to integrate into for instance another pipeline or another [pause] a full game engine for instance?

INVESTIGATOR:

[pause] So, no other thoughts from what you’ve seen so far?

10_30_03:

Not at the moment. It might come up when, depending on…

INVESTIGATOR:

Well, I just want to check.
INVESTIGATOR:
Could you briefly describe your process as a designer?
10_30_04:
As a lighter?
INVESTIGATOR:
As a lighter. Or if you would prefer to talk about [pause] Your expert background that you prefer to describe.
10_30_04:
So my process?
INVESTIGATOR:
Mmmm-hmmm.
10_30_04:
Well, for here at work we get ---- targets for our art director and then we have to go through, particularly in gaming, we have to pull it in and look at it, just a sort of diffuse, make sure everything’s working in a diffuse shaded mode. The textures, make sure they’re all aligned and everything. And then we do a basic lighting pass, and then we get the art director to look at it and we tweak things. And as a lead I have to help other people out, make sure their lighting kind of follows the art direction.
INVESTIGATOR:
What’s your general reaction to this?
10_30_04:
It’s cool, we’re trying to get some interactive lighting. We’re not going about it the same kind of way but it’s cool. I like the …
INVESTIGATOR:
Could you go into that a bit more specifically? As in, what are you specifically noticing that’s different [pause] That you said you’re not going about it the same kind of way?
10_30_04:

Well, because of memory constraints and stuff we have certain lights that are
dynamic and other ones that are baked. So, using spherical harmonics or whatever we
bake in the fill or we bake in two lights and we have one light [pause] Our key light
we’ll try to get dynamic or our rim light we’ll always keep dynamic. And then other
things will be baked per environment. So that we get a nice mix of shadow plus
occlusion [occlusion in this case refers to ambient occlusion. Several participants
appeared to use “occlusion” to refer to ambient occlusion and to infer the general
global illumination effect of ambient occlusion, not the literal definition of occlusion,
which results in a loss of light] and everything.
That’s one thing that I found that the darks are really dark. Like you’d never get
blacks that black on the character, that kind of thing. So you need some sort of
occlusion in there.

INVESTIGATOR:
Would you describe the lighting in this environment as satisfactory?
10_30_04:
Not for what we do. Just the lighting quality? No.

INVESTIGATOR:
Could you be a little more specific?
10_30_04:
It’s flat, there’s areas that are flat and things that are too dark. Stuff is blowing out
[10_30_04 is referring to areas where the color value is >1 and is being clamped
against the max value of low dynamic range displays], the textures are blowing out.
The textures are [pause] We would go in an equalize the textures when we bring
things in, the saturation and the contrast could be better, those are all things that we
do to balance it first. First of all, we do a texture pass and make sure everything is in
balance with everything and that the contrast is correct for the texture that you’re
seeing or using. We bring everything in and make sure it has proper Gamma
correction on it so that it doesn’t [pause] So that we have linear lighting. That’s one
thing that we’re doing a big push on so things don’t blow out in the world. So your
lighting is true. So your lighting isn’t…does that make sense?
INVESTIGATOR:
Ya..
10_30_04:
I’m not an SE so…
INVESTIGATOR:
That makes perfect sense and I understand what Gamma correction is. What’s your interpretation of the image in the upper right corner?
10_30_04:
Interpretation?
INVESTIGATOR:
What do you think it means?
10_30_04:
It’s just lighting direction, I assume where the key light’s coming from. Is this what that is?
INVESTIGATOR:
Well…so, any other thoughts that I haven’t mentioned or brought up?
10_30_04:
No.

10_30_05

INVESTIGATOR:
Could you describe your typical process as a lighting designer?
10_30_05:
I actually am not a lighting designer at this point. ---- I’m an art director.
INVESTIGATOR:
Well, I guess [pause] Pardon me. What I should say is [pause] Take an expert process that you’ve done in the past or that you’re particularly fond of and lay it out for me.
One of the normal processes we go through is to gather or to select a whole bunch of ----, ---- reference images. And depending on what kind of game it is we’re working on you look at your ---- reference and we look at what the game engine can actually support in terms of technology, what type of lighting it can support. And then we set up and try to [pause] If our goal is to replicate the ----, we’ll analyze difference methods that we’ve got on hand to generate that lighting or to simulate that lighting ‘cause a lot of our games are ----. So, we will generally go and get our ----, break down the ---- target into what the different components of lighting are and then try and map those to the different techniques that we’ve got available to try and recreate that type of lighting.

INVESTIGATOR:
What are your general reactions to this?

It’s a bit hard to see [pause] It’s a bit disconcerting because some of the rooms you’ve got pretty ambient lighting on them, so when you’re traversing through [pause] It’s hard to kind of make judgment calls on the lighting because the lighting doesn’t appear to have any direction. That’s why when I was stepping back and forth, I was trying to figure out where’s the ambient light, ‘cause if you look at her [pause] If you look at this, as it’s set right now, the ambient light is so wrong for this room because this room obviously is pretty flat lit. I guess what I’m looking for is how well does she fit within the lighting in the context of the room.

So, obviously when you go to the ambient mode and you start putting what I think is the key light following with the camera, that’s kind of very documentary style way of illuminating things from a kind of television point of view. That works most of the time. For me, it’s a safe way of lighting things because you know your character is going to be illuminated in a reasonable way most of the time. I think the problem with this is it’s so hard to discern or determine the light sources ‘cause there’s obviously potential light sources in this environment like this one up here in the dining room but that seems to theoretically give off a lot of light but it’s not really mapping through to some of the rooms. So it’s kind of hard to see. If that really was giving off hat much light, it wouldn’t fall off this way so it’s kind of funky.
INVESTIGATOR:
To kind of expand on that a little bit, would you describe the lighting in this environment as satisfactory?

10_30_05:
Nope.

INVESTIGATOR:
Ok. So you’ve mentioned quite a bit about that, anything else? About why it isn’t satisfactory?

10_30_05:
It’s not satisfactory ‘cause if the light was [pause] For me as an art director, I’m always looking for what is the source of the light and what’s the logic behind it. If this is a simulation, and the textures used in this indicate that this is a simulation because they’re photorealistic textures, so when you look at that you basically say, ok I’m looking at something that’s supposed to be real. So you’re looking for the logic and the validation in the lighting when you look at the content. So, if I was to be analyzing this room or these set of rooms, the only real logical source of light appears to be the chandelier light. This room [10_30_05 is referring to the middle room] has a light in the middle of it but I don’t know where that comes from. Like there’s no visible source for it. So, that’s when I’m art directing things and I see the things that don’t make sense to me in this kind of context I start to wonder well where is the light coming from? If [pause] And it’s not necessarily just a lighting thing in here it’s more [pause] It’s potentially a material thing as well. If you put her in this angle here and you have that kind of light there you would have a rim light around the back and there’s no rim light. It kind of starts to fall to pieces physically, sorry to burst your…I’m guessing you want real hard core feedback on it.

INVESTIGATOR:
This is exactly what I want.

10_30_05:
To be critical of the lighting in the room there’s no ambient, there’s no AO style [pause] We’re very big on AO…
INVESTIGATOR:

AO being Ambient Occlusion?

10_30_05:

Ya. ----. So this is lacking that [pause] It feels like it’s lacking that level of physical, I
know it’s not real [10_30_05 is referring to AO not being physically accurate], but it
feels more real when you have that blended in there.
So for example these [indecipherable] are very ambiently lit or very flat lit but there’s
no light source to do that.

INVESTIGATOR:

What is your interpretation of the image in the upper right?

10_30_05:

I’m not really sure what you’re trying to indicate there [pause] If it’s a direction of
light source because it can be connected and can be disconnected from what the
lighting in your scene is, then it breaks the logic for what it’s trying to tell me. I think.
I got confused when I was switching back and forth between the different modes as to
what this was actually trying to do. I guess [pause] I’m familiar with lighting things
in applications that have just a sphere to tell you where the light source is [pause] I’m
kind of used to seeing these kinds of things and I would expect that to be an uplight,
kind of very broad. Key light, rim light.

INVESTIGATOR:

Any other comments or anything else that hasn’t come up?

10_30_05:

Not really. I’m interested to know what you’re trying to do here.
Curriculum Vita: Joseph A. Zupko

Education

- Ph.D. in Information Sciences and Technology, May 2009, The Pennsylvania State University
- B.S. in Computer Science, May 2004, The Pennsylvania State University

Teaching

- IST 402: Game Design and Development, Teaching Assistant
- IST 412: Engineering of Complex Software Systems, Teaching Assistant

Journal Papers


Conference Papers

- Yucel, I., Zupko, J., Seif El-Nasr, M. 2006. Using Game Modding to promote and provide basic IT skills to a female audience. Games, Learning, and Society.