A Psychophysical Investigation of Global Illumination Algorithms Used in Augmented Reality

by

Timothy John Hattenberger

B.S. Rochester Institute of Technology, 2000

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Chester F. Carlson Center for Imaging Science
Rochester Institute of Technology

March 27, 2006

Signature of the Author _____

Accepted by _____
Coordinator, M.S. Degree Program Date

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Timothy John Hattenberger
has been examined and approved by the
thesis committee as satisfactory for the
thesis required for the
M.S. degree in Imaging Science

Mark D. Fairchild, Thesis Advisor

Carl Salvaggio

Garrett Johnson

Date

THESIS RELEASE PERMISSION
ROCHESTER INSTITUTE OF TECHNOLOGY
CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE

Title of Thesis:

A Psychophysical Investigation of Global Illumination Algorithms Used in Augmented Reality

I, Timothy John Hattenberger, hereby grant permission to Wallace Memorial Library of R.I.T. to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Signature _____

Date

A Psychophysical Investigation of Global Illumination Algorithms Used in Augmented Reality

by

Timothy John Hattenberger

Submitted to the
Chester F. Carlson Center for Imaging Science
in partial fulfillment of the requirements
for the Master of Science Degree
at the Rochester Institute of Technology

Abstract

Global illumination rendering algorithms are capable of producing images that are visually realistic. However, this typically comes at a large computational expense. The overarching goal of this research was to compare different rendering solutions in order to understand why some yield better results when applied to rendering synthetic objects into real photographs. As rendered images are ultimately viewed by human observers, it was logical to use psychophysics to investigate these differences.

A psychophysical experiment was conducted judging the composite images for accuracy to the original photograph. In addition, iCAM, an image color appearance model, was used to calculate image differences for the same set of images. In general it was determined that any full global illumination is better than direct illumination solutions only. Also, it was discovered that the full rendering with all of its artifacts is not necessarily an indicator of judged accuracy for the final composite image. Finally, initial results show promise in using iCAM to predict a relationship similar to the psychophysics, which could eventually be used in-the-rendering-loop to achieve photo-realism.

Acknowledgements

Thank you Dr. John Schott for providing me with amazing experience and knowledge, and allowing me to 'make pretty pictures' for my thesis. Thank you Dr. Mark Fairchild, Dr. Garrett Johnson, and Dr. Carl Salvaggio for your knowledge and guidance throughout this process. We got some great work done at the fairway office. Mark and Garrett, I will see you Scotland! Carl, the phrase 'reinvent yourself' will stick with me forever. Thanks to everyone in the DIRS and Munsell groups. I have made such great friends over last four years, but am tired of feeling so dumb around all of you. Thanks especially to Adam G., Willy, Lawrence, Emmett, Kremens, Scott, Niek, Nate (ring-deltas), Kate, Cindy, Val, and observers (expert and naive). Kris, Deb, and Lainey, thank you for everything you have done for us, especially for Reagan. Tony and Julia, you have given so selflessly of yourselves to help us. Mom and Dad, you have been so incredibly supportive from day one of my undergraduate right through my thesis defense. I realize and appreciate the sacrifices you have made to help me get so far. You have set a high mark as parents and I will strive to do the same for my children. Lauren, thanks for always having a smile on your face. Ed, you were my best friend growing up, let's stay that close. Thanks for the help and encouragement throughout my thesis ('just get it done'). Thank you Mark and Maggie for being so supportive and unselfish and for treating me like family since day one. Chris and Mark, thanks for treating me like a brother. Rachel, my wife, best friend, and legal counsel. Anyone who knew what I was like during finals time will know what you have had to put up with for the past four years straight. You have given so much of yourself to me, the least of which our beautiful daughter, Reagan, and now a baby boy. Thanks for making life so much fun. I love you and am so proud of you! Reagan, thanks for changing my life in the best possible way. Thank you God for blessing me with so much in my life and helping me focus on only the most important things.

Thanks to Kodak for the Seed grant that assisted in funding this research.

This work is dedicated to God, my wife Rachel, and my children Reagan and the little guy.

Contents

List of Figures

Chapter 1

Introduction

Computer graphics and vision research have become more intimate in recent years. It seems logical as a human observer is typically the final discriminator of the output imagery. One author even remarks that "[A]t Microsoft Research, the computer vision and graphics groups used to be on opposite sides of the building. Now we have offices along the same hallways, and we see each other every day" [34]. This area of research is currently of great interest. Perhaps as interesting is that it spans areas including, but not limited to, computer science, vision science, biology, digital image processing, perception, and psychophysics. The application of all of these areas amounts to a *system-level* approach which lends itself to a discussion using the 'imaging-pipeline'.

1.1 Computer Graphics and the Rendering Pipeline

The idea of the imaging pipeline is depicted in Figure 1.1. In this example, the input is a scene in the world, realizing the input could be purely imaginary as well. This could be an object in a light booth or a cluttered desk in an office. Typically this scene is imaged onto film or a CCD

in a digital camera. In order to create a synthetic image of the scene, certain characteristics must be acquired from the real scene. These include surface reflectances, object dimensions and global positions, radiometry and geometry of light sources, and camera geometry and characteristics. For the purposes of this discussion, assume that these measurements can be made with relatively high accuracy (unless otherwise stated). These parameters are input into a rendering engine, typically a global illumination solver. As the name implies, this software calculates the amount of light leaving the sources, interacting with the objects, and entering the camera to produce a final radiance image. (Camera models can be used to modulate this sensor-reaching radiance). The radiance image typically contains values (high-dynamic-range) that cannot be reproduced on a normal display device such as a CRT or LCD (low-dynamic-range). The radiance image must be *tone-mapped* in order to be displayed. The tone-mapped image is presented to a human observer and using a psychophysical technique compared to the real scene, or photograph of the real scene. Based on the results of these experiments, the parameters can be tweaked and the scene re-rendered or, ideally, the synthetic scene is perceived as indistinguishable from the real image of the scene and declared *realistic*.

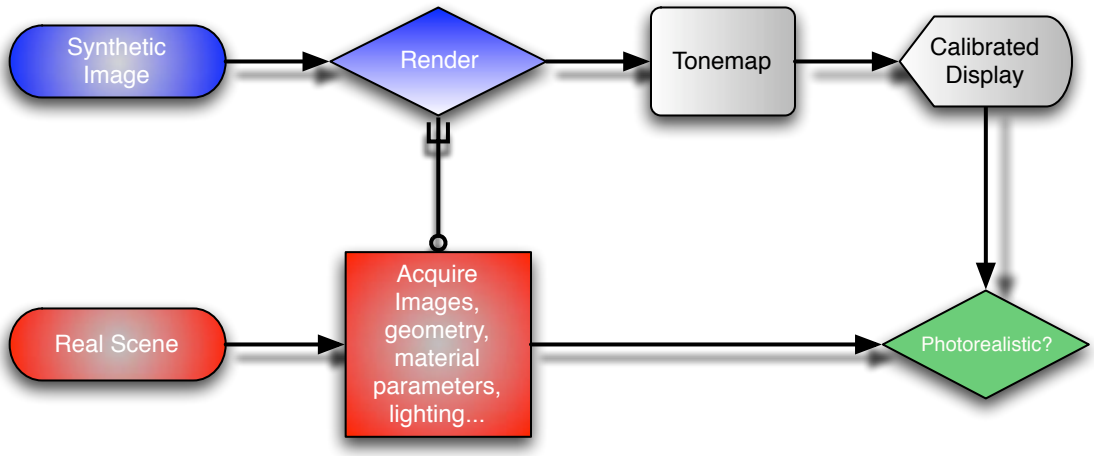


Figure 1.1: Photo-realistic Image Generation Pipeline

The specifics of each of the elements in this pipeline depend on the desired output as different applications warrant different results. As stated above, the ideal output is an image that is realistic. This begs the question, *What is meant by realistic?*

1.2 Realism in Computer Graphics

James Ferwerda of Cornell wrote a paper [20] that is meant to serve as a framework to help define realism in the context of computer graphics. As stated above, the final output of the rendering pipeline is a synthetic image. It is important to realize that an "image is a visual *representation* of a scene, in that it "re-presents" selected properties of the scene to the viewer with varying degrees of realism" [20]. The driving force for computer graphics has long since been about creating realistic images. Much of the past research used in this thesis starts with near identical discussions about this goal to create realistic images. As [20] states, and is realized in the literature, the need for realistic image synthesis is often questioned because there is not a standard set of metrics to delineate realism. The idea being that there is ambiguity in the term realism itself, and really three types of realism in computer graphics can be described: physical realism, photo-realism, and functional-realism.

Physical realism results in an image that "provides the same *visual stimulation* as the scene...this means that the image has to be an accurate point-by-point representation of the spectral irradiance values at a particular viewpoint in the scene...and is overkill if one's job is to create images for human observers" [20]. This is essentially the driving force behind DIRSIG, the Digital Imaging and Remote Sensing Image Generation tool, created within the Digital Imaging and Remote Sensing Laboratory at RIT [8].

Photo-realism "requires that the image has to produce the same *visual response* as the scene even though the physical energy coming off the image may be different than the scene...this

criterion allows us to take advantage of the limitations of vision to simplify the task of making realistic images" [20].

Ferwerda discusses valid pros and cons about adopting photo-realism as the criterion. However, it seems that this is currently many researcher's goal. He proposes functional realism, whose requirement is to provide the same *visual information* as the scene, and ideas of metrics for functional realism.

The idea of photo-realism is intriguing due to the intimate relationship between the human visual system (HVS), image synthesis, and image evaluation. For photo-realism, the HVS has been considered in the rendering, tone mapping, image metrics and image comparison. The primary focus of this research is dependent on the idea of photo-realism, and to a lesser degree, physical realism, and the relationship, if any, between the two.

1.3 Goals of this Research

To this point only purely synthetic images have been considered. There is however, a lot of research and interest in images that contain both real and synthetic components. Rendering synthetic objects into real images is sometimes referred to as augmented reality (AR). This idea is realized in most recent movies that combine computer generated images (CGI) with real human actors or scenery. In this case, the final composite image must meet the requirements of photo-realism as described in section 1.2.

This general idea of augmented reality will be used in this research. Several goals of this research are introduced here at a high level. A synthetic object will be rendered into a real scene. This will be performed in an environment (*e.g.*light booth), where parameters can be measured and controlled. Then, relationships between physical realism and photo realism will be explored through the use of radiometry, colorimetry, and psychophysics.

Another goal is to use an image appearance model to predict image differences. These results will be compared to human observers through the use of psychophysical experiments. While image appearance models may not fully replace human observers, they may provide an extremely useful tool in image quality and realistic image generation.

Chapter 2

Background

2.1 Synthetic Image Generation

This section addresses the process of synthetic image generation (SIG). Section 2.1.1 describes the theoretical background of SIG. It describes, at a high level, the interaction of electromagnetic radiation with matter in the world that eventually reaches a given sensor. Section 2.1.2 introduces the rendering process, and gives a high level introduction to global illumination algorithms. Finally, there is a discussion of `pbrt`, a physically based ray-tracer, that was used in this research in Section 2.1.3.

2.1.1 Radiation Propagation and Material Properties

Whether the goal of SIG is to calculate physically accurate radiance values, or create a visually pleasing stimulus, it is dependent at some level on radiation propagation. This section is not meant to be a complete, low-level description of all of the physics and optics necessary to describe these interactions. Rather it introduces the reader to the physical processes that must

be considered when developing reflection and illumination models. Most of these processes are captured in one or many of the available simulation tools at levels ranging from rigorous physical solutions, to approximations based on empirical results. Even so, a basic understanding of the physical phenomena should allow rendering parameter tuning to provide expected results.

The author must assume the reader has a knowledge of basic radiometry and physics. For a more thorough explanation of these phenomena and phenomena not discussed here, please consult [24], and associated references or [27], which starts with Maxwell's Equations to derive them.

Our current understanding of light allows it to be described both as a wave and as particle. The wave description is most useful when discussing computer graphics, as it describes the optical interactions of most concern. *Light* used in the context of this document refers to the visible portion of the electromagnetic (EM) spectrum, with wavelengths on the order of 0.4 to 0.7 [μm]. The wavelength of light corresponds roughly to the perceived hue, where 0.4[μm] is violet, and 0.7[μm] is red. It is a transverse wave containing both an electrical field and an orthogonal magnetic field. This allows electromagnetic radiation to propagate using itself as the medium, and thus has a speed of approximately $3x10^8$ [m/s] in a vacuum. The reflective portion of the EM spectrum refers to the visible (VIS) and short wave infrared (SWIR), roughly 0.3 to 3.3 [μm], and the thermal portion of the EM spectrum refers to the long wave infrared (LWIR), approximately 8-14 [μm]. The reflective region is characterized by the sun or some other external source as the dominant source of radiation, and the intensity of energy on surfaces is determined by the interaction of these sources with the surfaces. The thermal region is the region of the EM spectrum where the dominant source of photons is due to thermal self-emission [51]. While these terms will arise in the following discussion of the underlying physics and DIRSIG, the primary focus will be on *light*. The reason for this is that the images rendered as part of this research will be presented to human observers, where the

eye's sensitivity is in the VIS.

Interaction of Light and Matter

A robust model must account for all sources of photons, and the way in which they interact with materials in the environment. These interactions are governed by physical processes. As stated in [26], "[T]here are two illumination phenomena of major importance in generating imagery. The first is the interaction of light with the boundaries between materials. The second is the scattering and absorption of light as it passes through the material."

There are two general categories of materials that light can interact with: dielectrics (or insulators) and conductors. As indicated by the name, these materials are described primarily by their electrical properties. Dielectrics include materials such as glass and quartz crystals, which are largely transparent in the VIS to EM radiation. Their electrons are in stable orbits that are not affected by passing light. The main interaction of dielectrics with light is caused by the decrease in speed of the light wave. This causes an incident wavefront to change direction as it enters the material, provided it was not normal to the surface upon entering. This phenomena is described by Snell's law:

$$n_1 sin\theta_1 = n_2 sin\theta_2 \tag{2.1}$$

and is shown in Equation 2.1, where n is referred to as the index of refraction and is the ratio of the speed of light in a vacuum to the speed of light in the material. As light travels the fastest in a vacuum, this number will always be greater than 1.

Conductors, as their name implies have electrical interactions with EM radiation. This is due to the fact that there is a "sea of electrons" in loosely bound orbits. When light strikes the surface of a conductor, the $\overrightarrow{E}$ field causes the electrons of the conductor to oscillate and reradiate

the incident radiation in the form of reflected light. The electrons are not completely free to oscillate and can be thought of as being tethered by a dampening spring. This has the effect of absorbing some of the incident energy in the form of heat. Thus the reflected light is less than the incident light. In order to characterize this behavior, two new variables are introduced: the complex index of refraction k, and the absorption coefficient. "The absorption coefficient k/n, is a measure of the absorption characteristics of a conductor" [26]. Hall also provides some useful approximations of Fresnell's equations for conductors, and plots of several Fresnell relationships.

Regardless of the material, when light reaches a boundary or interface between two materials, one of two events occurs, it is reflected and/or transmitted. The Fresnell Equations [27], which are derived from Maxwell's equations provide a relationship of the percentage of the light that's reflected and transmitted. These equations are based on the indices of refraction of the two materials, and yield results in terms of the polarization states parallel and perpendicular to the plane of incidence. Hall [26] states that "[F]or the purposes of image synthesis, it is convenient to assume light is always circularly polarized, and that interactions are characterized as the average of the perpendicular and parallel components of polarized light." Energy conservation dictates the transmitted energy is simply 1 minus the reflected energy.

The angle at which the reflected component leaves the surface is a function of wavelength, surface roughness, and incident illumination angle. If the surface is a perfect mirror, for example, all of the incident energy will be reflected off of the surface into an angle equal to the negative, relative to the surface normal, of the incident angle. This is referred to as specular reflection. However, as the surface becomes rougher, incident light is scattered into more directions about the specular direction, until a surface is perfectly diffuse, or Lambertian and reflects light equally into all directions. Therefore it is not enough to specify a single number r that represents the surface reflectance if the surface is something other than Lambertian. The

reflectance r is thus a function of all combinations of input and output angles.

The function that describes this is called a bidirectional reflectance distribution function or BRDF. Figure 2.1 from [51] shows representative BRDFs for materials that range from perfectly diffuse to perfectly specular. "When we care only about opaque surfaces in vacuum (or homogeneous air which we are content to treat as vacuum), then we need only find a description of the... BRDF at each point on each surface. If the surfaces are partly transparent then the bidirectional transmission distribution function (BTDF) must also be considered. Together, these functions form the *bidirectional scattering distribution function* (BSDF)" [24]. All of these functions can be thought of as three-dimensional probability distribution functions. They specify the direction light is most likely to take based on the various combination of input and output angles. Hall and Glassner give generalized illumination expressions that account for coherent reflection and transmission, bidirectional incoherent reflection and transmission, diffuse reflection and transmission, and the emissivity of a surface.

It is important to understand that aggregate behavior may not be representative of the material. Hall gives several examples to illustrate this point. As stated above, many dielectrics appear transparent to light, like glass. However, if one looks at a pile of crushed glass, it will look white. On the microscopic scale, each little piece of glass is still as transparent as the original sheet. At a macroscopic scale however, the glass appears white due to scattering (*i.e.* all wavelengths are scattered equally into all directions). This is the same phenomenon observed in clouds and the blue sky. The blue, rather than white color of the sky is due to the fact that there is a wavelength dependence of scattered light due to atmospheric constituents. This change of observed behavior with scale introduces another level of complexity that must be captured. More information about scattering and optical depth can be found in [51].

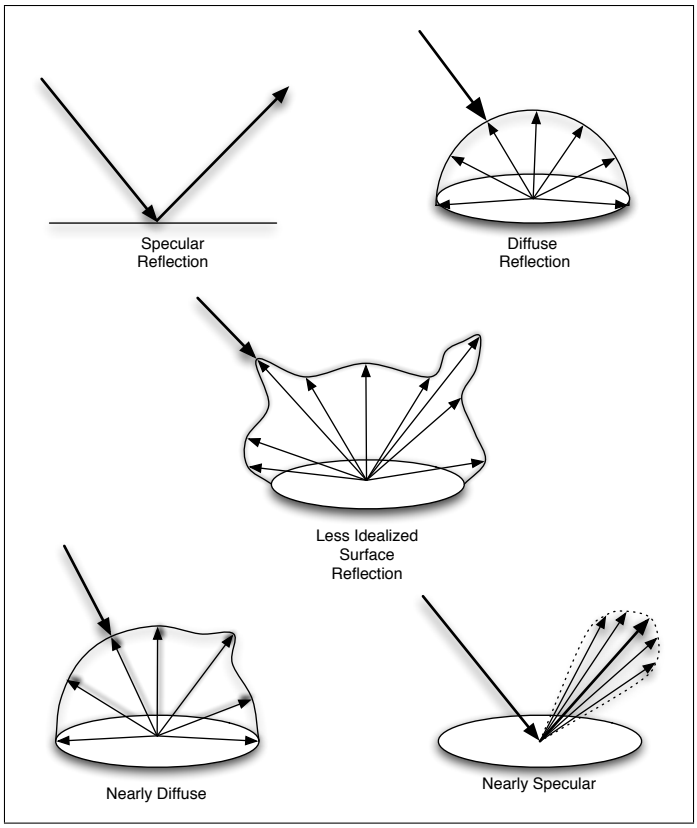


Figure 2.1: Reflectance characteristics for idealized surfaces [51]

The Big Equation

Now that the behavior at surfaces has been introduced, mathematical descriptions can be used to follow energy from the source to the sensor. *The big equation* is a term that Schott [51] uses for the governing equation for the spectral radiance reaching the sensor that is well suited for remote sensing. However, the equation is complete, and though the numerical methods may change, the physics remains the same whether the source is the sun or a desk lamp. This section will serve to introduce the energy paths accounted for and their corresponding terms in the *big equation*. A detailed explanation is given in [51].

Figure 2.2 shows all of the energy paths accounted for, and they are referred to as Types A through I photons. Generally the radiance reaching the sensor is given by Equation 2.2. As alluded to above, there are solar (or source energy paths), and thermal paths. The solar paths are shown in blue, and the thermal photons in red. As indicated by their names, the solar photons originate at the sun and interact with the world. The thermal photons can originate anywhere an object is above absolute zero Kelvin. They emit photons according to the Planckian blackbody function modulated by the emissivity of the surface, which is a complimentary term to reflectance (*i.e.* $\varepsilon = 1 - r$) for the thermal portion of the EM spectrum. Strictly speaking, the sun also emits photons according to the black body equation, however due to its temperature, the peak wavelength is in the visible and falls off very quickly in the SWIR and MWIR.

Equation 2.2 shows the total radiance, L, as the sum of all of the radiance due to photon paths A through H, where multiple bounce photons have been ignored. The definitions of the photon paths are given below.

$$L = L_A + L_D + L_B + L_E + L_G + L_H + L_C + L_F \tag{2.2}$$

A Direct solar photons originate at the sun (or source), pass through the atmosphere, interact with the material (reflect), and are redirected through the atmosphere back towards the sensor.

B Sunlight or skylight photons originate at the source and are scattered by the atmosphere towards the target, reflect off of the surface, and continue through the atmosphere towards the sensor. These photons account for light in the shadows of A type photons. As atmospheric constituents increase, there is an increased probability of scattering, and thus it is likely to have a higher ratio of B to A type photons.

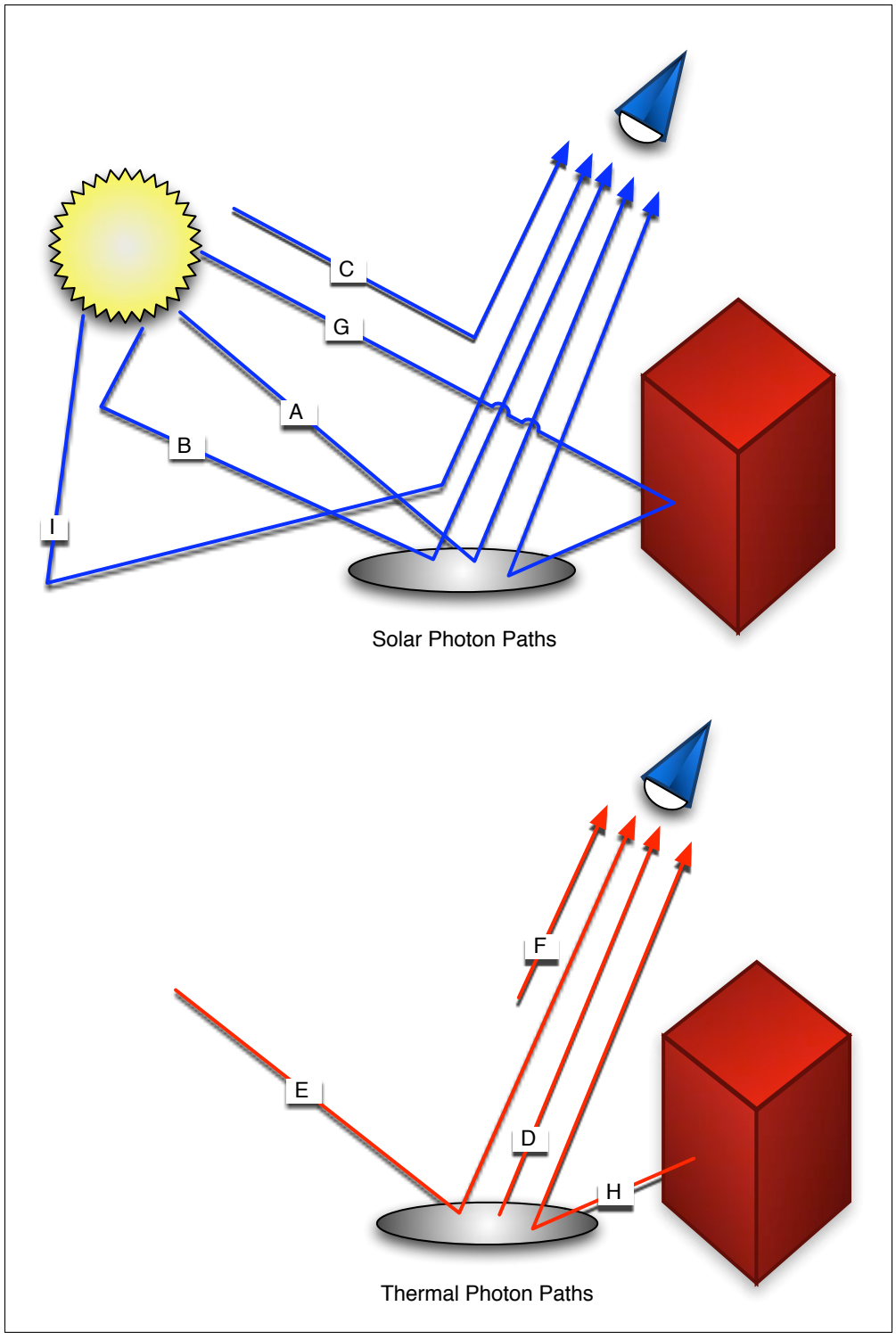


Figure 2.2: Photon Energy Paths in the *big equation* [51]

C Upwelled radiance photons originate at the sun and are scattered by the atmosphere at the sensor, without ever reaching the target. These photons therefore carry no information about the surface, and thus reduce the contrast of the image. They can be thought of as flare.

D Self-emitted thermal photons that originate at the target and propagate through the atmosphere to the sun.

E Thermal downwelled photons are emitted by the atmosphere due the fact that it has a temperature, are directed towards the target and again are direct back towards the sensor.

F Thermal upwelled photons that propagate directly from the atmosphere towards the sensor, again reducing the contrast as they contain no target information

G Solar background photons bounce off of background object before hitting the target, and then head back towards the sensor. As Schott says "... whether multiple bounce photons are important depends on the sensitivity of our measurements"

H Thermal background photons originate at some background object and then bounce off of the target, through the atmosphere to the sensor.

I Multiple bounce photons are scattered by surrounding objects, and then into the line of sight of the sensor without ever having reached the target. This phenomena is called the *adjacency effect*. As Schott says, if the background reflectance is slowly varying, then these photons can be considered part of C type photons. From the diagram, it can be seen that these photons contaminate the target radiance with non-target radiance.

Equation 2.3 expands each of the terms given in Equation 2.2.

$$L_\lambda = \Big\{ E'_{s\lambda} \cos\sigma' \tau_1(\lambda) \frac{r(\lambda)}{\pi} + \varepsilon(\lambda) L_{T\lambda} + F[E_{ds\lambda} + E_{d\varepsilon\lambda}] \frac{r_d(\lambda)}{\pi} + (1-F)[L_{bs\lambda} + L_{b\varepsilon\lambda}] r_d(\lambda) \Big\} \tau_2(\lambda) + L_{us\lambda} + L_{u\varepsilon\lambda} \tag{2.3}$$

where:

L_λ	Sensor reaching spectral radiance
$E'_{s\lambda}$	Exoatmospheric solar irradiance
σ'	Angle from target to the sun
τ_1	Atmospheric transmission from sun to the target
τ_2	Atmospheric transmission from target to sensor
r	Target reflectance
ε	Target emissivity
$L_{T\lambda}$	Self-emitted thermal radiance of target at temperature, T
F	Shape factor - Fraction of hemisphere above the target which is sky
$E_{ds\lambda}$	Solar downwelled irradiance
$E_{d\varepsilon\lambda}$	Self-emitted downwelled irradiance due to atmosphere
$L_{bs\lambda}$	Background radiance from scattering
$L_{b\varepsilon\lambda}$	Self-emitted background radiance
$L_{us\lambda}$	Solar upwelled radiance due to atmospheric scattering
$L_{u\varepsilon\lambda}$	Self-emitted upwelled radiance due to atmosphere

Chapter 4 in Schott's book gives derivations and detailed explanations of these variables. Most of the variables in the equations are the result of simplifications. For example the reflectance,

r, represents the full BRDF or the target. Additionally, the transmission terms, τ, are the comprised of scattering phase functions and optical depths (accounting for absorption and scattering) due to the atmosphere. As stated above, this equation is driven by the application to remote sensing, and as such does not account for radiation paths from man made sources and the moon and stars. These paths are accounted for in `pbrt`

The *big equation* is essentially the rendering equation referred to in the graphics community. The goal is to account for all energy paths, both self-emitted and reflected. The *big equation* is perhaps more general in that it captures the thermal energy paths explicitly. Derivations and detailed explanations of the rendering equation can be found in Kajiya's seminal work [33], as well as [25], [45], [26].

Conclusion

Reiterating, this section was not meant to be a complete rehash of physics and optics. In fact, not all processes were described, including fluorescence and phosphorescence. The intention was to quickly brush up on some of the important phenomena and to give the reader a sense of how quickly the problem expands if it is to be modeled by first principles physics. One thing to keep in mind is that in order to fully describe a surface in terms of physical processes, numerous properties must be known. It is often difficult to measure or find databases of materials. When the user does not have access to all of the required parameters, as is often the case, certain approximations must be made, often in the form of parameterized surface models. These approximations vary by application. Another thing to note is that modeling using a first-principles approach allows the simulation of phenomena such as rainbows, fog, and orange sunsets without the need for a talented artist.

2.1.2 Rendering

The previous section gave a brief overview of the physics of the interaction of light and matter, and the physical equation that captures the energy paths. In practice, the rendering equation is almost impossible to solve analytically, so approximations must be made. The broad categorization of these algorithms is global illumination. As the name implies, these algorithms "...take into account the distribution of light in the entire scene when deriving the color for any one surface point or image pixel" [24]. As an example, imagine a red box inside of a diffuse white box. As shown in Figure 2.3 the white walls appear reddish due to color bleeding. This effect can only be physically captured using global illumination. Global illumination is

Figure 2.3: Radiosity solution displaying color bleed phenomena

typically divided into two broad areas: radiosity and ray-tracing. Radiosity emerged from the field of thermodynamics and deals with calculating the energy transfer between patches in the scene. It is a view-independent solution, and makes an assumption that all objects in the environment are diffuse. Classical ray-tracing traces rays from the camera to the light sources, assuming objects in the path are purely specular. Since the rays are traced from the camera, it

is considered a view-dependent solution. An infinite number of rays is required to account for all possible sources of photons. It is obvious that neither solution independently can account for all interactions. Hybrid algorithms have been created, exploiting the strengths of each of the algorithms. The primary focus of this research will be ray-tracing algorithms, employing different techniques (but not radiosity), to capture the various diffuse-specular interactions.

Different ray-tracing algorithms have been developed based on first principles physics. Regardless of the algorithm, every ray-tracer must address the following areas. First, the scene geometry must be represented in three-dimensional (3D) space. This three dimensional geometry is then augmented with material parameters, and includes things like spectral reflectance and surface roughness, and how light is scattered from the surface. Of course, it would be a trivial image without light sources, so methods to model these must be incorporated. Perhaps the most important component is what is sometimes referred to as the *integrator* [45], which is used to solve the integral in the *big equation* 2.3. Finally, the radiance values are recorded as an image through the use of a sensor model.

2.1.3 `pbrt`

`pbrt` , which stands for *Physically Based Rendering Techniques*, was the software used in rendering the synthetic images. It was written over a period of approximately ten years at Stanford University [45], to aid in teaching physically based rendering to students. As such, there is extensive documentation and access to the source code. The authors of `pbrt` have implemented many computer science techniques to make it relatively straightforward for users to add plug-ins, modifying its behavior. In addition, there are many different built-in routines the user can select. For example, there are several different sampling techniques, integrators (ray-tracing algorithms), and surface reflectance models. This makes it an ideal test bed to use for psychophysical experimentation as only specific components can be changed, while all others are

held constant.

As the name implies, `pbrt` attempts to render "a 2D image from a description of a 3D scene..." using the "...principles of physics to model the interaction of light and matter" [45]. This is an important statement as many rendering systems used to create imagery, especially those in the entertainment industry, are at best loosely based on physics, rather than first principles based. There are advantages and disadvantages to each type of system. The non-physics based systems allow more degrees of freedom for the artist to manipulate the image throughout the entire process. With `pbrt` however, the user can only manipulate the input, including descriptions of the geometry, surface properties, and color of objects. `pbrt` uses this information as input to physics-based equations. Therefore, even those user specified inputs are expected to be based in reality. The result is a high quality, realistic image demonstrating real physical phenomenology, as in Figure 2.4. The tradeoff however is typically large amounts of computation time, and fewer degrees of freedom for the user. This type of rendering system is ideal for the author for several reasons. The first is that the author is not an artist, but rather a scientist. Physical properties can be measured using analytical devices (*i.e.* spectrophotometers) and those measurements can then be directly input into `pbrt`. Also, the final solution (including artifacts) is more easily understood by looking to the equations and methods of implementation.

How `pbrt works`

The intent of this section is to give a high-level introduction to `pbrt`, highlighting its features and data pipeline. Readers should not expect to find the minutia of the implementation or underlying physical methods and equations, and are encouraged to read `pbrt`'s accompanying text, [45], which contains many examples, images, lines of code, as well as the references to the seminal work upon which this system, and global illumination algorithms in general, are based. Much of the following description is based on the overview from the first chapter of the `pbrt`

Figure 2.4: Image rendered using `pbrt` demonstrating subsurface scattering through the use of photon mapping. [45]

book.

At its core, `pbrt` is a ray-tracing algorithm. "[I]t is based on following the path of a ray of light through a scene as it interacts with and bounces off objects in an environment" [45]. The authors go on to list the minimum requirements that a ray-tracing system must be able to simulate. These include:

- Cameras

- Ray-object intersections
- Light distribution
- Visibility
- Surface scattering
- Recursive ray-tracing
- Ray propagation

Cameras The camera is essentially like any camera one might use in real life. There is an aperture and a film plane at a bare minimum defining a *pinhole* camera. These two things and their dimensions define a viewing volume, outside of which, objects are not visible to the camera. Stated another way, once the camera is defined, so too is the portion of the scene that is visible to the camera. It makes sense therefore to propagate rays 'backwards' into the scene from the camera. In the case of a pinhole, one can think of this direction as a vector from the eye (or pinhole) through each pixel into the scene. Of course it is possible to stochastically trace rays from the light sources into the scene, let them bounce around and eventually reach the camera. However, it would take an extremely large number of rays in order for enough of them to randomly reach the camera, let alone produce a low-noise image. In addition to a simple pinhole, `pbrt` is capable of simulating a camera with real optics, in other words a non-infinitesmal aperture, and thus such phenomena as motion blur and depth of field. `pbrt` also supports orthographic and non-orthographic cameras, and environment cameras useful for creating environment maps. For the purposes of this research these advanced features were not required, and therefore not used in order to save computational time.

Ray-Object Intersections Rays originating from the camera, propagate into the scene and may or may not hit objects. The ray-tracer must test for these intersections, the details of which are provided in [45]. One interesting thing to note is that in a scene with multiple objects, it can quickly become computationally expensive to test a given ray against every single object. Therefore, `pbrt` uses specially designed data structures to store the scene in a spatially meaningful manner, eliminating unnecessary calculations, and accelerating required ones.

Light Distribution At this point, the camera has been defined, and objects have intersected by rays originating from the camera. The next step is to determine the amount of light interacting with that object and eventually reaching the film plane. Therefore light sources must be defined in terms of both their geometry as well as their power. `pbrt` being physically-based implies these light sources exhibit properties such as cosine projection and the $\frac{1}{r^2}$ falloff of light with distance. Light sources can be analytically defined as point sources, spheres or disks. `pbrt` can also use any user-defined geometry as a source, while still obeying the laws of physics. `pbrt` uses many techniques to stochastically sample these sources, which are described in detail in the book. Figure 2.5 shows examples of different geometries as light sources.

Visibility The previous section ignores one important thing, shadows. "Fortunately, in a ray tracer it is trivial to determine if the light is visible from the point being shaded. We simply construct a new ray whose origin is at the surface point and whose direction points toward the light...called *shadow rays*" [45].

Surface Scattering Recapping, we currently know both the incident lighting and location. The next step is to scatter that light off of the intersected surface. The scattering behavior of a surface is defined by its material parameters, specifically the *Bidirectional Reflection Distribu-*

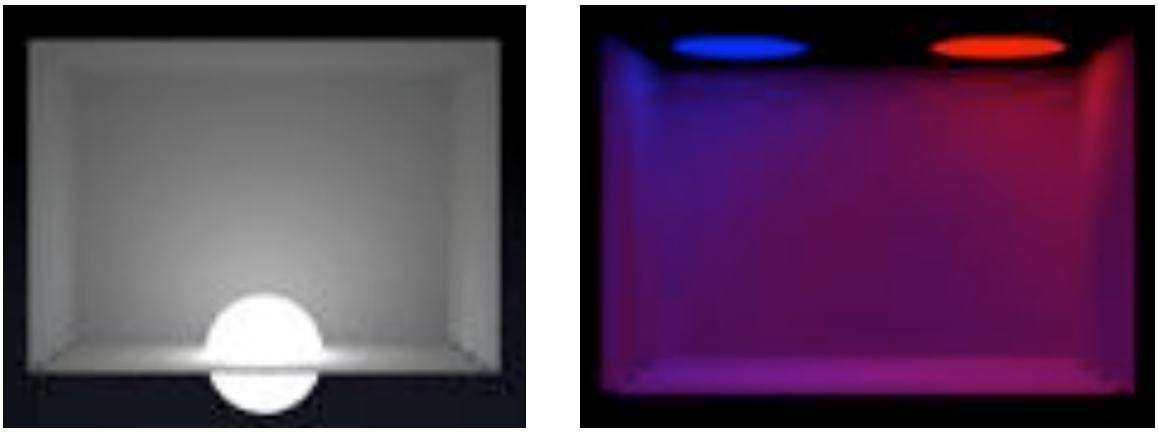


Figure 2.5: This figure shows light sources in `pbrt` using different geometries. The image on the left is a sphere illuminating a box. The image on the right shows two disk sources of different colors illuminating the inside of a box.

tion Function (BRDF), as described earlier in Section 2.1.1. This BRDF can be generalized for both reflective and transmissive surfaces to a function referred to as the *Bidirectional Scattering Distribution Function* (BSDF), again, an inherent property of the material. "`pbrt` supports a variety of both physically and phenomenologically base BSDF models" [45]. As is described later in Section 3.1, the scene designed for this research consisted of a majority of objects that were assumed to be *Lambertian*, the same reflectance value for any combination of input and output angles. Some, however, did use some of the more complex BSDFs.

Materials Materials in `pbrt` are modeled using combinations of surface reflectance functions. The three predefined `pbrt` materials used in this research were mirror, matte, and plastic. Mirror is self explanatory, and it uses a perfectly specular, or delta BRDF. Matte attempts to create a very dull looking material, essentially by eliminating any specular component. Depending on the user-defined parameters, a perfectly Lambertian or Oren-Nayar BRDF is used. An Oran-Nayar diffuse surface is described by a "collection of symmetric V-shaped grooves..." [45] This empirical approach captures the phenomena that real-world rough surfaces tend to

appear brighter as the illumination direction approaches the viewing direction, unlike a perfectly Lambertian surface. Finally, the plastic material is defined as a combination of glossy and diffuse scattering BRDFs. The user can control the color and the amount of diffuse and glossy reflection components with the surface roughness parameter. A Lambertian BRDF is used for the diffuse component while a Blinn microfacet distribution BRDF is used for the specular. The Blinn BSDF models the surface as a distribution of microfacets whose normals falls off exponentially perpendicular to the surface normal. Smooth surfaces fall off quicker than rough surfaces. Figure 2.6 shows examples of an object rendered using the matte and plastic materials.

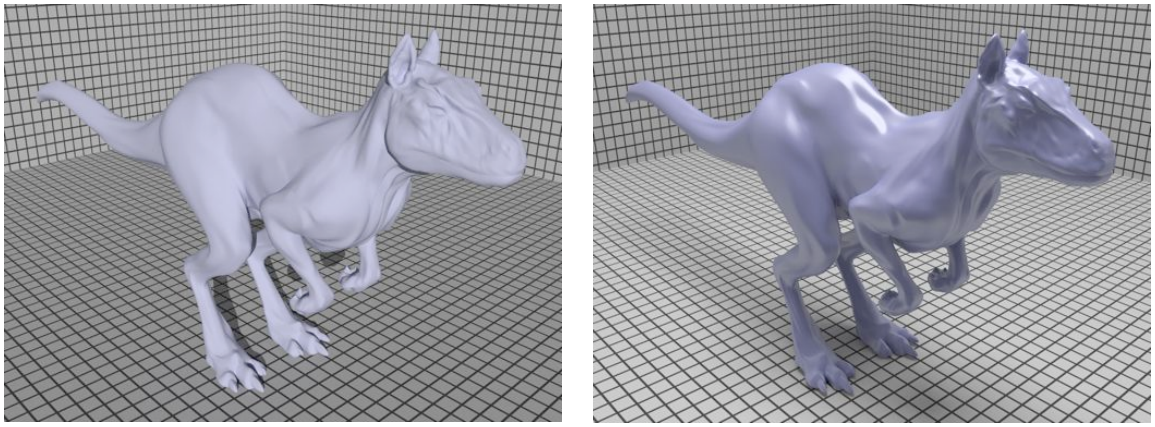


Figure 2.6: These images from [45] show the same object rendered with two different materials. The image on the left shows the killeroo with the matte material, and in this case a perfectly Labertian BRDF. The image on the right shows the killeroo rendered using the plastic material in `pbrt`. The plastic material adds a specular component to the object. Note that the illumination conditions are not identical in both of these images. The image on the left appears to be a spot or point source due to the hard shadows, and yet the matte material still exhibits no specular lobe.

Recursive Ray Tracing "In general the amount of light that reaches the eye from a point on an object is given by the sum of emitted light and reflected light" [45]. This idea was presented

earlier in Section 2.1.1, or as is referred to in the graphics community, the *light transport or rendering* equation, "...which says that the outgoing radiance $L_o(p, \omega_o)$ from a point p in direction ω_o is the emitted radiance at that point in that direction, $L_e(p, \omega_o)$, plus the incident radiance from all directions on the sphere S^2 around p scaled by the BSDF $f(p, \omega_o, \omega_i)$ and a cosine term:

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{S^2} f(p, \omega_o, \omega_i) L_i(p, \omega_i) |cos\theta_i \ | d\omega_i. \tag{2.4}$$

This equation is almost always too complicated to be solved by anything other than numerical integration techniques. The method used to solve this integral can be changed to one of several options included with `pbrt`. These *surface integrators* were the primary variable manipulated to create the stimuli for this research. The integrators are described briefly below.

Whitted As stated in [45], "Turner Whitted's original paper on ray tracing emphasized its recursive nature." Many of the original ray-traced images took advantage of this feature and were able to produce renderings of perfectly reflecting and refracting surfaces like glass and mirrored spheres. A ray is propagated from the camera to a perfectly reflecting mirror. In order to find what objects are being reflected onto that mirror, a ray is reflected about the mirror's surface normal, and the ray-tracing algorithm recursively called to add this contribution to the final solution. This results in the appearance of a reflection in the mirror. If we think about this further, it is apparent that Whitted does not solve the integral over the entire sphere, only in the specular direction. Stated another way, Whitted can only effectively render the direct and indirect illumination associated with objects that have a delta distribution BRDF. Figure 2.7 shows an image rendered using the Whitted integrator, capable of accurately simulating global illumination for the ideal point source and delta BRDFs.

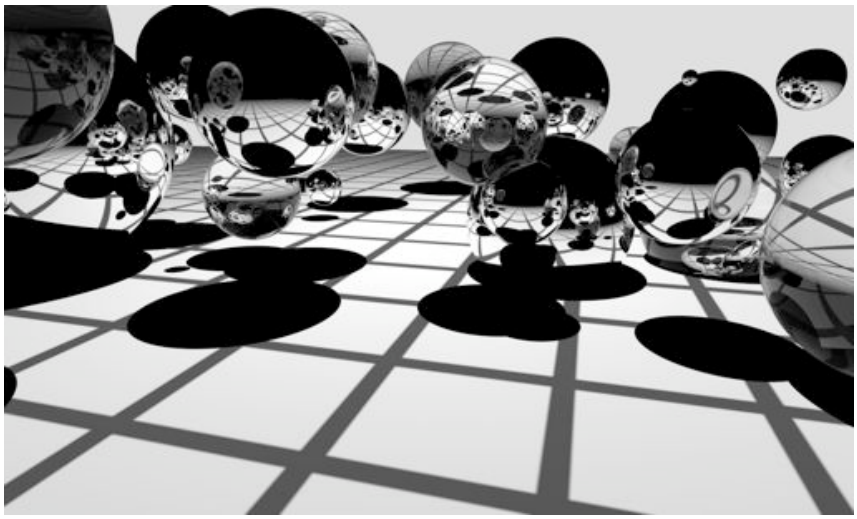


Figure 2.7: This image from [45] shows perfectly refractive and reflective spheres rendered using the Whitted surface integrator. Note also the hard shadows, indicative of point light sources.

Direct The direct lighting surface integrator only accounts for light arriving at a surface directly from a light source. In other words, it does not include indirect illumination from non-emissive objects. This reduced light transport equation can be written as:

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{S^2} f(p, \omega_o, \omega_i) L_d(p, \omega_i) |cos\theta_i \ | d\omega_i \tag{2.5}$$

where the only difference from 2.4 is that only the direct lighting, L_d is considered. It is different from the Whitted in that it solves the integral (using various sampling techniques) over the entire hemisphere. Therefore, the direct lighting solution of environments including objects with delta and non-delta BRDF's can now be solved. The solution presented in the book for solving the direct lighting can be used independently, or in conjunction with irradiance caching or photon mapping. In those cases, irradiance caching and photon mapping are used to solve the indirect illumination component which is added to the solution from the direct illumination surface integrator, yielding a complete solution to the light transport equation.

Path Tracing "Path tracing was the first general-purpose unbiased Monte Carlo light transport algorithm used in graphics" [45]. The simplest explanation of path tracing is that it is like Whitted, except that it supports both delta and non-delta BRDF's and light sources. It generates paths that begin at the camera and propagate into the scene, eventually ending at the source. At each vertex along a path there is a scattering event, based on the BSDF of the surface, from which a new path is propagated (see Figure 2.8). The radiance arriving at the

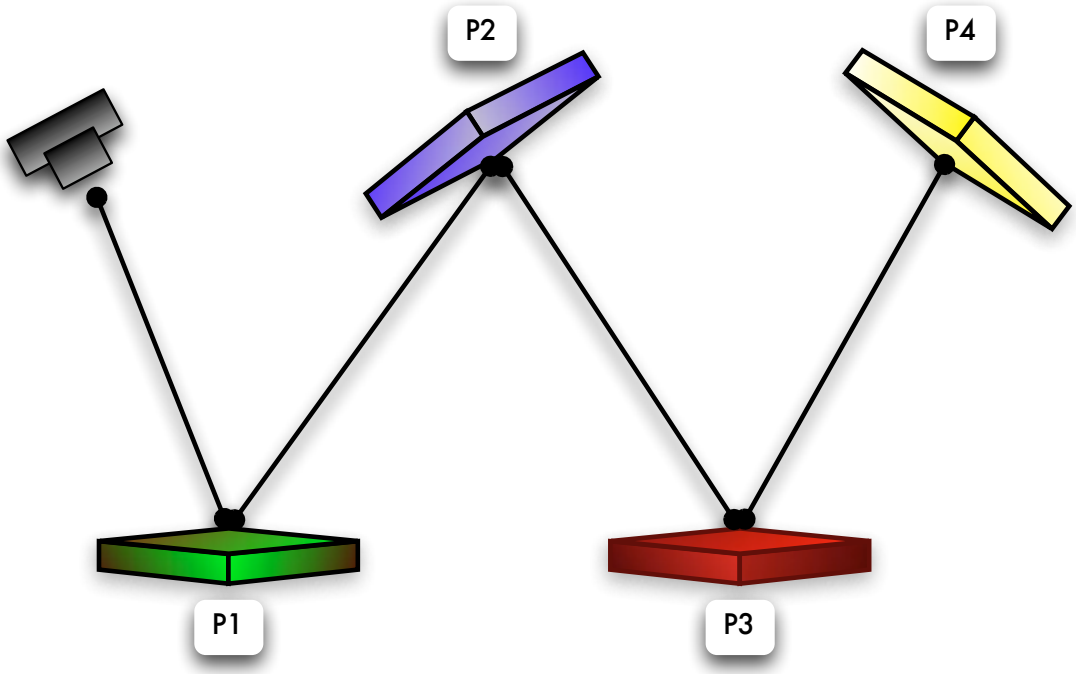


Figure 2.8: Iconic representation of path tracing.

camera is emitted radiance of the light modulated by the throughput of the path defined as the product of all of the vertex BRDFs. Different techniques are used to handle special cases such as delta BRDFs, as well as how to randomly terminate a given path so as not to trace

forever. Path tracing yields an unbiased solution to the sensor reaching radiance. While the mean value of an image may be correct, the variance may be high, resulting in an image with high frequency noise. The upside is that this can be predictably lowered by casting more rays into the scene, in other words, increasing the number of samples per pixel.

Irradiance Caching As stated above, path tracing is an unbiased algorithm, that decreases high frequency noise with an increased number of samples. This can be computationally prohibitive as it sometimes requires a large number of additional rays to reduce the objectionable noise to an acceptable level. Furthermore, if there is no noise present in the image, one can still be assured that the solution is not only visually pleasing, but also accurate. Figure 2.10 shows a direct comparison of the noise in irradiance cached and path tracing rendered images. On the other hand, there is a direct relationship between the number of rays, and the noise present in the final image.

This is not the case with biased algorithms. Biased algorithms also have artifacts, but are typically not the more objectionable high frequency noise as in path tracing. In addition, it is not true that increasing the number of samples will predictable reduce the noise present. On top of all of these problems, it is not even true to call an image with no visible noise an accurate representation to the actual scene radiance distribution. The reader should be asking themselves, '*Why would you ever use biased rendering algorithms?'*. These biased algorithms will produce a very realistic image, capturing the indirect illumination effects, while using a significant amount less computation time. "They can often create good-looking images using relatively little additional computation compared to basic techniques like Whitted ray tracing" [45]. Referring back to the definitions of realism presented in the Introduction, biased algorithms do not produce physically realistic images, but they may be photo-realistic, which is appropriate for this research.

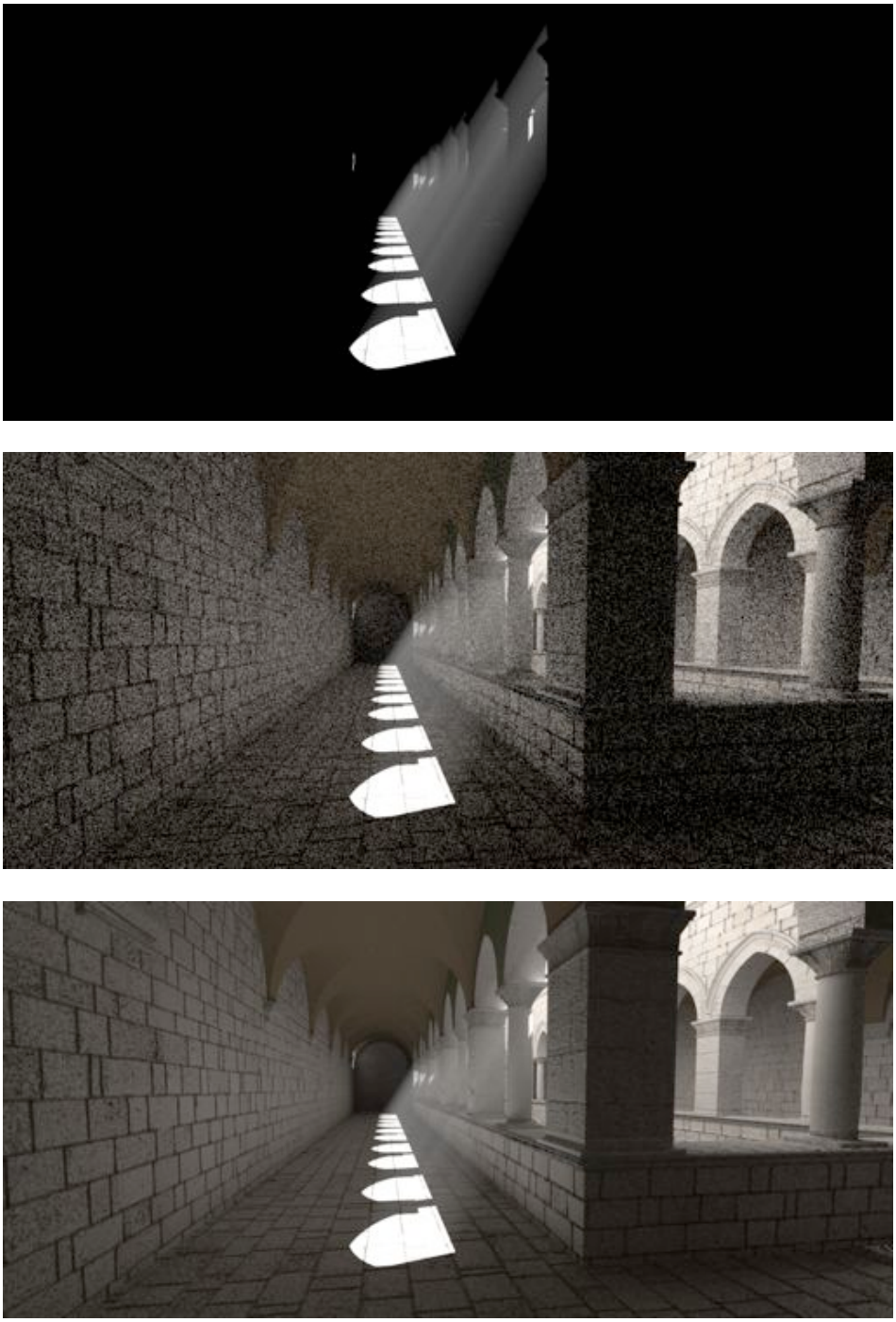


Figure 2.9: This image from [45] is dominated by indirect illumination. (a) was rendered using direct illumination only. (b) and (c) were rendered using path tracing, but with different numbers of samples per pixel. Image (b) rendered with fewer [spp] shows the noise of variance characteristic of an unbiased algorithm.

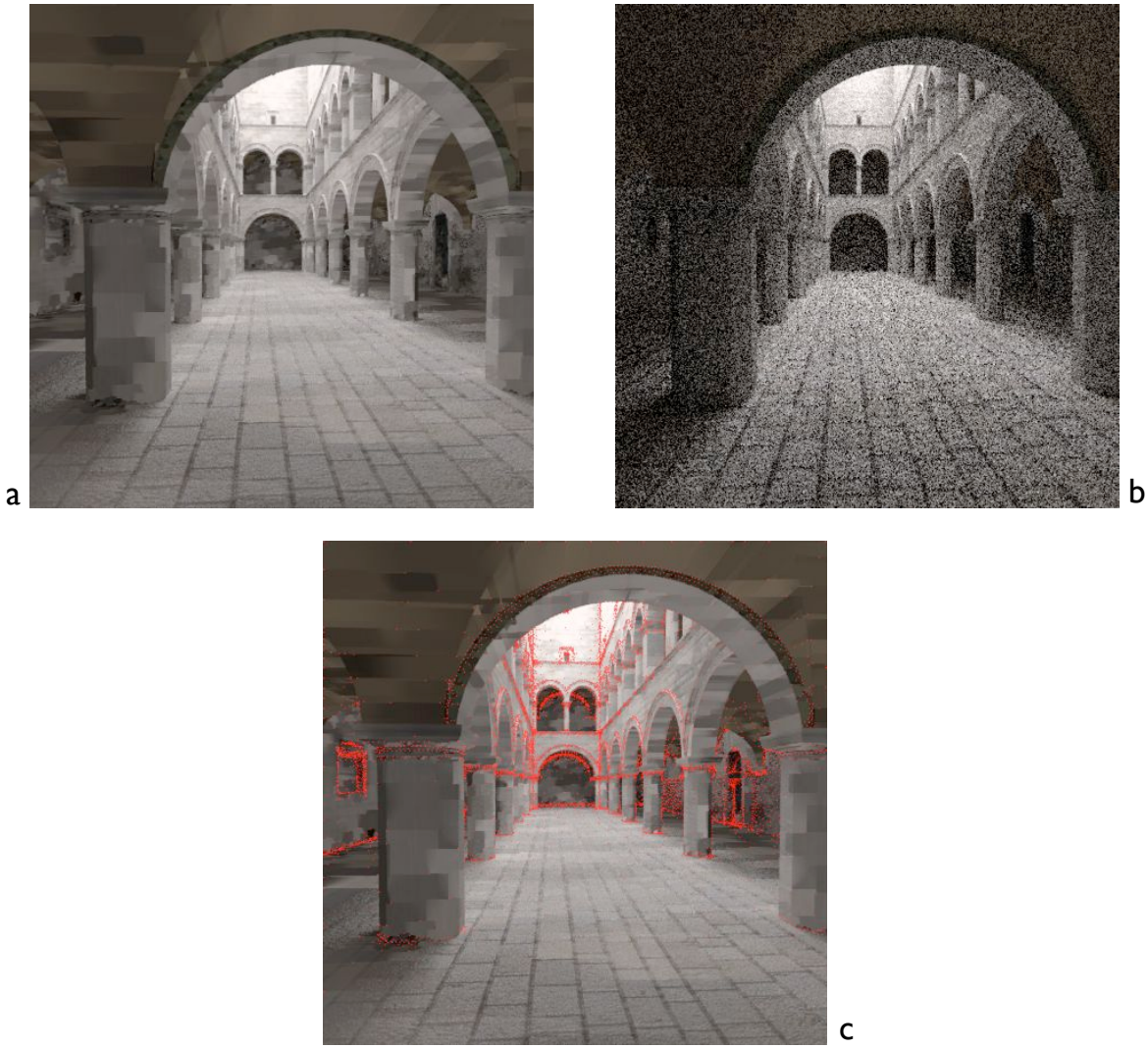


Figure 2.10: This image from [45] was rendered using irradiance caching (a) and path tracing (b) with approximately the same amount of computation time. Note how the noise manifests itself with each algorithm. (c) shows the locations of the precomputed irradiance cache samples.

Irradiance caching is one of these biased techniques. The idea behind irradiance caching is to pre-compute irradiance values at a select number of locations, typically locations where there would be the highest frequency change in the indirect lighting distributions. As the image is being rendered, these pre-computed irradiance samples are reused, interpolated, and averaged for intersected points that do not correspond directly with one of the stored samples. If the error is larger than a user-defined value, a new irradiance cache will be computed on-the-fly.

Irradiance rather than radiance samples are calculated and stored, which have smaller memory requirements. Recall that irradiance can be thought of as an average of the radiance over the entire hemisphere. In other words, irradiance caching assumes the surfaces are Lambertian. This is actually a good assumption for the most part for this research based on the scene construction of nearly Lambertian objects, described later, but not in general.

Photon Mapping Photon mapping is the final biased, surface integrator that was used in this research. Photon mapping is a two-pass technique. The first pass is to propagate no more than a user-specified number of rays into the scene from the light sources. These rays are propagated based on the surface reflectance properties and stored into a 3D data structure called a kd tree. The scene is then ray-traced as usual starting from the camera and propagating into the scene. The ray stops at the first object it hits, and the photon map is accessed. Based on some user-defined parameters, the photon map is searched for nearby photons, and the solution for the given point solved by averaging these photon within a given search radius. The photon map can be used to solve the indirect illumination component only (in conjunction with the direct surface integrator described above), or it can be used to solve both the direct and indirect as shown in Figure 2.11. Since photon mapping propagates rays from the source to the scene, it allows for computation of complex illumination phenomena such as caustics in water. One of the major disadvantages is the number of parameters that can be adjusted in order to *tune* the photon map. This can also be a distinct advantage over other algorithms.

Figure 2.11: This image is from [45]. (a) was rendered using the photon map for both indirect and direct illumination, while (b) used the photon map for the indirect only. Note the blotchy artifacts in (a) due to the photon map.

One of the most important settings in photon mapping is the use of what is called *final gathering*. Final gathering is a technique to reduce the visibility of artifacts. As stated earlier, the photons within a given radius are used to compute the exitant radiance at a point. Final gathering, however, samples the Bidirectional Scattering Distribution Function (BSDF) as that point, and traces rays back into the scene, and finds incident radiance along those rays. The error of interpolation is now at these rays, rather than the exitant radiance point, thus reducing the artifacts in the final image. Figure 2.12 shows the effect final gathering has on final image

Figure 2.12: This image is from [45]. (a) was rendered using photon mapping, and (b) was rendered using photon mapping with final gathering. Notice how the artifacts are greatly reduced with final gathering.

quality.

Ray Propagation The final requirement of a ray-tracer is the ability to propagate rays. `pbrt` has the ability to do this in a vacuum or with participating medium present such as smoke or fog.

A Note on Color in `pbrt`

In general, `pbrt` is designed to handle spectral representations for the materials and light sources, however, in its default configuration, it essentially reduces to a RGB triplet configuration, based on the primaries in the sRGB specification [3]. Therefore, if no changes are made, the user must specify the color of lights and objects using a triplet. The most straightforward way to calculate these triplets is to measure spectral reflectances of the objects and lights, calulate XYZ's from these spectra, and then use the sRGB 3x3 to transform these to the expected primaries of `pbrt`. The other option is to modify `pbrt` to read and write spectral files.

Conclusion `pbrt`

This section provided a high level description of `pbrt` a physically-based ray-tracer. `pbrt` is a sophisticated piece of software with many rendering options. It is very capable of producing an extremely realistic image. The quality of the final image however relies heavily on the input material parameters, and surface integrator settings.

2.2 Augmented Reality

To this point the discussion has focused on creating a purely synthetic image. There is, however, a growing body of research in the area of *augmented reality* [4]. Augmented reality attempts to fuse real and synthetic images together. The application areas relate closely to the definitions of realism in Section 1.2 and can include military training or guided surgery, where functional realism is important. Figure 2.13 shows examples of these application areas. It is clear that if a surgeon is operating on a patient's liver, a photorealistic rendition is not as important as a real-time image depicting the accurate location of the organ highlighted in an unmistakable color. Other applications of augmented reality include compositing live actors onto virtual sets

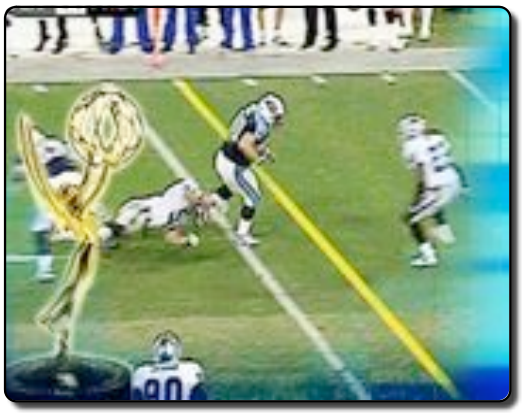
via a green-screen method. In this case, photo-realism is key, to provide the viewer with a sense that the actors were actually in that environment when the image was taken. This research is more concerned with the latter example, where the goal is photo-realism, with no emphasis on the ability to compute the solution in real-time, or to derive all of the necessary parameters from the imagery. This section highlights the general procedure, and specifically Debevec's technique [11], *Rendering Synthetic Objects into Real Scenes.* [1]

The problem addressed by his paper is to add synthetic objects to the image of a real scene. This is not a simple compositing problem. The synthetic objects must occlude and shadow the real scene, and vice versa. The novelty of this technique is to use a global illumination algorithm in conjunction with image-derived lighting and material parameters.

2.2.1 Modeling

Debevec proposes a solution based on global illumination utilizing image-based lighting, within the framework of a novel scene representation shown in Figure 2.14. The scene representation indicates the light interaction, required material and geometric parameters of each. The distant scene is represented by a *light-based model.* This is a term he uses "to refer to a representation of a scene that consists of radiance information [12], possibly with specific reference to light leaving surfaces, but not necessarily containing material property (BRDF) information" [11]. It is important not to confuse this with a material-based model, the pre-cursor to a light-based model, or an image-based model in which the values may not represent absolute radiance. The reason can be seen in the diagram (Figure 2.14), where the distant scene is used to illuminate the synthetic objects. Also noted is that light reflected back towards the distant scene is ignored. The local scene is the area of the real scene that interacts with the synthetic objects, both in terms of illumination and geometric considerations. Therefore, the geometry and material

[1] It is noticed that Debevec refrains to referring to the problem as augmented reality, as it has many connotations with virtual reality, not relevant here.



Sportvision's 1st and Ten

Augmented Reality Guided Surgery

Augmented Reality in Star Wars

Figure 2.13: Examples of Augmented Reality Applications

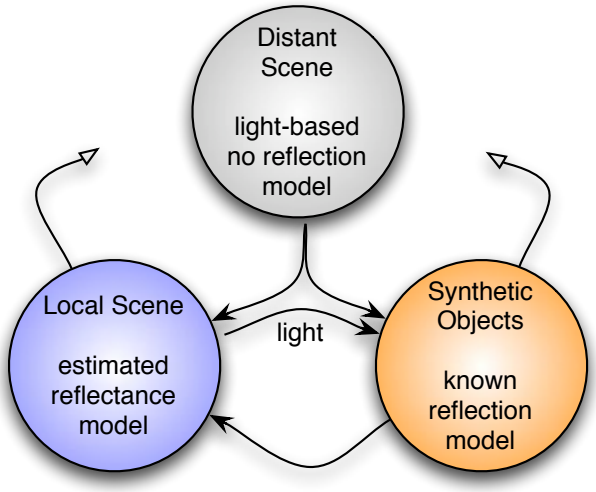


Figure 2.14: Interaction of the Scene Components in Debevec's Technique

properties must be known at some level. Debevec explains that the geometry can be determined both actively and passively, and there is a lot of literature on both. Additionally, there are methods for estimating the BRDF if accurate measurements are not available [11, 49, 50, 57, 23, 35, 13, 22]. The idea being that enough information should be gathered from the scene *in situ*, without separate instrumentation to measure these parameters. The local scene must be modeled accurately enough so that the global illumination algorithm yields an acceptable solution. The synthetic objects of course must be modeled accurately, and be represented by any material supported by the global illumination solution.

2.2.2 Rendering and Compositing

At this point, the scene has been divided into three components, and modeled at different levels. The next obvious step is to input this information into a rendering engine, and create images. The distant scene is mapped to an approximate model of the room (the inside of a cube for

example), and used as the illumination source. A global illumination solution like DIRSIG or `pbrt` , is run which creates the output image as well as a binary mask, where white corresponds to the local scene and synthetic objects, and black the distant scene. The resultant image is composited [7] with the background photograph using the mask to choose pixels from the background or rendered image. Debevec points out that occlusion by the distant scene can be included if there was some model of its geometry.

2.2.3 Differential Rendering

Debevec further refines the procedure using a technique called differential rendering. This is useful because using the rendering procedure above requires the geometry and material properties be measured to a high degree of accuracy. The reason being the global illumination solution is used in place of all local scene components in addition to the added synthetic components. It is often difficult to capture spatially varying textures, and an accurate, full BRDF. The background photograph in effect contains all of this information. The idea is therefore to only render in the difference between the local scene, and the local scene with the synthetic objects added. Using Debevec's notation, LS_b refers to the background image, LS_{noobj} is the rendered local scene without the synthetic objects, and LS_{obj} is the rendered local scene with the synthetic objects. The error of the rendered local scene, Err_{ls} is defined as:

$$Err_{ls} = LS_{obj} - LS_b \tag{2.6}$$

The final rendered local scene is then:

$$LS_{final} = LS_{obj} - Err_{ls} \tag{2.7}$$

$$LS_{final} = LS_b + (LS_{obj} - LS_{noobj}) \tag{2.8}$$

It can be seen from Equation 2.8 that nothing is added to the background when there is no difference in the rendered local scenes. If the term in parentheses is negative, light is subtracted from the background, implying a shadow. Conversely, light can also be added back into the images if that term is positive. Problems can occur if the error term is negative, in which case Debevec adjusts for the *relative error*:

$$LS_{final} = LS_b \left(\frac{LS_{obj}}{LS_{noobj}} \right) \tag{2.9}$$

2.3 The Human Visual System and Computer Graphics

As discussed in Section 1.2, there is difference between physical realism and photo realism, in that a perceptual match does not guarantee accurate radiance values. However, as the eye is typically the final discriminator, this may not cause a problem. In order to understand this discrepancy and which tradeoffs can be made, the human visual system (HVS) must be investigated.

2.3.1 The HVS: Relevant Properties for Photo-Realism

This section describes the Human Visual System and the properties that are leveraged to the advantage of image synthesis. Typically, a vision (computational) model is built from these properties. The vision models range from rather simplistic to very involved. In any case, these

models are based upon the current body of knowledge on the HVS. The HVS is not completely understood, and therefore a perfect model does not exist. The specific properties that will be discussed are adaptation, non-linearity of response, contrast sensitivity and masking are properties of the HVS that are used in these models.

General adaptation refers to the changes in the HVS in response to the overall luminance of the scene. This effect can be summarized by stating humans are not absolute light meters as they adapt to the current light level. This makes it possible to give the illusion of very bright or dark scenes on rather limited monitors. Depending on the scene, two different luminances may be mapped to the same or to vastly different pixel values [21]. In the natural world, humans are exposed to large dynamic ranges of luminance. The absolute dynamic range for humans is on the order of 10 million to 1 (comparing sunlight to startlight). At any give time the dynamic range can be on the order of 10,000 to 1. The human visual system is able to function over this broad range by the process of adaptation. Adaptation involves changes in the pupil diameter, rods and cones, photopigments and neural processing [18]. These systems allow the HVS to operate over 14 log units of luminance. It is important to understand that vision as a whole is not constant across that range. Visual acuity and color vision are better at increased luminance levels, but "absolute sensitivity is low and luminance differences have to be large to be detectable" [44]. Conversely, for scotopic conditions, acuity and color vision are decreased, and sensitivity to luminance differences is increased. This behavior is more formally known as Weber's Law, and was first described by Weber in the early nineteenth century [32]. This law states that the ratio $\Delta I/\mathrm{I}$ is equal to a constant K, where I is a given stimulus intensity level, and ΔI is the minimum change in intensity resulting in a perceptible difference. Intuitively this behavior makes sense, and is evident in everyday situations. Consider the example of a person talking in a quiet room who is easily heard, versus that same person in a loud room. In order to be heard, the person must speak even louder. Threshold experiments have been used to measure the adaptation effects, and in these conditions are well characterized and obey

Weber's law over wide luminance ranges. Additionally, humans are not relative light meters either. For example doubling luminance does not double the perception of brightness. This effect has to do with the non-linearity of response in the HVS.

Acuity is often characterized in terms of the Contrast Sensitivity Function (CSF)[2]. The CSF measures the sensitivity of the HVS to sinusoidal gratings at different frequencies. The CSF is measured through psychophysical experiments and standard results can be found in the literature [41]. There are several important, general results [44]. First, the spatial frequency response of the achromatic channel is like a bandpass filter while the chromatic channels act like low pass filters. Second, the high frequency cutoff for the achromatic channel is 60 cycles per degree [cpd] and slightly less for the chromatic channels. It is important to note that the CSF is a function of mean field luminance, retinal eccentricity, time, color, adaptation, distance, size and more. These parameters change the the shape of the CSF and the high frequency cutoff. For example, at higher luminance levels the CSF has a bandpass shape, and higher cutoff frequency than at lower luminance levels. The reader must also understand that the CSF is not a modulation transfer function (MTF), for many reasons. An obvious reason is that the HVS does not behave as a linear system. The CSF describes only the sensitivity of the HVS to sinusoidal gratings at different spatial frequencies.

Masking is the phenomenon where the visibility of a signal at a particular spatial frequency may be more or less visible due to the presence of another signal at a nearby frequency. Masking is due to spatial processing in the HVS. When a mask and a test have very close spatial frequencies, the test is difficult to perceive. The presence of the test is more visible when the spatial frequencies are different. There is also a contrast effect, and in general a mask facilitates the test detection at low contrasts and masks detection at high contrasts [54]. Computational models of visual masking have been created for computer graphics [19].

[2]Various researchers have performed experiments to understand the reasons the HVS has developed to be optimized for certain frequencies [42].

The descriptions of adaptation and spatial vision above are based on threshold experiments. Therefore they describe the performance of the HVS in its limits. However, real-world scenes provide conditions well above threshold, or *suprathreshold*. "Stevens' model of brightness and apparent contrast...summarizes much of what is known about the intensity dependence of surface appearance at suprathreshold levels" [44]. The relationship can be described by Stevens' Power Law (with a power less than 1), where brightness increases as a power of the luminance. In general, "as we turn up the lights, the world becomes more vivid" [44]. This suprathreshold model has been applied to both adaptation and spatial vision. It turns out that the threshold experiments are special cases of the general processes. Threshold versus intensity experiments, (TVI), demonstrate these effects under more natural conditions [44].

In depth research has been performed on each of those areas. Therefore this section was meant to serve as a high-level overview of some visual phenomena. Operational characteristics of the HVS described above at threshold and suprathreshold levels are important to creating complete visual models. The functionality of the model is application driven. Therefore, rendering engines will utilize different aspects of tone mapping operators due to different constraints and their placement along the pipeline for creating synthetic imagery.

2.3.2 HVS and Image Synthesis

Rendering the scene is computationally expensive. If the final discriminator is a human observer, then the HVS can be used during the rendering to reduce the rendering time by foregoing calculations that would result in information below the HVS thresholds. In addition to reducing calculations, using vision models should result in more photo-realistic images as the output has been 'tuned' for human observers.

There are two major categories of rendering algorithms: local and global illumination. Local illumination is computationally easier, but renders each object independent of the rest of the

scene. Global illumination techniques take object inter-reflections into account, or in other words calculate the transport of light on a global scale. In addition to inter-reflections, this allows for the calculation of shadows and other effects not possible with local illumination (volumetric lighting, refraction...). To reiterate, the goal is to achieve *photo-realistic* images. Therefore the focus here is on the global illumination techniques[3]. Ray-tracing, radiosity, and photon mapping are three global illumination rendering algorithms. The fundamental details of these techniques can be found in graphics texts [56, 24].

Common to all is the idea of sampling. This is due to the fact that computers can only store samples of continuous signals and not the analog signal itself. Sampling results in aliasing, where high frequencies are aliased to lower frequencies. Discussions of aliasing can be found in [25]. Aliasing can be mitigated by taking more samples. This does not help in computational costs however. Rather than uniformly taking more samples, people have been smart about where the samples should be placed. "Mitchell realized that deciding where to do extra sampling can be guided by knowledge of how the eye perceives *noise as a function of contrast and colour* [36]. The previous discussion of the CSF explained the HVS's sensitivity is a maximum around 4.5[cpd] and the cutoff (below the sensitivity of the HVS) is approximately 60[cpd]. Therefore the HVS is most sensitive to aliasing artifacts around 4.5[cpd] frequencies. Essentially, Mitchell used non-uniform sampling techniques based on the frequency content of the images. He used a contrast metric to decide where the extra samples should be placed. This technique does not eliminate aliasing, rather it causes aliasing at higher frequencies where humans are less sensitive. "Although this idea has the beginnings of a perceptual approach, it is at most a crude approximation to the HVS" [36].

Meyer and Liu again take advantage of the HVS's spatial vision, realizing that acuity is different for the chromatic and achromatic channels, specifically, the chromatic spatial acuity is worse.

[3]Tricks and hacks can be used in conjunction with local illumination to give the *effect* of global illumination by approximating shadows or object interactions.

The image is processed and stored in a kd-tree data structure. A kd-tree is a data structure that is often used for photon-mapping as it lends itself to non-uniform sampling [29]. In this case the kd-tree stores higher frequency information towards the bottom. When they were performing calculations they would obtain the image information from the kd-tree. For spatial color calculations, a complete traversal of the kd-tree was unnecessary based on results of psychophysical experiments. Their results showed some computation savings while maintaining image fidelity [36].[4]

The next in the progression of vision model complexity is a frequency-based ray tracer [6]. Bolin and Meyer's vision model takes into account contrast sensitivity, spatial frequency response, and masking. They control where the rays are cast into the scene based on these HVS properties. "A specific luminance difference at low intensity is considered to be more important than the same difference at high intensity" [6]. They use a Monte Carlo ray tracer to calculate the global illumination solution. Using their algorithm, more rays are spawned when low frequency terms are being determined than when high frequency terms are being found. Overall, they were able to remove the visible artifacts first, and any noise due to ray tracing is channeled into areas where it is less noticeable.

There have been several other rendering solutions that have incorporated vision models that primarily leverage spatial vision characteristics of the HVS. Details of their implementation have been omitted for brevity. The important thing to realize is how aspects of the HVS have been embedded into vision models which in turn are used to efficiently and photo-realistically render images. More complex vision models have also been tested, but at a certain point the complexity reduction in the global illumination solution is outweighed by the added complexity in the vision model. Therefore the overall computation is increased at no gain in visual fidelity [14, 36].

[4]It is important to note that they used psychophysics for rating the output of their algorithm. Often in the graphics community, the 'looks good' approach is used.

2.3.3 Tone Mapping Operators

As stated before, the output from the rendering solution contains values that cannot be reproduced on normal displays. The high-dynamic range of the input tones must be mapped into the range of values allowed by the display which is inherently a low-dynamic range device. There are many different techniques that are broadly categorized as global or local tone mapping operators. The global operators apply the same operator to every pixel in the image based on the entire scene content. Local operators on the other hand apply different operators to each pixel based on the spatially local scene content. There are advantages and disadvantages to each approach. This is a very popular area of research currently, and readers are encouraged to consult [47] for details on the subject of high dynamic range imaging, including examples and explanations of tone mapping operators.

2.3.4 Image Evaluation

At this point in the pipeline a synthetic scene has been rendered and tone-mapped to a display. This is arguably the most critical point in the pipeline; evaluating the image. It is at this stage where it is ascertained whether or not the correct tradeoffs were made. Ideally the output image will provide the same visual response as the original scene per the requirements outlined in Section 1.2. The obvious question is *How are these images evaluated?*. There are two answers. The images can be presented to human observers as part of a psychophysical experiment, or they can be computationally evaluated using a defined image quality metric.

Experiments using Psychophysics

The most obvious experiment is to compare a photograph of the real world scene to the synthetic image. Meyer et al. [39] used an image synthesis approach that is based on two specific modules:

physical and perceptual. Their approach was to achieve accurate light simulation before the image was degraded with a perceptual transform.

Their experimental scene consisted of diffusely reflecting objects placed in a small dark room. A synthetic scene was created using a radiosity solution. Radiometric values of the scene and image were measured using a spectral integrating radiometer and compared. In their paper they outline constraints they used for the radiosity solution to achive results that were in agreement with the real scene measurements.

After verifying the output of the model, the next step was to compare the physical model with an image on a television screen. The synthetic image was then converted to RGB values to create a color television image using color science techniques. The observers viewed the monitor and the real scene through a view camera. They justify this methodology stating it allows simultaneous side-by-side comparisons without introducing the effect of the observer's memory. Twenty observers (10 experts and 10 naive) performed the image comparison. Nine out of the twenty people (45%) selected the real scene when asked to pick the computer generated scene. They concluded that the observers did not perform better than if they were just guessing. The overall scene and color were judged good by the observers, however, there were weaknesses cited in the sharpness of shadows and in the brightness of the ceiling panel.

The results of this study (from 1986) show promise in achieving photo-realism. However, there are certain weaknesses in their approach. First, the scene was very simple consisting of simple shapes that were purely diffuse. Also, the methodology for comparison was not inherently controlled, and the view cameras reduced sharpness. McNamara [36] suggests a more robust approach is required.

They developed a technique for measuring the perceptual equivalence of a graphical scene to a real scene. They began by running several psychophysical experiments where human observers were asked to compare two-dimensional target regions of a real physical scene to regions of the

synthetic representation of that scene. The results of these experiments showed that the visual response to the real scene was similar to the high-fidelity rendered image [38].

This was extended to comparisons of complete three-dimensional objects, which inherently allowed comparisons of scene characteristics such as shadow, object occlusion and depth perception. Their scene consisted of seventeen test objects. In their paper [37], they describe the training procedure on Munsell chips prior to the experiment. The observers were presented with 10 images of the scene of which one was the photograph, and nine were created using different rendering solutions. They were asked to match the lightness of the 17 objects and 5 sides of the environment, to test patches, resulting in a total of twenty two matches. Their results indicate a difference between the rendering techniques. Three of the methods are of the same perceptual quality as the photograph of the scene in terms of lightness matches. More important is that they provide a framework for measuring the perceptual equivalence of a real scene and a rendered equivalent.

Realism Cues

Rademacher, et al. [46] approached the photo-realistic question from a different perspective. He proposed that the key to creating good rendering algorithms is to first understand the perceptual process. They proceeded to measure the visual realism in images using psychophysical experiments. The experiments focused on shadow softness, surface smoothness, number of light sources, number of objects, and variety of objects. It is clear that they wanted to hone in on which cues in an image contribute to the realism, not just that the final output appears realistic. The question posed to the observers was to rate a series of images as "real" or "not real." The problem is what definition do the observers use for real? The researchers, in an effort not to bias the observers, gave them minimal instructions, only using those two phrases in the context of computer-generated imagery. The images were of very simple scenes

and objects. The methodology explained in the paper is indicative of the method of constant stimuli. Rather than assign thresholds though, they created a scale of realism. The realism scale was defined as the proportion a feature was judged real, where 0 is not real, and 1 is real. For example, if a certain shadow was presented to the observers 20 times and judged as real 10 times, then the realism scale would be 0.5. The results are summarized as follows. First, sharper shadows were perceived as being less real. Second, diffuse surfaces were rated as being more real than spray-painted (more specular) surfaces. The observers realism response did not increase with an increase in the number of objects, variety of object shapes or number of lights. The interesting thing to note is that these first experiments all used images of real scenes, which did indeed look like computer-generated images. Different lighting was used to change the shadow hardness, and different material types and object types were also varied with real parameters. The same experiments were run using computer-generated imagery. The results were qualitatively in agreement with the experiments using the photographs. A χ^2 test confirmed a statistical significance between the computer generated imagery and the shadow softness and surface-smoothness experiments.[5]

Compositing Errors

Selan [52] performed psychophysical experiments most closely related to this research. He isolated four sources of lighting error in compositing: brightness errors, chromaticity errors, shading directionality errors, and case shadow directionality errors. In the experimenters stimuli, both components composited together were real, as opposed to rendering a synthetic object into a real scene. Also, a local, rather than global, illumination method was used to shade and render shadows on the composited elements. A set of stimuli was created varying one of these

[5]Interesting experiments have also been performed to distinguish paintings from photographs based on the photorealism of the images. The methods in [9] were also verified through psychophysics. There is also a lot of literature on other visual cues that affect three dimensional visualization and renderings. [28, 48, 55]

parameters along its continuum. Four separate two-alternative forced-choice experiments were conducted where the subject was asked to determine which element in the image was composited. Their results showed observers are very sensitive to chromaticity and brightness errors. By and large, observers were insensitive to illumination direction errors. This is a significant result because it indicates that compositors need to worry less about lighting effect. Finally, this experiment found that there was a dependence on whether the shadows were converging or diverging. Diverging shadow errors were less likely to be detected than converging shadow errors. This was primarily due to the fact that diverging shadows are more likely to exist in natural settings.

2.4 iCAM

To this point, several major topics have been reviewed: generating physically based synthetic images, augmenting real imagery with synthetic components, and evaluating the realism of these images through the use of psychophysics. The final topic that will be discussed is a specific model of human vision capable of predicting image differences, appearance, and overall image quality. This model is termed iCAM (Image Color Appearance Model), and was described as a part of Johnson's Ph.D. dissertation [30]. The motivation of his research was to devise a modular framework, that would mimic various properties of the HVS, thus creating a device-independent image quality model, with the ultimate goal "to predict perception" [30].

The first important aspect of this research was to make the model a collection of modules. This allows the use of current color difference research, while maintaining the flexibility to add new modules as more research is done. Johnson also presented the idea of a pool of modules, where each module accepts input, and provides output, while acting as a self-contained unit. The strength of this design is that it allows the modules to be linked together to provide an overall metric, and at the same time the individual modules can be used to determine the cause for

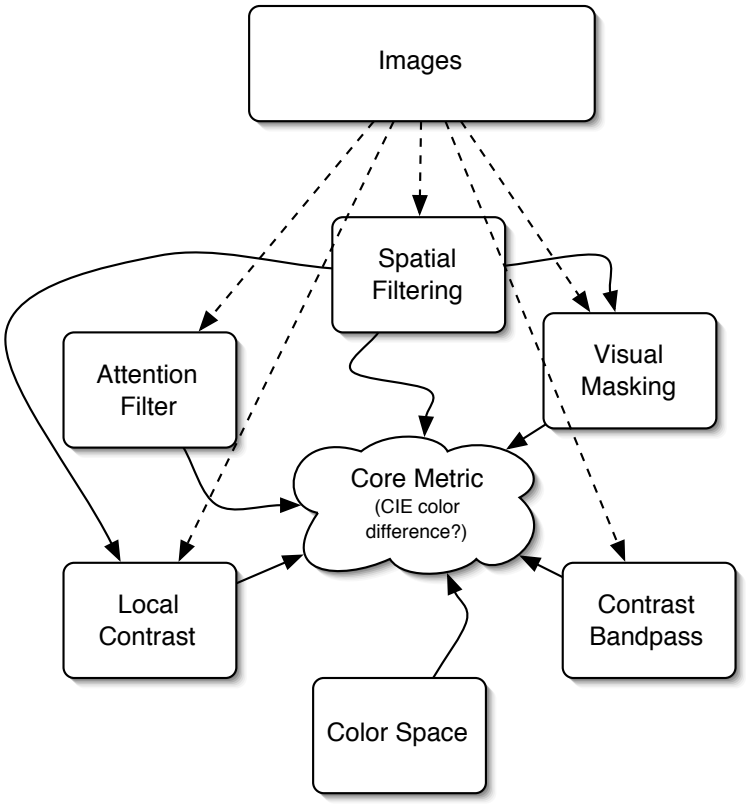


Figure 2.15: Modular Pool Concept [30]

the image difference. This idea is shown in Figure 2.15.

2.4.1 Modular Image Difference Framework

The image difference functionality of iCAM is based on decades of CIE color difference research. As Johnson points out though, traditional color difference equations were developed based on uniform color patches, not complex spatially varying stimuli like images. Therefore, applying these equations to images would essentially be treating the individual pixels as separate stimuli, not accounting at all for the coherence in the imagery. It is well known that the HVS keys on

certain features in imagery, like edges, in addition to the color difference. The idea then is to preprocess the images spatially in a similar manner as the HVS, and then apply the color difference equations. The first incarnation of this idea was called S-CIELAB, where S stands for *spatial*, accounting only for the CSF. Johnson extends the S-CIELAB concept to also account for spatial-frequency adaptation, spatial localization, and local and global contrast detection by introducing modules for each. Again, each of these modules provides its own results on the image differences. However, it is also beneficial in some cases to reduce all of these results into a single number, *the image difference*. This number is some weighted sum of the individual modules, giving a measure of how different a pair of images is in terms of image difference units. In reducing to a single number, information is inevitably lost, yet the result could be a scale of image differences along some variable continuum, which in turn can be compared to a psychophysical scale. Of course, in some situations this could replace, or at least pre-process the stimuli for a psychophysical experiment, reducing the burden on human observers. Johnson also points out that "...an image difference model is only capable of predicting magnitudes of errors, and not direction." This can be obviated by examining the output of each of the modules individually. Rather than using the traditional color difference equations which essentially measure a scalar distance, calculations of entities like ΔL^* can be performed. Figure 2.16 indicates specific causes for image differences, and the corresponding module that contains this information.

2.4.2 Image Appearance Modeling

In the same way color difference equations were extended to images, color appearance modeling research is extended to spatially complex stimuli, creating image appearance modeling. Color appearance attempts to quantify and predict attributes such as lightness, brightness, colorfulness, chroma and hue. Image appearance extends this to include things such as sharpness,

graininess, contrast and resolution [30]. The most logical step is to replace the core metric at the heart of the image difference framework with an appearance space. Johnson gives his argument for using the IPT color space. The overall framework for the image appearance model, iCAM , is given in Figure 2.17. The input is a colorimetric image as well as the surround adapting stimulus. The first stage is to account for the chromatic adaptation. Following this, the image is converted from cone-similar RGB space, to opponent-color signals (IPT), representative of color encoding within the brain. These are then non-linearly compressed via an exponential term, in order to "predict response compression that is prevalent in most human sensory systems" [17]. Johnson demonstrates how this model is capable of predicting such phenomena as simultaneous contrast, crispening, and spreading. He has also shown success in rendering high-dynamic range images to low-dynamic range displays using iCAM, and producing results consistent with psychophysical experiments. In addition, image difference calculations are also possible using this new framework, as it relies heavily on the modular image difference framework. The input is two images rather than one. The difference modules are applied after chromatic adaptation.

2.4.3 iCAM Conclusion

iCAM has been shown through [30] to be a robust tool for measuring and predicting overall image differences. Even more important is the modular nature of iCAM . The ability to examine the output from each module pinpoints specific causes for differences between images. The module output can also be weighted to create a scale of difference. In general, iCAM provides a new tool that extends traditional, robust color difference equations, with models of specific properties of spatial / spectral human vision, that can be used to examine complex stimuli.

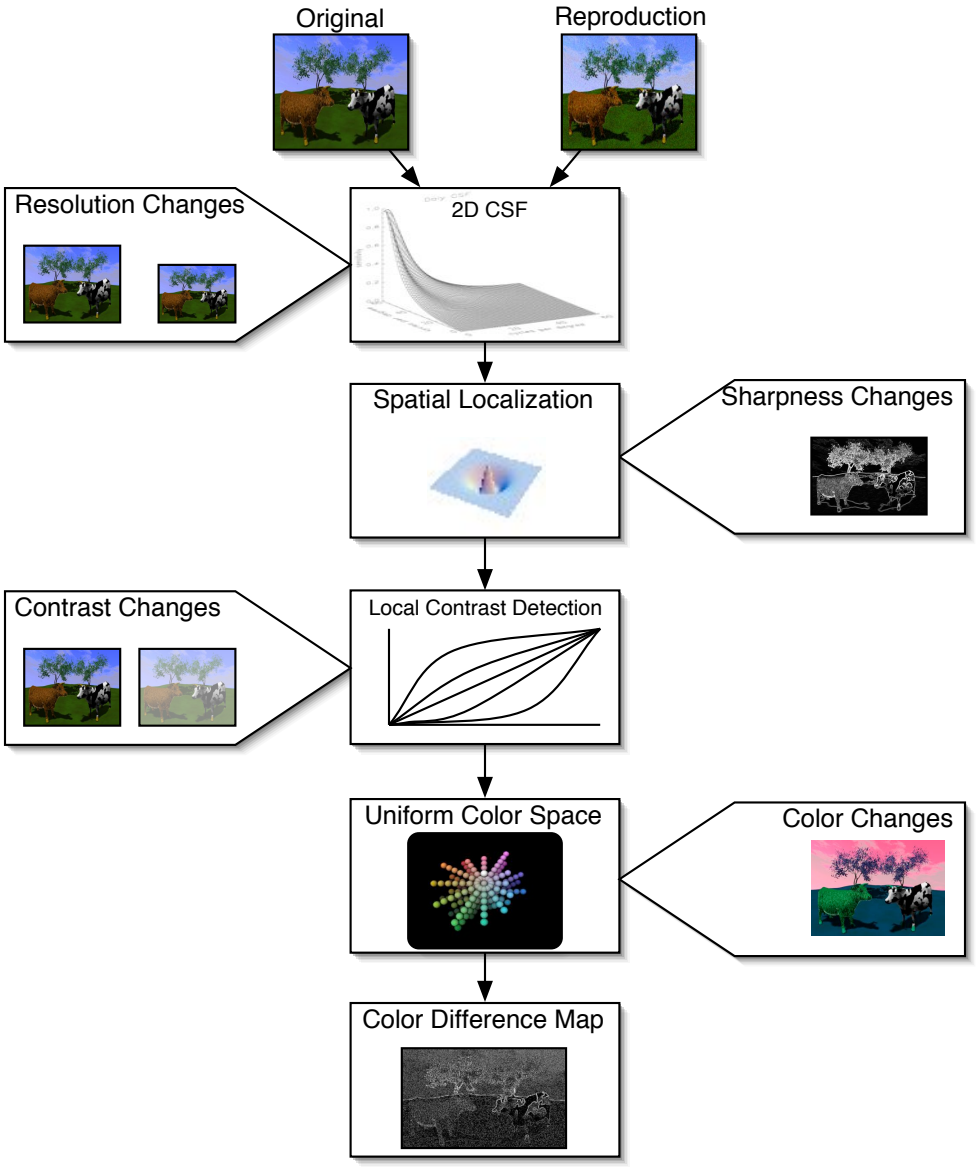


Figure 2.16: Image difference flow modules with associated causes of difference [30]

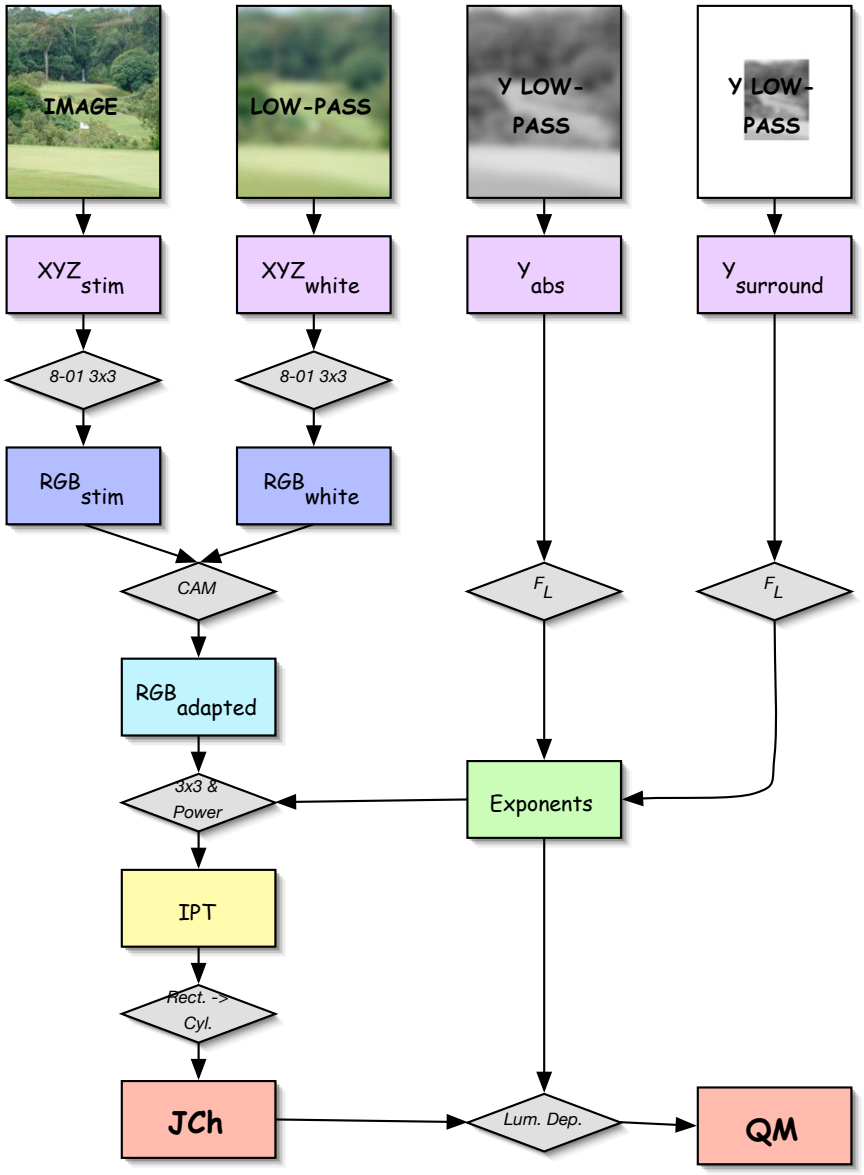


Figure 2.17: iCAM framework [30]

Chapter 3

Approach

This chapter describes the experimental approach used in this research. Overall the process can be divided into the following areas. First, the scene was constructed in a light booth. This scene was then photographed using special techniques and equipment. In parallel, a virtual model of the scene was built and used to render images. The renderings and photographs were then used in the compositing step to create the stimuli. The stimuli were presented to observers in a psychophysical experiment as well as iCAM. The results of those experiments were analyzed both individually, as well as in concert with each other. The details of each of the preceding areas is described below.

3.1 Lightbooth Scene Construction

3.1.1 Object Creation

There were several important requirements that influenced the design of the scene. First, it needed to be a controlled environment. Control in this context refers to the illumination both in terms of its color and geometry. This requirement was satisfied by constructing the scene in a standard viewing lightbooth. Second, in order to be rendered, the objects comprising the scene needed to be defined both geometrically and in terms of their material composition. As such, the author chose objects with well-behaved BRDFs, close to the idealized Lambertian or perfectly specular BRDFs, and simple in terms of their geometry, with the exception of the cow described later.

It was difficult finding a variety of objects that were approximately Lambertian, regardless of the geometric complexity. Several different approaches and materials were used. First, the author purchased wooden craft blocks and spheres, sanded them using fine grit sandpaper, and applied several coats of spray-paint primers [1]. This resulted in objects that were uniform and nearly diffuse, but with the most difficult type of BRDF to model. The painted wooden objects exhibited a BRDF with both specular and diffuse components as in Figure 3.1. The BRDF exhibited by the paint could be reasonably modeled using the built-in BRDF models in `pbrt`, however determining the values for the parameters is often not straightforward or physically based.

The author quickly realized that even approximately Lambertian surfaces are like point light sources and frictionless surfaces, existent only in the mind of a physicist. However, after continuously searching, it was noticed that racquetballs and foam bath toy blocks exhibited the 'nearly-Lambertian' surface the author sought and were included in the scene.

[1] The author consulted with local craftsman Mark Robinson regarding painting techniques to provide a uniform matte finish

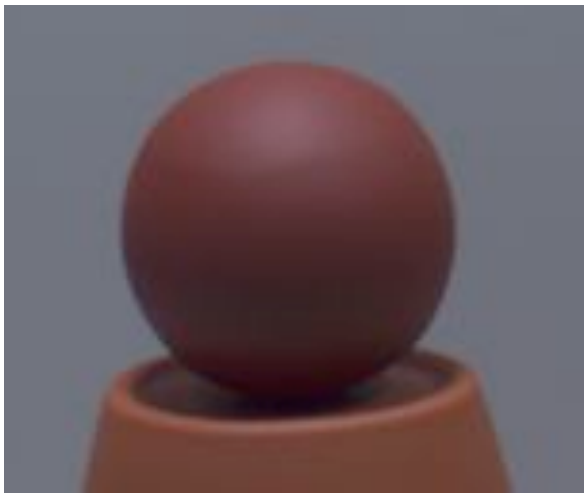


Figure 3.1: A wood sphere painted with a matte finish primer spray paint still exhibits a specular highlight so it cannot be assumed Lambertian.

Ultimately a photograph of the scene would be compared to a photograph of the scene including one rendered object. This object must exist both as a physical model, as well as a 3D model in the computer (see Figure 3.2). As this object was the focus for the research, the author wanted an object that was more realistic and intricate than a simple wooden block. The typical process is to choose a real object and then spend a large amount of time modeling that object in a modeling program such as Blender. This posed a problem as it would require great expertise in the use of a modeling package, and the author is a novice. This problem was averted by following the opposite path. First, a geometric file was purchased. This file was of a cow, and was extremely intricate, including textures. The file was emailed to a company called Stratasys [2], specializing in rapid prototyping. The company uses a process called Fused Deposition Modeling (FDM), which essentially prints a 3D version of the file by building up layer upon layer of extruded plastic. Different colors and materials can be chosen, and the final product is quite accurate and precise with resolutions in the thousandths of inches. The cow ordered was made of white ABS plastic, relatively strong and nearly opaque. In this case, nearly opaque

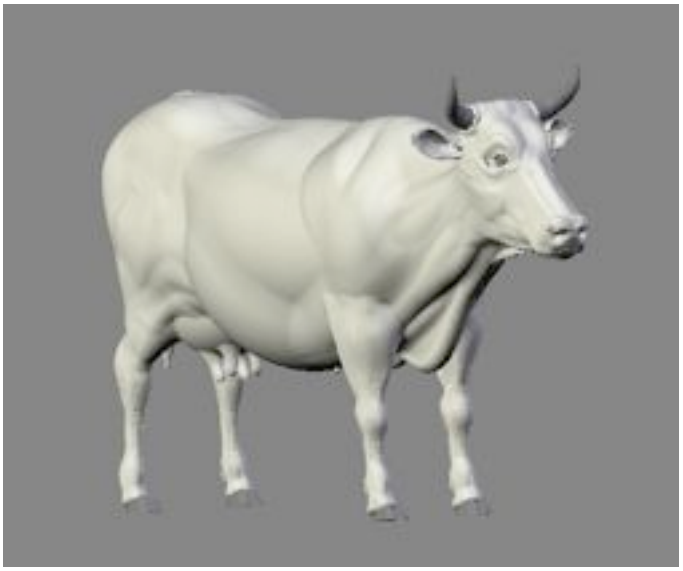


Figure 3.2: Screen shot from Blender showing a shaded view of the cow model.

implied the cow exhibited a fair amount of subsurface scattering, the impact of which will be discussed later. Finally, to complete the collection of objects to construct the scene, a mirror, terra cotta pot, and a clay vase were included.

3.1.2 Object Placement

As important as the object materials were the object placements within the booth and relative to each other. Recalling the ultimate objective of this research, the author wanted to accentuate differences in global illumination rendering algorithms. Therefore diffuse-diffuse and diffuse-specular interactions between the cow and other objects were created through careful object placement. First the cow was placed on a mirror, which created a significant amount of illumination on the underside of the cow, as well as a unique pattern of light on the large white vase in the back of the scene. Secondly, the multicolored foam blocks were placed almost in direct contact with the camera-side of the cow (see Figure 3.3). In addition to producing color

Figure 3.3: Photograph of the scene built inside of the light booth. Notice the reflection from the mirror onto the vase, as well as the color bleeding from the foam blocks onto the underside of the cow. These are the very visible indirect illumination effects reproduced in renderings physically through the use of a global illumination algorithm.

bleeding on the front of the cow, the object occlusion produces complexity that adds to realism and something an observer might encounter in real life. This scene could not be divided simply into local and distant scenes. In [11], the synthetic rendered objects were significantly distant from the surrounding scene and did not 'interact' with it. Additionally, the author was careful to place the objects with the less-idealized BRDFs further from the cow so as any interactions would be negligible. As will be discussed later in the compositing step, the effect of the cow on the scene is important as well, and includes effects such as shadowing and reflection on the multicolored blocks and the vase.

When looking at the final arrangement of the real scene and the objects, it looks like early computer-graphic renderings. The reason has to do with the nearly idealized BRDFs of the objects, which are significantly easier to model in a rendering package.

3.1.3 3D Model Constructions

After the scene was constructed, it needed to be modeled in the computer both geometrically and in terms of surface properties. Although the cow was the only object to be composited into the photograph, the entire scene required a virtual counterpart in order to model the object interactions. The entire scene was modeled in Blender, a freeware 3D software package [1]. As stated above, the cow model was purchased, and only needed the appropriate scaling to match the physical model dimensions printed using the FDM method. The remainder of the objects were all modeled individually. Some of these objects were very simple to model and included the wooden spheres and cubes, foam blocks, the mirror, and of course the racquetballs.

The physical dimensions of the vase were difficult to measure and model accurately for a novice such as the author. To aid in the modeling, a photo of the vase was taken, imported into Blender and used as a background image, analogous to tracing paper. The image was moved such that the vase was vertically bisected by the z-axis and seated on the xy plane. A Bezier curve was drawn over the photo that followed the profile of the vase. This curve was then 'spun' around the z-axis, forming a complete 3D model. The model was then scaled appropriately to match the proportions of the the real vase (see Figure 3.4).

Perhaps not obvious at first, the actual light booth needed to be modeled as well. At a minimum the interior and light geometry needed to be modeled. The author modeled the booth first and used it as a practice in learning Blender and thus accurately modeled the interior and exterior of the booth. This included the metal interlocking tabs on the floor of the booth. The final scene configuration changed throughout this work and in hindsight this level of detail in the

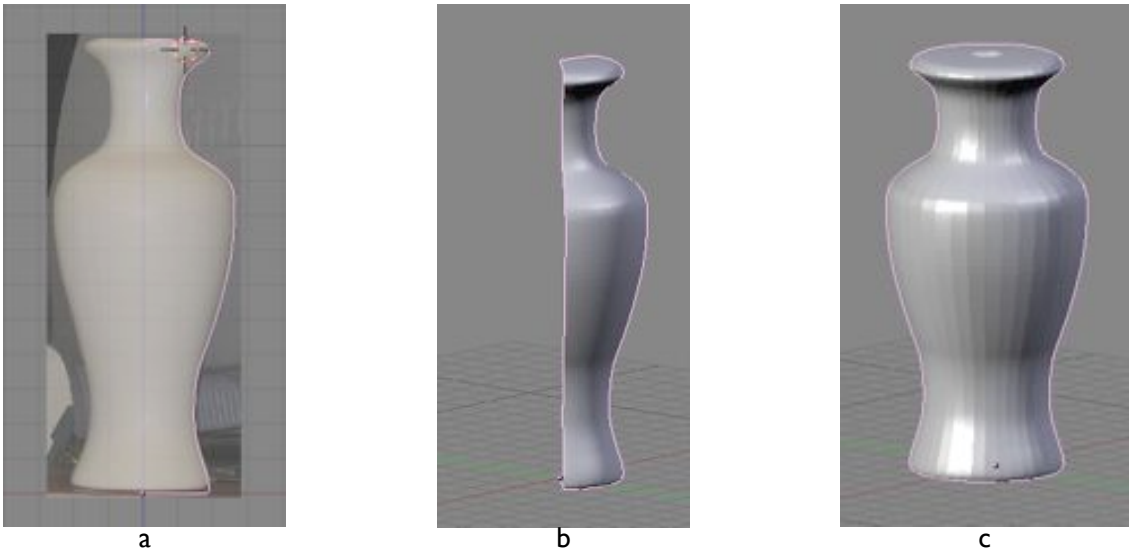


Figure 3.4: (a) is a screenshot from Blender using a photo of the vase as the tracing paper and the profile of the vase in pink. (b) shows the model after rotating the profile 135 degrees. (c) shows the complete vase model.

light booth model was excessive as the interactions of things such as the tabs and the cows are perceptually insignificant.

The virtual objects were placed inside of the virtual light booth to match the real scene. The reader can envision one of several ways to accomplish this such as direct measurement of all of the objects or a more complex, computer-vision approach. This was accomplished by using a technique similar to how the vase was modeled. First, the model of the light booth was inserted and centered in the scene. Next, a photograph of the scene was taken. This image was then imported into Blender and used as a backdrop. A virtual camera was placed in the scene, and its parameters (focal length and field-of-view) were matched visually to the imported photograph and known dimensions of the interior of the light booth. The virtual camera viewport was used to place models into the light booth. The are many more elegant solutions to this problem and can be found in the augmented reality literature. If the author was building multiple scenes,

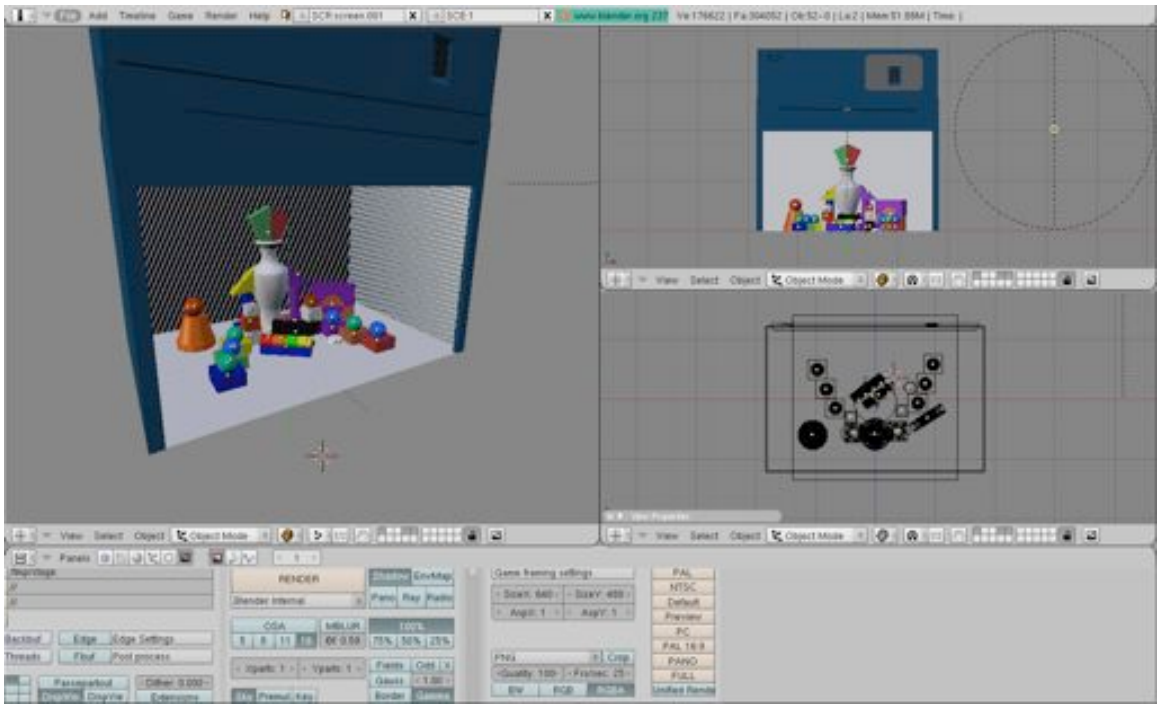


Figure 3.5: Screenshot from Blender showing various views of the complete scene model.

a alternate approach such as a calibration grid would have been useful. Since this is a very controlled instance of augmented reality, this approach was tractable and sufficient. Figure 3.5 shows the entire scene as it was being modeled in the software package Blender.

3.1.4 Material Parameters

To this point only the geometry of the objects has been discussed. However, `pbrt` must also know something about the optical properties of the object surfaces in order to render an image. These material parameters at a minimum include the 'color' and BRDF. The BRDFs of the objects were not measured for this research and were assumed 'idealized' in most cases. The surface reflectance characteristics (BRDF) were assumed Lambertian for the foam blocks, racquetballs, light booth interior, dark gray and black wooden blocks, and the cow. The mirror

was modeled using a perfect specular BRDF. The vase and red wooden objects were modeled using more realistic BRDFs. They were determined iteratively and visually.

The term color is perhaps not sufficient. The least ideal method (especially for a scientist) would be to adjust RGB sliders in the modeling package until the screen matched the object. There are many problems associated with this method and include things such as visual subjectivity, monitor calibration, and calibrated viewing conditions. Ideally, the author would have liked to render the images spectrally, and process the spectral cubes appropriately for a specific display. In this way, no information is discarded until the last possible step. This approach was not used, however, due to the extremely long rendering times required even for as little as 9 bands. The most reasonable approach was to measure the spectral reflectance of the objects using a spectrophotometer and then calculate XYZ values. These XYZ tristimulus values were then transformed to RGB triplets required by the renderer using a 3x3 matrix. This is a reasonable method in that it is based on instrumental measurements and standardized colorimetry. Therefore any color issues are likely a result of the renderer and not with the inputs, which is exactly what the author wants to exploit.

It is important to realize two important characteristics of the materials that help improve the accuracy of this approach. First, the materials chosen have a smooth spectral curve shape. If there were any spikes in the curves, they would not be appropriately sampled and color errors would result. Additionally, the materials chosen did not exhibit any fluorescence phenomena, which can only be accurately rendered spectrally [31].

The objects were measured on a Spectraflash 500 spectrophotometer with spectral coverage from 400 to 700 [nm]. The measured spectral curves are shown in Appendix A.1. XYZ values were calculated by multiplying the spectral reflectance ($r(\lambda)$) by the 1931 2 degree color matching functions ($\bar{x}, \bar{y}, \bar{z}$) by the spectral power distribution of the source ($S(\lambda)$) and then integrating as in Equations 3.1 - 3.3. From here, the values need to be input into `pbrt`. The authors

of `pbrt` make a point of stating how they do not use a simple RGB triplet model for their color. After looking through their source code, indeed they do perform calculations spectrally. However in the default configuration, `pbrt` is hardcoded to handle only 3 values, R, G, and B. The RGB values it expects are based on the International Telecommunication Union (ITU) HDTV recommendation (the same 3x3 as the sRGB matrix) [3]. Therefore, the XYZ values were multiplied by the 3x3 transformation matrix. Internally, `pbrt` takes those RGB values and immediately inverts them back to XYZ, performs the lighting calculations, and then converts back to RGBs using the XYZ to sRGB 3x3. It is a very roundabout process, but yields the expected results. Most importantly, it did not require any manipulation of their source code.

$$X = \int_\lambda r(\lambda)S(\lambda)\bar{x}(\lambda)d\lambda \tag{3.1}$$

$$Y = \int_\lambda r(\lambda)S(\lambda)\bar{y}(\lambda)d\lambda \tag{3.2}$$

$$Z = \int_\lambda r(\lambda)S(\lambda)\bar{z}(\lambda)d\lambda \tag{3.3}$$

The cow was originally measured using the procedure above. However, the results of the rendering process were not consistent with the photograph. After trying several different measurement techniques, the author noticed that the cow exhibited a fair amount of subsurface scattering. Therefore, when measuring with a spectrophotometer, some of the calibrated source was scattered out of the measuring aperture, and anomalous results were recorded. The author accounted for this by measuring the cow with the PR-650 under D65 simulated illumination. A piece of PTFE was also measured as the reflecting standard and ultimately the ratio of the two measurements was used as the spectral reflectance of the cow.

Finally, the source spectrum was mesaured. A PTFE diffuser was placed on the light booth floor. The booth was set to use the D65 (daylight) source (filtered tungsten). The PTFE was

measured using the PR-650. XYZ's and other triplets were calculated from the spectrum using principles of colorimetry as described above.

3.2 Scene Image Capture

After the scene was constructed, and material attributes measured, the scene was photographed. Typical consumer digital cameras make nice color photographs, but are not designed to perform accurate colorimetry. Since it was imperative to control the color throughout this process, it was necessary to use a camera over which the user would have more control and information. Another student in the lab was working with just such a camera for his Ph.D. research. The camera did not have a color filter array on the sensor, rather a color wheel in front of the camera. The filters were approximately a linear transform of the color matching functions.

The author assisted this student in calibrating the camera both in terms of its response and colorimetrically. The first step was to recover the response function of the camera for each of the three color filters. This was accomplished by photographing a target with a series of grayscale patches, for a range of camera exposures, and for each of the three color filters. The method is described in more detail in [12]. A series of photographs were then taken of the Macbeth ColorChecker DC. This data was used to determine the 3x3 matrix transform from camera digital counts to XYZ's. More details of this technique can be found in [5]. The result of the calibration work was a software package. This software package allowed multiple exposures through the three color filters to be combined into a colorimetrically calibrated, high-dynamic range XYZ photograph.

The camera was placed on a tripod in front of the light booth, looking downward slightly into the booth. Photographs of the scene were taken with exposures ranging from 1 second, to $1/60^{th}$ of a second for each of the three filters. Photographs of the booth with and without

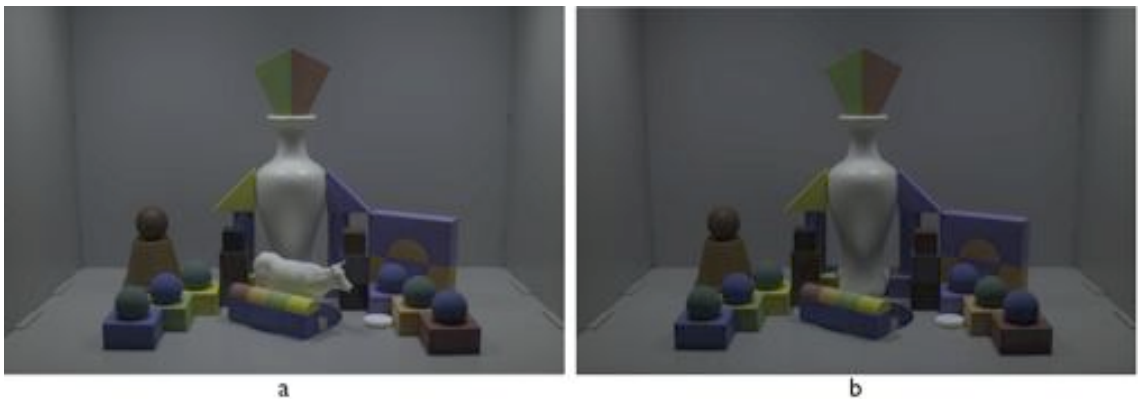


Figure 3.6: Photographs of the booth in linear XYZs and rendered as RGB directly, which is why they appear dark and incorrect in terms of color. Note the color banding in uniform areas, and the defined specular highlights on the top of the vase

.

the cow were taken using this method and can be seen below in Figure 3.6. Additionally, colorimetric measurements of the PTFE (XYZ for the 2 degree observer) were taken with the PR-650. The in-house camera software was then used to combine these photographs along with the measurement of the PTFE, into a high dynamic range image of XYZ values. At this point, the image contains high-dynamic range values. The image has not been tone-mapped in any way and still retains information in the very bright and very dark regions. For example, defined specular highlights can be seen at the top of the white vase at this point. However, in order to display these values in 8-bit monitor RGB's, values must be adjusted to fit within this range.

3.3 Rendering in `pbrt`

At this point, photographs of the booth have been taken and colorimetrically calibrated. The next step was to render the light booth scene in `pbrt` and use those results to composite the synthetic cow into photograph without the cow, Figure 3.6-b. It was not the author's intention to determine thresholds of realism for a given rendering method. Rather, the author wanted

to investigate coarse increments in several global illumination techniques to see how realism varied within and across algorithms. The goal was to span a large range of rendering times and realism, including the various algorithm artifacts. As stated earlier, this made `pbrt` an excellent research tool as it has a plug-in type architecture to test different techniques. `pbrt` comes with several surface integrators that were all used for this research and include direct, whitted, irradiance caching, photon mapping, and path tracing. In order to render an image, `pbrt` needs three things: a light source, objects, and a camera. The objects were geometrically modeled and attributed as described above. The light source also needs a geometry and color. In the real light booth, the manufacturer creates the daylight source by overdriving tungsten halogen lights and then filtering them to provide the appropriate spectral power distribution. The light bulbs were recessed into the ceiling of the booth and behind a diffuser used to create uniform illumination. In reality it would have been possible to geometrically model the diffuser and light source explicitly and have `pbrt` solve the illumination by tracing rays through the glass diffuser to the light sources. This approach was not taken, however, as it would have added a large level of complexity and rendering time. A simpler approach was taken. Ideally, if the diffuser was perfect, one could imagine using one single rectangular source the size of the ceiling of the light booth. This was clearly not the case and was observed by looking at the shadows formed by objects in the light booth. While the diffuser did distribute a fair amount of the light, faint shadows could be seen on either side of an object (see Figure 3.7). This indicated that there was slightly more power in and around the areas of the light sources directly behind the diffuser. In `pbrt` this was modeled using two disk sources 10 inches in diameter on either side of the booth ceiling center. An exact match is not required for this research, but the illumination needed to be fairly close so as not to be the dominant source of artifacts.

The color of the lights in `pbrt` were set to [10, 10, 10]. Recall that the default configuration of `pbrt` expects RGB triplets for the color of the objects based on the sRGB color transformation. If D65 XYZs are converted to RGBs, it results in equal R, G, B values.

Figure 3.7: (a) is the photograph of one of the blocks in the booth showing the shadows on both sides and (b) is the rendering of the booth demonstrating similar shadows.

.

Similar to the photographs, two renderings for each algorithm were completed, with and without the cow. Therefore, a total of 32 full-scene images were rendered. Two additional renderings were also done to assist in the compositing step, which will be described later. `pbrt` not only calculates pixel R,G,B's but also an alpha, α, value, which corresponds to the transparency of the material hit within a given pixel. For example, if an opaque object is hit, the $\alpha = 1.0$. If no object is hit, then $\alpha = 0.0$. Alpha values in between are possible even without transmissive objects. This occurs at the border of objects and is a result of oversampling and anti-aliasing. Figure 3.8 shows the alpha channel of the cow when it was rendered independent of anything else. In total, 34 images were required to produce the stimuli.

3.3.1 Computational Concerns

Computational resources were an important concern in rendering these images. Several different approaches were considered. Originally, all images were going to rendered on an Apple Powerbook G4 1GHz laptop. While possible, it would have completely monopolized that computer. Fortunately, `pbrt` is supported on most operating systems including Unix. Originally, `pbrt` was compiled for an SGI running Irix. This computer was ideal due to the 8 processors

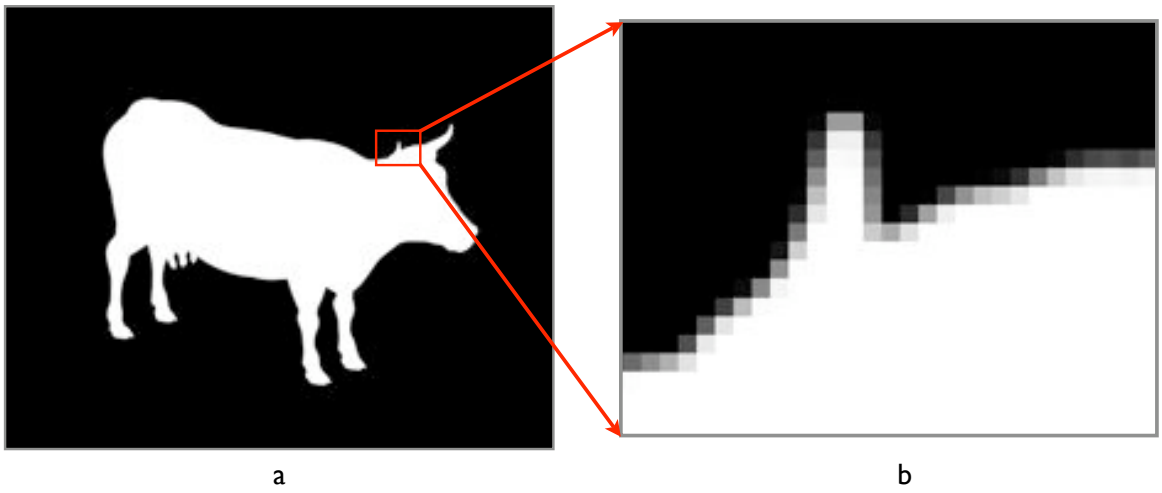


Figure 3.8: Alpha channel for the rendering of (a) the cow only and (b) zoom in of area around cow's horn, showing intermediate alpha values indicating partial transmission.

with 1 Gigabyte of RAM for each processor. However, Irix was one of the versions of Unix not officially supported, and eventually rendering errors precluded further use of the SGI computer. The author was eventually allowed exclusive use of two PowerMac G5 desktop computers with dual 2.0GHz processors and 2 Gigabytes of RAM each. `pbrt` is not multi-processor capable in its default configuration. However, it is possible to break a job into as many segments as desired and render them individually. This is possible because in ray tracing, each pixel is solved independently of other pixels. In combination with this feature, a Tcl script was written that divided a single job into n sub-jobs, one for each processor. Additionally, this script periodically checked rendering status, and would immediately start the next job when a processor was available. While not the most elegant solution, it sufficiently maximized and managed the rendering jobs.

3.4 Compositing

The basic idea of compositing is to layer various image elements into one complete final image. It is used in many motion pictures to combine computer generated elements with live action footage. At this point in the process there are two photographs and 34 rendered images. The idea was to use compositing techniques and differential rendering [11] in order to add the rendered cow to the photograph without the cow. Figures 3.9-3.11 show flowcharts of the different steps used to create the final composite image. Please refer to them throughout this section.

The Apple program Shake was used to assist in this process. Shake is a professional compositing package capable of manipulating high-dynamic range images with floating point computations, as well as processing sequences of images. Shake uses a node-based approach for its workflow. Nodes can be concatenated together to create an image processing workflow from beginning to end. Figure 3.12 shows a screen grab from the Shake program giving the user a sense of the node workflow concept. Additionally, the workflow can be exported and edited as a simple text script. The nodes operations include things such as color transformations, image filtering, and warping. While this research utilized relatively simple nodes, the use of Shake provided an extremely solid and optimized platform that saved the author a significant amount of time. The basic process is explained below.

First, the photographs were imported into the Shake. The photographs were normalized to the Y component of the PTFE (Y of the PTFE will be 100). This is what has previously been referred to as 'relative colorimetry'.

This next step involved extracting the cow from the renderings in order to composite it into the photograph. The simplest way to think about this step is to imagine a cookie cutter that cuts the cow out of the image. This is the basic idea of a multiplying the image by a matte.

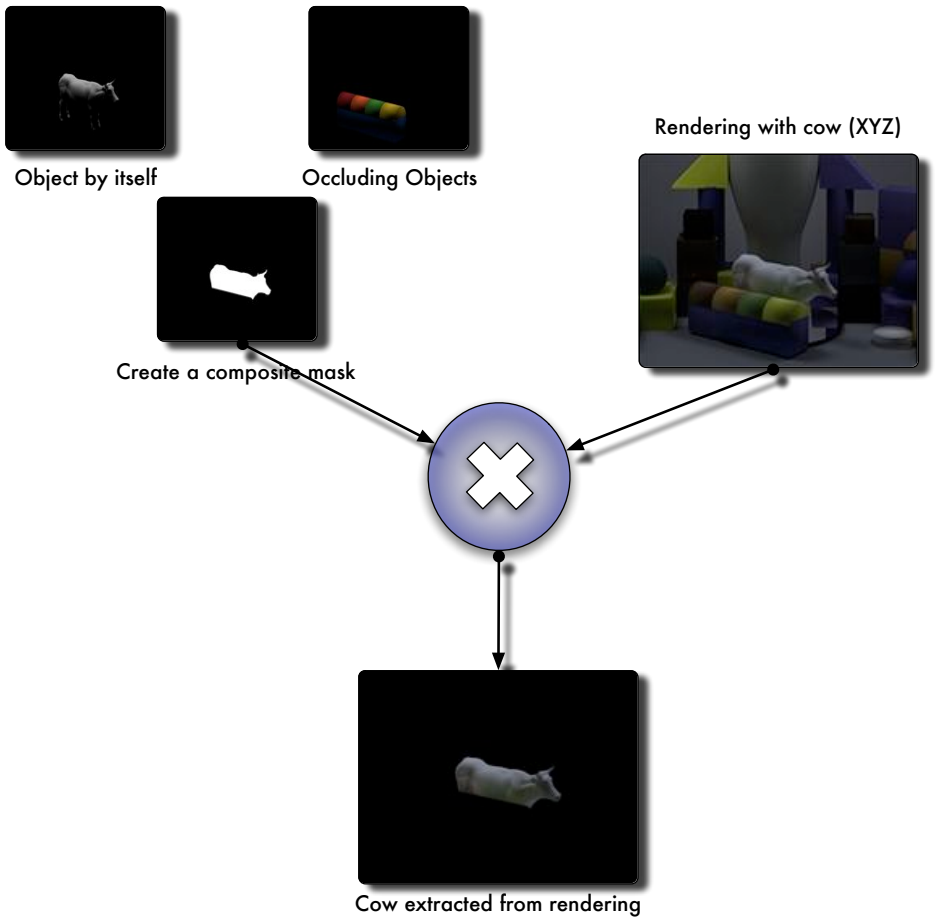


Figure 3.9: The steps required to extract the occluded cow from the renderings.

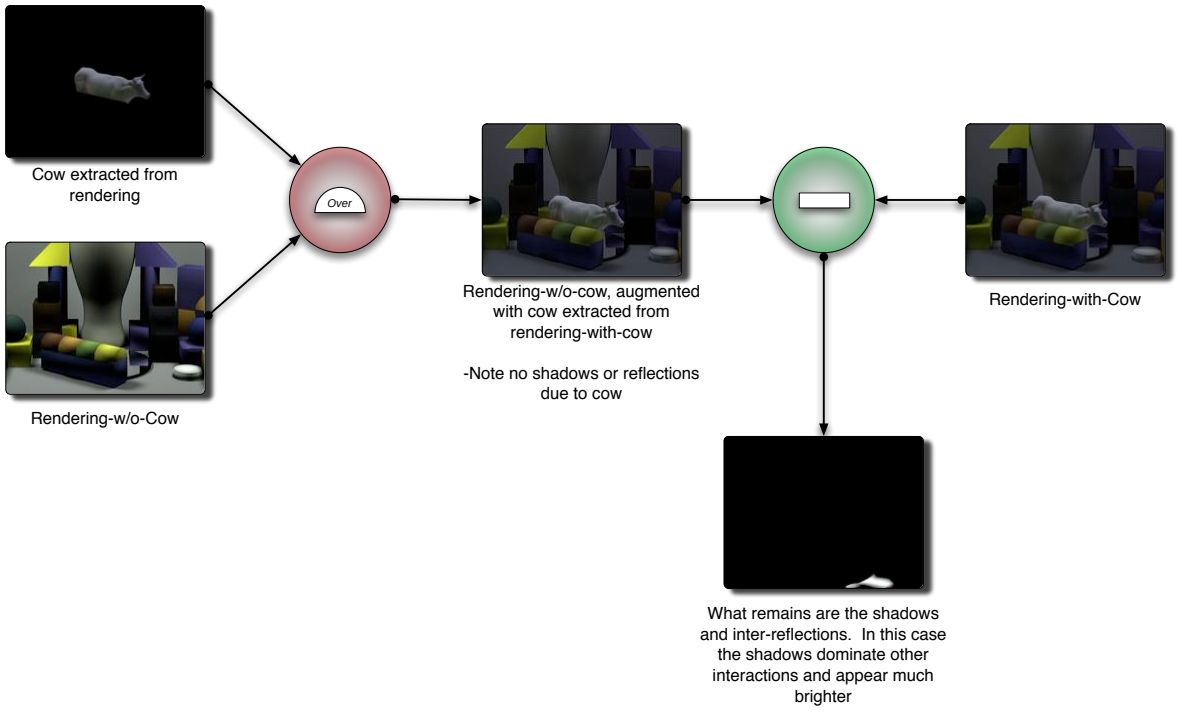


Figure 3.10: The steps required to extract the shadow and other lighting interactions not included in the cow.

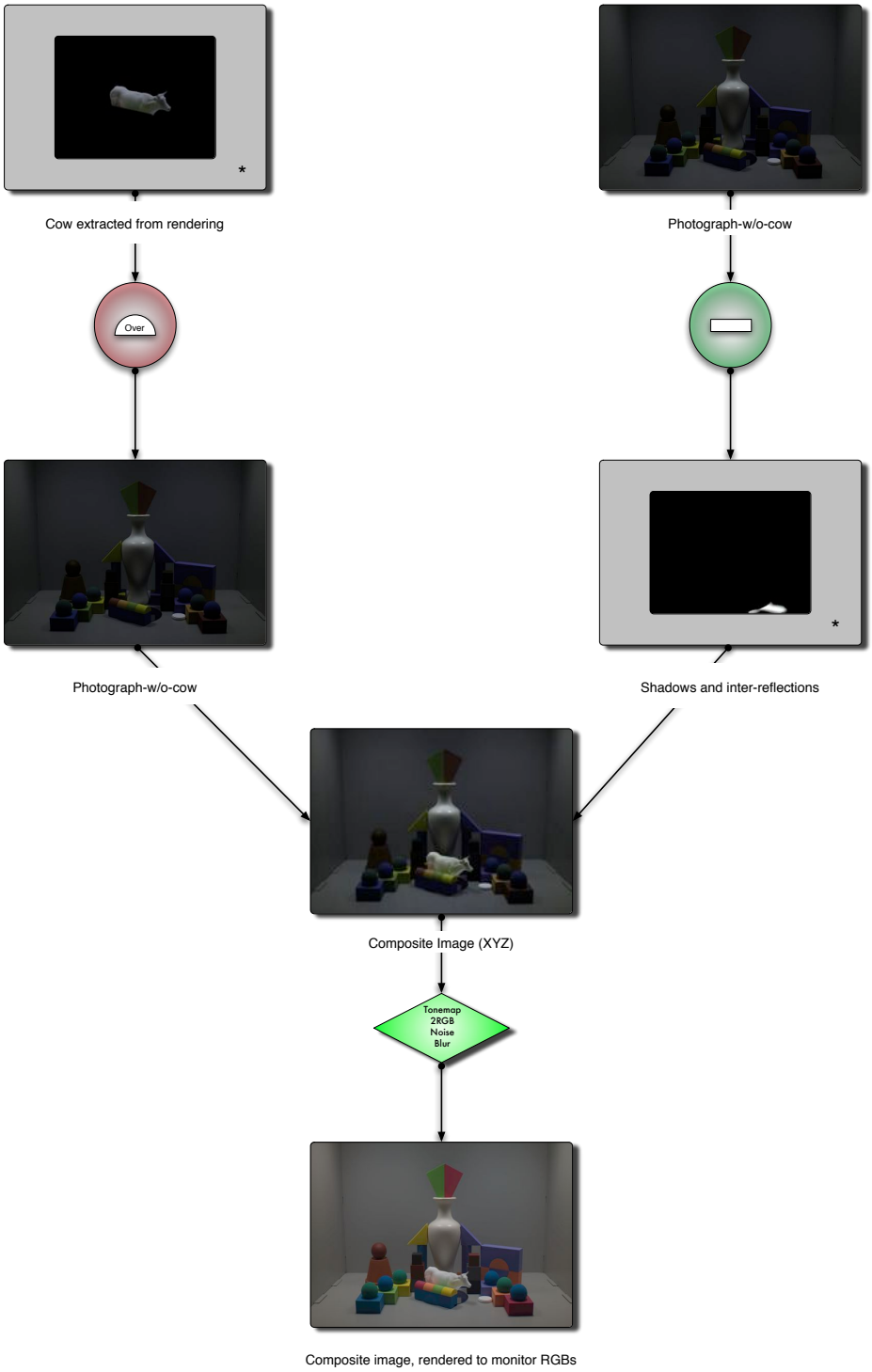


Figure 3.11: Steps to add the extracted object and other illumination interactions to the photograph to create the full composite image.

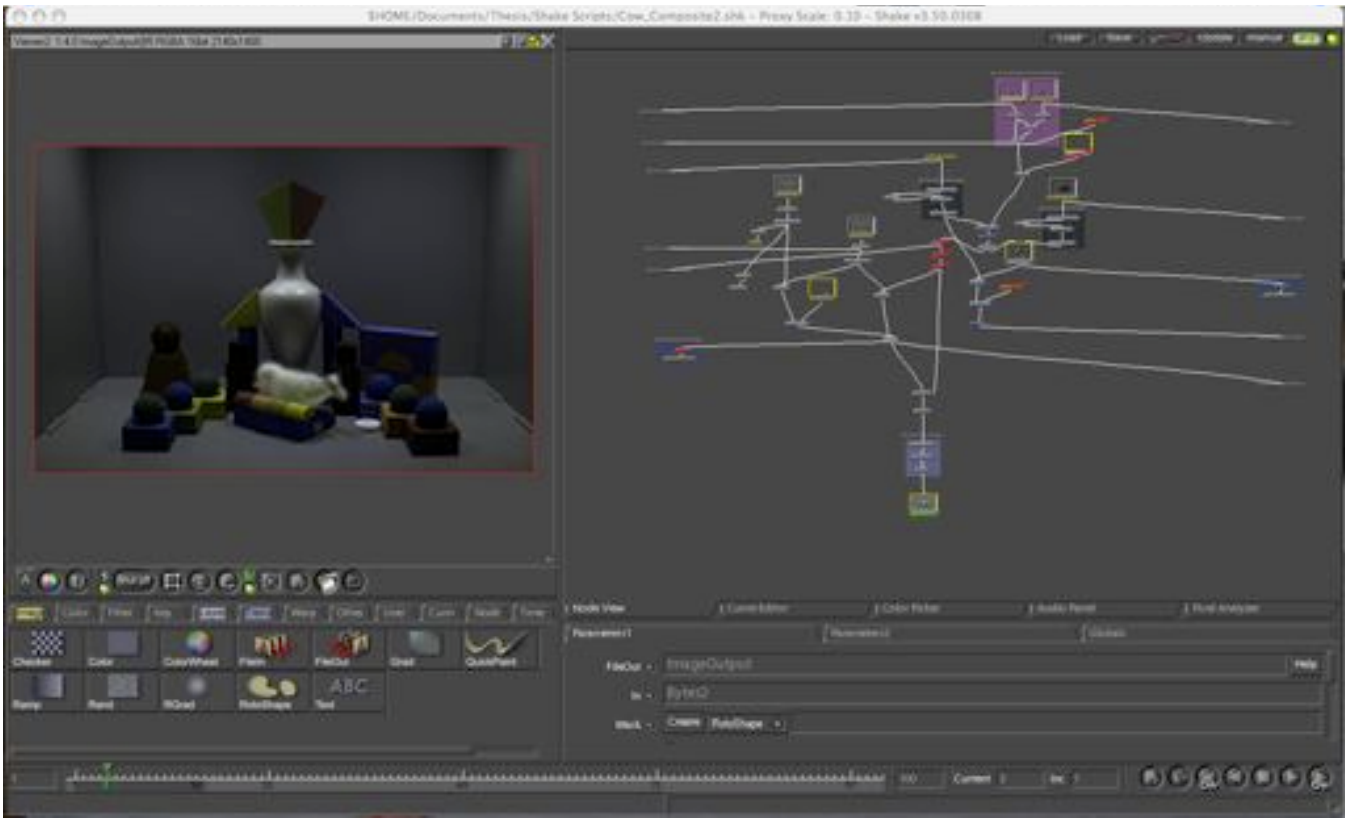


Figure 3.12: Screen grab from the Shake compositing program.

The matte contains zeros in areas to be excluded from the photograph, and values greater than zero (primarily ones), elsewhere. The scene was constructed such that the cow was behind an occluding object. Therefore, the matte must take that into account. The matte image itself was a combination of individual renderings of the the cow and occluding block objects.

Recall that the rendered images were in `pbrt`'s default RGB space. The first step in processing the renderings in Shake was to convert them to XYZ. This was done using the 3x3 matrix from `pbrt` as shown in [3]. Next a chromatic adaptation transform was applied. Due to the problems with the spectrophotometric measurements of the cow (Section 3.1.4), it was rendered in `pbrt` using the same RGB triplet as the measured PTFE. This resulted in a cow that was

too white. In order to account for this, a chromatic adaptation transform was applied in Shake using Equation 3.4 and 3.5. As [16] states, the process begins with a "transformation from CIE tristimulus values (XYZ) to sharpened cone responsivities (RGB) using the M_{CAT02} matrix transformation.

$$\begin{vmatrix} X_2 \\ Y_2 \\ Z_2 \end{vmatrix} = M^{-}1_{CAT02} \begin{vmatrix} R_{adapt2} & 0 & 0 \\ 0 & G_{adapt2} & 0 \\ 0 & 0 & B_{adapt2} \end{vmatrix} \begin{vmatrix} 1/R_{adapt1} & 0 & 0 \\ 0 & 1/G_{adapt1} & 0 \\ 0 & 0 & 1/B_{adapt1} \end{vmatrix} M_{CAT02} \begin{Vmatrix} X_1 \\ Y_1 \\ 0 \end{Vmatrix} \quad (3.4)$$

$$M_{CAT02} = \begin{vmatrix} 0.7328 & 0.4296 & -0.1624 \\ -0.7036 & 1.6975 & 0.0061 \\ 0.0030 & 0.0136 & 0.9834 \end{vmatrix} \quad (3.5)$$

The RGB values are then divided by the adapting RGB values for the first viewing condition and multiplied by the adapting RGB values for the second viewing condition prior to a linear transformation back to corresponding CIE tristimulus values." Figure 3.13 shows the XYZ composite images with and without the chromatic adaptation applied. A pixel analyzer node was then used to examine a region of pixels on the surface of the PTFE. These values were input into a node to normalize the image such that $Y_{PTFE} = 100$. The matte, created several steps earlier, was used to extract the cow from the rendering. This was done using a 'switchmatte' node in Shake, which simply copies the selected channels from the second image, to the alpha channel of the first image. By definition of an alpha channel, the image is multiplied by the values in the alpha channel (between 0 and 1) to determine a pixel's visibility. At this point, the cow has been properly extracted from the renderings and can be composited onto the photograph of the booth without the cow.

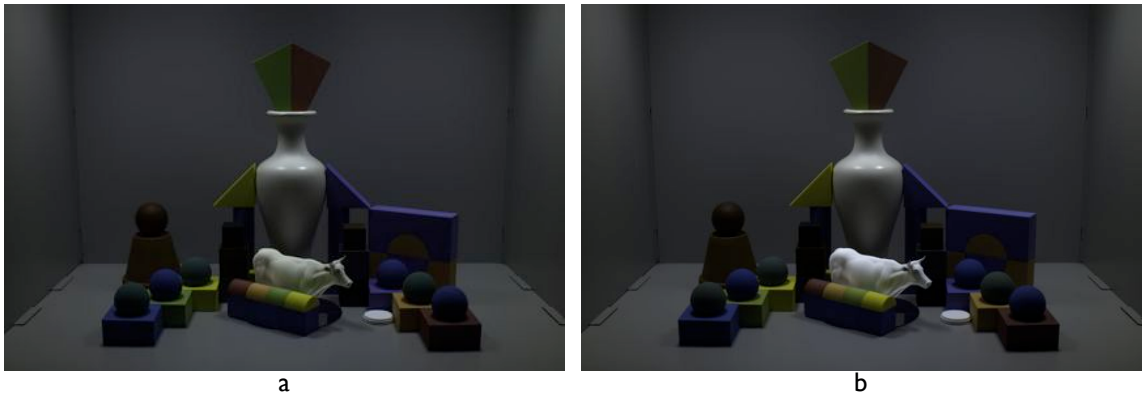


Figure 3.13: These two images show the difference (a) with and (b) without the chromatic adaptation applied. It is obvious even displayed in XYZ, that the cow is too white (b), as compared to Figure 3.6-a

In rigorous augmented reality applications the camera position in three-space is explicitly known, or determined through the use of computer vision techniques and/or calibration targets. Complex computer vision techniques were not used in this research. Rather, the gross location of the camera was measured relative to the center of the lightbooth. These coordinates were then used as a starting point, and refined visually through iterative rendering. Also in typical augmented reality, the camera used is thoroughly specified and measured. The internal geometry of the camera must be known. In this research, a complete geometric calibration of the camera used was not done. Instead, a couple of quick measurements were taken, and a fixed focal length used in order to determine the field of view. It was not necessary to measure every parameter because a simple camera model was used in `pbrt` and only required the field of view and camera position.

The renderings were created using these parameters. It is obvious that they are not an exact match to the photographs (see Figure 3.14). In general, they would be acceptable and believable if they were not being compared side-by-side to the original photograph. Additionally, these large errors would cause problems with iCAM image difference calculations. In order to mitigate

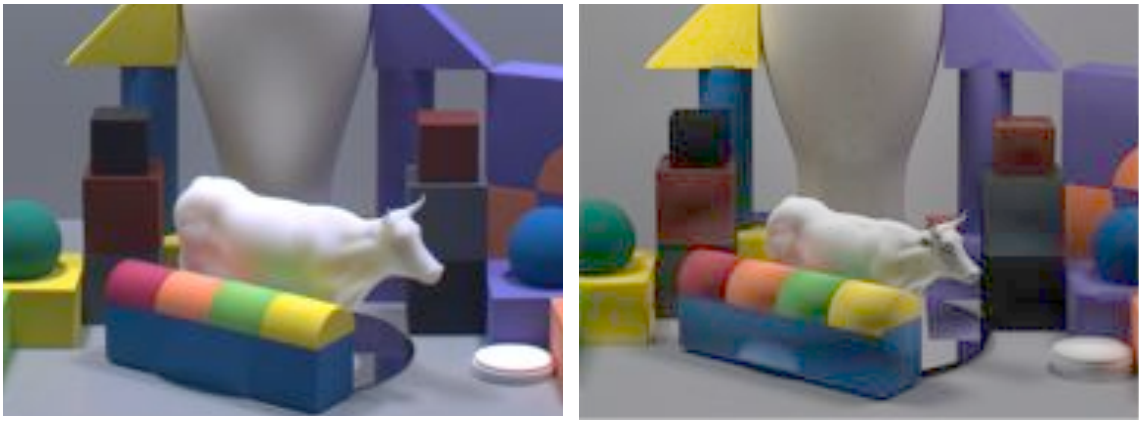


Figure 3.14: This figure shows approximately the same area in the (a) photo, and (b) one of the renderings. It is noticed that the camera geometry was not exactly matched between the two.

these problems, the extracted cow was moved and warped in Shake, rather than completely re-rendering the images. The justification for this was two-fold. First, all of the renderings were manipulated identically. Second, if the camera position and geometry were perfectly specified, an advanced renderer such as `pbrt` would have no problem calculating the correct image. Again, this research was not to compare `pbrt` against another renderer, rather it was a comparison of global illumination algorithms within `pbrt`. Therefore the decision was made to make a simple calculation in Shake that would take less than a minute to calculate instead of re-rendering, which could take several days in certain cases.

Just as the cow was extracted from the rendering, the same was done for the interactions. These interactions included things like shadows and object inter-reflections. The general procedure to isolate these interactions was to simply subtract the rendering without the cow from the rendering with the cow. However, using the arguments above for the camera position errors, it was decided not to incorporate these interactions from the renderings. Rather, the shadow from the cow was extracted from the photograph, and composited into the photograph with

the rendered cow. The remainder of the interactions were perceptually negligible due to scene design, or were not captured in the renderings and thus not included.

At this point the photograph contains the rendered cow with shadows, and is ready to be saved from Shake. Shake supports floating point pixel values, but typical displays only support 8-bit integers. In order to prepare the images for display they were normalized to the brightest pixel in the image, which turned out to be the specular highlights of the sources on the vase. The images were then converted to 16-bit and written out as 16-bit TIFF files. Finally, the images were resized to [859 by 564] pixels using the Mitchell filter by default. The resize was performed to prepare the images for display in the psychophysical experiment. The images were named sequentially, which allowed Shake to automatically process all of the images without user interaction.

3.5 Display Characterization and Rendering Images to Display

The stimuli for the experiment have been created. The next step was to convert them from the 16-bit TIFF format, to an 8 bit image based on the colorimetric calibration of 22" Apple Cinema liquid crystal display (LCD). The calibration procedure was based on [10]. The general idea was to display carefully selected color patches on the liquid crystal display and measure them with an LMT colorimeter. Three one-dimensional look up tables (LUTs) were created based on measurements of primary ramps. These LUTs were used to invert the nonlinearities of the display, manufacturer imposed in the case of LCDs. A matrix is calculated in this linear space to convert XYZs to monitor RGBs based on the input versus measured values. The matrix was further refined through an optimization routine to minimize CIEDE2000, an equation to calculate the difference between colors in an approximately uniform color space [5]. The three one-dimensional LUTs and the matrix constitute the required parameters to construct a monitor model, necessary to convert between XYZ values (floats) and monitor RGBs (0-255).

The 16-bit TIFF XYZ images were read into Matlab and immediately converted to double precision. The pixel value of the PTFE in the image was retrieved and a scale factor computed based on the measured XYZ of the PTFE in the light booth with the PR-650 (0.9605, 1.040, 1.11120). The image was divided by this scale factor and then multiplied by 100, effectively scaling the image such that the PTFE pixel values are X = 97.3668, Y = 102.2528, Z = 111.5436, very close to a perfect diffuser under a perfect D65 illuminant. The image was reshaped to two dimensions, (height * width, 3), and the measured LCD model applied. It is important to understand that any XYZ can be run through the matrix and three one-dimensional LUTs to obtain an RGB value specific to that monitor. However, care must be taken to ensure the value is within the monitor gamut, and if not, handled appropriately. In the case of this research, any values outside the monitor gamut were simply clipped. After all of the images had been processed by this model, they were ready for the psychophysical evaluation.

3.6 Comparing the Images

The final step is to compare the images against one another. The stimuli were presented to observers using a paired-comparison psychophysical experiment. In paired comparison, the observer is presented with two stimuli, and asked to choose one based on some criterion. The paired-comparison experiment was analyzed using Thurstone's Law of Comparative Judgement, Case V. The result was an interval scale of quality.

In this experiment, the observer sat 36" from the screen, and was presented with three images on screen, the original photograph on top, and the pair of images on the left and right halves of the bottom of the screen, see Figure 3.15. The observer was asked "Choose the image on the bottom of the screen by clicking on it, that is most like the image on the top. This is an accuracy, not preference judgement. You may focus your attention in and around the area of the cow" After the observer clicked on one of the images, three noise images were displayed

for one second, and the next pair was presented. There were 17 total images, 16 that included the rendered and composited cow, and the completely real photograph. This implies that there were trials where the photograph was presented along with one of the renderings. Also, all trials were unique in that there were no trials where both images were the same. In total there were $(n)(n-1)/2$ total trials, where n is 17, for a total of 136 trials. It should be noted that although the real photograph was the image presented on top for every trial, it was not explicitly stated. Also, the observers were told to look in and around the area of the cow. This was to reduce any noise due to the randomized presentation order. In other words there were trials where it was very obvious that the only difference between images was the cow, and there were others where this was not the case. After a few trials though, every observer would be focusing their attention around the cow and not in the areas of the image that remained constant.

The images were also evaluated using iCAM in the image difference configuration. Each of the 16 composite images were compared against the original photograph, resulting in image difference maps (see Figure 3.16). Bright pixels indicate a larger difference between the photograph and the test image. One of the inputs to the image difference model is the viewing distance. This is used to calculate the pixels per degree as shown in Equation 3.6, which in turn blurs the image appropriately to provide the spatial adaptation stimulus. The resulting contrast sensitivity function (CSF) used by iCAM enhances perceptible frequencies and modulates those frequencies that are less perceptible. In the equation, 100 represents the pixels per inch of the monitor and ppd is the pixels per degree of visual angle. Note the multiplier of 2, which converts from cycles per degree to pixels per degree. Each image was processed by adapting to the frequency content and luminance of themselves. After the images have been processed they were compared to the original processed photograph. Differences were only calculated for the cow and surrounding area. This was to match the psychophysical experiment where observers were instructed to focus their attention in that area.

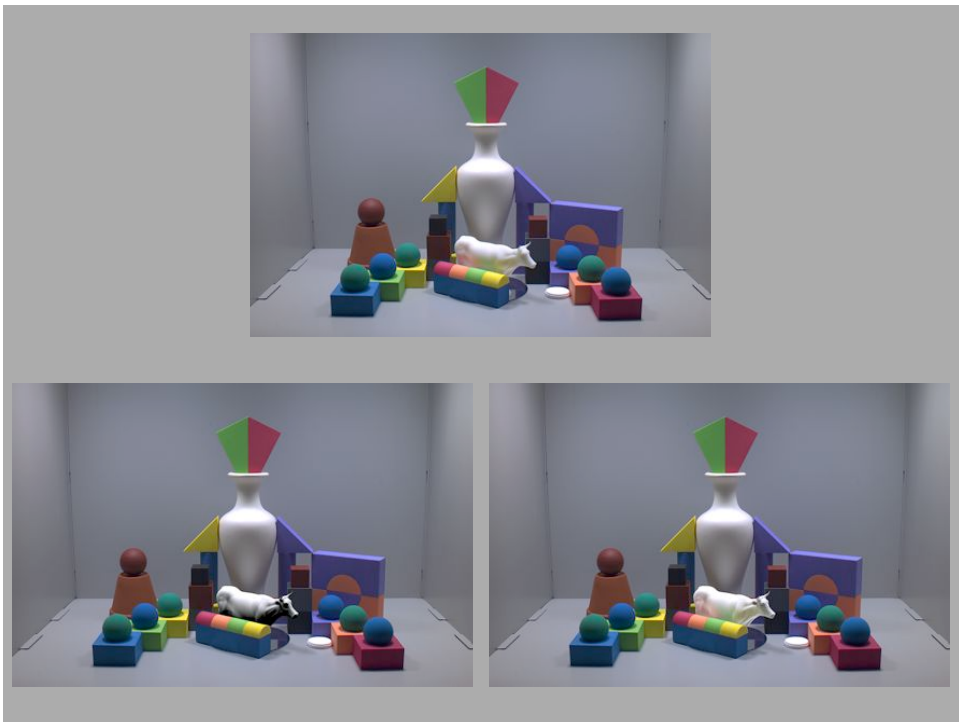


Figure 3.15: This image depicts the display presented to observers for the psychophysical experiment. The reference photograph was placed in the top-middle portion of the screen, and the two photographs to choose from were placed at the bottom of the screen, over a neutral gray background.

$$ppd = \frac{100[ppi]}{\frac{180}{\pi}\arctan(\frac{1}{viewdist})}2 \tag{3.6}$$

The difference calculation was performed as follows. First the images were individually processed both spatially and colorimetrically using the iCAM modules. The resulting images were in the IPT [15] color space, an opponent color space. Then each of the composited images were subtracted band-by-band from the photograph. The bands are squared, summed, square rooted and then cropped to the area around and including the cow, resulting in the image difference map as shown in Figure 3.16. Two types of analysis can be done on these images. The first

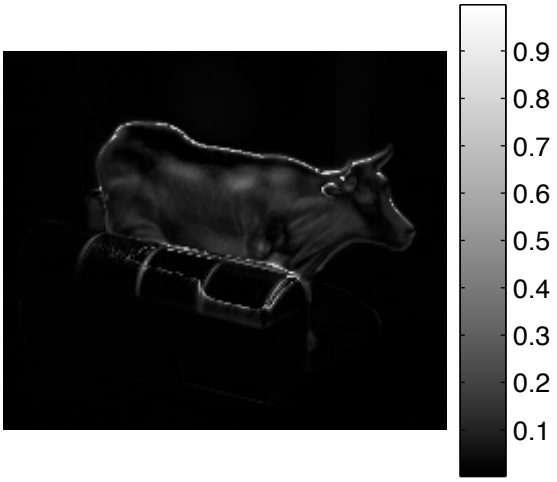


Figure 3.16: This image shows an image difference map between the photograph and one of the renderings.

was to look at each difference map individually to look for trends. The second was to reduce these maps into a single number so relationships with other variables can be explored. Several different techniques were performed for the latter. They included finding the maximum value, median value, as well as several different percentile values. There are a limitless number of ways to reduce this data, however only a few make sense. The idea was to choose methods that can be explained at least initially in a perceptual manner, with the intent for others to continue this research. These results are in Chapter 4.

Chapter 4

Results and Discussion

4.1 Renderings

In this section, the unprocessed renderings are shown. They have not been colorimetrically adjusted, and therefore are in `pbrt` RGBs. Additionally, they have not been normalized by the PTFE and therefore appear darker in order to display as much detail as possible. Figures 4.1-4.3 show the `pbrt` parameters set for each rendering, and the abbreviated name that will be used in the rest of this document.

Image	1	2	3	4	5	6	7
maxdepth	5						
GI	W	W	D	D	P	P	P
SPP	1	16	1	16	16	128	1024
Time [s]	7.4	88	103.7	1738	225	1909	16536
Name	Whitted_1	Whitted_16	Direct_1	Direct_16	Path_16	Path_128	Path_1024

Figure 4.1: The rendering settings for the Whitted, Direct, and Path tracing integrators.

Image	8	9	10	11
spp	16			
maxspeculardepth & maxindirectdepth	5			
maxerror	2.0	2.0	0.02	0.02
nsamples	256	4096	256	4096
Time [s]	2912	3031	4066	40,000
Name	Irrad_2_256	Irrad_2_4096	Irrad_02_256	Irrad_02_4096

Figure 4.2: The rendering settings for the Irradiance Caching integrator.

Image	12	13	14	15	16
spp	16				
maxdepth	5				
directwithphotons	T			F	F
finalgather	T	F		T	F
directphotons	10M			-	
indirectphotons	1M				
causticphotons	20k				
nused	150				
maxdist	1.0	2.0	1.0	2.0	2.0
finalgathersamples	64	-		64	-
Shoot Time [s]	411	411	388	131	114
Render Time [s]	54K	1463	1214	53K	2395
Name	Photon_d_1.0_nfg	Photon_d_2.0_nfg	Photon_d_1.0_fg	Photon_i_fg	Photon_i_nfg

Figure 4.3: The rendering settings for the Photon Mapping integrator

The first two images (see Figure 4.4) are the renderings using the Direct Illumination surface integrator with 1 and 16 samples per pixel respectively. The difference in the number of samples per pixel essentially changes the total number of rays cast. Increasing the number of samples per pixel using this integrator is essentially anti-aliasing the final image. This can be seen by looking at the boundary between the top of the cow's back and the vase. In the image using only one sample per pixel, [spp], that line is jagged, as opposed to the smooth line seen in the image with 16 [spp]. Looking closely, one can see this effect on the borders of all of the objects. Additionally, more noise is seen in the shadows of the image with 1 [spp] image. It is also noticed that the images rendered using the Direct illumination are very dark, especially on the underside of the cow and on the vase. This is due to the fact that as the name suggests, this surface integrator only accounts for the illumination that reaches a surface directly from the sources, and not indirect illumination such as light that bounces off of the mirror, onto the underside of the cow.

Figure 4.4: Direct illumination integrator renderings with (a) 1[spp] and (b) 16 [spp] respectively.

The next pair of images (see Figure 4.5) were rendered using the Whitted integrator. The image on the left corresponds to using the Whitted integrator with 1 [spp], and the on the right is the

Whitted with 16 [spp]. There are several things to notice in these images. First, as with the Direct, the difference in the samples per pixel changes the number rays cast. Thus the image with 16 [spp] shows smoother edges on the objects. It is also noticed that these images are very dark, again like the Direct illumination surface integrator. Unlike the Direct, Whitted does not explicitly exclude all indirect illumination effects. Whitted supports recursion for perfectly specular or perfectly refractive surfaces, in other words, surfaces with delta type BRDFs. Since all but the mirror are non-delta BRDFs, there is seemingly no indirect illumination effects. The other obvious artifact in the Whitted images is the increased variance noise. The reason for this is that since Whitted assumes simple recursive ray-tracing, it only samples the light source once for each ray, as opposed to the Direct illumination where even with only 1 [spp], it still computes the integral, sampling both the surface BRDF and the light source with 32 [spp] as specified in the `pbrt` configuration files.

Figure 4.5: Whitted integrator renderings with (a) 1[spp] and (b) 16 [spp] respectively.

The next three images were rendered using the path integrator. The path integrator is similar to the Whitted in that it traces a ray into the scene, but will sample non-delta BRDF's and non-point sources to get an estimate of the indirect illumination. The only variable for the

path integrator is again the samples per pixel, essentially the number of rays cast into the scene. The images from let to right were rendered with 16, 128 and 1024 samples per pixel respectively. Again, with the path integrator the dominant source of noise, is noise of variance. The variance noise is prevalent in Figure 4.6.a and is seen as the high-frequency pixel brightness variation. With only 16 [spp], the noise is very apparent and distracting. As more rays are cast, the variance decreases towards the mean value in a predictable manner. As a further demonstration of this, a region corresponding to the back wall of the light-booth was selected and the mean and standard deviation were calculated for each of the three images. In Figure 4.7, the standard deviation is shown decreasing while the mean remains virtually the same for all three images.

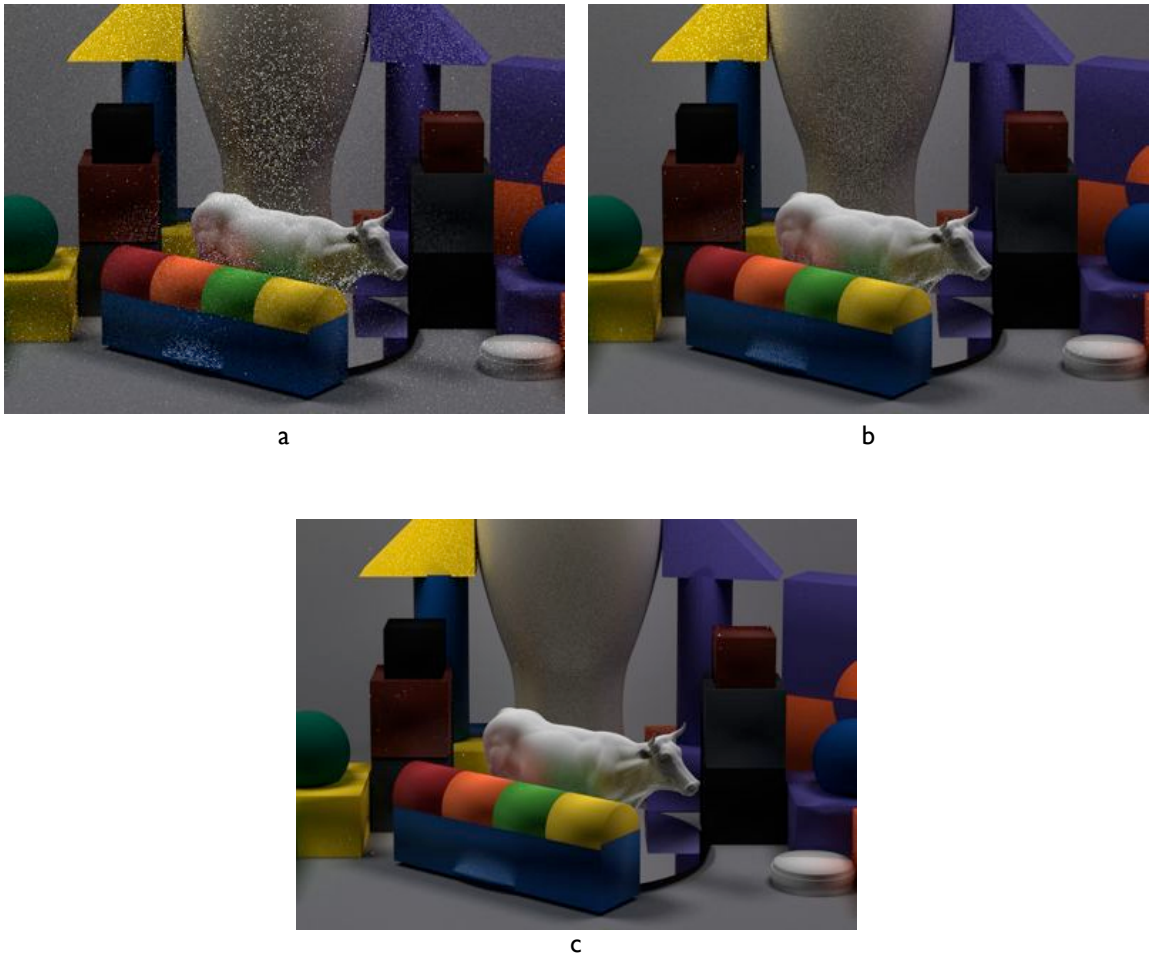


Figure 4.6: Path integrator renderings with (a) 16[spp] and (b) 128 [spp], and (c) 1024[spp]. Notice the decrease in high frequency noise (variance) with an increased number of samples.

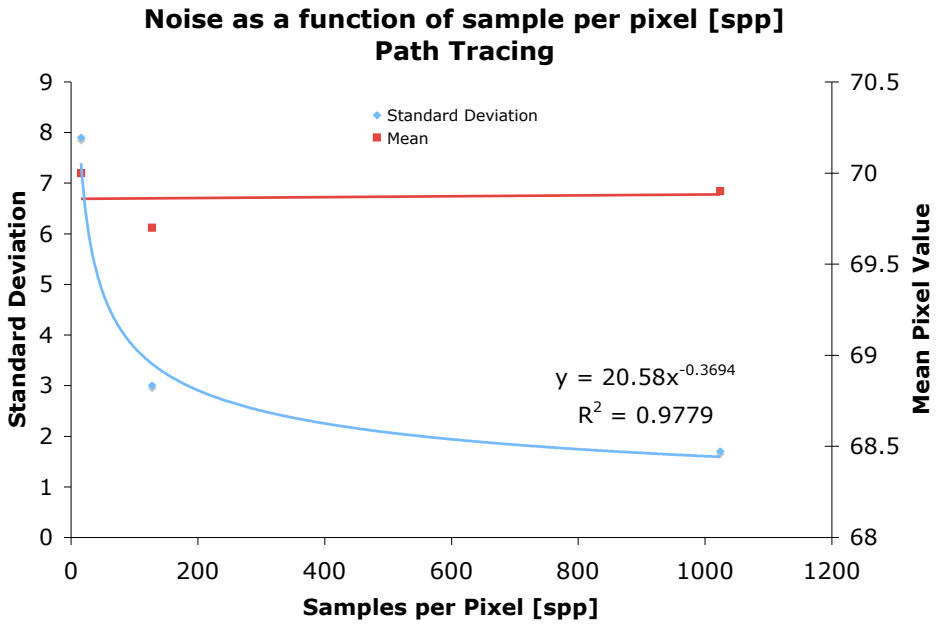


Figure 4.7: The noise decreases predictably with an increase in samples per pixel for the path tracing integrator. In addition, the average pixel value is approximately constant regardless of the number of samples per pixel.

The next four images were rendered using irradiance caching (see Figure 4.8) to calculate the indirect illumination component. The two parameters adjusted were the error metric value, and the number of samples per pixel. Two levels of each were chosen; 0.02 and 2.0 for the error metric, and 256 and 4096 samples per pixel. Four total images were rendered using the combinations of these values. The absolute values of the error metric are perhaps meaningless, but their relative relationship is not. As described earlier in Section 2.1.1, irradiance caching pre-calculates irradiance samples at various locations. The error metric is used as a threshold to determine whether or not additional samples should be calculated at render time. The larger the value, the more error the algorithm is 'willing' to allow before calculating more irradiance samples on the fly. In other words, the ray tracer will calculate the final result using fewer cached samples.

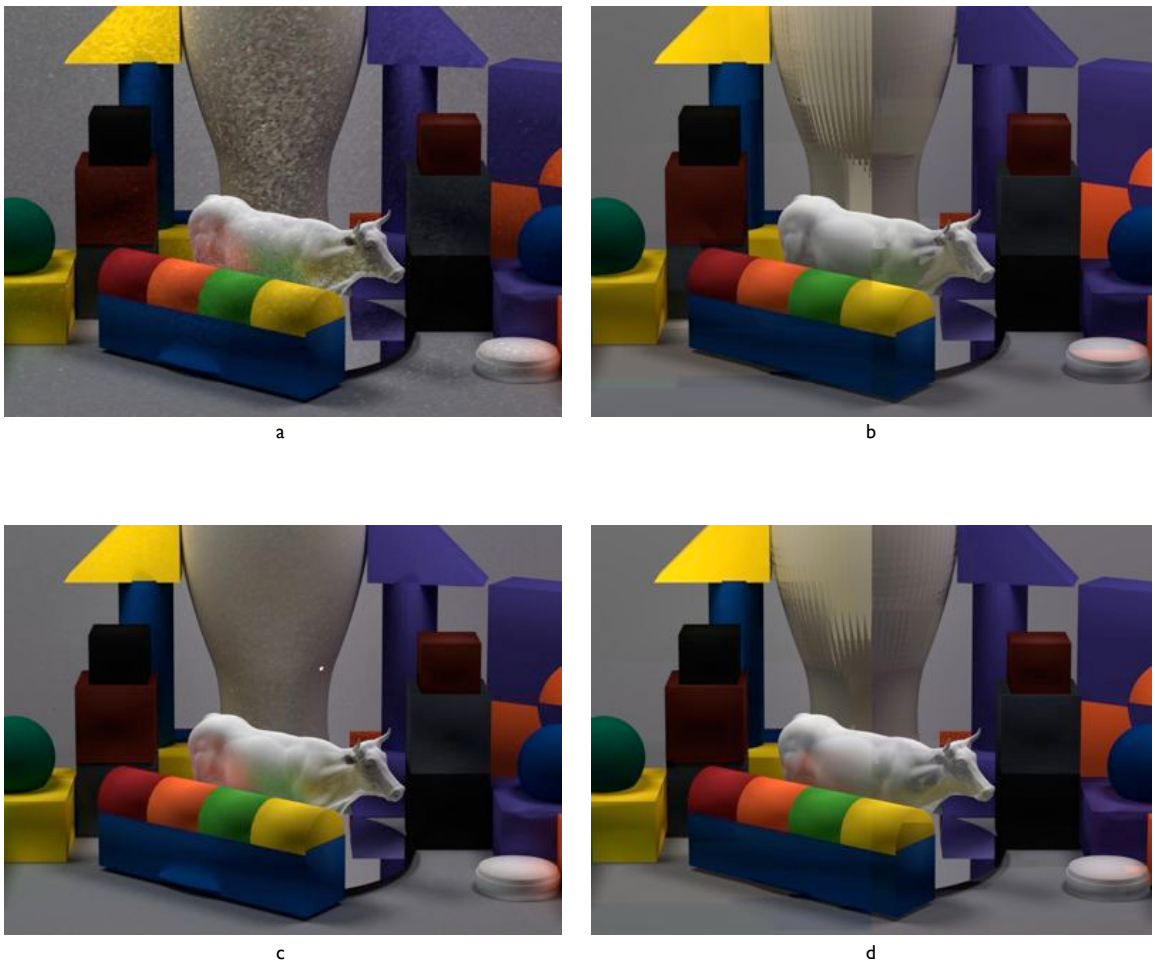


Figure 4.8: Irradiance caching integrator renderings. (a) and (b) were both rendered using 256 [spp] but with an error metric of 0.02 and 2.0 respectively. (c) and (d) were rendered using 4096 [spp] and the same 0.02 and 2.0 error metric respectively.

The effect of the error metric is readily apparent in this set of images. In Figure 4.8-b the error metric was set to 2.0, a relatively large value. One can almost see the area over which a given irradiance sample is used. In Figure 4.8-a, the noise is of a higher spatial frequency due to the increased number of irradiance samples cached using the error metric of 0.02. The other images use show how the image artifacts change when the number of samples is changed. As shown in previous examples, the increased number of samples reduces the noise and smoothes the image.

The final five images were rendered using the photon mapping surface integrator (Figure 4.9). Within `pbrt`, there are many parameters that can be varied for photon mapping, but only a couple were chosen, and within those parameters, only a few discrete points were used. The parameters varied were whether *final gathering* was used, the *maximum distance*, and whether or not the photon map was used to solve the indirect and direct lighting interactions, or only the indirect. `pbrt` defines the maximum distance as the "maximum distance between a point being shaded and a photon that can contribute to that point." The first two images show the photon mapping integrator being used to solve the entire radiative transport equation, with a maximum distance of 1.0 and 2.0 respectively. As shown, there is essentially no visual difference between the two images. The reason being, that the maximum distance of 1.0 was already too large, so doubling it to 2.0 has no effect on the quality. The poorly chosen maximum distance is what leads to the 'splotchy' appearance of the renderings when using photon mapping to solve the direct interactions. However, in Figure 4.9-b, the parameters are the same, with the exception of using final gathering. The final gathering option goes a long way to increasing the quality. The final two images used the photon mapping algorithm to solve the indirect portion of the integral only. The first image did not use final gathering. While it looks significantly better than Figures 4.9-a,b, there is still some "splotchiness", as well as other large errors such as the purple spot to the right of the vase and the dark area on the PTFE. The last image used photon mapping with final gathering to solve only the indirect interactions. In this image the final solution produces a very nice image, with no splotches, or large artifacts. The cow is very

smooth, and the indirect, colored interactions are displayed clearly on its belly. To this point, one may choose this as the best rendering. However, as will be shown later, that was not the case for the augmented reality images, due in part to the fact that the full-rendered images were not shown to observers, only the cow composited into a photograph.

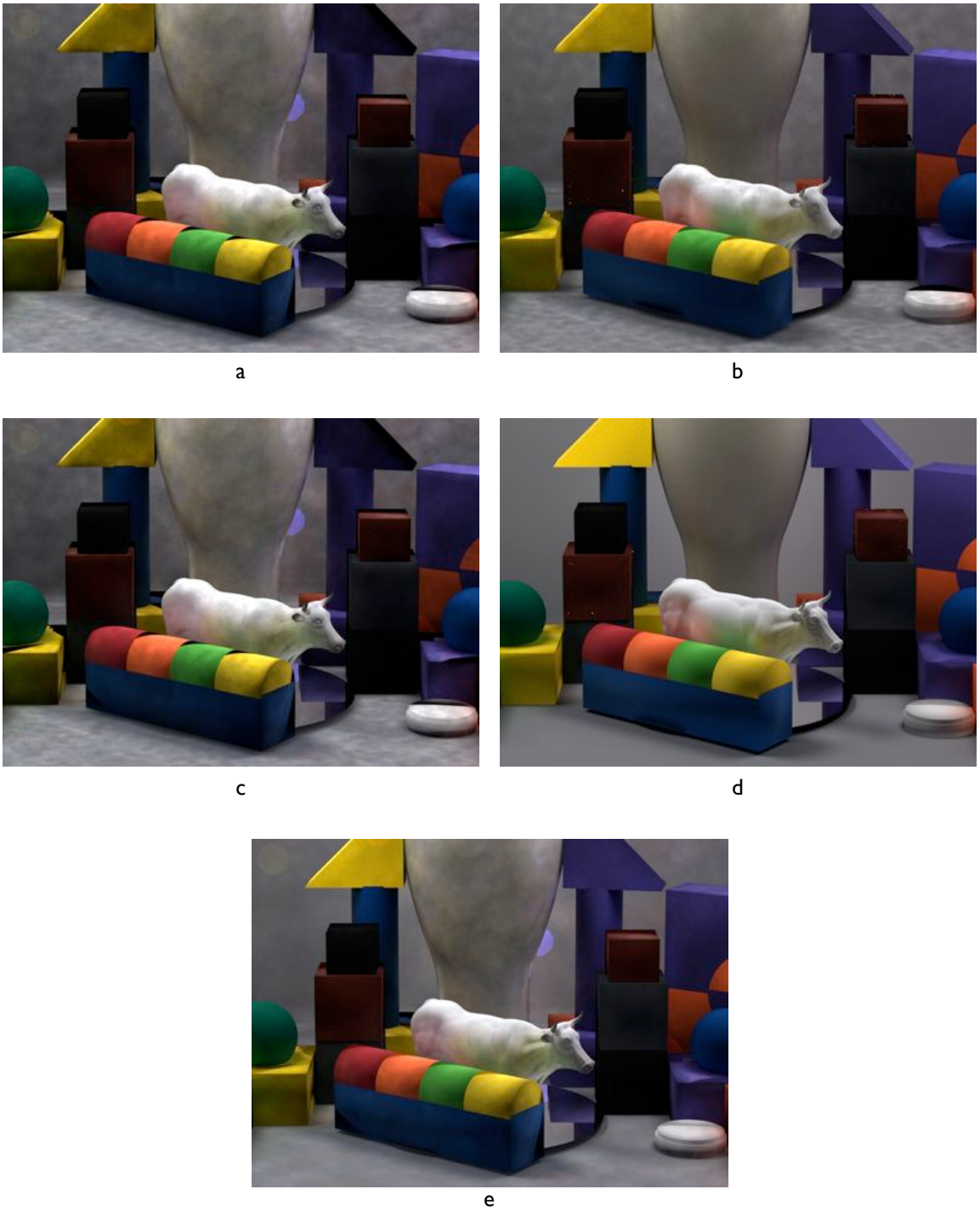


Figure 4.9: Photon mapping surface integrator renderings. (a), (b) and (c) all used the photon map to solve both the indirect and direct illumination. In addition, (c) used final gathering to reduce the visible artifacts. (d) and (e) used the photon map to solve the indirect component only with and without using final gathering respectively.

As stated earlier, the primary noise associated with path tracing was noise of variance. Photon mapping, however, exhibits bias. In other words, the final solution may have little noise, but an incorrect average pixel value, unlike path tracing, which predictably gets more accurate with more samples. So, image quality is not necessarily correlated to the correct pixel value. From the path tracing example shown above (Figure 4.7), the correct value for the average pixel value is approximately 70. Even with a large amount of noise, the average value is still very close to 70. With photon mapping however that value jumps from 65.2 to 87.2, while the standard deviation varies from 8.4 to 2.6. Additionally, it will be shown later that with photon mapping, these statistics do not correlate with the best perceived image.

4.2 Composite Renderings

This section discusses the final composite renderings that were presented to observers during the psychophysics experiment. The rendered cow was composited onto the photograph of the lightbooth scene and color calibrated for the Apple Cinema Display. A few things should be noted before presenting the results of the experiments. First, it should be apparent that the shadow of the cow onto the lightbooth floor is the same for all images. As explained in Section 3.4, it was decided early on to use the shadow of real cow from the photograph to avoid any additional error, primarily geometric distortion. The justification was that any artifacts visible in the shadows would also be visible in the cow, and thus including the rendered shadows should not change the results. Also, recall that the goal of this research was not to determine absolutes about specific rendering algorithms, but more importantly, the relationship between psychophysics and iCAM predictions.

The images are labeled below corresponding to the rendering algorithm used (see Figure 4.10). In general, the information presented in the previous section sufficiently describes the images and the artifacts present in the final composite images. However, there are a couple of things

to discuss. Prior to display on the 8-bit monitor, all of the images, including photographs and renderings, were stored in formats that supported at least 16-bits of information. This larger bit-depth sufficiently captured the dynamic range of the original scene in the lightbooth. In other words, detail was preserved in the shadows and specular highlights on the vase. In order to preserve as much of that information as possible on an 8-bit display, complex tonemapping algorithms would need to be used. However, since the images were only slightly higher dynamic range than could be displayed on an 8-bit monitor, no-such tonemapping algorithm was used. Rather, as the raw images were being processed through the LCD model, out-of-gamut colors were simply clipped. This had the biggest impact on the specular highlights on the top of the vase, as well as the top of the cow. Pixel saturation can be seen if one pays close attention. After all images had been processed, observers performed the experiment.

4.3 Psychophysics

Thirty one observers took part in this experiment comparing the composite images against each other and the real photograph. The observer pool consisted of 15 expert, including the author, and 16 naive observers. The typical time to complete the experiment ranged from approximately 10 to 20 minutes, although the exact time taken for an observer to complete the 136 comparisons was not recorded. The observers were allowed to take as much time as necessary to make a decision for a given trial pair. They were limited only by the brief time the noise images were shown between trials. The error bars used in this research were calculated using the method described in [40]. This method is empirical and takes into account not only the number of observations, but also the number of stimuli. In the case of this research, there were 31 observers and 136 pairs, both fairly large numbers, which should lead to smaller 95% confidence intervals (CI).

Figure 4.11 shows the combined results of naive and expert for the paired comparison experi-

Figure 4.10: The composite images, along with a map of the rendering algorithm used. These images are being shown in RGBs optimized for the Apple Cinema LCD that was used for the psychophysics experiment. Additionally, only the entire photograph is shown due to size constraints.

ment. This plot shows the interval scale value for each of the rendering algorithms used, as well as the photo. Large negative numbers represent the worst images based on the question asked, and increase with increasing quality of the image. There are a couple of important results that can be seen. First is the importance of using a full-global illumination algorithm. Direct illumination, and for the purposes of this scene, Whitted surface integrators, were chosen by the observers as less like the original. This result is not a surprise remembering the question for the observers was to choose the image most like the original, and not which they preferred. However, observers consistently chose the noisier results of algorithms such as path tracing, to the smooth, but dark cows created by the Direct and Whitted algorithms. The only exception to this is the extremely noisy result of the path tracing integrator using only 16 [spp] and the direct illumination integrator also using 16 [spp]. The path tracing image demonstrates significant amount of noise in capturing the indirect illumination effects, whereas the direct illumination produces a cow that is dark, but with very smooth tone transitions. As the plot shows, the observers did not think one image was closer to the photograph than the other within the 95% CI's.

The final important result shown in the graph is the image created using photon mapping to solve both direct and indirect illumination with final gathering was chosen (outside of the 95% CI's) by observers to be a better reproduction of the photograph, than the photograph itself! The composited image using photon mapping does indeed look a lot like the original, with the exception that it is a bit brighter. The author proposes the following explanation for this result. The observers were unaware of the fact that the image on the top was the original photograph, or that it never changed. Additionally, they were unaware that the original photograph, the same as the image at the top of the screen, was randomly being presented in the test pairs. Perhaps then the observers, particularly the naive, were assuming that every image presented to them was in some way manipulated from the original image. Therefore they switched their criteria from image accuracy to preference, thereby choosing the image with the brighter cow.

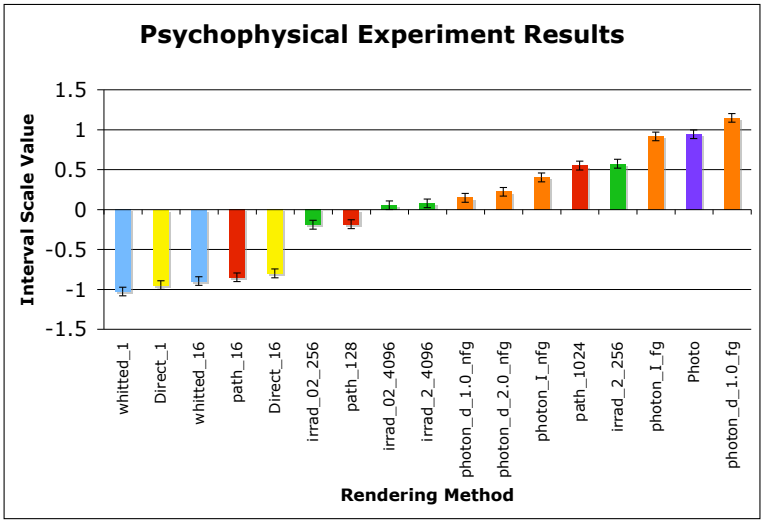


Figure 4.11: The interval scale plotted against the rendering algorithm for the entire pool of obsevers.

This is still valid, because in cases where they are very different or very similar, the observer is being asked to make a comparison where two images may be equidistant along two different axes, and therefore must choose one image or the other, resulting in a 50% likelihood of either.

In order to explore this further, the data were divided into naive and expert observers and analyzed again. The following plots in Figures 4.12-4.14, show the results when analyzed this way. The first plot shows the naive results plotted versus the interval scale, with recalculated error bars. There are a couple of interesting results. First, the photograph is ranked even lower on the scale, implying all rendering algorithms above it are at least the same visually as the photograph. Also, the naive observers tended rank all of the dark (*i.e.* no indirect illumination) as the worst in terms of accuracy to the original photograph with the exception of the noise path tracing rendering (path_16) which is equally bad as whitted_16 and direct_16.

The next plot (Figure 4.13), shows the results of the psychophysical experiment for the expert observers only. Interestingly, the experts disagree with the naive observers in that the path

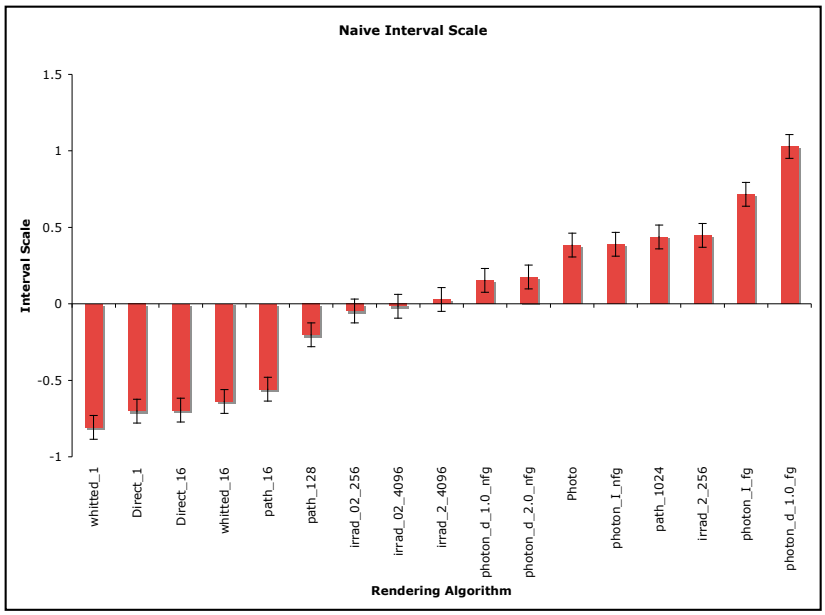


Figure 4.12: The interval scale plotted against the rendering algorithm for the naive obsevers.

tracing rendering with 16 [spp] is a lot worse than the direct with 16 [spp], however it is still within the error bars. Also, the experts ranked the irrad_02_ 256 (irradiance cache, 0.02 error metric, 256 samples per pixel), as being in the low grouping for accuracy. Both the path_16 (path tracing, 16 samples per pixel), and irrad_02_256 displayed a significant amount of high frequency noise in the cow. Also, the experts ranked the photograph at the top, which one would expect. However, it is still within the error-bars of the next best rendering using photon mapping and final gathering, the same image the naive group judged as most accurate. In the case of the experts, they chose the smoothest photon mapping image. The final plot in Figure 4.14 shows all of the psychophysical experiment results together. In nine out of seventeen stimuli, the expert and naive groups disagreed significantly. If nothing else it points out the need to look within the observer pool to understand the trends. When the two groups are examined separately, it is obvious that to the naive observers believe the photo is not the best statistically, and to the expert observers, it is. In either case, it is possible to create augmented reality images using

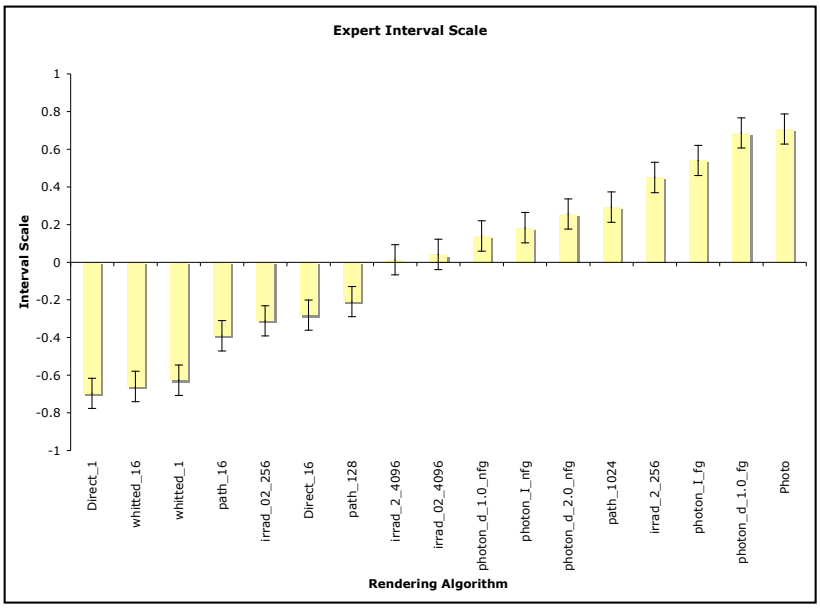


Figure 4.13: The interval scale plotted against the rendering algorithm for the expert obsevers.

a global illumination algorithm such as photon mapping, irradiance caching or path tracing, yielding a realistic result, indistinguishable from the original to human observers.

4.4 iCAM versus the Psychophysics

This section discusses the results using iCAM to compute an image difference. In general, one would expect that iCAM should predict a large image difference where the psychophysics scale value is small as well as the converse. Several different analyses were completed using iCAM. The first is a general plot (Figure 4.15), similar to the ones in the preceding section. However, keep in mind that low image difference values imply a closer match to the original photograph. Of course, in this computational example, the photograph will always receive an image difference of exactly 0. Additionally, as one might expect, all of the Whitted and direct integrator images have a significantly larger image difference than all of the other algorithms

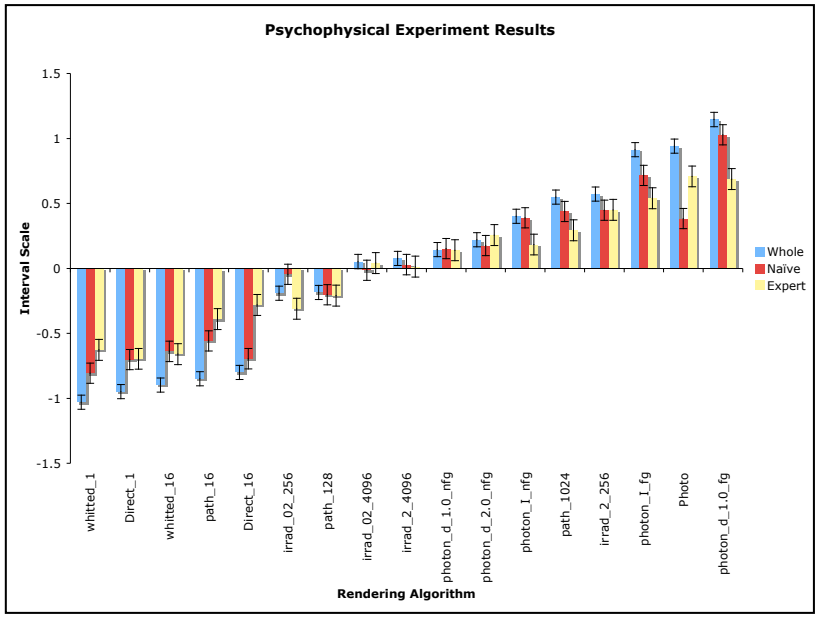


Figure 4.14: The interval scale plotted against the rendering algorithm for the entire pool of obsevers along with the naive and expert separately.

due to the fact that they do not incorporate indirect illumination effects. Other than that, the only thing that can be said about this plot, in general, is that it appears that the irradiance caching and photon mapping algorithms yield images with a larger difference than the path tracing. It seems then that iCAM calculates a smaller image difference for the noisy unbiased path tracing algorithms rather than the biased algorithms. The spatial noise is weighted less than the absolute color difference.

The next logical thing to do is to compare the iCAM results against the psychophysics results. Recall that iCAM inherently produces an image difference map as shown in Figure 3.16, yet in the plots a single number is used. In all cases it was determined that the 92^{nd} percentile of the image difference map was a reasonable method to reduce the data. It highlighted the area of the image that observers considered when making a decision. If one looks at those other percentiles, either too much or too little of the cow and surrounding area is included. Other

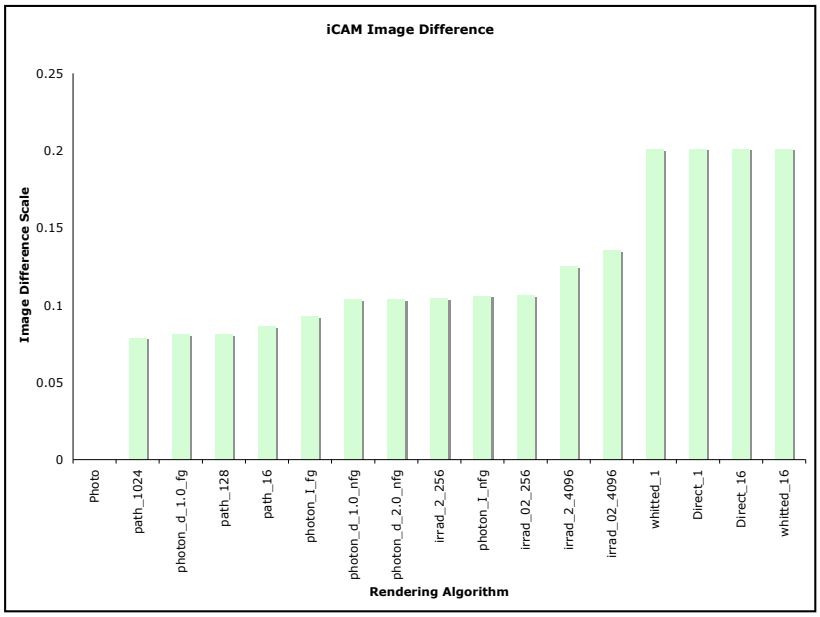


Figure 4.15: The 92% threshold of the iCAM image difference maps for each of the rendering algorithms is shown. Higher values indicate less accuracy to the original photograph.

statistics of the image difference images were calculated and included the mean and median, but were quickly discarded. The rationale was because statistics, such as the mean, would not correspond to real observers who performed the experiment, and imply a person is able to average all of the color differences and then make a judgement. This is typically not the behavior observed. Most people search for the first differences they see, which is more analogous to the percentile concept. Remember that the observers were asked to look at two images and choose the one that was closest in terms of accuracy to the photograph. Humans do not look at all of the differences for each image and then take an average. They start by looking at the most extreme differenes. As they need to examine the images more closely for differences, they are looking at the less extreme color differences, until one image looks worse than the other.

Figure 4.16 shows all of the image difference maps along with a scale for the magnitude of the pixel values for each rendering algorithm. As expected, large, uniform differences are shown for

the Direct and Whitted algorithms especially underneath the cow where no indirect illumination is calculated. Also interesting is the high frequency noise especially prevalent in the path_16 and irrad_02_256 images. The noise has been blurred slightly from the original images due to the iCAM spatial filtering. Figure 4.17 shows the image difference maps with the 92^{nd} threshold applied. Again one can see the pixels in white above that threshold. As stated earlier, this threshold does a good job of highlighting the differences that are most perceptible to a human observer.

The plot (Figure 4.19), shows iCAM versus the paired comparison interval scale for all observers combined. Recall that an inverse relationship (negative slope) is desired. In general, the plot shows this relationship, with an $r^2 = 0.55$ if all data points are included. It is apparent that there are three significant outliers, shown in red. These three correspond with irradiance caching (*error = 0.02, 256 [spp]*), path tracing (*16 [spp]*) and path tracing (*128 [spp]*). If these three images are removed from the data set, a much stronger relationship exists with a $r^2 = 0.9217$ as well as a higher slope. It is obvious that the common characteristic of these three outlier images is a significant amount of high frequency noise. Referring back to the plot, according to the psychophysics, these three images scored low on the interval scale. One would expect a large image difference value calculated by iCAM, which is not the case. Recall that iCAM computes an image difference map, which is then reduced to a single number using the 92^{nd} percentile statistic. This procedure of reducing the difference map to a single value does not explicitly include any spatial information such as high frequency noise. This is an extremely important as one of the major advantages of using iCAM versus a simple color difference equation is the incorporation of the spatial dimension.

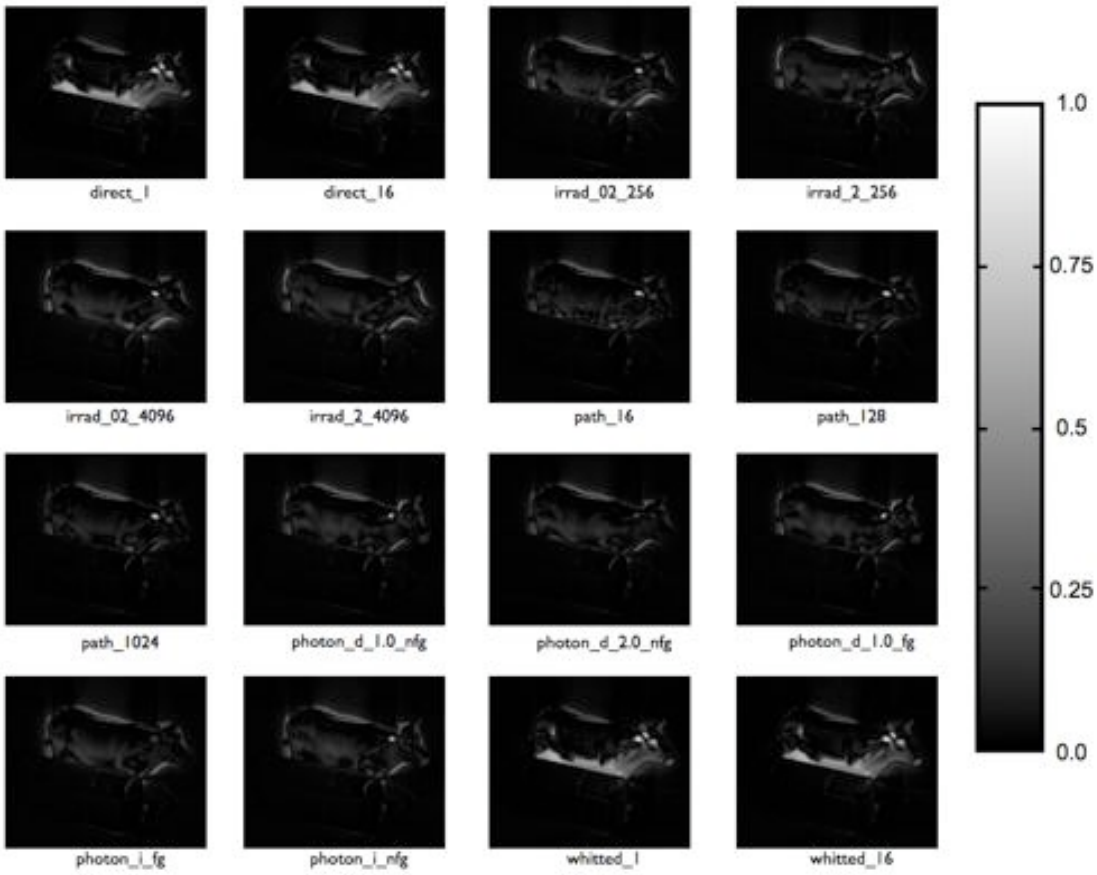


Figure 4.16: Image difference maps calculated using iCAM, between the all-real photograph, and each of the composite photographs.

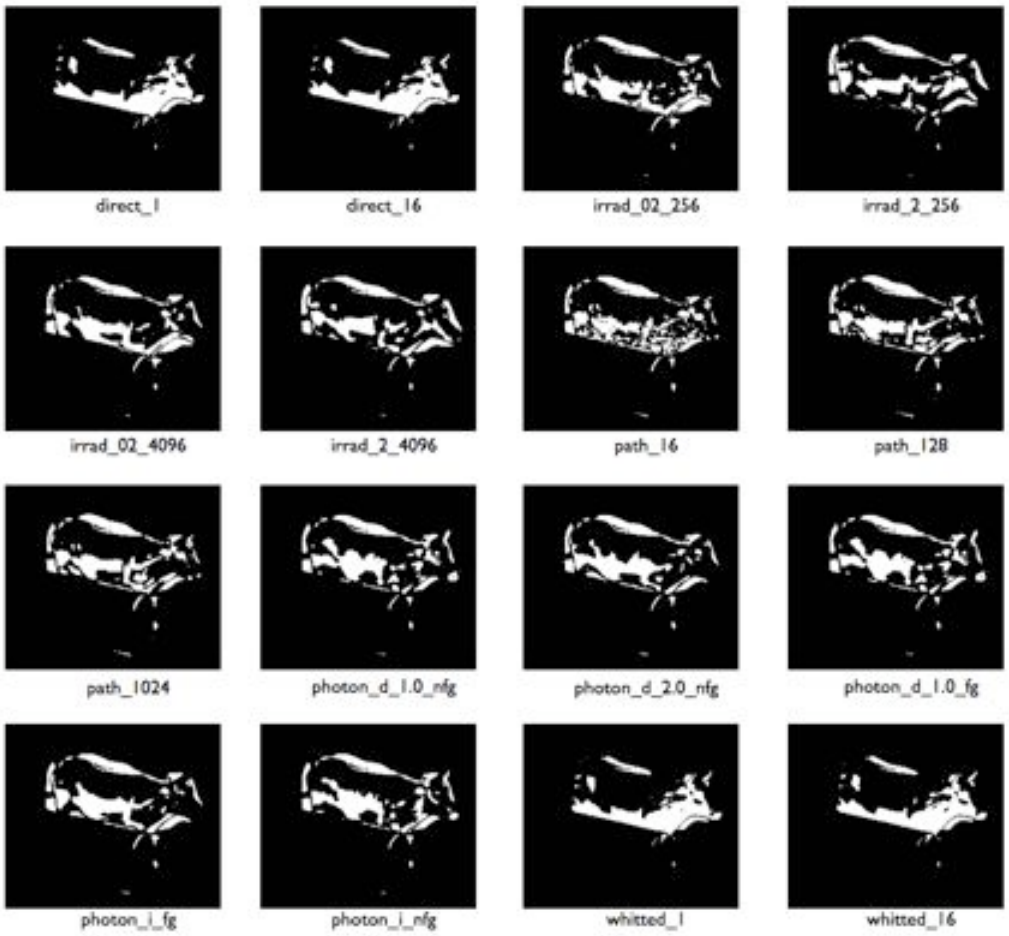


Figure 4.17: Image difference maps with the 92^{nd} percentile threshold applied. White pixels indicate image difference values above the threshold.

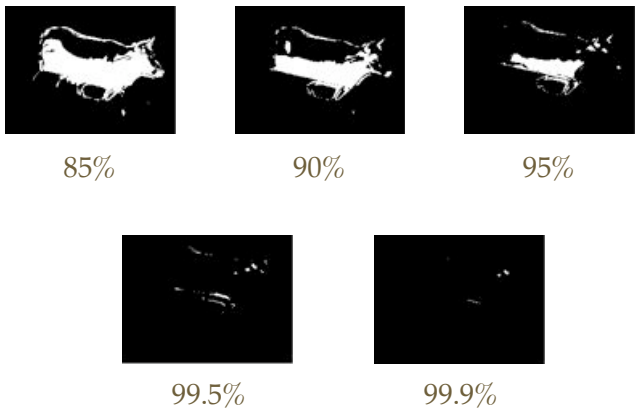


Figure 4.18: Image difference map where white are all of the image difference values above the annotated threshold level.

4.5 Rendering Time versus Accuracy

This section presents a plot showing the iCAM image difference scale (Figure 4.20) versus the rendering time in seconds. An analytical relationship is difficult to derive from this plot due to an insufficient number of data points within a given rendering algorithm. In general, the algorithms that took longer to compute yielded images that the observers judged as more like the original photograph. The only exceptions are the Direct and Whitted data points which took a longer time to render due to more samples per pixel. For approximately the same computation time, any number of algorithms can be chosen to give better results such as photon mapping using final gathering. Another way to look at it is for a given computation time, say approximately 3000 seconds, there is a large variance in the perceived accuracy within and between algorithms. This leads to the idea of specifying the maximum amount of compute time available, and adjusting the algorithm settings until an acceptable image is produced.

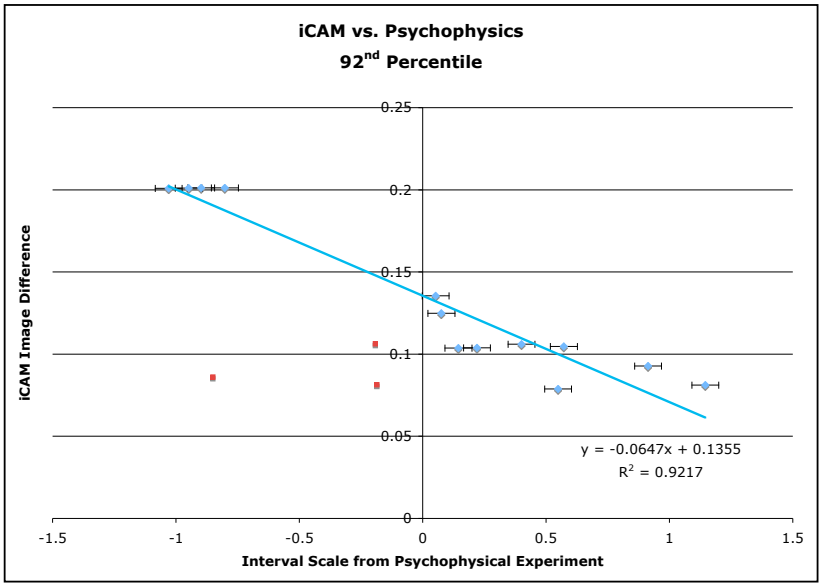


Figure 4.19: iCAM vs. the psychophysical experiment results, with the three outlier data points removed from the data fit.

4.6 Tolhurst's method

This section is presented purely out of interest. The author met researchers from the UK while at a conference, who were developing an algorithm similar to iCAM, but from a completely different starting point, primarily physiology and psychology. The details of the algorithm are given in [53]. Essentially "a low-level model calculates differences in local contrast between pairs of images within a few spatial frequency channels with bandwidth like neurons in V1 (primary visual cortex)" [43]. Their original research was looking in to how human discrimination changes as a function of the 'naturalness' of an image. Naturalness is defined in terms of the Fourier spectrum having a stable relationship between the frequency and the amplitude of that frequency. This idea shown as an equation is:

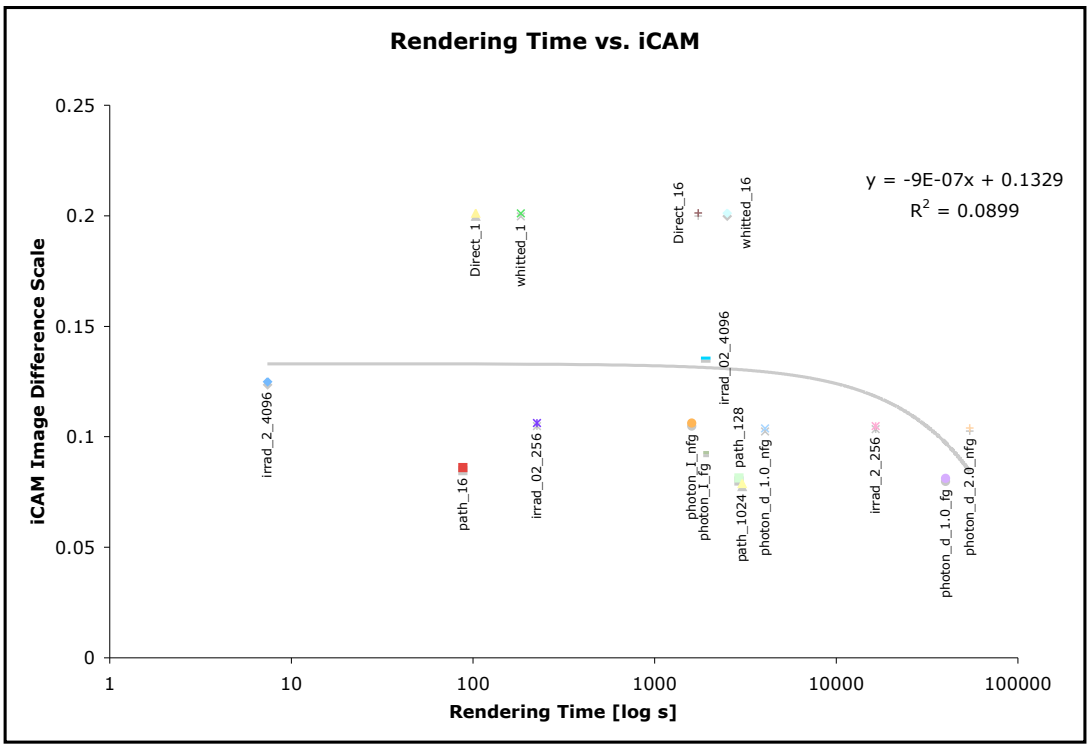


Figure 4.20: iCAM versus the rendering time.

$$Amplitude(f)\alpha\frac{1}{f^{\alpha}}. \tag{4.1}$$

From this research they developed a model of visual discrimination, and were interested in processing this data with their model. The author provided them with the cropped images in XYZ space, as predicted by the LCD forward model of the monitor RGB images (clipping maintained), the maximum luminance of the display, as well as the the maximum luminance of the Apple Cinema Display (100 $\frac{cd}{m^2}$). Results were emailed back, and are presented below. The first plot shows their scale, called Tolhurst's Values, plotted against the psychophysics interval scale. As can be seen, similar results are obtained using their discrimination model or

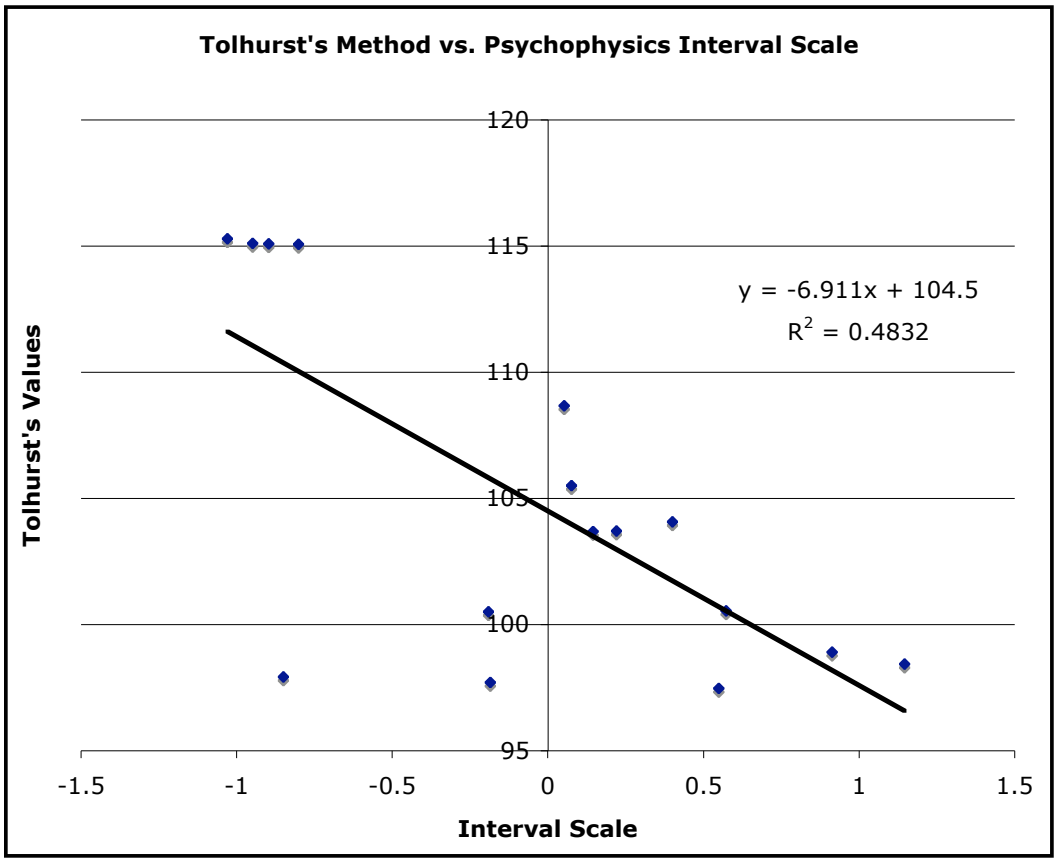


Figure 4.21: The psychophysical interval scale vs Tolhurst's method with the outliers included in the regression.

iCAM. This is encouraging in the fact that two different approaches based on human vision are yielding similar results. Additionally, it is also noticed that there are the identical outliers in a similar pattern. This could indicate that neither model is capable of capturing some perception phenomena, or perhaps the psychophysics yielded an error. More research will need to be completed to answer this question.

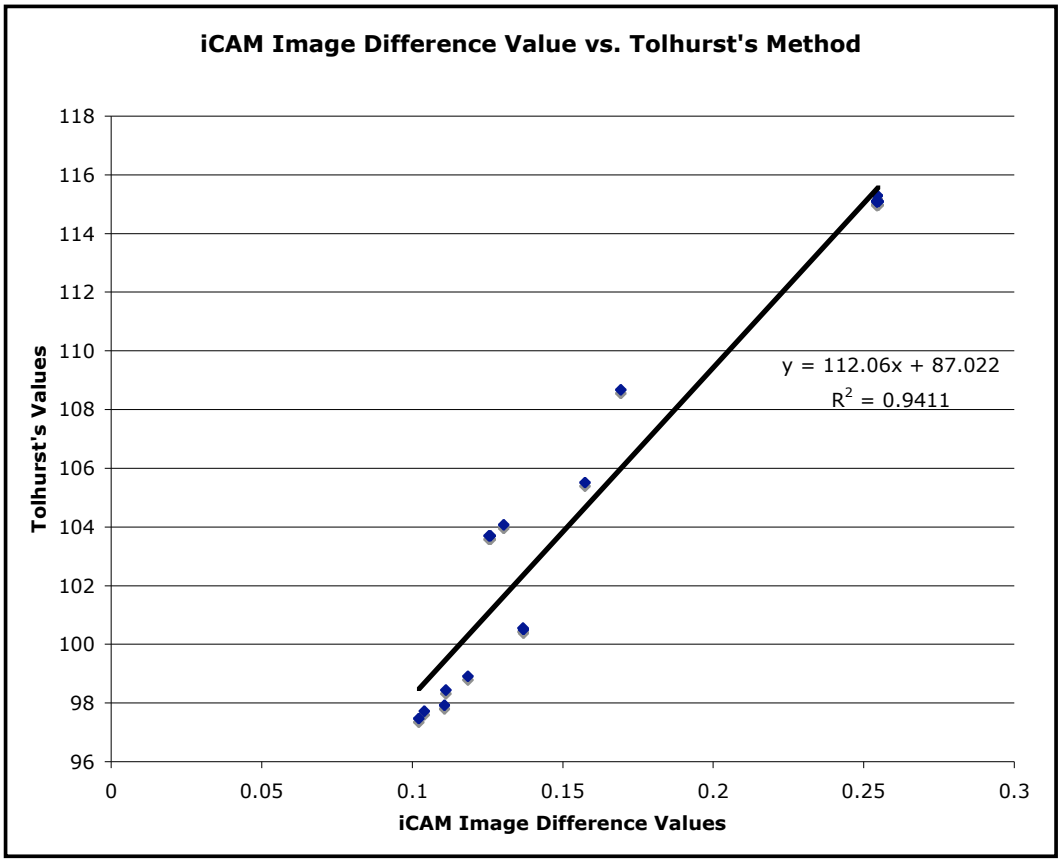


Figure 4.22: The iCAM image difference values vs Tolhurst's Method with outliers included in the regression.

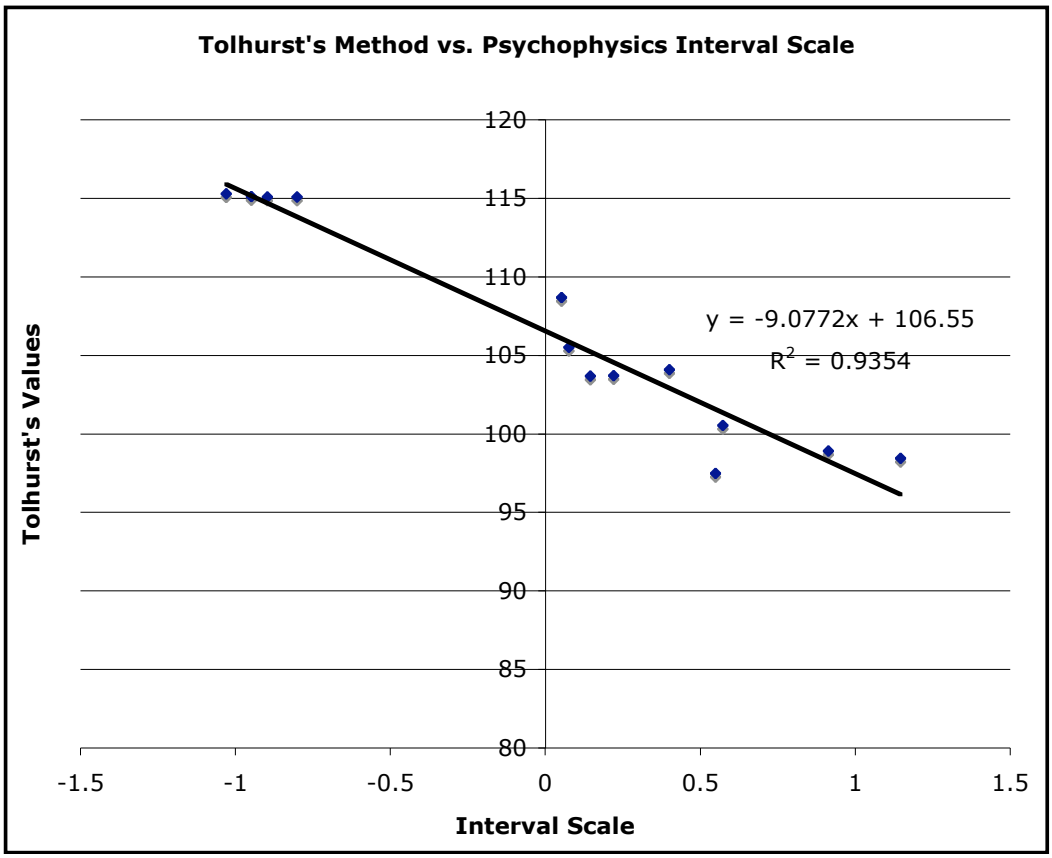


Figure 4.23: The psychophysical interval scale vs. Tolhurst's method with the same three outliers removed.

Chapter 5

Conclusions and Future Directions

The goal of this research was to begin to explore various global illumination rendering algorithms, specifically those based on ray-tracing, as applied to the rendering synthetic objects into real photographs. Furthermore, these algorithms in conjunction with augmented reality were analyzed through the use of psychophysical experiments and iCAM, a computational model of human vision. Through all of this, one could hopefully learn something about image synthesis, the human visual system and perception, and perhaps the confluence of the two.

In terms of image rendering, several things were learned, specifically when applied to rendering synthetic objects into real photographs. First, it seems that any global illumination algorithm will perform better than one that does not account for indirect illumination, except in the presence of significant noise of variance. This may not seem like a significant effect, until one considers the stimuli for the experiment. Consider the images rendered with the Direct illumination integrator as compared to the path integrator. Both the expert and naive observers ranked the noise path tracing image lower than the direct integrator with 16 [spp]. However, iCAM calculates the greatest image difference for all four algorithms that do not consider indirect illumination. There were trials where the observer was presented with two images that

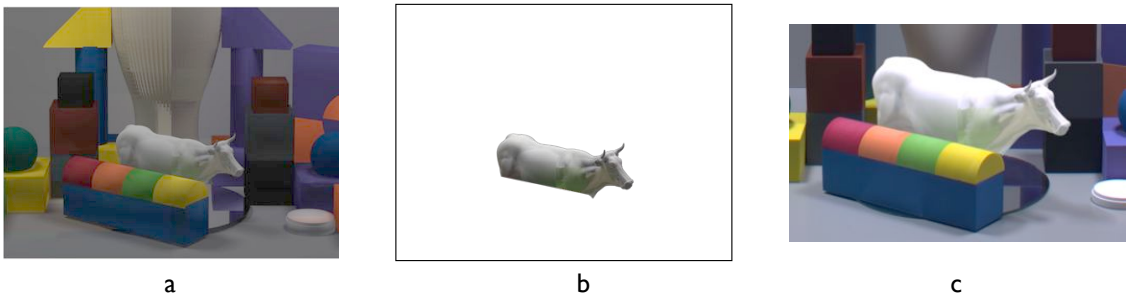


Figure 5.1: (a) The irrad_2_256 rendering, notice the artifacts on the vase and cow. (b) is the extracted cow, the artifacts are harder to see, unless one compares to the rendering. (c) The final composite with some clipping applied, making the artifacts harder yet to see.

may have been equally 'wrong,' leading them to judge a preference rather than accuracy.

Secondly, the experiments concluded that rendering time alone is not a direct indicator of the most accurate match to an original. It is generally true that more samples (*i.e.* more time) are required to achieve a better rendering especially for unbiased algorithms such as path tracing. However, when biased and unbiased algorithms are pooled, this is no longer the case. Stated differently, Figure 4.20 shows algorithms that take roughly the same time to render, but vary wildly both psychophysically and to iCAM. Related to this result is the fact that the most refined or 'tuned' rendering will always rank the best. Again this is true both for iCAM and the psychophysics. Clarifying, this result is most likely the case for the augmented reality application only. In other words, let us assume one looks at the entire rendering with all of the artifacts, and then extracts one object such as the cow and composites that into a photograph and compares the two images (see Figure 5.1). The cow does not necessarily appear as bad as the entire rendering because the artifacts are not as pronounced. This could be because of the material or lighting, or a number of other things. The converse is probably also true that some rendered objects show much more of the artifacts than the original rendering, thus making the final composite appear worse.

One of the most promising results was the correlation between iCAM and the psychophysical experiment. The expected relationship was present, with some outliers. The outliers all tended to be the renderings that had high frequency artifacts, where as the other images contained lower frequency artifacts. In other words, it seems as though the human observers judged these images using different criteria than the other 13 stimuli. Remember that there was strong agreement of these results with those from the researchers in the UK with those obtained from iCAM. Not only did they produce a similar relationship with iCAM, there were also the same artifacts present. Again this could be indicative of a deficiency in the models, or something with the manner in which the observers were making judgments during the experiment.

This last result leads to a discussion of future recommendations. First, it would be interesting to work further with these researchers in the UK and complete more psychophysical experiments to see under what conditions iCAM and their method agree and disagree. This would likely further the refinement of both. In doing so, the conclusions from this research could also be solidified by testing a variety of scenes. The scene used in this experiment appeared non-realistic in real life, which may have impacted the results.

In addition, it would be nice to enhance the pipeline for creating an augmented reality image. This includes more algorithms to calibrate the scene and camera. A more general compositing process could be implemented as described in the background and theory. More specifically, the effects of the object on the scene and scene on the object could be generically included. Also, more rendering algorithms could be used, including local illumination shaders that use a hack for the ambient term. It would be interesting to see where along the continuum these algorithms would fall. It is also possible to implement some of these algorithms and other shaders in GPU's (graphics processing unit), allowing even more possibilities of real-time rendering and interaction with the real scene. Ideally, it would be interesting to use this research to extract a baseline 'threshold for reality.' This could be used with iCAM in the rendering loop to produce

images that are believed to be within an acceptable accuracy to an original. Of course, all of these results would then be analyzed psychophysically. This is, of course, more tractable when the image rendering is on the order of minutes (GPU) rather than days with a ray-tracer.

Perhaps the most important recommendation is to continue research in reducing the data in image difference maps to a single number. The strength of iCAM is that it produces a *map*, of image differences. In other words, it calculates color differences spatially on complex spatial stimuli images. It seems counterintuitive to discard that spatial information in order to determine a relationship with a psychophysical experiment. This research clearly points out the need for more study into the reduction of the map into a single number. The author believes all of information is there at various stages of the iCAM image processing. Parameters could be derived at these various steps, and a multi-variable equation derived that reduces the difference map, including the color and the spatial characteristics. The outliers in the iCAM / psychophysical relationship may not be outliers at all, but just not completely described by the 92^{nd} percentile statistic.

In completely different direction, it would be very interesting to apply the compositing technique (in a more refined mode) to the remote sensing modality. For example, there are many existing image data sets flown over areas with known sensors. This information could be input into a spectral radiometric renderer such as `pbrt` or DIRSIG along with a local model of an area to be augmented and a composite image created. There are a lot of details to be wary of including sensors and absolute radiometry, but this could bridge the gap between doing large scale modeling and using existing data sets as they are.

Appendix A

Appendix

This appendix shows some of the ancillary files, scripts and data used in this research.

A.1 Spectral Reflectance of Materials

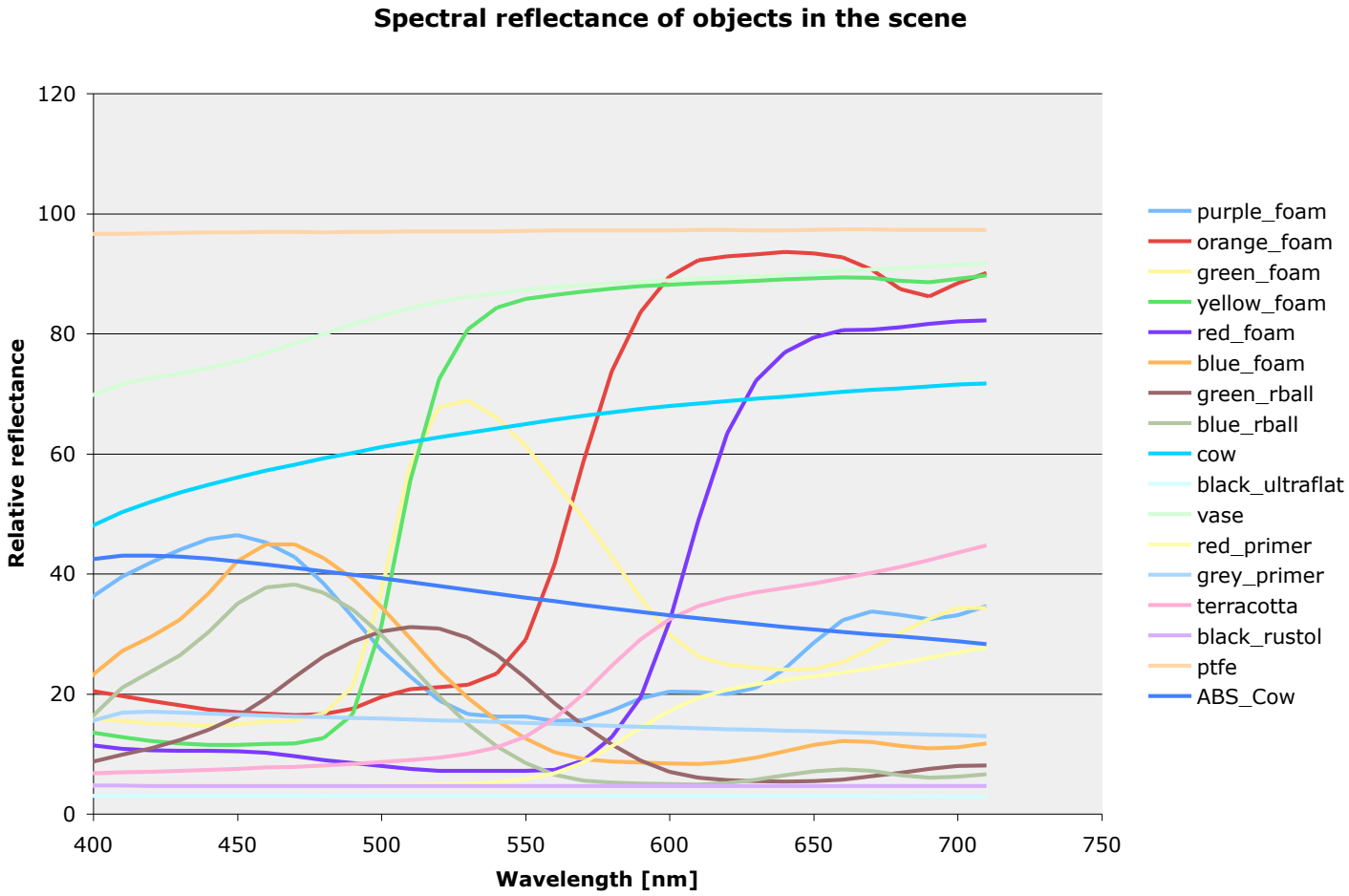


Figure A.1: Measured spectral reflectance of the objects used in the research.

A.2 Tcl Script to break and monitor rendering jobs

```
#!/usr/bin/tclsh
#create a master list of pbrts to run?



#Need to add the ability to create cow and no cow images
#Grab the pbrt filename from the command line arguments
proc split_jobs {fname nimages} {

    #set the number of images to break the pbrt cfg file into
    #set inc [ expr 1.0 / $nimages ]
    set inc 0.1869
    #open the pbrt file for reading
    set templateId [open $fname r]
    #decided to read the file using gets rather than read, which reads the file in
    #one line at a time and appends each line, as a string, to the list infile
    while {[eof $templateId] < 1} {
     lappend infile [gets $templateId]
    }
    close $templateId

    #search for cropwindow in each of the entries of the infile list
    #which returns the index of the occurrence
    set croplinenumber [lsearch -regexp $infile cropwindow]
    #hard coded to strip off [0 1 0 1] from the end of the cropwindow line
```

```
    set minuswindow [string range [ lindex $infile $croplinenumber  ] 0 23]

    #search for filename in each of the entries of the infile list
    #and strip off the original filename assumed to be $fname.exr"]
    set exrfilename [string trimright $fname .pbrt]
    set filename_linenumber [lsearch -regexp $infile filename]
    set filenameline [lindex $infile $filename_linenumber]
    set minusexrfilename [string trimright $filenameline "$exrfilename\.exr\"\]" ]
    #puts stdout $minusexrfilename

    #set boo [lindex $crap $croplinenumber]
    #puts stdout $boo

    #Maybe just loop through and make nimages lists and then write out the
    #nimages lists to the nimages filenames created below

    #Initializing the variables to create the files
#   set upinc 0
    set upinc 0.3178
    set newname [string trimright $fname .pbrt]
    set dirname [string toupper $newname]
    file mkdir $dirname
    cd $dirname

    #This loop is the meat of the program.  It first creates tempstr, which is
    #the new cropwindow values in the square brackets, [0 0.125 0 1] for example.
```

```
#The upinc is then  incremented by the inc (0.125 for 8 images), and then
#a new string is created called nline which is the concatenation of
#the line from above with the [0 1 0 1] removed, and the tempstr just created.
#A new list is created called outfile, which is essentially the same as
#the infile list read from the pbrt file, with the cropwindow line (a string variable)
#replaced with the new cropwindow line.  Then a file name is created of the form
# "oldfname_i.pbrt" inside of the $dirname and the file is opened for reading.
#The outfile, a list variable, is written line by line to that filename.
#I also execute pbrt from within this script, and catch any errors


for {set i 0} {$i < $nimages} {incr i} {
    set poo "$newname\_[expr $i + 1]"
    set tempstr "\[$upinc  [expr $upinc + $inc] 0.4693 0.8883\]"
    set upinc [expr $upinc + $inc]
    set nline "$minuswindow $tempstr"
    set exrline "$minusexrfilename$poo\.exr\"\] "
    set outfile [ lreplace $infile $croplinenumber $croplinenumber $nline]
    set outfile [ lreplace $outfile $filename_linenumber $filename_linenumber $exrline]

    set tempId [open "$poo\.pbrt" w]
    for {set j 0} {$j < [ expr [ llength $outfile] - 1] } {incr j} {
 puts $tempId [ lindex $outfile $j ]
    }
    close $tempId
```

```
        #puts stdout [pwd]
        set command1 "$poo\.pbrt"
        set command2 "$poo\.log"
        #puts stdout " $command1 $command2"



        if [ catch { exec nohup pbrt $command1 >>& $command2 &} result]  {
            global errorInfo
            puts stderr $result
            puts stderr "***Tcl TRACE****"
            puts stderr $errorInfo
        } else {
            #command body ok, result of last command is in result
        }

        unset poo
    }



}

#This loop is the main program that loops through the major pbrt files
#and then calls the split_jobs procedure described above.  It then invokes the
#after command, and sets x to the list of exr files in the directory where pbrt
```

```
#is rendering. Once the number of exr images is nimages, that image is completely
#rendered and the program can then move on to the next pbrt file and repeat the process
set pbrtfiles [ glob  *.pbrt]
set nimages 2
#set waittime 300000
set waittime 60000
for {set i 0} {$i < [llength $pbrtfiles]} {incr i} {
    split_jobs [lindex $pbrtfiles $i] $nimages
    set dname [string toupper [string trimright [lindex $pbrtfiles $i] .pbrt] ]
    set x 0
    #set x [glob -nocomplain $dname\/*exr]
    while {$x < $nimages} {
     #   puts stdout [pwd]
        after $waittime {set x [glob -nocomplain *exr] }
        vwait x
     #   puts stdout [ llength $x ]
    }
    cd ..
    puts stdout [pwd]
}
```

Bibliography

[1] blender3d.org :: Home.

[2] Stratasys, inc - rapid prototyping, cad plastic prototyping, digital prototypes, cad plastic prototype engineering.

[3] Recommendation itu-r bt.709.5: Parameter values for the hdtv standards for production and international programme exchange. Electronic, 04 2002.

[4] Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, August 1997.

[5] Roy S. Berns. *Billmeyer and Saltzman's Principles of Color Technology*. John Wiley and Sons, New York, 3rd edition edition, 2000.

[6] Mark R. Bolin and Gary W. Meyer. A frequency based ray tracer. In *SIGGRAPH*, 1995.

[7] Ron Brinkmann. *The Art and Science of Digital Compositing*. Morgan Kaufman, 1999.

[8] Scott Brown. The DIRSIG homepage, 2003.

[9] Florin Cutzu, Riad Hammoud, and Alex Leykin. Estimating the photorealism of images: Distinguishing paintings from photographs. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[10] Ellen A. Day, Lawrence Taplin, and Roy S. Berns. Colorimetric characterization of a computer-controlled liquid crystal display. *Color Research and Application*, 29(5):365 – 373, October 2004.

[11] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *SIGGRAPH*, 1998.

[12] Paul E. Debevec and Jittendra Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH*, 1997.

[13] George Drettakis, Luc Robert, and Sylvain Bougnoux. Interactive common illumination for computer augmented reality. In *Proceedings of the 8th Eurographics on Rendering*, 1997.

[14] Reynald Dumant, Fabio Pellacinin, and James A. Ferwerda. Perceptually-driven decision theory for interactive realistic rendering. *ACM Transactions on Graphics*, 22(2):152–181, 2003.

[15] Fritz Ebner and Mark D. Fairchild. Development and testing of a colro sapce (ipt) with improved hue uniformity. In *The Sixt Color Imaging Conference: Color Science, Systems, and Applications*. Imaging Science and Technology, 1998.

[16] Mark D. Fairchild. *Color Appearance Models*. Wiley IST Series in Imaging Science and Technology, 2005.

[17] Mark D. Fairchild and Garrett M. Johnson. Image appearance modeling. In *SPIE/IS&T Electronic Imaging Conference*, pages 149–160, 2003.

[18] J.A. Ferwerda, S. Pattanaik, P. Shirley, and D.P. Greenberg. A model of visual adaptation for realistic image synthesis. In *SIGGRAPH 1996*, pages 249–258, 1996.

[19] James A. Ferwerda. A model of visual masking for computer graphics. In *SIGGRAPH*, 1997.

[20] James A. Ferwerda. Three varieties of realism in computer graphics. In *Proceedings SPIE Human Vision and Electonic Imaging*, 2003.

[21] James A. Ferwerda, Holly Rushmeir, and Benjamin Watson. Frontiers in perceptually-based image synthesis: Modeling, rendering, display, validation. 2003.

[22] Alain Fournier. Illumination problems in computer augmented reality. *Journal of Analysis and Synthesis*, 1994.

[23] Simon Gibson, Toby Howard, and Roger Hubbold. Flexible image-based photometric reconstruction using virtual light sources. In *EUROGRAPHICS*, 2001.

[24] Andrew S. Glassner. *Principles of Digital Image Synthesis*, volume Two. Morgan Kaufman, 340 Pine Street, Sixth Floor, San Francisco, CA 94104, USA, 1995.

[25] Andrew S. Glassner. *Principles of Digital Image Synthesis*, volume One. Morgan Kaufman, 340 Pine Street, Sixth Floor, San Francisco, CA 94104, USA, 1995.

[26] Roy Hall. *Illumination and Color in Computer Generated Imagery*. Springer-Verlag, New York, 1989.

[27] Eugene Hecht. *Optics*. Addison-Wesley, Reading, Mass, second edition, 1987.

[28] Geoffrey S. Hubona and Matthew Brandt. The relative contributions of stereo, lighting, and background scenes in promoting 3d depth visualization. *ACM Transactions on Computer-Human Interaction*, 2000.

[29] Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping*. AK Peters, 2001.

[30] Garrett M. Johnson. *Measuring Images: Differences, Quality and Appearance*. PhD thesis, Rochester Institute of Technology, 54 Lomb Memorial Drive, 2003.

[31] Garrett M. Johnson and Mark D. Fairchild. Full-spectral color calculations in realistic image synthesis. *IEEE Computer Graphics and Applications*, 19(4):47–53, 1999.

[32] Garrett M. Johnson and Mark D. Fairchild. *Digital Color Imaging Handbook*, volume One, chapter Two. CRC Press, 2003.

[33] James T. Kajiya. The rendering equation. In *SIGGRAPH 1985*, pages 143–150, 1985.

[34] Jed Lengyel. The convergence of graphics and vision. *IEEE Computer*, 31(7):46–53, July 1998.

[35] Celine Loscos, George Drettakis, and Luc Robert. Interactive virtual relighting of real scenes. *IEEE Transactions on Visualization and Computer Graphics*, 6(4):289–305, October-December 2000.

[36] Ann McNamara. Visual perception in realistic image synthesis. In *EUROGRAPHICS*, 2001.

[37] Ann McNamara and Alan Chalmers. Comparing real and synthetic scenes using human judgements of lightness. In *Proceedings of the Eurographics Workshop on Rendering*, 2000.

[38] Ann McNamara, Alan Chalmers, Tom Troscianko, and Erik Reinhard. Fidelity of graphics reconstructions: A psychophysical investigation. In *9th Eurographics Rendering Workshop*, 1998.

[39] Gary W. Meyer, Holly E. Rushmeier, Michael F. Cohen, Donald P. Greenberg, and Kenneth E. Torrance. An experimental evaluation of computer graphics imagery. *ACM Transactions on Graphics*, 5(1):30–50, January 1986.

[40] Ethan Montag. Empirical formula for creating error bars for the method of paired comparison. *Journal of Electronic Imaging*.

[41] K.T. Mullen. The contrast sensitivity of human color vision to red-green and blue-yellow chromatic gratings. *Journal of Physiology*, 359:381–400, 1985.

[42] C.A. Parraga, T. Troscianko, and D.J. Tolhurst. The human visual system is optimized for processing the spatial information in natural visual images. *Current Biology*, 10:35–38, 2000.

[43] C.A. Parraga, T. Troscianko, and D.J. Tolhurst. The effects of amplitude-spectrum statistics on foveal and peripheral discrimination of changes in natural images, and a multiresolution model. *Vision Research*, 45:3145–3168, 2005.

[44] S. Pattanaik, J.A. Ferwerda, M.D. Fairchild, and D.P. Greenberg. A multiscale model of adaptation and spatial vision for realistic image display. In *SIGGRAPH*, pages 289–298, 1998.

[45] Matt Pharr and Greg Humphreys. *Physically Based Rendering From Theory To Implementations*. Elsevier, 2004.

[46] Paul Rademacher, Jed Lengyel, Edward Cutrell, and Turner Whitted. Measuring the perception of visual realism in images. In *Proceedings of the 12th Eurographics Workshop on Rendering*, 2001.

[47] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul E. Debevec. *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann, 2006.

[48] James C. Rodger and Roger A. Browse. Choosing rendering parameters for effective communication of 3d shape. *IEEE Computer Graphics and Applications*, March/April 2000.

[49] Imari Sato, Yoichi Sata, and Katsushi Ikeuchi. Acquiring a radiance distribution to superimpose virtual objects onto a real scene. *IEEE Transactions on Visualization and Computer Graphics*, 5(1):1–12, January-March 1999.

[50] Yoichi Sato, Mark D. Wheeler, and Katsushi Ikeuchi. Object shape and reflectance modeling from observation. In *SIGGRAPH*, 1997.

[51] John R. Schott. *Remote Sensing The Image Chain Approach*. Oxford University Press, 1997.

[52] Jeremy Adam Selan. Merging live video with synthetic imagery. Master's thesis, Cornell University, 2003.

[53] David J. Tolhurst, C Alejandro Parraga, P George Lovell, and Tom Troscianko. A multiresolution color model for visual difference prediction. In *ACM SIGGRAPH Symposium on Applied Graphics and Visualization*, pages 135–138, 2005.

[54] Brian A. Wandell. *Foundations of Vision*. Sinauer Associates, Inc., 1995.

[55] Leonard R. Wanger, James A. Ferwerda, and Donald P. Greenberg. Perceiving spatial relationships in computer-generated images. *IEEE Comuter Graphics and Applications*, 1992.

[56] Alan Watt. *3D Computer Graphics*. Addison-Wesley, 2000.

[57] Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkings. Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *SIGGRAPH*, 1999.