Comparing Integrators on Perceived Realism and Quality

Rob Post

Sunday, July 08, 2012

Comparing Integrators on Perceived Realism and Image Quality

R.J.M. Post
1358383

Master Thesis
Media and Knowledge Engineering
Computer Science
July 2012

Computer Science – Media and Knowlegde Engineering
Interactive Intelligence Group
Faculty EEMCS, Delft University of Technology
Delft, The Netherlands

Visual Experiences Group
Philips Research
High Tech Campus
Eindhoven, The Netherlands

Graduation Committee:

Prof. Dr. I. Heynderickx	Philips Research / Delft University of Technology
M. Murdoch	Philips Research
Dr. Ir. G. de Haan	Delft University of Technology
Dr. J. A. Redi	Delft University of Technology

Preface

In front of you lies my master thesis in computer science. For this master thesis I did research in the area of rendering at the Philips High Tech Campus in Eindhoven in the Visual Experiences group. I have been working on finding a relation between the quality of renderings and the effort that is needed to create the renderings. The knowledge gained from this work can be used in the future to create renderings that look more realistic.

During my research study I learned how to conduct and report research projects and I gained experience in the field of 3D modeling, rendering and image quality.

My stay at the Visual Experiences group was most pleasant and educational with a good atmosphere. I would especially thank my mentors Michael Murdoch and Ingrid Heynderickx for their excellent input and support during my research. I also would like to thank everybody else at Philips Research who helped me out with my experiments.

Rob Post
Eindhoven, July 8, 2012

Summary

Nowadays almost everything is digital. Even testing new products can be done by simulating them digitally, also called virtual prototyping. This report describes research that is done to improve virtual prototyping at Philips Research. Two experiments have been performed with the aim to evaluate how image quality and realism are affected by the integrator, i.e. the method used to calculate the light at each point in the scene, and the time spent on the rendering process.

The first experiment was conducted to evaluate whether a different integrator than an implementation of a standard path tracer might result in more realistic renderings. Mitsuba was used as render software and support integrators both based on path tracing and photon mapping. Two images of the same scene were simultaneously shown to 20 participants separately. The participants had to indicate which of the two they considered most realistic. They had to do this for different lighting conditions and with and without a reference to the real scene. The results showed that people preferred a photon based integrator over the path based integrators.

The second experiment focused more on how material characteristics influence the rendering time. Renderings of a teapot with different materials and under different lighting conditions were created with Mitsuba and Indigo software. Participants had to compare a rendering to a reference rendering, which was part of a standard quality ruler, based on one type of material and light condition. As a consequence, they had to compare quality across different materials and lighting conditions. The results were widely spread across the 20 participants, mainly because comparing quality of different materials was a difficult task. The task became even more complicated because of the different kinds of noise generated by the different integrators.

To overcome the issue found in the second experiment, a follow-up study was conducted. In this follow-up study the participants not only saw a stimulus and a ruler rendering but also the perfect versions of those two. The task was now to identify if the stimulus or the ruler rendering had the smallest difference with their perfect version. The results for the 4 participants that performed this follow-up study were less spread than those of the second experiment. Therefore the interface used in the follow-up study is considered to be useful for future experiments where comparisons across different materials are needed.

Table of Contents

1 Introduction

Sometimes it is desirable to display images of a scene that doesn't exist in real life. In that case a virtual model of the scene can be made. Creating images of a virtual scene is called rendering. Different methods of rendering exist, of which the most popular ones are described in the Literature Study section of this report. Images created using rendering methods are called renderings. A lot of rendering is used in the field of Virtual Prototyping.

Virtual Prototyping (VP) is very useful during the development process of a product. Many ideas for a product may arise, but usually only a few of these ideas will result in an actual prototype that is built. Many of the other ideas are rejected. It would be too expensive or too time consuming to build prototypes of all the ideas. A method to overcome this limitation is VP.

There exist many definitions of VP in literature (Wang, 2002). The definition of VP according to (Wang, 2002) is:

"Virtual prototype, or digital mock-up, is a computer simulation of a physical product that can be presented, analyzed, and tested from concerned product life-cycle aspects such as design/engineering, manufacturing, service, and recycling as if on a real physical model. The construction and testing of a virtual prototype is called virtual prototyping."

So with VP it is relatively easy to create prototypes and evaluate them with a test panel. The main aspect in VP is that the virtual prototype should accurately mimic a real prototype in such a way that it behaves as a psychical prototype would do in the real world.

VP can be used in almost all application fields. One of the application fields important for the research of this report is lighting design, and more particularly, testing certain luminaires in a scene. With the help of light simulation software, images of how specific luminaires illuminate a given scene can be constructed, and the effect of placing the luminaires differently or using different luminaires can be investigated. This is exactly what is done in this research.

The main aim of VP to the purpose of lighting design is to achieve that people perceive light in renderings the same as they would do in real life. One way to accomplish this is to simulate light physically correct. When using a physically based render technique (PBRT) (Pharr & Humphreys, 2010) all materials and light sources must be modeled physically correct, such that the integrator is able to calculate the physical behavior of the light in the virtual scene. An alternative to PBRT is photo-realistic rendering (Agusanto, Li, Chuangui, & Sing, 2003), where the emphasis lays on the output rather than the techniques used to derive it. The only important aspect of photo-realistic rendering is that the final result looks nice. The first aim of the current research is to evaluate the difference in perceived realism between different PBRT rendering methods.

Many experiments on how to render the most realistic images have been reported in literature. (Salters, Murdoch, Sekulovksi, Chen, & Seuntiens, 2012) and (Salters & Seuntiens, A Comparison of Perceived Lighting Characteristics in Simulations Versus Real-Life Setup, 2011) In these studies one particular light simulating method for rendering that produces quite realistic images has been used a lot. This method is compared to other existing methods in the research reported here. In this comparison the focus lies in the relation between effort that is needed to create the image and quality or realism of the image. There might be light simulating methods that simulate light as good as the currently used method, but require less time to complete the rendering. Also, the quality of a different method may be much higher when run for the same amount of time. Hence, the second aim of the current research is to investigate the balance between quality of renderings and time needed to accomplish them.

A comparison between different light simulation methods is done by means of two different experiments. Before it is possible to describe the experiments done in this research, some terminology and techniques must be explained. This is done in chapter 2, giving an introduction of the literature about rendering, realism and how to judge the quality of images. Chapter 3 describes software that can be used to render actual images. This software is used to create the renderings for the experiments that are described in chapter 4 and in chapter 5. Finally, the main conclusions are described in chapter 6 and suggestions for future work are given in chapter 7.

2 Literature Study

The chapter gives an introduction to the research area of rendering and some aspects on how to judge the quality of renderings compared to real world scenes. The information in this chapter is needed to fully understand the experiments that have been done in this research. Much more is already known and studied on rendering and that information can be retrieved from the references in this report.

This chapter starts by introducing the rendering equation and then describes some integrators which try to solve this rendering equation. Once an image is rendered, it must be tone mapped before it can be shown on a display. Tone mapping is described in section 2.8. After the tone mapping step the images can be rated on their realism. Different types of realism are described in section 2.9 and possible methodologies on how to rate the quality of images are described in section 2.10.

2.1 Light Simulation

Seeing objects is all about receiving light in the eye. Therefore to make renderings of a 3D scene, light must be simulated. Most light simulation techniques attempt to do this by solving the rendering equation introduced by (Kajiya, 1986). The rendering equation is based on the law of conservation of energy and reads:

$$I(x, x') = g(x, x')\left[\epsilon(x, x') + \int_S \rho(x, x', x'')I(x', x'')dx''\right]$$

Where:

$I(x, x')$	is related to the intensity of light passing from point x' to point x.
$g(x, x')$	is a "geometry" term.
$\epsilon(x, x')$	is related to the intensity of emitted light from x' to x.
$\rho(x, x', x'')$	is related to the intensity of light scattered from x'' to x by a patch of surface at x'.
S	is the union over all surfaces.

It basically says that at each particular position and direction on a surface, the outgoing light is the sum of the emitted light and the reflected light. The reflected light itself is the sum of the incoming light from all directions, multiplied by the surface reflection and cosine of the incident angle. Light paths must be calculated in order to solve the rendering equation. Light paths are calculated by integrators.

Simulating light in a scene is usually referred to as calculating global illumination. Global illumination takes into account direct and indirect illumination. Direct illumination considers only the light that comes directly from a light source and hits a surface. Indirect illumination takes also reflection and refraction of light on surfaces into account. Direct illumination is much easier to calculate than indirect illumination, but using only direct illumination will of course not result in a physically correct rendering. Direct illumination can be used to clearly see the light distribution of a luminaire without being distracted by the indirect illumination. It can also be used for making a quick rendering of a scene to see how it looks like. Both direct and indirect illumination must be used to render physically correct images.

It should also be considered that the medium light is traveling through can influence the light paths. These so called participating media, such as dust, smoke, fog or even air, can have a huge impact on the rendering process. In very large scenes light gets scattered by the air it travels through, and as a consequence, more and more light paths must be calculated in order to obtain an accurate rendering. Due to the higher computational load the rendering process will take much longer. Participating media are usually omitted from the scene unless they should be visible in the rendering. Most rendering software requires a specific integrator to deal with participating media.

2.2 Projections

Before light paths can be calculated, a scene and a viewpoint must be defined. A scene describes a 3D environment with objects in it. It describes where the objects are, how they are orientated and what material they consist of. The viewpoint, from now on referred to as virtual camera, defines how the user looks into this 3D environment. The camera has simple attributes such as position and orientation, but it can also have more advanced attributes such as focus distance and aperture settings. Two important settings of the camera are the positions and sizes of the near and far clipping planes. These planes define a volume in which the camera can actually 'see' the scene.

In order to make an image of a 3D scene, the 3D scene must be projected on a 2D plane. Two types of projections are usually used in computer graphics: perspective projection and orthographic projection (Shreiner, 2009). A schematic overview of both projections can be found in Figure 1.

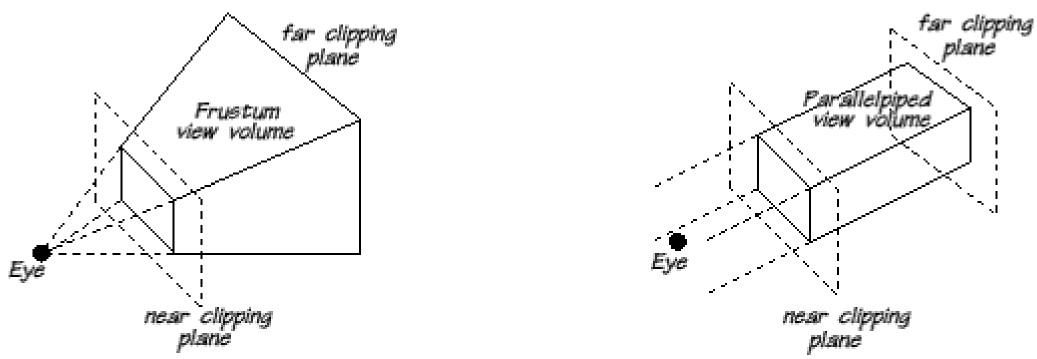


Figure 1: Left: Perspective projection; this projection uses a viewing frustum and projects the scene on the top of a pyramid. This projection is the same as how humans see the world. Right: Orthographic projection; the scene is projected on the top of a parallelepiped viewing volume. This projection is used for architectural blueprints. (Bazley, n.d.).

The perspective projection is the same as how humans see the world: the farther away an object is from the camera, the smaller it appears, because the viewing volume is a frustum of a pyramid. Objects that fall within the viewing volume are projected towards the top of the pyramid, where the camera is. Objects closers to the camera appear larger, because they occupy a proportionally larger amount of viewing volume than objects that are father away in the frustum. This projection is typically used in renderings as it corresponds to how humans see the world.

In the orthographic projection the viewing volume is a rectangular parallelepiped or simply a box. Because the size of the viewing volume doesn't change from one end to the other, objects that are closer to the camera do not appear bigger than object that are farther away. This type of projection is usually used for creating architectural blueprints. It can be used for creating renderings, but it doesn't match the view that people have in the real scene and thus will not be very realistic.

In order to create an image from a scene or an environment an image plane is inserted into the scene in front of the virtual camera. The 3D scene is projected onto the image plane to create the image that is the result of the rendering process. The goal of each rendering technique is to calculate a color for each pixel in the image plane, making use of integrators and material models to calculate the light distribution in the scene.

2.3 Material Models

Before it is possible to render realistic images, material models must be applied to objects and surfaces in the scene. These models describe the properties of the material and should match with the physical world as closely as possible. Material models specify how light is reflected, refracted or absorbed by an object or surface.

There are simple material models, such as diffuse and specular reflection models, and more complex material models, such as sub surface scattering models. An example of a diffuse and specular reflection model can be found in Figure 2. Diffuse reflection scatters light in all directions, while specular reflection reflects light in the direction that has the same angle to the surface normal as the incoming light. The latter is the kind of reflection that occurs in mirrors. In reality most materials have some mixture of both, i.e. diffuse and specular reflection.

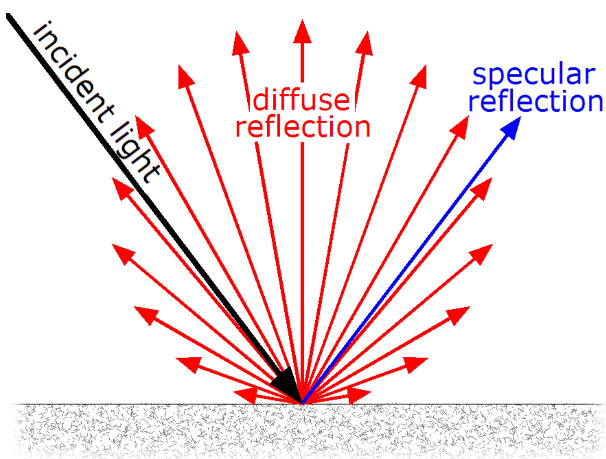


Figure 2: Diffuse and specular reflection. Diffuse reflection reflects light in all directions. Specular reflection reflects the light in the direction that has the same angle as the incident light with respect to the surface normal. (Wikipedia, Diffuse reflection, n.d.).

For creating the most realistic images these material models should match the physical world as close as possible. It is possible to measure the reflectance distribution of a material and create a model of it. For some materials these models may be relatively simple, but for other materials the related models may be very complex. Some material models therefore require a lot of parameters in order to simulate their physical world behavior.

2.4 Integrators

Integrators are essential in the rendering process as they construct light paths between the viewpoint or virtual camera, on the one hand, and the light sources in the environment, on the other hand. The performance of the integrators may vary depending on the technique used and some perform better with interiors than exteriors. So based on the scene that must be rendered, an optimal integrator must be selected.

Integrators can roughly be divided into two categories: ray based integrators and radiosity based integrators. Ray based integrators shoot rays from the viewpoint into the scene and calculate how the rays are bounced around in the scene until they hit the light source. Radiosity based integrators start at the light source and calculate how much light falls on every surface in the scene. The latter is useful when the light source is very small or hidden in the scene.

Some of the most popular integrators are described below. The first three integrators are ray tracing based integrators while the others are radiosity based.

2.4.1 Path Tracing

Path tracing calculates the color of each pixel in the image plane by tracing the physical paths of light traversing in the scene. The very basics of the path tracing algorithm are listed here (Jensen & Christensen, 2007):

- Shoot a ray from the viewpoint through a pixel position into the scene (view ray).
- Check if that ray intersects with an object in the scene.
- Set the pixel color to the color of the intersected object (if any).

This is the basic idea of path tracing, but the result is far from realistic. The quality of the image can already be improved by shooting multiple rays per pixel, but for realistically looking images, shading and reflection models to generate shadows and highlights in the image should be incorporated as well.

When a ray hits a surface with a diffuse or specular reflection, computing the color of the pixel requires shooting secondary rays from the point of impact of the first ray. These new rays should be shot in all direction where the light is coming from. As such, the number of rays for computing the pixel color increases rapidly.

Shadows are calculated by shooting so called shadow rays from the point of impact of the first ray to the surface of the light source. If these shadow rays can’t reach the light source, there must be an object that blocks that path. In that case there is shadow on the object that was hit by the first ray. These basic steps in path tracing algorithms are visualized in Figure 3.

These steps discussed above are only the basics of path tracing. A lot of other techniques have been invented to further increase the realism of the generated image. These techniques include soft shadows, depth of field and glossy reflection. (Jensen & Christensen, 2007) describes these techniques in more detail.

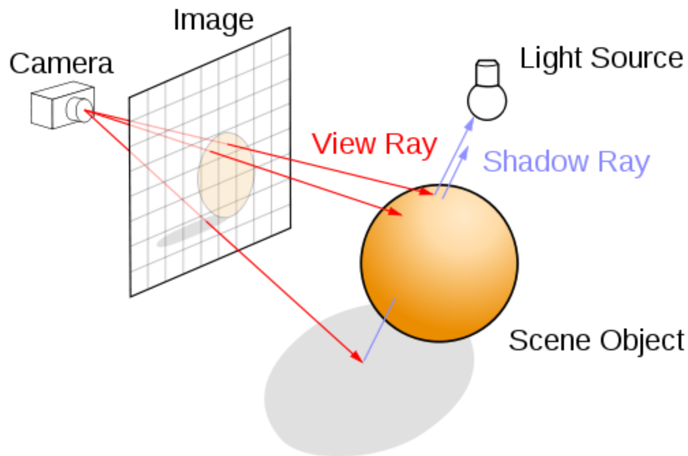


Figure 3: Visual representation of Path Tracing. Rays are shot through every pixel of the image plane and are reflected according to the material properties. Also shadow rays are visible to calculate shows. Shadow rays are shot in the direction of the light source and if they can't reach the light source, there must be shadow. (Wikipedia, Ray tracing (graphics), n.d.)

2.4.2 Bidirectional Path Tracing

Path tracing works pretty well for most scenes, but in scenes where light is hard to find, it can be problematic. An example may be a scene where a light source only illuminates a small part of the ceiling such that the rest of the room is only illuminated by indirect lighting. In such case only a few rays that are shot hit the small lit part of the ceiling, which results in an image with very bright spots while the rest of the image is black. Bidirectional path tracing provides a solution for such scenes.

With bidirectional path tracing rays are not only shot from the viewpoint but also from the light sources (Pharr & Humphreys, 2010). The path tracing rays are connected with the rays from the light source with so-called visibility rays. In this way more rays find light, and so the quality of the image is improved, while the computational time doesn't have to increase much.

2.4.3 Metropolis Light Transport

Another technique that improves standard path tracing is Metropolis Light Transport (MLT) and was first introduced by (Veach & Guibas, 1997). This integrator uses Metropolis sampling, first introduced by (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953). Metropolis sampling is a method for obtaining random samples from a probability distribution for which direct sampling is difficult. It can be used to approximate the distribution of the scene.

MLT uses a transition function to generate a sequence of paths through the scene. In addition, it makes use of a mutation strategy for local explorations. When a path that contributes significantly to the image is found it is easy to find other paths which are likely to also contribute significantly to the image. Hence, small perturbations to the original path are explored, while keeping the distribution of the sampled paths proportional to the power contribution they make to the image (Pharr & Humphreys, 2010).

An advantage of MLT is that it uses less rendering time to less important areas of the scene. A disadvantage, however, is that the time spent on darker regions of the image may be too low, such that these parts of the rendered scene become too noisy. As a result, noise may be non-uniformly distributed over the rendered scene.

2.4.4 Photon Mapping

Photon mapping starts at the light source rather than at the viewpoint and consists of two phases. In the first phase a predefined number of photons is emitted from the light source into the scene. Using the same principles as for path tracing these photons bounce around in the environment. At each bounce with a diffuse surface the intersection point, the photon power and the incoming direction of the photon are stored in a photon map. In addition, based on the properties of the material, it is decided whether the photon is either absorbed, diffusely reflected or specularly reflected. Once all photons are absorbed by the materials in the scene or have bounced more than a specified number of times the creation of the photon map is completed.

For efficiency reasons it pays off to divide the photon map into three different photon maps (Jensen & Christensen, 2007):

- *Global photon map*: an approximate representation of the global illumination of the scene with all diffuse surfaces.
- *Caustic photon map*: contains information about the photons that have been through at least one specular reflection or have been refracted before hitting a diffuse surface.
- *Volume photon map*: indirect illumination of a participating medium.

A reason for keeping different photon maps is that caustics, an envelope of light rays reflected or refracted by a surface of an object, should be of high quality. By separating the diffuse and caustic maps it is for example easier to put more effort in rendering caustics than in rendering the rest of the scene.

The second phase of the photon mapping technique is the actual rendering of the image. It consists of shooting rays from the viewpoint through a pixel into the scene. At each intersection of a ray with a surface the radiance leaving the surface in the direction of the ray can be extracted from the photon maps. The radiance leaving the surface in a particular direction is calculated as follows:

- Gather the N nearest photons near the intersection point.
- Let S be the sphere that contains these N photons.
- Divide each photon by the area of S and multiply with the surface material properties.
- Sum the results over the N photons.

The values of N can of course influence the quality of the rendering. By using too many photons, photons of different surfaces can be included in the sphere and materials can get mixed. By using too few photons the radiance might not be accurate.

An advantage of photon mapping is that the first phase is view independent. The photon map can be used for rendering images with different viewpoints and view angles.

2.4.5 Progressive Photon Mapping

Progressive Photon Mapping (PPM) is an improvement over standard photon mapping, since it doesn't store the complete photon map in memory (Hachisuka, Ogaki, & Jensen, Progressive Photon Mapping, 2008). This is possible because PPM consists of multiple passes. The first pass is an ordinary path tracing step to identify which surfaces are visible from the viewpoint through the set of image pixels. This pass is followed by multiple photon tracing passes. In each of these passes the standard photon mapping algorithm is applied, but with fewer photons. After each pass the radiances at the intersections of the path tracing step are computed and added to the radiance of the previous passes.

The radiance is estimated by looking at the density of photons within a sphere. After each pass more and more photons end up in such a sphere. Hence, the radius of the sphere should be reduced so that not too many photons are taken into account and that the radiance value converges to the correct value.

The photon map can be deleted as soon as the radiances are updated. Then a new photon mapping pass can begin and this process can be repeated until the quality of the image is satisfactory. This technique uses less memory, because smaller photon maps are used and memory is used over and over again.

2.4.6 Stochastic Progressive Photon Mapping

Stochastic Progressive Photon Mapping (SPPM) is an extension to PPM (Hachisuka & Jensen, Stochastic Progressive Photon Mapping, 2009). It introduces an extra distributed path tracing pass after each photon tracing pass. These distributed rays help to compute the radiance at the intersection points of the first path tracing pass. The idea is to compute the average radiance over a region.

The difference between SPPM and PPM is graphically represented in Figure 4. Both integrators use a path tracing pass to identify which surfaces are visible from the viewpoint and to store the intersection points. PPM takes the photons around the intersection points into account when calculating the radiance at an intersection point. SPPM does an extra path tracing pass after each photon pass and uses these intersection points to calculate local radiances. These local radiances are used to calculate the radiance at intersection points of the very first path tracing pass. So, the main difference is that PPM uses a fixed set of intersection points while SPPM uses intersection points randomly generated by the distributed path tracing pass. The use of a distributed ray tracing pass enhances features such as motion blur and depth-of-field.

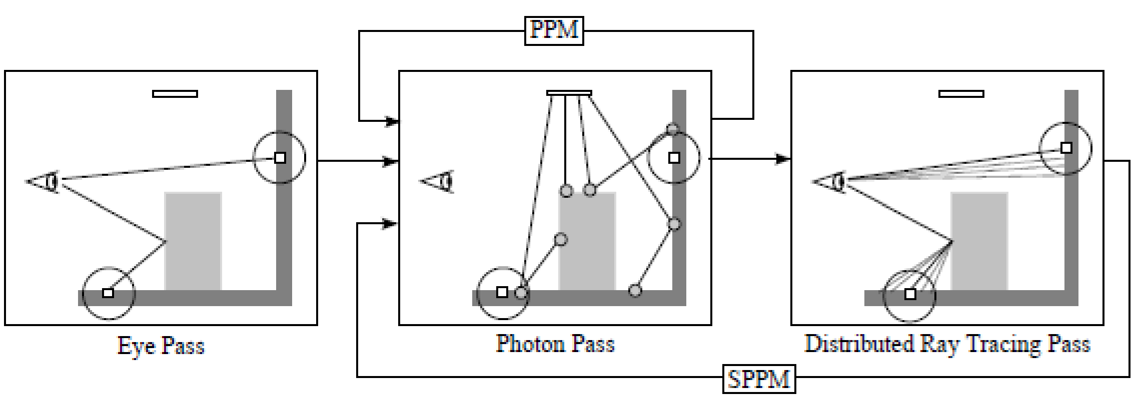


Figure 4: Difference between PPM and SPPM (Hachisuka & Jensen, Stochastic Progressive Photon Mapping, 2009). PPM only uses multiple photon passes while SPPM also uses an extra distributed ray tracing pass to calculate the average radiance over a bigger area.

2.4.7 Virtual Point Light Renderer

The Virtual Point Light Renderer was first introduced by (Keller, 1997). The basic idea of this integrator is fairly simple. It takes a random point on every light source and calculates the radiance in the scene with that point as a point light source. Any integrator that can handle radiance can be used for this approach. The images created from the different point light sources are then accumulated. The resulting image only represents the direct illumination part of the algorithm. Calculation of the indirect illumination part works in a similar way. Paths are traced from the light sources into the scene. When these paths intersect for the first time with an object in the scene, a new point light source is placed at that intersection. This way of working is visualized in Figure 5. The new virtual point light sources are used to calculate the radiance they distribute. The resulting images are again accumulated. The final rendering is obtained by accumulating images from the direct and indirect illumination parts of the algorithm.

Obviously, the quality of the rendering increases with the number of virtual point light sources, since the radiance in the scene is calculated more accurately with more point light sources. Also the path depth at which the virtual point lights are created can be increased to increase the quality of the rendering.

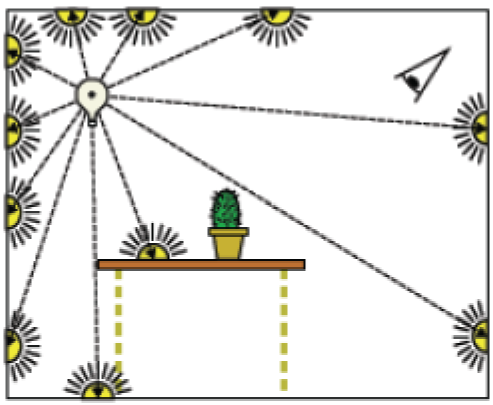


Figure 5: Virtual point lights in a scene. Virtual point light sources are created at the first intersection with rays from the light source and the environment. (Kinkelin & Liensberger, 2008).

2.4.8 Adjoint Particle Tracer

The Adjoint Particle Tracer (APT) starts just as the standard Path Tracer at the viewpoint, but instead of shooting rays, it shoots photons. So, an implementation of a Photon Mapping algorithm can be used. In the case of normal photon mapping the photons are shot from the luminaires and sensed by the camera. The APT exchanges the functions of the luminaire and the camera. The latter is possible due to the fact that light transport is self-adjoint (Morley, Boulos, Johnson, Edwards, & Shirley, 2006), which means that the light is closed under the involution operation. In other words, if function f is applied twice after each other to a light path, $f(f(x))$, then the result is the original light path.

The APT has some advantages over the standard Path Tracing and Photon mapping algorithms:

- Indirect and direct lighting are not separated, so there are no explicit shadow rays, making luminaires with reflectors not any more difficult than diffusely emitting surfaces.
- All adjoint photons have one wavelength rather than a spectrum, so reflections that vary in shape with wavelength present no difficulties.
- No photon maps are used, only the photons that interact with the sensor are logged.

Figure 6 illustrates the differences between a normal Photon Mapping algorithm and the APT. The direction of the photons is reversed and photons with different wavelengths are reflected differently.

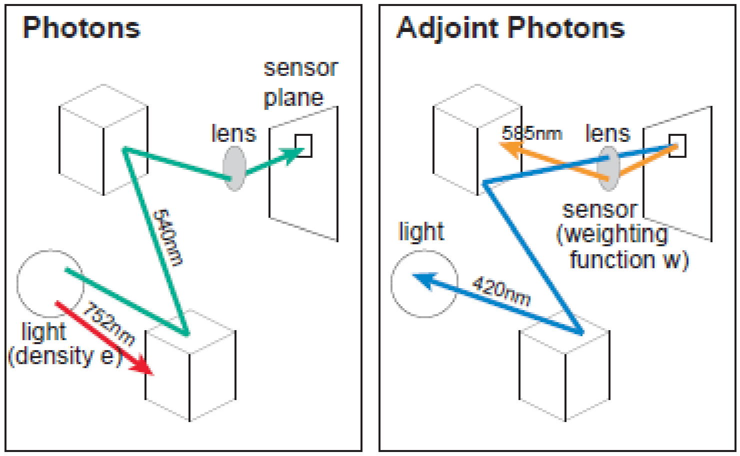


Figure 6: Difference between Photon Mapping and APT (Morley, Boulos, Johnson, Edwards, & Shirley, 2006). Where normal photons originate at the light source, adjoint photons start at the camera.

2.5 Improvements in Rendering Techniques

While the render techniques described above already produce nice images, there is always room for improvement. Some improvements decrease the rendering time, while others increase the quality of the renderings. A few of the most common improvements are described in this section.

2.5.1 Russian Roulette

Russian Roulette is a different way of tracing rays. Where normal path tracing scales the intensity of rays according to the reflection properties of the material model of the surface it intersects with, Russian Roulette scales the probability of actually shooting a ray. For example, if a surface has a specularity of 0.2, the normal path tracer scales the intensity of all rays by 0.2, while in the case of Russian Roulette rays with original intensity are only generated 20% of the time. This results in less rays to be shot, which improves render time.

2.5.2 Importance Sampling

Importance Sampling is a method for improving the quality of images during rendering with a Photon Mapping renderer (Christensen, 2003). It starts with emitting importance particles from the viewpoint. They are, just as in a Photon Mapping algorithm, stored as they hit a diffuse surface. After this pass, the normal photon tracing pass commences. Photons are probabilistically stored based on the importance map constructed in the first pass. The result of this extra pass is that it is more likely that more photons are stored in the part of the scene that the camera can see, thus providing more detail in the final image. Fewer photons are stored in parts of the scene that are not visible in the image.

2.5.3 Irradiance Caching

A way to speed up the path tracing process is to make use of the irradiance caching algorithm. Irradiance caching is based on the property that indirect lighting does usually not change rapidly from point to point, while direct lighting does (Pharr & Humphreys, 2010). With this in mind, the indirect lighting component only has to be calculated at some points in the scene, and can be further interpolated at intermediate points. It can be hypothesized that the error introduced by irradiance caching is not that big, while the computational time needed can be reduced significantly.

2.5.4 Adaptive Integration

Adaptive integration is a technique that runs once the integrator is finished. It starts a second integrator that only samples the pixels that have a certain luminance difference with their neighboring pixels. This luminance difference threshold and how many samples the adaptive integrator should do can be specified by the user. By looking at the luminance difference of neighboring pixels the adaptive integrator basically focuses on anti-aliasing and decreases noise in problematic areas of the rendering. An example is given in Figure 7. The white pixels in this rendering will be sampled by the adaptive integrator.

Figure 7: The white points in the scene have a luminance difference with the neighboring pixels greater than a certain threshold and are therefore sampled by the adaptive integrator to reduce aliasing (YafaRay, n.d.).

2.6 Properties of Rendering Techniques

Sometimes it is necessary to know if an integrator makes errors, and if it does how big these errors are. The latter can be estimated depending on if an integrator is biased or unbiased and consistent or inconsistent.

2.6.1 Biased versus Unbiased

Integrators can either be biased or unbiased. (Crane, n.d.) gives a nice explanation of the difference between a biased integrator and an unbiased integrator. An integrator is unbiased if it produces the correct answer on average. So, if an image is rendered a lot of times, then averaging the images would give the right answer. Thus, the expected error of an unbiased integrator is zero. If all this is not the case, then the integrator is biased. The error of a biased integrator is proportional to the variance of the estimator.

A common source of bias in renderings is the fact that integrators ignore some type of lighting effect. Not all bounces of a light path can be calculated and also some light paths may be too complicated for the integrator. A complex light path may go through multiple glass surfaces where each time the refraction has to be calculated. Another source of bias is that surfaces are approximated with triangles which may be not accurately enough. Bias is usually intentionally introduced to speed up the rendering process.

2.6.2 Consistent versus Inconsistent

An integrator is consistent if the approximation approaches the correct solution as computation time increases. So by letting these integrators run for a long time the approximation gets better and better. A disadvantage, however, is that nothing can be said about how fast the approximation converges to the correct solution. It is also impossible to give a bound to the error.

2.6.3 Overview

(Crane, n.d.) gives an overview of a few commonly used integrators and indicates if they are biased and/or consistent or not. This overview is reproduced in Table 1.

Integrator	Unbiased	Consistent
Path Tracing	✓	✓
Bidirectional Path Tracing	✓	✓
Metropolis Light Transport	✓	✓
Photon Mapping		✓

Table 1: Overview of integrators and if they are unbiased and/or consistent or not.

There are multiple reasons why the photon mapping algorithm can be biased. When there are too few photons in the caustic map, caustics will appear blurry, and averaging over blurry images won't result in a sharp image. Another source of bias occurs when too many photons are used. In that case, the region used to estimate the radiance shrinks to a point. In the limit only one photon is used to estimate the radiance at the end of a light path.

2.7 Light Distribution

Until now it has been discussed how light is sampled and traced through an environment. What is still missing is where and with what distribution that light is generated. The spatial distribution of the light leaving a luminaire is dependent on the reflector of the luminaire and on the light source. Sometimes it is very hard to model the reflector of a luminaire and as a consequence the light leaving the luminaire does not match with reality. Manufacturers provide a digital model of how a luminaire distributes light in so-called Illuminating Engineering Society (IES) profiles. An IES profile has a standard file format that holds information on the distribution of the light intensity from a light source. Using IES profiles makes the modeling of luminaires a lot easier.

With a simple luminaire model and an IES profile, the luminaire may look the same as a real life luminaire. Figure 8 demonstrates the effect of using an IES profile for one particular example. The left part of the figure shows a square light source. Light that is emitted from the square is evenly distributed over the environment, just as expected. When an IES profile is added to the square, as is done in the right part of Figure 8, the light emitted by the square has a different and more realistic distribution.

There are, however, some limitations to IES profiles. An IES profile describes the light distribution from a point light source. When a luminaire is bigger, multiple point light sources with the same IES profile are applied to that luminaire. Another limitation is that the light distribution in an IES profile is symmetric, while this is not always the case with real luminaires.

Figure 8: Left: Square luminaire without IES profile; Right: Luminaire with IES profile. It can be seen that the light distribution is changed by the IES profile.

2.8 Tone Mapping

In real world scenes the range of light intensities people experience is very large, varying from star light to sun lit snow. However, the range of light intensities that can be reproduced by prints or by screens is much smaller. Physically realistically rendered images also span a bigger range of light intensities than screens or prints can reproduce. Such images are called High Dynamic Range (HDR) images. Most displays have a smaller dynamic range, and therefore, are called Low Dynamic Range (LDR) displays.

In order to display HDR images on LDR displays, the range of light intensities in the image has to be reduced to match the range of light intensities the display can reproduce. The latter can be done by a so called Tone Mapping Operator (TMO). TMOs can be classified in two classes: global operators and local operators (Yoshida, Blanz, Myszkowski, & Seidel, 2005) (Villa & Labayrade, 2010).

Global operators work uniformly on the whole image: each pixel is treated the same. Local operators adjust the treatment of a pixel based on its intensity and the intensity of the surrounding pixels. Both (Yoshida, Blanz, Myszkowski, & Seidel, 2005) and (Villa & Labayrade, 2010) did experiments to find out what TMO operator was preferred by participants. From these experiments they concluded that the TMOs by (Reinhard, Stark, Shirly, & Ferwerda, 2002) and by (Drago, Myszkowski, Annen, & Chiba, 2003) are rated as most realistic.

Also Philips Research already investigated which TMO resulted in the most realistically looking images (Salters, Murdoch, Sekulovksi, Chen, & Seuntiens, 2012), and recommended to use the TMO by Reinhard. As this TMO, referenced to as Reinhard02, is used in the experiments described in this report, it is explained in more detail here.

2.8.1 Reinhard02

Reinhard02 starts by determining the key value of a scene. The key value is an indication whether a scene is subjectively light, normal or dark, with a high key value corresponding to a bright scene. The key value is approximated, just as many other TMOs do, by computing the log-average luminance of the scene, using the equation:

$$\bar{L}_w = \frac{1}{N} exp\left(\sum_{x,y} log(\delta + L_w(x,y))\right)$$

Where:

$\bar{L}_w$	is the log-average luminance of the scene to approximate the key value.
$L_w(x,y)$	is the "world" luminance for pixel (x,y).
N	is the total number of pixels in the image.
δ	is a small value to avoid singularity that occurs if black pixels are present in the image.

The scene now has to be mapped to the subjective middle brightness range of the scene. This mapping is done with the following equation:

$$L(x,y) = \frac{a}{\bar{L}_w} L_{w(x,y)}$$

Where:

$L(x,y)$	is a scaled luminance.
a	is the key value to map the scene to. This value usually is 0.18 but can be defined by the user.

The value of a depends on the key value of the scene and 0.18 is usually used if the scene has a normal key value. Higher values of a will result in brighter images.

The problem with this last equation is that many scenes have a normal dynamic range with a few high luminance regions. Traditional photography compresses both low and high luminance values. Modern approaches, however, compress and clip mainly the high luminance values. Compressing low luminance values, while clipping high luminance values can be done with the following equation:

$$L_d(x,y) = \frac{L(x,y)\left(1 + \frac{L(x,y)}{L_{white}^2}\right)}{1 + L(x,y)}$$

Where:

L_{white}	is the smallest luminance that will be clipped to pure white.

2.9 Realism

To understand how to render realistic images, a definition of realism has to be given. People may have a different understanding of what they define as real. (Ferwerda, 2003) describes three types of realism in computer graphics:

- Physical realism: The image provides the same **visual stimulation** as the scene.
- Photo realism: The image produces the same **visual response** as the scene.
- Functional realism: The image provides the same **visual information** as the scene.

Each type of realism has its own advantages and disadvantages. They are described in the next sections.

2.9.1 Physical Realism

An image with physical realism tries to provide the same visual stimulation as the real scene, which can only be achieved if the image is a point-by-point representation of the irradiance values at a particular viewpoint in the scene. In order to do this, the model of the scene must contain accurate descriptions of the shapes, materials and illumination properties of the scene. In addition, renderers must be able to understand these descriptions in order to make a realistic image.

Although this technique creates physically realistic images, it is very computational expensive and therefore not suitable for interactive applications. Moreover, the images that are created using this technique cannot be displayed accurately with common displays, since some form of tone mapping is usually needed. Another disadvantage is that the limitations and capabilities of the human eye are not taken into account. This may result in spending too many resources in something that humans can barely see. Therefore, research has investigated the concept of photo realism.

2.9.2 Photo Realism

A photorealistic image is indistinguishable from a photograph of the scene if it produces the same visual response. In other words, different images may look the same to humans as long as they produce the same visual response. A lot of research has been devoted to understand how the Human Visual System (HVS) responds to certain input. Considerable progress has been made, but still visual processes, such as masking and adaptation, are not fully understood yet.

Renderers can be optimized based on knowledge of the HVS, by focusing on those parts of the image to which the HVS is most sensitive. This will probably reduce the rendering time of images, but the images won't be physically realistic anymore. Still these techniques are too slow for interactive applications. Moreover, also in the case of photo realism displays can't accurately display the generated images.

2.9.3 Functional Realism

The third type of realism is functional realism in which the image has to provide the same visual information as the scene. A good example of functional realism can be found in almost any *do it yourself* manual. These manuals usually make use of illustrations instead of photographs to indicate how a certain task should be performed. Making use of drawings has certain benefits:

- Drawings can omit certain elements that are not important for performing the task.
- Drawings can easily group certain elements in the picture for example by giving them the same color.
- Drawings can show viewpoints which are difficult or even impossible to create in real scenes.
- Drawings can make use of special effects to make things clear, for example artificial transparency.

2.10 Perception Judgments

Image perception can be assessed objectively and subjectively on different attributes, such as quality, naturalness, realism, etc. Objective measures are meant to replace time-consuming subjective judgments, but are only able to do so if they properly include HVS characteristics. Specifically for realism, objective metrics are not available yet. Hence, we have to rely on subjective testing to investigate how realistic rendered images are. This chapter describes various methodologies to judge the rendered images subjectively.

2.10.1 Psychophysics

Psychophysics investigates the relation between a parameter of a physical stimulus and its sensation. A possible result of this research is a so called psychometric function. This function describes the relationship between a parameter of a physical stimulus and the response of a person on a certain aspect of that stimulus. The psychometric curve usually resembles a sigmoid function, an example of which is given in Figure 9. The x-axis corresponds to the intensity of the stimulus and the y-axis to the probability of yes-responses. If, for example, the x-axis is the volume of a sound, then the y-axis corresponds to the proportion of people that could hear that volume. With psychophysics it is for example possible to find out how strong certain stimuli must be in order to be perceived by the majority of the people.

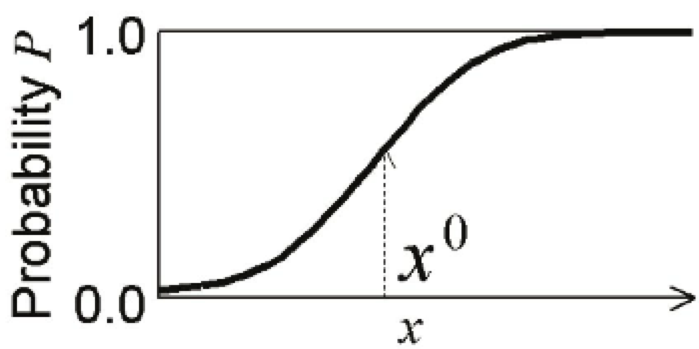


Figure 9: Example of a psychometric curve (Matsuo & Fujimoto, 2011) with on the y-axes the probability that a participant responds yes to a stimulus with an corresponding intensity on the x-axes.

The X_{50} point on the psychometric curve is the point where half of the participants responded yes, e.g. to the question whether they saw a certain light spot or heard a certain volume. This point is usually referred to as the absolute threshold. For some tasks, however, 50% is just the point that is obtained by chance. For example, in case people are forced to choose between two options, the 50% point is achieved in the condition where the two options are equal. In such a situation the X_{75} point is a more suitable threshold estimate (Kaernback, 1991).

2.10.2 Paired Comparison

In a paired comparison experiment two stimuli, A and B, are simultaneously presented to a participant (David, 1988). In its basic form the participant has to indicate which of the two stimuli he prefers. It is also possible to provide, for example, a seven point scale and to ask the participant to scale his preference towards one of the two stimuli. Possible points on such a scale may be “Strong preference for A", "Preference for A", "Slight preference for A", "No preference" and of course the same for stimulus B.

The experiment can become lengthy when a lot of stimuli have to be compared with each other. When there are n stimuli to be compared there are $n(n-1)/2$ possible pairs that have to be judged. Based on this formula it is evident that the number of pairs increases rapidly with n.

One way of reducing the number of pairs to be judged is to ask participants to only assess a subset of the pairs. This subset should be different for every participant. In this way all the pairs can be judged, but not every pair is judged by every participant.

Another way to reduce the number of pairs is to make use of a rank order methodology. In that case multiple stimuli are presented to the participant at the same time. The participant is then asked to rank the stimuli according to his preference. From the reported rank order it is possible to deduce which stimulus is preferred over any other stimuli. As a consequence, the collected data in a rank order experiment is an approximation of paired comparison data.

2.10.3 Quality Ruler

A good way for measuring image quality is by using a quality ruler (Keelan & Urabe, 2004). A quality ruler consists of a series of reference images which vary widely in a single attribute. There is a known quality difference, just notable difference (JND), between the samples of the series. Because the JND values of the reference images are known, any other image that is placed on the ruler can get a numerical value as well. The quality ruler can be implemented both in a hardcopy or a softcopy version. Only the softcopy version is discussed here, because this research focuses on virtual prototyping.

The task for placing a given image on a quality ruler can be made similar for the user as a paired comparison task. In that case a ruler image is displayed on the left, while a test image is displayed on the right. The first ruler image and test image are selected randomly. The observer now has to select the image he considers of higher quality. This can either be done pressing the corresponding buttons on the screen or by pressing the left or right arrow key. The quality ruler then chooses a new ruler image, using a binary search algorithm. The selected image lays approximately halfway between the lowest image quality ruler image that has 'won' a paired comparison and the highest quality ruler image that has 'lost'. At the start of the trial, the highest and lowest quality images are used as defaults. The process is terminated when the test image is rated differently with respect to two adjacent ruler images. The quality of the test image is then the average of the quality of the two adjacent ruler images.

2.10.4 Staircase Method

The paired comparison method can be used to order stimuli with respect to, for example, preference by the participants. However, sometimes it is necessary to find a threshold at which people find a given stimulus good enough. A good method for finding a threshold is the staircase method (Ehrenstein & Ehrenstein, 1999).

In its simplest form the process starts by presenting a stimulus to the participant and the participant is asked to indicate whether that stimulus is good or not. In case the stimulus is good, its quality is decreased in a next presentation. This continues until the stimulus becomes not good enough. At this point the quality of the stimulus increases again until the participant considers it good enough again. Usually the process continuous until the participant showed six to nine reversals in his judgment. The threshold is then determined by taking the average of the last four to six reversal points.

The stimuli presented in the staircase method are all concentrated around the threshold, and as a consequence, fewer stimuli have to be presented to the participants. However, there is a problem with the staircase method. Participants may become aware of the intended sequence of stimuli that is presented to them. As a result they may anticipate their answer on the knowledge of the sequence. A solution to this is to interleave multiple staircases and present those to the participant in a random order. A visual comparison between a standard staircase and interleaved staircases is shown in Figure 10. This figure also illustrates how the step size may vary over time. The steps in the beginning of the staircase are bigger than the steps at the end. As such, the assessment procedure quickly jumps to the region around the threshold, and then refines the threshold value using smaller steps.

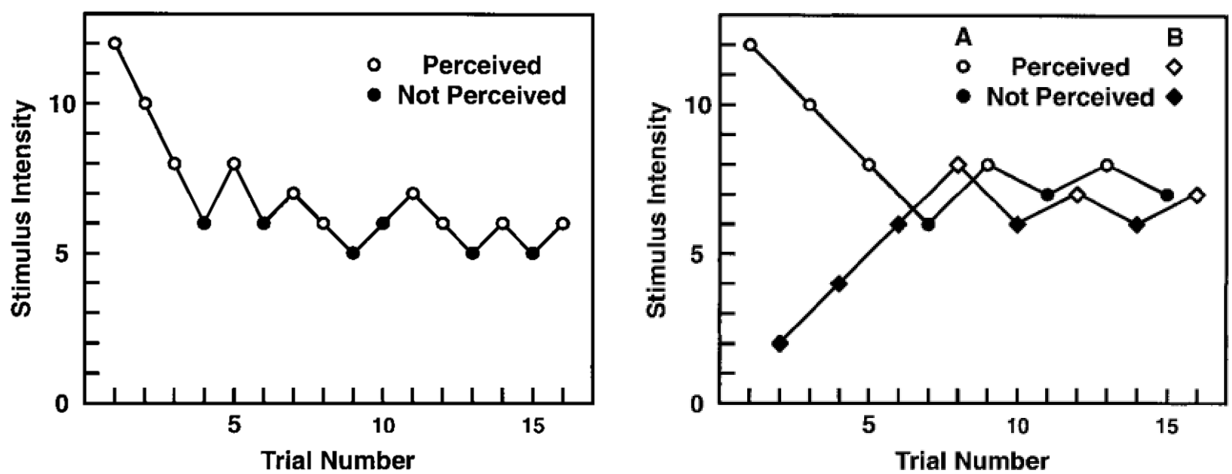


Figure 10: Left: Single staircase with reversions and decreased step size over time. Right: Two interleaved staircases to exclude any learning effect.

Where a simple up-down method (1 up, 1 down) converges to the X_{50} point of the psychometric curve, a weighted up-down method as proposed by (Kaernback, 1991) converges to the X_{75} point. The key difference is to use different step sizes for up and downward steps. The stimulus level is decreased by 1 step after each yes response and increased by 3 steps after each no response.

3 Render Software

There is a lot of software available, both commercially or as free software, that simulates light in a virtual scene. Simulating light is still academic research, so it is possible that software packages get updated with new or better algorithms. Most of the current software packages available are reviewed in the literature. A performance comparison is done with some of them, for example in (Roy, 2000). In this paper four light simulation packages are compared for use in architectural design.

This chapter describes two of the software packages that can be used for rendering: Mitsuba Renderer and Indigo Renderer. Both are used for creating the renderings in our experiments.

3.1 Mitsuba Renderer

Mitsuba[1] is a physically based renderer that has a wide range of rendering techniques available. It is created by Wenzel Jakob and is an open-source research-oriented renderer. Mitsuba is relatively young and freely available. As a consequence, it misses some features that are expected in rendering software. This section describes many of the aspects of Mitsuba in order to import 3D scenes, adjust the scenes and create renderings of them.

3.1.1 3DS Max to Mitsuba

Many 3D scenes are created using 3D Studio Max or any other 3D modeling software. In order to render these scenes with Mitsuba, these scenes have to be ported to a format that Mitsuba understands. Mitsuba works with an Extensible Markup Language (XML) file which gives a description of the scene. All the meshes that are used in the scene are stored in a separate file. The XML file specifies the translation, scale and rotation of all the meshes. It also defines which mesh uses which material and it gives the material definitions. Finally it specifies the location and orientation of the camera.

At this point in time it is not possible to directly export scenes created in 3DS Max to Mitsuba. Fortunately Mitsuba can import COLLADA files and convert them to Mitsuba's own file format. 3DS Max can export scenes to COLLADA. So via the COLLADA file format the scene was converted from a 3DS Max scene to a Mitsuba scene. Unfortunately, not every aspect of the scene was converted properly. Materials definitions, camera properties and textures were not always imported properly.

In order to overcome these problems some work had to be done manually. First of all, the definitions of the materials, such as the specular and diffuse reflection parameters, had to be copied from the 3DS Max scene. This step also included putting the right textures on surfaces and specifying the right textures for, for example, bump mapping. Finally, the camera had to be put in the correct position and with the correct orientation to match the viewpoint of the real scene.

3.1.2 XML files

Mitsuba renderer provides some additional useful features with its XML file format. It is possible to link separate XML files to each other. This may be used to define the whole scene in one XML file, and in

[1] Mitsuba – physically based renderer – http://www.mitsuba-renderer.org/ (Accessed: June 18, 2012)

addition, use different XML files for the different light settings. If something has to be changed in the scene, then only one file has to be edited. The change then becomes visible for every light setting.

Another useful feature is the possibility to use variables in XML files. A variable can be used for, for example, the position of an object or the color of a luminaire. At the start of the rendering process a value can be given to this variable. The advantage is that only one scene file is needed if only a small element of a scene changes a lot of times.

3.1.3 Usage

When a scene is loaded in Mitsuba it displays a quick preview of that scene based on a fast rendering using the virtual point light renderer. In this preview the viewpoint is placed at the position of the camera, as specified in the XML file describing the scene. With the mouse it is possible to freely move the camera around in the scene, which is useful in the case one wants to have a closer look to objects that were changed. It is possible to save the new camera position in the XML file, so that the next time the scene is loaded the camera is at the exact same location. It is also possible to select objects in the preview by clicking on them. However, at this point in time, it is not possible to do any operations on these objects. Just their name is provided, which is helpful to find their specifications in the XML file and to change their material properties.

Once the camera is in the right position it is time to specify the appropriate rendering settings. The first thing that has to be specified is the type of integrator that has to be used. Mitsuba supports different integrators which are listed in section 3.1.4. This section also describes some important parameters to tweak the integrators in order to get the best result for every type of scene. Once all parameters are set, it is time to start the rendering process.

3.1.4 Supported Rendering Techniques

Mitsuba currently supports the following rendering techniques:

- Direct illumination
- Path tracer
- Adjoint particle tracer
- Virtual point light renderer
- Photon mapping
- Progressive photon mapping
- Stochastic progressive photon mapping

All of them are described in more detail in section 2.4. Every rendering technique has its own parameters to configure the integrator to the best performance. These parameters can be different for every software package, however most of the time they share a lot of settings, sometimes just under different names. The most important parameters for each of the supported rendering techniques in Mitsuba are explained here.

3.1.4.1 Important Parameters

The following parameters are important for the path tracer and the adjoint particle tracer:

- *Maximum depth*: Specifies the longest path depth in the generated output image.

- *Russian Roulette starting depth*: Specifies the path depth, after which the Russian Roulette algorithm is applied to the path.
- *Samples per pixel*: Number of rays to be shot per pixel.

The important parameters for the virtual point light renderer are:

- *Number of virtual point lights*: The number of virtual point lights created in the scene.

The parameters that are used for the standard path tracer can be used for the photon mapping as well. In addition, the following parameters are important:

- *Global, caustic and volume photons*: Number of photons to collect for the global, caustic and volume photon maps.
- *Global, caustic and volume map lookup size*: Number of photons to consider in a global, caustic or volume photon map lookup.

The PPM also uses the same parameters as the standard path tracer. In addition it has the following important parameters:

- *Photons per iteration*: Number of photons to shoot in each iteration.
- *Size reduction parameter*: The speed at which the photon radius is reduced.

The SPPM integrator uses the same parameters as the progressive photon mapping integrator.

3.1.5 Output

During the rendering process the image of the scene gets better and better in terms of quality. However, this only holds for some integrators, while others divide the scene in equally sized cells and start rendering cell by cell. In the latter case the image gets updated cell by cell. The image resulting after rendering is not tone mapped yet, and therefore may look weird, for example all white or all black. Mitsuba can render HDR images and can also tone map them in order to display them on LDR devices. It is possible to specify the tone map parameters in Mitsuba, however this is done only roughly with a slider. Therefore it is better to save the final image as a HDR image and to perform the tone mapping off-line with an external program, such as *Luminance HDR*[2].

3.1.6 APT Flaw

Unfortunately, there is a flaw in the Adjoint Particle Tracer integrator of Mitsuba. It renders the scene and the light paths correctly, but light sources itself appear dark. An example of this flaw can be found in Figure 11, in which part of a scene is rendered both with the APT and the Path Tracer. In order to overcome this limitation, *Photoshop* is used to copy the light source from the image rendered with the Path Tracer to the image rendered with the APT.

[2] Luminance HDR – http://qtpfsgui.sourceforge.net/ (Accessed: June 18, 2012)

Figure 11: Same image rendered with Adjoint Particle Tracer and Path Tracer. The light source in the rendering of the Adjoint Particle Tracer appears dark.

3.2 Indigo Renderer

Indigo is in contrast to Mitsuba not free and not open-source. It is already on the market for a much longer time than Mitsuba and therefore has a greater user base. It is also more user friendly with a nice GUI for every part of the software package. This section describes some of the advantages of Indigo over Mitsuba.

3.2.1 Integrators

The integrators supported by Indigo are all path tracer based. In contrast to Mitsuba, it supports the more advanced path trace integrators, such as MLT and bi-directional path tracing. It even supports a combination of MLT and bi-directional path tracing for extra performance. Indigo doesn't support a photon based integrator.

3.2.2 Rendering View

Just like Mitsuba, Indigo has a preview window where the rendering up to any point in time is shown. This preview gets better in quality as more time passes. It is also possible to tone map the rendering with different TMOs and parameters and the result is visible in the preview window.

3.2.3 Material Editor

Indigo comes with an integrated material editor. Once it loads a material it instantly renders it. In the GUI of the material it is possible to adjust the parameters of the material. The adjusted material is rendered instantly. This is much more convenient than editing XML files and having to render a whole scene in order to view its result as must be done with Mitsuba.

3.2.4 Light Layers

Where Mitsuba only saves the total image, Indigo saves how much radiance is received by every part of the scene. This has many advantages. The first advantage is that rendering can be continued at a later time. When a rendering is continued, Indigo just adds the radiance to the already existing radiance and scales it appropriately.

Another advantage of storing radiances is the possibility to use light layers. In the construction of a scene it is possible to assign every luminaire to a different light layer. During and after rendering it is possible to switch these light layers on and off without having to start the rendering from the beginning. This basically switches on and off luminaires in the scene and the result is immediately visible in the rendering preview. It is also possible to adjust the color and the power of each light layer. Also IES profiles can be added to the light layers. So with only one scene setup it is possible to create renderings of the same scene under different lighting conditions and the lighting can be adjusted once the rendering is finished.

3.2.5 Exporter

Most 3D modeling software has an exporter for Indigo included. An exporter basically exports the whole scene that is modeled to Indigo for rendering. If an exporter for Indigo is included in the modeling software, it is possible to work with materials that Indigo understands. There is no need for editing the scene in XML files.

4 Experiment 1

4.1 Introduction

The goal of the first experiment was to investigate which integrator produced the most realistic images considering the effort that was needed to create these images. To this end, participants were asked to compare a real world scene with a rendering of that same scene, where the scene used was the Light Lab of Philips Research. An accurate three dimensional (3D) model of the Light Lab was already available from the research reported in (Salters & Seuntiens, A Comparison of Perceived Lighting Characteristics in Simulations Versus Real-Life Setup, 2011). Renderings under different lighting conditions were generated with the different integrators, and the resulting images were compared to the real scene on their perceived realism.

4.2 Method

In order to compare the renderings with each other a paired comparison method was used. Two images each based on a different rendering process were shown next to each other to each participant. The participant had to indicate which of the two he considered most realistic. Each participant had to do this twice: once while seeing the real scene and once without seeing the real scene. For every participant the order of the rendering pairs, the placement left - right of the stimuli, and the order of the two repetitions were randomized.

4.2.1 Participants

A total of 20 participants voluntarily participated in this experiment. Ten of them were male and the other ten were female. Eighteen of the participants were students and the other two were regular employees of Philips. The average age of the participants was 24.5 years with ages ranging from 21 to 33 years. One of the participants was red-green colorblind, but nonetheless included in the analysis and results.

4.2.2 Equipment and Setup

As explained above, the experiment consisted of two parts. One part of the experiment took place in the Light Lab of Philips Research. Participants had to take place in a tent that limited their field of view, such that it matched what they saw in the renderings. Figure 13 illustrates the view a participant had of the Light Lab. The only other object in the room was a small table, placed next to the door. On the left side of the participant a display showed the various renderings of the same scene. The participant was asked to choose the rendering that had the most resemblance with the real scene. He could indicate his choice by pressing the corresponding key on the keyboard that was in front of the screen. A top view of this experiment setup is given in Figure 12.

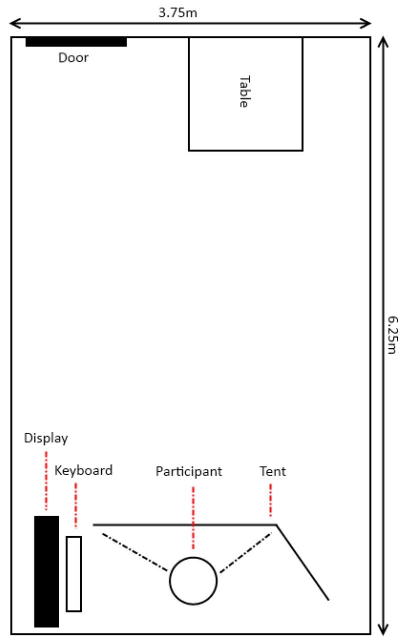


Figure 12: Top view of the experiment setup with the participant placed in a tent to limit his field of view and a display with keyboard to the left of the participant.

Figure 13: Participants view of the light lab.

The second part of the experiment took place in the same room, but could as well have taken place in a different room. In this part of the experiment the tent was closed so that the participants could not look to the real room. Apart from this difference, the experimental setup for this part of the experiment was similar to the first part. The participant was asked to take place in front of the same screen as used in the first part of the experiment. By using the paired comparison method, the participant was asked to indicate which image of the two was considered most realistic. The same renderings were used in both parts of the experiment.

For each participant it was randomly selected with which part of the experiment to start.

The Light Lab was equipped with different luminaires, namely Philips Savio down lights, spot lights that face downwards, from now on referred to as down lights, and spot lights that directed at the wall, from now on referred to as spot lights. There were six luminaires of each type and they were positioned according to the layout, shown in Figure 14. The Savios consisted of 2 CCFL lamps with a color temperature of 2700K and 4 CCFL lamps with a color temperature of 6500K. When the Savios were used in the experiment, only the high color temperature lamps were used. The down lights and the spot lights had a color temperature of 3000K. The six down lights had an opening angle of 18 degrees and the six spot lights an opening angle of 11 degrees.

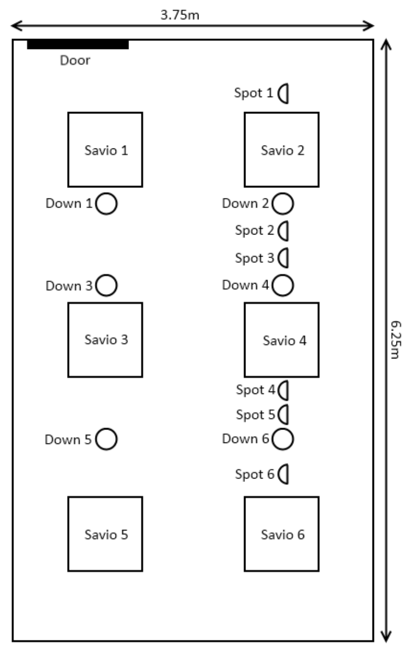


Figure 14: Light layout of the Light Lab.

All participants were exposed to different lighting configurations, namely:

- All six Savios on.
- Savio 1, Savio 2, Savio 5 and Savio 6 on.
- Savio 1, Savio 4 and Savio 5 on.
- Savio 3 and Savio 4 on.
- All six spot lights on.
- All six down lights on.

In order to exclude any order effect, the order in which the light configurations were assessed, was randomized for each participant.

4.2.3 Stimuli

Mitsuba renderer 0.3.0 was used for rendering the images of the different lighting settings. Four different integrators were used to generate the renderings from the 3D model. For each integrator most of the settings were kept the same:

- Sampler: Independent sampler
- Resolution: 960x540 (for side by side comparison on a full high definition display)
- Reconstruction: Windowed Gaussian filter
- Other features: No Irradiance Cache and no Adaptive Integration

The following integrators with their specified settings were used. Note that when settings are not specified the default value of Mitsuba was used.

- Path Tracer
  - Maximum depth: -1 (infinity)
  - Russian Roulette starting depth: 10
  - Samples per pixel: 4096
- Adjoint Particle Tracer
  - Maximum depth: -1 (infinity)
  - Russian Roulette starting depth: 10
  - Samples per pixel: 4096
- Photon Mapper
  - Global photons: 100000000
  - Global photon map lookup size: 700
  - Russian Roulette starting depth: 10
  - Samples per pixel: 4096
- Stochastic Progressive Photon Mapper
  - Photons per iteration: 500000
  - Maximum depth: -1 (infinity)
  - Russian Roulette starting depth: 10
  - Samples per pixel: 4096

Also renderings created with Indigo as previously used by (Salters & Seuntiens, A Comparison of Perceived Lighting Characteristics in Simulations Versus Real-Life Setup, 2011) were included in the experiment. These renderings had the same light configurations as the ones rendered with Mitsuba. However, they were still based on the light lab with old carpet. These renderings were included in order to have some reference to the previous experiment. Some examples of renderings for the light setting with all 6 Savios on are added in appendix 10.2.

The time needed to complete the renderings is given in Table 2 per integrator. The rendering time for the Indigo software is much longer than for the Mitsuba software. The extra rendering time was due to the fact that the Indigo renderings used a higher resolution, that all the different light configurations were computed at the same time and that the rendering had to be free of noise.

As stated in literature the SPPM integrator might run for an infinite amount of time (Hachisuka & Jensen, Stochastic Progressive Photon Mapping, 2009). Therefore the rendering time for that integrator is not given. In order to make the comparison between the integrators more fair, also renderings over the same rendering time were included. Therefore, the APT and the path tracer renderings were repeated for the same amount of time as the photon mapping integrator. In order to have two different quality levels for the SPPM integrator, the result was saved after the average rendering time of the APT and the path tracer and after the rendering time of the photon mapping integrator.

So, the whole process resulted in two batches of renderings: one where most of the rendering settings were equal and one where the rendering time was equal. Note that in the remainder of this report the integrators that ran for the same amount of time have the prefix “time_” in front of their name.

Integrator	Render time (hours)
Indigo	72
Adjoint Particle Tracer (APT)	1
Path Tracer	1.75
Photon Mapper	5.25

Table 2: Render time per integrator.

After some pilot renderings it turned out that the Path Tracer combined with the down light and spot light scenes didn’t perform well, since it was very hard for the rays to find the small light sources. In order to create nice looking pictures the Path Tracer had to render those scenes for about a dozen times longer than the other integrators. Therefore the renderings for these combinations were left out of the experiment.

The renderer produced HDR renderings of which the luminance range had to be reduced in order to be able to display them on a LDR screen. The software package *Luminance HDR* was used to convert the HDR images to LDR images. The TMO defined by (Reinhard, Stark, Shirly, & Ferwerda, 2002) was used with the following parameters:

- Key value: 0.27
- Phi: 1.00

The key value was determined by the experimenter as being the one that gave the most realistic looking brightness of the rendering. The phi value was the standard value being used for this TMO by *Luminance HDR*.

Before displaying the renderings on the screen, the color of the images was corrected for the display they were shown on. This had to be done in order assure that the actual colors of the renderings were correct at the display. An already existing MATLAB script took care of this process.

4.2.4 Measurement

For this experiment three independent variables were used; namely the integrator that was used to create the renderings, the light configuration that was displayed to the participants and whether the participants had a reference to the real room or not. The dependent variable of the experiment was the perceived realism of the renderings.

4.2.5 Procedure

Participants were asked to take place inside the tent and read the instruction form which can be found in appendix 10.1. Additional explanation and answers to any questions were given by the experimenter. The task of the participants was to select the rendering that they thought was most realistic. They had to do this twice: in one session they could see the real room, while in the other session they couldn’t. Each session started with two test pairs in order to see if the participants understood their task. The average time it took the participants to complete the experiment was 23 minutes. After the experiment, participants were asked to indicate where in the room they paid the most attention to. It was also possible for the participants to give feedback to the experimenter.

4.3 Results

The actual data of the experiment were just ones and zeros that indicated which rendering was chosen by the participants. These data were used to construct a frequency matrix having in each cell the number of times the rendering in the column head was preferred over the rendering in the row head. Such a matrix was constructed for every light scene and for each session of the experiment separately. So in total 12 of these matrices were constructed. These matrices already gave an indication of which rendering was preferred by the participants. It turned out that the time_sppm integrator was preferred in 10 of the 12 different settings. The difference with the other integrators was in some cases not very big. In order to evaluate which differences were significant, the matrices were converted to quality scores with error bars.

The conversion was done using a method described by Montag (Montag, 2006). The values in the frequency matrix were first converted into proportions by dividing every cell by N, where N was the total number of observations in the frequency matrix. These proportions were then converted into normal deviates and the averages over the columns created the quality scores for the stimuli. The standard deviation was calculated with the following equation:

$$\sigma_{obs} = b_1(n - b_2)^{b_3}(N - b_4)^{b_5}$$

where n is the number of stimuli and b_1 = 1.76, b_2 = -3.08, b_3 = -0.613, b_4 = 2.55 and b_5 = -0.491. This equation was derived using data from more than 10,000 simulated experiments.

The results of the Montag method can be represented in a graph, of which a typical example is given in Figure 15. The example given here corresponds to the results of both sessions of the experiment for the light setting in which two Savios were switched on. The graphs for the quality scores of the remaining five light conditions can be found in appendix 10.3.

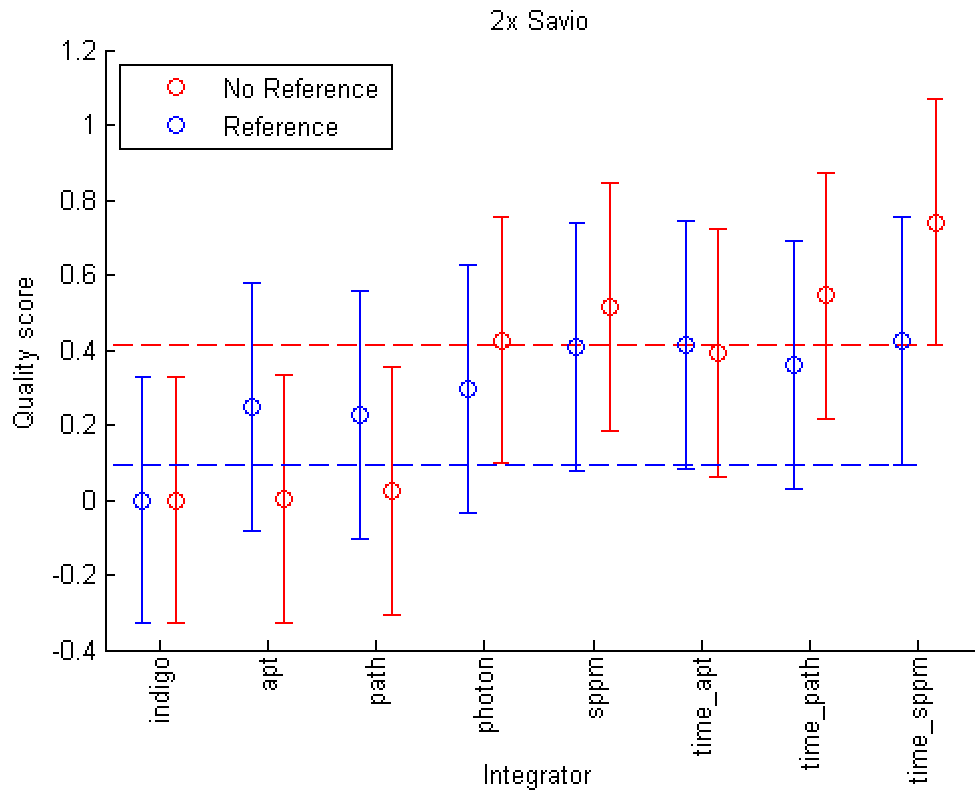


Figure 15: Quality scores of the renderings of the 2x Savio light setting. This figure shows that the realism increases as render time increases. Also the SPPM integrator is preferred over all other integrators when all the settings are the same and over all integrators when the render time is the same. This holds when participants had a reference and when they didn't.

All graphs in the appendix have the integrators on the x-axis ordered in the same way, and show a similar increasing quality trend for both parts of the experiment. In other words, the order in realism over the different integrators seems to be consistent over the various light conditions. In addition, all graphs confirm that the realism of the renderings increases for longer rendering times. Apparently, the extra computational time spent on the renderings pays off in the form of extra realism, which is of course in accordance to our expectations.

The increase in quality score over time per integrator is further illustrated in Figure 16. This graph shows the quality scores of the renderings averaged over all light conditions at two different amounts of time. The numbers in the legend represent the steepness of the lines. The SPPM integrator definitely has the highest quality score, but the lowest increase in quality over time. The APT and Path Tracer integrators are rated almost equal in quality and have a very similar increase in quality as a function of rendering time.

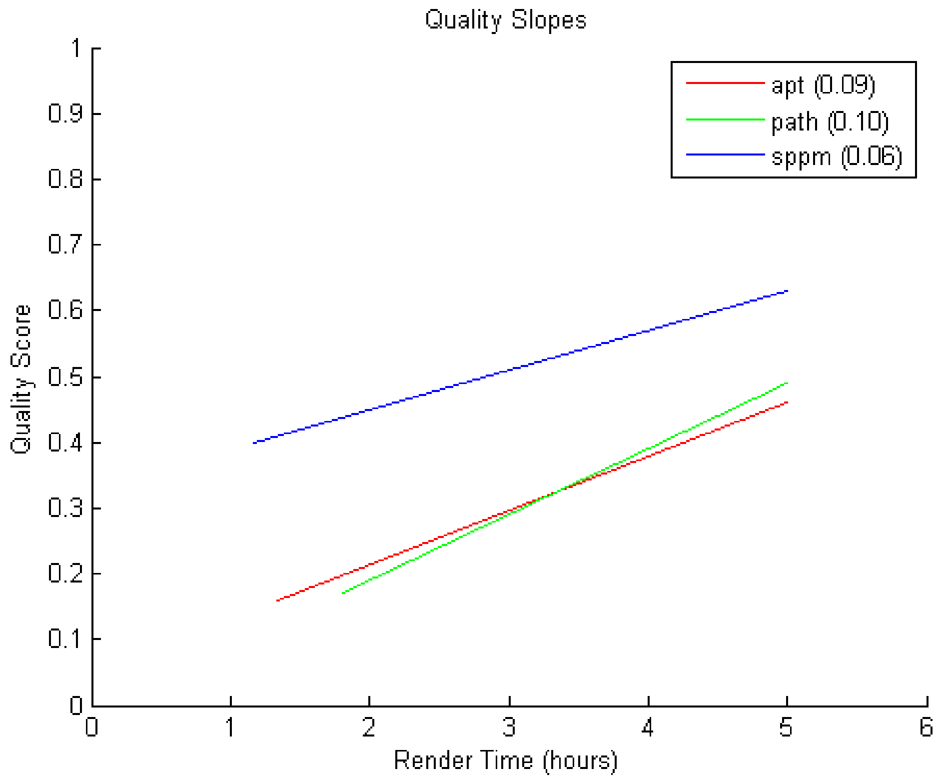


Figure 16: Quality score increase over time per integrator. SPPM had the highest quality score, but the lowest increase in quality over time. APT and Path Tracer are similar in rated quality and in increase in quality over time.

The graphs in appendix 10.3 also show that the renderings obtained with Indigo were always rated as having the lowest realism. This result may be due to the fact that these renderings didn't have the right texture for the carpet. Another difference was that the Indigo renderings were created with IES profiles, whereas all other renderings were not. An often heard comment from the participants was that the Indigo renderings were too 'bluish'.

The same graphs also show that the SPPM integrator was rated as the best one, independent on whether the same settings or the same time was used for the renderings.

To determine which of the independent variables had a significant effect on the perceived realism of the renderings an ANOVA was performed. The results for the renderings created with the path tracer were omitted from the data used for the ANOVA, because the path tracer was not used in the spot light and down light scenes. If these data would have been included, it would have resulted in incomplete data, which would have complicated the ANOVA. The ANOVA was performed with Reference, Light Scene and Integrator as independent variables and Quality Score as dependent variable, including also the 2-way interactions of the independent variables. Variables were considered significant for a p-value equal to or smaller than 0.05. Note that the quality scores used as dependent variable were an integrated z-score over the (paired-comparison) preferences of all participants. Hence, spread in the data only resulted from the different levels in the independent variables.

Analysis of Variance					
Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Reference	0.0561	1	0.0561	4.6063	0.0418
Integrator	2.5023	5	0.5005	41.1197	2.84E-11
Light Scene	0.1723	5	0.0345	2.8319	0.0369
Reference*Integrator	0.0789	5	0.0158	1.2969	0.2969
Reference*Light Scene	0.2107	5	0.0421	3.4616	0.0163
Integrator*Light Scene	0.3381	25	0.0135	1.1111	0.3972
Error	0.3043	25	0.0122		
Total	3.6626	71			

Figure 17: ANOVA of experiment 1. Reference, Integrator, Light Scene and the interaction of Reference and Light Scene has a significant effect on the realism of the renderings.

The results of the ANOVA are summarized in Figure 17, and illustrate that the type of integrator, the availability of a reference and the light setting all had a significant effect on the perceived realism of the renderings. Only the interaction between Reference and Light Setting was significant, indicating that the way the quality was affected by having a reference or not was different for the various light settings, as is also obvious from Figure 15 and the graphs in appendix 10.3.

Most important for our research is the significant effect of the integrator on the perceived realism of the renderings. Therefore, a Tukey's HSD Post Hoc test was used to further investigate which integrator was significantly better than the others on perceived realism. The result of this post hoc test is shown in Figure 18.

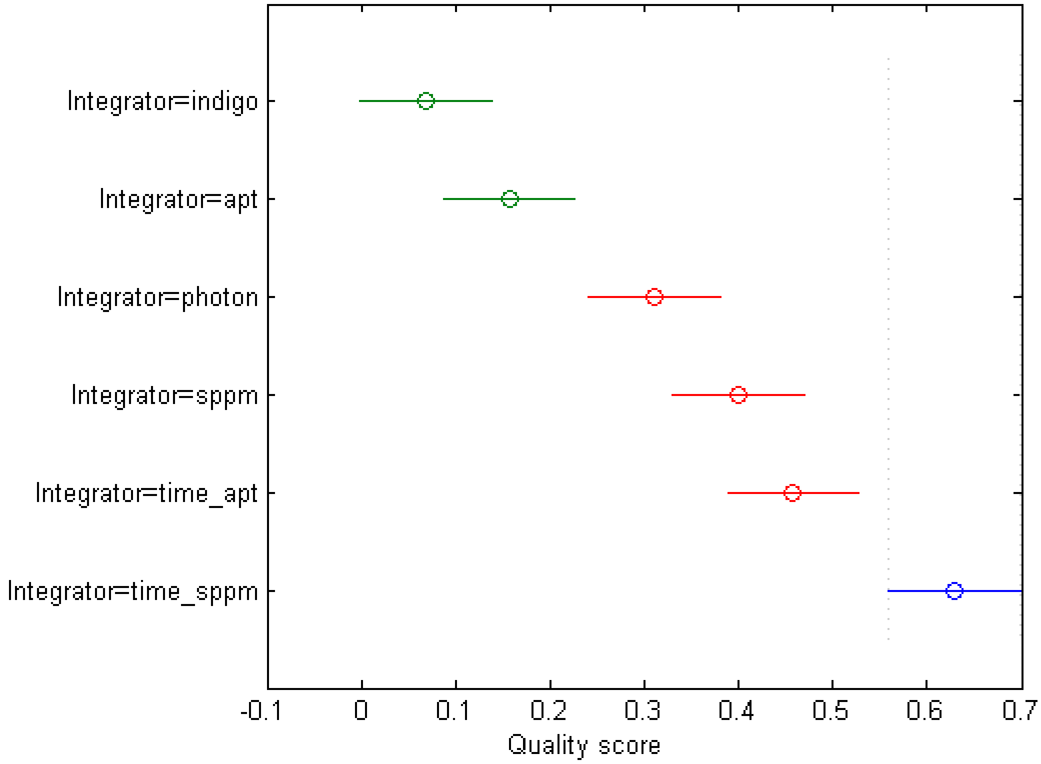


Figure 18: Tukey's HSD Post Hoc test of the ANOVA of experiment 1. This figure shows that the time_sppm integrator is significant different than the other integrators. Also Indigo and APT are significant different from the other integrators.

The Tukey's HSD Post Hoc results in Figure 18 show that the integrators can roughly be divided into three groups. In each group the means are relatively close to each other. The time_sppm integrator forms a group on its own and is significantly different from the other integrators.

4.4 Discussion

The results of the experiment show a clear conclusion with respect to the worst and the best integrator from the point of view of realism of the resulting renderings: the Indigo integrator was the worst, while the time_sppm integrator was the best. The results of the ANOVA, however, were to some extent tricky, since they were based on integrated scores over all participants. Hence, the spread resulting from differences between participants was not included in the analysis. Nonetheless, the conclusions could also been drawn from the graphs in appendix 10.3.

The observation that the Indigo renderer performed worst should be put in the right perspective. This renderer used a different 3D model of the room with a different carpet texture and used IES profiles for the light distribution in the room. As a consequence, a direct comparison with the other integrators might have been not fully fair. More research is needed to better understand whether the bad performance of the Indigo renderer is intrinsic to the renderer, or a consequence of the other differences.

The Path Tracer integrator didn't perform very well for the light configurations containing spot lights. Even for a prolonged rendering time, the resulting rendering was poor. Therefore, this integrator was omitted from the comparison with the other integrators for these light configurations. The latter complicated the analysis as the data became incomplete. However, it didn't affect the conclusions we could draw.

One of the participants spontaneously indicated that he was colorblind. Since the occurrence of color blindness was not explicitly tested with the other participants, we decided to include the data of the color blind participant in the analysis. The influence of his results on the overall conclusions was tested by repeating the analysis without the data of the color blind participant. It turned out that the results didn't change significantly.

4.5 Conclusion

Based on the results of the first experiment it can be concluded that the integrator has a significant effect on the perceived realism of a rendering. The participants prefer the Stochastic Progressive Photon Mapping integrator in most of the different light configurations. They prefer this integrator over the others when the same settings are used and when they are run for the same amount of time. It does not matter if people physically are in the scene or not. The perceived realism of this integrator is significantly better than of any other integrator, as shown in a Tukey's HSD Post Hoc test.

The SPPM integrator, however, appears to have the slowest increase in quality over time. The APT and Path Tracer increase in quality faster and might result in better quality than the SPPM when run for longer times.

5 Experiment 2

5.1 Introduction

Based on the results of the first experiment it seems that some materials and some types of light sources improve faster in realism than others. In order to investigate this further renderings of an object were created for different materials of the object and under different lighting conditions. Two different integrators were used for creating the renderings, and depending on the integrator the renderings were saved either after a fixed time interval or when a certain amount of samples per pixel was reached. The goal of the resulting experiment was to determine combinations of material and lighting that were significantly faster or slower to render and how these combinations influenced the perceived image quality.

5.2 Method

The task for the participants was a two alternative forced choice comparison. Underlying this paired comparison was a quality ruler. A stimulus rendering and a rendering from the quality ruler were shown to the participants at the same time. The participants indicated which of the two renderings they preferred in terms of image quality of the rendered object. Based on the choice the stimulus rendering moved up or down on the quality ruler with a staircase-type of algorithm. This process stopped after the participant made 5 reversals on the quality ruler. The quality level of the last four reversals was averaged and considered as the quality score for the stimulus rendering. The images used for the ruler are explained in section 5.2.3.

The participants had to complete a staircase for each stimulus rendering. These staircases were intermingled to exclude any order or fatigue effects. Also the starting point on the staircase and whether the stimulus rendering was displayed at the left or right side of the display was selected randomly.

5.2.1 Participants

A total of 20 participants voluntarily signed up for this experiment. 7 of the 20 participants were female. The participants were not colorblind. Most of the participants were students and the average age of the participants was 24.75 with ages ranging from 22 to 30 years.

5.2.2 Equipment and Setup

The experiment took place in a standard experimental room. The room was equipped with a computer and a keyboard to control the program that was used. The display used was a Philips 42” full high definition display. The participant had to take place right in front of the display at a distance of about 1 meter.

5.2.3 Stimuli

Several renderings, all based on the same shape of teapot on the same background, were presented to the participants. An example of such a rendering is given in Figure 19. The dimensions of the bounding box of the teapot are 40x25x20 centimeter.

Figure 19: A standard teapot as object used in this experiment.

Figure 20 gives an overview of the scene of which the renderings were created. The scene was created using Google SketchUp 8.0[3]. It shows which luminaires were used to illuminate the rendered object. To create the actual renderings the scene was closed by putting a box around the scene.

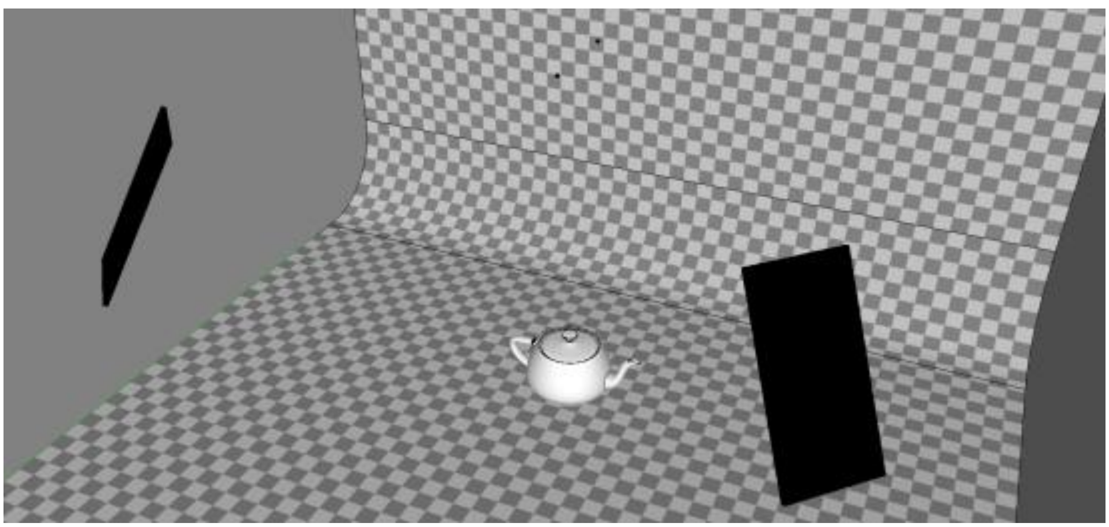


Figure 20: Scene in which the teapot was placed. This figure also shows the two area luminaires and the two spot lights that were in the scene.

The luminaires in the scene consisted of:

- Two spotlights on both sides of the teapot at a height of 1 meter. Both spotlights had a color temperature of 5000K.
- Two planar luminaires hanging at the same height as the spotlights, which were rotated over 45 degrees to the left and the right, and positioned 62.5 centimeter away from the middle of the teapot. The left luminaire had a color temperature of 2500K and the right a color temperature of 5000K. The dimensions of the luminaires were 40x20 centimeters. This light setting is further referred to as ‘warmcold’.

[3] Google SketchUp: http://sketchup.google.com/intl/en/ (Accessed: June 18, 2012)

The renderings were created using two different integrators:

- Indigo with the bidirectional path tracer with metropolis light transport enabled.
- Mitsuba with the stochastic progressive photon mapping algorithm.

These two integrators were chosen because, on the one hand, the bidirectional path tracer with metropolis light transport enabled is commonly used, and on the other hand, the stochastic progressive photon mapping algorithm was rated as the most realistic in the previous experiment. Another reason for selecting these two integrators was that one integrator was path tracing based, while the other was photon mapping based.

The teapot was rendered with the following material models:

- Diffuse material: A diffuse material with RGB color 220, 220, 220.
- Glossy material: A Phong material with an exponent value of 10000 and RGB color 220, 220, 220.
- Glass: A not solid glass materials with IOR 1.33 and no absorption.

The renderings created with Indigo were saved after they had reached the following number of samples per pixel: 25, 50, 100, 200, 500 and 1000. Unfortunately, there was no similar measure as samples per pixel when using the stochastic progressive photon mapping algorithm in Mitsuba. Therefore the renderings that were shown to the participants were saved after the following rendering times: 5, 10, 20, 30, 45 and 60 minutes of rendering. Examples of stimuli can be found in appendix 10.5, where the stimuli of the Indigo 'spot glossy' are shown.

Using two different light settings, two different integrators, three different materials and six different quality levels resulted in 72 renderings in total. As a consequence, the participants had to complete 72 staircases in order to determine the quality of the renderings.

Since no quality ruler for rendered images was available, we used one of the renderings of the set described above as the reference for the quality ruler, and compared the other renderings against this reference. The rendering selected for the quality ruler was the teapot with a glossy material in the 'warmcold' light setting rendered with Indigo. Since it was desirable that the ruler spanned a wider variety of quality than the stimuli renderings, more images were generated for the reference rendering. More particularly, for the glossy teapot in the 'warmcold' light setting Indigo rendered images with the following number of samples per pixel: 5, 10, 15, 20, 25, 30, 40, 60, 80, 100, 133, 166, 200, 250, 300, 400, 600, 800, 1000 and 2000. All images were rendered using a resolution of 620x350 pixels.

The renderings were created on a computer with an Intel Xeon W3680 processor running at 3.33GHz. Rendering times were recorded using this computer with four of the six cores dedicated to the rendering process.

5.2.4 Measurement

In this experiment there were three independent variables; namely the material of the rendered object, the lighting condition the object is exposed to and the integrator that was used to create the rendering.

The dependent variable of the experiment was the quality score resulting from the staircase along the quality ruler.

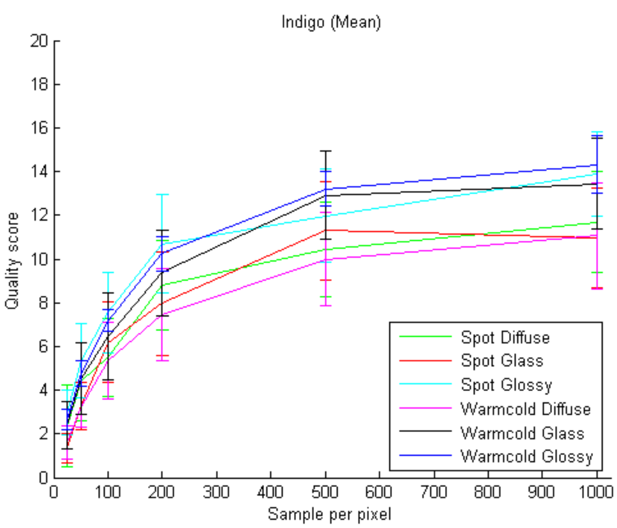
5.2.5 Procedure

The participants were asked to sit in front of the display and to sign an informed consent form, which can be found in appendix 10.4. The experimenter then verbally explained the experiment and answered questions if the participant had any. As soon as everything was clear, the participant started the experiment and had to indicate for each pair of renderings which one he preferred in terms of quality of the teapot.

5.3 Results

The experiment resulted in 72 quality values per participant that may be grouped over light condition, material type and integrator type. For two participants, however, the resulting quality values were independent of the rendering time, while this was clearly not the case for all other participants. The two participants either always gave a high or always a low quality score, depending on their preference for the type of material and lighting condition. This observation indicated that they didn't understand the task correctly. As a consequence, their results were omitted from the rest of the analysis.

The quality scores averaged over all remaining participants are given as a function of rendering time in Figure 21. The left graph shows the results for the Indigo renderer, whereas the right graph shows the results of the Mitsuba renderer. Each curve represents a given combination of material of the teapot and lighting condition. The error bars represent the 95% confidence interval on the averaged quality score. The Indigo renderings clearly progress in quality when the rendering time increases. However, for the Mitsuba renderer the trend is completely different. For some combinations of material and light condition the image quality increases with extra rendering time, while for most others it doesn't. In addition, the Indigo renderings start at a low quality for short rendering times, whereas the Mitsuba renderings start already at a higher quality level.



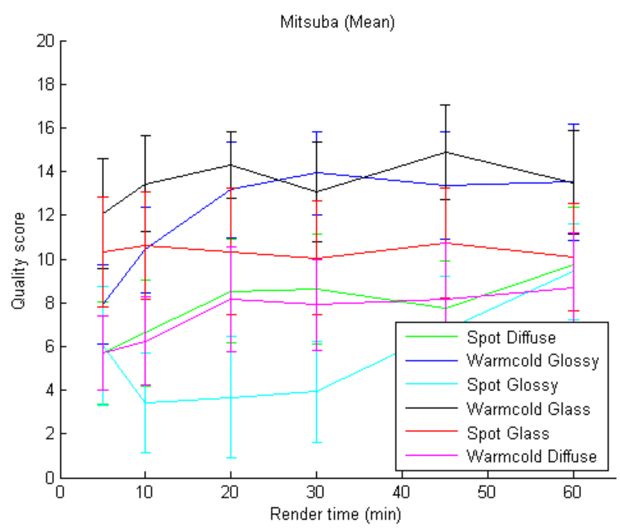


Figure 21: Mean quality scores for Indigo (left) and Mitsuba (right). The increase in quality of the Indigo renderings was as expected while that of Mitsuba not. Mitsuba renderings are rated higher in the beginning of the rendering process but don't increase much over time.

The constant quality as a function of rendering time for some of the materials with the Mitsuba renderer might be caused by artifacts introduced by the integrator. These artifacts may be different depending on the integrator, as is illustrated in Figure 22. The artifacts in the Mitsuba renderings are a bit 'lumpy' and the participants may think that this is part of the material type of the teapot. The artifacts of the Indigo renderings appear as pixel noise. As such, participants might not recognize the artifacts of the Mitsuba renderings as noise, and so, consider the Mitsuba renderings less noisy than the Indigo renderings, while both types of renderings are still far from perfect.

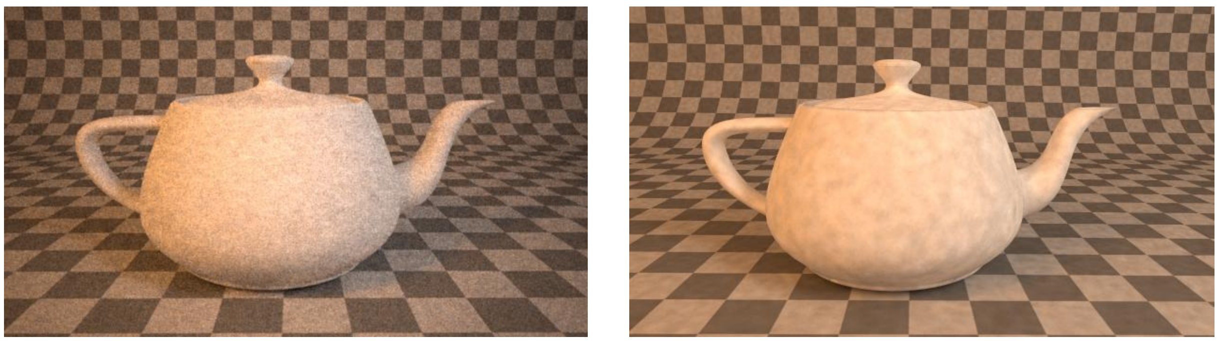


Figure 22: Artifact comparison between Indigo (left) and Mitsuba (right). The artifact of Indigo appears as pixel noise while the artifact of Mitsuba appears as a lumpy texture.

An ANCOVA analysis was performed to see if there is a significant difference between the different material and lighting combinations and the quality score for the Indigo renderings. In this analysis the log of samples per pixel was the predictor variable, the quality score was the response variable and the combinations of material and lighting was used as grouping variable. It turned out that the material/lighting combination does have a significant effect on the quality score, with a F value of 8.69. While this is statistically significant at the 0.05 level, the effect is much weaker than the significant effect of log samples per pixel (F=736.27). Perhaps due to this weakness, despite the overall significance of the grouping variable, a Tukey Post Hoc test was unable to reveal significant differences between the different groups.

Since some of the stimuli renderings were part of the ruler images we can also plot the quality values of all stimuli against the quality values of the ruler images. Doing so allows us to conclude which combination of material and lighting condition increases faster in quality than the 'Warmcold Glossy' setting. These plots for both the Indigo and Mitsuba renderer are given in Figure 23. For the Indigo renderings it is clear that all scenes increase in quality in a similar way. In addition all scenes, except for the 'Spot Glossy' scene, are of lower quality than the 'Warmcold Glossy' scene. This result is counter-intuitive. We had expected to find a steeper increase in quality for the simpler scenes compared to the more complex scenes. However, the relatively simple diffuse scenes have the slowest increase in quality. On the contrary, the 'Spot Glossy' scene, being more difficult because of its smaller light sources than the 'Warmcold' scenes, has nonetheless a same increase in quality.

For the Mitsuba renderings the conclusions are the same as based on Figure 21. When compared to the increase in quality of the 'Warmcold Glossy' scene, all other renderings fluctuate a bit in quality, but mainly stay constant. In other words, the renderings of all scenes have a more or less constant quality level with increasing rendering time.

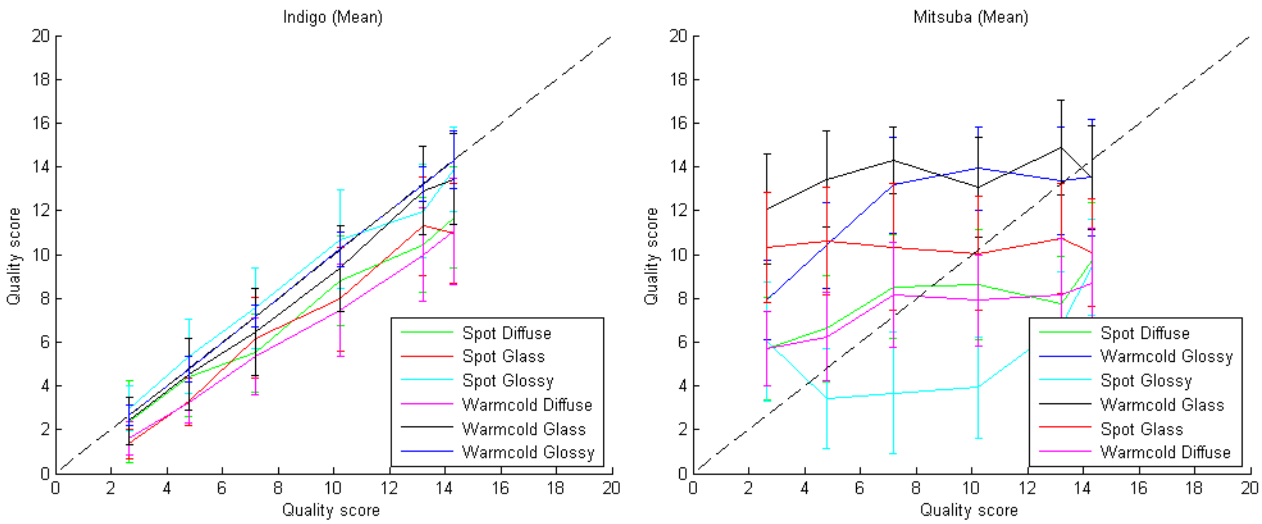


Figure 23: Mean quality scores plotted against the quality scores for Indigo 'Warmcold Glossy' for Indigo (left) and Mitsuba (right). This figure shows that almost all Indigo scenes increase slower in quality than the 'Warmcold Glossy' scene. There is not much change in the graph of Mitsuba scenes when compared to Figure 21.

Visually inspecting the course of the staircases through the quality ruler for all participants shows that the results are not very consistent. The latter is illustrated in Figure 24, where the left graph shows the staircases in quality of the Indigo rendering (100 spp) for the 'Spot Glossy' scene, while the right graph gives the staircases in quality of the Indigo rendering (100 ssp) for the 'Warmcold Glossy' scene. In both graphs the red horizontal lines represented the converged values of quality for each of the participants, whereas the black horizontal line represents the average of the red lines, i.e. over all participants. Clearly the staircases for the 'Spot Glossy' scene jump up and down in quality with a wide spread, whereas the staircases for the 'Warmcold Glossy' scene seem to converge more or less to the same quality level.

The stimuli in both graphs of Figure 24 represent the same material, but the stimuli in the right graph correspond to the same light condition as the ruler images, while this is not the case for the stimuli in the left graph. Hence, when the rendered light condition (or material) differs between stimuli and reference images, the staircases are more spread across the quality range. This indicates that comparing the quality of different material or lighting conditions is a difficult task for the participants.

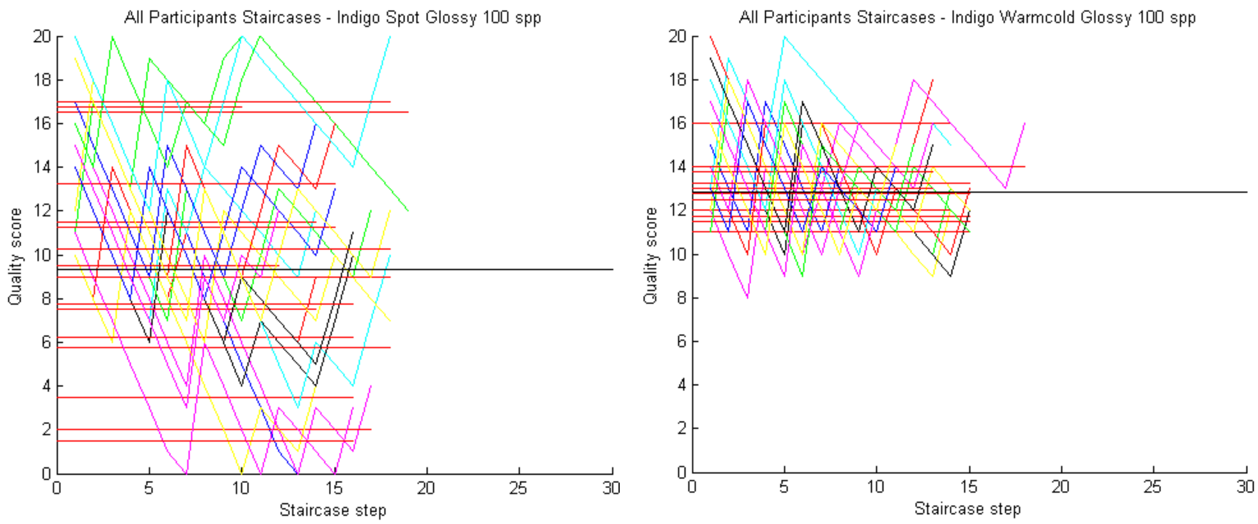


Figure 24: Staircase comparison between Indigo Spot Glossy 100 spp and Indigo Warmcold Glossy 100 spp. When the stimulus is the same as the ruler the staircase converge. When the material and or the lighting condition is different than the ruler, the staircases are spread over the possible quality levels.

Some possible solutions towards better converging staircases were tried out. The left graph of Figure 24 clearly shows that a lot of staircases hit the bottom and top level of the quality ruler. This observation suggested that the quality range of the reference renderings was, according to the participants, not wide enough for covering the stimuli renderings. To improve the accuracy of the data, we removed the staircases from the data set that hit the top or bottom quality level. This action, however, did not reduce the spread in quality values across the staircases. A second possible solution was to reduce the length of the staircases.

At the end of each staircase the difference in quality between the ruler images and the stimuli images became smaller and smaller and thus harder for the participants to make a decision. The first part of the staircases, on the other hand, contained the bigger steps in quality, and therefore, resulted in more reliable judgments. We tried to only look to the first part of the staircases for calculating the quality values, but this didn't result in more converging quality values.

As some of the staircases were looking good, it wasn't the methodology used in this experiment that caused the weird results. The general thought of the experimenters was that the participants had too much trouble in comparing the quality of different materials and different lighting conditions. In addition, not having a reference to a 'perfect' rendered version of the object with the same material might made it difficult for the participants. It was our hypothesis that having a reference to a 'perfect' rendered version of the object may lead to better results as participants can better look through the different artifacts introduced by the two integrators.

5.3.1 Follow-up Experiment

This hypothesis was tested in a small follow-up experiment. In the new experimental set-up not only a ruler image and a stimulus rendering were shown to the participant, but also a very high quality version of the ruler image and the stimulus rendering. The participants now had to indicate which rendering had the smallest difference with the very high quality version of the corresponding rendering. This also ruled out the possibility that participants chose the rendering they personally preferred based on the material.

This follow-up experiment was conducted with 4 participants and with only half of the stimuli of the real experiment. The results for the Indigo rendering (100 spp) of the ‘Spot Glossy’ scene are displayed in Figure 25. Comparing this graph to the left graph of Figure 24 shows that the staircases are now more consistent, and less spread over the quality range.

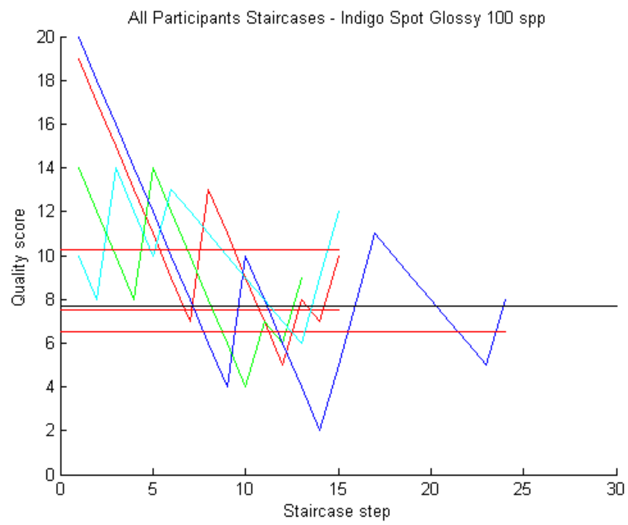


Figure 25: Staircases of the Indigo Spot Glossy 100 spp rendering. When compared to Figure 24 it can be seen that the staircases are more converging.

The mean quality scores resulting from the follow-up experiment are plotted in Figure 26. The results for the Indigo renderer are very similar to the results shown in Figure 21. Interestingly the Indigo renderings end up at a higher quality score, but this may be a consequence of using only a limited number of subjects. The curves for the Mitsuba renderings don’t change very much; they are still widely spread over the quality range. However, both the ‘Spot Diffuse’ scene and the ‘Warmcold Glossy’ scene begin to show the same shape as obtained with the Indigo renderer, which suggests that Mitsuba renderings also improve as rendering time increases.

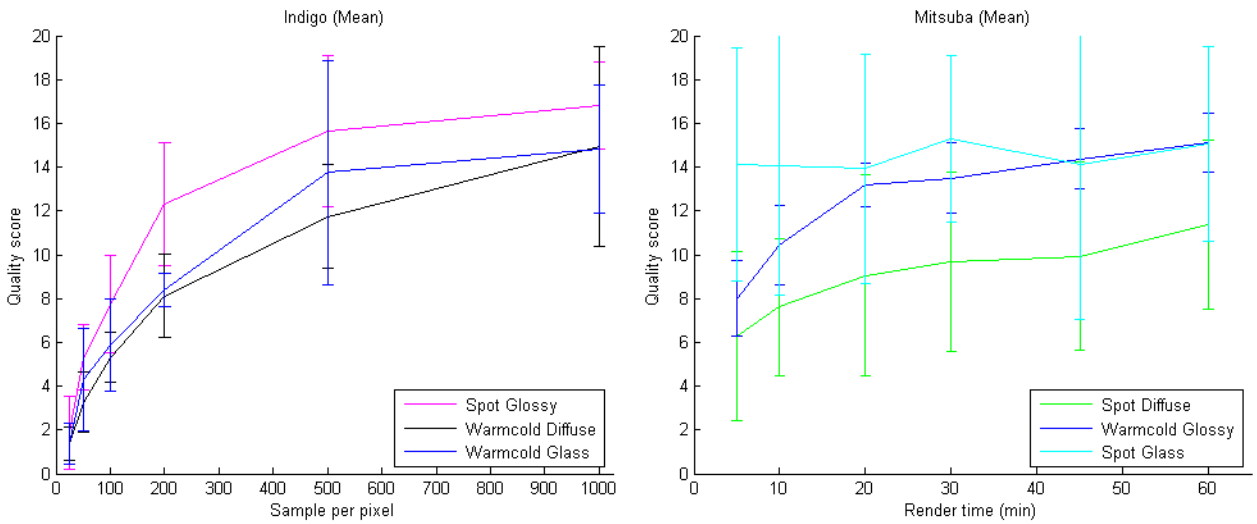


Figure 26: Mean staircase scores for Indigo (left) and Mitsuba (right) for the follow-up study. Quality scores for the Indigo scenes are higher when compared to Figure 21. The curves of the Mitsuba scenes seem to more or less get the same shape as the curves of the Indigo scenes.

5.4 Discussion

It appeared that the hardest part of this experiment was to compare quality across different material types and different lighting types. An attempt to make the task easier we used a different interface in a follow-up experiment. This already improved the results. Yet another possible interface may be one, in which renderings of a given material type have to be rated along a quality scale ranging from 0 to 100. Renderings of very low and very high quality of the same material are then placed as a reference at both ends of the scale. This interface also rules out the possibility that participants choose the rendering they personally prefer based on the material. A disadvantage, however, is that people may try to space the renderings equally on the scale instead of giving them an actual score. Another disadvantage is that scores of renderings of different materials are not comparable due to the fact that these renderings are scored on a different scale. It seems desired to use the same renderer for the ruler images. In this case Indigo renderings were used as ruler images, and that didn't lead to stable results for stimuli rendered with the Mitsuba software. Using Mitsuba renderings as ruler images might lead to more stable results for Mitsuba stimuli, however then the results for the Indigo stimuli might be not so stable.

The scene used in the experiment was also not optimal. The scene was surrounded by a box, made of a diffuse material, so that the teapot was not only lit from the luminaires but also with the reflective rays from the box. In addition, the spot luminaires were not real point luminaires. Although they were much smaller than the big planer luminaires, they were not real point light sources.

The results of this experiment are not directly applicable to real scenes. Real scenes most likely contain lots of different materials. This experiment focused on limited material types. However, the results of this experiment may give an indication on which material types increase the rendering time of a rendering.

5.5 Conclusion

It turned out that the material/lighting combination does have significant effect on the quality score. This effect is rather weak when compared to the effect of the render time on the quality score. These statements at least holds for the Indigo renderings. It was expected to see a bigger difference between the integrators and some results were surprising as simple diffuse scenes performing worse in quality at a given rendering time than more complex scenes, such as glass.

An additional conclusion is that renderings created with Mitsuba seem to be of higher quality at the beginning of the rendering process than renderings created with Indigo. However, the quality of the Mitsuba renderings don’t seem to improve much with increased rendering time. Indigo renderings start with a low quality but that quality rapidly improves over time.

Another interesting finding of this experiment is that people find it hard to compare quality across different materials and lighting conditions. It helps if people have a reference to a perfect version of the renderings they have to judge.

6 Conclusions and Discussions

By comparing the renderings created with different integrators of Mitsuba it can be concluded that the Stochastic Progressive Photon Mapper is rated as the integrator that produces the most realistic renderings. As this integrator is fundamentally different from the integrators used in most research, it might be of interest to evaluate if other photon based integrators also score higher in realism compared to path tracer based integrators. A side note with this experiment is that the scene used for the experiment is an indoor scene with a lot of relatively simple materials in it. The results might be very different for outdoor scenes, scenes with daylight or scenes with more complex materials, such as glass and cloth.

Although the Stochastic Progressive Photon Mapper of Mitsuba produces higher quality renderings right from the start, they don't quickly improve over time. Indigo renderings start at a lower quality, but improve quickly over time. This might be explained by the artifacts introduced by the two integrators. Mitsuba's SPPM introduces some kind of lumpy artifacts and Indigo creates actual pixel noise. Although no tests have been done on how this influences the realism of renderings it makes sense that pixel noise is not perceived as realistic. Therefore renderings that are rendered for short amount of time with Mitsuba's SPPM are rated more realistic than renderings rendered for a short amount of time with Indigo.

The results from the second experiment show that a more difficult scene doesn't necessarily mean an increase in rendering time. Almost all different scenes increase in quality at about the same speed when the rendering time is increased.

Another fact that was discovered with this research is that apparently people find it hard to compare the quality of renderings of different materials and the quality of renderings of the same material under different lighting conditions. The same material under different lighting conditions may be perceived as being a different material. Some difficulty in comparing the material quality may be introduced by the artifacts generated by the two integrators. In addition, personal preference of the participants for a given material look may play a role in rating the quality of an image. These difficulties may be ruled out using a different interface for the experiment. Such a different interface basically changes the task for the participants. Showing a perfect version of the stimuli and their reference renderings in a quality ruler set-up helps the participants in their decision and leads to more stable quality results.

It's hard to use these results in real world scenes. Real world scenes usually have dozens of different materials and various light sources in it. However, diffuse, glossy and glass materials using spotlights and diffuse lights shouldn't negatively influence rendering times.

7 Future Work

More research is needed to really underpin the conclusion towards the best integrator and rendering software. One extension of the current research may be a study to evaluate if a photon based integrator also results in more realistic renderings than a path based integrator in the same amount of time when different rendering software is used. Of course, for such an evaluation both integrators should be supported by the same rendering software. It may also be useful to compare other rendering software that has implemented the SPPM algorithm and to evaluate how those renderings compare to the Mitsuba renderings. Another extension may be to check if the SPPM still outperforms path tracer based integrators on the realism versus effort rating once the bi-directional path tracer and metropolis light transport are implemented in Mitsuba. An final extension to this research may be the evaluation the SPPM algorithm on outdoor scenes, scenes which have daylight in it and scenes with more complex materials in it.

At this point of time it is not recommended to replace Indigo with Mitsuba. Although people might prefer the renderings created by Mitsuba, the user friendliness is way below that of Indigo. Mitsuba also lacks some parameters to specifically model a material. More research in the render qualities and the usage of Mitsuba is needed before it can replace Indigo.

It may be very worthwhile to focus on the question why the renderings rendered with SPPM are preferred over other renderings. If that aspect is better understood, the knowledge may be used to create even more realistic renderings.

Finally more knowledge on the relation between material type and rendering time needed to realistically render the material would be highly desirable. To obtain this knowledge, however, a good experimental interface that supports people in comparing material quality across different materials is needed. The interfaces suggested in this report might be helpful. The interface used in the follow-up experiment after the second experiment already proved to be useful.

8 Abbreviations

3D	Three Dimensional
APT	Adjoint Particle Tracer
COLLADA	COLLAborative Design Activity
HDR	High Dynamic Range
HVS	Human Visual System
IES	Illuminating Engineering Society
JND	Just Notable Difference
LDR	Low Dynamic Range
MLT	Metropolis Light Transport
PBRT	Physically Based Render Technique
PPM	Progressive Photon Mapping
SPP	Sample Per Pixel
SPPM	Stochastic Progressive Photon Mapping
TMO	Tone Mapping Operator
VP	Virtual Prototyping
XML	Extensible Markup Language

9 References

Agusanto, K., Li, L., Chuangui, Z., & Sing, N. (2003). Photorealistic Rendering for Augmented Reality using Environment Illumination. *IEEE and ACM International Symposium on Mixed and Augmented*, 208-216.

Bazley, C. (n.d.). *Design of GL module*. Retrieved June 18, 2012, from A 3-D Graphics Library for ARM Systems: http://www.starfighter.acornarcade.com/mysite/articles/P3Report/Chapter5.html

Christensen, P. H. (2003). Adjoints and Importance in Rendering: An Overview. *IEEE Transactions on Visualization and Computer Graphics*, 329-340.

Cline, D., & Egbert, P. (2005). *A Practical Introduction to Metropolis Light Transport.* Technical report: Brigham Young University.

Crane, K. (n.d.). Bias in Rendering.

David, H. (1988). *The Method of Paired Comparisons* (2nd ed.). London: Oxford University Press.

Drago, F., Myszkowski, K., Annen, T., & Chiba, N. (2003). Adaptive Logarithmic Mapping For Displaying High Contrast Scenes. *Proceedings of Eurographics*, 419-426.

Ehrenstein, W. H., & Ehrenstein, A. (1999). Psychophysical Methods. (U. Windhorst, & H. Johansson, Eds.) *Modern Techniques in Neuroscience Research*, 1211-1241.

Ferwerda, J. (2003). Three Varieties of Realism in Computer Graphics. *Human Vision and Electronic Imaging VIII*, 290-297.

Hachisuka, T., & Jensen, H. W. (2009). Stochastic Progressive Photon Mapping. *SIGGRAPH Asia '09*, 1-8.

Hachisuka, T., Ogaki, S., & Jensen, H. W. (2008). Progressive Photon Mapping. *SIGGRAPH Asia '08*, 1-8.

Jensen, H. W., & Christensen, P. (2007). High Quality Rendering Using Ray Tracing and Photon Mapping. *ACM SIGGRAPH*.

Kaernback, C. (1991). Simple adaptive testing with the weighted up-down. *Perception and Psychophysics*, 49:227-229.

Kajiya, J. T. (1986). The Rendering Equation. *Computer Graphics (SIGGRAPH '86 Proceedings)*, 143-150.

Keelan, B. W., & Urabe, H. (2004). ISO 20462, A psychophysical image quality measurement standard. *Image Quality and system Performance*, 181-189.

Keller, A. (1997). Instant Radiosity. *SIGGRAPH*, 49-56.

Kinkelin, M., & Liensberger, C. (2008). Instant Radiosity: An Approach for Real-Time Global Illumination.

Matsuo, T., & Fujimoto, T. (2011). *E-Activity and Intelligent Web Construction: Effects of Social Design.* IGI Global.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21:1087-1092.

Montag, E. D. (2006). Empirical formula for creating error bars for the method of paired comparison. *J Electron Imaging*, 15:222-230.

Morley, R. K., Boulos, S., Johnson, J., Edwards, D., & Shirley, P. (2006). Image Synthesis using Adjoint Photons. *Graphics Interface*, 179-186.

Pharr, M., & Humphreys, G. (2010). *Physically Based Rendering: From Theory to Implementation, second edition.* Morgan Kaufmann Publishers.

Reinhard, E., Stark, M., Shirly, P., & Ferwerda, J. (2002). Photographic Tone Reproduction for Digital Images. *ACM Transactions on Graphics*, 21(3):267-276.

Roy, G. G. (2000). A Comparative Study of Lighting Simulation Packages Suitable for use in Architectural Design. Perth: Murdoch University.

Salters, B., & Seuntiens, P. (2011). A Comparison of Perceived Lighting Characteristics in Simulations Versus Real-Life Setup. *Society of Photo-Optical Instrumentation Engineers*.

Salters, B., Murdoch, M., Sekulovksi, D., Chen, S.-H., & Seuntiens, P. (2012). An Evaluation of Different Setups for Simulating Lighting Characteristics. *Proc. SPIE 8291, 82911M*.

Shreiner, D. (2009). *OpenGL Programming Guide.* Addison-Wesley.

Veach, E., & Guibas, L. J. (1997). Metropolis Light Transport. *SIGGRAPH '97*, 65-76.

Villa, C., & Labayrade, R. (2010). Psychovisual Assessment of Tone-Mapping Operators for Global Appearance and Colour Reproduction. *CGIV*, 189-196.

Wang, G. G. (2002). Definition and Review of Virtual Prototyping. *ASME Journal of Computing and Information Science in Engineering*, 232-236.

Wikipedia. (n.d.). *Diffuse reflection*. Retrieved December 15, 2011, from Wikipedia: http://en.wikipedia.org/wiki/Diffuse_reflection

Wikipedia. (n.d.). *Ray tracing (graphics)*. Retrieved December 15, 2011, from Wikipedia: http://en.wikipedia.org/wiki/Ray_tracing_(graphics)

YafaRay. (n.d.). *Render Settings*. Retrieved June 18, 2012, from YafaRay: http://www.yafaray.org/documentation/userguide/rendersettings#adaptivesampling

Yoshida, A., Blanz, V., Myszkowski, K., & Seidel, H.-P. (2005). Perceptual Evaluation of Tone Mapping Operators with Real-World Scenes. *Human Vision and Electronic Imaging X*, 192-203.

10 Appendices

10.1 Appendix A1: Information and Informed Consent Form Experiment 1

Information & Informed consent form

Participant number: _____

Thank you for participating in this study!

In this experiment you are going to judge image quality. Each time, two images are shown on the screen next to you. The images resemble the room you see in front of you.

You have to select the image that you think is the most **realistic** looking. Selecting an image is done by pressing either the **arrow left** button for the left image or the **arrow right** button for the right image. You always have to select an image and you can take as much time for it as you need. There is no right or wrong in choosing an image.

You will judge images of different light conditions. In will take some time to adjust the lights in the room to the new light conditions. Therefore, if a new light condition will start, the screen will stay black for five seconds.

The experiment consists of two parts. In one part of the experiment you can look in to the real room while evaluating the images and in the other part you cannot. Before the start of each part, there are two image pairs that are there for training purposes. Make sure, in the first part, that during these two pairs your field of view matches with what you see in the images and make sure, in the second part, that you sit comfortable in front of the screen.

Participation in this study is strictly voluntary. You may choose to stop participating and withdraw from this study at any time, without providing reason. Please tell the researcher if you wish to stop.

All results will be handled confidentially, and your name will not be included or in any other way associated with the data collected in this study.

Consent:

I acknowledge that I have read and understand this consent form, and I voluntarily participate in this research study. I understand that I may stop participation at any time, without providing a reason. If I have any questions or concerns now or in the future regarding the study, I may ask the researcher's supervisor, Michael Murdoch.

Participants name:	Participants signature:	Date:
______________________________________	_________________________	______________

10.2 Appendix A2: Lightlab Renderings per Integrator for 6x Savio Light Setting

Indigo

Photon Mapper

APT

Time APT

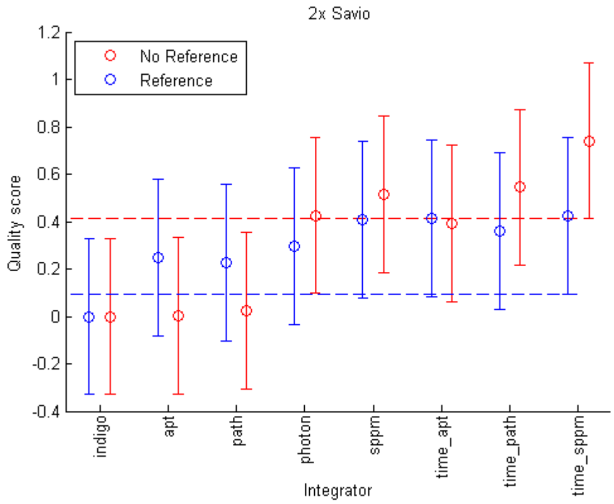
Path Tracer

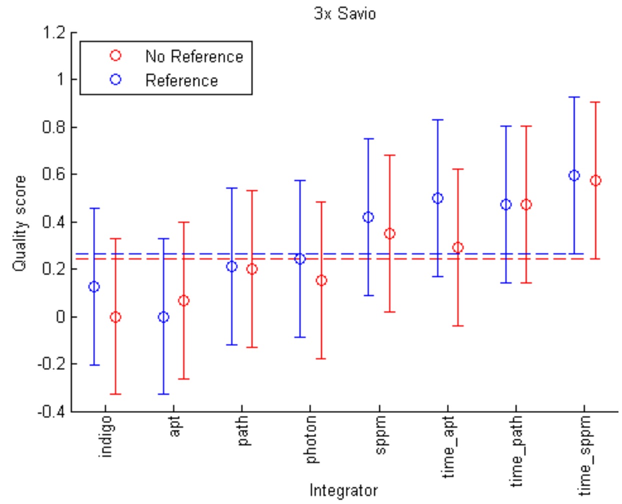
Time Path Tracer

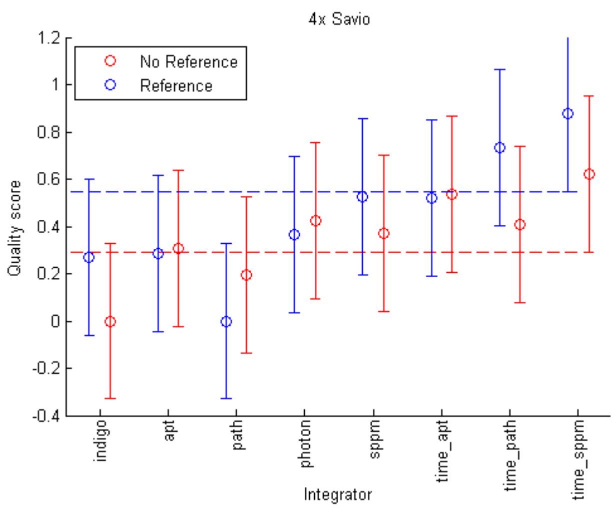
SPPM

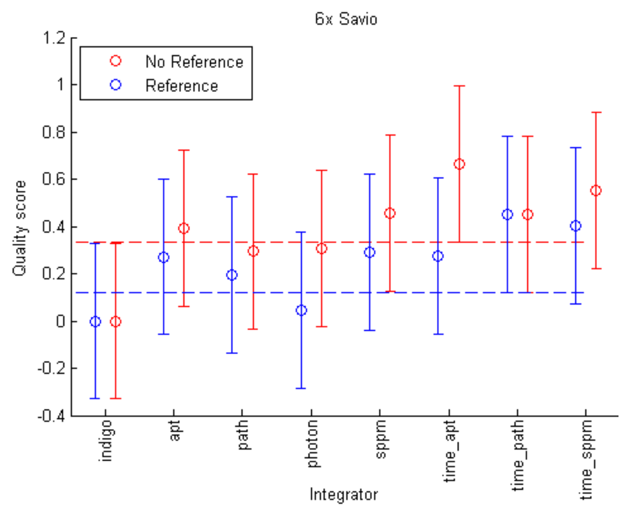
Time SPPM

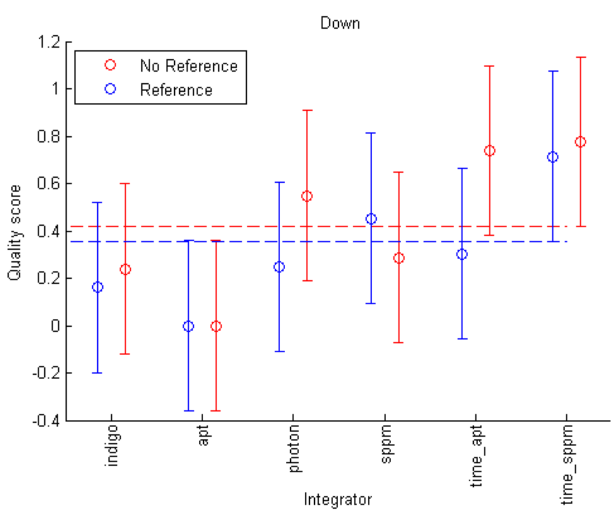
10.3 Appendix A3: Montag Plots Experiment 1

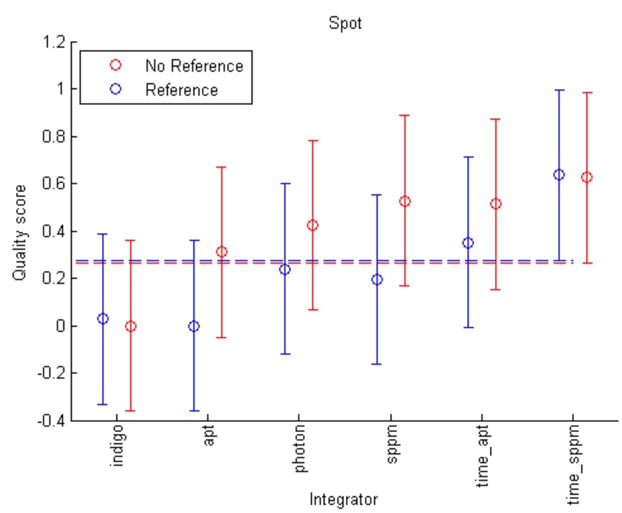


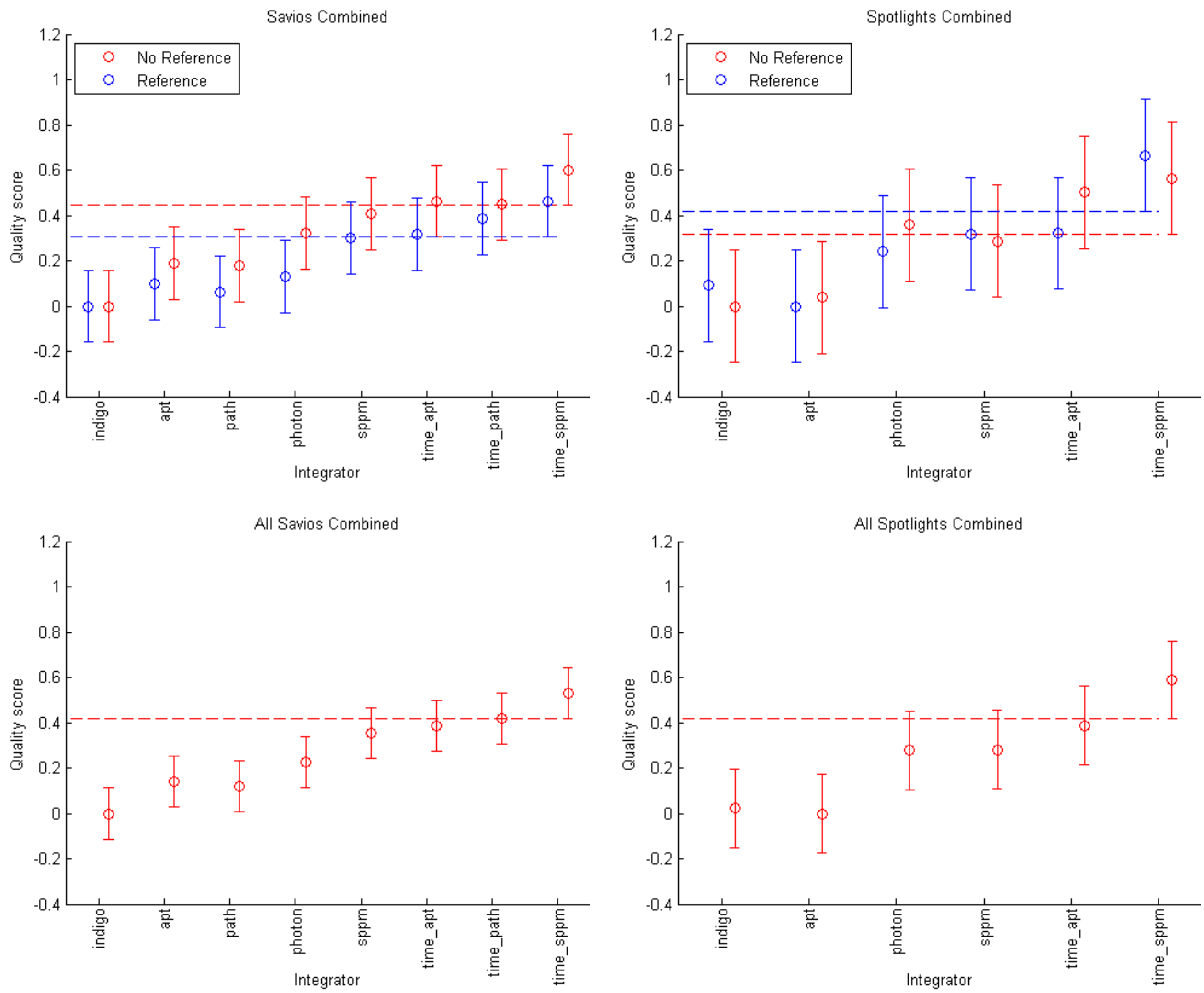











Savios Combined
No Reference
Reference
Quality score
Integrator
indigo
apt
path
photon
sppm
time_apt
time_path
time_sppm
Spotlights Combined
No Reference
Reference
Quality score
Integrator
All Savios Combined
Quality score
Integrator
All Spotlights Combined
Quality score
Integrator

10.4 Appendix B1: Information and Informed Consent Form Experiment 2

Information & Informed consent form

Participant number: _____

Thank you for participating in this study!

In this experiment you are going to judge image quality. Each time, two images are shown on the screen in front of you. These two images show a teapot of different materials. You have to select the image that you think has the highest **material quality**. Selecting an image is done by pressing either the **left arrow** button for the left image or the **right arrow** button for the right image. You always have to select an image and you can take as much time for it as you need. There is no right or wrong in choosing an image.

Before the real experiment starts you get three trial comparisons to get used to comparing the quality of the materials of the teapot.

Participation in this study is strictly voluntary. You may choose to stop participating and withdraw from this study at any time, without providing reason. Please tell the researcher if you wish to stop.

All results will be handled confidentially, and your name will not be included or in any other way associated with the data collected in this study.

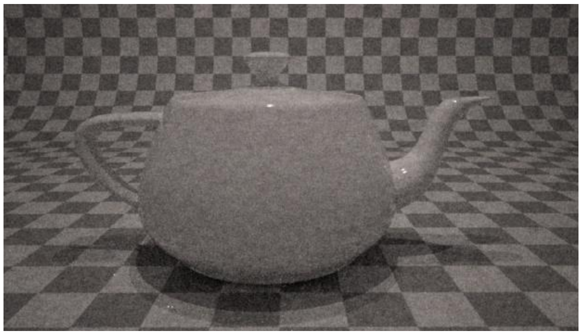
Consent:

I acknowledge that I have read and understand this consent form, and I voluntarily participate in this research study. I understand that I may stop participation at any time, without providing a reason. If I have any questions or concerns now or in the future regarding the study, I may ask the researcher's supervisor, Michael Murdoch.

Participants name:	Participants signature:	Date:
______________________________________	_________________________	______________

10.5 Appendix B2: Indigo Spot Glossy Renderings

25 spp



50 spp

100 spp

200 spp

500 spp

1000 spp