Nicolae Denut

Optimal photon behaviour in virtual environment

A Case study using 3ds Max for architectural visualization

Helsinki Metropolia University of Applied Sciences

Information & Communication Technology

Media Engineering

Bachelor's Theses

Date 27/5/14

Author(s) Title Number of Pages Date	Nicolae Denut Optimal photon behaviour in virtual environment 54 pages + 1 appendices 27 May 2014
Degree	Bachelor of Engineering
Degree Programme	Media Engineering
Instructor(s)	Antti Laiho, Lecturer

A Virtual Environment (VR) is a three dimensional representation of the real world, a computer generated simulation. This simulation has multiple uses among which this study is focusing on architectural visualization.

Architectural visualization plays an important role in today's construction and engineering field because projects need to be evaluated and foundations ensured. In this situation, having a simulated copy of the final product - ranging from small objects to district sized habitation complexes - greatly reduces costs and enhances production.

While adequate know-how and technological requirements are fulfilled, the resulting visualization is not necessarily as life-like as hoped. This gap between the quality of the resulting image (render) or video presentation and the high quality of computer built models can, among other things, be derived from the difference of computer-generated and real light conditions.

Photon behaviour in real environments are studied in great details and well understood. However, when it comes to modelling the same behaviour on computers, the developer may face a multitude of problems that are going to be discussed and addressed in this study.

This study aims to give an understanding and a guide to Global Illumination and determine the optimal technology to be used in architectural visualization.

Keywords	architectural, computer graphics, global illumination, render engine, visualization

Contents

Abbreviations and Terms 1

1 Introduction 3

2 Basics of Computer Graphics 4

2.1 Brief History of 3D Computer Graphics and 3D Studio Max 4
2.2 Workflow using CG Software 5
2.2.1 Interpretation of the Blueprints 5
2.2.2 Modelling and Material Assignment 6
2.2.3 Illuminating of the Scene 8
2.2.4 Basic Render Set Up and Test Renders 9
2.2.5 Camera Set Up 9
2.2.6 Final Render Set Up and Final Render 10

3 Methods, Materials and Theoretical Background 10

3.1 Research Methodology 10
3.2 Illustrations and Images 11
3.3 Light as an Object 11
3.4 Light in the Real Environment 11
3.5 Simulations of Light 12

4 Illumination Options 13

4.1 Local Illumination 13
4.1.1 Standard Spotlight 14
4.1.2 Standard Directional Light 14
4.1.3 Standard Omni Light 15
4.1.4 Standard Skylight 16
4.1.5 Photometric Light 17
4.1.6 IES Lights 19
4.1.7 Image Based Lighting 20
4.1.8 Ambient Light 21
4.2 Global Illumination 22
4.2.1 Reflections 23
4.2.2 Light Bouncing 24
4.2.3 Ambient Illumination and Occlusion 25

5 Methods for Global Illumination 26

5.1 Scanline Rendering 26

5.1.1 Radiosity 27

5.1.2 Ray-Tracing and Shadows 28

5.2 Mental Ray 29

5.2.1 Skylights 29

5.2.2 Final Gather 30

5.2.3 Caustics 31

5.2.4 Photon Mapping 33

5.2.5 Ray-tracing 34

5.2.6 Ambient Occlusion 35

5.3 V-Ray 36

5.3.1 Brute Force 37

5.3.2 Irradiance Map 37

5.3.3 Photon Map 38

5.3.4 Light Cache 39

5.3.5 Ambient Occlusion 40

5.3.6 Caustics 40

6 Case Study on Global Illumination 41

6.1 Solution with Scanline Rendering 41

6.2 Solution with Mental Ray 42

6.3 Solution with V-Ray 43

7 Summery 45

7.1 Materials 45

7.2 Lights and the Camera 46

7.3 Results 47

8 Conclusions 49

References 51

Appendices

Appendix 1. Renders of Case Studies (3 pages)

Abbreviations and Terms

Architectural Visualization

The visualization of architectural plans using 3D computer graphics software.

Camera

The camera is the observer in the Virtual Environment. The two dimensional images are created from the point of view of the observer.

Computational Power

Computational power is the ability of a given computer to complete a certain number of calculations in a given time. When higher computational power is available, the time required to compute a given task is less than for a system with lower computational power. The best indicator of computational power is the CPU's clock speed in the computer.

Computer Graphics (CG)

Graphics created using computers. In this study computer graphics refer to 3D computer graphics in contrast to other graphics software.

Global Illumination (GI)

An algorithm or set of algorithms that calculate particle behaviour in accordance with the well understood physics of light and optics.

Model and Modelling

The model is a three-dimensional representation of a structure or an object. Modelling is the creation of three-dimensional models in the virtual environment.

Photons

Photon (from Greek phōs, phōt- ‘light,’) is a term coined by the American chemist G.N. Lewis to Albert Einstein’s idea that the electromagnetic radiation itself is consisting of small grains or particles (photons) thus proving previous mathematical constructs. While the idea is a hypothesis, it makes mathematical calculations work and is used in computer science to calculate the behaviour of photons resulting in realistic light simulations. [1]

Render

An image generated from the three-dimensional model. A render is created from the point of the view of the camera.

Render Engine

Render engine is a piece of software that is responsible for generating an image from a three-dimensional model. These softwares usually allow the developer to fine tune different aspects of the image creation. Render engines commonly provide additional functionality to the main computer graphics software such as materials, lights and cameras.

Scene

The scene is a place within the virtual environment where the subject of observation is situated. To determine the accuracy of the concepts and methods, a scene called Cornell box was used. A Cornell box is a simple environment that was created by Cindy M. Goral, Kenneth E. Torrance, and Donald P. Greenberg for the 1984 paper "Modelling the interaction of Light Between Diffuse Surfaces" [2]. Ever since the publication of that study the Cornell Box has been widely used to model and simulate light behaviour.

Virtual Environment (VE)

Virtual Environment or VE is discussed in the sense of being a representation of reality. A simulation generated by computer or computers that mimic the real world environment. However, it is important to note, that VE as a term can be used to describe a number of things that are unrelated to this study. Furthermore, Virtual Reality (VR) is essentially a VE and thus the concepts discussed herein are applicable to it, as well.

1 Introduction

Although the physics and the mathematical base of photon behaviour are well understood, to simulate an accurate copy of reality many factors have to be taken into consideration. For a start when photos hit a surface they do not simply bounce back to the observer illuminating the object, but scatter and interact with other objects while, at the same time, a portion of them gets absorbed. The matter is further complicated by the fact that light or photons do not travel to only one direction and that certain surfaces, on the other hand, do make photons travel to one direction only. [3] This complicity makes simulation of light very difficult and takes a huge draw on computational power. While a VE can be illuminated effectively without using complicated algorithms to calculate every bounce, absorption and re-bounce, the result often looks unrealistic to the observer.

In many cases, the purpose is not to create a hyper-realistic representation. In the case of Architectural Visualisation, however, the resulting image calculated by the computer from the virtual model placed in a virtual environment is required to be as accurate to reality as technologically possible. A developer thus faces a dilemma. It has to be assessed whether more time or more computational power is available?

In addition to all of this, realistic lighting is often neglected. While illumination itself is thought to be important, realism is only an added benefit rather than a goal. For architectural visualization, realistic lighting is vital. The goal of architectural visualization is to enable the architect to see a realistic image of the project and it being physically created. Flaws in the plan, such as too small windows as an arbitrary example or too little built in lights must be discovered. This information is critical, thus real behaviour based light simulation is desired, but a balance between effectiveness and effects has to be maintained. To achieve this balance, Global Illumination algorithms are used. These algorithms are not always part of a given 3D computer graphic software and may come as a complementary package, a render engine.

The purpose of this study is to introduce global illumination as a candidate group of algorithms to create a more realistic lighting to virtual scenes and to a lesser extent to try to determine the optimal technology or an optimal combination of algorithms to be used in architectural virtualization with different render engines. This study focuses on the practical side of using GI and the most widely used technologies for architectural visualization in CG.

The best resource for an in-depth overview of the global illumination technologies used currently in CG softwares is the Global Illumination Compendium by Philip Dutré that is available at http://people.cs.kuleuven.be/~philip.dutre/GI/ .

2 Basics of Computer Graphics

2.1 Brief History of 3D Computer Graphics and 3D Studio Max

Computer Graphics and 3D Computers graphics share a common history and can be originated from the creation of the first digital image of a baby by Russell Kirsch. This digital image made in 1957 is amongst the 100 photographs that changed the world, a collection by Life magazine. [4] The phrase “computer graphics” is credited to William Fetter, who created the first perspective CG of a human body in 1965. According to him at that time there were only a room-full of people doing computer graphics, him being the only one with an art background and a degree in the arts. [5]

In the same year at the University of Utah the computer science faculty was founded. This faculty produced among others the algorithms nowadays commonly used as shaders for 3D materials as well as texture mapping and many more technologies. [6] A graduate student at the same university made the first animation in 1972. He titled it “A Computer Animated Hand”. The animation was of his hand. [7] However softwares before the 80’s were not widely available for the public, nor were computers, thus the revolution in CG coincides with the commercial availability of the personal computer.

Figure 1. The Autodesk logo in 1990. [8]

In the 1980's Autodesk published The Yost Group's modelling and rendering application. The software was called THUD. At the time CG and Autodesk were very different from today, not only their logo that can be seen in Figure 1, but the market as well. CG was mainly used as experimentation and was not thought to replace legacy visualization done by a graphic artist. THUD was fast superseded by Autodesk's 3D Studio the very first "ancestor" of the software that is the most popular choice today in architectural visualization: Autodesk 3d Studio Max. [9]

Today most people working to create 3D architectural visualization are referred to as 3D artists.

2.2 Workflow using CG Software

In order to give a comprehensive picture of the subject of this study, a simplified workflow of creating CG is demonstrated here. While this workflow is ideal, it is rarely the case. This study focuses on lighting since illumination is a core concept that follows through the entire creation of CG.

2.2.1 Interpretation of the Blueprints

As a starting point in an ideal project, the company responsible for creating the CG is presented with the blueprints of the structure or object that needs to be modelled and rendered.

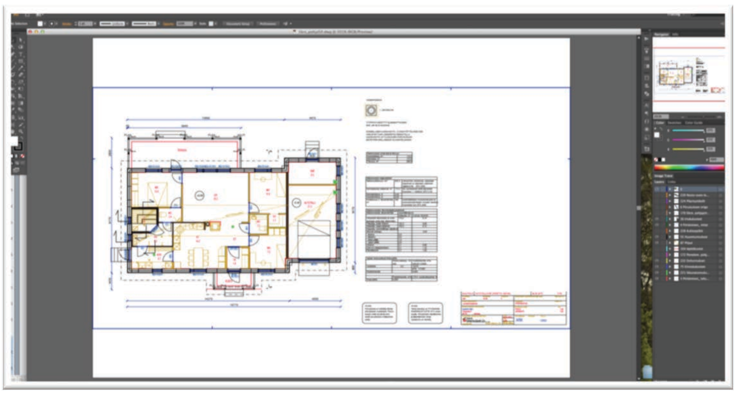


Figure 2. Blueprint of a building.

Figure 2 shows the architectural plans or blueprints of a house opened in a graphics software for inspection. The blueprints are examined and cleared of unnecessary lines and information by the CG artist and imported to the CG software to serve as a base to create a virtual model. If the unit sizes are correctly set, a blueprint can serve as a base to create the models upon it.

2.2.2 Modelling and Material Assignment

A model of the structure or object is created using basic geometric shapes such as spheres, plains, boxes, toroids, cylinders and pyramids as it can be seen in Figure 3. Editing their polygons can modify these basic geometries.

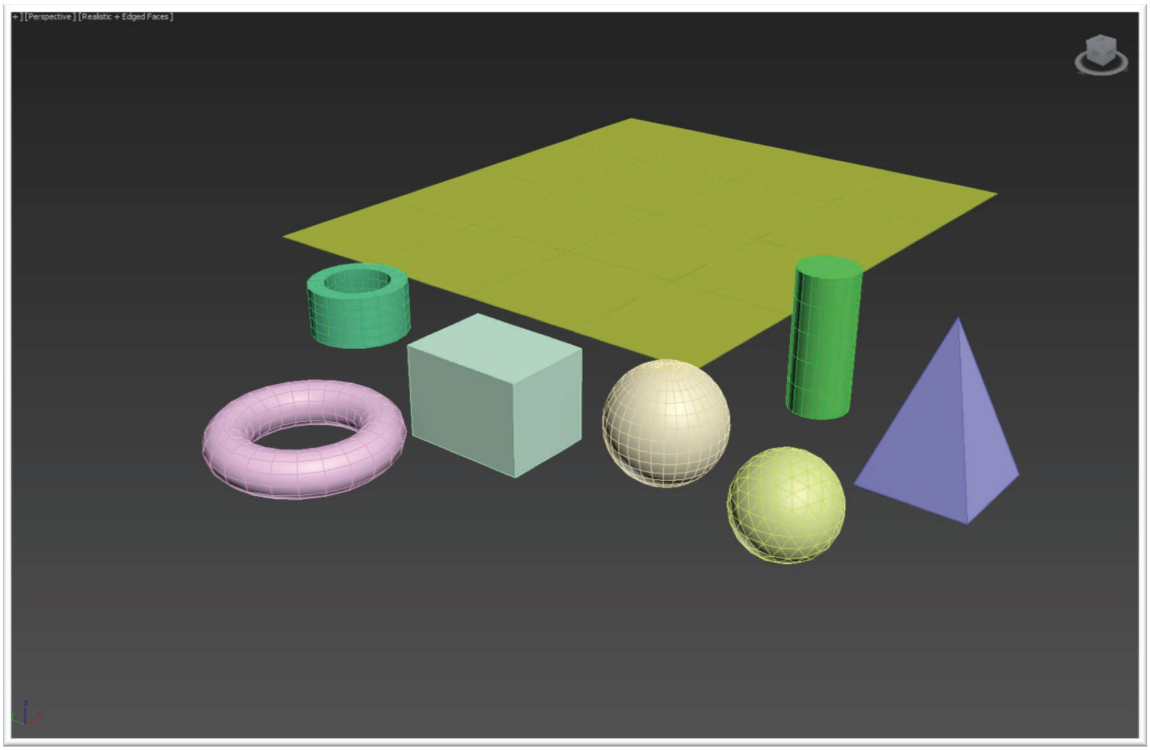


Figure 3. Basic geometric shapes.

A model is built up of vertices or points that are connected by edges, that are in turn defining faces. Faces in a plane are called polygons and polygons in line create surfaces that define a model as it is shown in Figure 4. [10]

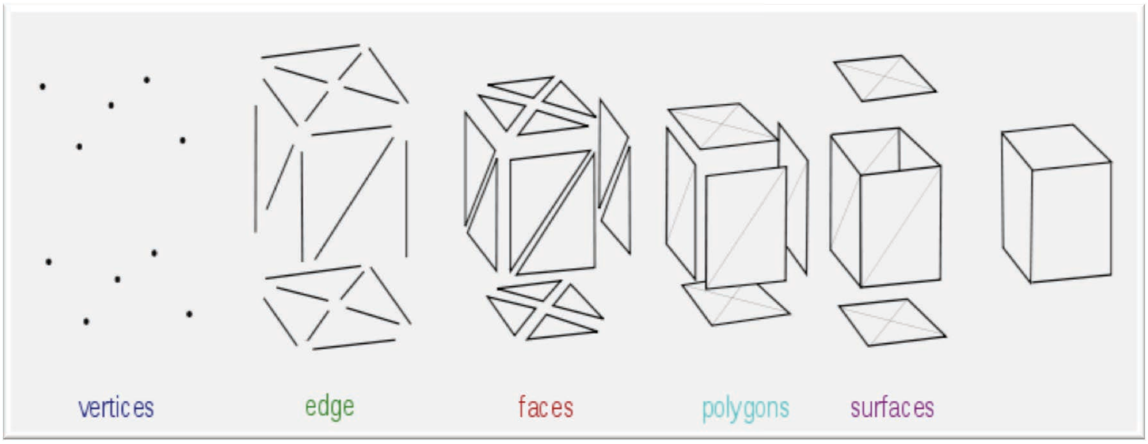


Figure 4. Elements of a 3D model. [10]

As part of the modelling process, the models are assigned materials (or shaders) that are indicative of the real world equivalent of the model's materials. Materials need to be set up in the CG software's material editor.

Figure 5. Material editor, reflection and refraction settings in the V-Ray render engine.

For materials, a number of settings and attributes are required starting with colour or texture (image) through its reflection and refraction values to determine how light interacts with it (Figure 5). Reflection values are indicative of how mirror-like the surface is, and refraction is responsible for how light is split. In practice, a glass has a higher refraction than air; thus light bouncing off the glass or liquid drop surface may split into rainbow-like colours.

2.2.3 Illuminating of the Scene

Once modelling and material assignments are done, the CG artist usually inserts the light objects to the scene to illuminate the model. Up until this point the model is illuminated by ambient light, a non-physically accurate light that has no source and serves only to make created objects visible for the developer.

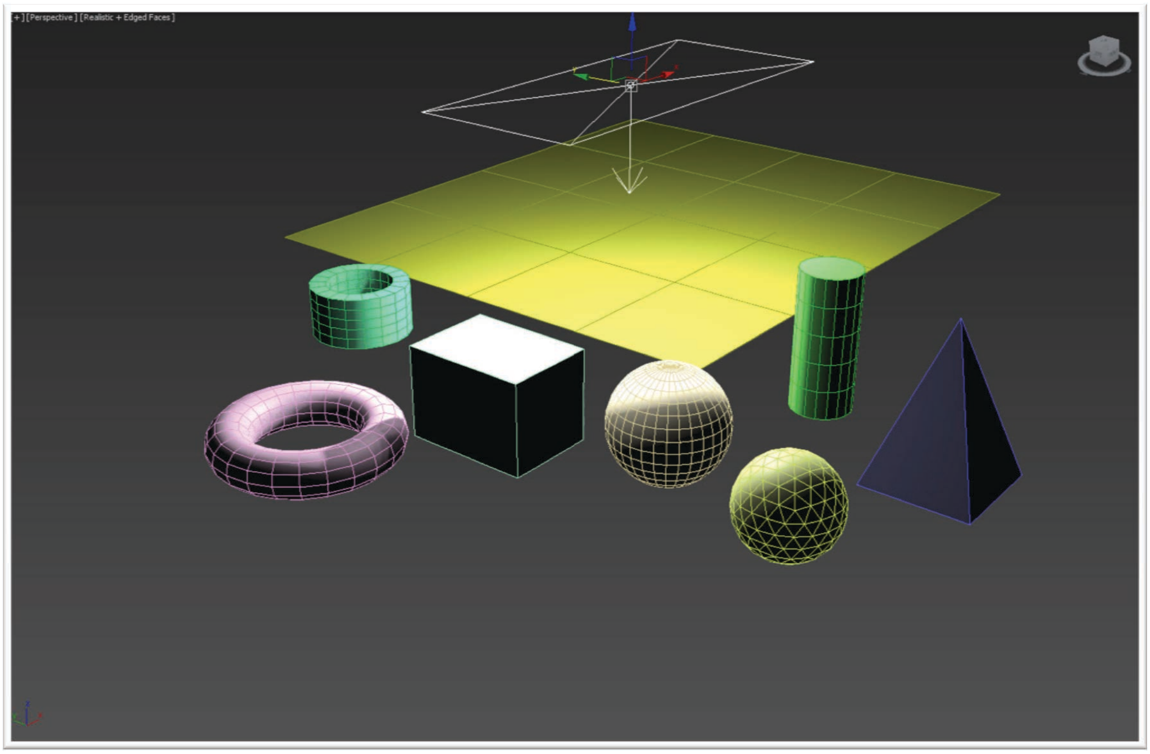


Figure 6. Light object in a scene.

As it is shown in Figure 6, the light object can be inserted just like any other basic geometric shape. The shape of the area that illuminates and the direction of illumination can be edited thus mimicking many type of light sources.

2.2.4 Basic Render Set Up and Test Renders

At this point, the CG model is almost done and the settings for rendering need to be determined. Test renders are created usually using the software's viewport (the framed area of the display) as an observer or camera.

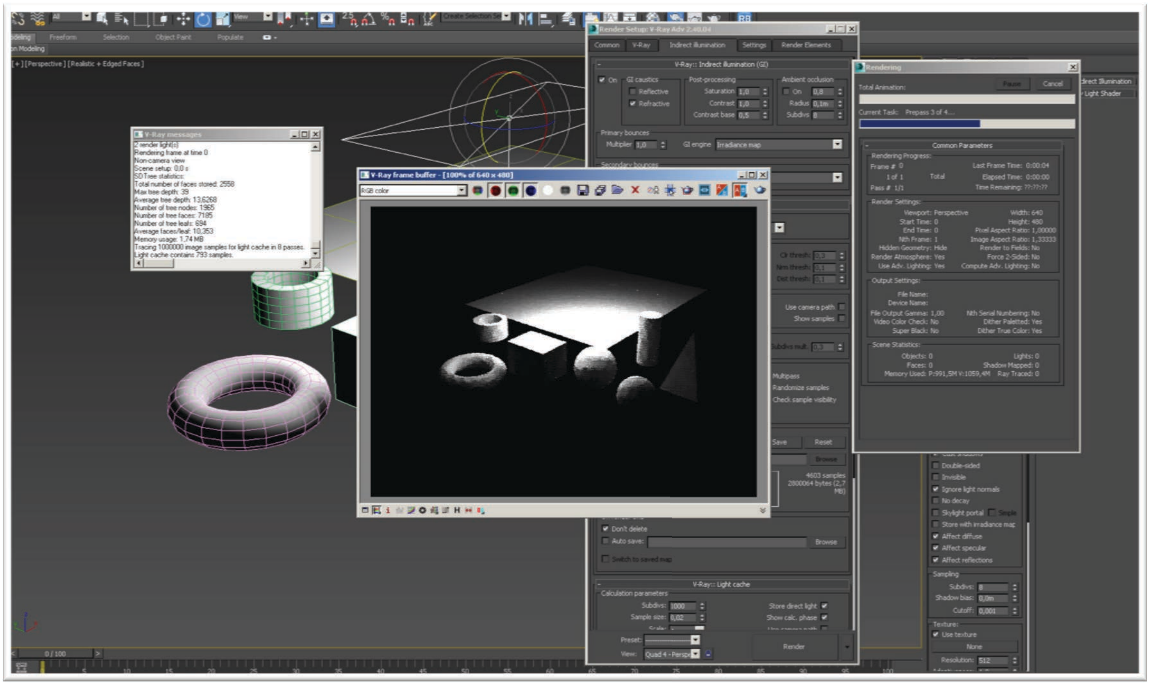


Figure 7. Rendering using the viewport as camera.

In Figure 7 the CG software 3ds Max's graphical user interface is shown with a render window open. If the developer notices any mistakes, then previous steps will be undertaken again as long as the scene meets the requirements. Often it is not required to wait through the render process, mistakes and problems can be seen "on the fly" and the render process can be cancelled to make the required changes right away.

2.2.5 Camera Set Up

If the test renders yield the desired result, observation points are created. The developer or 3D artist inserts cameras into the scenes that can be used to render an image from their viewpoint.

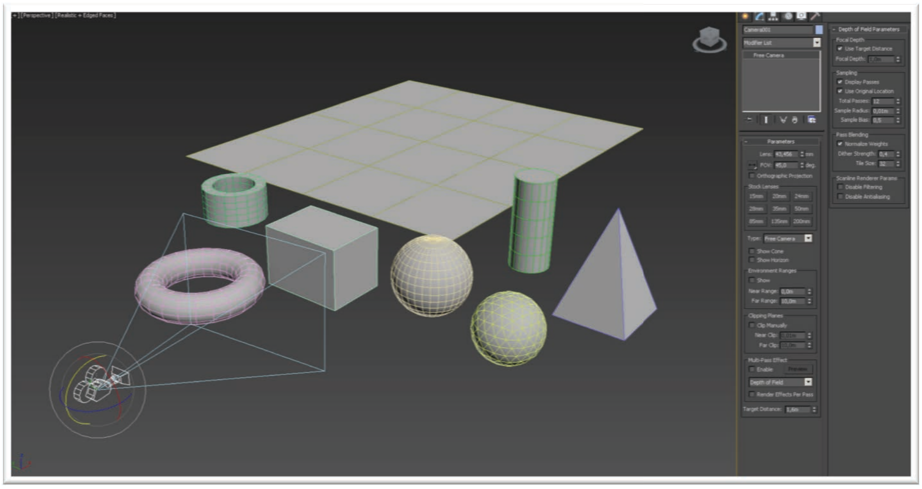


Figure 8. Camera set up.

Camera set up as it can be seen in Figure 8 is a multi-fold process, where direction and focal point can be set among others. Different render engines can have their own pre set cameras that mimic real world devices and their characteristics, like aperture and distortion.

2.2.6 Final Render Set Up and Final Render

As a last step, the rendering settings are fine-tuned, and final renders are created. While test renders are usually set to a low resolution, the final renders are run in higher resolution. Rendering high-resolution images take considerably longer time. Because final renders are usually time consuming, all corrections should be done beforehand.

In reality, however, it is rarely the case that final renders are final. Normally final renders are sent to the architect, and they come back with the architect's notes to be corrected. For this reason a good workflow and a reasonable rendering time is desired.

3 Methods, Materials and Theoretical Background

3.1 Research Methodology

The primary strategy to conduct the research was to systematically review the wide literature available in the field of CG. The discussed methods and concepts were tested, and their results are included in the study. For the purposes of this study 3ds Max, the most commonly used CG software in the field of architectural visualization was used. For the rendering of the test scenes, three different render engines were reviewed. These 3D render engines were the **Scanline Rendering** also know as the default render engine in 3ds Max, **Mental Ray** developed by NVidia Corporation and **V-Ray** developed by the Chaos Group.

3.2 Illustrations and Images

If not otherwise referenced, illustrations and test scenes were created by the company RealTime Suomi. The models used in the scenes if not otherwise stated or referenced are also made by and are the property of said company. RealTime Suomi is located in Espoo and was established in 2012. The company primarily creates 3D architectural visualization for architectural firms in Finland, Germany, Italy and Hungary. The company employs 5 persons at most. RealTime Suomi is lead by the writer of this study.

3.3 Light as an Object

Light is everywhere and while we all understand what light is we cannot explain, "what is it" even today. As Benjamin Franklin put it "...I must own that I am much in the Dark about Light". [11, 264.] Light has two principal natures, one is its wave nature and the other one is its particle nature. While neither of these two natures fits light perfectly, both of them combined gives a fairly good approximation. [3, 3-8.] For the purposes of modelling the behaviour of light, it is assumed that its nature is particle-like thus mathematical calculations on its behaviour can be made and used in a practical way. When light is referred to as being made up of particles, it is called photons.

3.4 Light in the Real Environment

As it was pointed out earlier, light scatters and "falls" into the eye. This behaviour is true only within the environment of the Earth where atmosphere is present. In a vacuum light travels from the source such as the sun in a straight line or rays until it hits an object, like a planet or the moon then it changes direction and hits an observer, like the human eye. However, not all objects reflect light, for instance a clean glass is invisible because light passes through it. In the vacuum of space, light travelling just in front of the observer but not hitting it, will be invisible. [3, 4.] These facts are well demonstrated in Figure 9, an illustration from the book "Seeing the light".

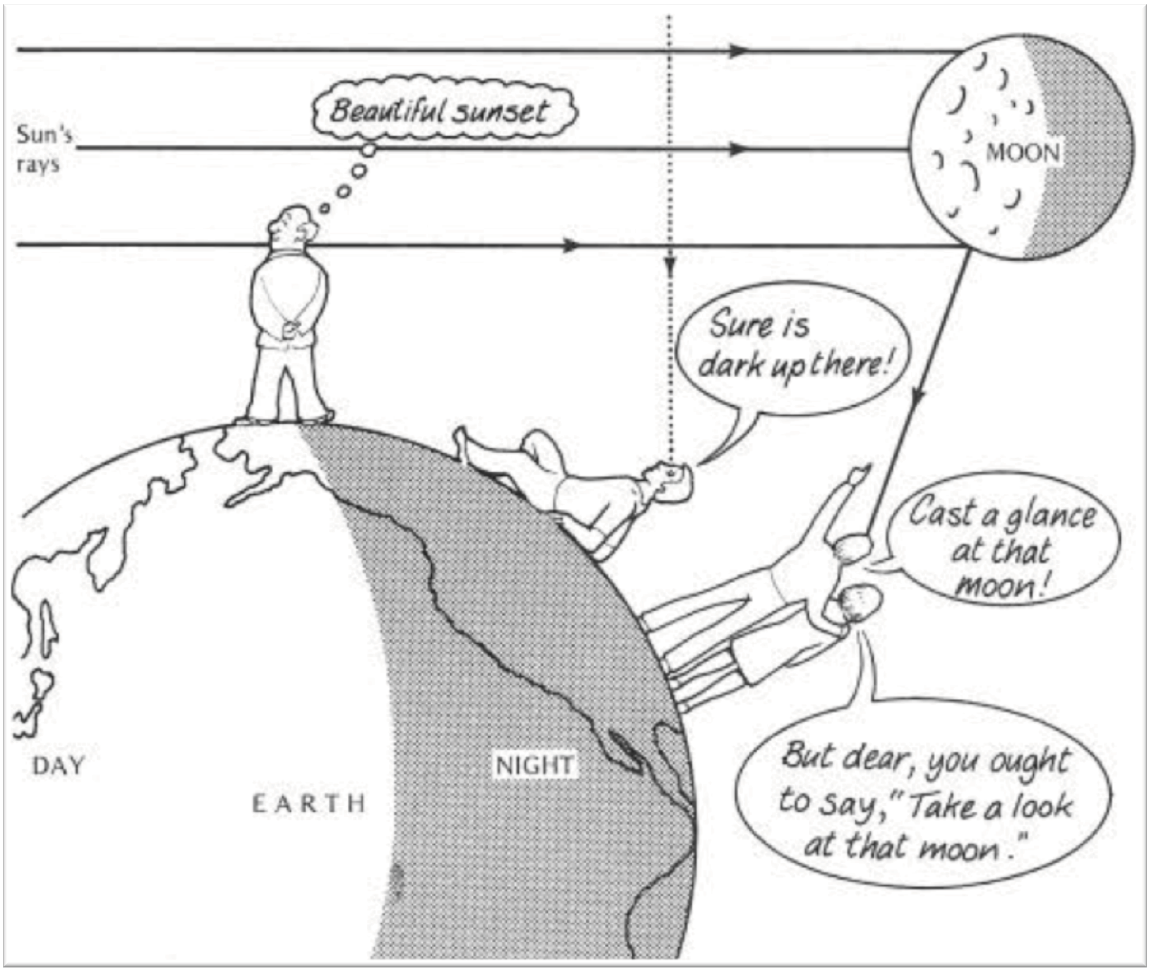


Figure 9. Light behaviour in real environment. [3, 4.]

On the Earth, however, light cannot travel undetected because of the tiny particles, like dust and water droplets in the atmosphere that scatter it. In the mist or through dusty air, the property that light travels in straight lines can be well observed. [3, 5.]

3.5 Simulations of Light

In a virtual environment, the basic property of light still applies. It travels in straight lines. Light directed at an object will not illuminate other objects that are not in its path because it does not scatter but changes direction only once to hit the observer. A virtual environment does not have an atmosphere and light originating from basic light sources will not take into account all the bounces that photons would do in reality.

4 Illumination Options

4.1 Local Illumination

Lighting is nothing more than the creation and setting of light objects in a virtual scene. Local illumination is not physically accurate. The concept of local illumination is very simple: if an object is pointing towards the light it will be illuminated, if it points away from the light it will not be illuminated. This oversimplified way of lighting is far from realism, but is a useful base for lighting a scene and when it comes to rendering, local illumination is very fast.

With local illumination, there is no bouncing effect or reflections, but these can be "faked" by using multiple light sources and coloured light sources to imitate the effect of bouncing light. A scene is using multiple light sources to "fake" real light behaviour renders considerably faster than the same scene using real physics to simulate the same effect. When illuminating a scene, the developer is presented with multiple options for the sources of light. For instance in 3ds Max lights are divided into two categories.

"Standard lights" are objects that simulate common light sources, like lamps and the sun. The intensity of standard lights can be changed, however, their value has no correlation with physical reality. [12]

"Photometric lights" are the other category that differs from standard light in that its propagation through an environment is based on physically accurate simulation.
Their values can be given in

- luminous flux
- illuminance
- luminance
- luminous intensity [13].

In addition to these lighting possibilities, different render engines may provide additional lighting possibilities and settings, such as photon counts and render engine specific light sources. [14;15] Depending on the CG software or technology used the following objects are most commonly available as light sources.

4.1.1 Standard Spotlight

Spotlight casts a beam of light on an object or area. As it can be seen from Figure 10, spotlight is a directional light that behaves like a flashlight. The smaller image within Figure 10 shows the shape of the light, while underneath the actual render using spotlight illumination can be seen.

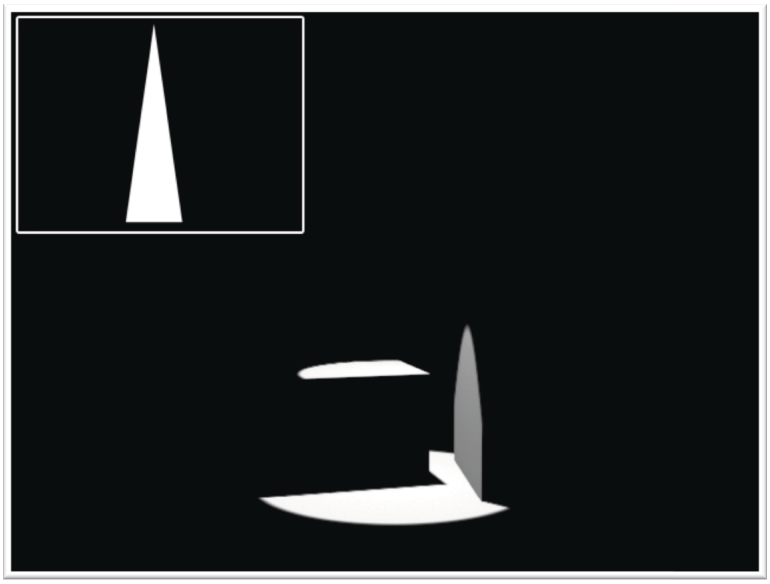


Figure 10. Spotlight.

Spotlights can be targeted or freely moved around. This light type is normally available in all the CG softwares and technologies that are used to create 3D visualization. [16]

4.1.2 Standard Directional Light

Directional light is primarily used to simulate the sun or sun-like behaviour. This light object casts multiple light rays in a given direction, as it is shown in Figure 11.

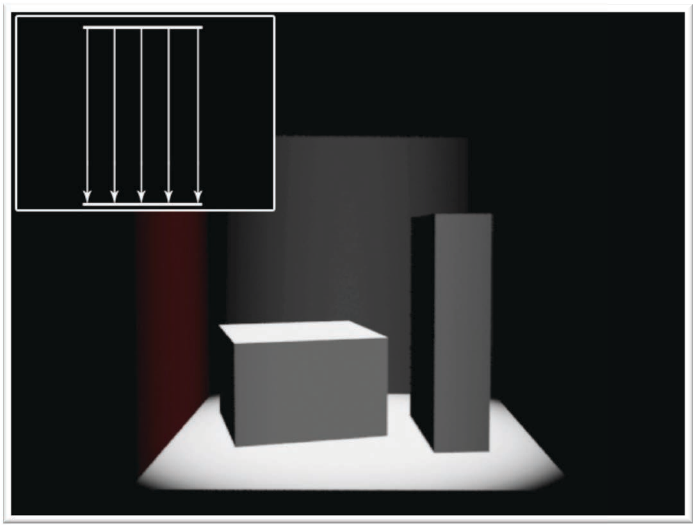


Figure 11. Directional light within an enclosure.

As with the spotlight object, directional light can be also freely rotated, moved or targeted to a certain area. [16]

4.1.3 Standard Omni Light

An omni light or omnidirectional light objects cast multiple light rays from a single source to all directions. The source of an omni light can vary from a single point to cylindrical objects. [16]

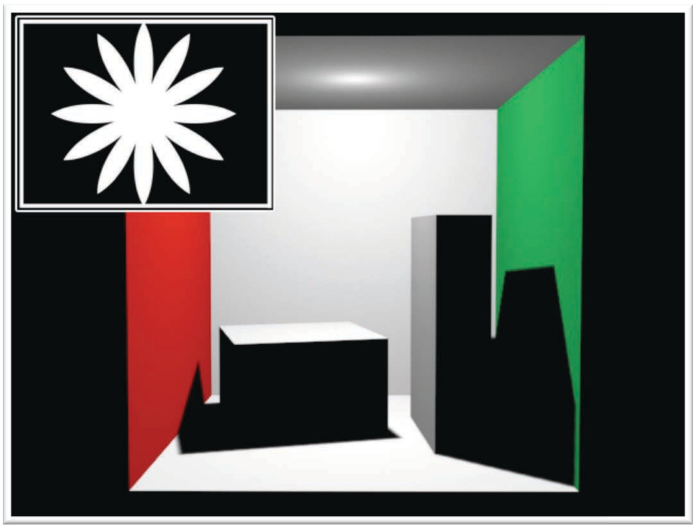


Figure 12. Omni light.

Omnidirectional light objects are also readily available in most 3D softwares or technologies. Figure 12 shows an illustration of omnidirectional light and a scene illuminated from top with one.

4.1.4 Standard Skylight

The skylight object is to model daylight; it is created as a dome above the scene. [16] Figure 13 shows skylight as a dome over an object and a render created by using only skylight as illumination. The Cornell-box is open to the camera, thus the light from the dome can illuminate the objects contained within.

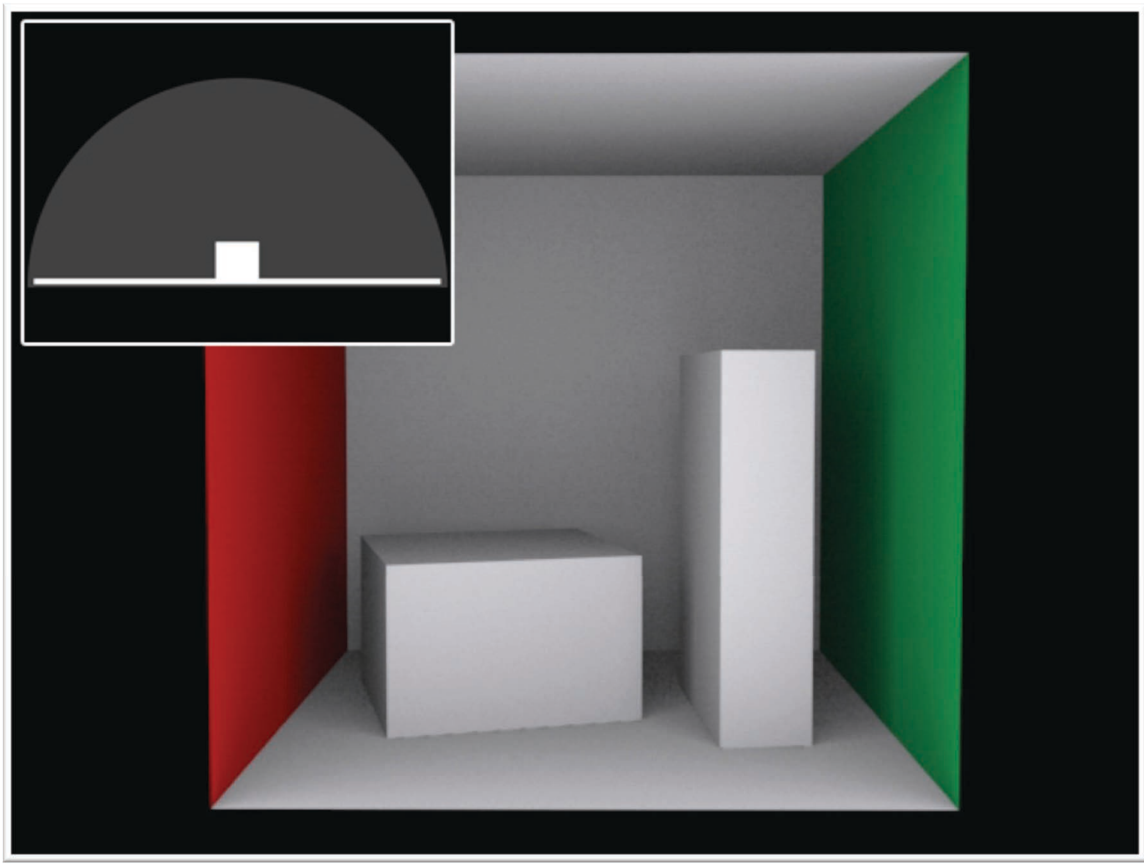


Figure13. Skylight.

Skylight can be very useful but together with GI options and additional lights it is important to ensure that walls have closed corners, otherwise light can "leak" in through the corners.

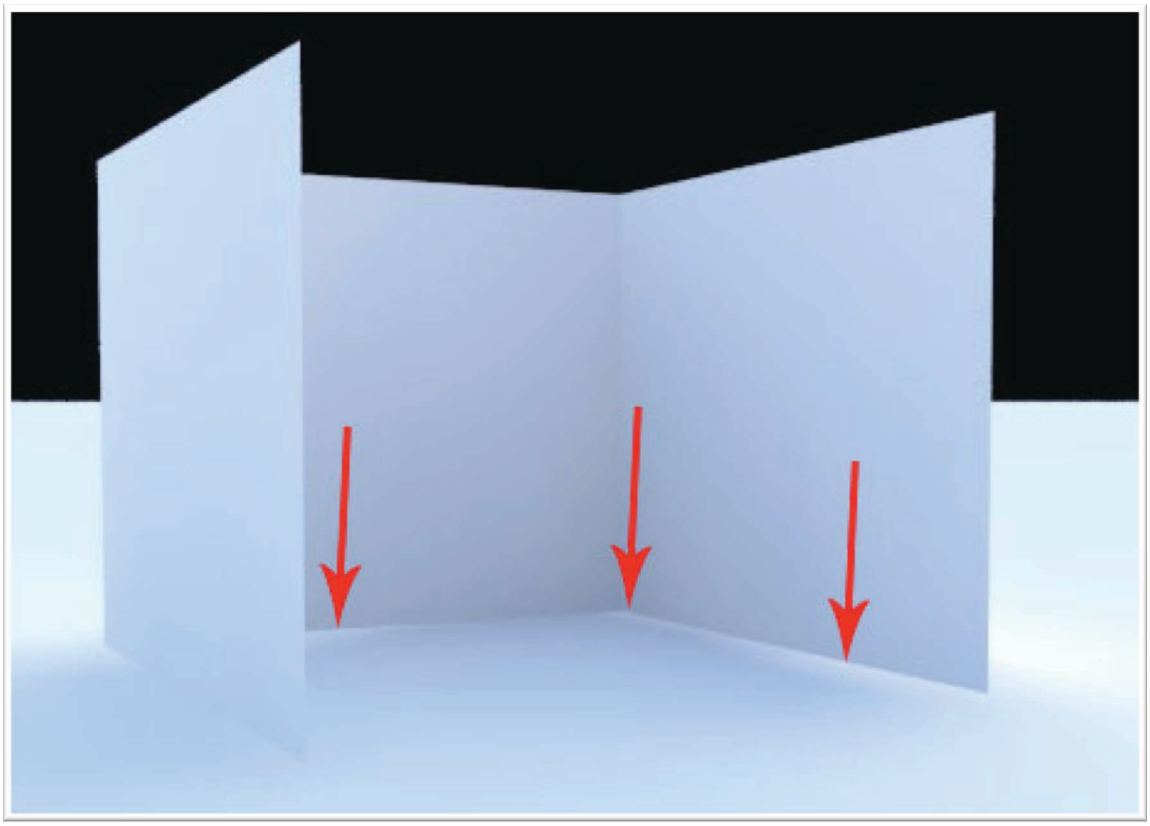


Figure 14. Light leaks marked by red arrows. [17]

Figure 14 shows light leakages appearing at the bottom of the walls. Walls without width or edges can cause this effect when GI is enabled and a lot of light is present, such is the case of skylight in combination with other lights. Some algorithms are more likely to produce light leaks than others. Usually photon mapping type of algorithms are the ones that need to be inspected.

4.1.5 Photometric Light

Photometric lights use real values for intensity, such as candela (cd), lumens (lm) or luminous flux (lx). These lights are capable of mimicking reality. Photometric Target light is a targeted light object that tries to simulate real world light in 3ds Max. It has a target sub-object to aim the light. Figure 15 shows a scene illuminated by a photometric target light.

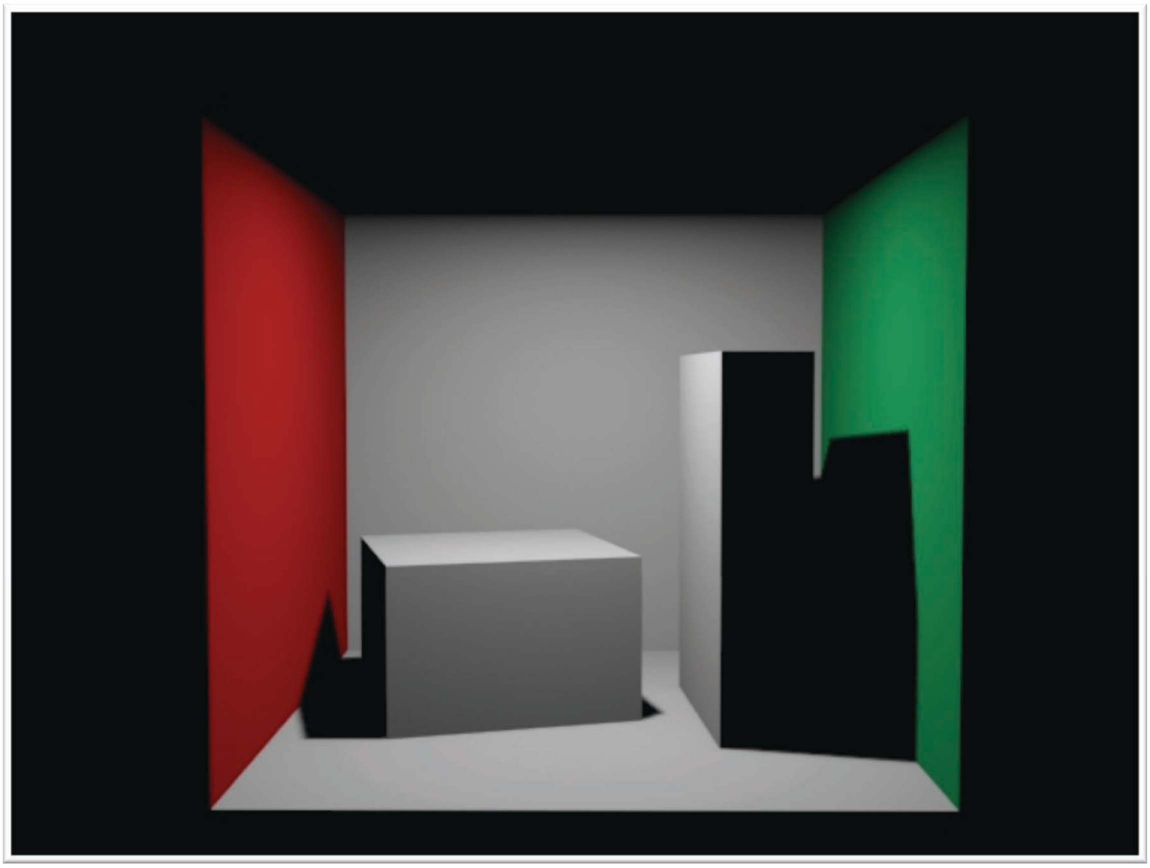


Figure 15. Photometric target light with 10000 lumen intensity.

Its distribution can be chosen in the software: from uniform, meaning that it casts light equally in all directions, spotlight that casts a focused beam of light as a flashlight or web distribution that is based on a geometric mesh to model the intensity pattern of the light source. [16] It can be assigned photometric light information from real light sources as well.

Photometric Free light is one of the Photometric lights in 3ds Max, it has the same parameters as target light does but without targeting. [16]

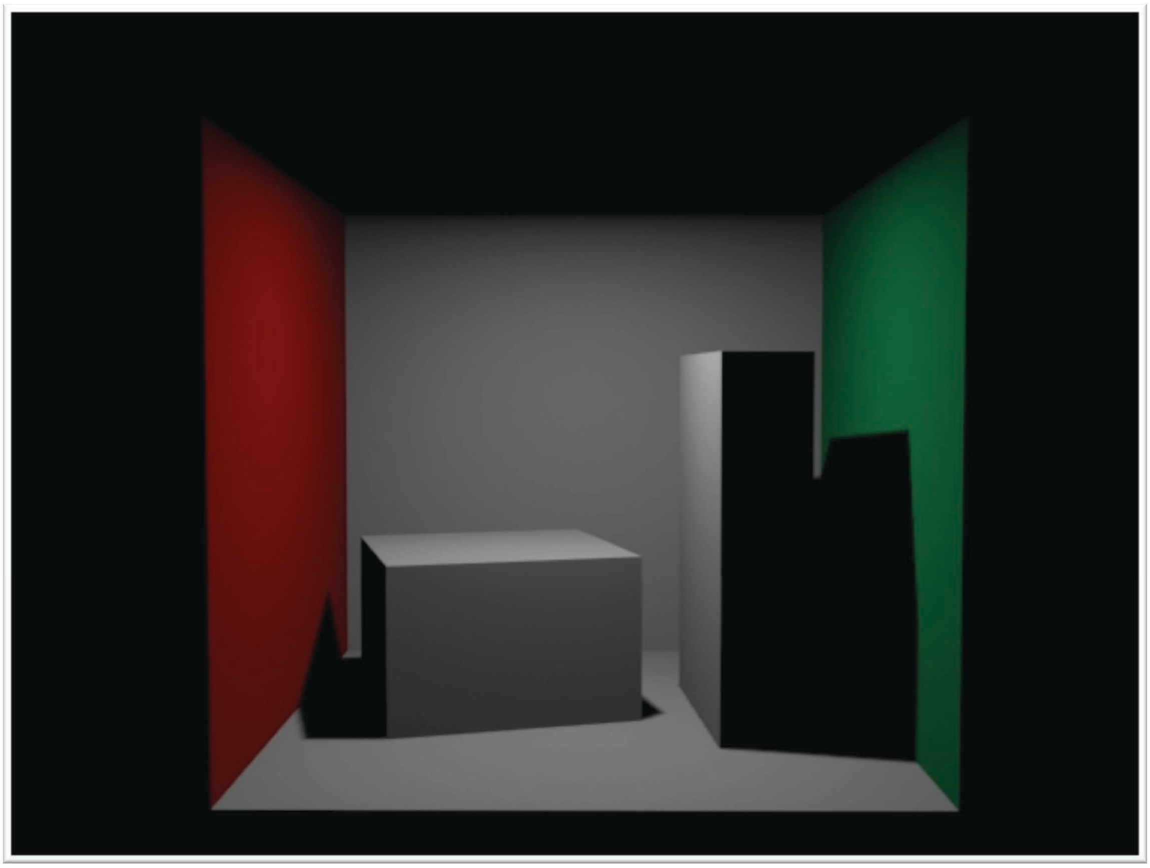


Figure 16. Photometric free light with 3000 lumen intensity.

The transformer object has to be used to position a photometric free light in the CG software 3ds Max. Once rendered there cannot be a visible difference between a photometric target or photometric free light. Compare Figure 16 with Figure 15.

4.1.6 IES Lights

IES stands for Illuminating Engineering Society and is a standard file format to transfer photometric data electronically. The IES photometric data can be added to the photometric light's attributes to simulate a real light source. Figure 17 shows examples of IES lighting profiles that can be added to photometric lights.

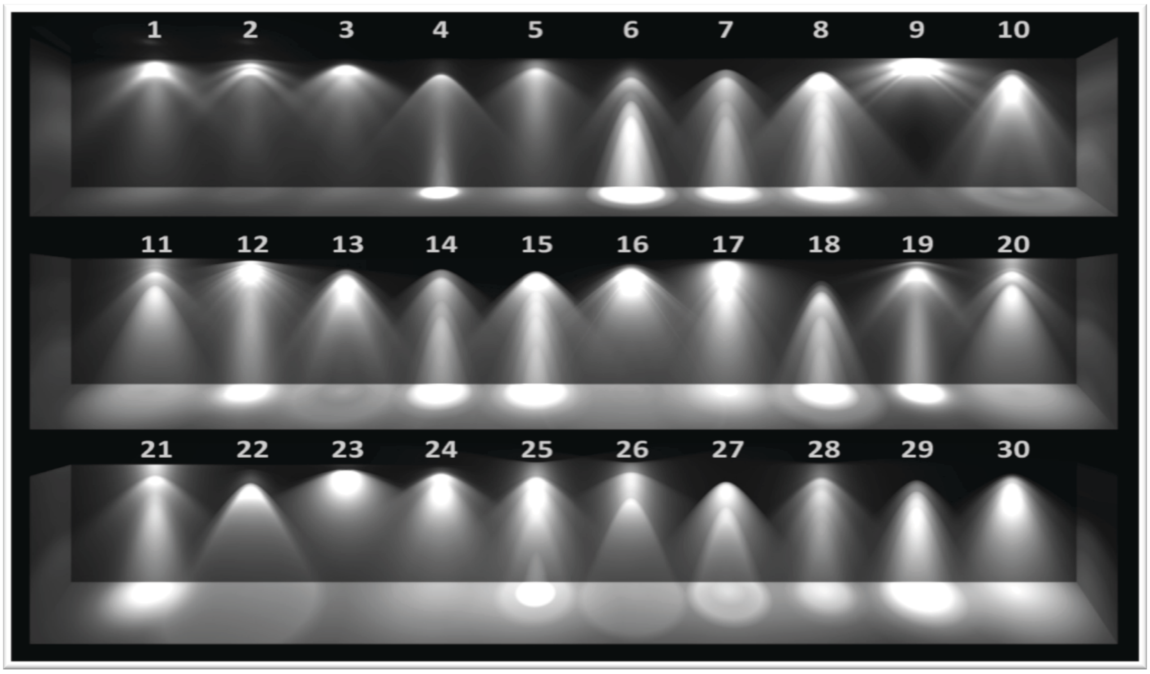


Figure 17. Examples of IES lighting profiles. [18]

These files are commonly available from lighting manufacturers. While IES is very effective when it comes to light effects, it is rarely used unless the effect itself is important to portray. [18] The reason for its relatively low usage comes from the time demanding task to research and set up usually a high number of lights.

4.1.7 Image Based Lighting

Image based lighting or IBL is a rendering technique that is gaining popularity since processing power has risen. The idea of IBL is to map an image, ideally a high-dynamic-range image (HDRI) on a light, such as Skylight, which is a dome around the scene. Most 3D computer graphic software has dedicated options to use image based lighting. Figure 18 shows image based lighting in Autodesk's Maya software. Maya is a CG software like 3ds Max, aimed at the Macintosh platform.

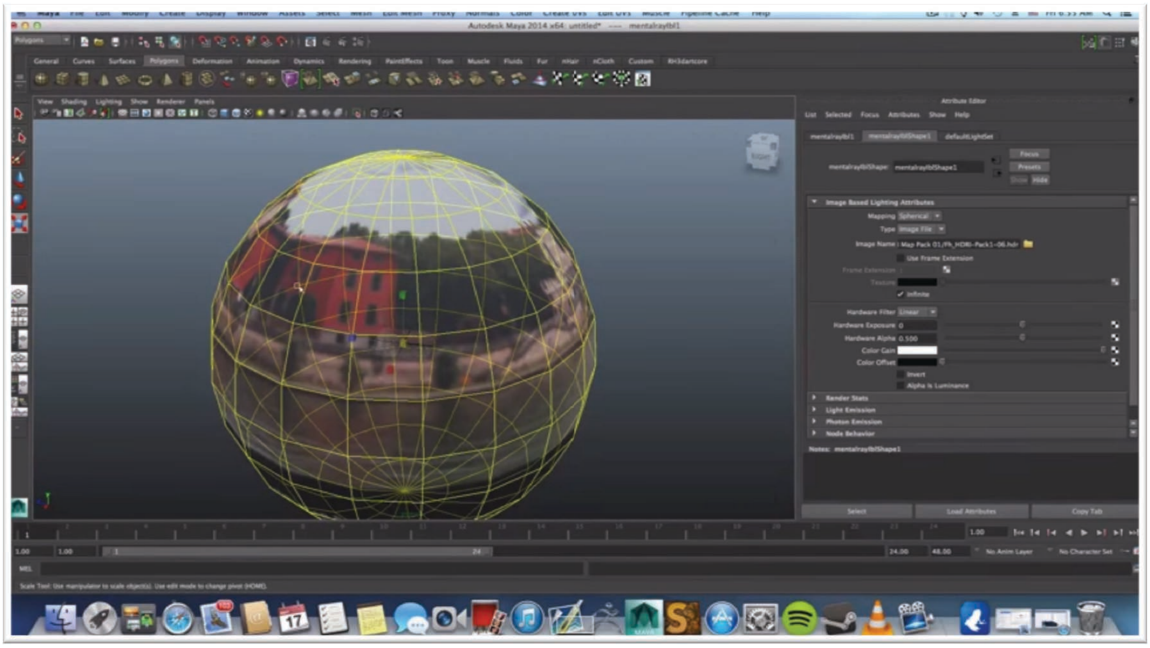


Figure 18. HDRI image wrapped around a 3D model. [19]

Once the image is mapped to the light the render engine can generate environmental lighting from the image information. For instance if a phone boot from a real location is to be rendered from the outside, a panoramic HDRI image can be taken at the location and wrapped around the model so that the generated environmental lighting fits the real lighting conditions perfectly. [20]

4.1.8 Ambient Light

Ambient light is a component of surfaces. In effect, it emulates the reflection of lights without a light source. When using standard lights ambient light is usually neglected. Ambient light can be controlled in many 3D computer graphics software. Also it is turned on by default so that the developer can see the model.

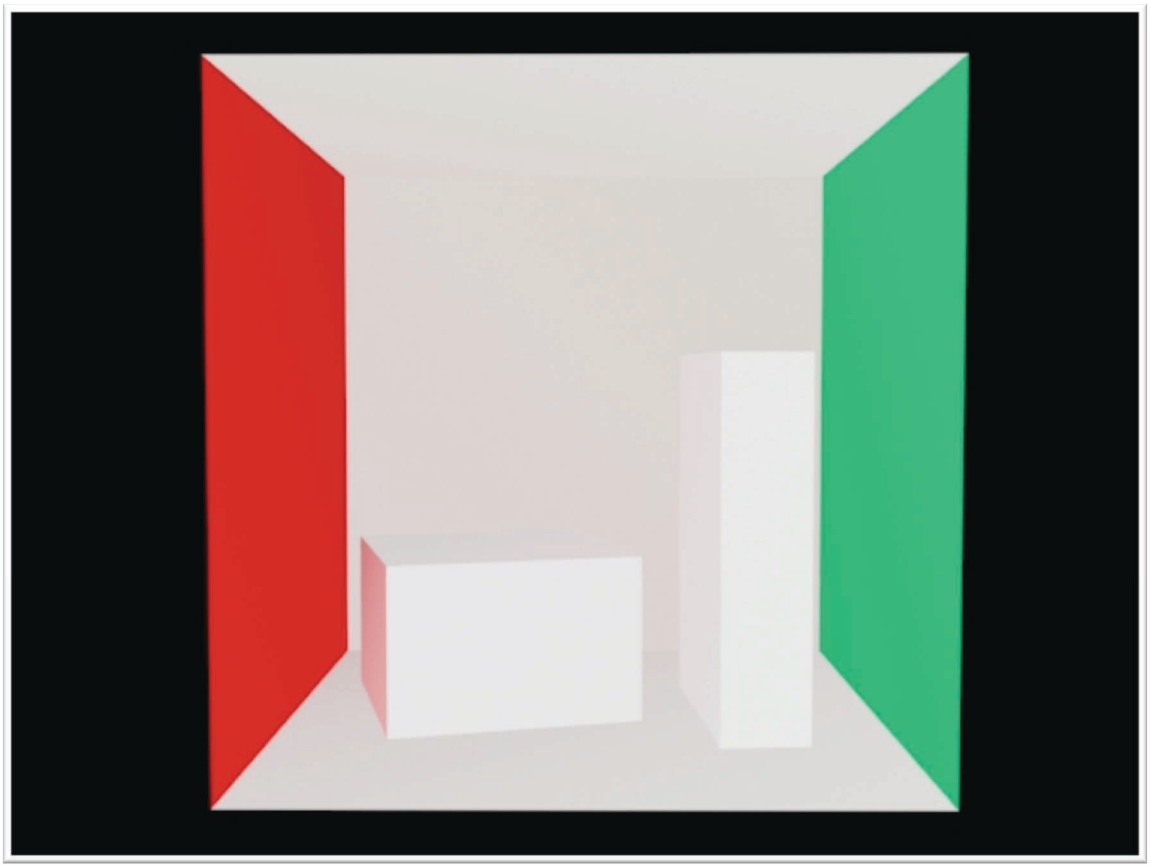


Figure 19. Ambient light.

The intensity and colour of ambient light can be adjusted to better simulate reality. While the colour of ambient light can be set, it is greatly affected by the surface colour of objects in the scene. [21;22] Ambient light produces dull and uniform surfaces therefore it is usually disabled once light sources are manually added to the scene, see Figure 19. Some render engines disable it by default once lights are added.

4.2 Global Illumination

Global illumination is the technology that tries to mimic realistic lighting conditions in CG. In most computer graphics software, the realism is achieved by utilizing multiple methods of calculation for different scenarios of light behaviour. While render engines have dedicated settings for global illumination, in most of the cases it proves to be insufficient. Rather than using one setting to affect the calculation of light behaviour in a given scene, simultaneous use of multiple technologies is more productive.

There are algorithms for calculating reflections via tracing light rays, light bounces by calculating diffuse reflections, ambient illumination and occlusion determining how exposed each point in a given scene is to light [23]. These algorithms could be used as stand alone solutions. This study is going to illustrate the result of such approach in contrast with using these algorithms combined in the case study.

The main problem with using GI methods as stand-alone solutions is that light behaves in certain ways with different materials. While light is reflected almost a 100% in a mirror surface, it also bounces and exposes other surfaces. While the algorithm can calculate the perfect reflection by design, it is not going to include another behaviour, such as bounces. This study is going to discuss three main areas of light behaviour that are vital to successfully model realistic light behaviour.

4.2.1 Reflections

Reflection is the main focus of GI and has to be calculated accurately for the desired realism of the scene. There are different types of reflections and while modelling the virtual environment the material settings determines these types.

These types can be

- mirror reflections
- metallic reflections
- blurry reflections
- glossy reflections [24].

These reflections are shown in the same order from left to right on the spheres in Figure 20.

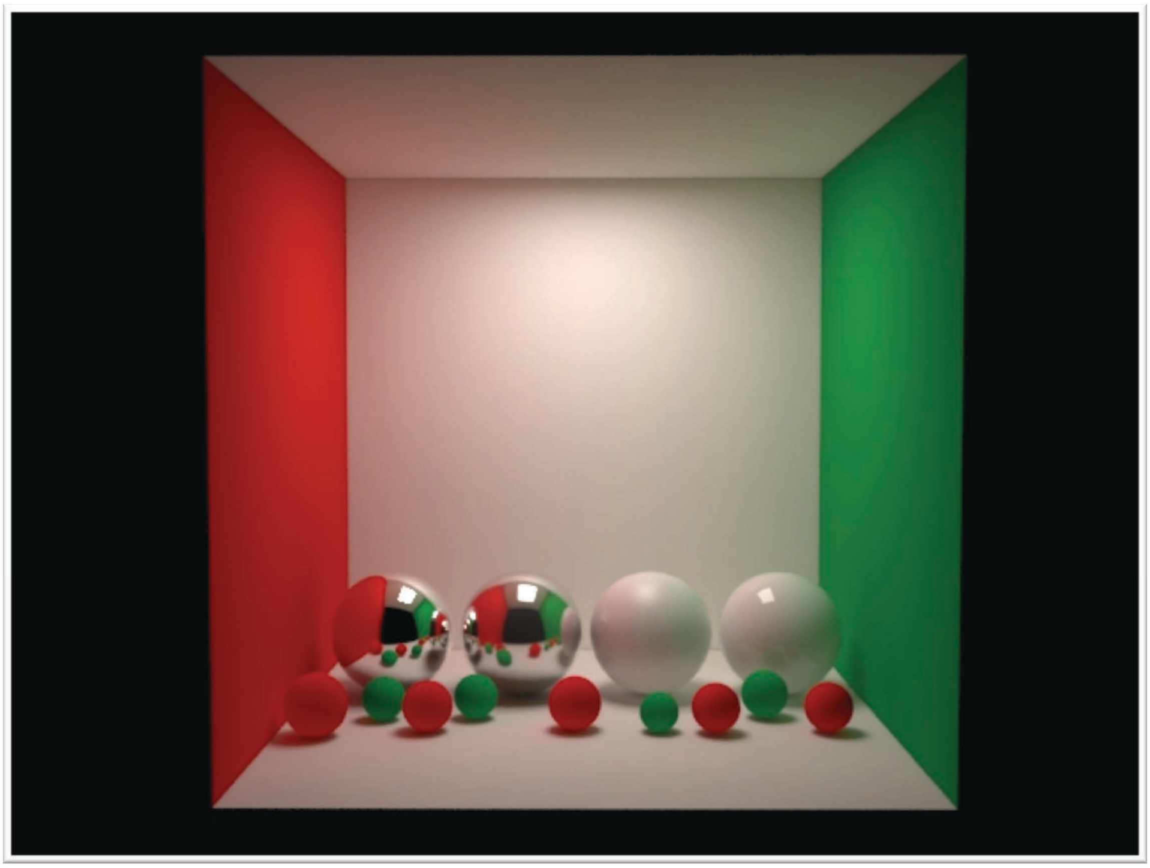


Figure 20. Different types of reflections.

Reflections can be categorised also as specular (reflected in one direction) or diffuse (reflected in all directions). [25] All of these have to be calculated precisely and accurately to mimic reality in the VE.

4.2.2 Light Bouncing

In 3D computer graphics, light bouncing or scattering is commonly referred to as diffuse reflection. It is due to the behaviour of light that while it is reflected it also spreads over a wide area. When the surface is smooth, light reflects back at an equal angle to its incoming direction and the resulting reflection is called a specular reflection.

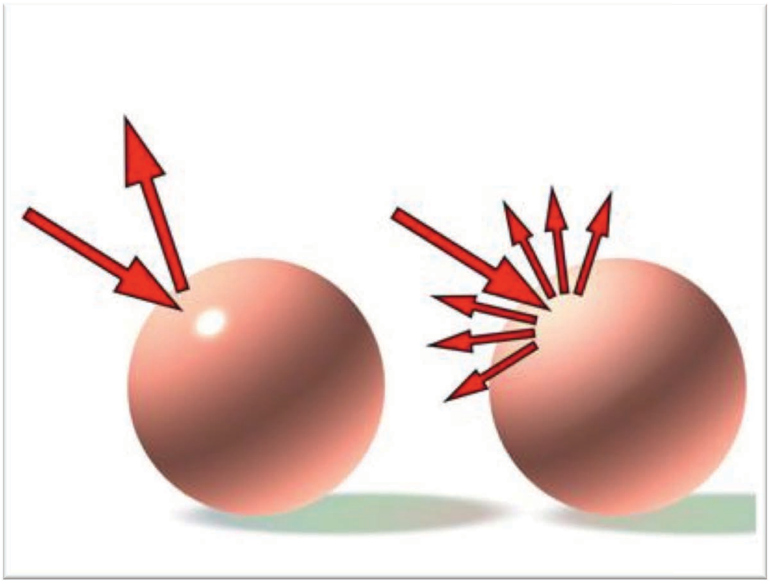


Figure 21. Specular and diffuse reflection. [25]

When the surface is rough light is reflected in all directions. While the diffusion, scattering or bouncing of light is also a reflection. It is calculated by different algorithms than the specular reflections; thus it is critical to be aware of the difference here. [25] In Figure 21 the first sphere shows specular reflection while the second sphere shows diffuse reflection.

It is important to note that CG softwares may not use the terms “specular” or “diffuse” for their accurate meaning. Often these terms are used as pointers to values that can be tuned, even if it is not physically accurate, for instance when the goal is to achieve some effect.

4.2.3 Ambient Illumination and Occlusion

Ambient illumination is the approximation of light reflected from objects and ambient occlusion is the approximation of the effect of darkening when light is blocked [26]. While ambient occlusion is only a crude approximation of real global illumination, it can be very useful to set up and calculated to add additional depth to the materials in the scene. Ambient illumination and occlusion are not real global illumination algorithms in that they are not based on reality.

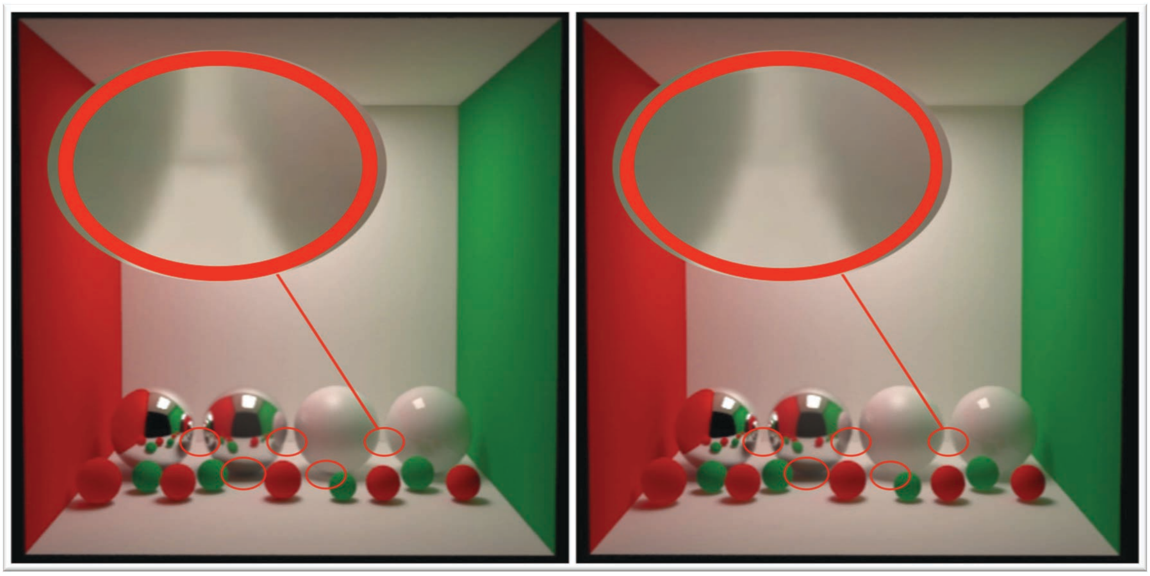


Figure 22. Render with (left) and without (right) ambient occlusion.

It is, however, commonly recognized that this method enhances the final renders and improves on the results that are yielded by GI algorithms. Sometimes the difference is subtle, as it can be seen in Figure 22. However, a slight darkening under the third sphere from the left is noticeable when the renders are compared. When ambient occlusion is calculated, a small darkening in the shadow occurs under the white sphere while, without ambient occlusion, the white sphere in the second image looks more uniform.

The enlarged circles show a very subtle difference at the edge. Often these small differences are neglected but the human brain notices very small differences even if its not consciously aware of it. These subtle differences can be the difference between the acceptance of a project or its rejection.

5 Methods for Global Illumination

5.1 Scanline Rendering

Global illumination is a concept that can be achieved in multiple ways. The default render engine in 3ds Max is the Scanline Rendering. This render engine has improved over the years to the point that in a limited way it supports GI as well. This render engine is fast, but the results are not as accurate to realism as those made with dedicated render engines. [20] Scanline rendering supports GI via two principal algorithms; radiosity and ray-tracing. Radiosity is responsible for diffuse light interreflections in other words for indirect light. Ray-tracing is enabled by default for reflections and refractions and advanced ray-traced shadows are given as an option. These two algorithms are going to be demonstrated on their own. It is important to note, however, for optimal renders the two algorithms should be run complementing each other.

5.1.1 Radiosity

In conjunction with using photometric lights radiosity aims to simulate real world light behaviour, such as indirect light and colour bleeding. Radiosity calculations are shooting calculations meaning that they are calculated from the point of origin (light source) in contrast to gathering calculations that are calculated from the observer's view (camera).

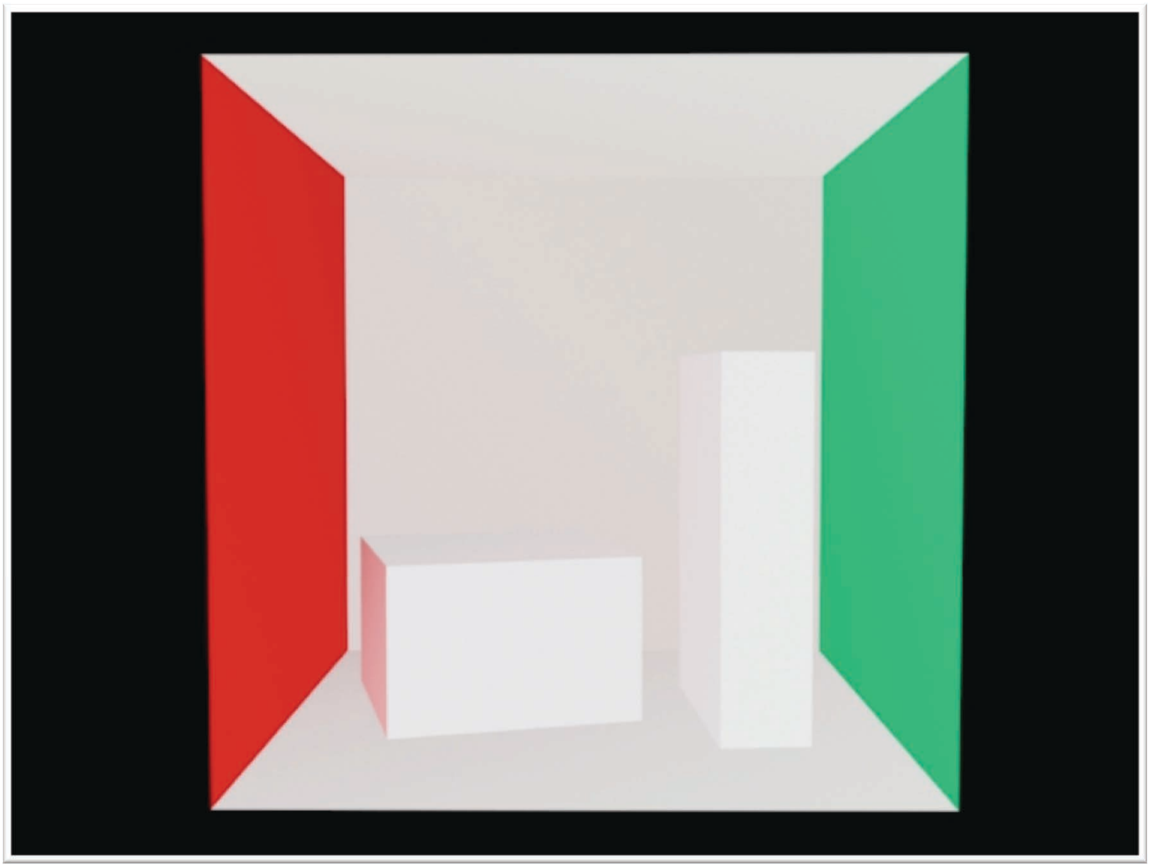


Figure 23. Render using Scanline rendering and radiosity.

In Figure 23 colour bleeding onto the left box can be clearly detected. Shadows are turned off since the ray-tracing algorithm is not run, and standard shadows are not enabled. Radiosity can be applied via the Advanced Lighting tab of the Render Set-Up window.

5.1.2 Ray-Tracing and Shadows

Ray-tracing is a gathering algorithm meaning that it calculates light paths from the point of view of the camera. The Ray-tracing global parameters allow to turn it on or off and also to tweak its settings. It is, however, on by default and responsible for reflections and refractions. Turning it off would result in no reflections or refractions with the expectation of materials that has been set up with ray-traced materials. The "Advanced Ray-traced shadows" parameter is responsible for generating realistic shadows. Shadows can be generated without ray-tracing, but shadows calculated like that will not have real correlation to reality.

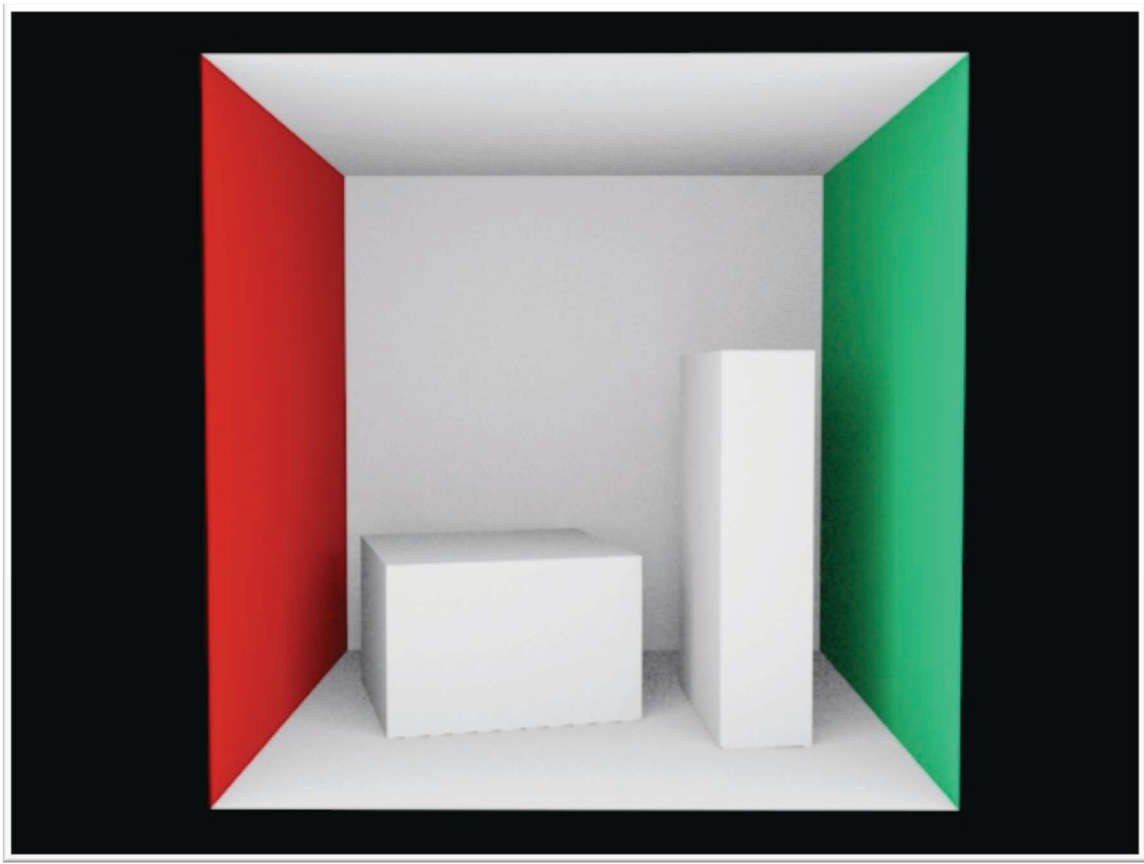


Figure 24. Render using Scanline rendering and ray-tracing.

In Figure 24 colour bleeding does not occur since the radiosity algorithm is not run. Ray-tracing shadows can be specified not in the render options but as a method to render shadows in the light sources. [27]

5.2 Mental Ray

In the Mental ray render engine used in the CG software 3ds Max, there are multiple options to achieve the effect of GI. Some of these options can work on their own or to be more effective; side-by-side. Mental ray is more advanced than the Scanline rendering engine, and it supports a wider variety of GI algorithms than the default Scanline rendering. While some of the GI calculations can be freely specified there are defaults that cannot be changed. Skylight is such a default algorithm and thus, cannot be turned off. This fact hinders somewhat the effort to compare the power of different algorithms within this render engine.

5.2.1 Skylights

The Skylights method is used to simulate skylight either by deriving it from a skylight light object using IBL or by generating it through the calculations of the Final Gather algorithm. [28]

As it can be seen in Figure 25, the Skylights algorithm on its own - when calculated from a skylight light object - produces an unrealistically light-flooded scene with very unnatural shadows. This effect can easily be observed at the left side of the second box where the shadow is cast from the right box. The shadow at the floor level leaks under the second object and there is light at the bottom of it.

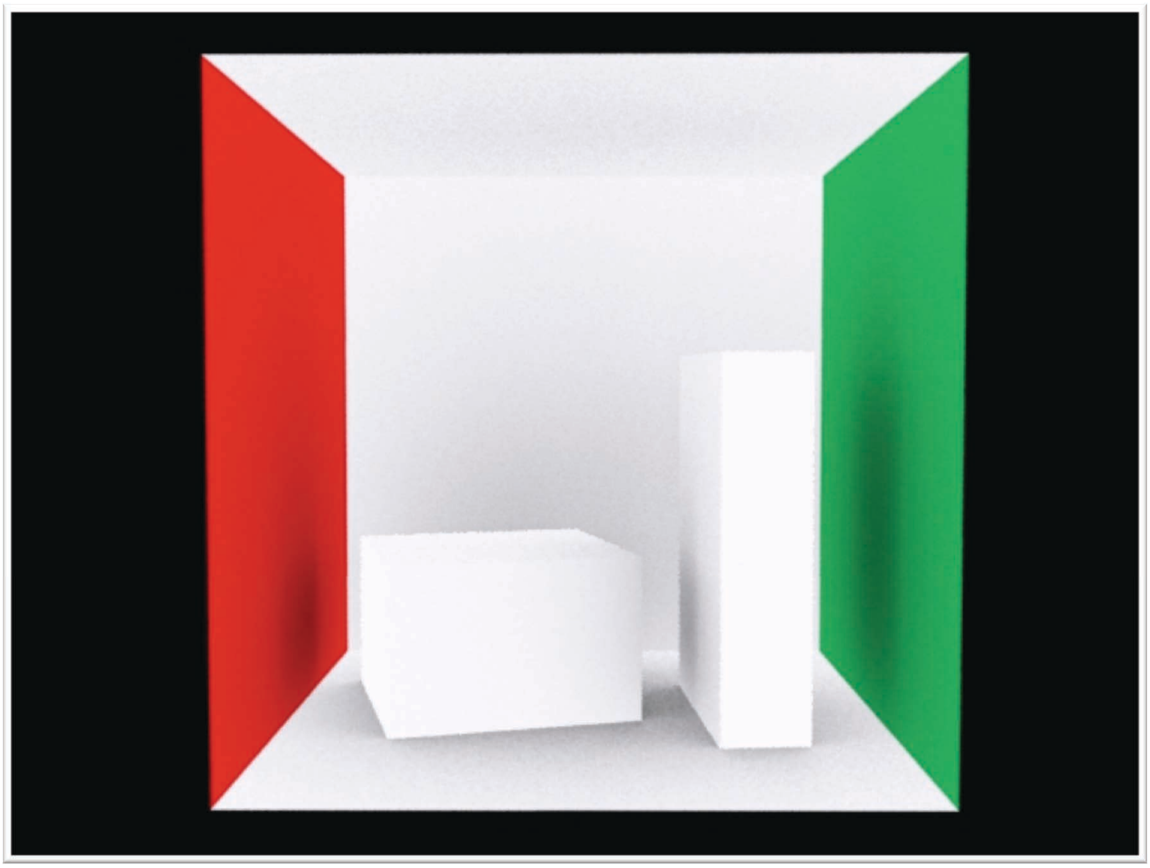


Figure 25. Render using mental ray and skylight.

Even though the light leaks in the scene are subtle, this algorithm run on its own demonstrates how unrealistic the results can be when applying only one type of GI calculation.

5.2.2 Final Gather

Final gather is an algorithm in mental ray that is based on ray-tracing, but focuses on light bounces or diffuse reflections. Final gather shoots rays or points randomly on objects and measures their colour and intensity. It is a gathering algorithm as its name suggests so the rays are shot from the camera's perspective, regardless of the lights position on the scene. [28] Because the shooting of rays is random, Final Gather is not the best choice for animation, since for every frame a different set of points will be measured and it can lead to unrealistic effects.

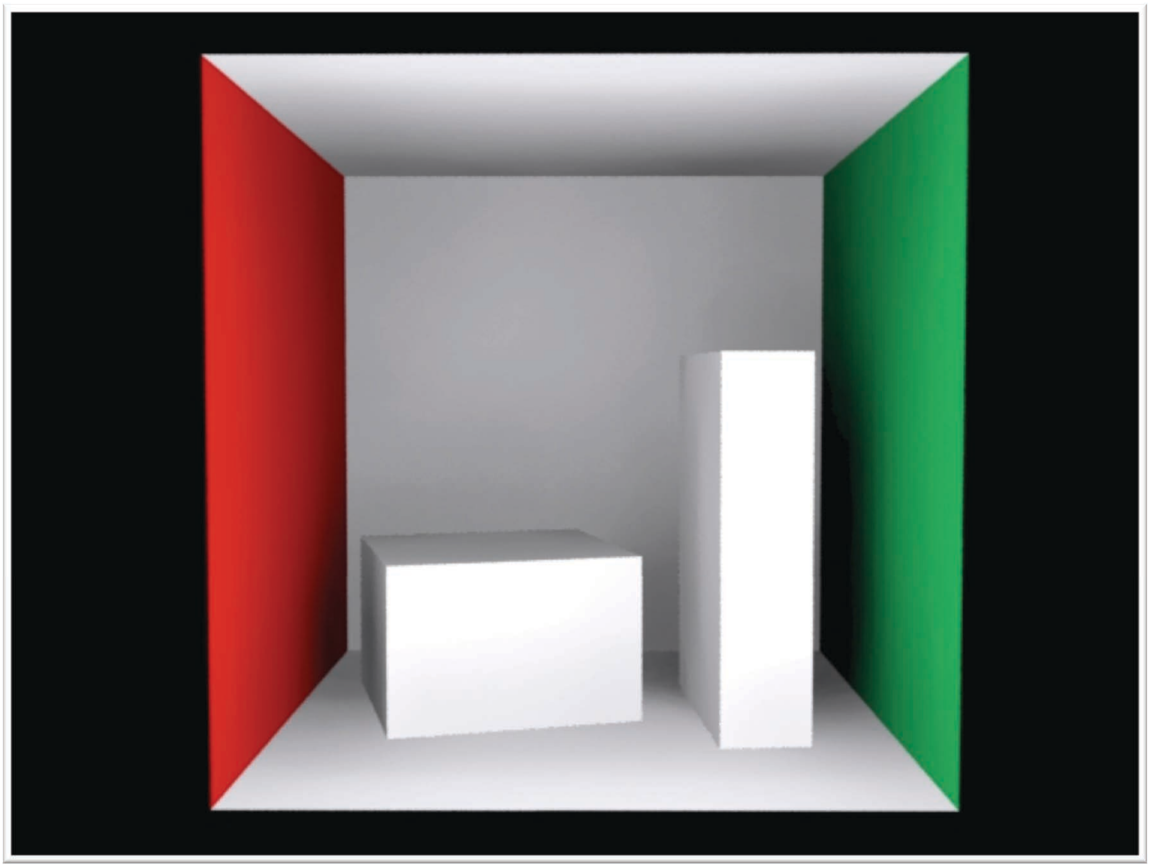


Figure 26. Render using mental ray and final gather.

In Figure 26 the realism of the scene is greater than it was in Figure 25, even though Skylights as a default calculation is still run from the skylight light object. It is an algorithm that is enabled by default when using the global illumination option in this render engine, but canalso be turned off. [28]

5.2.3 Caustics

This method calculates the caustic effect via photon mapping. Caustic effect is mostly visible when light shines on a glass of water, for instance, causing light to be collected and reflected off the glass. [28]

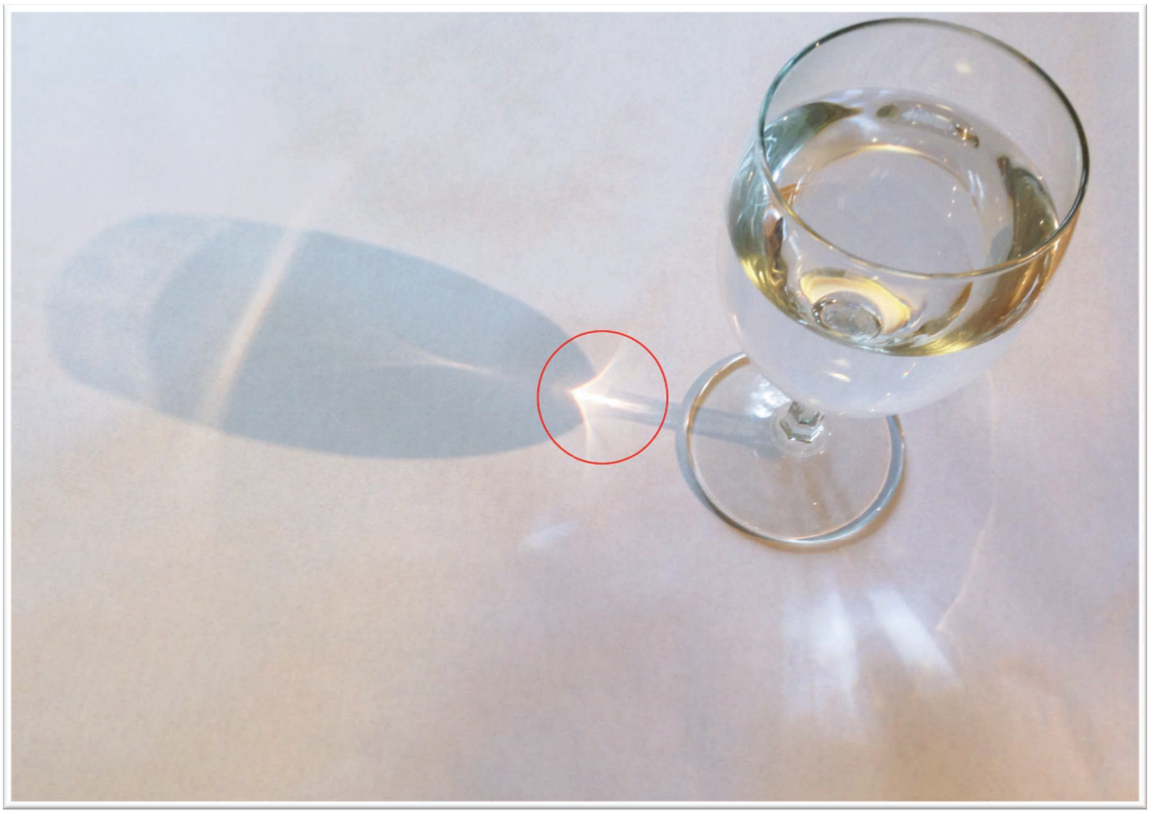


Figure 27. Caustics effects on a real glass of water.

In Figure 27 caustics effect can been seen in reality. The picture was taken on a cloud-covered day at noon inside, but caustics still occur. The water and glass focuses the photons like a prism or a lens.

While there are no liquid or glass objects on the scene, basic photon mapping still occurs as it can be seen in Figure 28. This render uses only the Skylights algorithm as a default and the Caustics algorithm so that Final Gather is turned off. The result is clearly different from using only the Skylight calculation and, therefore, photon mapping can be seen. Shadows look more realistic. Multiplying the value of caustics would result in a clearer and lighter render.

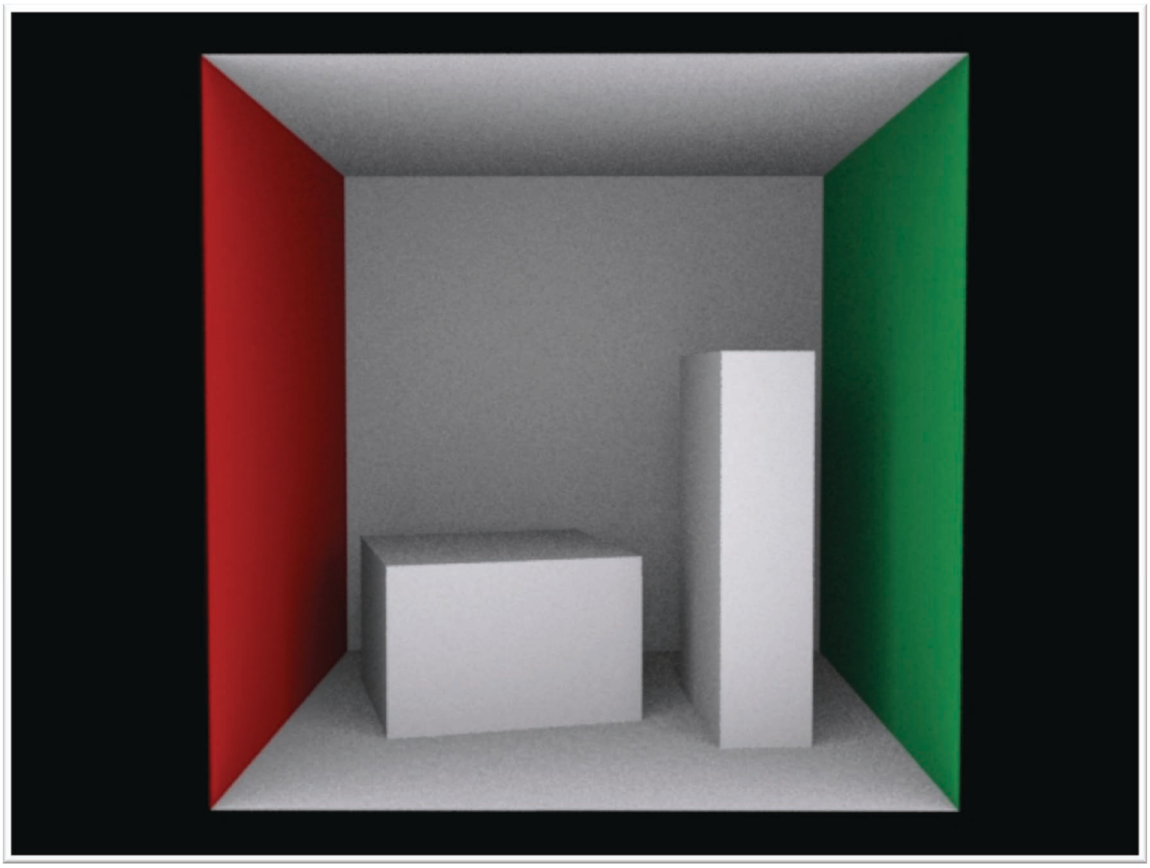


Figure 28. Render using mental ray and caustics.

Since caustics have a very narrow field of use, this GI algorithm is not useful in scenes that do not contain objects that would create the caustic effects. Caustics may also calculate diffractions. Because the main focus of this algorithm is caustics with normal values, photon bounces are not followed excessively, and this explains the darkness of the scene. The algorithm focuses on areas where photon density is high enough for caustics to occur.

5.2.4 Photon Mapping

Photon Mapping calculates the bounces of the photons coming from the light sources. This algorithm by default has higher values for following photon paths than caustics and it results in a lighter scene even though it uses a similar algorithm. Sometimes this algorithm is sufficient to mimic basic caustic effects so that - given the emphases is not on this particular effect - Caustics can be neglected. [28]

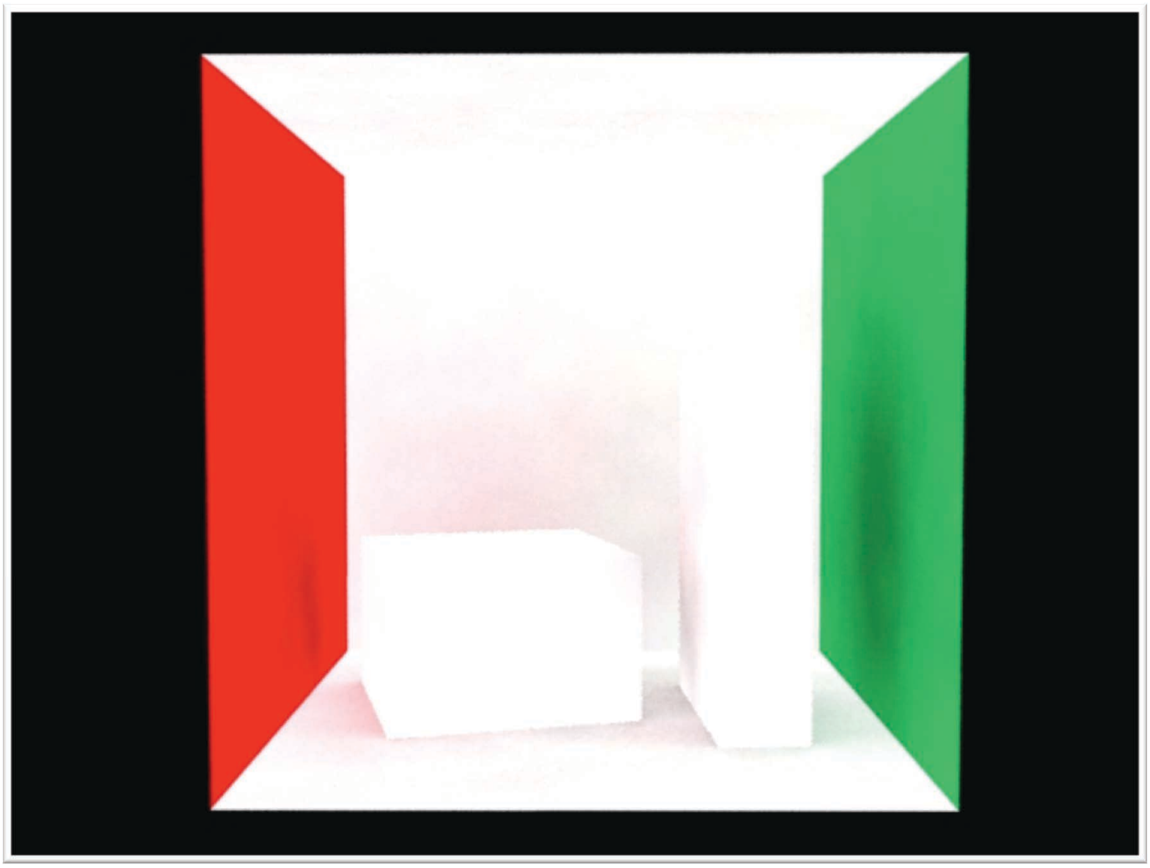


Figure 29. Render using mental ray and photon mapping.

In Figure 29 light is excessive, and shadows are not realistic. This algorithm is not capable of creating realistic scenes by itself.

5.2.5 Ray-tracing

Ray-tracing is essentially the technology used to calculate reflections in the mental ray render engine. While true ray-tracing is capable of simulating multiple effects, such as reflections, refractions, light bouncing and scattering, in mental ray it primarily focuses on reflections. In the current Mental ray render engine it is enabled by default among the render algorithms.

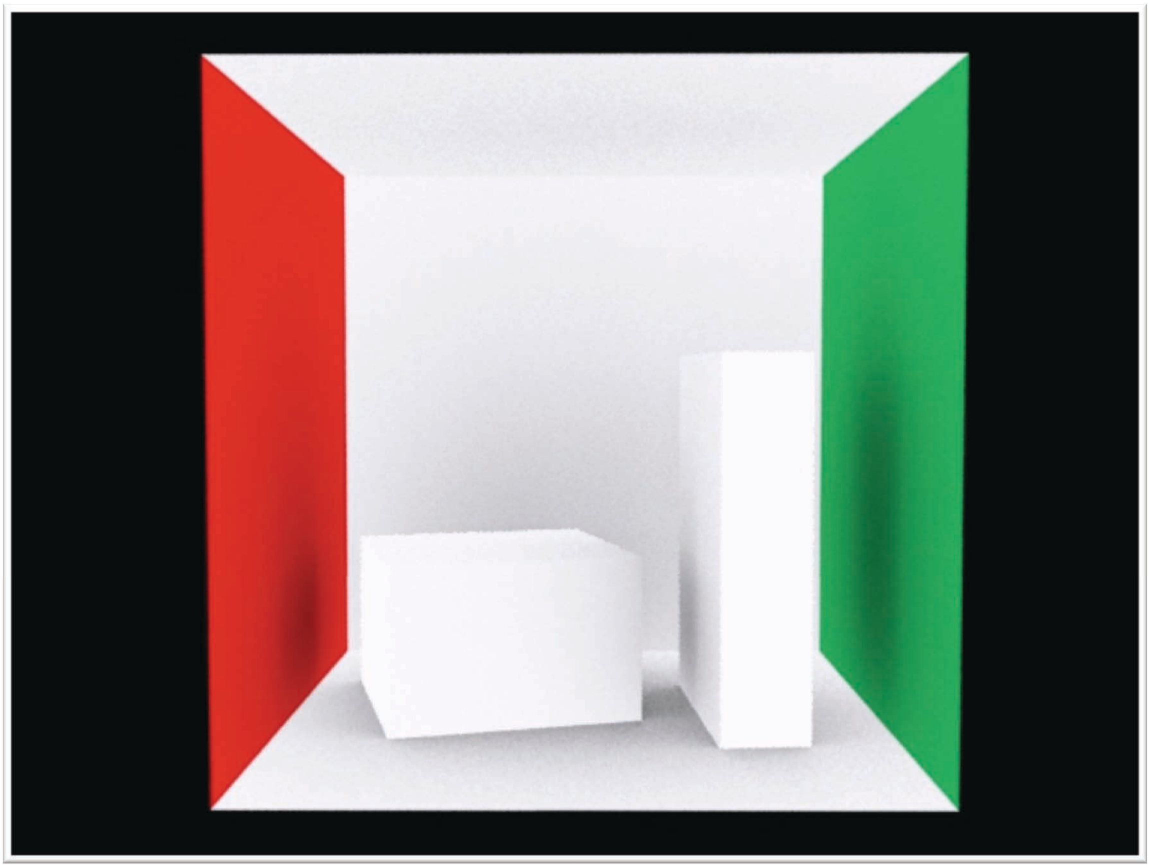


Figure 30. Render using mental ray and ray-tracing.

Ray-tracing suffers similar artefacts as using skylight alone or photon mapping, but it results in ambient occlusion style sharp edges. That is the reason why ambient occlusion is considered a secondary priority. This effect, however, combined with other algorithms becomes less pronounced and, therefore, ambient occlusion is a calculation to be considered.

5.2.6 Ambient Occlusion

Ambient Occlusion is not part of the group of real GI algorithms, however, in architectural visualization it can be vital. Ambient occlusion is a simplified calculation and does not reflect reality. It is generally thought that in case, final gather is used, ambient occlusion can be left out since final gather is supposed to calculate similar effects as ambient occlusion. [29] In practice, however, ambient occlusion and final gather complement each other. Final gather improves the realism of the scene by adding light distribution of indirect light and colour bleeding while ambient occlusion will enhance the shadows at the edges of objects. [30]

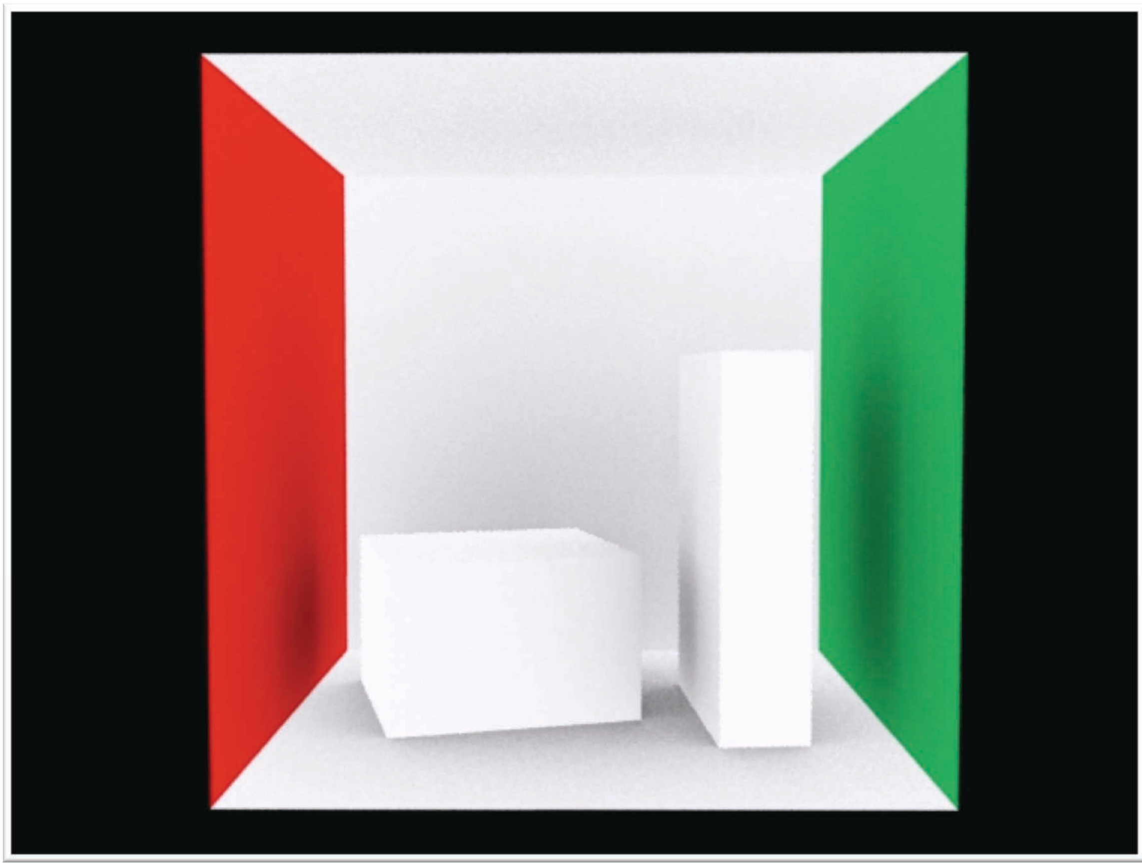


Figure 31. Render using mental ray and ambient occlusion.

In mental ray, ambient occlusion is a render element that can be added in the render setup. Once the scene is rendered an additional file that can be imposed onto the rendered image is created. The additional file contains the ambient occlusion information. To merge the two images, other graphics software - such as Photoshop - is generally used. The result can be seen in Figure 31.

5.3 V-Ray

In the V-Ray rendering engine, GI is referred to as indirect illumination. V-Ray uses two passes to calculate global illumination, primary bounces and secondary bounces. In addition to this, ambient occlusion can also be turned on under the "Indirect Illumination" option in the render setup. To compare the results of different GI algorithms within the V-Ray render engine, the methods are used on their own.

5.3.1 Brute Force

The Brute Force method computes global illumination for all surfaces independently via ray-tracing. The result seen in Figure 32 is very good for only one algorithm being used.

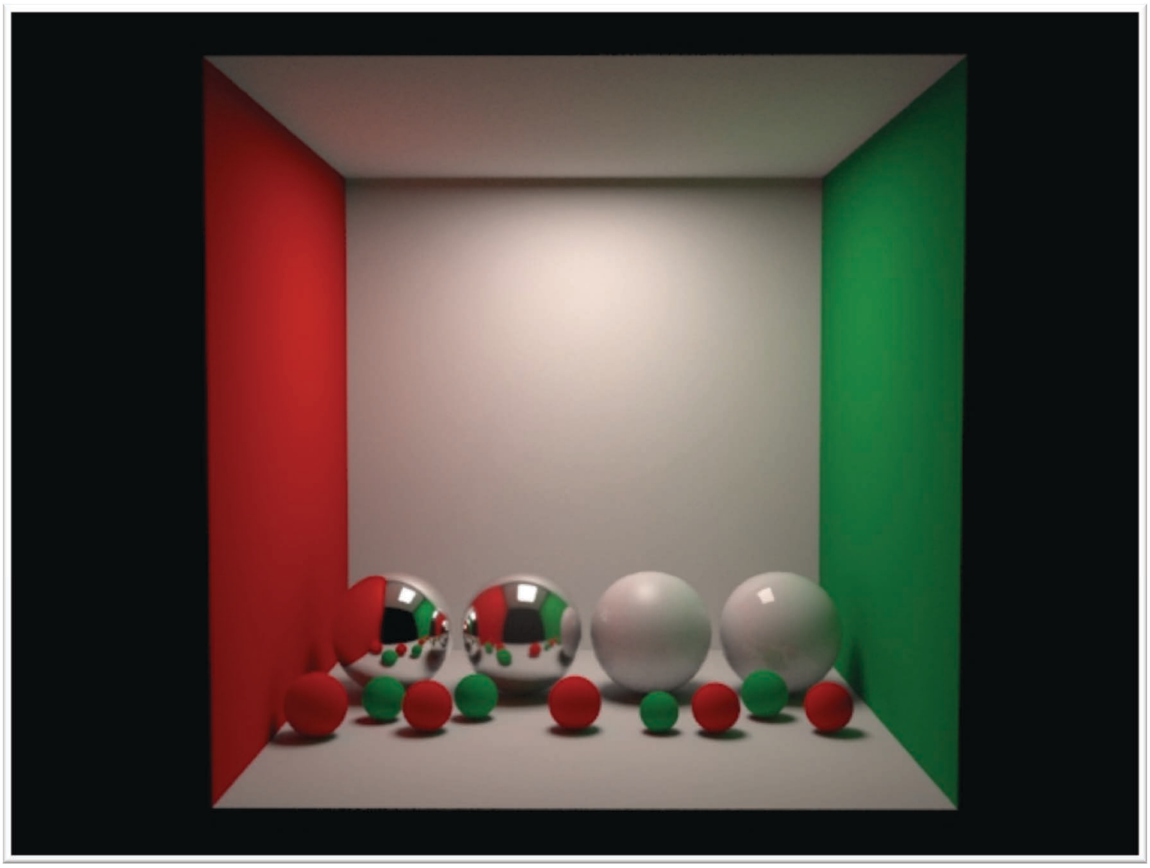


Figure 32. Render using v-ray and brute force.

This method is slow and may produce artefacts on the rendered image, but usually the result is smooth and realistic. [31]

5.3.2 Irradiance Map

The Irradiance Map method computes GI only for certain points in the scene and then constructs new data points within the range of the computed values.

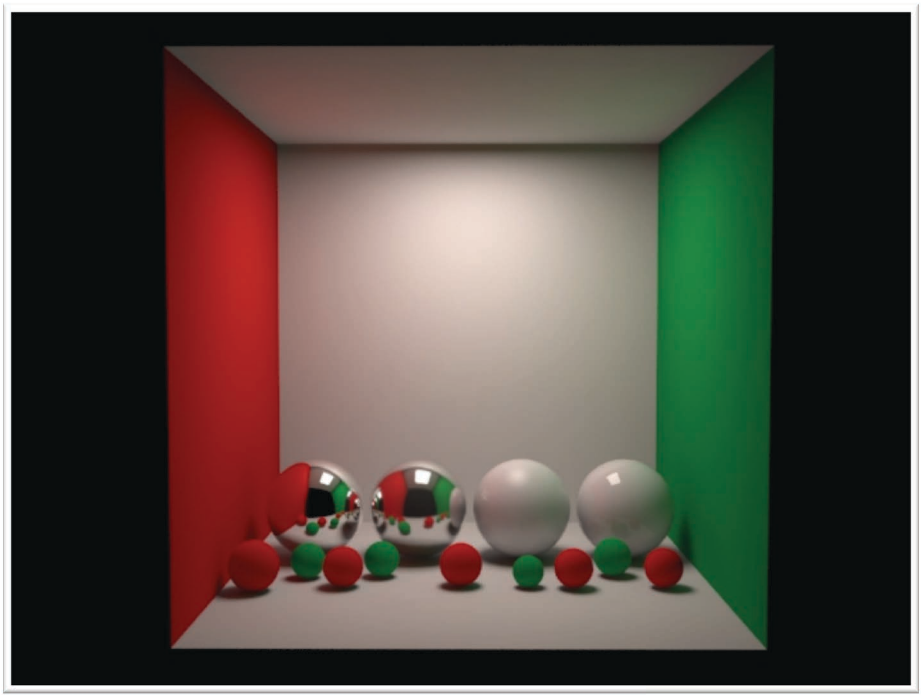


Figure 33. Render using v-ray and irradiance map.

It makes the irradiance mapping very effective and fast with the trade off of accuracy. It can be seen by comparing Figure 32 and Figure 33, the resulting images are nearly identical. [31]

5.3.3 Photon Map

As the name suggests this method traces photons or particles from the light source and calculates their bounces. This algorithm is best used with photometric lights.

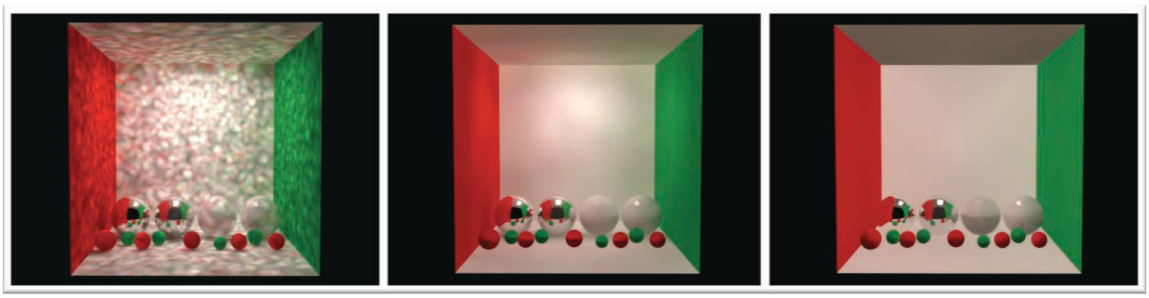


Figure 34. Render using v-ray and photon map with different settings.

Photon map with basic settings results in an awkward scene as can be observed in Figure 34. The basic settings work with 30 photons and one bounce (left). If the number of photons is raised to a higher number such as 10000 and the number of bounces are set for 3, the resulting render shows high quality reflections, but the shadows disappear and the image looks flat (right). Lowering the bounces of photons will flatten the image, while raising the number of photons will eradicate light patches together with shadows. A good compromise is to keep bounces high - 10 - and photon numbers moderate - around 1000 (image in the centre).

Compared to other methods that are gathering methods, photon mapping is a shooting method. Gathering methods start from the camera, and shooting methods start from the light source. This method can simulate caustics in a limited way. [31;32]

5.3.4 Light Cache

The Light Cache method is similar to photon mapping but instead of being a shooting method, it originates from the camera. A big advantage of the algorithm that it is able to use any standard or photometric light source compared to photon mapping that prefers photometric lights. Light cache is easy to set up and can be pre-run to calculate from the view of the selected camera or even to calculate for all the cameras. Light cache's ability to calculate for every camera makes it a prime candidate algorithm for animations. It is important to note that Light cache used on its own created unrealistic image and may distort optically the geometry as shown in Figure 35.

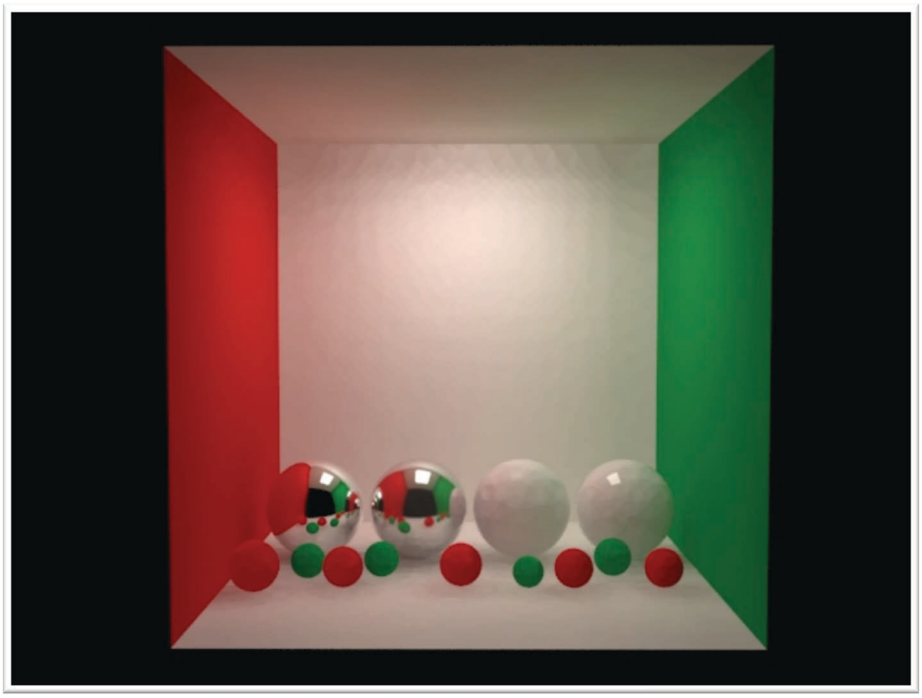


Figure 35. Render using v-ray and light cache with v-ray materials.

Using only light cache will produce splotches, however, combined with other algorithms it can be very effective, because it is a very fast process. The big disadvantage of the light cache is that it works only with V-Ray materials, meaning that objects in a virtual scene have to be assigned a V-Ray material. [31]

5.3.5 Ambient Occlusion

Ambient occlusion is an on/off option at the GI rendering setup in the V-Ray render engine. Its amount and radius can be set as well the number of samples to be taken for the calculations. [31] Ambient occlusion cannot be run by itself without setting at least one GI algorithm.

5.3.6 Caustics

Caustics are also an on-off option at the rendering in the V-Ray render engine. It uses a photon mapping algorithm and is a highly time consuming calculation. It has no real benefits if the main focus in the scene is not on the objects that cause caustics and the

Photon Map or Light Cache render method is in use. [33] Caustics cannot be run on its own.

6 Case Study on Global Illumination

6.1 Solution with Scanline Rendering

A Nokia headset in a Cornell box has been chosen for the case study. [34] To compare the different GI algorithms a standard photometric free light, standard materials and a default camera was set up. To better illustrate GI, the model was placed in a Cornell box with two of its sides coloured and the rest textured to mimic cardboard.

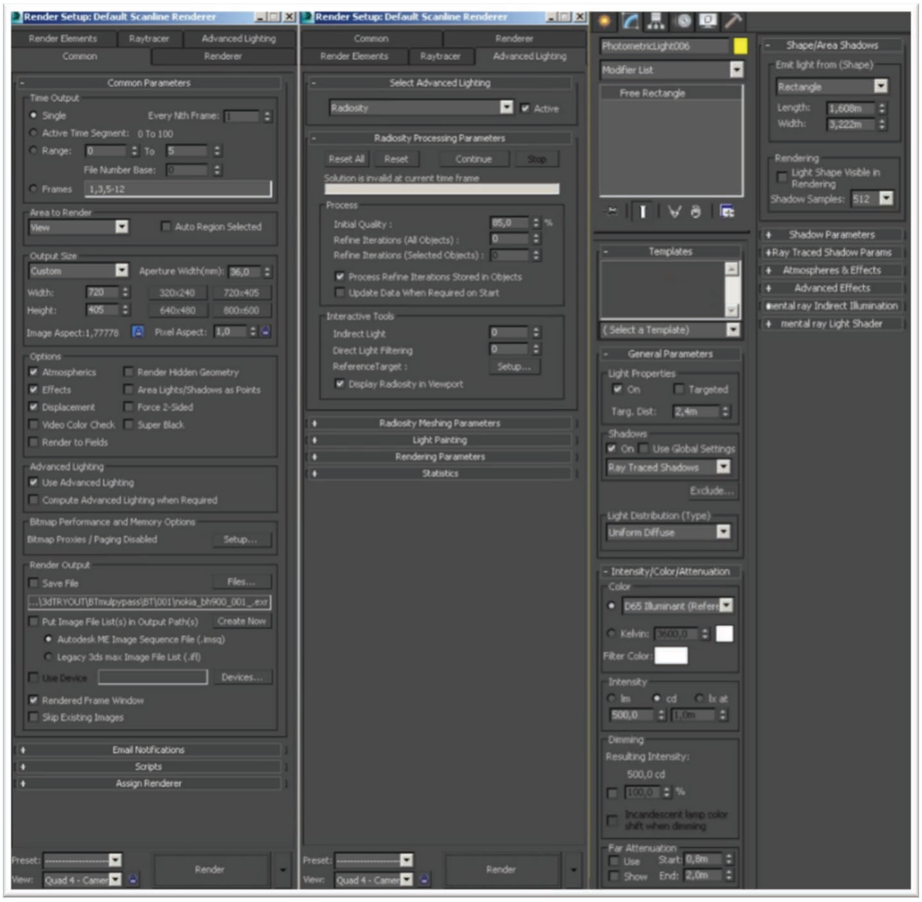


Figure 36. Settings used in Scanline Rendering.

To render with Scanline Rendering, shadows were set to be ray-traced. To use GI in Scanline rendering the global ray-tracing was left on default and Advanced Lighting was enabled. Within the advanced lighting Radiosity was chosen as the algorithm to calculate photon behaviour. Scanline rendering has no other GI capable algorithms. Radiosity once enabled has to be pre-calculated before the rendering; however, it is done only once regardless of the camera's position since it is a shooting calculation. Once radiosity is calculated, an unlimited number of renders can be made from any camera view until the scene is unchanged. All settings can be seen in Figure 36. The final render can be seen in the Appendix.

6.2 Solution with Mental Ray

In Mental Ray three algorithms were used with the same scene settings as previously described.

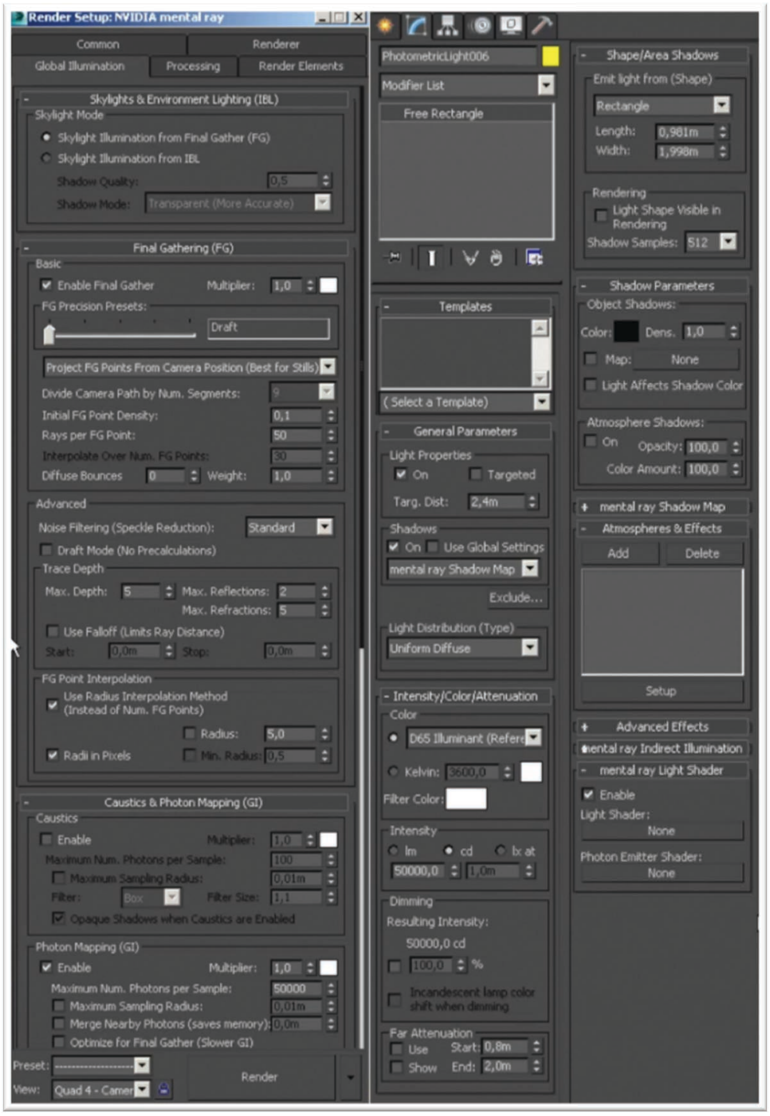


Figure 37. Settings used in Mental Ray.

The Skylight and Environmental Lighting is a default algorithm that cannot be turned off. Since the scene did not contain environmental lighting other than the one light source above the object, this algorithm was set to calculate its values from the Final Gather algorithm.

Final Gather and Photon Mapping calculated photon behaviour. Final Gather Point Interpolation has been set to Radius Interpolation and the number of photons per sample to 50 000. Under 5000 photons the render starts to have excessive light splotches. Mental Ray had a very difficult set up process and after multiple trials the resulting render looked acceptable. Using Final Gather without Photon Mapping resulted in a black void, while using Photon Mapping alone resulted in excessive and unrealistic light. Unfortunately Mental Ray was unable to use standard materials correctly. Multiple trials were undertaken to correct this to no avail. All settings can be seen in Figure 37. The final render can be seen in the Appendix.

6.3 Solution with V-Ray

With the V-Ray render engine the same scene and setting were used. For GI calculations Irradiance map and Light cache has been set up to calculate primary and secondary bounces respectively. While the resulting render was of a high quality and realistic, the edges of the box were less enhanced compared to the other two engines. To overcome this difficulty Ambient Occlusion had to be turned on.

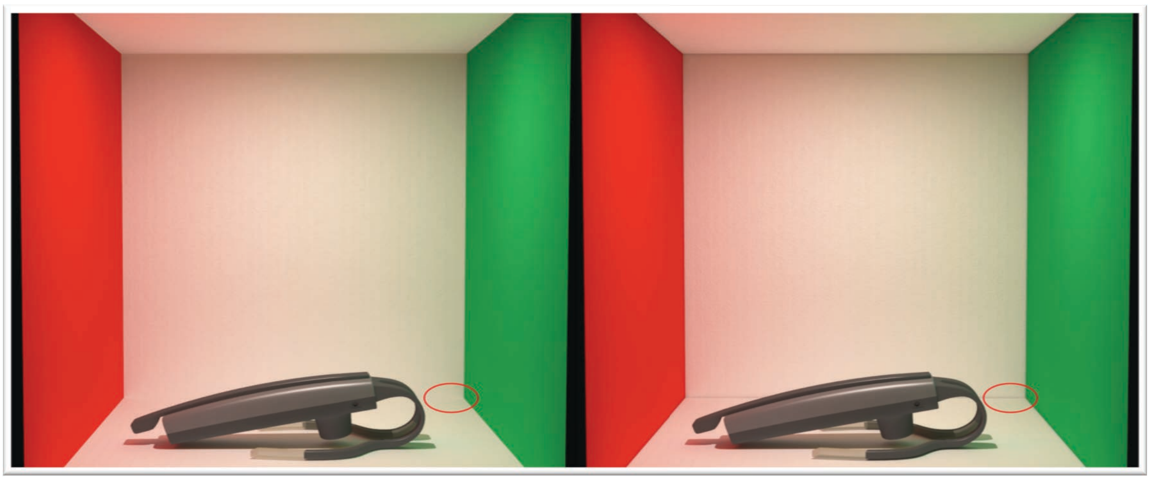


Figure 38. Final render without (left) and with Ambient Occlusion (right).

As it can be seen in Figure 38 marked by a red circle, the edges are nearly invisible on the render that was created without using Ambient Occlusion. The V-Ray render engine used V-Ray shadows since it cannot operate with standard or ray-traced shadows.

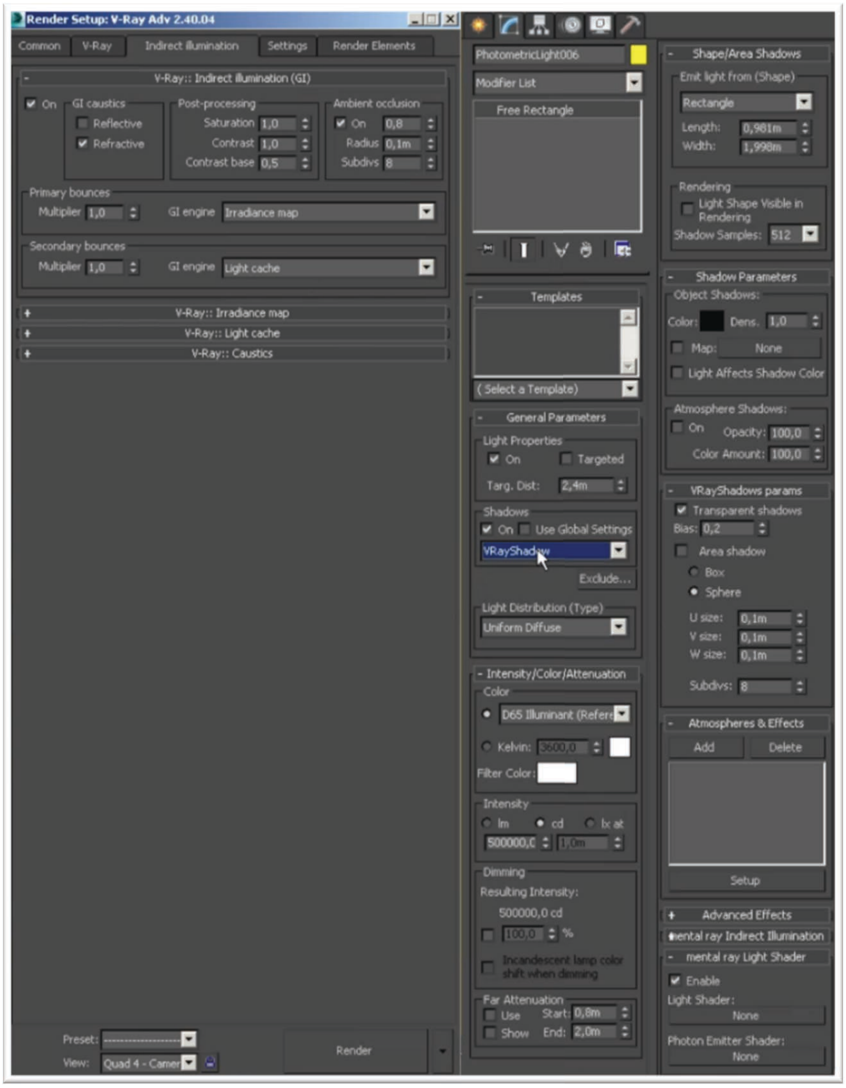


Figure 39. Settings used in V-Ray.

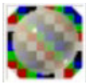
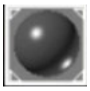
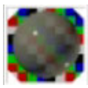
Settings for Ambient Occlusion have been left on their default. GI caustics has been set to be refractive, since there are no objects capable of causing caustics on the scene. All settings can be seen in Figure 39. The final render can be seen in the Appendix.

7 Summery

7.1 Materials

In the case study to be able to compare purely the different algorithms that are used for GI the same material, camera and light settings were used. On the case study scene there are 7 materials, all of them are Blinn shaders. The Blinn shader is a material shading type. A shader is an algorithm that specifies how surfaces interact with light [35]. The Blinn Shader is the most commonly used shading type for materials and was developed by Jim Blinn. It is also an OpenGL standard. [36]

Table 1. Materials used in the case study scenes.

Name	**Image**	**Colour**	**Spectacular Level**	**Glossiness**	**Amount of Transparency**
Material 1			120	36	0
Material 2			125	72	56
Material 3			110	46	0
Material 4			333	78	0
Material 5			97	26	0
Material 6			127	79	0
Material 7			0	10	0
Material 8			0	10	0
Material 9			0	10	0

In Table 1 on material 7 a spectacular and bump map has been defined from an image file. The image file is a cardboard texture. Defining these maps requires no difficult set up other than browsing for the image file.

The idea was to depict the scene to appear as a coloured sided cardboard box containing a Nokia headset.

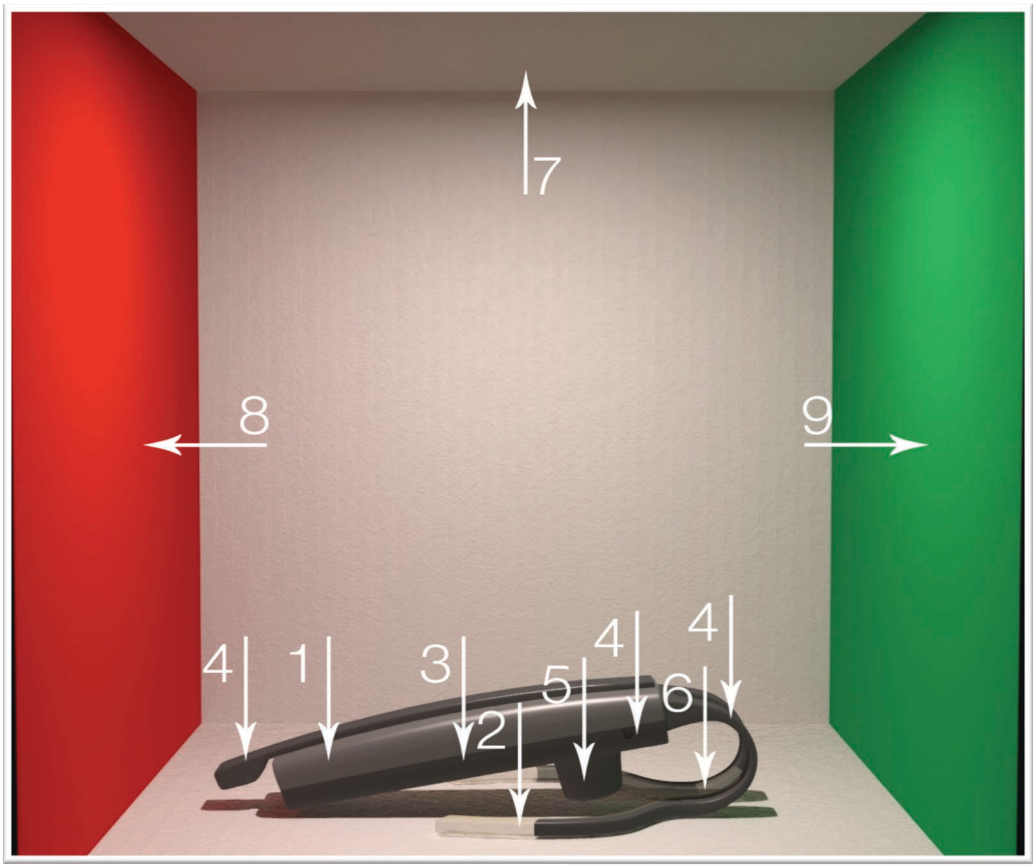


Figure 40. Placement of used materials to compare with Table 1.

Figure 40 shows the placement of different materials used on the model. Some of the materials were used on multiple surfaces, as it would be expected in real production of a similar object as well.

7.2 Lights and the Camera

There is only one photometric free light used in the scene. The luminous intensity of the light has been defined in Candela (cd) to be 500. However, using different render engines the 500 cd proved to be insufficient for all 3 engines. For Scanline Rendering 500 cd worked sufficiently, for Mental Ray 50 000 cd was required to illuminate the scene, while V-Ray required 500 000 cd.

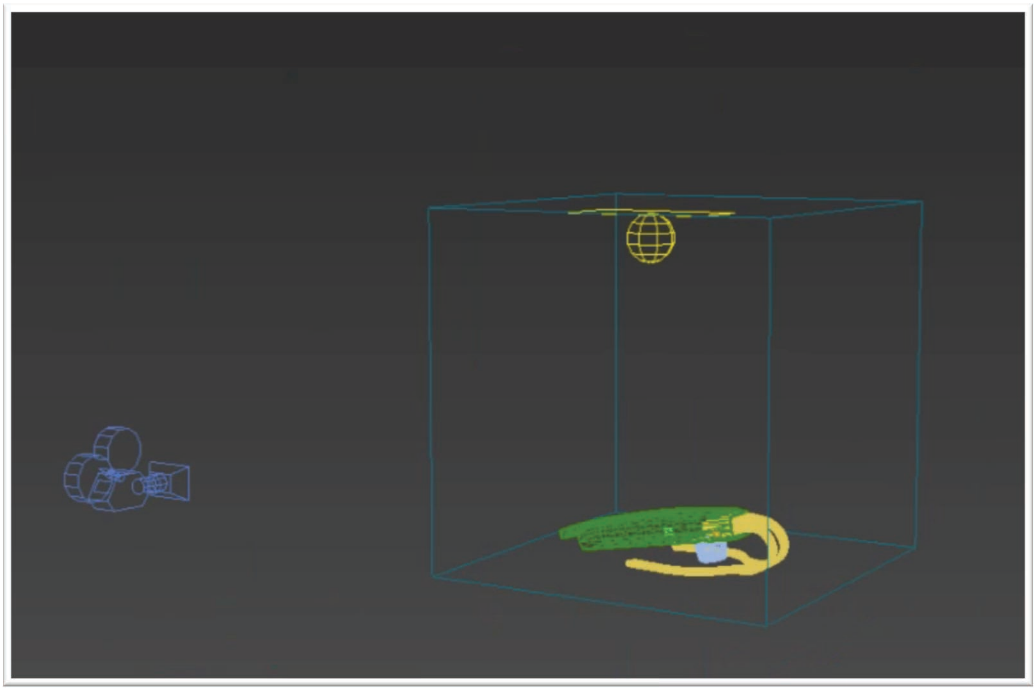


Figure 41. The light and the camera in the scene.

The light is directed downwards to the object and its size is roughly half of it. The light is placed in the centre from left to right and to the 2/3 of the front on the top face of the box. The exact placement of objects is displayed in Figure 41. While it was necessary to change the intensity of the light, its placement remained the same on every render.

There are different settings for the camera and some render engine have their own camera that mimics the distortions, aperture and colour management of real physical ones. The camera used to observe the scene is a default camera with default settings and was unchanged throughout the case study.

7.3 Results

The case study can be regarded partly as a failure and partly as a success. While the different GI's working mechanism is well demonstrated and an optimal light scene has been achieved, the illuminated object is far from being photorealistic. The main areas of interest in comparison are colour bleeding, edges, shadow, material use and time. To render the images a PC with 16,0 GBs of DDR 3 1866 MHz RAM and an Intel Core i7-2700K 3.50GHz processor were used.

Materials play a vital role in CG and different render engines use different materials. Scanline Rendering with its standard materials yielded an acceptable result where the specular highlights could be seen clearly and the materials gave a somewhat realistic impression. Transparent materials behave as expected and textures were also handled appropriately. The two huge coloured walls produce a bit of colour bleeding that can be seen in the beautifully sharp shadows and in the soft light just next to the coloured surfaces. The edges of the box are clearly visible and distinguishable. A bit of light leaks into the back of the object shadow that indicates the position of the light source accurately. However, keeping in mind the size of the coloured areas, the colour bleeding is unfortunately minimal. The rendering process, on the other hand, was very fast and took only a minute and 17 seconds for a render of 1920x1080 pixels.

Mental Ray produces very light and soft shadows, but dramatically more colour bleeding. Edges are clearly visible. Shadow positions are also accurate but surprisingly soft, despite all efforts to correct it. Material colours are used, but properties are not retained, thus no reflections or transparencies are displayed. While Mental Ray seems to correctly calculate light behaviour when it comes to colour bleeding, it fails to make use of standard materials and textures as well as it over-softens shadows. Edges are also correctly calculated and beautifully visible. This render engine should be used with its own materials for optimal results. The render time for mental-ray was a devastating 4 minutes and 55 seconds, making it the least effective of the three engines compared.

V-Ray combined the advantages of Scanline Rendering and Mental-Ray. As it can be seen in the Appendix, the render created by V-Ray has a realistic colour bleeding on surfaces and in shadows. The shadow at the back of the object lightens, indicating clearly the position of the light source. Materials are used correctly and transparency of the plastic placed on the cardboard is more natural than it was with Scanline Rendering. Edges, however, are near invisible without Ambient Occlusion. When Ambient Occlusion was turned on, edges became acceptable. The rendering time was considerably more than Scanline rendering, but much less than Mental Ray with a surprisingly good result. The total time to render the full HD (1920x1080 pixels) image was 3:38 minutes.

While a minute here and there might not mean a lot for the scope of this study, when it comes to rendering complex, even district sized models into images, the render times will grow exponentially. With growing render times the time difference grows as well. These are real factors that need to be considered in architectural visualization. Modelling and illumination the scenes can take weeks and months. A poorly chosen render engine might add days or even weeks of idle waiting for the final renders.

8 Conclusions

Photon behaviour in Virtual Environments is an important area to focus when creating architectural visualization. The widely used technologies seem to be capable of producing optimal result, but adequate know-how, experience and experimentation are also required.

Because render engines are evolving continuously, it is important to follow their progress. While Scanline Rendering is widely considered inferior to the others, it shows clear improvements and it is fast. Mental Ray is widely used but complicated with own materials and a difficult GI setup. It could yield good results, but the trade offs have to be calculated. V-Ray is capable of using standard lights and materials accurately but it is not a very fast engine, and this has to be taken into account with bigger projects.

Architectural visualization has to be approached depending on the requirements of the project, the time given for completion and the developer's level of knowledge of said rendering technologies.

According to the case study and experiments discussed in this study, the most sensible approach for architectural visualization is to use at least two render engines when creating a final render of a scene. Since results can very greatly this approach seems to be a necessity rather than a choice. It must be noted that this necessity cannot be satisfied easily using todays CG softwares. Material, light and on occasion camera incompatibility hinders these efforts.

Scanline Rendering and V-Ray can easily share materials, lights and cameras, while Mental-Ray struggles with materials defeating the point of comparative renders. Comparing the three, Scanline rendering proved to be the fastest, followed by V-Ray. Mental-Ray failed on every level, but it is important to note that Mental-Ray used with its own materials was not part of this study and thus far fetched conclusions of its effectiveness in such a work scenario cannot be drawn.

As a best practice this study suggest using Scanline Rendering and the V-Ray render engine as the most time effective and quality aware solution.

References

For references articles, web sources and books were used.

1 Glenn S. Physics (light). [online]. Light (physics). [online]. Chicago, Illinois, U.S; Encyclopedia Britannica, Inc. January 2014
URL: http://global.britannica.com/EBchecked/topic/340440/light/258429/Photons. Accessed 27 May 2014.

2 Cornell University. History of the Cornell box. [online]. New York, New York, U.S; February 1998
URL: http://www.graphics.cornell.edu/online/box/history.html. Accessed 27 May 2014.

3 Falk DS, Brill DR, Stork DG. Seeing the light: Optics in nature. New York, New York, U.S; John Wiley & Sons, Inc, 1986. (C)

4 Newman EM. Fiftieth Anniversary of First Digital Image Marked. [online]. Gaithersburg, MD, US; National Institute of Standards and Technology, May 2007
URL: http://www.nist.gov/public_affairs/releases/image_052407.cfm. Accessed 27 May 2014.

5 Experiments in Art and Technology. Northwest Chapters. [online]. Seattle, WA, US; University of Washington, September 2002
URL: http://courses.washington.edu/eatreun/html/history/h_nw.html. Accessed 27 May 2014.

6 Geometric Design and Computation. History. [online]. Salt Lake City, UT, US; University of Utah, November 2000
URL: http://www.cs.utah.edu/gdc/history/. Accessed 27 May 2014.

7 Sean P. Pixar founder's Utah-made 'Hand' added to National Film Registry. [online]. Salt Lake City, UT, US; The Salt Lake Tribune, December 2011
URL: http://www.sltrib.com/sltrib/mobile/53193670-90/film-catmull-computer-animation.html.csp. Accessed 27 May 2014.

8 Clay E. Corner Stone (1990) 3D Studio promotional video. [video online]. April 2010,
URL: https://www.youtube.com/watch?v=pidl3rzdhUw. Accessed 27 May 2014.

9 Autodesk, Inc. History of Autodesk 3ds Max. [online]. San Rafael, California, U.S;
URL: http://area.autodesk.com/maxturns20/history. Accessed 27 May 2014.

10 Grafton J. Geometric theory. [online]. Manchester, UK; December 2011
URL: http://jessgrafton.wordpress.com/3d/geometric-theory/. Accessed 27 May 2014.

11 Franklin B. The Works of Benjamin Franklin, Vol. II Letters and Misc. Writings 1735-1753. New York, New York, U.S; G.P. Putnam's Sons, 1904.

12 Autodesk, Inc. Autodesk 3D Studio Max 2014 Documentation: Standard lights. [online]. San Rafael, California, U.S; March 2013

URL: http://docs.autodesk.com/3DSMAX/16/ENU/3ds-Max-Help/. Accessed 27 May 2014.

13 Autodesk, Inc. Autodesk 3D Studio Max 2014 Documentation: Photometric lights. [online]. San Rafael, California, U.S; March 2013
URL: http://docs.autodesk.com/3DSMAX/16/ENU/3ds-Max-Help/. Accessed 27 May 2014.

14 Autodesk, Inc. Autodesk 3D Studio Max 2014 Documentation: Guidelines for lighting. [online]. San Rafael, California, U.S; March 2013
URL: http://docs.autodesk.com/3DSMAX/16/ENU/3ds-Max-Help/. Accessed 27 May 2014.

15 Autodesk, Inc. Autodesk 3D Studio Max 2014 Documentation: Mental ray indirect illumination rollout (for lights). [online]. San Rafael, California, U.S; March 2013
URL: http://docs.autodesk.com/3DSMAX/16/ENU/3ds-Max-Help/. Accessed 27 May 2014.

16 Autodesk, Inc. Autodesk 3D Studio Max 2014 Documentation: Lights. [online]. San Rafael, California, U.S; March 2013
URL: http://docs.autodesk.com/3DSMAX/16/ENU/3ds-Max-Help/. Accessed 27 May 2014.

17 Mertzel E. Common problems. [online]. Pasadena, California, US; February 2011
URL: http://www.lookinglass.us/vray4rhinoWiki/tiki-download_file.php?fileId=71. Accessed 27 May 2014.

18 Solid Angle S. L. Ai Photometric Light. [online]. Madrid, Spain, 2014
URL: https://support.solidangle.com/display/AFMUG/Ai+Photometric+Light. Accessed 27 May 2014.

19 Horol D. HDRI environment in Maya. [video online]. May 2013,
URL: https://www.youtube.com/watch?v=RH7hZwTXYnQ. Accessed 27 May 2014.

20 Autodesk, Inc. Autodesk 3D Studio Max 2014 Documentation: Skylights & environment lighting (IBL) rollout (mental ray renderer). [online]. San Rafael, California, U.S; March 2013
URL: http://docs.autodesk.com/3DSMAX/16/ENU/3ds-Max-Help/. Accessed 27 May 2014.

21 Autodesk, Inc. Autodesk 3D Studio Max 2014 Documentation: Lighting in 3ds Max. [online]. San Rafael, California, U.S; March 2013
URL: http://docs.autodesk.com/3DSMAX/16/ENU/3ds-Max-Help/. Accessed 27 May 2014.

22 Udacity, Inc. Ambient Lighting. [video online]. Mountain View, California, U.S; April 2013
URL: https://www.youtube.com/watch?v=cj0NUlzqziw. Accessed 27 May 2014.

23 Whitehurst A. Depth map based ambient occlusion lighting. [online]. London, U.K; 2004
URL: http://www.andrew-whitehurst.net/amb_occlude.html. Accessed 27 May 2014.

24 Render Plus Systems, Inc. Reflection. [online]. Denver, Colorado, U.S; 2014
URL: http://www.renderplus.com/wp2/wk/Reflection.htm. Accessed 27 May 2014.

25 Autodesk, Inc. Autodesk 3D Studio Max 2014 Documentation: Modelling global illumination with radiosity. [online]. San Rafael, California, U.S; March 2013
URL: http://docs.autodesk.com/3DSMAX/16/ENU/3ds-Max-Help-Archive/. Accessed 27 May 2014.

26 McGuire M. High Performance Graphics: Ambient occlusion volumes. [online] The Eurographics Association, Saarbrücken, Germany, June 2010
URL: https://research.nvidia.com/sites/default/files/publications/McGuire10AOV.pdf. Accessed 27 May 2014.

27 Autodesk, Inc. Autodesk 3D Studio Max 2014 Documentation: Photometric lights. [online]. San Rafael, California, U.S; March 2013
URL: http://docs.autodesk.com/3DSMAX/16/ENU/3ds-Max-Help/. Accessed 27 May 2014.

28 Autodesk, Inc. Autodesk 3D Studio Max 2014 Documentation: Final gathering (FG) rollout (mental ray renderer). [online]. San Rafael, California, U.S; March 2013
URL: http://docs.autodesk.com/3DSMAX/16/ENU/3ds-Max-Help/. Accessed 27 May 2014.

29 Autodesk, Inc. Autodesk Maya 2013 Documentation: Final gather and ambient occlusion. [online]. San Rafael, California, U.S; March 2013
URL: http://download.autodesk.com/global/docs/maya2013/en_us/. Accessed 27 May 2014.

30 AutoDesSys, Inc. When to use final gather. [online]. Columbus, Ohio, U.S; 2011
URL:
http://www.formz.com/manuals/renderzone/!SSL!/WebHelp/html/1_6_1_when_to_use_fg.html. Accessed 27 May 2014.

31 Chaos Group. Indirect illumination (GI). [online]. Chaos Software, Sofia, Bulgaria; 2014
URL: http://help.chaosgroup.com/vray/help/150SP1/render_params_gi.htm. Accessed 27 May 2014.

32 Chaos Group. GI methods. [online]. Chaos Software, Sofia, Bulgaria; 2014
URL: http://help.chaosgroup.com/vray/help/150SP1/gimethods.htm. Accessed 27 May 2014.

33 Chaos Group. Caustics. [online]. Chaos Software, Sofia, Bulgaria; 2014
URL: http://help.chaosgroup.com/vray/help/150SP1/render_params_caustics.htm. Accessed 27 May 2014.

34 Csikos A. Nokia BH900 – bluetooth headset. [online]. RealtimeCGI, Budapest, Hungary; February 2008
URL: http://www.realtimecgi.com/?p=2706. Accessed 27 May 2014.

35 Autodesk, Inc. Autodesk 3D Studio Max 2014 Documentation: Shading Type. [online]. San Rafael, California, U.S; March 2013
URL: http://docs.autodesk.com/3DSMAX/16/ENU/3ds-Max-Help/. Accessed 27 May 2014.

36 Seymour M. Founder series: industry legend Jim Blinn. [online]. Los Angeles, California, U.S; FXGuide.com, July 2012
URL: http://www.fxguide.com/featured/founders-series-industry-legend-jim-blinn/. Accessed 27 May 2014.

Renders of Case Studies

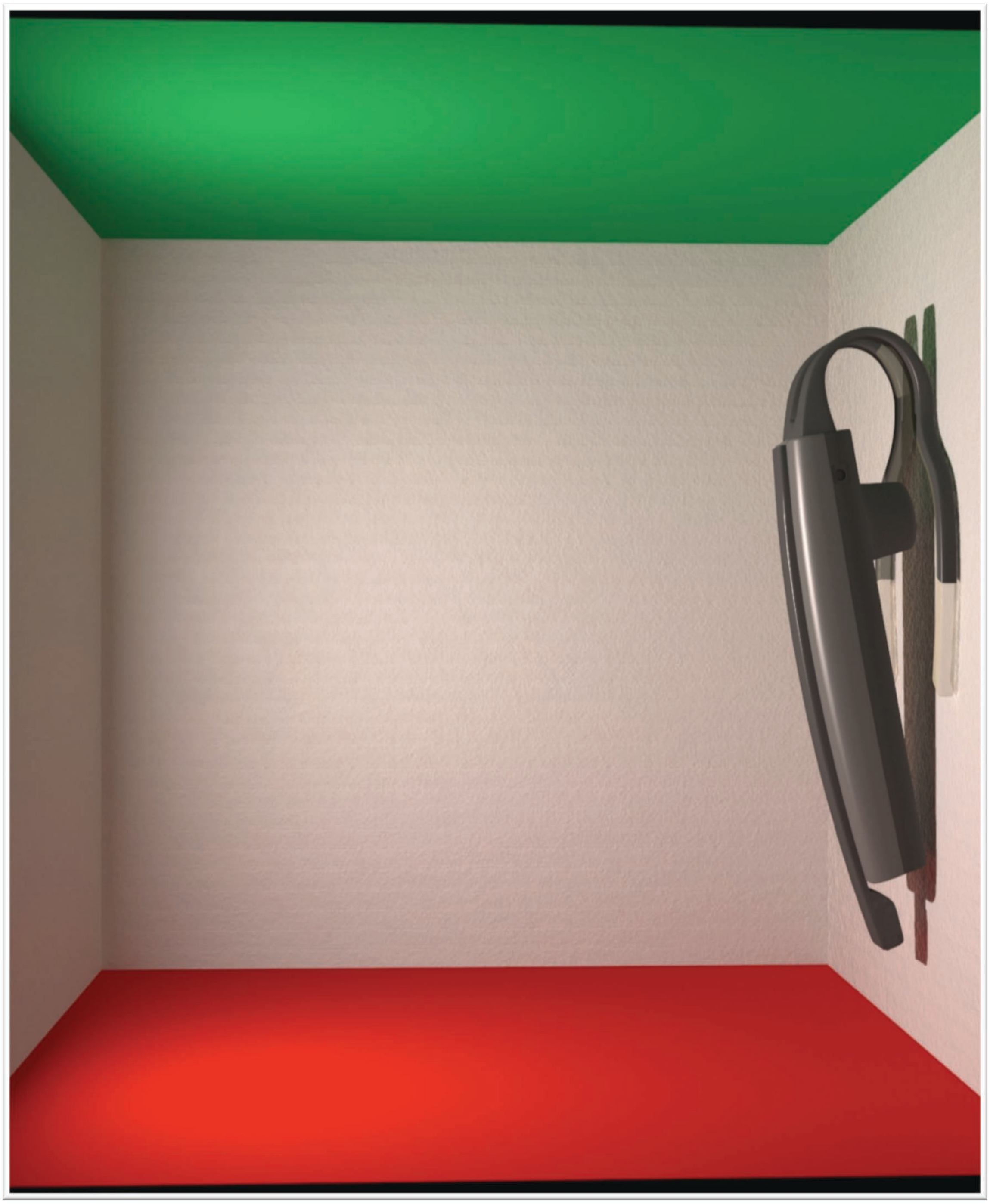


Figure 42. Case Study Render using Scanline Rendering.

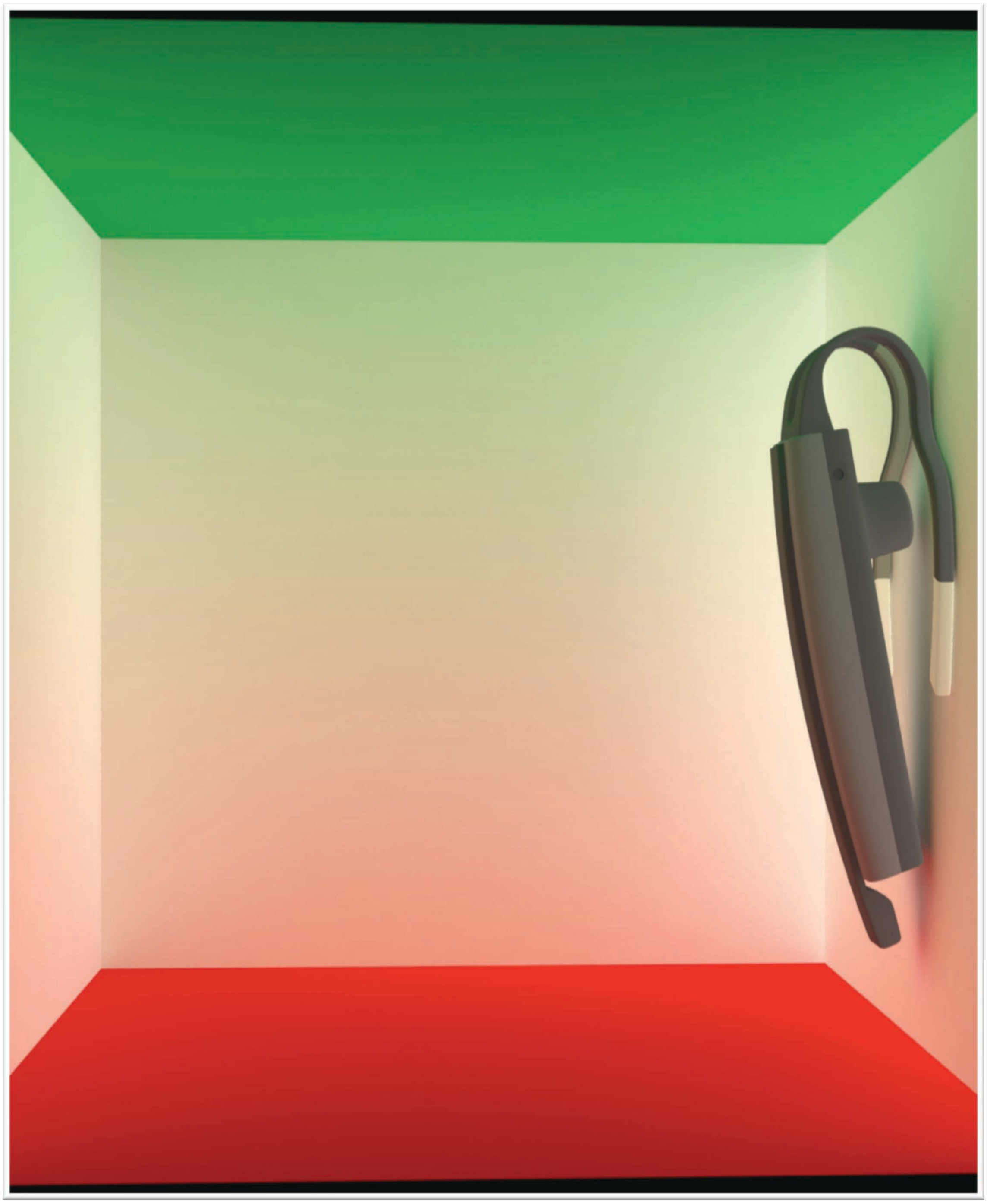


Figure 43. Case Study Render using Mental Ray.

Figure 44. Case Study Render using V-Ray.