FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Perception of Light in Virtual Reality

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Visual Computing

eingereicht von

Laura R. Luidolt, BSc
Matrikelnummer 01427250

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Associate Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer
Mitwirkung: Univ.Ass. Dipl.-Ing. Katharina Krösl, BSc

Wien, 2. Februar 2020

Laura R. Luidolt　　Michael Wimmer

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Perception of Light in Virtual Reality

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Visual Computing

by

Laura R. Luidolt, BSc
Registration Number 01427250

to the Faculty of Informatics

at the TU Wien

Advisor: Associate Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer
Assistance: Univ.Ass. Dipl.-Ing. Katharina Krösl, BSc

Vienna, 2nd February, 2020

Laura R. Luidolt   Michael Wimmer

Erklärung zur Verfassung der Arbeit

Laura R. Luidolt, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 2. Februar 2020

Laura R. Luidolt

Acknowledgements

First of all, I would like to thank Prof. Michael Wimmer for his guidance during this work.

Furthermore, I want to thank Katharina Krösl for letting me take part in her project on simulating vision impairments and her valuable input while proofreading this thesis.

I would also like to thank Prof. Michael Pircher from the Medical University of Vienna for answering my questions about the scattering of light in the human eye.

Moreover, a big thanks to all my user study participants, for giving me helpful feedback on my application.

This work was enabled by the Competence Centre VRVis. VRVis is funded by BMVIT, BMDW, Styria, SFG and Vienna Business Agency in the scope of COMET - Competence Centers for Excellent Technologies (854174) which is managed by FFG.

The hardware used in this project was funded by the TU Wien Informatics *Förderungsstipendium.*

My personal thanks goes out to my mother, for her never ending support, during my studies and especially my whole life. Also, thank you, for constantly sending me pictures of our cat.

Last but not least, I want to thank my boyfriend, Lukas, for preserving my last three molecules of serotonin by giving me food and attention.

Kurzfassung

Die Lichtwahrnehmung und der Lichteinfall in das menschliche Auge sind grundlegend anders in Szenarien der wirklichen Welt als in Virtual reality (VR) Simulationen. Gewöhnliche Displays, wie sie beispielsweise in VR Headsets verwendet werden, können den vollen Umfang der menschlichen Helligkeitswahrnehmung nicht abbilden. Deswegen müssen Licht-Phänomene, wie die zeitliche Augenadaption, blendendes Licht, vermindertes Sehvermögen und skotopisches Sehen, simuliert werden, um realistische Bilder zu erzeugen. Obwohl eine Simulation basierend auf physikalischen Grundlagen, die wahrgenommene Echtheit von VR Applikationen wesentlich erhöhen könnte, wurde dieses Thema noch nicht vollständig wissenschaftlich erforscht.

Wir stellen einen Post-Processing Arbeitsablauf vor, für VR und augmented reality (AR), mithilfe von Eye Tracking, welcher auf medizinischen Untersuchungen von gesunden, menschlichen Augen basiert und außerdem Echtzeit Laufzeiten erzielen kann, um Lichteffekte so real wie möglich zu simulieren. Wir stellen neue und schnellere Algorithmen vor, um die Simulation von blendenden Lichtern und vermindertem Sehvermögen im Dunkeln realistischer wirken zu lassen. Wir adaptieren die Intensität des in das Auge einfallenden Lichts abhängig von der Blickrichtung des Nutzers. Zusätzlich simulieren wir die Anpassung des Auges an verschiedene Lichtverhältnisse, sowie die Veränderung der Farbwahrnehmung im Dunkeln.

Wir haben die erste Versuchsstudie abgehalten, welche eine Alltagsszene mit niedrigen Lichtverhältnissen direkt mit einer entsprechenden VR Simulation vergleicht. Viele Teilnehmer erwähnten, dass die Simulation der meisten Effekte ähnlich, beziehungsweise gleich, wie ihre eigene Wahrnehmung waren. Es hat sich jedoch herausgestellt, dass weitere Arbeit an dem Erscheinungsbild des blendenden Lichts sowie dessen Bewegungen notwendig sind. Wir schließen, dass unsere Arbeit eine Grundlage für weitere Forschung bezüglich der Simulation und individuellen Anpassung an Lichteffekte, gelegt hat.

Abstract

The perception of light and light incidence in the human eye is substantially different in real-world scenarios and virtual reality (VR) simulations. Standard low dynamic range displays, as used in common VR headsets, are not able to replicate the same light intensities we see in reality. Therefore, light phenomenons, such as temporal eye adaptation, perceptual glare, visual acuity reduction and scotopic color vision need to be simulated to generate realistic images. Even though, a physically based simulation of these effects could increase the perceived reality of VR applications, this topic has not been thoroughly researched yet.

We propose a post-processing workflow for VR and augmented reality (AR), using eye tracking, that is based on medical studies of the healthy human eye and is able to run in real time, to simulate light effects as close to reality as possible. We improve an existing temporal eye adaptation algorithm to be view-dependent. We adapt a medically based glare simulation to run in VR and AR. Additionally, we add eye tracking to adjust the glare intensity according to the viewing direction and the glare appearance depending on the user's pupil size. We propose a new function fit for the reduction of visual acuity in VR head mounted displays. Finally, we include scotopic color vision for more realistic rendering of low-light scenes.

We conducted a primarily qualitative pilot study, comparing a real-world low-light scene to our VR simulation through individual, perceptual evaluation. Most participants mentioned, that the simulation of temporal eye adaptation, visual acuity reduction and scotopic color vision was similar or the same as their own perception in the real world. However, further work is necessary to improve the appearance and movement of our proposed glare kernel. We conclude, that our work has laid a ground base for further research regarding the simulation and individual adaptation to the perception of light in VR.

Contents

"Reality is merely an illusion, albeit a very persistent one."

– *Albert Einstein*

CHAPTER 1

Introduction

1.1 Motivation and Problem Statement

With the increasing numbers of virtual reality (VR) and augmented reality (AR) applications and simulations, many new research topics and challenges arise. Continually improving VR hardware allows for techniques that would not have been possible just a few years ago. A big focus of VR is to give the user an immersive experience, and while various factors can have an impact on that (such as the ease of use, cybersickness, or interactivity) [MH17], realistic graphics play an essential role. A significant aspect that has to be considered when trying to achieve realism is light. The human vision is dependent on light and the lighting of a scene in order for us to be able to understand it correctly. An appropriate simulation of perceived light in VR can be used in many different areas, such as gaming, applications focused on perception, or for visual effects. Furthermore, real-world applications, such as driving simulators, the analysis of eye diseases or architectural planning tools, can be enhanced to achieve more accurate simulations with correct light perception and glare effects.

Most previous lighting and visual-perception algorithms [KMS05, YBMS05] were developed (and are still mainly used) for 2D displays. The light calculations themselves are done in 3D space, but the output media is still merely two dimensional. Displaying an image on a 2D screen does not give the impression of actually being in the scene – an image only gives a small insight into a much larger environment, like a window to the world. Using VR, these restrictions do not apply: a user can freely experience the whole virtual scene the way they want to. VR leads to a more immersive and realistic simulation than when looking at a standard display. To be as immersive as possible, very detailed algorithms are necessary to avoid the break of immersion through rendering artifacts. Viewing a scene in VR is perceptually different from normal monitors, due to the complete human field of view (FOV) being covered by the head mounted display (HMD), blocking all other light influences on the eyes, and the shorter distance between the eyes and the display. An HMD should reproduce what actual human eyes perceive and not simulate a camera's vision. However, the differences in visual perception of common displays and VR are not a key focus in current VR rendering – the same algorithms as for 2D screens are employed. The perceptual differences of standard displays and HMDs are neglected in today's approaches. There is no real-time VR simulation tool, that is able to display the full range of the human visual system true to human perception.

Implementing medically based approaches regarding the perception of light in VR hasn't been explored at all. To the best of our knowledge, there are no large-scale analyses of individual user perception in VR and the differences to common displays. Furthermore, detailed medical algorithms require a lot of processing power for the exact calculation of all kinds of influences on perception. Especially in VR, the runtime is a critical issue due to the high pixel count of HMDs and a necessary minimum of 90 frames per second (FPS) to avoid visible stutters. Therefore, perceptually realistic algorithms in VR have high requirements and are not always fully able to run in real time on current hardware.

Furthermore, a user's light perception depends on many circumstances, for example, pupil size, visual acuity, or a person's age [GPB^{+}13, ZDRCLG19]. Nevertheless, these factors are not accounted for in common algorithms. General approaches for displaying light on 2D devices do not suffice for the three-dimensionality as provided by HMDs, because of inherently different perception in VR and higher demands regarding performance. We need more accurate light simulations in VR, to allow for more realism and enable a user to actually feel like they are in the scene. Multiple influences on the human eye in natural situations have to be included in order to be able to represent a scene as realistic as possible.

The human eye is designed to adapt to different light conditions of the real world. Even though we can only distinguish between about 100 levels of brightness, we are still able to see in very dark environments, as well as in really bright surroundings. The eye is continuously adapting to different lighting conditions: For vision in dark rooms our pupils dilate, and the rods in our eyes are activated. This is called scotopic vision. In bright sunlight, the iris constricts the size of our pupils, and we are capable of color vision, i.e., photopic vision, due to activated cones [KMS05]. This means that we adjust to various levels of light, depending on the brightness of the area that is observed. Therefore, the brightness perception of humans is gaze-dependent [Man17]. The current viewing direction influences the way the light enters the eye, especially in low-light scenes with single light sources.

Consequently, the simulated illumination of a scene needs to be adjusted accordingly for displaying it in an HMD. The calculated luminance values have to be mapped to the available range of brightness levels in a perceptually pleasing way, e.g., with tone mapping. Yet, the major perceptual difference in VR from common displays is not included in today's VR rendering. To achieve a proper mapping of all brightness levels, the viewing direction has to be known to simulate the eye adaptation interactively, depending on whether the user looks at bright or dark areas. Current approaches that rely on the viewing direction were developed for conventional 2D displays [MM13]. However, interactive adaptation is currently not commonly considered, because the detection of the viewing direction requires additional eye-tracking hardware. Moreover, 2D displays are not fully immersive; thus, there is no need for the scene to be entirely perceptually correct, because the perception is not close to real life regardless. Approaches designed for computer screens are not able to control the the complete human visual field as in VR and rendering algorithms might excessively compress the brightness values of the whole scene.

To achieve full immersion and realistic perception in VR, the gaze direction is needed to determine the exact incidence of light in the eye. In order to display light in a natural way, similar to real-world perception, efficient VR algorithms are required based on medical research, which take interactive factors, like pupil size and viewing direction, into account. This means additional hardware (specifically an eye tracker) and dedicated methods are necessary to simulate realistic perception of light.

1.2 Aim of the Work

In this project, it is our goal to develop an application that can simulate perceptually realistic scenes under any lighting condition in VR. We aim to simulate light in a natural way, as the human eye perceives it. Using an eye tracker, we are able to measure the "state" of the eye, e.g., size of the pupil or viewing direction, and in response, adapt the displayed image. With VR, we are entirely in control of what the user sees – there is no other influence of the real world possible as there would be with standard displays. Therefore, we can measure and control the incidence of light into the eye of a user precisely. Other perceptually influenced algorithms [JS17, KMS05] are not capable of including or adapting to the state of the eye or to control ambient light of the room other than the monitor itself. Therefore, they make generalizations that possibly lead to less perceptually accurate results. By combining VR and eye tracking with medical research of the human eye, we aim to increase realism and create more immersive and authentic applications.

1.3 Contribution

We have developed a method that can build upon any traditional light calculation of a scene and adapt the displayed image locally and depending on the view direction. This simulation includes not only the adaptation to light or dark areas but the impact of light on the whole image, in particular glare effects and star-shaped artifacts perceived when looking into bright light.

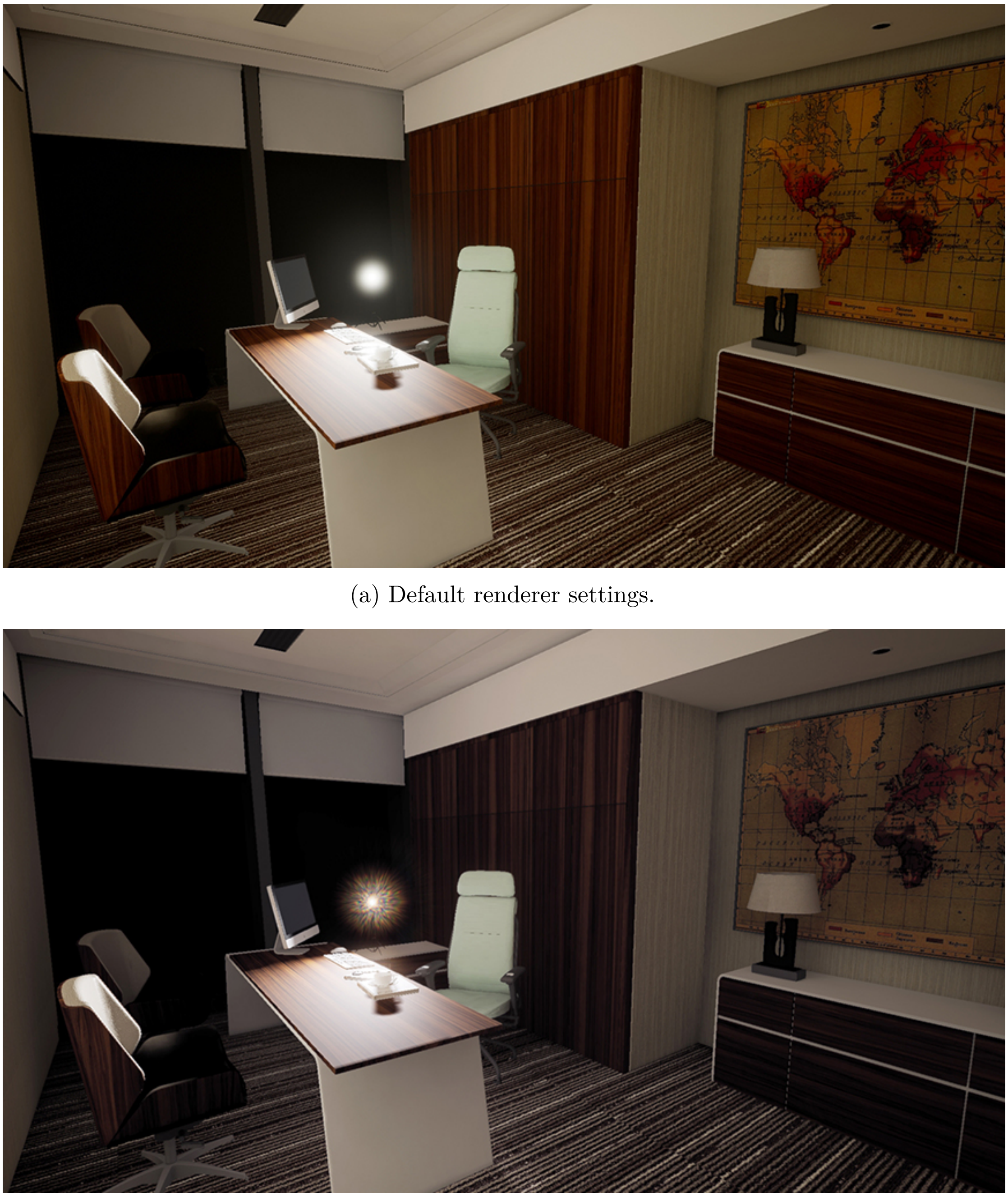
As rendering engine we use Unreal Engine (UE), where we were able to add our adaptations as post-processing stages. This thesis builds upon algorithms intended for 2D displays and adapts them for VR rendering. We rely on the perceptual tonemapper proposed by Krawczyk et al. [KMS05] and update their temporal eye adaptation, visual acuity reduction and scotopic color vision for VR, to run in real time and achieve perceptually realistic results. Furthermore, we have modified the perceptual glare by Ritschel et al. [RIF+09] to run in real time with an eye tracker.

We make the following contributions:

- We developed a new post-processing workflow for more accurate light simulation in VR and AR, which can be used for vision impairment studies, evaluating lighting conditions for architectural room planning, as well as more realistic computer games.
- We adjusted the simulated image to reproduce temporal eye adaptation, as well as visual acuity reduction and scotopic color vision in dark areas.
- We modified the "Temporal Glare" proposed by Ritschel et al. [RIF+09] to run in real-time VR/AR following an optometrist's advice, resulting in our perceptual glare.
- We used an eye tracker to adjust our simulation to light characteristics in real time. We adapt the scene brightness depending on the viewing direction or the glare effect and the pupil size.
- We conducted the first user study that compares a real-world low-light situation to a VR simulation using a perceptual evaluation. Our results show that visual perception is different for each participant, yet overall our simulation was able to give the impression of a realistic scene.

This approach should lead to a more natural impression in VR and will allow us to conduct user studies on perception and human vision more efficiently and realistically than before. In future work, our methodology can be used, in combination with vision-impairment simulations [KEW+19], to allow for more realistic applications when simulating eye diseases. Additionally, with glare effects based on real user perception, it will be possible to improve the simulation of light during the planning period of buildings for architects and lighting designers. Problematic areas regarding the placement of luminaries could be identified more easily, and handled during the early planning stages, to avoid creating uncomfortable glare effects.

An example comparison of the default settings of a common renderer to our contribution can be seen in Figure 1.1.



(a) Default renderer settings.

(b) Our approach.

Figure 1.1: Comparison of a standard rendering method to our perceptually adapted method of a low-light scene.

1.4 Structure of the Work

In Chapter 2, we explain the necessary background knowledge of this thesis. This includes medical details on the human eye and how we perceive light and color. Furthermore, technical details on brightness compression in computer graphics are explained, as well as the Fast Fourier transformation, which we use in our approach. In Chapter 3, we explain the current state of the art in related work. Approaches similar to our method, which aim to simulate light realistically, and algorithms that approximate natural effects in standard vision are presented.

In Chapter 4, we give an overview of our proposed methodology. An introduction to all the topics we deal with in this thesis and their dependencies are made. In Chapter 5, we lay out the methodology of this thesis in detail. The four main effects of human light perception, i.e., temporal eye adaptation, perceptual glare, visual acuity reduction, and scotopic color vision, and their application to a rendered image are explained. In Chapter 6, we give all details regarding the implementation of this work. We describe the software we used and specific rendering details for our methodology.

In Chapter 7, the results of this thesis and its evaluation are presented. We give insight into the conducted user study, with the necessary hardware. Furthermore, we discuss the results at hand and compare their benefits as well as shortcomings in comparison to similar works. The runtimes of the application at hand are also presented. In Chapter 8, we conclude our thesis and present some possibilities for future work.

CHAPTER 2

Background

In this chapter, all the background knowledge applied in this thesis is described. First, we discuss the anatomical medical details of the human eye, where the relevance of each part to this work is analyzed. Secondly, the human perception of light is explained, including the physical properties leading to human light perception, as well as the influence of anatomical characteristics on different visual effects. Next, the technical specifics of fitting the full range of human light vision onto a standard display are illustrated. Finally, we explain the mathematical details of the Fast Fourier transformation (FFT) used in this work.

2.1 Anatomy of the Human Eye

In this section, we will describe the anatomical basis of the human eye. The parts of the eye that influence our perception regarding light incidence and processing are explained and how they might differ between humans or change with age.

The eye is an ellipsoid with an average diameter of 24 mm along its longest axis. A sketch that indicates all parts of the human eye, that are significant for our work can be seen in Figure 2.1. We describe these parts in the following sections.

2.1.1 Cornea

Light entering the eye first hits the cornea, a transparent protection layer. It accounts for approximately 70% of total light refraction in the eye [Sri18], due to the vast difference in refractive index between air and the cornea's tissue. Irregularities in the surface structure can cause optical aberrations and scattering errors [SMV$^+$16]. When light rays deviate from their original path (but less than 90°), it is called *forward light scatter* or *straylight* and can cause changes in luminance or lead to glare effects. Backward scattering (more than 90° deviation) reflects rays away from the eye. Hence, it leads to a loss of these light rays for the human eye.

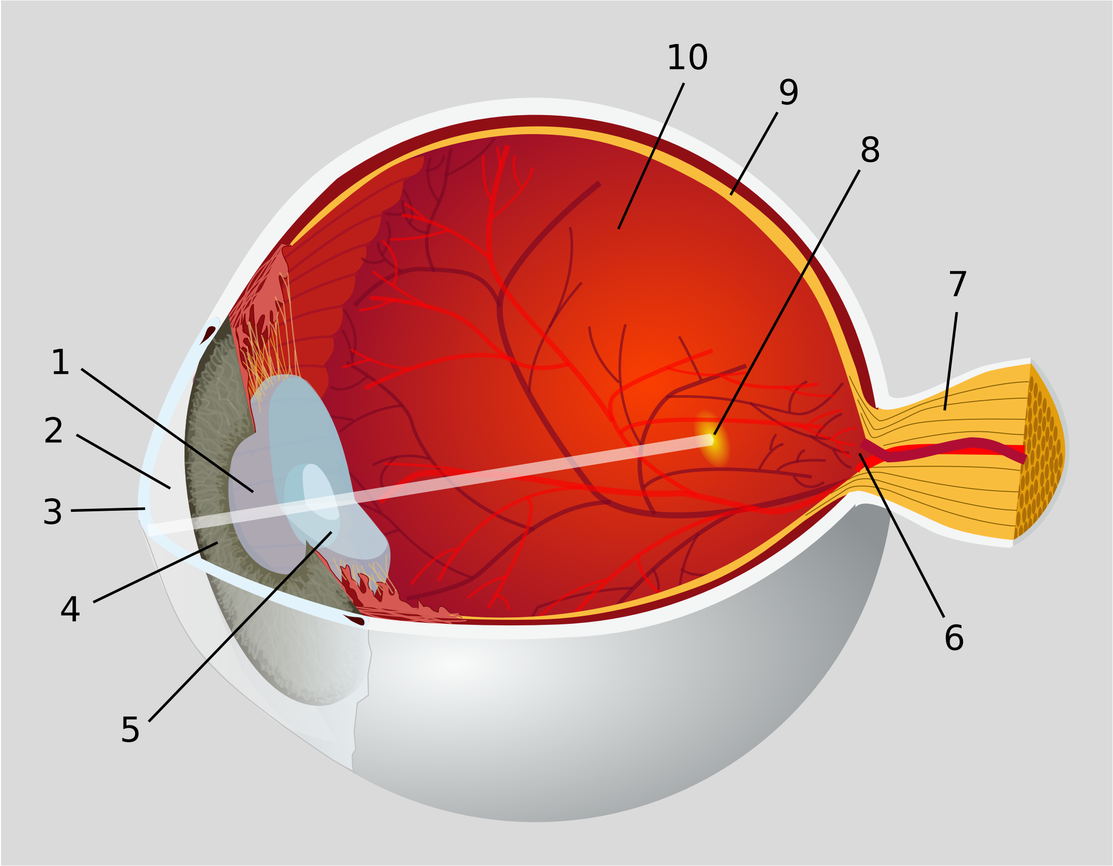


Figure 2.1: Anatomy of the right human eye. (1) Pupil; (2) Anterior chamber; (3) Cornea; (4) Iris; (5) Lens; (6) Optic disc; (7) Optic nerve; (8) Fovea; (9) Retina; (10) Vitreous body. Adapted from [Cha07].

2.1.2 Anterior Chamber

The anterior chamber is located between the cornea and the iris and is filled by the aqueous humor [IG09]. Light scattering or glare in this area only appears in infected or inflamed eyes [Tho19]. Since we base our approach on healthy and normal eyes, we do not take this part into further consideration.

2.1.3 Iris and Pupil

The (usually) colorful iris surrounds the pupil of the eye. Muscles connected to the iris allow it to contract and dilate like a diaphragm, as shown in Figure 2.2. The pupil acts as aperture that lets light into the eye and adjusts to the amount of light entering. The pupil size ranges from approximately 2 to 8 mm, and responds to luminance changes of the viewed scene, which is called the *pupillary light reflex* [PAF$^+$08].

According to a study by Rozema et al. [RdBT10], the color of the iris only has a minor

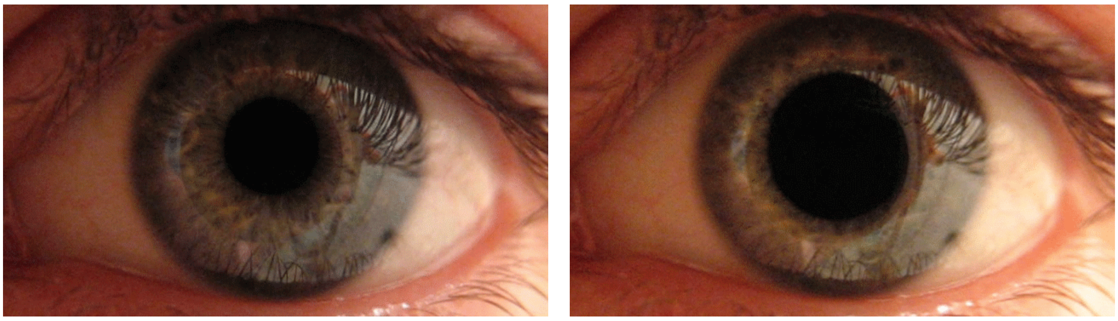


Figure 2.2: Pupil constriction and dilation. Reprinted from [Gre06].

influence on the scattering of light and is neglectable for a model of straylight. However, it has been shown that retinal straylight is dependent on the pupil diameter under all lighting conditions [FTCvdB07]. In detail, more light can enter the eye and scatter with wide pupils than small pupils. Therefore, we include the pupil size as a parameter for our glare model.

2.1.4 Lens

The lens is a biconvex structure of transparent tissue with a diameter of approximately 10 mm, which consists of protein structures and is mainly responsible for generating an image on the retina [IG09]. It is located behind the iris and pupil and connects them to the vitreous body. It is held in place by suspensory ligaments and ciliary muscles, which are also used for deforming, i.e., focusing, the lens.

The lens consists of the lens nucleus and the lens cortex. The nucleus contains long, thin cells, which are circularly ordered around the cortex. However, the nucleus does not contribute to light scattering [vdBS99].

The protein structures in the cortex are spherical particles in a highly ordered arrangement. Irregularities in this ordering or particles of increased size lead to interferences, which results in unevenly scattered light and, therefore, in retinal straylight [Tuc97]. In general, large particles dominate the forward scattering [vdBS99], with a median radius of 724 nm and taking up a fraction of 6×10^{-6} of the lens volume. Therefore, many thousand particles of increased size are present in healthy eyes [vdBHC05].

2.1.5 Vitreous Body

The vitreous body is a transparent, gelatinous substance that takes up approximately 80% of the eyeball's volume [YS20]. It holds the lens and retina in place and maintains eye pressure to keep the eyeball's near-spherical shape. The vitreous body transmits light with minimal scattering due to a low concentration of particles in healthy eyes [Rem12] and is therefore not taken into account for our simulations.

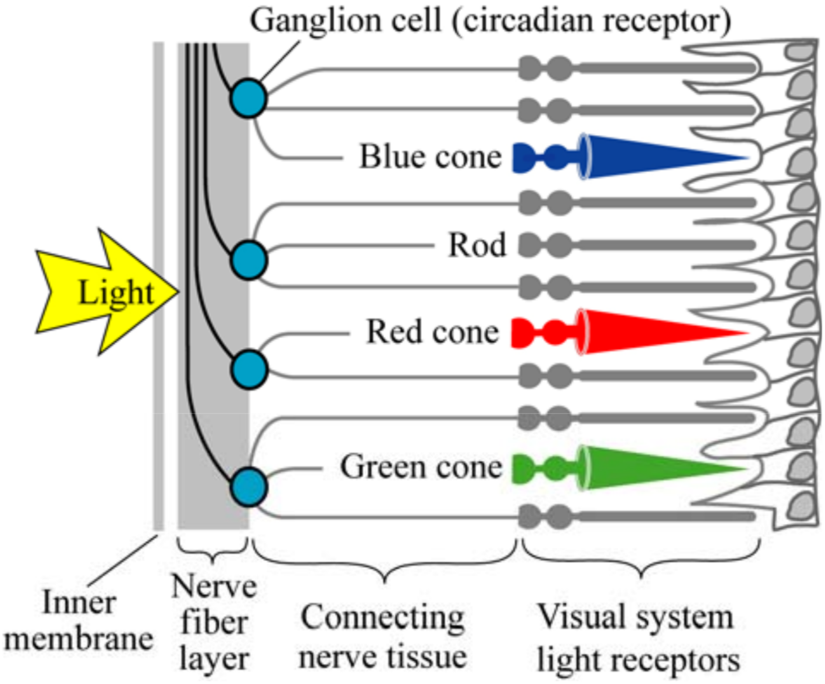


Figure 2.3: Structure of the retina's layers and cells. Reprinted from [Sch06].

2.1.6 Retina and Fovea

The retina covers most of the inner eyeball's backside and is responsible for capturing the image that is projected onto it and the first processing of the image [IG09]. It is a light-sensitive tissue with multiple layers, e.g., the ganglion cell layer, the receptor layer, and other connecting layers (see Figure 2.3). The fovea is a small area on the retina along the central axis of light incidence and has the highest concentration of cells and is, therefore, the point of sharpest vision.

There are two types of photoreceptors, rods and cones, with a significantly higher amount of rods (approximately 110–125 million), as compared to cones (6.3–6.8 million) [IG09]. Photoreceptors are specialized nerve cells with light-sensitive tips that absorb the incoming light and transform it into electrical impulses, which is then sent to the brain [Pal99].

The rods are responsible for light and dark vision and are distributed over the whole retina except the fovea (see Figure 2.4).

The cones are used for color vision and are mainly located in the fovea, and only very sparsely present in outer regions. They can be divided into three different types, L-cones, M-cones, and S-cones, which process long ($\sim$ 564 nm), medium ($\sim$ 534 nm) and short ($\sim$ 420 nm) wavelengths respectively, for red, green and blue colors [Pur19]. The distribution of cones in the fovea can vary between different people, but on average, have a relation of $\mathrm{L} : \mathrm{M} : \mathrm{S} \approx 10 : 5 : 1$. Outside the fovea, most of the cones are blue-sensitive,

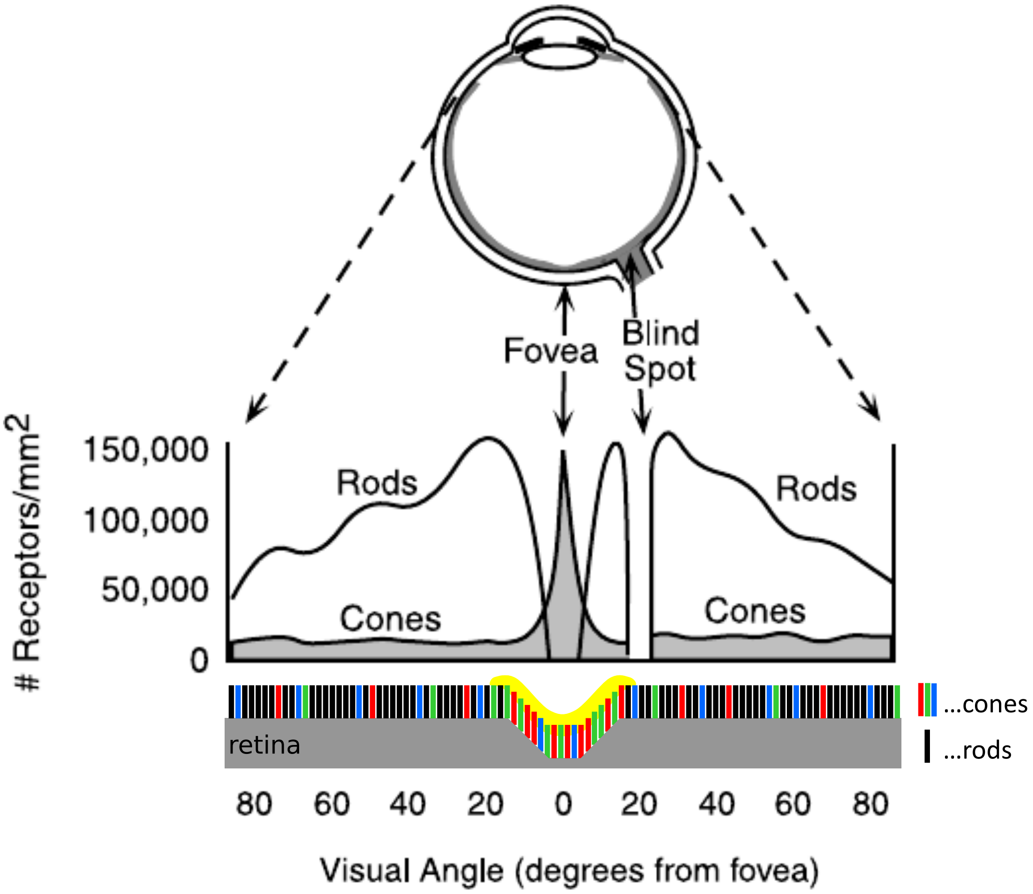


Figure 2.4: Amount and distribution of rods and cones in the eye. Adapted from [Pur19] and [Pal99].

and only a small amount of red- and green-sensitive cones are present, as shown in Figure 2.4. In theory, this means that we are only able to detect color clearly in the current point of focus. However, the human brain can inter- and extrapolate the colors. Therefore, we also perceive the side regions as colorful.

The anatomy of retina and fovea, as well as the distribution of photoreceptors significantly influences our perception. During standard lighting, we are able to see clear shapes and distinguish colors, while in dark scenarios our vision is less detailed, edges are blurred and colors are shifted (for details see Section 2.2.2).

2.1.7 Optic Disc and Nerve

All nerve fibers of the eye that transport information to the brain converge to the optic disc, located approximately 3 mm to the nasal side of the fovea [IG09]. The optic nerve is the connection from the eye to the brain and is also responsible for the blood supply of the eye. The disc does not contain any photoreceptors and, therefore, is also known as the blind spot. It appears at 15 degrees of visual angle on the temporal side of each eye. However, the blind spot only takes effect in monocular vision and our brain interpolates the missing data at this spot. Therefore, the optic disc and nerve do not influence our perception and are not included in our simulation.

2.2 Perception of Light

In this section, the basics of the average human perception of light are described and its different influences used in this thesis, i.e., ocular straylight resulting in perceptual glare, the reduction of visual acuity in the dark and scotopic color vision.

2.2.1 Light Incidence and Processing

When bundling light through a small hole, an inverted image can be seen on the other side, as shown in Figure 2.5. This effect is generated by a so-called *pinhole camera*. If the hole is too big, too many rays can enter, and the image appears blurry; if it is too

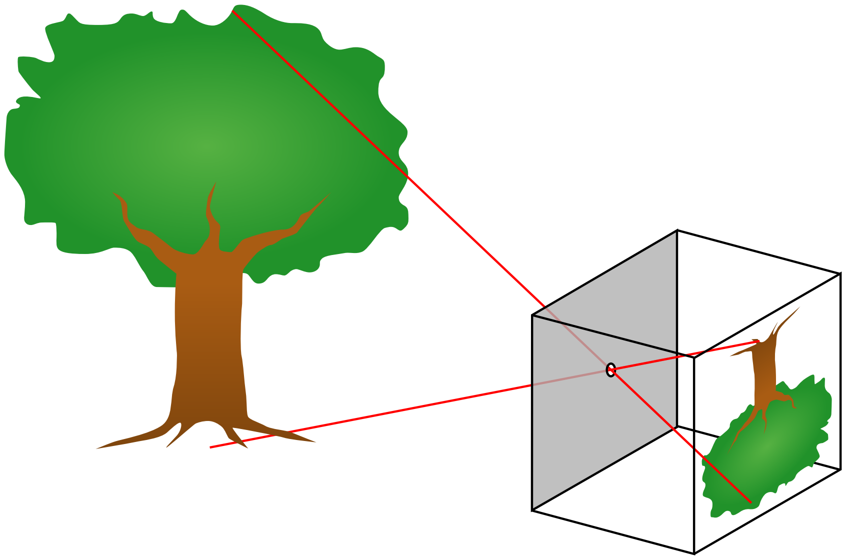


Figure 2.5: Example of a simple pinhole camera. An inverse image is visible on the backside of a dark box with a small hole. Reprinted from [MP08].

small, then the light waves break on the borders and diffract, which also leads to a blurry result.

In modern cameras a lens is used to bundle incoming light at a specific distance, to allow for a bigger hole. That hole is then called *aperture* and the resulting image benefits from more light entering and is brighter, while still maintaining a sharp result.

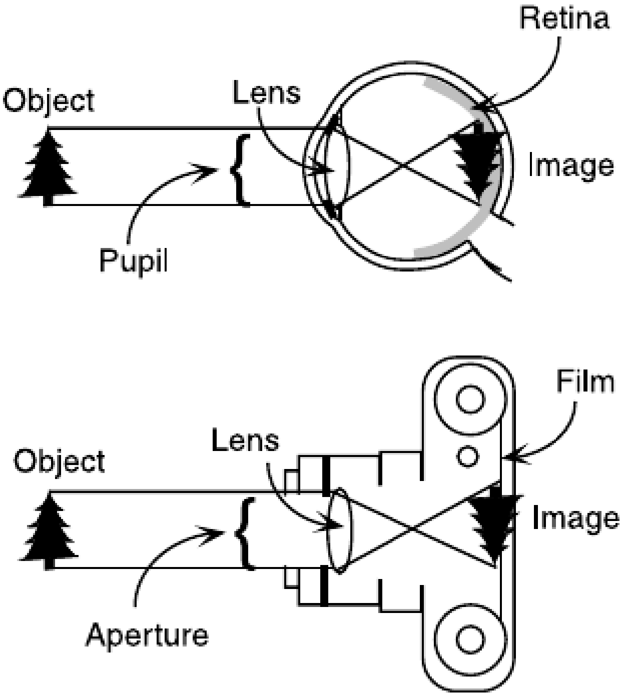


Figure 2.6: Comparison of a human eye and a camera. Reprinted from [Pal99].

Our eyes work similar to such a camera, except our lens focuses the light on the retina, which then processes it further (see Figure 2.6). The human lens is shaped precisely to focus light on the fovea in a way that most light rays hit this area of sharpest vision. Additionally, the pupil works as an aperture – it lets in more light rays in dark and less in bright environments.

As show in Figure 2.6, the human brain continually "receives" an inverted image on the retina, but is accustomed to this and allows us to see the right side still up/down.

Due to different wavelengths of light entering the eye, there are small differences in refraction in the lens, and not every "color" hits the retina precisely at the same point, as shown in Figure 2.7. This effect is especially prominent in the scattering of bright and intense light, which leads to the bright corona we perceive when looking at a bright light.

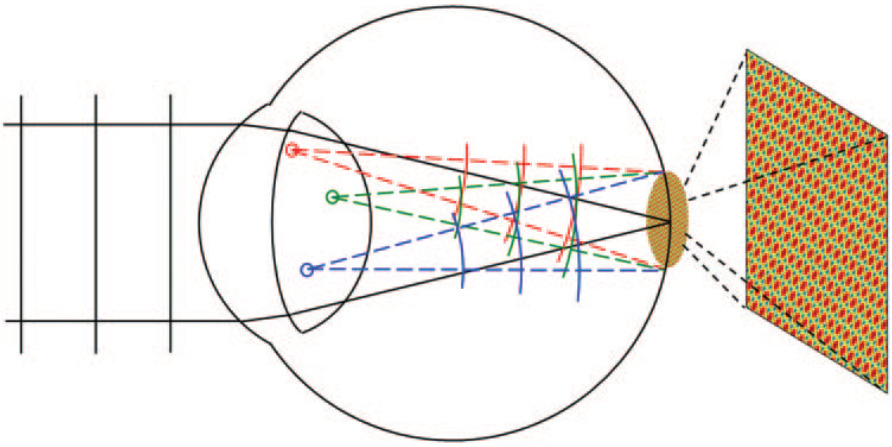


Figure 2.7: Side view of the eye with incoming light. Different wavelengths scatter at particles in different angles, therefore, creating interferences. Reprinted from [vdBHC05].

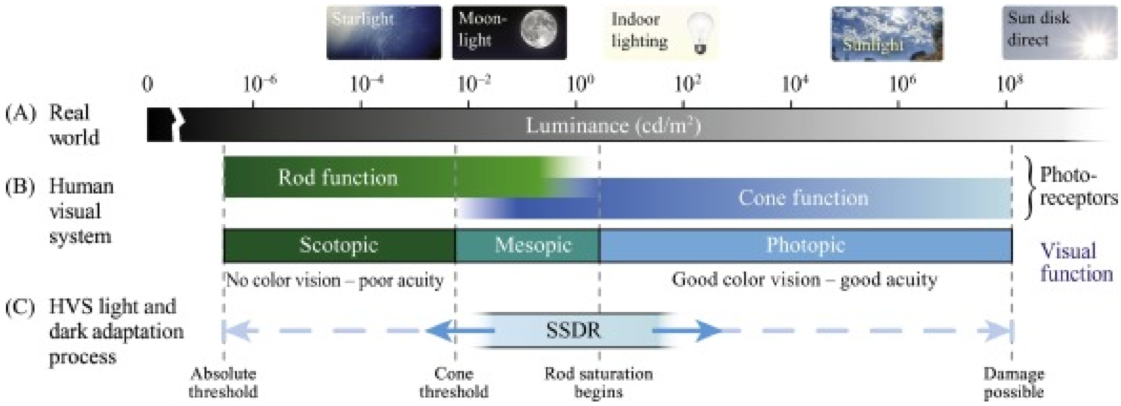


Figure 2.8: Luminance levels in the real world and the human visual system. Reprinted from [KDMF16].

2.2.2 Characteristics Regarding Eye Anatomy

We can explain many phenomena in our vision only via considering the characteristics in the anatomy of the eye. We aim to cover these in the following.

2.2.2 i. Photoreceptors Efficacy

Cones are able to process luminances from 10^{-6} to $10\frac{cd}{m^2}$, while rods are active from 0.03 to $10^8\frac{cd}{m^2}$ [KDMF16] (see Figure 2.8). Our perception changes depending on which cells are primarily active. We distinguish between photopic, mesopic, and scotopic vision. Photopic vision is our "everyday" vision. We are used to seeing colors and sharp outlines under different illuminants, i.e., simple indoor lighting or a bright and sunny day. Mesopic vision is usually active at twilight and dawn. This is the time when both our receptor

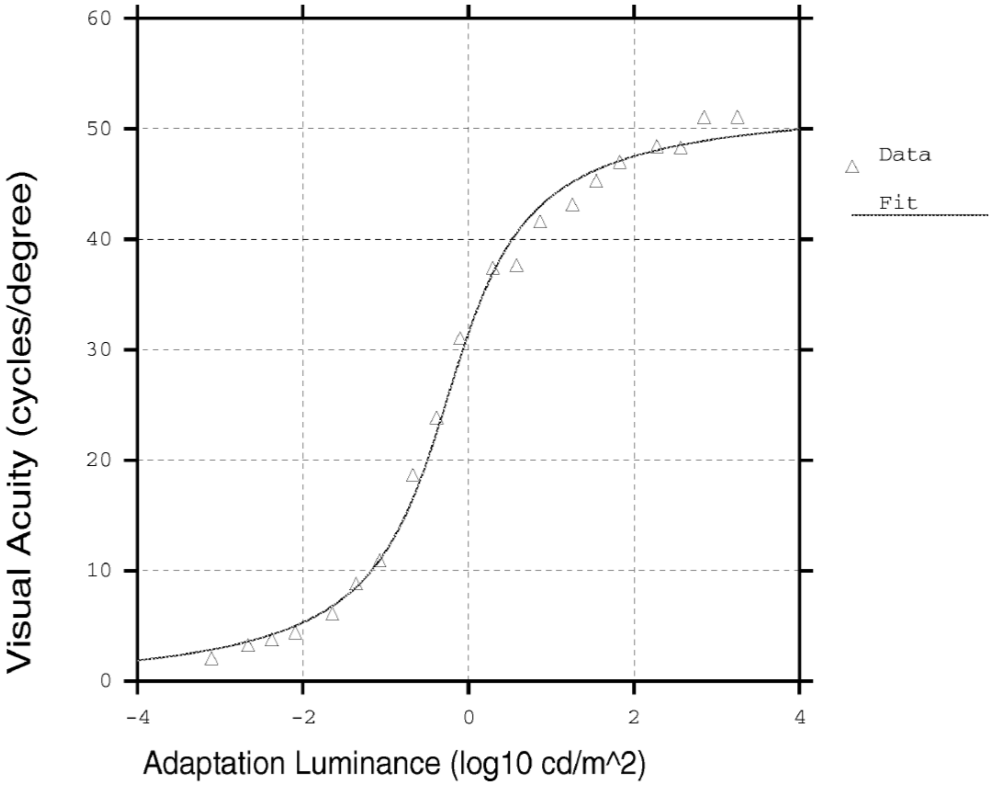


Figure 2.9: Function fit to human visual acuity. Reprinted from Larson et al. [LRP97].

cells are active, and we can still distinguish colors, but they appear shifted and washed out. Scotopic vision is our night vision, where only rods are active since the incoming light is too faint to be detected by the cones.

2.2.2 ii. Visual Acuity Reduction in Scotopic Vision

Since there are no rods located in the fovea (i.e., our point of sharpest vision, see Section 2.1.6), we lose visual acuity at luminances where no cones are active: at less than $0.03\frac{cd}{m^2}$ as shown in Figure 2.9. The rods are still able to discern shapes and especially movement in the dark, but we are not able to fully perceive sharp edges like during photopic vision.

2.2.2 iii. Colors in Scotopic Vision

During mesopic and scotopic vision, the rods in the eye become more prominent than the cones. Due to the increasing efficiency of rods, a perceptual color shift towards green and blue colors occurs. This impression is called the *Purkinje effect*, which states that the luminance sensitivity of photoreceptors in scotopic vision is the highest at approximately 507nm, while the photopic sensitivity peaks at 555nm [BDE$^+$09], as shown in Figure 2.10.

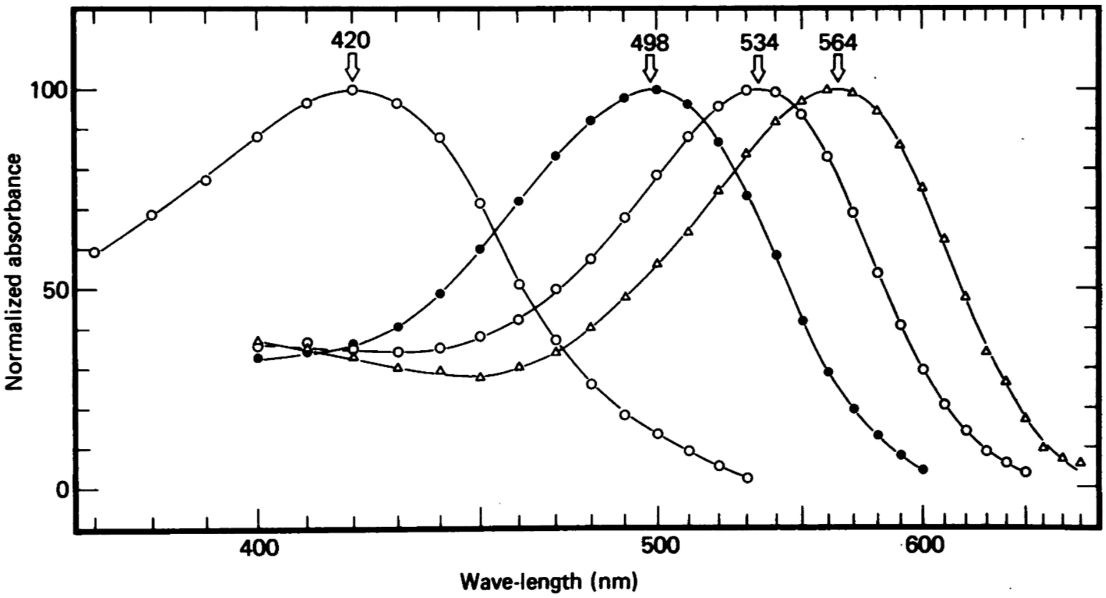


Figure 2.10: The normalized absorbance of blue cones (maximum at 420 nm), rods (498 nm), green cones (534 nm) and red cones (564 nm) at different wavelengths of incident light. Reprinted from [BD80].

This effect leads to a perceptual change of color vision. In detail, red colors become darker since no cells can "capture" their wavelengths. Blue-green colors become more prominent than during photopic vision, because rods are still able to detect these wavelengths.

2.2.2 iv. Noise in Scotopic Vision

The light-sensitive molecules in the rods sometimes fire randomly [Pur19]. Due to the high amount of rods in the eye, this frequently happens during the scotopic vision, which we then perceive as random, small flashes. These errors lead to an overall perception of a "noisy" image.

2.2.2 v. Temporal Adaptation

Human perception is not only dependent on the current scene's luminance but also the time spent under specific lighting conditions. For example, being in a dark room and turning on a lamp leads to oversaturation of the dark-adapted photoreceptors, and everything is too bright for a few moments. After a few seconds, we adapt to the new illumination. It takes the human eye approximately 5 minutes to fully adapt in such a scenario [Bri87]. On the other hand, it can take 20 to 30 minutes to adapt from bright sunlight to darkness, due to slower reaction times of the rods (see Figure 2.11).

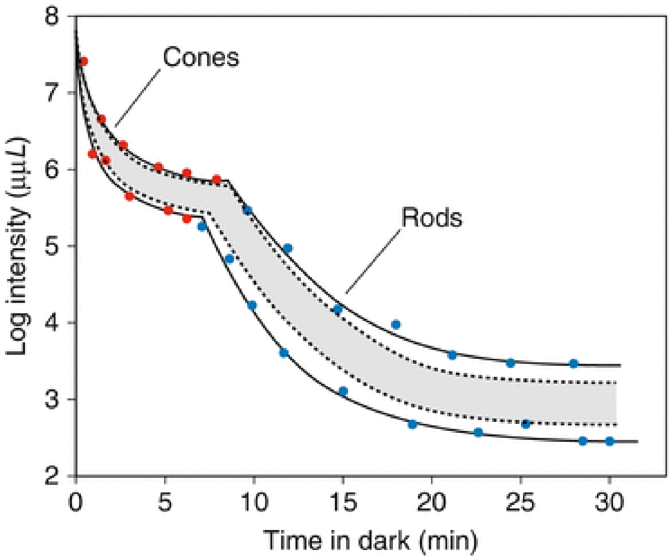


Figure 2.11: Adaptation to the dark over time. 80% of test subjects lie in the shaded area. Reprinted from [Lak16].

2.2.3 Impurities of Cornea and Lens

What we perceive as glare is forward-scattered light. Scattering usually happens due to the eye's impurities and irregularities, mostly in the cornea and lens, as shown in Figure 2.12. These small errors (further called particles) break the light waves at different angles for different wavelengths, which then strike the retina at different positions. This scattering of particles leads to a common glare phenomena with a corona and a rainbow halo, as shown in Figure 2.13.

2.2.4 Light Diffraction Inside the Eye

In addition to seeing light as a single particle, it is commonly known that light also has wave properties. This is called the *light wave-particle-duality*, and light is the only known medium in which this phenomenon appears. In the following, we view the refraction on particles as a diffraction of waves on an obstacle.

The incidence of light through a circular hole creates an airy disk, which is defined by the wavelength of light and the aperture's size [Air35]. This airy disk can be generated by calculating the Fourier transform of the aperture.

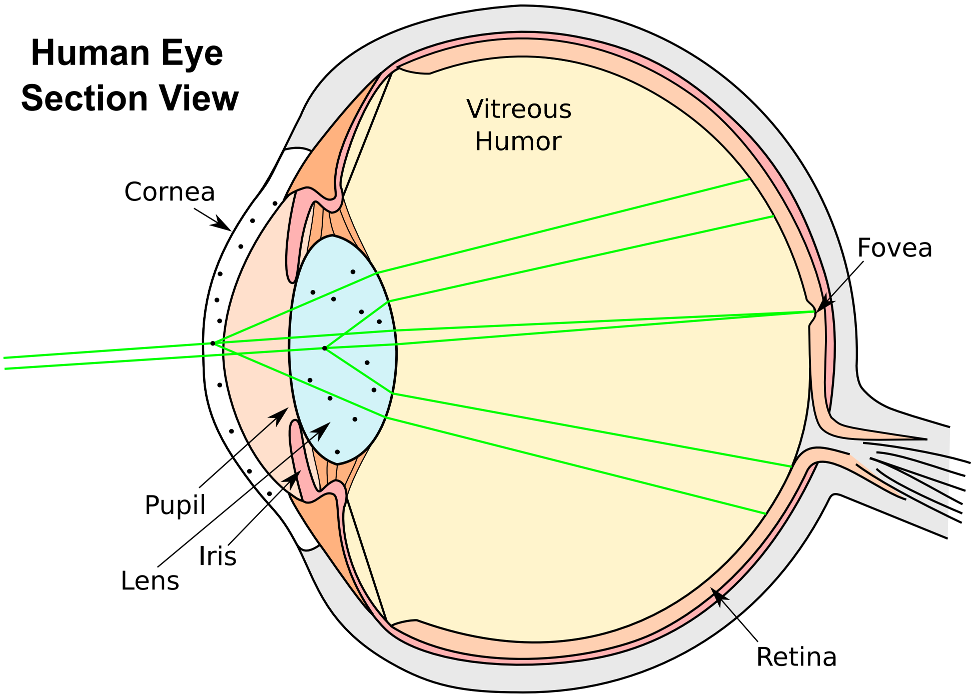


Figure 2.12: Light scattering in the human eye of a monochromatic light, only regarding the particle characteristic of the light, after Ritschel et al. [RIF$^+$09]. Adapted from [SZ07].

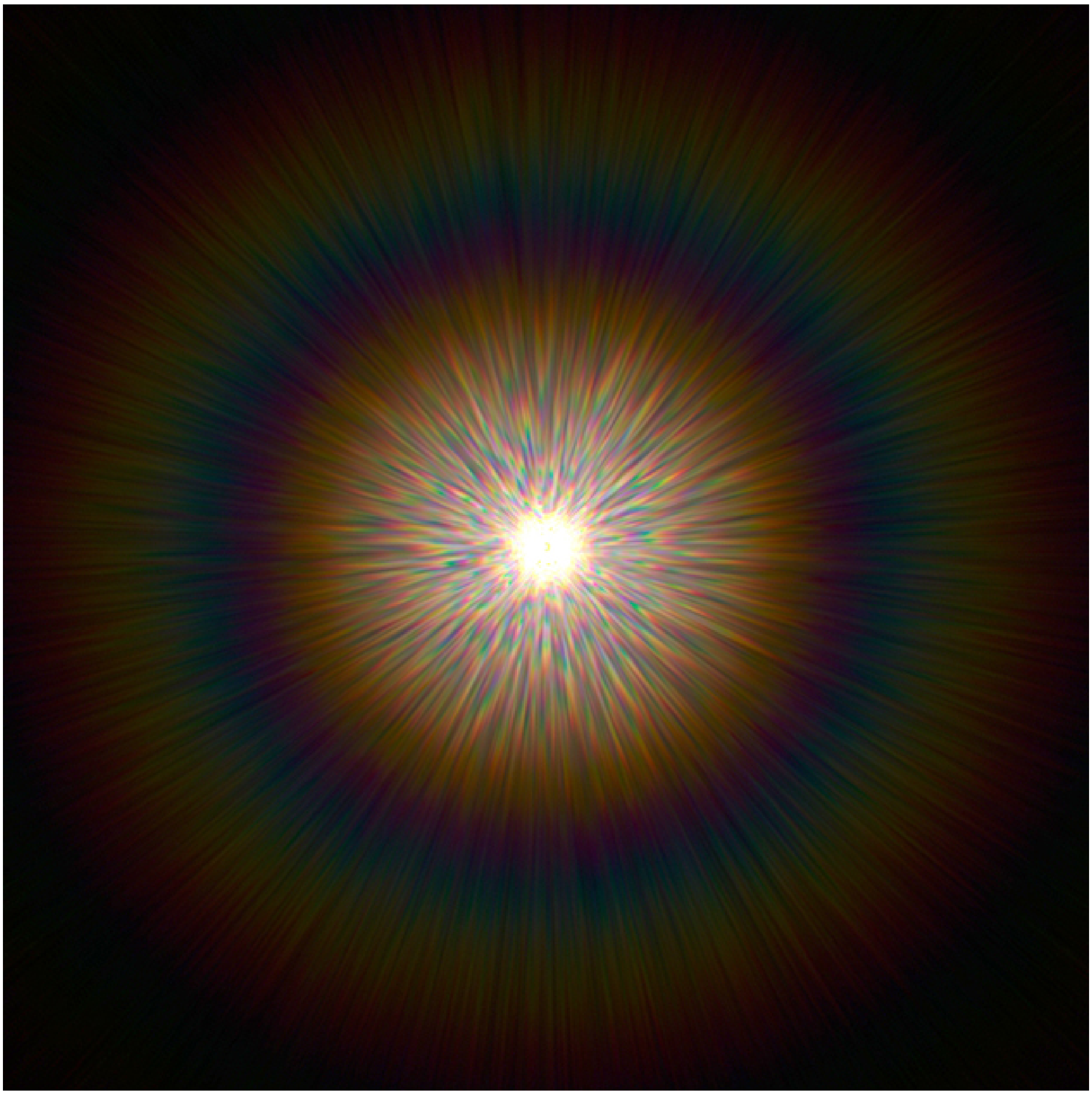


Figure 2.13: Simulation of a common glare phenomenon.

2.3 Brightness Range Compression for LDR Displays

Calculating the full humanly visible brightness range from 10^{-6} to $10^{8}\frac{cd}{m^2}$ does not pose any significant problems anymore for current hardware. A challenge that remains is the displaying of high dynamic range (HDR) scenes, on low dynamic range (LDR) monitors. An average, modern display has a brightness of $250\frac{cd}{m^2}$ with 1000 steps in contrast. These specifications are still significantly below the full brightness range humans can perceive. On the other hand, the human brain can distinguish only approximately 100 shades of gray at once. Therefore, it is, in theory, possible to compress the full brightness of a scene in a way that it can be displayed perceptually "correct". Many tone mapping algorithms aim to achieve proper perceptual compression.

2.3.1 Tone Mapping

The process of compressing the full brightness range of a scene in order to be displayed on a LDR monitor is called *tone mapping*. It is necessary to use perception-based algorithms to achieve visually acceptable results. A simple mapping of ranges usually leads to washed-out images, and clamping the maximum values will result in bright and overexposed areas.

Even though a lot is already known about the anatomy of the eye, it is still not completely understood how our brain processes the information it receives from the eyes. Due to the complexity of color and brightness perception, it is still challenging to generate an overall perfect solution for brightness compression. Furthermore, it has to be noted that perception is subjective and might differ from person to person. Multiple color-comparison tests show that the same person under the same conditions might still have different results on a different day [Pur19]. Therefore, it is challenging to generate a completely "correct" mapping of brightness levels and colors. Many algorithms aim to solve this task in different ways, however, some fundamentals regarding perception are retained.

2.3.2 General Approach

In common tone-mapping approaches (see Section 3.4), the fact that human light perception is not linear, but logarithmic, is exploited. We can distinguish more shades of darker colors than brighter ones [GW07]. Therefore, the bright regions of an image can be compressed more than the dark regions without loss of information.

Furthermore, human perception is very local and changes when focusing on one area or the other. Our brain can perceive two distant pixels with the same intensity as different brightnesses due to their surroundings. Therefore, algorithms should only compress the values of distinct regions and not the whole image.

2.4 Fast Fourier Transformation

The Fast Fourier transformation (FFT) is a computationally efficient way to calculate a discrete Fourier transform [Nus81].

A two-dimensional discrete Fourier transform Y of an m-by-n image X is defined as [MAT20]:

$$Y_{p,q} = \sum_{j=0}^{m-1} \sum_{k=0}^{n-1} \omega(m)^{j \cdot p} \cdot \omega(n)^{k \cdot q} \cdot X_{j,k}, \tag{2.1}$$

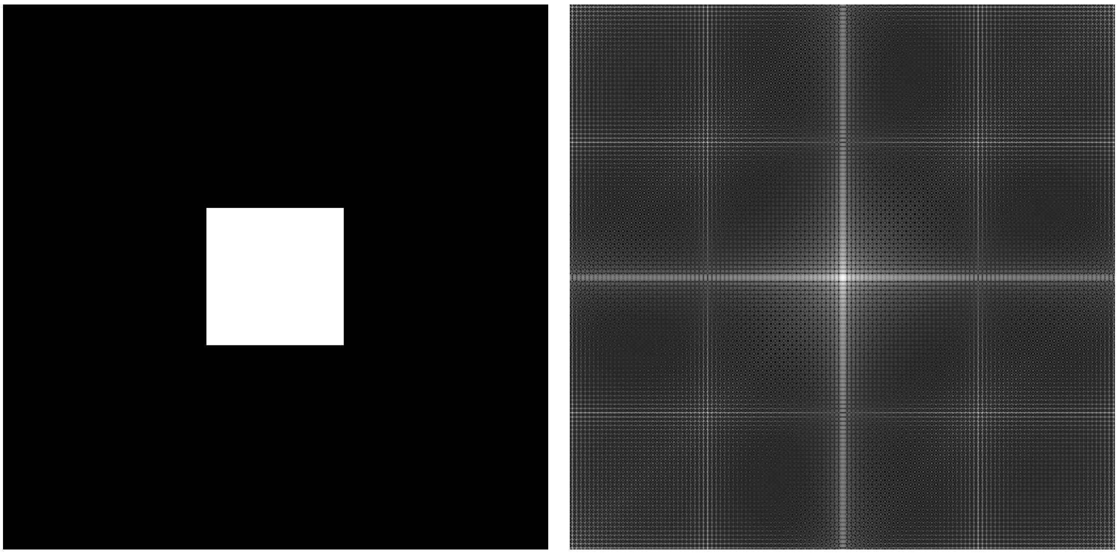
$$\omega(x) = e^{-2\pi \mathrm{i}/x}. \tag{2.2}$$

It is a method to transform discrete data (e.g., an image) into the frequency domain, which can then be processed further and eventually be transformed back into the spatial domain without loss of information.

Every image has a unique representation in the frequency domain; some samples can be seen in Figures 2.14-2.17. This transformation can be used for many practical applications, for example removing high frequencies which leads to a blur of the image.

A naive 2D discrete Fourier transform has a runtime of $O(N^2)$, where $N = \max(m, n)$, which is mostly too slow for real-time applications. Using an FFT, a runtime of $O(N \cdot \log N)$ can be achieved.

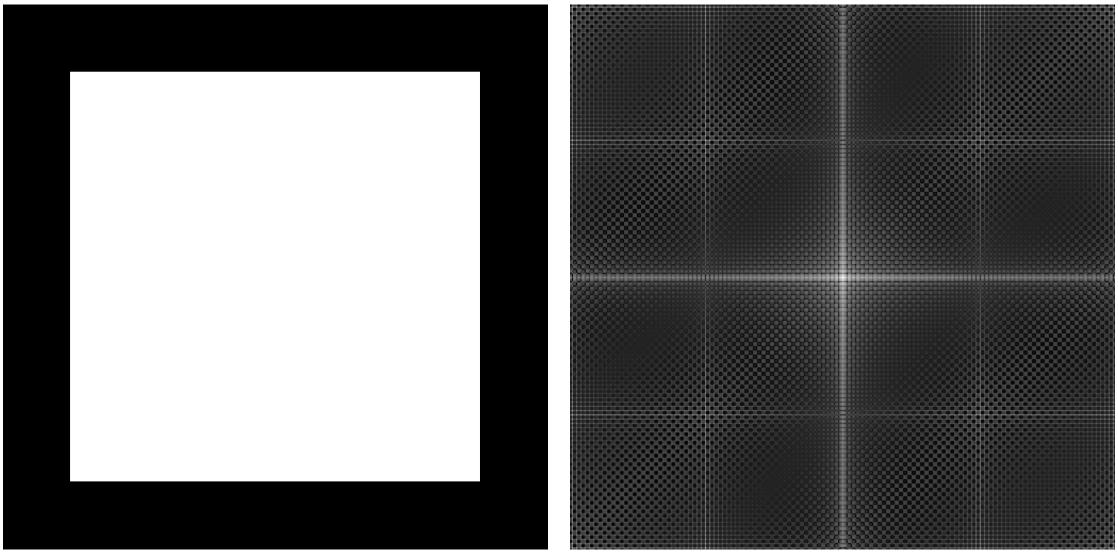
There are multiple algorithms proposed to solve this transformation as fast as possible. Most algorithms make use of the divide-and-conquer method, which recursively calculates the solution of smaller sections and then combines them into one final output [Nus81]. This method inflicts no error, but the precision of the processor used. Since our approach is an approximation nonetheless, we are not influenced by these minor errors.



(a) Discrete data.

(b) Fourier transform.

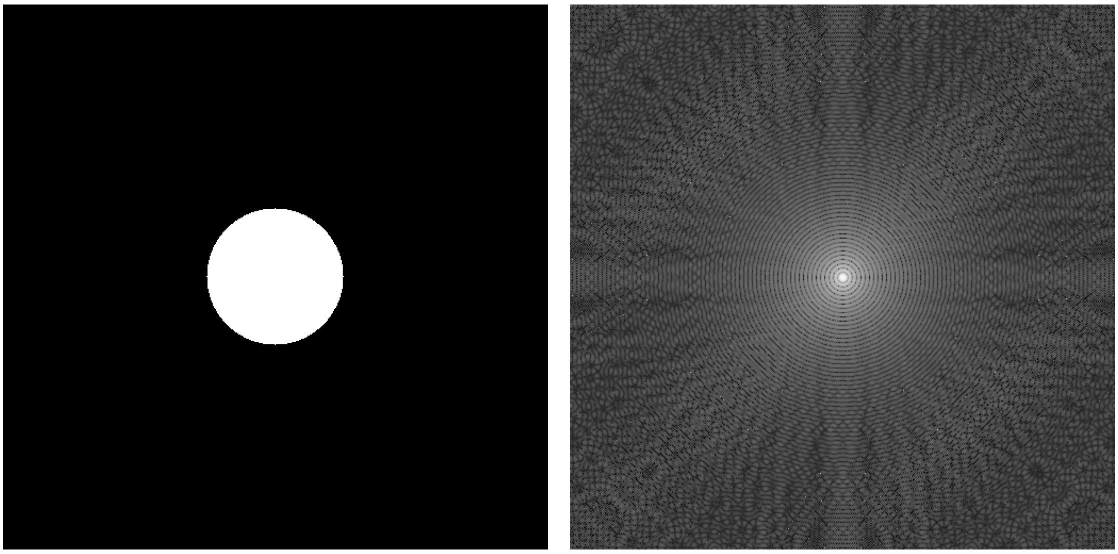
Figure 2.14: Comparison of discrete data and Fourier transform of a small square.



(a) Discrete data.

(b) Fourier transform.

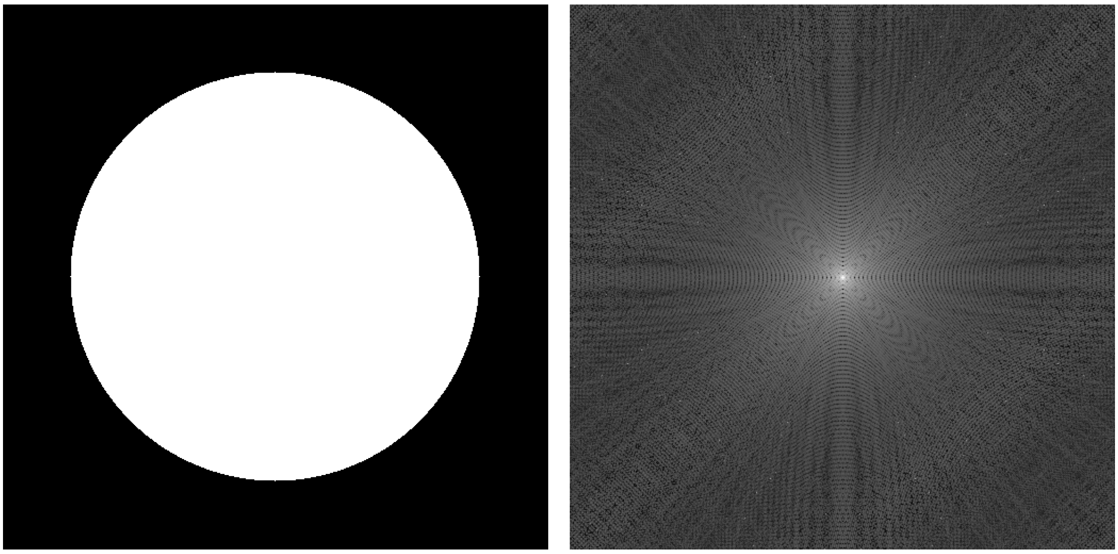
Figure 2.15: Comparison of discrete data and Fourier transform of a big square.



(a) Discrete data.

(b) Fourier transform.

Figure 2.16: Comparison of discrete data and Fourier transform of a small circle.



(a) Discrete data.

(b) Fourier transform.

Figure 2.17: Comparison of discrete data and Fourier transform of a big circle.

CHAPTER 3

State of the Art

In this chapter, the current state of the art in related work regarding the simulation and perception of light is explained in detail.

Human perception depends on the following influences:

- Luminance adaptation
- Glare
- Visual acuity
- Photopic, mesotoic, and scotopic vision

The approaches presented in the following aim to simulate one or more of these perceptual effects.

First, some recent applications that require correct light simulations are presented. Secondly, approaches that aim to simulate very bright light on a LDR screen are explained. Next, the adaptation of the human eye to light or dark illumination, as well as day and night vision and their simulations, are discussed. Finally, perceptually inspired tonemappers and their methods are introduced.

3.1 Light Simulation Tools

Simulating light in a physically inspired and, therefore, perceptually pleasing way can be used in various applications. These tools include, but are not limited to, architectural design tools, realistic driving simulations, or the study of eye diseases. In this section, such applications and their methods are presented.

Krösl et al. [KEW$^+$19] simulated view-dependent eye diseases using an eye tracker. Different types of cataracts were implemented to allow for better planning of escape route signs or placement of luminaries in a 3D scene. Eye tracking was used to simulate intense blinding effects and different intensities of the eye diseases, whether the user was looking at bright or dark areas on the scene. The authors aimed to generate an improved understanding of the perception of a vision-impaired person for better handling for architects, doctors, as well as normal-sighted people. However, this approach does not include physically realistic simulation of glares or scotopic vision.

Meyer et al. [MGVM16] used perception-based methods to simulate glare situations at night-time on conventional displays. They focused their research on contrast reduction for night-driving simulations and glares from oncoming vehicles' headlights. Finally, their work was used to study and test reaction times of drivers under realistic conditions. This research does not include VR simulations, but is merely targeted at 2D displays.

Similarly, Bolling et al. [BSJ10] used an actual lamp in a virtual environment setup with a projector to simulate dazzling light on different car wind-shields. In their study, this method proved to simulate a low-lying sun well enough. However, this kind of setup is very specific for this sort of simulation and therefore difficult to adapt to other applications.

Jones and Reinhart [JR16] proposed an application that gives predictions of daylight glare probability, task luminance, and contrast, in real time for architectural design. Their work was intended as an addition to offline rendered scenes, in order to provide feedback on the impact of architectural decisions on visual perception faster and more detailed than before. In contrast to the work of Jones and Reinhart, our simulation of human vision is able to run in real time in a VR application and therefore provides a higher level of immersion.

3.2 Simulating Bright Light Sources

Simulating the glare that surrounds bright light sources is used to artificially increase the perceived brightness range of a displayed scene – we perceive lights with glare as more intense than those without [YIMS08]. Since most common displays are not able to replicate the full humanly visible brightness (i.e., LDR screens), a way of increasing the perceived brightness of the (should-be) dazzling parts of a digital image is necessary to improve realistic perception.

The most straightforward solution to simulate bright light sources is a Gaussian convolution, i.e., a blur, applied to the bright pixels of a scene, as commonly used in many game engines [YIMS08]. This results in a uniform glare or glow around light sources (see Figure 3.1). While in some cases this might look quite similar to what we perceive, applying a Gaussian blur does not take any physical properties into account. Applications that just use a Gaussian convolution do not aim to approximate a real-world glare effect, but merely try to generate an visually reasonable impression.

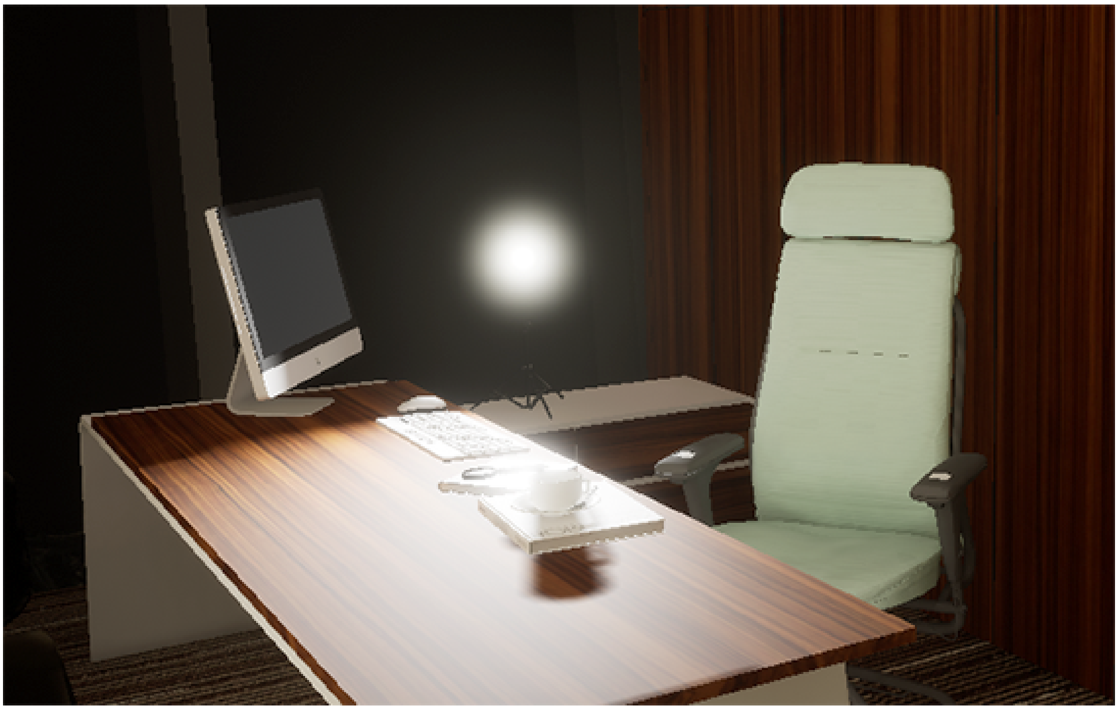


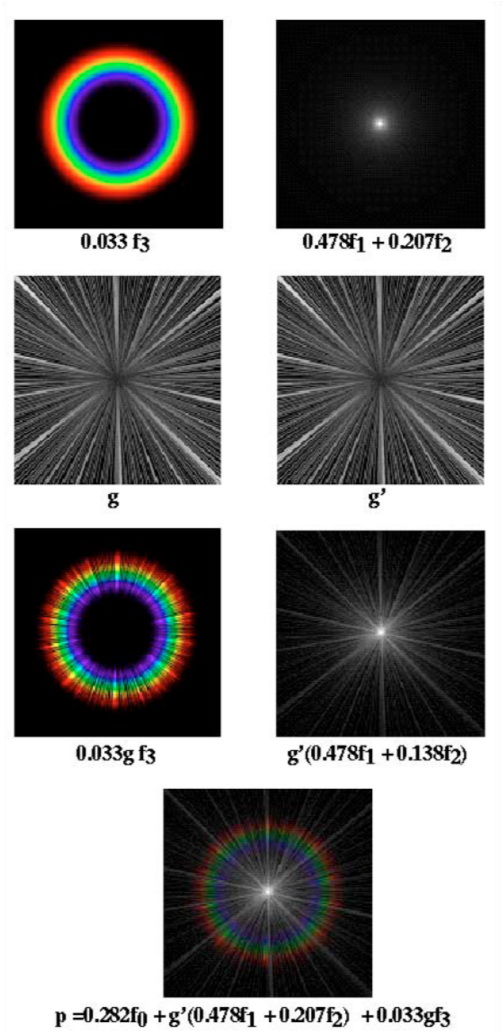
Figure 3.1: Example of a simple bloom. Generated using UE.

Algorithms generating a physically based glare effect [KMN+04, RIF+09, vdBHC05] often make use of a point spread function (PSF). A PSF models the distribution of light on the retina in the eye. It is usually rendered into a texture and then applied to the bright pixels of the final output image, e.g., via multiplication in the frequency space.

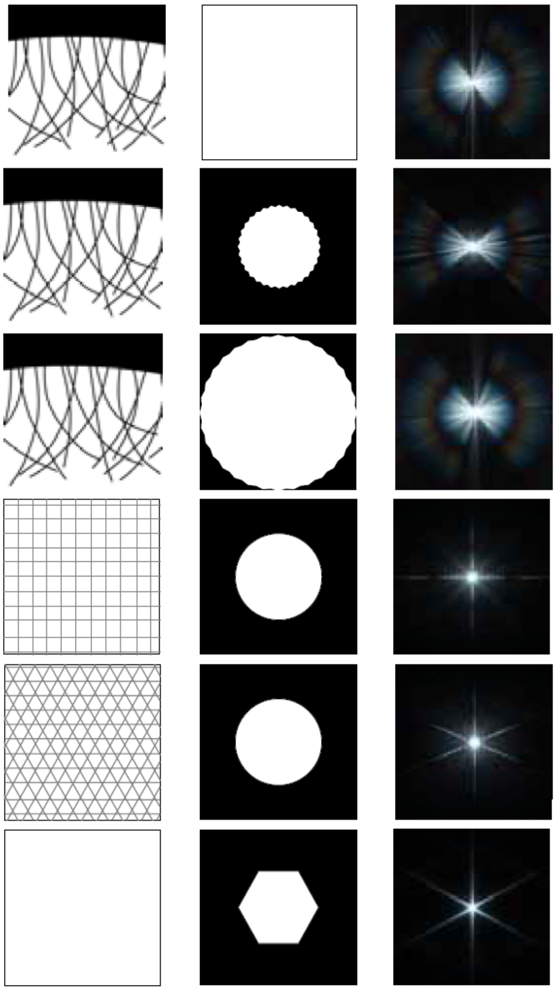
Yoshida et al. [YIMS08] conducted a user study with ten subjects and concluded that the perceived brightness of a glare simulation with a Gaussian kernel is of the same or higher intensity than convolution with a PSF. This study did not include the realism of the glare but only the perceived brightness. Depending on the aim of the developed application, i.e., more brightness, a simple Gaussian bloom would be sufficient as glare. In order to simulate high realism, further studies are necessary.

Similarly, Facchin et al. [FDZ17] conducted a manual glare-effect test with 101 printed cards that show different bloom intensities. Their results show that the threshold for perceiving something as glaring was higher for older people (60–75 years old) than for younger ones and on average lies between 48.86 % and 58.24 % glare-effect intensity. This study was held via printed cards – to our best knowledge, no comparable study on a display or in VR has been conducted.

Spencer et al. [SSZ+95] generated a scotopic pointspread filter based on quantitative formulae of the perceptual contribution of the bloom to light for an average viewer. The computation includes different visual effects like the lenticular halo and flare lines, which are then combined into one PSF (see Figure 3.2a). While the authors based their



(a) Different perceptual formulae and their influences on the final PSF. Image reprinted from Spencer et al. [SSZ+95].



(b) Different filters and apertures on the left, generate the PSFs on the right, based on Fourier optics. Images reprinted from Kakimoto et al. [KMN+04].

Figure 3.2: Different methods that generate a PSF.

work on medical observations, different influences were generated separately and then combined into one output PSF. Nevertheless, when looking at the anatomy of the eye (see Section 2.1) and the way light propagates, it is clear that those influences do not form separately, but in an interrelated way. A method that is able to calculate the glare effect as one, taking this interdependence into account, is to generate a PSF via Fourier transforms relying on the basics of wave optics.

Kakimoto et al. [KMN+04] generated physically based glare images using Fraunhofer diffraction [Goo05]. They used aperture masks, i.e., binary images of different layouts of apertures, determining the opacity of each point, where 0 is opaque and 1 is transparent. They applied Fourier transforms to those masks at different wavelengths for the three different color channels (red, green and blue), which were then combined into one RGB image. This approach includes the influences of the pupil size as well as eyelashes surrounding the eye. Furthermore, this technique can also be used for cameras with different filters and diaphragm sizes, as shown in Figure 3.2b. However, their approach is static and does not include real-time adaptations to changing parameters.

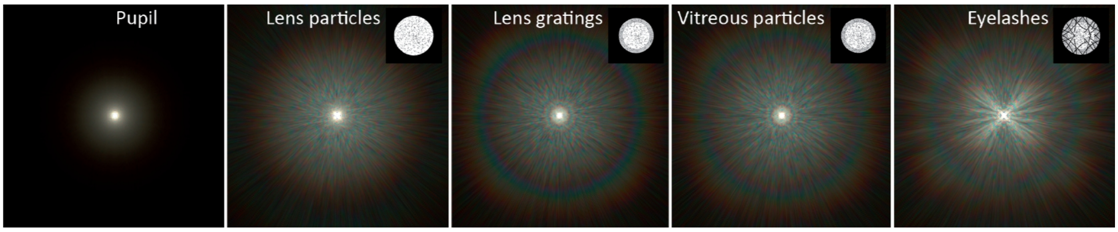
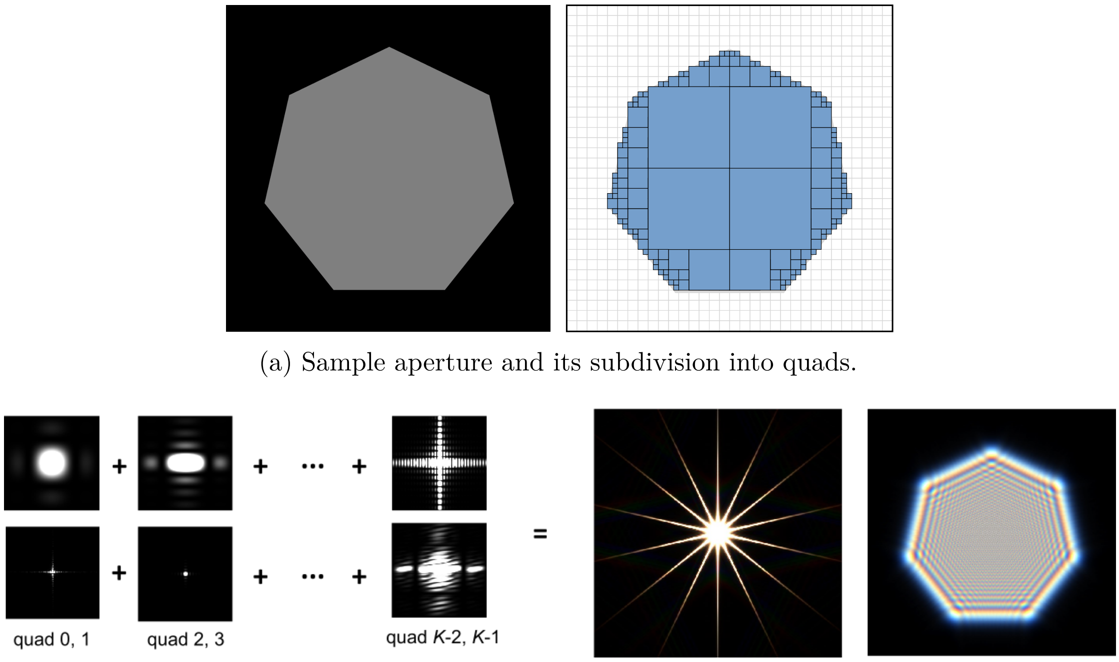


Figure 3.3: Different influences on the PSF by Ritschel et al. [RIF$^+$09].

Ritschel et al. [RIF$^+$09] used the Fresnel approximation to Huygen's principle, which is a more generalized diffraction equation that includes Fraunhofer diffraction as a special case. It describes the principle that every point on a wavefront is a source of wavelets, and is more accurate than Fraunhofer diffraction when the diffraction pattern is viewed at a short distance (near field) [Goo05]. Ritschel et al. considered Fresnel diffraction more appropriate, due to the short distance between the pupil and the retina. In their approach, the aperture mask is multiplied with a complex exponential before the Fourier transform at a single wavelength at the average of the visible human field. This monochromatic PSF is then converted into a spectral PSF by averaging over multiple scaled copies along the visible human range. Their approach takes multiple influences into account such as the pupil size, lens particles and gratings, vitreous particles and eyelashes, as shown in Figure 3.3. The perceptual glare effect implemented in this thesis builds upon this approach and adapts their method to run in real time VR.

Scandolo et al. [SLE18] proposed a new real-time method for a more efficient calculation of the Fourier transform used to generate the PSF. They subdivide the aperture image into multiple quads, which are then solved using a closed-form solution. Finally, the different quads are then re-combined into far- and near-field diffraction patterns (see Figure 3.4). This method works well on uniform circles or boxes, i.e., any shape that can be approximated with large squares. However, the authors do not include eye particles or other noise in their method. We assume that their approach would not work as efficiently with an aperture that takes varying pupil sizes and particles into account, such as the one we use in our approach.

Other effects that greatly influence the perception of light in a scene are lens-flares, like starbursts and ghosts, as shown in Figure 3.5. These effects generally appear when pointing a camera lens at a very bright light source, due to the interreflection of light



(a) Sample aperture and its subdivision into quads.

(b) The individual Fourier transforms of the quads (left) can be summed to generate the far-field diffraction (middle) and near-field diffraction (right) of the aperture.

Figure 3.4: Different stages of the PSF generation proposed by Scandolo et al. Reprinted from [SLE18].

in the lens itself. The human eye only consists of a single lens, so generally no internal reflection occurs. However, the principle of an aperture, which results in star-shaped artifacts when light diffracts, remains the same.

Hullin et al. [HESL11] proposed a full model that renders different lens systems in real time, including physically based camera aperture f-stops. However, they generate the starburst texture in a precomputation step and apply it to the image during runtime. This approach is therefore static and not able to adapt to varying parameters in real time.

Walch et al. [WLS$^+$18] proposed a real-time workflow for physically plausible lens-flare rendering, using measurements of real cameras. This algorithm does not take starbursts into account and, therefore, does not represent perceptual glare properly, i.e., their results do not sufficiently resemble looking into bright light with human eyes.

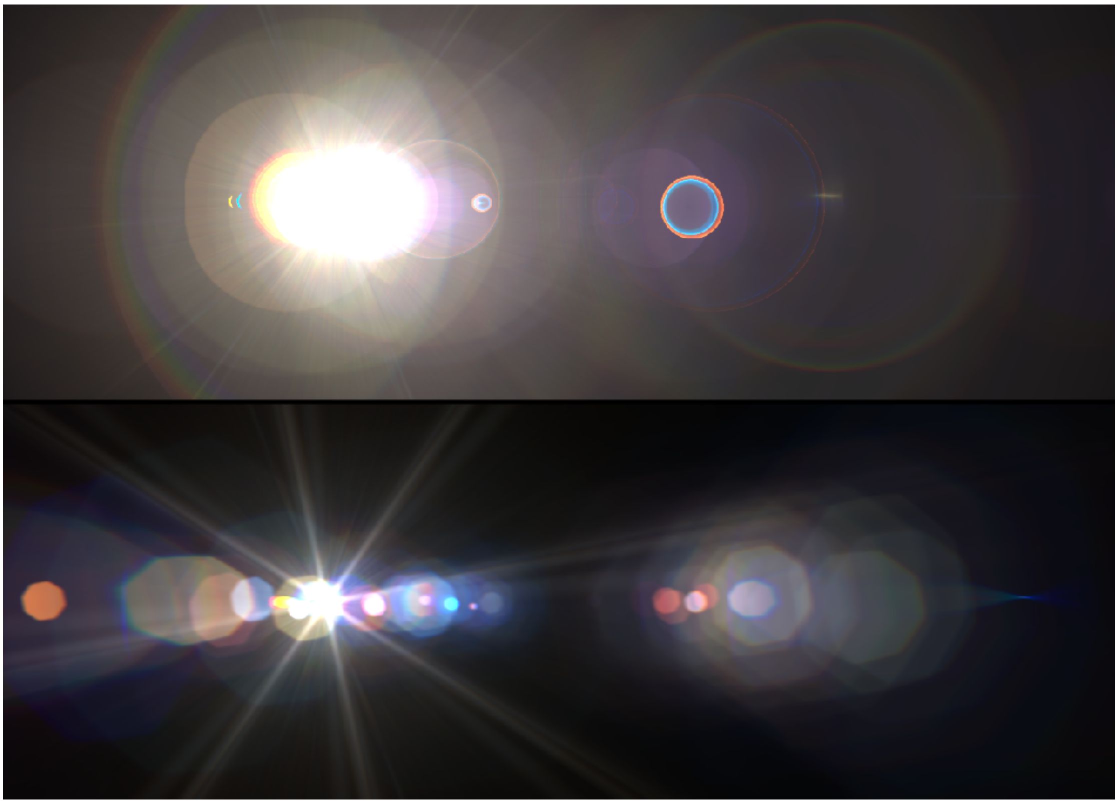


Figure 3.5: Renderings of lens flares. Reprinted from [HESL11].

3.3 Luminance Adaptation and Scotopic Vision

The brightness levels perceivable by humans range from 10^{-6} (faint starlight) to $10^{8}\,\frac{cd}{m^2}$ (sunlight) in total. However, we are not able to take in such a broad range at the same time. In order to see under different brightness levels, the human eye needs to adapt. We can distinguish between approximately 100 shades of gray, from white to black, under unchanged conditions. Our eye's process of changing from one lighting condition to another is called *temporal luminance adaptation* and has to be modeled in computer graphics applications [KDMF16]. The works presented in the following focus on the simulation of rod vs. cone vision. Some approaches already include tonemappers in their methodology. However, they are not focused on compressing the human visible luminance to the displayable range of a monitor like traditional tone mapping, but aim to reproduce the human response to light and dark.

Larson et al. [LRP97] proposed a global tone reproduction operator, based on local adaptation, using a histogram with real-world luminances, where 1 pixel represents 1° in the visual field. The authors matched a custom curve in order to preserve contrast, based on human perception, for a target display. This approach also included veiling luminance (i.e., glare), color sensitivity and the loss of visual acuity in their tonemapper.

However, a perceptual glare or scotopic color vision is not included in this approach.

Ledda et al. [LSC04] proposed a model for local eye adaption, which can be used as a static or temporal effect on HDR images and videos. They relied on the basics of rod and cone vision with respect to current scene intensities, i.e., they first transform an HDR RGB sample to the respective photopic and scotopic luminance based on a linear transform of an XYZ stimulus. Then they transform the intensity of the active stimulus via a perception-based equation to achieve the final pixel brightness. Even though they identify regions of photopic, mesotopic and scotopic vision, no color adaptation is added to this approach. Therefore, intense, saturated colors in a dark region might appear as incorrectly shaded.

Ferradans et al. [FPBC09] presented another model based on human visual adaptation. They extended the Naka-Rushton equation [SEC84] to improve an image's local contrast similar to human vision. They combined the mapping of luminances as well as colors in order to achieve an even more realistic look. While mapping colors and luminances separately might give a better impression of the overall saturation, this also leads to four times higher runtimes, which is not feasible for VR applications.

Meyer et al. [MGVM16] simulated the adaptation of an eye to short, bright flashes of light as well as the re-adaptation to darker environments in their nighttime driving simulator. They measured subjective contrast perception curves and used those results in order to generate an algorithm that adapts to different input situations over time. This approach has been specifically developed for driving simulators. While their results seem promising, they have only been tested in one specific scenario, and no general evaluation has been made.

Wernikowski et al. [WMP19] presented a VR framework that employs the natural eye adaptation process as well as a linear adaptation process and compared those in a user study. They concluded that their test subjects preferred a 25-second linear adaptation over a medically determined perceptual adaptation equation. When evaluating their perceptual adaptation algorithm only, the most preferred length was 5 seconds, which is significantly less than real-world temporal eye adaptation (up to 30 minutes). This study only relies on the intuitive assessment of their participants and does not compare the methods to other works or real-world scenarios.

3.4 Perceptual Tonemappers

When looking at recent research [EUM16, KEW$^+$19, YIMS08] regarding the perception of light and lighting situations, a very common topic is HDR tone mapping. This is the process of adapting an image (either natural or synthetic) with a broad range of light information to be displayed on LDR output devices, like standard screens, preferably without loss of visual information. Numerous algorithms have been proposed [EMU15, KMS05, RSSF02, TMM19] and compared in scientific literature [EUM16, ČWNA08]. The majority of the presented algorithms aim to compress HDR images as good as possible, however, they might neglect perceptual effects, like rod or cone vision. A recent review by Eilertsen et al. [EUM16] concludes that tone-mapping algorithms have come a long way, and while there have been many approaches on a universally applicable model, there are still problems regarding the fine-tuning of the algorithms' parameters, as well as the treatment of colors.

One of the most commonly used tonemappers was proposed by Reinhard et al. [RSSF02]. It uses a technique, inspired by traditional printing, that enhances contrast while compressing an image's luminance range. A Gaussian pyramid is generated in order to detect the biggest area around a pixel where no contrast changes appear, and the luminance of that pixel is scaled in a way that increases the local contrast.

An early approach regarding perceptual tone mapping by Krawczyk et al. [KMS05] took effects such as visual acuity, glare, day, and night vision into account. In their work, they also made use of a Gaussian pyramid to construct three maps for adaptation, acuity, and glare, where different scales allow for individual spatial processing. Finally, applying an equation based on the current pixel's luminance and tone-mapped luminance, the effect of scotopic vision is simulated. This algorithm can be applied in real time to any frame, due to the discovered spatial similarities of the effects, i.e., by using a Gaussian pyramid to compute three maps at once. In contrast to other common tone-mapping algorithms, this method focuses not only on mapping the full contrast of an image onto a smaller range, but includes visual effects that also influence the perception of scenes.

Čadík [Čad07] proposed a global and local hybrid method to achieve perceptually realistic tone-mapping results. They used a first global pass and simultaneously generated an enhancement map of very bright and very dark areas, which were then improved further in a local pass. They, therefore, ensured the preservation of overall image attributes as well as local contrast enhancement.

Mantiuk and Markowski [MM13] employed an eye tracker to be able to simulate the human perception compression of light more accurately. They calculated a saturation constant depending on a Gaussian function centered at a temporally filtered gaze point in the image.

Vangorp et al. [VMGM15] generated a local adaptation map based on their psychophysical experiments, by using the best fitting model out of a large-scale set of models. Similar to Mantiuk and Markowski [MM13] they applied a tonemapper based on a user's viewing direction, using their model of local adaptation.

Yoshida et al. [YBMS05] conducted a psychophysical experiment that compared real-world scenes with HDR tone-mapped images on an LDR display. They evaluated five features, i.e., brightness, contrast, detail reproduction in the dark as well as bright image region and "naturalness", for seven different tone-mapping algorithms. They showed that the features in question were perceived very differently for different algorithms. For overall naturalness, Drago et al. [DMAC03] achieved the best average value of all algorithms.

In our approach, the Academy Color Encoding System Filmic tonemapper [DFD+15] is used. While this tonemapper does not take perceptual details into account, it is highly customizable and has efficient runtimes. Furthermore, it is an industry standard for television and film. We believe that this tonemapper in combination with our perceptual adaptations can lead to perceptually accurate results.

CHAPTER 4

Overview

In this chapter, the main stages for simulating human vision in a medically based way, to achieve perceptually pleasing applications, are discussed. Human vision is highly dependent on the way light enters the eye. Different effects only appear under specific lighting scenarios. For example, we see a pulsing corona when looking at bright light sources, or our color perception changes during night-time. Since these phenomena do not appear naturally when looking at a screen – due to the displayable luminance range only covering a small part of the range perceivable by a human – we need to simulate them to achieve a realistic representation of bright light sources or light in general.

Therefore, perceptual algorithms need to be able to respond to different lighting conditions in a similar way as the actual human eye responds to them. We have evaluated all major influences on visual perception and explained them in detail in Chapter 2. We consulted an optometrist for additional information on the visual processing of light in the eye and the importance and impact of light rays on our vision. We conclude that the four most significant influences on vision are the following:

- The temporal eye adaptation to bright and dark scenarios, due to the adaptation of rods and cones (see Section 2.2.2 v.).
- The perceptual glare we perceive when looking at a bright light source – colorful patterns that appear through the scattering of light in the eye, i.e., ocular straylight (see Section 2.2.1).
- In dark surroundings, specifically at less than $0.03\frac{cd}{m^2}$, we experience a visual acuity reduction – edges blur, and details become hard to discern (see Section 2.2.2 ii.).
- Scotopic color vision (i.e., night vision) is significantly different from standard day vision – colors change in intensity and seem more washed out (see Section 2.2.2 iii.).

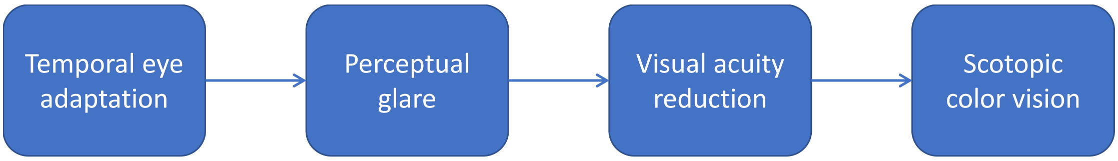


Figure 4.1: Overview of all the stages to improve the perception in simulations.

We defined these visual effects as different stages of a renderer, their order of implementation can be seen in Figure 4.1. In the following sections, we give an overview of the simulation of temporal eye adaptation, perceptual glare, visual acuity reduction, and scotopic color vision, as well as their interdependencies. A more detailed discussion about these effects and how they work can be found in Chapter 5. Further details about the practical implementation of these effects via a standard renderer can be found in Chapter 6.

4.1 Simulation of Temporal Eye Adaptation

Temporal eye adaptation occurs when the luminance of the focused region changes – depending on external influences like a light switching on or off, or simply our viewing direction [Pur19]. Our eyes can adapt from looking out a window into a sunlit scene over to glancing at an object that is only dimly illuminated by indirect light within milliseconds. However, the adaptation to large changes can take up to 30 minutes (see Section 2.2.2 v.). The adaptation of the eye enables us to see in dark surroundings as well as in bright sunlight and enlarges the humanly perceivable brightness range significantly.

The adaptation of the human eye to changing illuminations does not happen instantly but converges over time to the optimal adjustment for a given brightness. This is due to the varying destruction of proteins in the eye when hit by light and their constant reconstruction speed [Pur19]. Following Krawczyk et al. [KMS05], we simulate this by applying an increase or decrease of the average luminance of the frame following an exponential function over time. In contrast to their approach, which is uniform over the whole frame, we calculate the frame luminance weighted by the user's viewing direction. Since this effect influences the whole human perception, it has to be applied first, to be able to generate realistic simulations for all the following effects.

4.2 Simulation of Perceptual Glare

The humanly perceived glare is generated by particles in the eye scattering light rays. Due to this, humans do not only see a single sharp, bright area but flickering rays and a corona around a light source [RIF^{+}09]. This is medically based on the fact that there are micrometer-sized particles, i.e., impurities in structure, inside the human eye, that diffract the incoming light and create these scattered appearances, which are then called ocular straylight. This straylight can be physically based on the wave characteristic of the light, i.e., wave optics. Light rays diffract inside the eye, which leads to glare patterns [Goo05]. For more background details please refer to Section 2.2.1.

Using the diffraction of light on a slit or aperture, such as the pupil, and including the impurities of the eye, the resulting pattern can be simulated quite accurately by a Fourier transformation [KMN+04]. This is also known as the PSF of the eye, which approximates the spreading of light on the retina. This generated image or kernel can then be applied to the bright regions of a rendered scene to simulate perceptually correct glare.

Previous glare simulations are static due to computational reasons. However, what we see in reality is not entirely static, but flickers and is highly dependent on constant eye movement, the pupil size, or the viewing direction. Due to the angle at which light enters the eye, different scattering patterns occur, and it is not sufficient to only assume that the user is looking straight ahead since our eyes are continually moving. Furthermore, the bigger the pupil size, the more light can enter the eye and a more intense glare phenomena can be perceived, i.e., glare generally is more present in darker surroundings than in brighter ones. Therefore, in order to simulate human glare perception as correctly as possible, a real-time approach that takes these factors of movement into account is necessary.

We base our perceptual glare on the approach by Ritschel et al. [RIF+09], who use Fresnel diffraction to simulate the scattering of light. We adapted their approach to run in real-time VR/AR and added eye tracking to increase realism. Our simulation of perceptual glare is dependent on the perceived brightness of lights or light sources. Therefore, it has to be applied after the temporal adaptation.

4.3 Simulation of Visual Acuity Reduction

Especially in dark scenes, the human eye suffers from loss of visual acuity. The photoreceptors in the eye that are mainly active during mesopic and scotopic vision, the rods, are not present in the fovea, which is the point of sharpest vision in the eye [Pur19] (for details see Section 2.2.2 i.). Therefore, in low light conditions, we only receive signals from regions in the eye that are not optimally in focus (see also Section 2.2.2 ii.). Hence, the acuity is reduced, and edges from objects and borders seem more blurred than during photopic/day vision.

This effect does not naturally occur when looking at a display, since screens are not able to represent such low luminances. The cones are always triggered by a monitor, especially with liquid crystal displays (LCDs) as used in our work. Therefore, we need to simulate a visual acuity reduction to maintain realism in low-light scenes. This effect influences only the pixels that are below a specific luminance – for humans $10\frac{cd}{m^2}$. These pixels have to be adapted by blurring them. We build upon the functionality proposed by Krawczyk et al. [KMS05] and apply a Gaussian filter depending on the pixel's luminance. Our approach uses a faster function fit than Krawczyk et al. [KMS05] to approximate the loss of visual acuity in the dark. No eye tracking is necessary to implement this effect since the output is merely dependent on the luminance, and this does not change whether a pixel is in focus or not.

4.4 Simulation of Scotopic Color Vision

Similar to the loss of visual acuity in darker scenes, color perception also changes under low-light conditions. This is because rods are only able to see "one color", and are not able to distinguish between thee different wavelengths of light, like cones are (see Section 2.2.2 iii.). Therefore, we perceive a blue to grayish color shift at night time or dimly illuminated scenes.

This effect cannot appear naturally when looking at a monitor since the minimum displayable luminances still trigger the cones in the eye. However, to realistically simulate a scene that is not well lit, the shift in color vision in mesopic and scotopic conditions has to be taken into account.

Following the approach by Krawczyk et al. [KMS05] and using the sensitivity function of scotopic vision, a color shift can be applied to the image. This has to be done last, to avoid modified pixel luminances beforehand, which would affect other stages.

CHAPTER 5

Simulating Human Vision

In this chapter, the methodology behind simulating human vision in VR is explained. First, we describe the temporal eye adaptation to dark and bright scenes. Secondly, the perceptual glare and its generation algorithm are discussed. Next, we explain the simulation of the loss of visual acuity during night time or dark scenes. Finally, the scotopic color vision is illustrated.

5.1 Temporal Eye Adaptation

Temporal eye adaptation is the process of adjusting to light or dark environments. The human eye needs this adaptation to be able to see in low-light scenes and still not be entirely blinded by bright sunlight. However, this process is not instant but converges towards an optimum over time. When turning off a light switch, so the room becomes dark, we are not able to see anything at first, yet after a few seconds, we can make out silhouettes and some minutes later, we are fully adapted. In this section, our approach to simulate temporal adaptation is explained in detail.

As a first post-processing step, global temporal eye adaptation is calculated based on the HDR values in linear RGB scene values. This step is done before we apply our glare effect, in order to base the calculation solely on the scene luminances and not on additional simulated effects. First, we weight the RGB colors with a 2D Gaussian function centered at the viewing point. Therefore, lights at the periphery have little to no influence on our temporal adaptation. Second, we downsample our scene to a size of 128×128 and generate a Gaussian pyramid until 16×16 pixels. We then take the average over these 16×16 pixels as a single frame average. We convert the RGB value to luminance using a D65 white point as

$$Y = 0.2125 \cdot R + 0.7154 \cdot G + 0.0721 \cdot B, \tag{5.1}$$

after Poynton [Poy98].

The luminance, i.e., the Y-value, is then given as $\frac{cd}{m^2}$, assuming the pixel intensities are calibrated.

Following the equation of adaptation after Krawczyk et al. [KMS05], using the luminance value L_{i-1} from the previous frame, the temporally filtered luminance L_i for the current frame i can be calculated as:

$$L_i = L_{i-1} + (Y - L_{i-1}) \cdot \left(1 - e^{-\frac{ft}{\tau(Y)}}\right) \tag{5.2}$$

$$\tau(Y) = \sigma(Y) \cdot \tau_{rod} + (1 - \sigma(Y)) \cdot \tau_{cone} \tag{5.3}$$

$$\sigma(Y) = \frac{0.04}{0.04 + Y} \ , \tag{5.4}$$

where ft is the current frametime in seconds, and $\tau_{rod} = 0.4$s and $\tau_{cone} = 0.1$s are the adaptation times for the rods and cones respectively. The function $\sigma(Y)$ denotes the sensitivity of rods with regard to the luminance. This adaptation luminance L_i can then be applied to the image by multiplying it with the linear RGB values.

This approach leads to a temporal adaptation to the luminance of the viewing direction over time (see Figures 5.1 and 5.2).

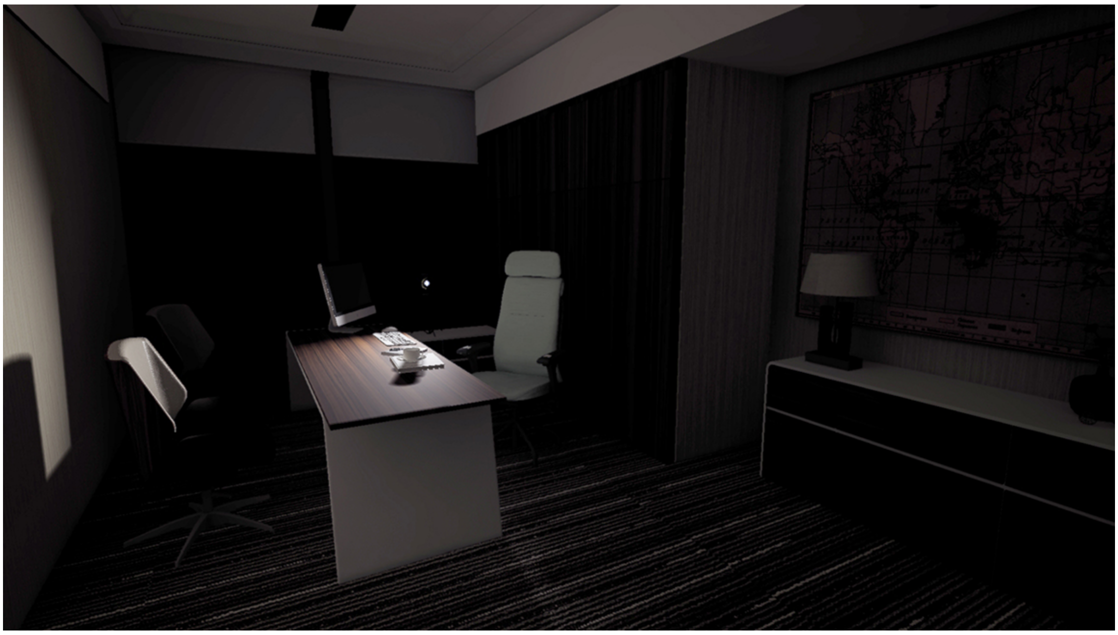


Figure 5.1: Initial scene adaption.

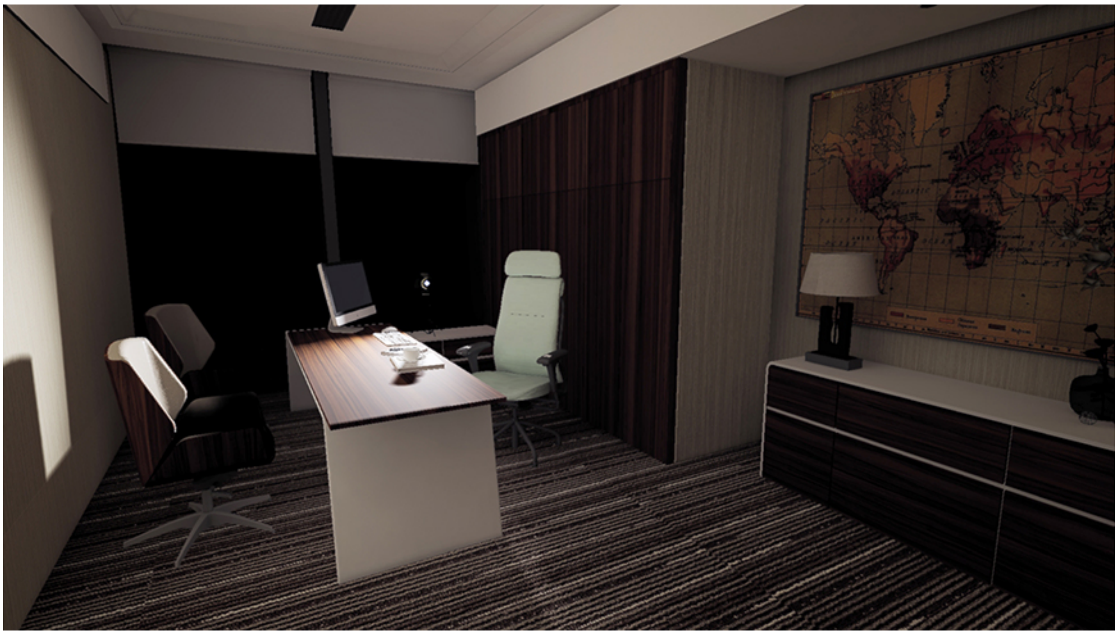


Figure 5.2: Scene adaption after 5 seconds.

5.2 Perceptual Glare

This section will explain the generation and the rendering of a glare, using the principles of wave optics and scattering. First, the generation of a PSF is discussed. Second, the mathematical basis of a glare kernel is explained. Next, the application of a glare kernel onto a rendered image is analyzed, and finally, the different settings, i.e., personal influences of a user, on the glare perception, are laid out.

Light gets scattered in the human eye; therefore, we perceive a glare around very bright lights [RIF^{+}09]. Light-rays entering the eye, scatter at different obstacles inside the cornea and lens, i.e., micrometer-sized particles. Therefore, some rays are redirected and hit the retina at a different spot than when going straight. This scattering creates a halo in our perception in the vicinity of bright light sources (see Figure 5.3). This can be modeled using a PSF.

5.2.1 The Point Spread Function

Our glare effect is based on the glare proposed by Ritschel et al. [RIF^{+}09]. The performance of their approach was not sufficient enough to run in real time in VR or AR at the necessary minimum of 90 FPS.[1] We adapt this approach for a VR application, and make it view-dependent by using eye tracking. To achieve 90 FPS runtimes, we need to reduce processing time. Hence, we cut down expensive computations and only keep

[1] https://www.vive.com/de/product/vive-pro/

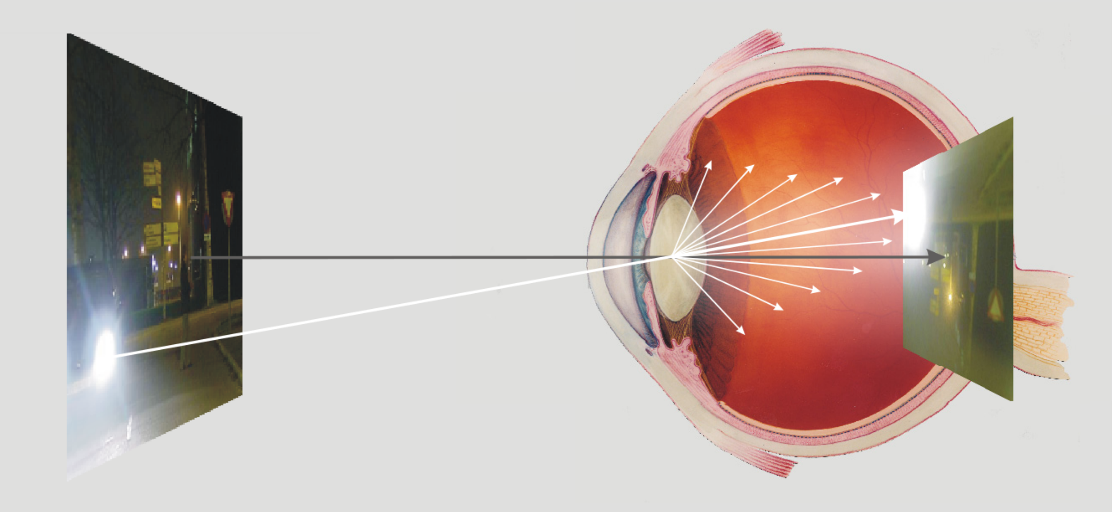


Figure 5.3: The creation of glare patterns in the eye. Reprinted from [FC07].

the minimum requirements for an accurate output result. We consulted an optometrist and identified the parts of the algorithm that have the most influence on the final effect, according to his expertise, and adapted the approach accordingly, as shown in Table 5.1.

Table 5.1: Influences on glare perception and whether they were included in our algorithm. The anterior chamber was not included as influence for our glare simulation, because it induces no scattering for healthy eyes. The Lens Fibers and Vitreous Body were not included due to their low influence on the scattering of light in the eye. For medical background see Section 2.1.

Effect	Included
Cornea Particles	Yes
Anterior Chamber	No (no scattering)
Varying Pupil Size	Yes
Pupil Aperture	Yes
Lens Particles	Yes
Lens Fibers	No (low scattering)
Vitreous Body	No (low scattering)

We generate a glare kernel based on the anatomy of the human eye and the scattering of light inside the eye, using medical research from studies of healthy eyes [vdBHC05, FC07, IG09, RdBT10, SMV$^+$16]. Particles inside the lens and cornea are simulated by generating a user-defined amount at random positions and projecting them onto a plane (see Section 5.2.1 i.). The aperture of the eye lens is included as well, as shown in Figure 5.4. The resulting image is then converted to the spectral domain, following the principles of wave optics. Using this method, a spectral PSF is obtained, which can then be used as a glare kernel (see Figure 5.5e).

5.2.1 i. Particles

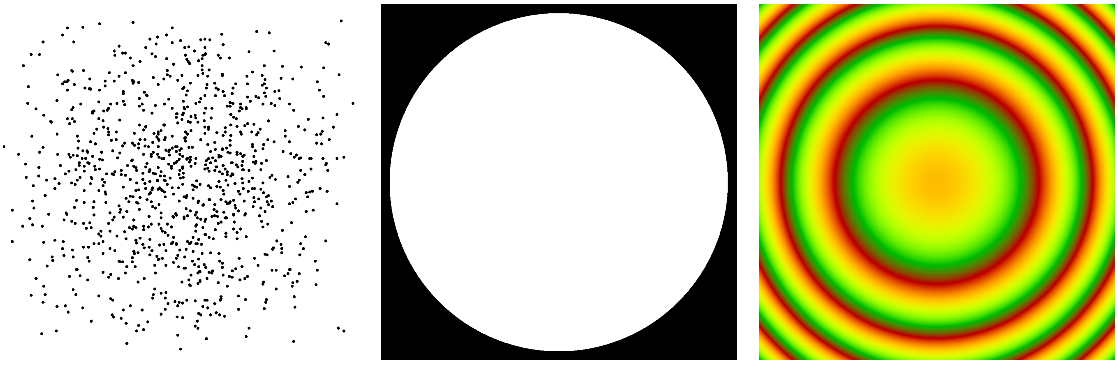
We focus our proposed model on the particles in the cornea and the lens of the eye, as those are two of the main influences on glare perception. In total, they account for around 70% of the glare phenomenon in the eye [RIF^{+}09].

When a ray of light hits a particle in the eye, it scatters and gets bent along a reflection and a refraction angle. According to Ritschel et al. [RIF^{+}09], the backward scattering of light, i.e., the reflection, is “relatively unimportant” concerning glare. Therefore, we also neglect this property. When the refracted ray finally hits the retina, it has less intensity (due to the lost part of reflection) and is not aligned with the original path anymore. We perceive this as a faint glow around the light source.

In practice, we simulate static, circular particles on a 2D plane, which is a projection of the 3D-eye-space onto the pupil plane (see Figure 5.4a). Ritschel et al. [RIF^{+}09] stated that a multi-plane approach (simulating particles at different slices of the eye and adding them up in frequency space) has no obvious advantage over a single plane approach. Therefore, we also use the single-plane approach for our method.

5.2.1 ii. The Pupil

The pupil of the eye resembles a camera aperture – it is a small opening in front of a sensor that only lets in a certain amount of light [Pal99]. Different sizes of aperture allow for different lighting conditions, i.e., bigger pupils in a dark environment let more light into the eye. However, the more light enters the eye, the more straylight (see Sections 2.1.1, 2.1.4, and 2.2.3) occurs, and we perceive glare as more intense at night time [VDB86]. Therefore, glare perception is highly dependent on the size of the pupil.



(a) Simulated particles.

(b) Simulated aperture.

(c) Separate Fresnel term remapped from [-1;1] to [0;1] for displaying.

Figure 5.4: Different components of the PSF.

The adaptation of the size of a human pupil to light and dark is done automatically, involuntarily, and is continuously changing. In order to simulate glare realistically, this adaptation has to be taken into account. In contrast to Ritschel et al. [RIF$^+$09], who calculate an artificial pupil size via time and field luminance, based on measured data, we use our eye tracker. We can measure the user's current pupil size and react to changes in real-time. We do not need to estimate the pupil size – we are able to track the actual pupillary response of the user. Therefore, we are able to simulate the glare pattern in a more realistic and natural way.

Similar to the particles (see Section 5.2.1 i.), the pupil aperture is simulated in 2D space, as a simple circular opening (see Figure 5.4b). We calibrate our eye tracker to a user's pupil size, by showing a black or white screen for 5 seconds each. We take the average of the pupil size during the white frame as minimum pupil size and the average over the black frame as maximum. During runtime, we then generate the pupil aperture by mapping the eye tracker output to our predefined range of 2–8 mm diameter. Finally, we combine it with the simulated particles.

5.2.1 iii. Non-Included Influences

According to the optometrist we consulted, the influence of the lens fibers (see Section 2.1.4) is negligible, as these only have a low influence on light scattering. Furthermore, we chose not to include moving particles as these are mainly located in the vitreous humor, which hardly affects the human perception of glare.

5.2.1 iv. Wave Optics

In order to transform the simulated particles and the aperture (i.e., the aperture mask) into an actual spectral PSF, we use an approach based on wave optics, as proposed by Kakimoto et al. [KMN$^+$04]. Wave optics aims to describe the interference and the diffraction of light in a physical way. The authors base their approach on the diffraction of light on obstacles, such as the pupil aperture, to simulate the incident light in the human eyes. They show that the Fraunhofer diffraction equation can be transformed into an equation with a Fourier transform for the integral, which can be solved in real-time with a 2D FFT:

$$I_f\left(x_f, y_f\right) = \frac{A^2}{\lambda^2 f^2} \left|\mathcal{F}\left[t_0\left(x_0, y_0\right) P\left(x_0, y_0\right), \lambda f\right]\right|^2, \tag{5.5}$$

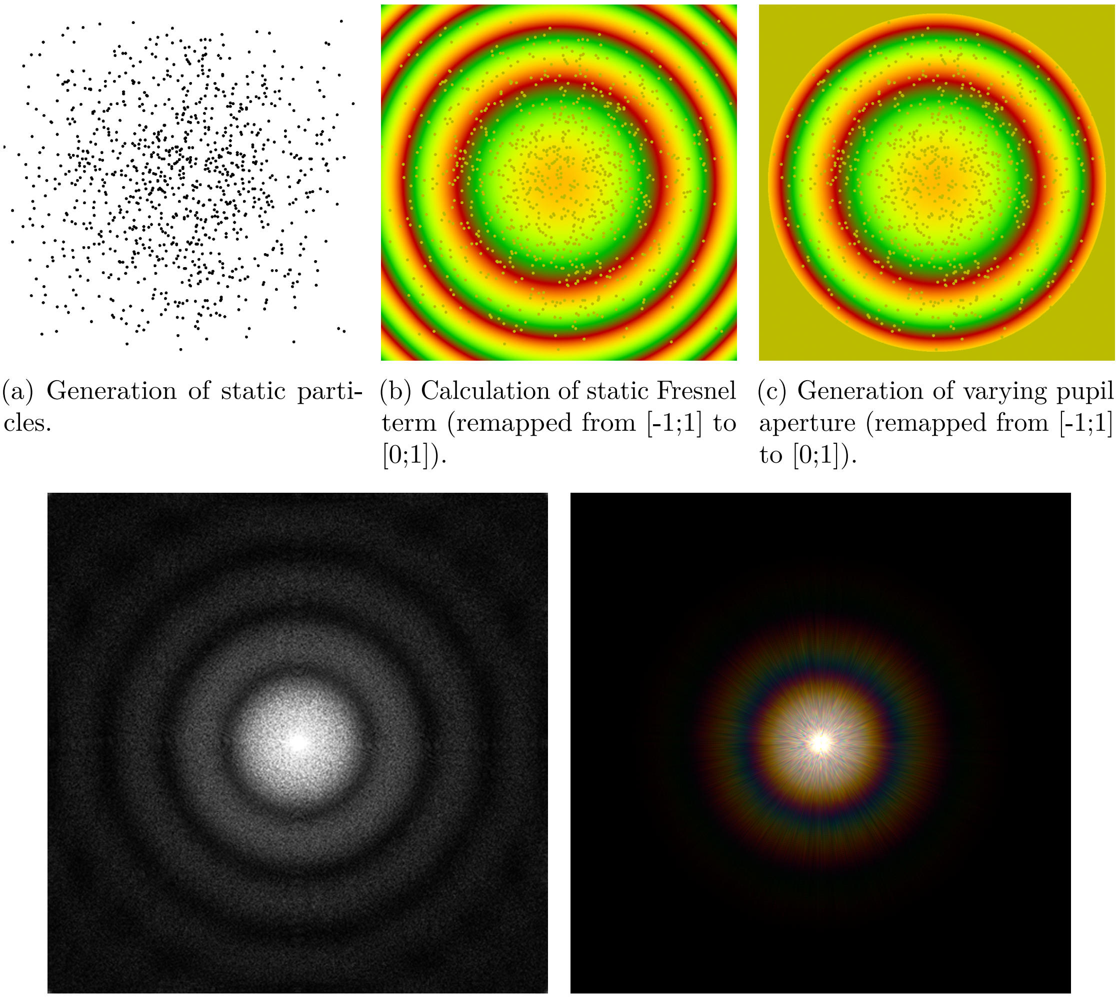
where I_f is the resulting PSF at coordinates (x_f, y_f), A is the amplitude of the wave, λ is the wavelength, f is the focal distance and $t_0(x_0, y_0)$ is an image masked by the pupil function $P(x_0, y_0)$.

This approach can be enhanced by adding a Fresnel term (see Figure 5.4c), according to Ritschel et al. [RIF$^+$09], which is more accurate than Fraunhofer diffraction, due to the short distance between the pupil and the retina.

5.2.2 Glare Kernel Generation

The generation of the glare kernel contains the following steps:

1. Generation of static particles, Figure 5.5a
2. Calculation of static Fresnel term, Figure 5.5b
3. Generation of varying pupil aperture, Figure 5.5c
4. Fourier transformation to monochromatic PSF, Figure 5.5d
5. Conversion to spectral PSF, Figure 5.5e



(a) Generation of static particles.

(b) Calculation of static Fresnel term (remapped from [-1;1] to [0;1]).

(c) Generation of varying pupil aperture (remapped from [-1;1] to [0;1]).

(d) Fourier transformation to monochromatic PSF.

(e) Conversion to spectral PSF.

Figure 5.5: Steps of the glare kernel generation.

All following equations are evaluated in the normalized device coordinate space, where $x = [-1;1]$ and $y = [-1;1]$. The subscript $\mathbb{C}$ denotes complex numbers.

In a pre-processing step the two-dimensional static particles $P(x, y)$ are generated as a binary random dot pattern with constant particle/dot size s.

The static Fresnel term, following Huygen's principle after Ritschel et al. [RIF$^+$09], is defined as

$$F_{\mathbb{C}}(x,y) = e^{\mathrm{i}\frac{\pi}{\lambda d}\left(x^2+y^2\right)}, \tag{5.6}$$

where λ is the wavelength and d the distance from the pupil to the retina. In order to avoid having to calculate an FFT for every visible wavelength, the mean value of the human spectrum is used, specifically $\lambda = 575$ nm. Furthermore, $d = 24$ mm is used as the human focal length [Hec02].

During runtime, the pupil aperture $A(x, y)$ is generated, depending on the currently measured pupil size of the user, as binary mask.

Finally, these three functions are multiplied, which results in the complete lens equation for simulating glare effects:

$$L_{\mathbb{C}}(x,y) = P(x,y) \cdot F_{\mathbb{C}}(x,y) \cdot A(x,y). \tag{5.7}$$

We then transform this equation into frequency space to generate the monochromatic PSF. The lens equation has to be mapped to a positive range,[2] but adding a constant before the Fourier transform $\mathcal{F}$ does not change the final result, because as shown by Wang [Wan09]:

$$\mathcal{F}[ax(t) + by(t)] = a\mathcal{F}[x(t)] + b\mathcal{F}[y(t)] \tag{5.8}$$

and Weisstein [Wei20]:

$$\mathcal{F}[1](\mathbf{x}) = \delta(\mathbf{x}), \tag{5.9}$$

where $\delta(\mathbf{x})$ is Dirac's delta function, which is 1 at $\mathbf{x} = (0, 0)$ and 0 everywhere else.

Since in our case the pixel at position $(0, 0)$ will be overwritten by a brightness multiplier for implementational reasons (see Section 6.3.2), we can conclude:

$$T_{\mathbb{C}}(x,y) = \mathcal{F}[L_{\mathbb{C}}(x,y) + 1] = \mathcal{F}[L_{\mathbb{C}}(x,y)] + \mathcal{F}[1] \approx \mathcal{F}[L_{\mathbb{C}}(x,y)], \tag{5.10}$$

[2] Due to implementational reasons, for more details, see Section 6.3.1.

where $T_{\mathbb{C}}(x, y)$ is the lens aperture in frequency domain.

Moreover, we can define the monochromatic PSF $M(x, y)$ after Ritschel et al. [RIF+09] as:

$$M(x, y) = \frac{1}{(\lambda d)^2} \left| \frac{1}{N} \cdot T_{\mathbb{C}}(x, y) \right|^2 , \tag{5.11}$$

where N is the size of the input. The squared length of the complex input $T_{\mathbb{C}}$ is normalized by the image's size and by the wavelength λ and pupil length d. The result is a single channel 2D image, representing the scattered, incoming light on the retina of a single wavelength.

Finally, the spectral PSF is generated via summing up $M(x, y)$ at different wavelengths φ_i in the visible human range:

$$S_{\text{XYZ}}(x, y) = \frac{1}{k+1} \cdot \sum_{i=0}^{k} \text{Wavelength2XYZ}\left(\varphi_i\right) \cdot M\left(x\frac{\lambda}{\varphi_i}, y\frac{\lambda}{\varphi_i}\right), \tag{5.12}$$

$$\varphi_i = 380 + \frac{i}{k}(770 - 380), \tag{5.13}$$

where k is a constant that defines the "fineness" of the interpolation. We multiply the intensity of the incoming light at $M\left(x\frac{\lambda}{\varphi_i}, y\frac{\lambda}{\varphi_i}\right)$ with the XYZ color of the respective wavelength after Wyman et al. [WSS13]. By using XYZ colors, we get a perceptually based average for each pixel.

Finally, this is transformed into linear RGB space, which results in the spectral RGB kernel $S_{\text{RGB}}(x, y)$, which can then be used as the glare kernel.

5.2.3 Convolutional Bloom

The glare kernel, as described in Section 5.2.2, is applied to the image, using a convolutional FFT bloom, where both image and kernel are transformed to the frequency domain and multiplied. The result is then transformed back into linear RGB image space. This approach does not only blend the rendered image smoothly with our glare kernel, but also results in different glare intensities, depending on the intensity of the light source.

Since we need two FFTs per eye (forward and inverse) and another one for the kernel, this results in five 2D FFT transformations per frame (in addition to the first transformation of the aperture into the kernel), which is very costly and not well suited for real-time VR or AR applications.

During a consultation with an optometrist, we found out that the glare intensity is dependent on the current viewing direction. Therefore, we apply our bloom effect according to the viewing direction in a smaller window and blend it with a falloff with the original image. In contrast to Ritschel et al. [RIF$^+$09], who apply the glare effect uniformly over the whole image, we achieve a less pronounced glare effect at the periphery and a more intense effect in the center. Therefore, we are able to approximate human perception more accurately, because the perceived glare is greatly influenced by our viewing direction [FA53]. The glare phenomenon is more intense when looking directly at a light source. We use a 1024 × 1024 pixel window, which additionally has the benefit that we can save computing time. It leads to faster runtimes due to the smaller size of the necessary FFTs.

Since our eyes never stay still, we also do not see a glaring light as a purely static phenomenon. The perceived movement of a glare mainly comes from the constant unconscious movement of the eye, so-called microsaccades. These saccades are approximately 10ms long. Our eye tracker can capture images at 200Hz,[3] which means an image every 5ms. Therefore, we can capture those short saccades, and they are included in our simulation just like normal eye movement.

5.2.4 Settings

The resulting glare effect can be adjusted by manually changing various parameters in order to achieve a more realistic output for individual users, as each person has a different perception. Generally, default values can approximate the average user very well, but tweaking the following parameters allows for more personalized effects.

Pupil size The size of the pupil is individually calibrated for each user. We measure the average diameter of the pupil when looking at a black and at a white screen for approximately five seconds (see Section 5.2.1 ii.). A bigger pupil admits more light into the eye than a smaller one. Therefore, the resulting glare is more intense. For comparisons of our glare effects for different pupil sizes, see Figure 5.6.

Number of Particles The number of particles in the human eye, i.e., in the lens or cornea, varies from person to person. Generally, the number of particles increases with age, which leads to a more pronounced glare effect for older people [vdBHC05].

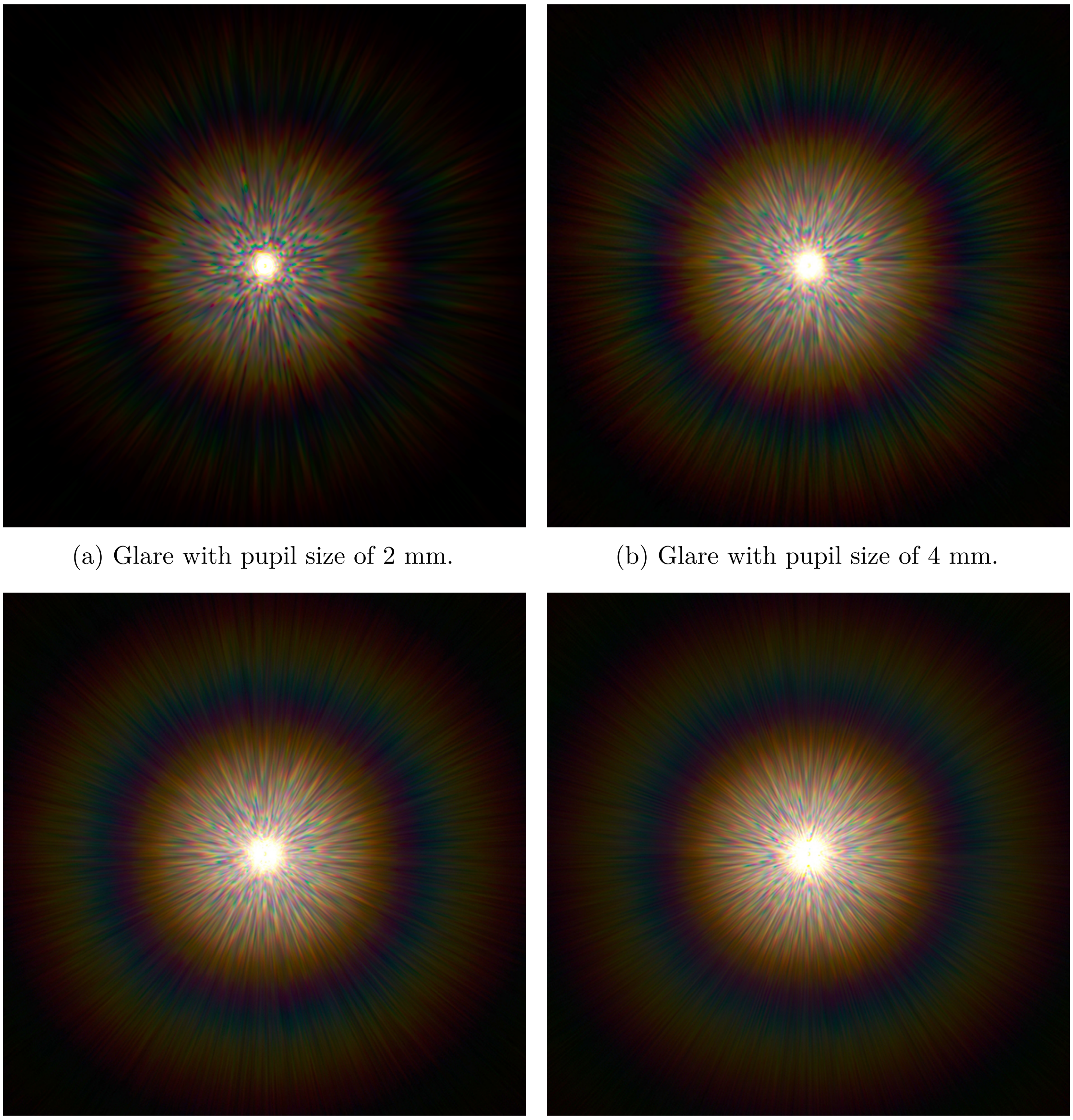
In the simulations of Berg et al. [vdBHC05], 1000 particles were used. This number also leads to very suitable results in our implementation. Therefore, we used this value as the default. Different particle numbers are compared in Figure 5.7.

Particle Size The size of different particles also varies for different users. We explored the parameter space through empirical experiments and found 0.042 mm to be a feasible value for the particle size.

However, Berg et al. [vdBHC05] state that the median particle size in the human eye is 0.724 μm. This value would lead to particles below one pixel with our current target resolution. In future work, with further advancement of technology

[3] `https://pupil-labs.com/products/vr-ar/`

and improved methodology it should become possible to simulate even such small impurities in real time.



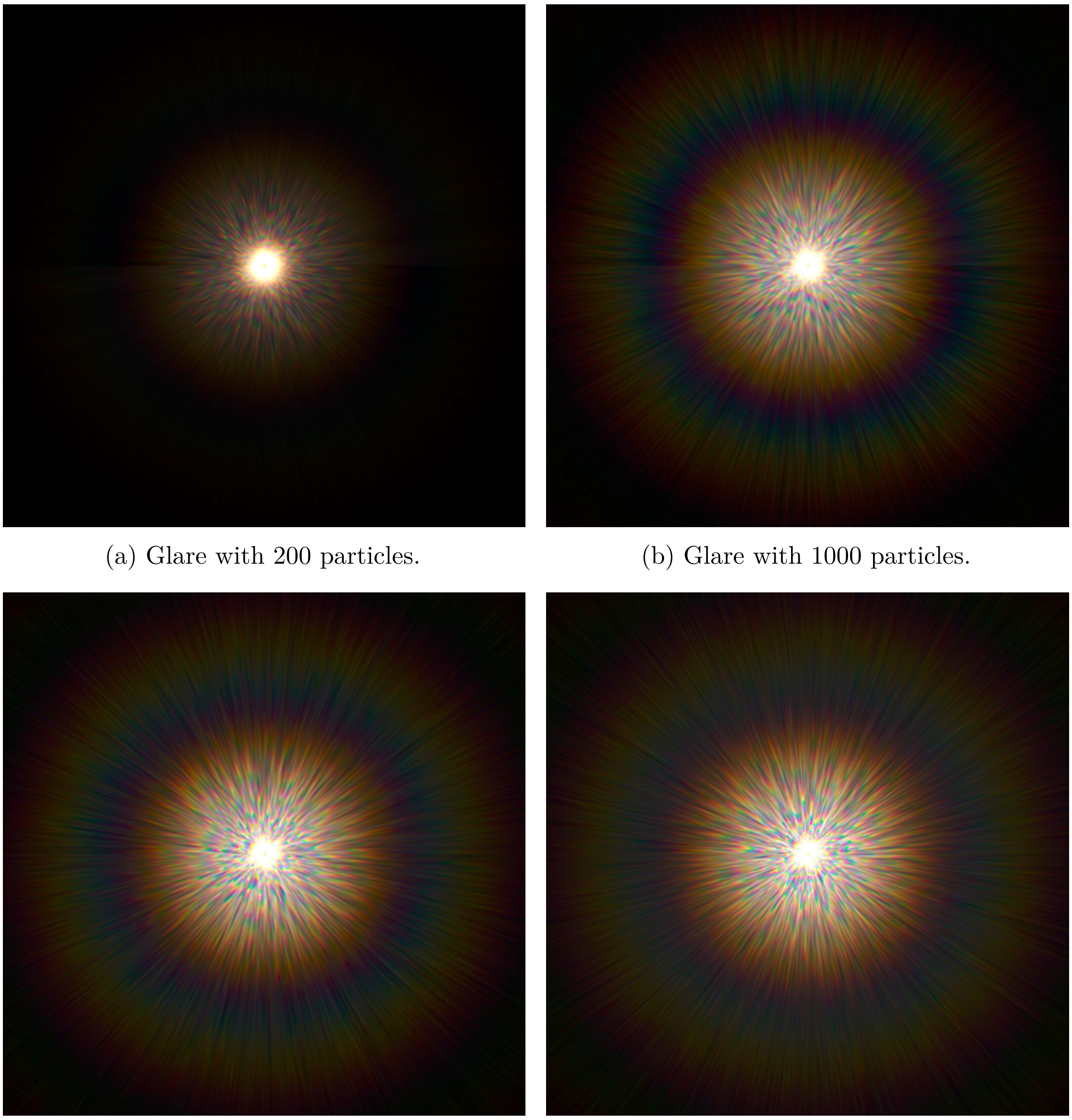
(a) Glare with pupil size of 2 mm.

(b) Glare with pupil size of 4 mm.

(c) Glare with pupil size of 6 mm.

(d) Glare with pupil size of 8 mm.

Figure 5.6: Different glare sizes of the pupil, with 1000 particles.



(a) Glare with 200 particles.

(b) Glare with 1000 particles.

(c) Glare with 2000 particles.

(d) Glare with 4000 particles.

Figure 5.7: Different particle numbers, with constant pupil size of 5 mm.

5.3 Visual Acuity Reduction

Human visual acuity is highly dependent on the illumination in a scene. With decreasing luminance values, the perceived acuity is reduced as well. To simulate this phenomenon, a simple blur can be applied to a rendered image, depending on the current point of focus. However, we use a more sophisticated method for the whole simulated image – we blur the brightness values depending on a visual acuity function. Since this leads to different intensities of reduced visual acuity in darker and brighter areas, no knowledge of the viewing direction is necessary.

5.3.1 Loss of Visual Acuity

In our approach, we simulate the loss of visual acuity based on the method proposed by Krawczyk et al. [KMS05]. In a post-processing step, we convert the linear HDR values (i.e., before tone mapping) of the image to the CIE L*a*b* (Lab) color space, where L is the lightness of the pixel. Secondly, the lightness values are blurred, in horizontal and vertical direction. The intensity of the blur for each pixel aims to approximate the function fit of real foveal acuity data, as proposed by Larson et al. [LRP97]. This approach results in a small error, due to the non-uniform blur kernel, yet this can be neglected, as explained in Section 5.3.3. Converting the processed values back to RGB colors results in an image where dark regions are blurred.

The reduction of visual acuity consists of the following steps:

1. Convert linear HDR RGB values to Lab color space
2. Horizontal pass: blur L (lightness) with 1D Gaussian blur kernel, with sigma dependent on the lightness value
3. Vertical pass: blur L again with 1D Gaussian blur kernel, with sigma dependent on the (new) lightness value
4. Convert Lab colors back to linear RGB color space

5.3.2 New Function Fit

We propose a faster function fit to approximate the loss of visual acuity in the dark. This is relevant due to the high performance necessary for VR applications.

Larson et al. [LRP97] propose the function fit

$$R(Y) = 17.25 \cdot \arctan\left(1.4 \cdot \log_{10}(Y) + 0.35\right) + 25.72, \tag{5.14}$$

where $R(Y)$ the maximum perceivable visual acuity at the given luminance Y in cycles/degree.

We approximate the formula

$$\sigma(L) = \frac{width}{fov \cdot 1.86 \cdot \sqrt{2} \cdot R(\mathrm{LtoY}(L))}, \tag{5.15}$$

from Krawczyk et al. [KMS05], where $\mathrm{LtoY}(L)$ is the conversion from L in Lab to Y in XYZ color space, *width* the width of the rendered image and *fov* the horizontal FOV in degrees. The resulting σ denotes the variance for the 2D Gaussian blur.

We use

$$\sigma(L) = \max(1 - L, 0), \tag{5.16}$$

to approximate the blur intensity based on the approach by Krawczyk et al. [KMS05], as shown in Figure 5.8.

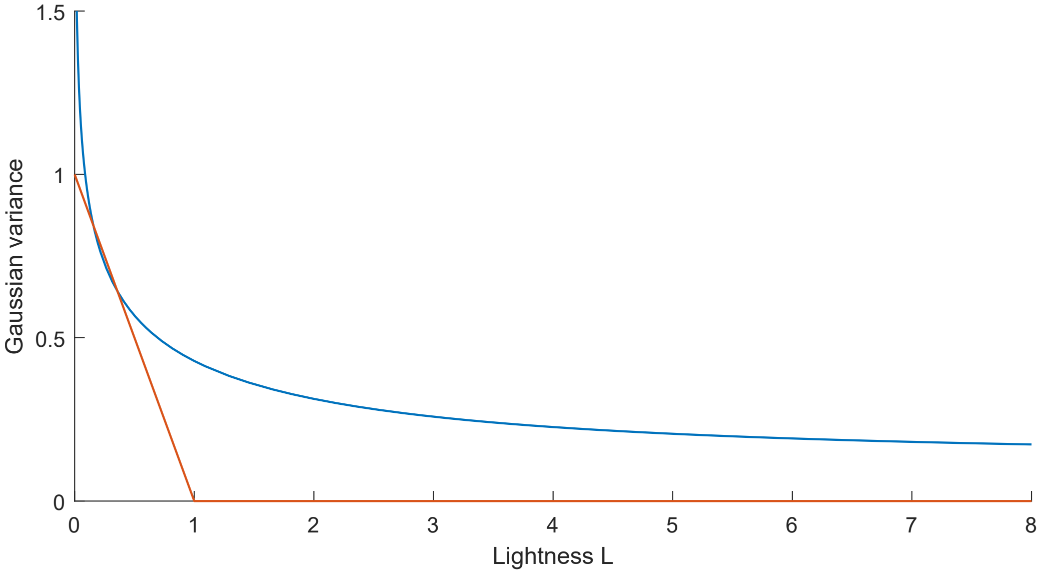


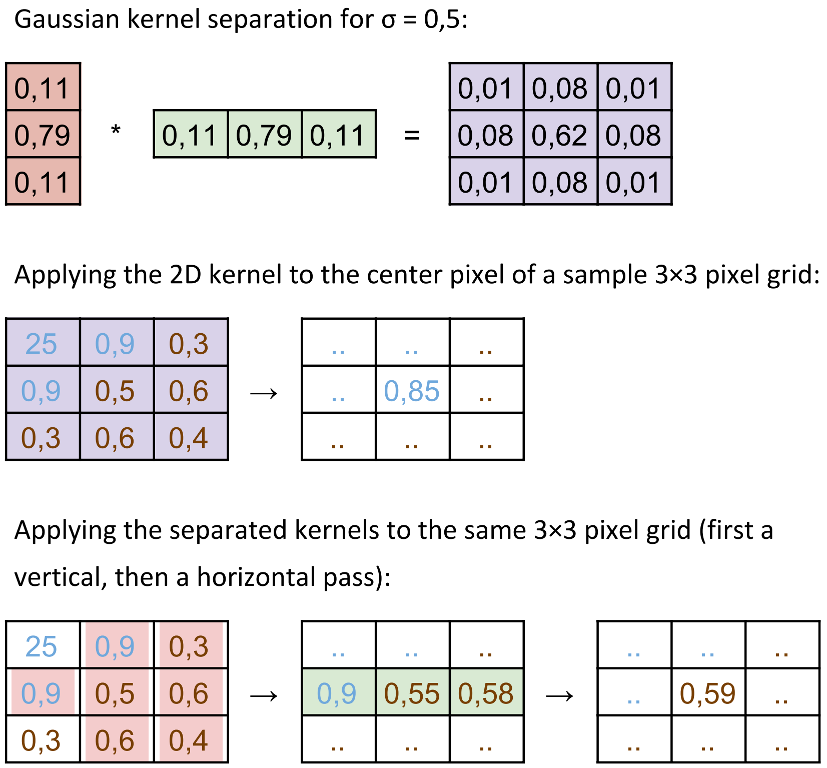
Figure 5.8: Comparison of Gaussian σ size, for different lightness values, of Krawczyk et al. [KMS05] (blue) and our approach (orange).

In bright regions of the image, the visual acuity stays unchanged, therefore, $\sigma = 0$ is reasonable. However, in the dark regions of the image, visual acuity reduction takes place up to a variance of $\sigma = 1$. The benefit of this function is that no blur is applied at lightness values greater than one because our function is 0 for those values. In very bright areas, there is no significant difference in human visual acuity anyhow. Furthermore, the maximum displayable acuity of our VR HMD is $\frac{width}{fov} = \frac{1440}{104} \approx 13.85$ pixel per degree (PPD). By rewriting Equation 5.14 we get

$$L = \mathrm{YtoL}\left(\exp\left(-1.64 \cdot \tan(1.49 - 0.06 \cdot R) - 0.575646\right)\right). \tag{5.17}$$

Setting $R = 13.85$, we can conclude that the highest displayable resolution of our HMD corresponds to approximately 1.31 L. Therefore, our new approximation is able to replicate real-world acuity well, even though we only handle $L \leq 1$. Differences in visual acuity in brighter regions would be too small, to be noticeable anyhow. Our function leads to fewer computations necessary at bright pixels or generally for a bright image.

The maximum of our approximative function $\max(\sigma(L))$ is 1, at $L = 0$. This upper boundary occurs because the Lab system allows no negative lightness values. This is well suited for real-time applications because an upper function boundary leads to an upper limit for a convolution kernel size. Following the common rule [Ngu07] of $s = 2 \cdot \lfloor 3 \cdot \sigma \rfloor + 1$ for the kernel size s, gives a maximum size of 7, which proved to be sufficient in our user study for a realistic simulation of loss of visual acuity in VR.



Gaussian kernel separation for σ = 0,5:

$$\begin{array}{|c|}\hline 0{,}11 \\ \hline 0{,}79 \\ \hline 0{,}11 \\ \hline \end{array} * \begin{array}{|c|c|c|}\hline 0{,}11 & 0{,}79 & 0{,}11 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0{,}01 & 0{,}08 & 0{,}01 \\ \hline 0{,}08 & 0{,}62 & 0{,}08 \\ \hline 0{,}01 & 0{,}08 & 0{,}01 \\ \hline \end{array}$$

Applying the 2D kernel to the center pixel of a sample 3×3 pixel grid:

$$\begin{array}{|c|c|c|}\hline 25 & 0{,}9 & 0{,}3 \\ \hline 0{,}9 & 0{,}5 & 0{,}6 \\ \hline 0{,}3 & 0{,}6 & 0{,}4 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline .. & .. & .. \\ \hline .. & 0{,}85 & .. \\ \hline .. & .. & .. \\ \hline \end{array}$$

Applying the separated kernels to the same 3×3 pixel grid (first a vertical, then a horizontal pass):

$$\begin{array}{|c|c|c|}\hline 25 & 0{,}9 & 0{,}3 \\ \hline 0{,}9 & 0{,}5 & 0{,}6 \\ \hline 0{,}3 & 0{,}6 & 0{,}4 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline .. & .. & .. \\ \hline 0{,}9 & 0{,}55 & 0{,}58 \\ \hline .. & .. & .. \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline .. & .. & .. \\ \hline .. & 0{,}59 & .. \\ \hline .. & .. & .. \\ \hline \end{array}$$

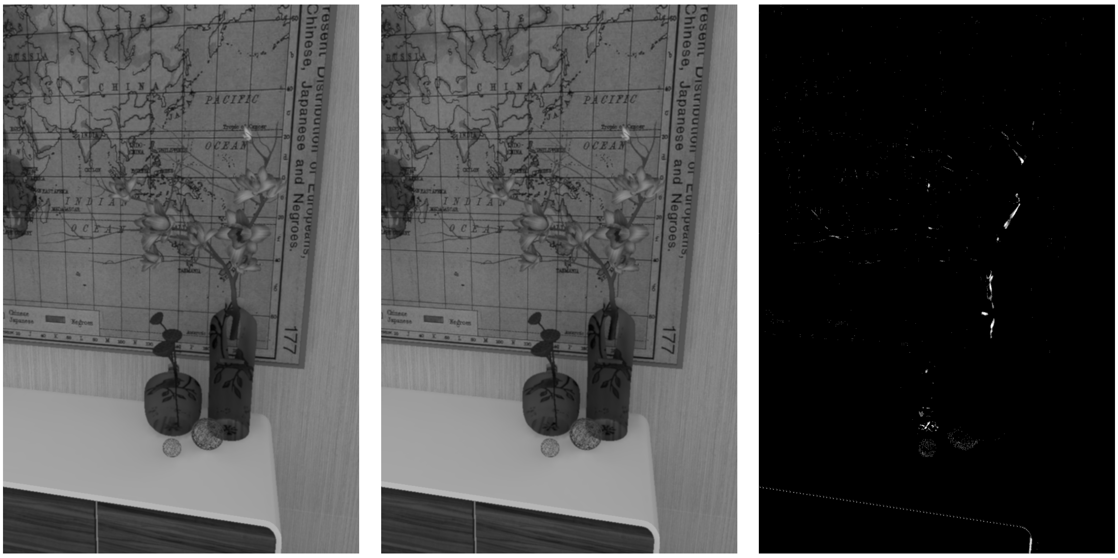
Figure 5.9: Applying a Gaussian filter with varying kernel sizes following our proposed methodology. The differences of a single 2D filter and separating the Gauss kernel into two 1D filters are shown. Usually the errors are very small unless abrupt changes in intensity occur (e.g., near edges) as in this example. Pixel values in light blue highlight bright pixels (with a kernel size $s = 1$), brown font color marks darker pixels ($s = 3$). Purple cells signify the convolution with the 2D filter, red for vertical 1D convolution and green for 1D horizontal convolution.

5.3.3 Applying the Blur

The application of a 2D blur kernel needs s^2 texture lookups for every pixel, where s is the kernel size. Since this is not well suited for real-time applications, we separate the 2D kernel into two 1D kernels and apply them horizontally and vertically, which gives us $2 \cdot s$ texture lookups. Especially for bigger kernels, this is a significant speedup ($7^2 = 49$ lookups, whereas $2 \cdot 7 = 14$ lookups).

Separating a Gaussian blur with a static kernel over the whole frame, does not change the final output, i.e., no errors occur through separation. However, due to the varying size of the blur kernel for every pixel, which we use, errors are induced at the edges between bright/dark changes, as demonstrated by the example shown in Figure 5.9. Furthermore, additional irregularities occur by determining the filter size of the second pass based on the already 1D filtered image. Both of these errors are minimal and hardly recognizable with the human eye, as shown in Figure 5.10.

Using the previously applied FFT to remove frequencies and therefore achieve a blurring is not possible in this case, as we blur each pixel with an individual sigma, depending on its brightness.



(a) 2D filter.

(b) 2 times 1D filter.

(c) Differences.

Figure 5.10: Comparison of the lightness of an image, when (a) applying a 2D visual acuity filter vs. (b) two 1D filters, shown as (c) the absolute difference between (a) and (b).

5.4 Scotopic Color Vision

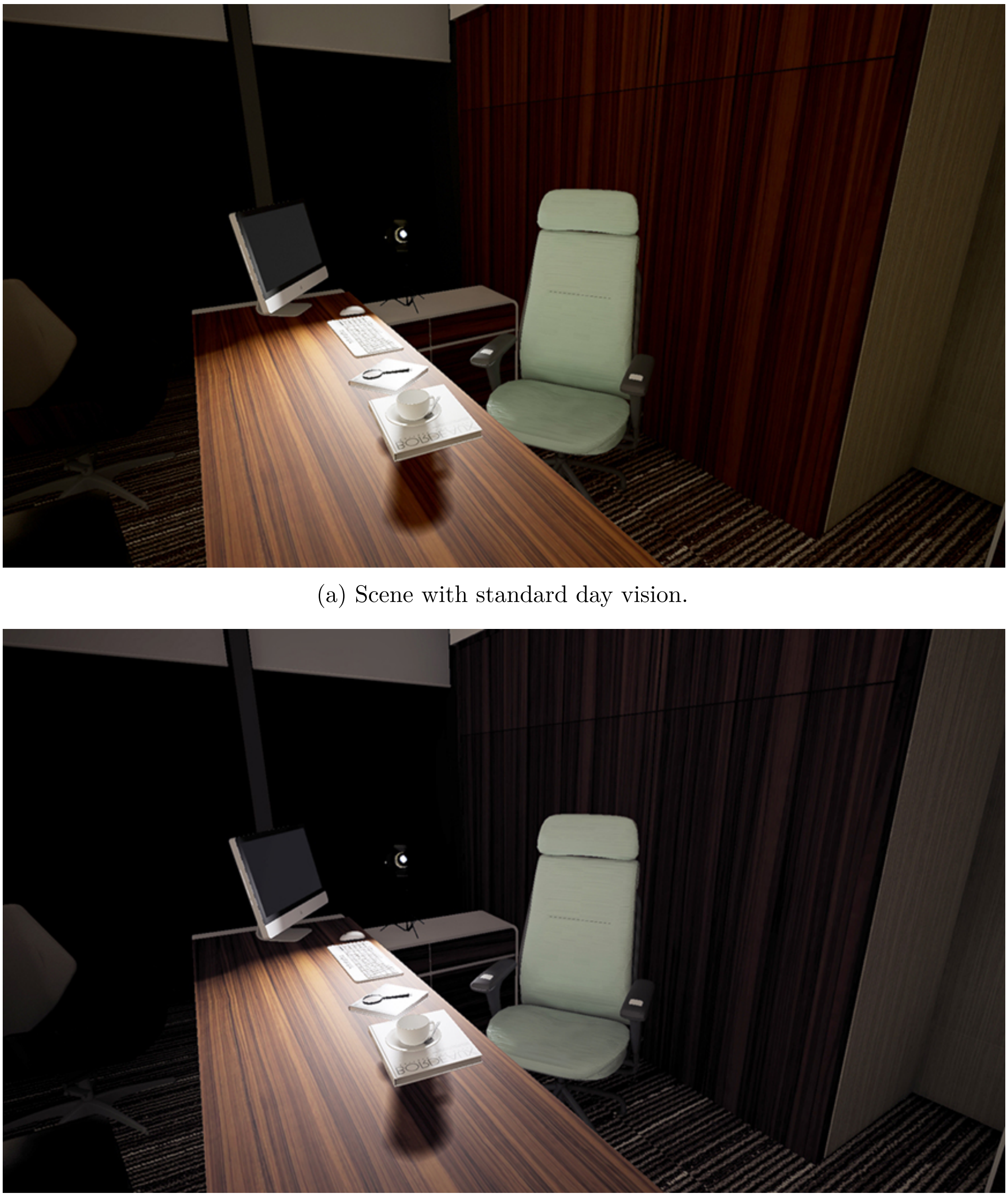
Scotopic color vision describes the color shift we perceive in darker environments (see Section 2.2.2 iii.). We implement scotopic color vision by adding a color shift, depending on the temporally adapted scene luminance calculated beforehand (see Section 5.1). Additionally, the absolute scotopic sensitivity of photorecptors for a given luminance in the scene is taken into account – the more sensitive the rods, i.e., the less sensitive the cones, the more intense the color shift. We calculate this after tone mapping.

If the intensity of incoming light is too low to trigger the (mainly red and green-sensitive) cones, rod-vision is activated. Then mostly rods and blue-sensitive cones are enabled, leading to a color shift, because colors like blue and purple are the most intense, i.e., brightest, in this case. Therefore, we apply a color shift towards a lavender purple color, because of the perceptual change towards light blue to purple at night (see Section 2.2.2 iii.), to simulate a loss of color perception (see Figure 5.11).

In order to do that, we use the original scene luminance (i.e., not tone-mapped) and the sensitivity of rods, given by equation 5.4. Depending on that sensitivity, we add a color shift, after Krawczyk et al. [KMS05], such as

$$SV = RGB \cdot (1 - \sigma(L_i)) + [1.05, 0.97, 1.27] \cdot L_i' \cdot \sigma(L_i), \tag{5.18}$$

where RGB is the tone-mapped LDR color, L_i the temporally filtered luminance of the current frame, L_i' the tone-mapped luminance and $[1.05, 0.97, 1.27]$ the purple tint.



(a) Scene with standard day vision.

(b) Scene with scotopic vision applied.

Figure 5.11: Scene changes in regards to scotopic vision.

CHAPTER 6

Implementation

In this chapter, the technical details of this thesis are explained, i.e., the implementation with the software used. Figure 6.1 shows the basic outline of our implementation. We explain the issues and pitfalls encountered when realizing our proposed methodology, as well as how to overcome them.

Our workflow is separated into two main parts: pre-processing, and run-time processing. In the pre-processing step, we calculate all data, that does not change during our

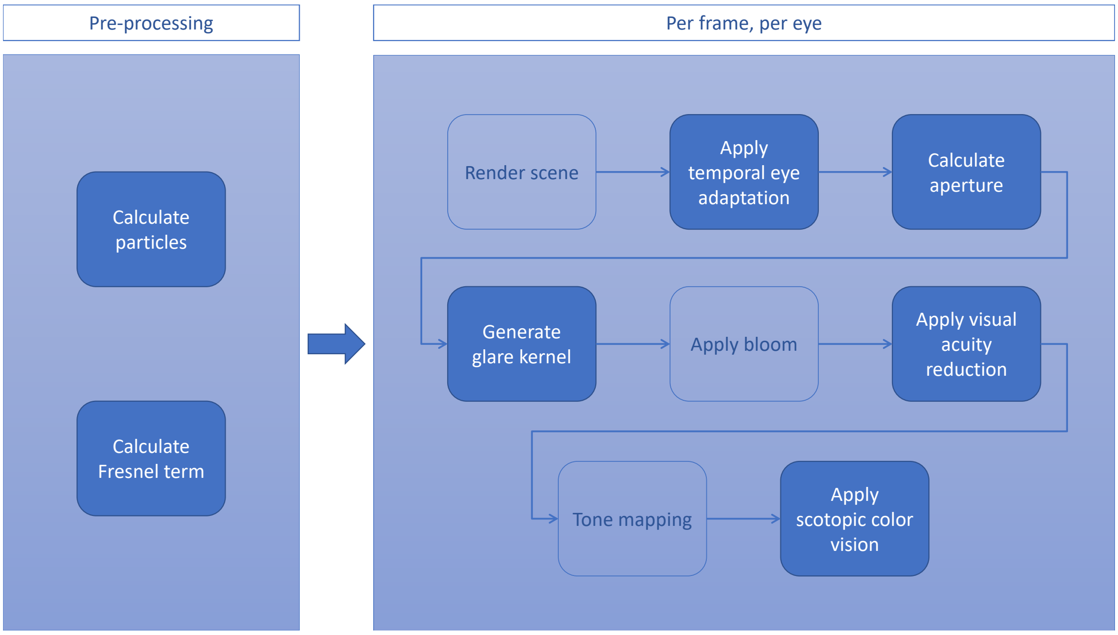


Figure 6.1: Flowchart of the necessary steps for realistic glare and night rendering. Filled fields signify adapted stages and non-filled fields are standard renderer stages.

simulation, such as the size and distribution of particles or the Fresnel term. During runtime, we add multiple stages to a common renderer's post-processing pipeline. All these stages have to be done per eye, i.e., twice per frame. First we apply a temporal eye adaptation to the HDR rendered scene. Additionally, we calculate the lens aperture and generate a glare kernel, that can then be applied as a convolutional bloom. After the glare, a visual acuity reduction is computed for the darker pixels of the image. Finally, after tone mapping, we apply a scotopic color shift to simulate night vision.

6.1 Software

In this section, the software this thesis builds upon is presented.

6.1.1 Renderer

In this project the *Unreal Engine (UE) 4.20*[1] is used as a renderer. Even though this software is mainly intended for game development, it includes numerous rendering features with various adjustment options. One of its main advantages is that UE is fully customizable, as the source code of the engine is available and can be adapted and expanded. Furthermore, it has all the typical benefits of a game engine, such as model loading, a material editor, as well as built-in VR-integration for conventional headsets.

We use UE to process and render a scene in VR and apply our proposed post-process adaptations. We have modified multiple stages of the renderer's post-processing pipeline, such as the convolutional bloom, which was adapted to only render in a specific window depending on the viewing direction. Furthermore, we added temporal eye adaptation by employing medically based formulas and inserted additional steps in between already available post-process stages for custom control.

6.1.2 Eye-Tracking Software

In order to access the features of our eye tracker, we use *Pupil Capture* by *Pupil Labs* [KPB14]. It is an open-source eye-tracking platform that generates a 3D model of the eye, based on the eye movements, to then calculate the viewing direction, the pupil diameter or eye accommodation.

Unfortunately, no official UE plugin for Pupil Capture was available at the time of writing this thesis. Therefore we use a custom Python script integration to access the data provided by our eye-tracker.

6.1.3 AR Module

While the VR integration is already provided by UE, an additional plugin is necessary to render AR scenes. We use *SRWorks SDK 0.8.0.2* by *HTC*,[2] which can simply be

[1] https://www.unrealengine.com/

[2] https://developer.vive.com/resources/knowledgebase/vive-srworks-sdk/

integrated into UE. The tool uses the cameras of the HMD and renders the images on a plane in VR to create a video-see-through AR scene.

The AR module, however, is not compatible with NVIDIA's Touring-series[3] and can only be used with graphic cards from the previous series.

6.2 Temporal Eye Adaptation

To implement a correct simulation of human vision, many of UE's already available features can be utilized and updated. For example, the process of temporal eye adaptation was already implemented in the engine, but based on approximations, not on medical data. UE provides an eye adaptation algorithm with manually adjustable up and down speed. While this gives a game designer more freedom to adapt the settings to an application's needs individually, this does not perfectly reproduce human adaptation. UE's temporal eye adaptation algorithm is implemented in a post-processing shader, with a function that uses the previous adaption value, the target value and the current frame time. With these three parameters, we implemented a medically-based temporal adaptation, following Equations 5.2–5.4. Due to the usage of medically evaluated data of human adaptation to light and dark, the parameters of up and down speed are no longer necessary.

6.3 Perceptual Glare

We generate a medically based glare via simulating the impurities in the human eye and the amount of light entering the eye. We generate static particles in a pre-processing step and combine them with a user's real-time pupil size. This is our lens aperture, which we use to generate our glare kernel via FFT. The glare kernel can then be applied to a rendered image with a convolutional bloom in real-time.

6.3.1 Pre-Processing Step

In our proposed methodology, the Fresnel equation, that is used for the diffraction of waves at short distances (like inside the eye), and the particles are static. Therefore, they can be computed and multiplied in a pre-processing step at startup and stored in a single texture for access during runtime. Since the Fresnel term is a complex number, the result is stored in a two-channel texture, where the red channel is the real part and the green channel the imaginary part.

UE is only able to store positive values in a render target (all negative values are clamped to zero), but the Fresnel term is in the range of $[-1, 1]$. Therefore, we shift the output into the positive domain, i.e., by adding the minimum of that range to the result.

[3] https://forum.vive.com/topic/5747-list-of-supported-gpus/?tab=comments#comment-25927

This yields a 2D-RG-Texture, representing the result of $F_{\mathbb{C}}(x,y) \cdot A(x,y)$ (see Section 5.2.2), with values in the range of $[0,2]$, which can be accessed during runtime.

6.3.2 Real-Time Kernel Generation

The non-static influence of the glare kernel, i.e., the pupil size, is generated during runtime. The size of the aperture is based on the user's current pupil size, which has to be calibrated beforehand.

The pre-computed texture $F_{\mathbb{C}}(x,y) \cdot A(x,y)$ is then shifted back into its actual domain $[-1,1]$, multiplied with the pupil aperture, and shifted into the positive domain again for storing in another render target. The resulting lens texture, $L_{\mathbb{C}}(x,y)$ (see Sec. 5.2.2), is then used as input for the Fourier transform. Note that the shifting of the values has no significant influence on the output (see Section 5.2.2).

As a next step, the FFT of this lens texture is computed by calling UE's internal GPU FFT function and copying the output into another render target for further processing. This function saves the output in a side-by-side manner, where $[0, \frac{N}{2}-1]$ is the complex output for the previous red channel and $[\frac{N}{2}, N-1]$ for the green channel as shown in Figure 6.2. By calculating the "mirrored" complex conjugate for each channel, the complete Fourier transformed output can be constructed.

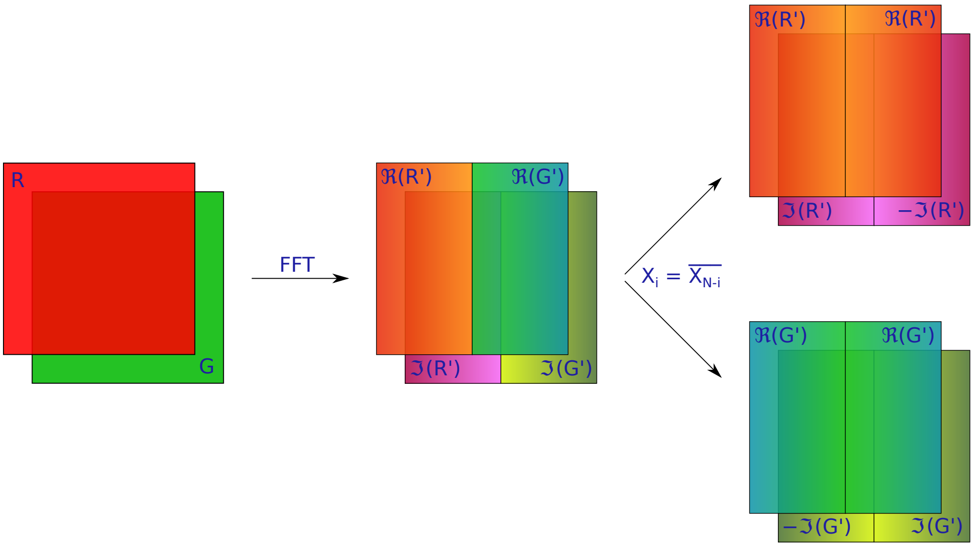


Figure 6.2: The UE FFT process visualized for a two-channel texture: The 2D Fourier transformation of an input image RG generates a complex side-by-side output, where the real values $\Re$ of R' and G' are saved in the first channel and their imaginary components $\Im$ in the second channel. They can be separated by taking the complex conjugate X_{N-i} of each row X_i, i.e., mirroring one side along the middle and inverting the mirrored imaginary component.

This result is then normalized. Furthermore, by taking the absolute squared value, a single-value, monochromatic PSF is generated. Before storing in a render target, it has to be clamped to UE's maximum texture value to avoid overflow.

This monochromatic PSF is then used to generate the spectral PSF by iterating over multiple wavelengths and summing them up, as shown in Figure 6.3.

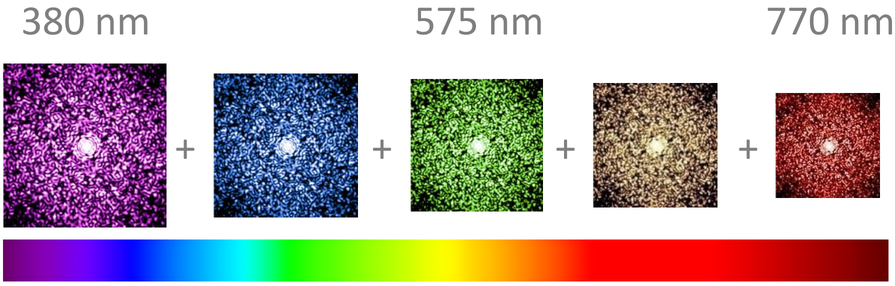


Figure 6.3: Scaled copies of one monochromatic PSF can be combined (added up) to a spectral PSF. Reprinted from Myszkowski and Tursun [MT19].

Finally, in order to be able to use our kernel as input for UE's bloom function, we manually add a very bright center, as advised in UE's documentation.[4] While the PSF generally already has a very bright center, it mainly is too big (i.e., multiple pixels wide) and would blur the rendered image on convolution. Therefore, we add a smaller, brighter center to the kernel before convolution. In future work the correlation of a too large center, as generated by our methodology, to the blurring of the image should be evaluated further.

6.3.3 Unreal Bloom

UE's built-in convolutional bloom can be used to combine the glare kernel with the rendered image. Since it is very costly to transform the whole rendered image for both eyes into the frequency space due to the high pixel count, we adapted the engine to apply the FFT only to a specific window. A square, power-of-two-sized area around the current viewing point, is selected, and the bloom is only applied to this part of the image. This approach results in better runtimes, and when using the eye tracker, we can simulate the dependence on the angle of incident light by creating a falloff from the center for more realistic simulation.

In order to allow a more pleasant appearance than cut-off edges at the corners of the FFT window, a custom post-processing stage was added, that blends the original (unconvolved) image with the bloomed window using a Gaussian falloff. These edges are only visible at the periphery of our visual field, which has very low influence on our bloom perception in VR. However, in the periphery we mainly react to movements or high frequency noise. Therefore, a blending is necessary to avoid disturbing artifacts.

[4] https://docs.unrealengine.com/en-US/Engine/Rendering/PostProcessEffects/Bloom/index.html#kernelimagebestpractices

Furthermore, the native UE bloom expects a static glare kernel (most likely to avoid costly recomputation every frame). However, we want a dynamic kernel. Therefore, we adapted the engine to take dynamic render targets as input, and trigger the recomputation of the kernel FFT manually every frame.

6.4 Visual Acuity Reduction

The loss of visual acuity was implemented using two sequential post-process materials before tone mapping. The first one transforms the image into the Lab color space and blurs the lightness in a horizontal direction, while the second one then blurs the lightness in a vertical direction and converts all values back to RGB colors.

Accessing the scene texture at multiple coordinates in UE's post-processing materials is intricate, but possible by using the post-processing shader's internal function *SceneTextureLookup* in a *Custom Node* within the material.

When writing negative values from a post-process material to the scene texture, these numbers get clamped to zero. Since the Lab color space can also have negative values, we need to preserve these values. In this case we cannot simply shift our values by adding the minimum value, because we cannot determine this value without iterating over the whole image, which is very slow and not well suited for VR. Therefore, we add a mask via the alpha channel, using a binary format for the three channels converted into decimal numbers, as shown in Algorithm 6.1. Then, we can pass all values as absolute and unmask it before further processing, as in Algorithm 6.2. Additionally, UE only can pass values between 0 and 1 in the alpha channel. Therefore, a division by the maximum mask value (i.e., seven) is necessary.

Algorithm 6.1: Mask Negative Values

```
Input: A vector $(L, a, b)$, with possible negative values.
Output: A vector $(L^+, a^+, b^+, M)$, with positive values only and negative values
        mask $M$.
$M = 0$;
if $L < 0$ then
    $M+ = 1$;
end
if $a < 0$ then
    $M+ = 2$;
end
if $b < 0$ then
    $M+ = 4$;
end
return $(|L|, |a|, |b|, \frac{M}{7})$;
```

Algorithm 6.2: Unmask Negative Values

```
Input: A vector $(L^+, a^+, b^+, M)$, with positive values only and negative values
       mask $M$.
Output: A vector $(L, a, b)$, with possible negative values.
$M = 7 \cdot M$;
$(L, a, b) = (L^+, a^+, b^+)$;
if $M\%2 == 1$ then
    $L = -1 \cdot L$;
end
$M = \lfloor M/2 \rfloor$;
if $M\%2 == 1$ then
    $a = -1 \cdot a$;
end
$M = \lfloor M/2 \rfloor$;
if $M\%2 == 1$ then
    $b = -1 \cdot b$;
end
return $(L, a, b)$;
```

6.5 Scotopic Color Vision

A scotopic color shift is applied as a post-processing material after tone mapping. The intensity of this color shift is dependent on the sensitivity of the rods. Therefore, we need the temporally filtered scene luminance (i.e., before the tone mapping).

In UE's post-processing materials only the tone-mapped color is available by default after tone mapping. However, the engine still stores the non-tone-mapped output for debugging reasons. Therefore, the non-tone-mapped color and luminance input can be activated and accessed alongside the default inputs. This allows us to implement scotopic vision as UE material, following Equation 5.18.

CHAPTER 7

Results and Discussion

In this chapter, the outcome of this thesis is discussed. First, we will describe the hardware used for the project and our user study. Secondly, all details of the user study and its results are laid out. Then we discuss the limitations of this thesis. Moreover, we compare our work to similar approaches and discuss its advantages and disadvantages. Finally, the runtimes of our algorithm are listed.

7.1 Hardware

We used the *HTC Vive Pro*[1] for our project, because it currently has one of the best pixel per inch/PPD ratios, and is also suited for AR applications. It has two 1440×1600 pixel AMOLED displays with a 90Hz refresh rate. Furthermore, two cameras are located at the front of the HMD that allow for video-see-through AR. It has an outside-in tracking system via Vive Lighthouse base stations.

For the eye tracking, the Binocular Add-on for the HTC Vive by Pupil Labs[2] is used. It can be clipped into the Vive or Vive Pro without any further mount. Adding an eye tracker to the inside of a HMD inherently reduces the FOV, since the lenses have to be moved back to make room for the cameras of the eye tracker. In practice, it is hardly distinctive for an unexperienced user from the full FOV of the HMD.

In order to allow for smooth applications an *Nvidia GTX 1070* and an *Intel i7-3770K CPU*, equivalent or better, are necessary for 90 FPS. For optimal user experience, better hardware is beneficial. Therefore, we used an *Nvidia RTX 2080 Ti* for our user study.

[1] https://www.vive.com/de/product/vive-pro/

[2] https://pupil-labs.com/products/vr-ar/

7.2 User Study

Five participants took part in this user study, aged 23 to 32, where one participant was female and were four male. One of the participants wore glasses (also in the HMD), two wore contact lenses, and another two had no visual aid at all. All of them were Caucasian and had relatively bright eye colors (blue, green, or light brown), with a distinctive difference of the pupil from their retina. One participant had very slight deuteranomaly, a red-green weakness, but they described it as barely noticeable in their everyday life.

In this pilot study we focused on collecting qualitative feedback of a few participants to investigate differences of human vision in real life and VR. We did not aim to obtain extensive statistical data, but detailed verbal feedback from the participants.

The user study protocol consisted of the following parts:

Study Information We explained the aim and general procedure of our study to the participant.

Real-World Experiment We set up a low-light scene to get a more accurate comparison, as well as to sharpen the user's attention. All the blinds in our Lab were shut and a small but bright light source was placed on a table approximately 5 m in front of the users. They were then asked to describe their perception of the light in regards to looks, body movement, eye movement, as well as the distance to the light. We asked, for example: What does the glare look like? How does it change, when you move around the room?

Eye Tracker Calibration The participants put on the HMD and we started the eye tracker calibration. First, they had to look at a green dot circling the center of the display, to calibrate the viewing direction. Second, we showed a black screen and a white screen for 5 seconds each, to calibrate the pupil size.

VR Experiment We showed the participants a low-light glare scene in VR, and asked them to compare our simulation to their vision during the real-world experiment. They could freely explore the VR scene and analyze the differences. We used an office scene[3] with only a single desk-lamp as light source.

Questionnaire We asked the users to compare the simulation to their real perception, while they were still in VR. They graded all implemented effects on a Likert Scale from one to seven, where one means not similar at all and seven stands for exactly the same. Additionally, they gave verbal feedback on perceived differences and gave reason to their ratings.

[3] https://4real3d.com/model/unreal-engine/ue4-scene/ue4-office/unreal-studio-office-scene-datasmith-beta-4/, with minor adaptations, especially regarding light emission.

The eye tracker generated volatile data for two participants due to glasses or mascara worn during the user study. It resulted in heavy flickering of the simulated glare – zero was returned as pupil size when the pupil could not be detected in that frame, which lead to unrealistic results. Our algorithm was not able to adapt to this unstable data.

7.2.1 Evaluation of Results

All participants were asked to rate six different categories on a Likert Scale from one to seven, where one means "completely different" and seven stands for "exactly the same" when comparing their real perception during the real-world experiment, to the one in the VR experiment. They were asked to evaluate and to explain their decisions the following categories:

Glare Appearance How does the effect of the glare around a bright light source look like?

Glare Movement When moving around the scene, or not looking straight at the glare, how does the glare change?

Eye Adaptation How does the illumination change, when looking from bright to dark regions or vice versa?

Visual Acuity How well are the details in dark regions perceived?

Scotopic Vision How do the colors change in dark regions?

Overall How would you rate the overall scene?

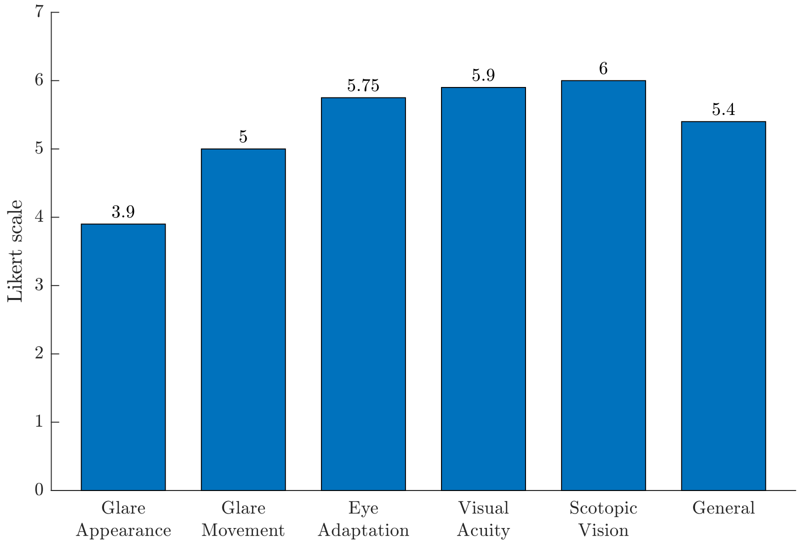


Figure 7.1: Average user study ratings, of five participants.

The average ratings of all participants per the different categories can be seen in Figure 7.1. Five of the six categories were, on average, voted five or above out of 7, with only the glare appearance voted 3.9 on average.

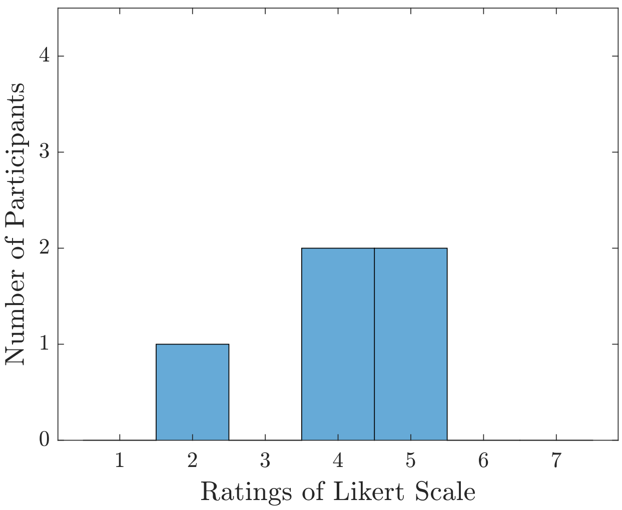


Figure 7.2: Distribution of ratings of Glare Appearance.

The distribution of the votes on glare appearance can be seen in Figure 7.2. In general, all participates stated that their perception of glare was much more delicate in regards to light streaks and much more detailed than our simulated glare. It was noted multiple times that the corona or the center of the glare seemed too intense in VR. Furthermore, the colors were described as too intense and saturated in VR. They appeared more reddish in VR than in real life and the center of the glare appeared whiter when looking at the real-world scene. However, some participants stated that the rainbow-like circles on the outside of the glare were very realistic and similar to their perception. On the other hand, one participant noted that they perceived more colored circles in real life than in VR – this, therefore, seems to be a very user-specific phenomenon. Additionally, it was mentioned by one participant, that their vision of glare was much more circular and less jagged than our simulation. Furthermore, one person noted that they found the slight pulsing of the glare irritating and disturbing. They described their glare perception as more fluid, like water, while they perceived the VR simulation as too jerky. Contrary, another participant mentioned that the pulsation of the glare in VR seemed very nice, while not entirely similar to their real perception; they still deemed it a realistic effect.

The participants rated the appearance of the look of the glare when moving around in the scene or changing gaze direction. The distribution of ratings can be seen in Figure 7.3. One participant claimed the glare disappeared too quickly when glancing away from it, another one claimed it was still too intense when focusing on something nearby. Another person rated the effect of eye movement in the scene to be just perfect. Furthermore, it was stated by one participant that they perceived fewer colors when looking away from

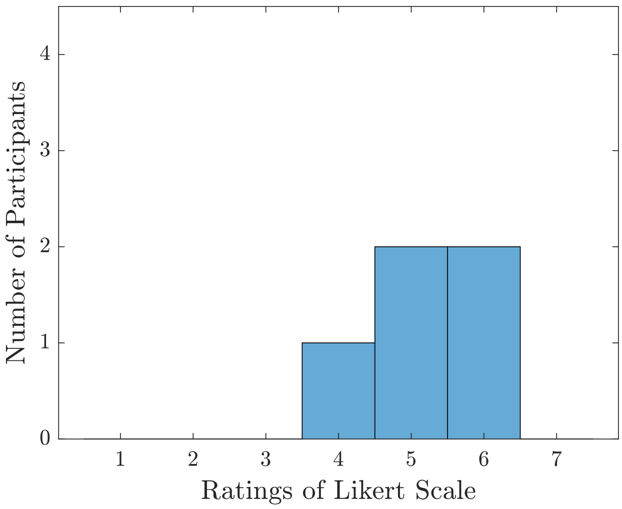


Figure 7.3: Distribution of ratings of Glare Movement.

the light in real life, while the simulation did not change color at all. Additionally, the effects of spacial movement in the scene were evaluated in this category – it has been noted multiple times that the glare effect did not seem to tilt correctly with the user, i.e., the participants expected some deformation when viewing the glare from the sides, but our glare stayed circular.

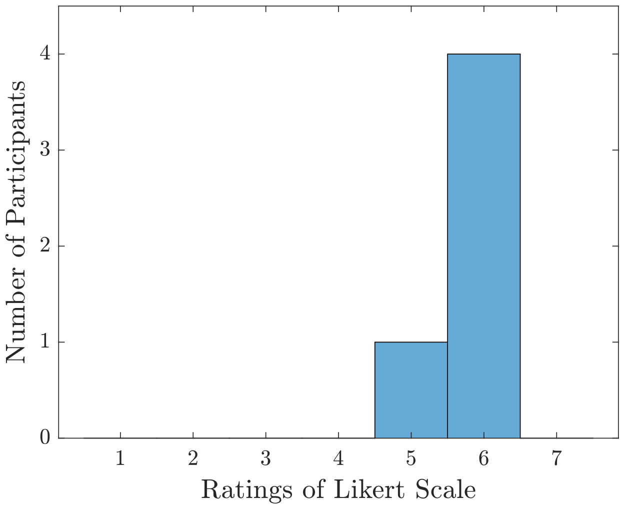


Figure 7.4: Distribution of ratings of Eye Adaptation.

The distribution of votes for the eye adaptation can be seen in Figure 7.4. Generally, this effect was commented on very positively, i.e., most participants said it resembled their perception a lot, it seemed natural and they were very comfortable with it. The main point of critique was that the adaptation from bright to dark scenes was too slow.

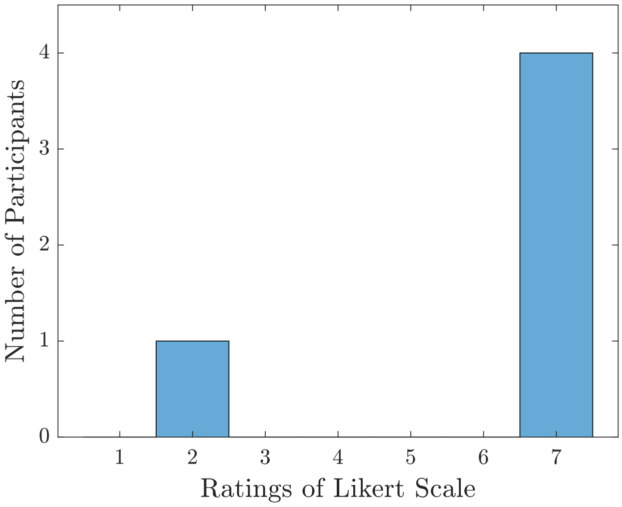


Figure 7.5: Distribution of ratings of Visual Acuity.

The distribution of the perception of visual acuity can be seen in Figure 7.5. This effect was voted as entirely equal to their perception by four out of five participants. All of those noted that they hardly had noticed it at all before being asked to grade it. One participant added that when viewing a dark scene, more "noise", i.e., irregularities, appear. On the other hand, one participant noted that this effect was way too intense in the scene at hand, and while the simulated loss of visual acuity might be realistic in a dark scene without light, they said that it did not match the scene at hand at all.

The votes for the color shift in scotopic vision was evaluated in Figure 7.6. Two participants noted that the colors completely matched their perception in a dimly lit scene. Another one noted that the scene, in general, seemed to be very warm with brown and red colors. Therefore the color shift in the dark areas seemed too intense. On the other hand, another participant mentioned that they thought the colors of the darker regions were too intense, i.e., they said a more intense shift would be necessary. Another one mentioned that some shadowed areas seemed too bright, while others appeared completely black – yet, this could also be influenced by the quality of the used lightmap generated by UE.

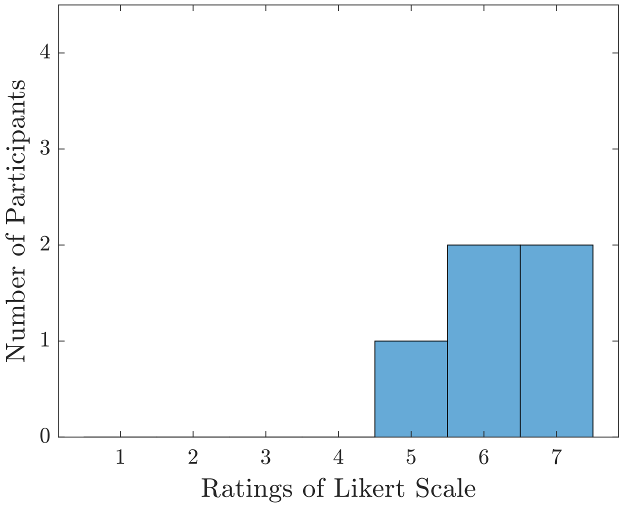


Figure 7.6: Distribution of ratings of Scotopic Vision.

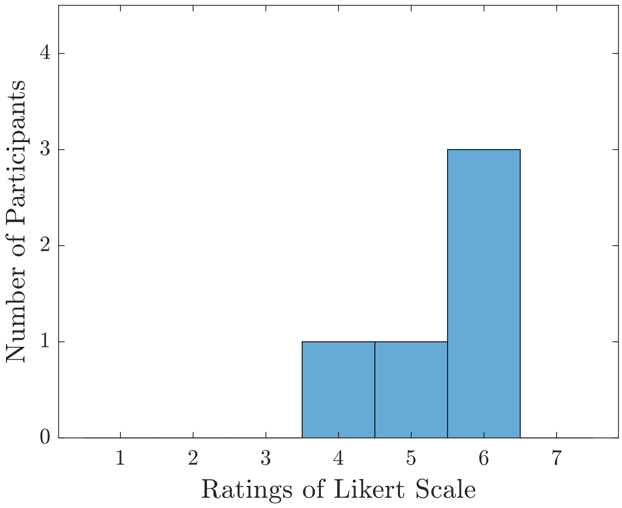


Figure 7.7: Distribution of ratings of Overall.

The distribution of the overall ratings of the simulation can be seen in Figure 7.7. Most participants rated our simulation above average. One participant rated it to be neither good nor bad, all the others found it good or very good. However, no one rated it as perfect. Three out of five participants noted that all the effects were harmonious and generated a realistic scene. One of the most significant downsides mentioned in this category was the appearance of the glare as being too coarse, especially the rays from the center, as well as being too big in size. Furthermore, participants mentioned that, in

VR, the lamp as a light source was not visible behind the intense glare, while in reality, the lamp was visible.

In summary, it can be said that the user study gave valuable insights into the perception of glare and night scenes in VR. While most participants agreed on the main arguments, like the glare appearing as too coarse and intense, or the proper simulation of night vision, some outliers confirmed our assumption that human perception is hugely subjective. Therefore, it is difficult to generate a generally acceptable model. We argue that a method, which allows individual adjustment of each effect for each user, like the one we propose, could overcome this problem.

7.3 Discussion and Limitations

Evaluating the results of the user study showed that the most significant improvements for future work could be made regarding the glare. Other phenomena, like scotopic vision and temporal adaptation, have already been widely researched, and by building upon this research we were able to generate plausible simulations. Glare effects are highly dependent on the internal eye anatomy, like particle count and size, which greatly varies from person to person and cannot be measured without specialized equipment. We can adapt our simulation to external eye parameters like pupil size and view direction. In future work, we could integrate more details into our simulation. However, this would require substantial medical measurements of the characteristics of the user's eye, and calibrations beforehand.

All user study participants mentioned that the glaring pattern was too coarse. They noted that their perceived glare, in reality, had more detailed and delicate light streaks and the simulated glare, therefore, seemed very thick and unnatural. While this might be an issue in the algorithm, it has to be mentioned that the used VR-HMD is not able to display details as well as the human eye can perceive them. The Vive Pro has an PPD value of 13.85, while the human eye at 20/20 vision can process approximately 60 PPD [SLWT19].

Another limitation of our work is that the glare kernel is always based on a yellow-white light, independent of the rendered light's color. Different colors and, therefore wavelengths of light, influence the perceived glare in different ways. While in theory, convolving the glare kernel with colored light should influence the final output color, this happens only very slightly and does not sufficiently resemble the real perceived glare of colored light.

One element not yet included in our simulation was the influence of eyelashes on the perception. The eye tracker is able to detect blinking, with a duration on average of 300–400 ms [Vol04]. It returns a pupil size of zero when the eyes are closed, as well as a decreased size when the lid is halfway closed, as shown in Figure 7.8. Therefore, the size of the glare aperture is reduced accordingly until the minimal size and increased again when opening the eyes. For this reason, our glare kernel adapts in some way to a blink of the eye. However, the influence of eyelashes interfering with the incident

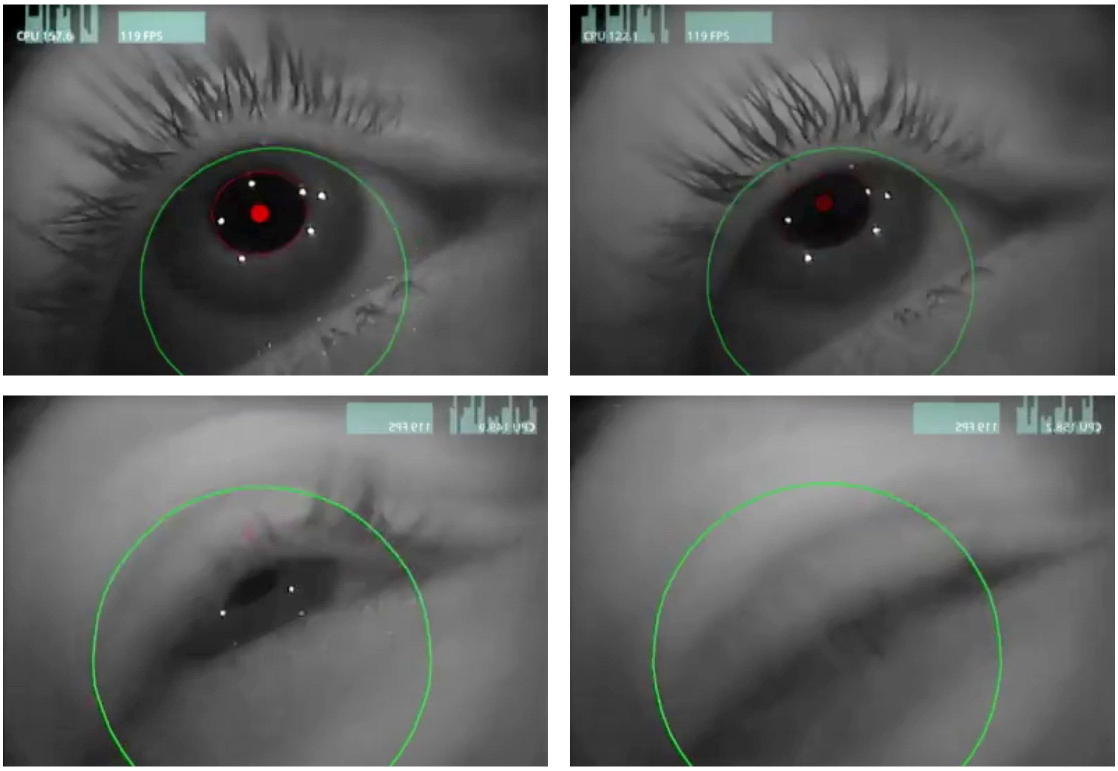


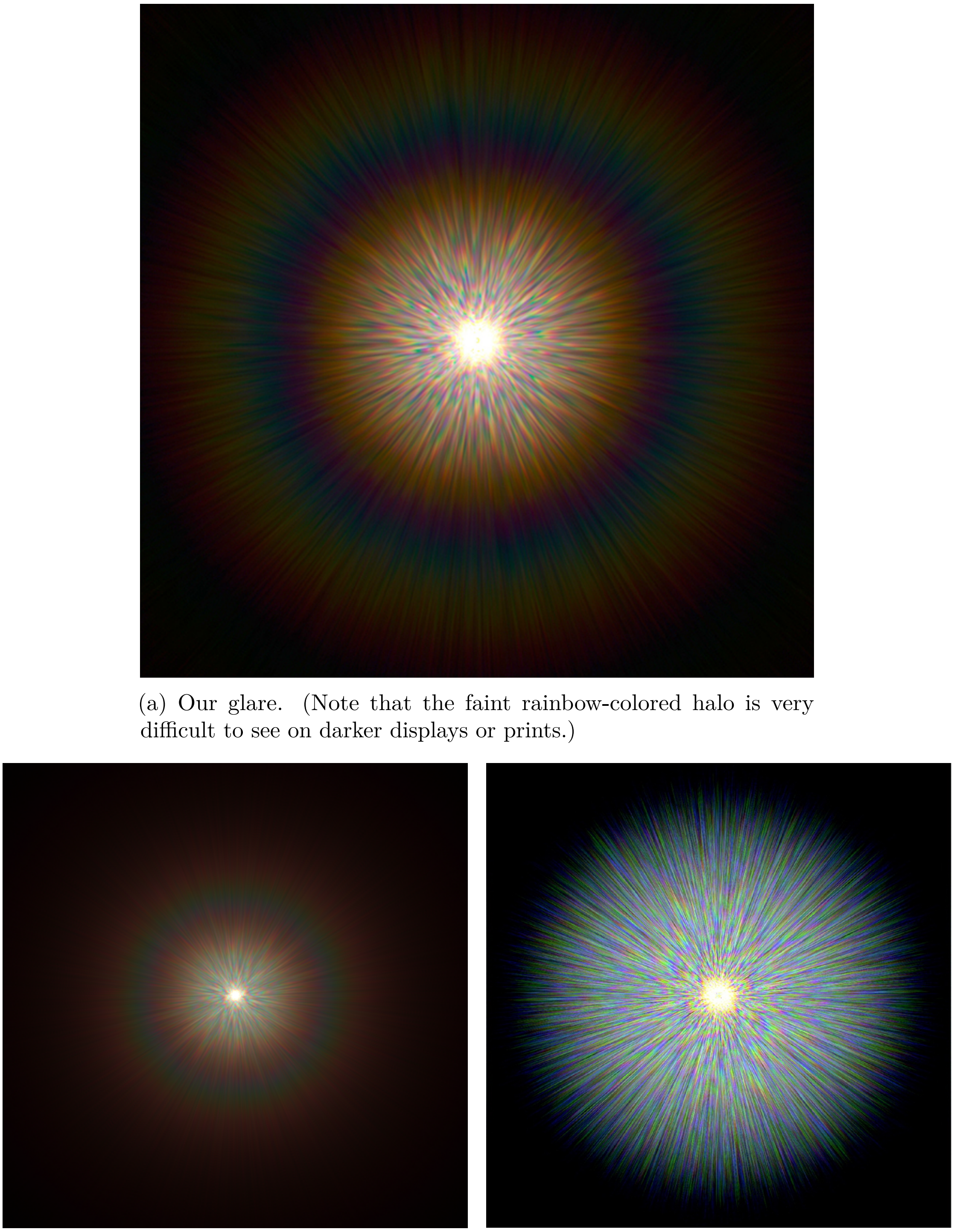
Figure 7.8: A blinking eye captured by our eye tracker. The red dot and red circle highlight the detected pupil.

light was not accounted for in our current implementation. However, none of the user study participants mentioned that phenomenon as “missing” or described any differences noticed.

7.4 Comparisons to Related Work

We produce a real-time glare effect, that depends on the user’s pupil size and is based on wave optics. In the proposed methods by Ritschel et al. [RIF$^+$09] and Berg et al. [vdBHC05] similar approaches were presented. Their effects are either based on static calculations or preset functions of the pupil size in regards to changes in lighting. A comparison of the three different produced glare kernels is shown in Figure 7.9. Our glare has a very intense center, similar to Berg et al. [vdBHC05]. A colorful, but fainter, corona is visible, similar to the kernel proposed by Ritschel et al. [RIF$^+$09].

Similar to Ritschel et al. [RIF$^+$09], we concluded that the glare perception could greatly vary between different users and there is no universally correct solution. Our user study participants also described similar phenomena as the participants of the study conducted by Berg et al. [vdBHC05]. While their simulated glare seems to be a lot more blueish



(a) Our glare. (Note that the faint rainbow-colored halo is very difficult to see on darker displays or prints.)

(b) Ritschel et al. [RIF+09]

(c) Berg et al. [vdBHC05]

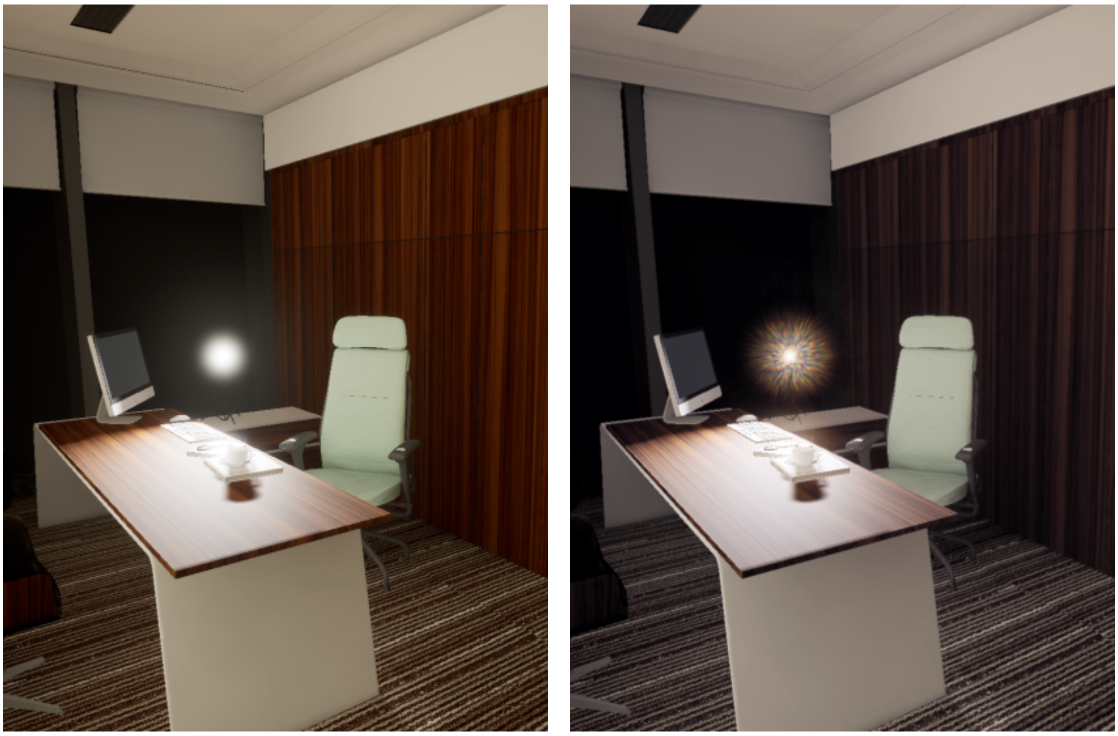
Figure 7.9: Comparison of different glare kernels.

than ours, still the same distinct features can be seen.

Neither Ritschel et al., nor Berg et al. conducted experiments with users to compare their simulated glare to a real-world situation. To the best of our knowledge, we are the first to let users compare a VR simulation of a glare to a glare effect perceived in the real world. As our results showed, our simulation achieved some promising initial results, but more work is needed regarding the glaring pattern itself. This can be achieved by a revised user study – employing the results of our first one, in order to generate a better understanding of the perception of glare in real life and in simulations, like VR.

Our simulated temporal eye adaptation, visual acuity reduction and scotopic color vision were generally well accepted by all user-study participants. Our results indicate that our method is able to enhance the user's experience of the scene in low lighting conditions.

When comparing an image of a low-light scene generated by a default renderer to our implementation, as shown in Figure 7.10, the differences can clearly be seen. We approximate the simulation of night vision better, i.e., our approach is closer to human perception than when using the default settings. The work by Krawczyk et al. [KMS05] showed similar results. They used a Gaussian pyramid due to the spacial similarities for all their implemented effects (temporal luminance adaptation, visual acuity and veiling



(a) Default UE settings.

(b) Our adaptations.

Figure 7.10: Comparison of a renderer's default settings to our implementation.

luminance). In contrast to their approach, we only generate four steps (128 × 128 to 16 × 16 pixels) of a Gaussian pyramid to calculate the average frame luminance. We use a distinct glare method via a PSF which can not be integrated into their workflow. We can still compute the loss of visual acuity efficiently in real-time by splitting the kernel into two 1D filter processes. Therefore, our approach can display all the effects proposed by Krawczyk with an advanced glare in real-time in VR.

Our user study shows that these additions, i.e., temporal luminance adaptation, perceptual glare, visual acuity reduction and scotopic color vision, to the scene are perceptually significant and necessary for realistic night scenes. Like stated in most of the perception-based methods, every user experiences low-light vision slightly differently. This work has laid a ground base for further research regarding individual adaptation for everyone in rendering of night scenes or low-light scenes.

7.5 Runtimes

Our application can run in real-time on current PCs and has quite robust performance on high-end hardware. In the following, the rendering times, of a PC with an *Nvidia GTX 1070* graphics card, for different stages of our method are evaluated, see Table 7.1.

Table 7.1: Runtimes for different rendering stages.

Stages	**Count**	**Avg. GPU times**
Aperture	Once	0.04 ms
Spectral PSF	Once	2.34 ms
Visual Acuity Reduction	Per eye	0.36 ms
Temporal Eye Adaptation	Per eye	0.18 ms
Glare Convolution	Per eye	1.47 ms
Scotopic Colors	Per eye	0.14 ms
Overall Post-Processing	Per eye	2.42 ms
Overall GPU	All-together	9.57 ms

CHAPTER 8

Conclusion and Future Work

In this chapter, we summarize the main contributions of this thesis, and discuss possible improvements for future work.

8.1 Conclusion

In this thesis, we propose a tool for simulating human perception of light in VR. We adapted the glare methodology proposed by Ritschel et al. [RIF^{+}09] to run in real-time VR and AR. Additionally, we added eye tracking to their approach to adapted the simulation to the user's viewing direction and pupil size. We also incorporated and adapted the work by Krawczyk et al [KMS05] to perform efficiently in our framework, in order to add more realism in low-light scenes.

Our work has made some significant improvements to standard simulations of low light scenes and added further realism to the simulation of glare. The results of our pilot study indicate that our developed application can simulate temporal eye adaptation, reduced visual acuity and scotopic color vision similar or very close to what humans perceive in real-life situations. To the best of our knowledge, our glare is the first perceptual glare to run in real-time VR and AR, that is based on medical research and implements an on-line PSF. We also identified areas of improvement for glare appearance and glare movement, to increase realism even further in future work.

With our user study, comparing a real-life situation to a VR simulation, we were able to show the importance of medically-based, perceptional effects to increase the quality of visual perception in VR, which leads to more realism of the simulation and potentially also better immersion.

Our work can be added to any standard real-time rendering tool or framework as independent post-processing steps for each effect or a general post-processing tool.

8.2 Future Work

In this thesis, we presented a real-time application for VR and AR to approximate human visual perception inside an HMD, based on medical studies and using eye tracking. We have identified the following possibilities to further improve realism in future work.

8.2.1 Further Medical Specifications for the Glare

While we aimed to include all relevant medical details in our glare algorithm, the following improvements can be made.

Lens Deformations A medical feature that has not yet been exploited in our approach is the deformation of the lens in the human eye, depending on eye accommodation and contractions of the ciliary muscle [RIF^{+}09]. These movements are determining the animation of the glaring pattern, in addition to the pupil size. We are already able to simulate the glare animation quite accurately, according to our user study participants. However, the deformation of the lens might be able to increase the realism of our simulation further.

Glare Sizes Depending on Light Source In our simulation, the size of the glare is only dependent on the size of the rendered image and therefore constant for all types, sizes and intensities of light. We assume a single direction of light incidence, i.e., parallel to the visual axis of the eye. Additional research regarding the external influences of light, i.e. the angle of light incidence or the size of the luminary, is necessary for improved realism for all lighting situations.

More Detailed Glare Texture Multiple user study participants noted that the glare they perceived in real life, was more delicate than in our simulation. While this might be an issue of the HMD's resolution (see Section 7.3), we could try to generate more detailed and delicate light streaks by using bigger textures for the generation of the glare kernel. However, this would also lead to the necessity of bigger FFTs, which would result in slower run times.

8.2.2 Additional Effects

Our approach already includes the most prominent effects on human vision. However, additional features could increase the realism in VR still.

Billboards for Glare Application A possibility to avoid many costly FFTs would be to not convolve the glare as a post-processing step with the image, but render it on a semi-transparent plane in front of the light source. This would additionally add more perspective to the glare, as it was mentioned multiple times in our user study. However, this approach would only be well suited for point lights, not area lights.

Afterimages A very prominent effect that has not been accounted for in this thesis are so-called afterimages or after effects [Pur19]. Looking into a very bright light source for an extended amount of time temporarily bleaches the photoreceptors in the eye. This leads to images appearing in our vision even after the luminaire has vanished, hence the name. With a more detailed simulation of the absorption and reproduction of the photoreceptors, the simulation of these effects could further increase the realism of a VR simulation.

Noise in Scotopic Vision Erroneously firing rods during low lighting ambiance leads to perceived noise in the human visual system (see Section 2.2.2 iv.). We see this as a “filter” over our vision. As mentioned by one participant in our user study, simulating this noise could help increase realism.

User’s Custom Tonemapper It has been shown in many studies, that the perception of light and lightness is unique to every person and even varies in the same person at different points in time [Pur19]. In order to allow a personally optimal tone mapping, a user’s custom curve of brightness perception can be constructed via visual tests with multiple repetitions. The benefit of doing so in VR is that the surroundings and environment are completely controllable, due to the enclosed HMD.

8.2.3 User Studies

Larger user studies should be conducted on the topic of light simulation to allow for a direct comparison of the perceived phenomena and simulated phenomena.

Using an HDR Display Further assessment of the proposed methodology could be made via an HDR display. Through simple tests using black images with small, white circles, we have concluded that glare is only minimally present on average LDR monitors. However, on HDR monitors, an evident glare phenomenon was observed. This has led us to believe that while our methodology was mainly developed for VR applications, a further assessment of the glare appearance on rendered images could be done on an HDR display to be able to compare the rendered glare to real life at the same time.

Using a High-Quality HDR Camera Evaluating the “correctness” of our implementation could also be done by applying it to real-world HDR images and comparing it to a user’s perception similar to [YBMS05] or [WC06]. In contrast to the evaluation method above, this allows for comparison in AR scenes.

AR User Study While our application included an AR mode, it was not reviewed in our conducted user study. The comparison of a real-world scene to an AR simulation could give further insights into the differences of real-world perception and the perception in an HMD. However, the cameras of our currently used HMD only have a resolution of 640 × 480 pixel and are not able to capture HDR scene information.

List of Figures

List of Tables

List of Algorithms

Acronyms

AR Augmented reality. ix, xi, 1, 4, 37, 41, 47, 58, 59, 65, 77–79

FFT Fast Fourier transformation. 6, 7, 21, 44, 46–48, 54, 59–62, 78, 82

FOV Field of view. 1, 52, 65

FPS Frames per second. 2, 41, 65

HDR High dynamic range. 20, 32, 39, 51, 58, 79

HMD Head mounted display. xi, 1, 2, 52, 53, 59, 65, 66, 72, 78, 79

Lab CIE L*a*b* color space. 51–53, 62

LDR Low dynamic range. xi, 20, 25, 26, 32, 33, 55, 79

PPD Pixel per degree. 52, 65, 72

PSF Point spread function. 27–30, 37, 41–47, 61, 76, 77, 82

UE Unreal Engine. 3, 27, 58–63, 70, 75, 81, 82

VR Virtual reality. ix, xi, 1–4, 26, 27, 29, 32, 37, 39, 41, 47, 51–53, 58, 59, 61, 62, 66–68, 72, 75–79

Bibliography

[Air35] George Biddell Airy. On the diffraction of an object-glass with circular aperture. *Transactions of the Cambridge Philosophical Society*, 5:283, 1835.

[BD80] J K Bowmaker and H J Dartnall. Visual pigments of rods and cones in a human retina. *The Journal of Physiology*, 298(1):501–511, 1980.

[BDE$^+$09] Michael Bass, Casimer DeCusatis, Jay Enoch, Vasudevan Lakshminarayanan, Guifang Li, Carolyn Macdonald, Virendra Mahajan, and Eric Van Stryland. *Handbook of optics, Volume II: Design, fabrication and testing, sources and detectors, radiometry and photometry.* McGraw-Hill, Inc., 2009.

[Bri87] Encyclopaedia Britannica. Sensory reception: human vision: structure and function of the human eye. *Encyclopedia Brittanica*, 27:179, 1987.

[BSJ10] Anne Bolling, Gunilla Sörensen, and Jonas Jansson. Simulating the effect of low lying sun and worn windscreens in a driving simulator. *Proceedings of the Driving Simulation*, pages 23–31, 1 2010.

[Čad07] Martin Čadík. Perception Motivated Hybrid Approach to Tone Mapping. *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2007.

[Cha07] Chabacano. Diagram of A Human Eye. https://commons.wikimedia.org/wiki/File:Eye-diagram_no_circles_border.svg, accessed on 17.12.2019, 3 2007.

[ČWNA08] Martin Čadík, Michael Wimmer, Laszlo Neumann, and Alessandro Artusi. Evaluation of HDR tone mapping methods using essential perceptual attributes. *Computers and Graphics (Pergamon)*, 32(3):330–349, 2008.

[DFD$^+$15] Haarm Pieter Duiker, Alexander Forsythe, Scott Dyer, Ray Feeney, Will McCown, Jim Houston, Andy Maltz, and Doug Walker. ACEScg: A common color encoding for visual effects applications. In *Proceedings - DigiPro 2015, Digital Production Symposium*, page 53. Association for Computing Machinery, Inc, 8 2015.

[DMAC03] F Drago, K Myszkowski, T Annen, and N Chiba. Adaptive Logarithmic Mapping For Displaying High Contrast Scenes. *Computer Graphics Forum*, 22(3):419–426, 2003.

[EMU15] Gabriel Eilertsen, Rafał K Mantiuk, and Jonas Unger. Real-time Noise-aware Tone Mapping. *ACM Trans. Graph.*, 34(6):198:1–198:15, 10 2015.

[EUM16] G Eilertsen, J Unger, and R K Mantiuk. Chapter 7 - Evaluation of Tone Mapping Operators for HDR Video. In Frédéric Dufaux, Patrick Le Callet, Rafał K Mantiuk, and Marta Mrak, editors, *High Dynamic Range Video*, pages 185 – 207. Academic Press, 2016.

[FA53] G. A. Fry and M. Alpern. The effect of a peripheral glare source upon the apparent brightness of an object. *Journal of the Optical Society of America*, 43(3):189–195, 1953.

[FC07] Luuk Franssen and Joris Eduard Coppens. *Straylight at the retina : scattered papers*. PhD thesis, University of Amsterdam, 2007.

[FDZ17] Alessio Facchin, Roberta Daini, and Daniele Zavagno. The Glare Effect Test and the Impact of Age on Luminosity Thresholds. *Frontiers in Psychology*, 8:1132, 2017.

[FPBC09] Sira Ferradans, Edoardo Provenzi, Marcelo Bertalmio, and Vicent Caselles. TSTM: A two-stage tone mapper combining visual adaptation and local contrast enhancement. *IMA Preprint Series # 2253*, 2009.

[FTCvdB07] Luuk Franssen, Juan Tabernero, Joris E Coppens, and Thomas J T P van den Berg. Pupil Size and Retinal Straylight in the Normal Eye. *Investigative Ophthalmology & Visual Science*, 48(5):2375–2382, 1 2007.

[Goo05] Joseph W Goodman. *Introduction to Fourier optics*. Roberts and Company Publishers, 2005.

[GPB$^+$13] Harilaos S Ginis, Guillermo M Perez, Juan M Bueno, Alexandros Pennos, and Pablo Artal. Wavelength Dependence of the Ocular StraylightWavelength Dependence of the Ocular Straylight. *Investigative Ophthalmology & Visual Science*, 54(5):3702–3708, 1 2013.

[Gre06] Orlando Greyson. Animation that illustrates the pupillary light reflex. https://commons.wikimedia.org/wiki/File:Eye_dilate.gif, accessed on 08.01.2020, 12 2006.

[GW07] Rafael C Gonzalez and Richard E Woods. *Digital Image Processing (3rd Edition)*. Pearson, 2007.

[Hec02] E Hecht. *Optics*. Pearson education. Addison-Wesley, 2002.

[HESL11] Matthias Hullin, Elmar Eisemann, Hans-Peter Seidel, and Sungkil Lee. Physically-based real-time lens flare rendering. In *ACM Transactions on Graphics (TOG)*, volume 30, page 108, 2011.

[IG09] Kristina Irsch and David Guyton. Anatomy of Eyes. *Encyclopedia of Biometrics*, 1 2009.

[JR16] Nathaniel L Jones and Christoph F Reinhart. Real-time visual comfort feedback for architectural design. In *PLEA 2016 Los Angeles–32nd International Conference on Passive and Low Energy Architecture*, pages 659–664, 2016.

[JS17] Cheolkon Jung and Tingting Sun. Optimized Perceptual Tone Mapping for Contrast Enhancement of Images. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(6):1161–1170, 6 2017.

[KDMF16] T Kunkel, S Daly, S Miller, and J Froehlich. Chapter 15 - Perceptual Design for High Dynamic Range Systems. In Frédéric Dufaux, Patrick Le Callet, Rafał K Mantiuk, and Marta Mrak, editors, *High Dynamic Range Video*, pages 391 – 430. Academic Press, 2016.

[KEW+19] Katharina Krösl, Carmine Elvezio, Michael Wimmer, Matthias Hürbe, Steven Feiner, and Sonja Karst. ICthroughVR: Illuminating cataracts through virtual reality. In *26th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2019 - Proceedings*, pages 655–663. Institute of Electrical and Electronics Engineers Inc., 3 2019.

[KMN+04] Masanori Kakimoto, Kaoru Matsuoka, Tomoyuki Nishita, Takeshi Naemura, and Hiroshi Harashima. Glare generation based on wave optics. In *Proceedings - Pacific Conference on Computer Graphics and Applications*, pages 133–142, 2004.

[KMS05] Grzegorz Krawczyk, Karol Myszkowski, and Hans Peter Seidel. Perceptual effects in real-time tone mapping. In *Spring Conference on Computer Graphics, SCCG 2005 - Conference Proceedings*, pages 195–202, 2005.

[KPB14] Moritz Kassner, William Patera, and Andreas Bulling. Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-based Interaction. In *Adjunct Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp ’14 Adjunct, pages 1151–1160, New York, NY, USA, 2014. ACM.

[Lak16] Vasudevan Lakshminarayanan. Light Detection and Sensitivity. In *Handbook of Visual Display Technology*, pages 105–112. Springer International Publishing, Cham, 2016.

[LRP97] Gregory Ward Larson, Holly Rushmeier, and Christine Piatko. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics*, 3(4):291–306, 1997.

[LSC04] Patrick Ledda, Luis Paulo Santos, and Alan Chalmers. A Local Model of Eye Adaptation for High Dynamic Range Images. In *Proceedings of the 3rd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, AFRIGRAPH '04, pages 151–160, New York, NY, USA, 2004. ACM.

[Man17] R Mantiuk. Chapter 10 - Gaze-Dependent Tone Mapping for HDR Video. In Alan Chalmers, Patrizio Campisi, Peter Shirley, and Igor G Olaizola, editors, *High Dynamic Range Video*, pages 189 – 199. Academic Press, 2017.

[MAT20] MATLAB Documentation. 2-D fast Fourier transform, 2020.

[MGVM16] Benjamin Meyer, Steve Grogorick, Mark Vollrath, and Marcus Magnor. Simulating Visual Contrast Reduction During Nighttime Glare Situations on Conventional Displays. *ACM Trans. Appl. Percept.*, 14(1):4:1–4:20, 7 2016.

[MH17] Joschka Mütterlein and Thomas Hess. Immersion, Presence, Interactivity: Towards a Joint Understanding of Factors Influencing Virtual Reality Acceptance and Use. *AMCIS 2017 Proceedings*, 8 2017.

[MM13] Radosław Mantiuk and Mateusz Markowski. Gaze-dependent tone mapping. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7950 LNCS, pages 426–433, 2013.

[MP08] Bob Mellish and Pbrks. How a pinhole camera works. https://upload.wikimedia.org/wikipedia/commons/3/3b/Pinhole-camera.svg, accessed on 21.12.2019, 5 2008.

[MT19] Karol Myszkowski and Okan Tarhan Tursun. Perception for Computer Graphics (Lecture, Winter Semester 2018/2019), 2019.

[Ngu07] Hubert Nguyen. *Gpu Gems 3*. Addison-Wesley Professional, first edition, 2007.

[Nus81] Henri J Nussbaumer. The Fast Fourier Transform. In *Fast Fourier Transform and Convolution Algorithms*, pages 80–111. Springer, Berlin, Heidelberg, 1981.

[PAF$^+$08] Dale Ed Purves, George J Augustine, David Ed Fitzpatrick, William C Hall, Anthony-Samuel Ed LaMantia, James O McNamara, and Leonard E White. Neuroscience. *Sinauer associates*, 2008.

[Pal99] Stephen E Palmer. *Vision science: Photons to phenomenology.* MIT press, 1999.

[Poy98] Charles Poynton. Frequently asked questions about gamma. *Rapport Technique, janvier*, 152, 1998.

[Pur19] Werner Purgathofer. Lecture Farbe (Color) - Human Perception Basics Slides, 2019.

[RdBT10] Jos J Rozema, Thomas J T P den Berg, and Marie-José Tassignon. Retinal Straylight as a Function of Age and Ocular Biometry in Healthy Eyes. *Investigative Ophthalmology & Visual Science*, 51(5):2795–2799, 1 2010.

[Rem12] Lee Ann Remington. Chapter 6 - Aqueous and Vitreous Humors. In Lee Ann Remington, editor, *Clinical Anatomy and Physiology of the Visual System (Third Edition)*, pages 109 – 122. Butterworth-Heinemann, Saint Louis, third edition edition, 2012.

[RIF$^+$09] Tobias Ritschel, Matthias Ihrke, Jeppe Revall Frisvad, Joris Coppens, Karol Myszkowski, and H-P Seidel. Temporal glare: Real-time dynamic simulation of the scattering in the human eye. In *Computer Graphics Forum*, volume 28, pages 183–192, 2009.

[RSSF02] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic Tone Reproduction for Digital Images. *ACM Trans. Graph.*, 21(3):267–276, 7 2002.

[Sch06] E Fred Schubert. Human eye sensitivity and photometric quantities. *Light-emitting diodes*, pages 275–291, 2006.

[SEC84] Robert Shapley and Christina Enroth-Cugell. Visual adaptation and retinal gain controls. *Progress in retinal research*, 3:263–346, 1984.

[SLE18] Leonardo Scandolo, Sungkil Lee, and Elmar Eisemann§. Quad-Based Fourier Transform for Efficient Diffraction Synthesis. *Computer Graphics Forum*, 37(4):167–176, 2018.

[SLWT19] Susannah Soon, Artur Lugmayr, Andrew Woods, and Tele Tan. Understanding head-mounted display FOV in maritime search and rescue object detection. In *Proceedings - 2018 IEEE International Conference on Artificial Intelligence and Virtual Reality, AIVR 2018*, pages 116–119. Institute of Electrical and Electronics Engineers Inc., 1 2019.

[SMV$^+$16] Leopoldo Spadea, Giorgia Maraone, Francesca Verboschi, Enzo Maria Vingolo, and Daniele Tognetto. Effect of corneal light scatter on vision: a review of the literature. *International journal of ophthalmology*, 9(3):459, 2016.

[Sri18] Mittanamalli S Sridhar. Anatomy of cornea and ocular surface. *Indian journal of ophthalmology*, 66(2):190, 2018.

[SSZ^{+}95] Greg Spencer, Peter Shirley, Kurt Zimmerman, Donald P Greenberg, and others. Physically-based glare effects for digital images. In *SIGGRAPH*, volume 95, pages 325–334, 1995.

[SZ07] Sathiyam2k and ZStardust. Section view of the human eye. https://commons.wikimedia.org/wiki/File:Eyesection.svg, accessed on 21.12.2019, 7 2007.

[Tho19] Liji Thomas. What is an Aqueous Flare?, 2019.

[TMM19] Ba Chien Thai, Anissa Mokraoui, and Basarab Matei. Contrast enhancement and details preservation of tone mapped high dynamic range images. *Journal of Visual Communication and Image Representation*, 58:589 – 599, 2019.

[Tuc97] Valerii V Tuchin. Light scattering study of tissues. *Physics-Uspekhi*, 40(5):495–515, 5 1997.

[VDB86] T. J T P Van Den Berg. Importance of pathological intraocular light scatter for visual disability. *Documenta Ophthalmologica*, 61(3-4):327–333, 1 1986.

[vdBHC05] Thomas J T P van den Berg, Michiel P J Hagenouw, and Joris E Coppens. The Ciliary Corona: Physical Model and Simulation of the Fine Needles Radiating from Point Light Sources. *Investigative Ophthalmology & Visual Science*, 46(7):2627–2632, 1 2005.

[vdBS99] Thomas J T P van den Berg and Henk Spekreijse. Light scattering model for donor lenses as a function of depth. *Vision Research*, 39(8):1437 – 1445, 1999.

[VMGM15] Peter Vangorp, Karol Myszkowski, Erich W Graf, and Rafał K Mantiuk. A Model of Local Adaptation. *ACM Trans. Graph.*, 34(6):166:1–166:13, 10 2015.

[Vol04] Nicholas J Volpe. Adler's Physiology of the Eye: Clinical Application, Tenth Edition. *Journal of Neuro-Ophthalmology*, 24(4), 2004.

[Wan09] Ruye Wang. Properties of Fourier Transform, 2009.

[WC06] Jan Wienold and Jens Christoffersen. Evaluation methods and development of a new glare prediction model for daylight environments with the use of CCD cameras. *Energy and buildings*, 38(7):743–757, 2006.

[Wei20] Eric W. Weisstein. Fourier Transform–1, 2020.

[WLS$^+$18] Andreas Walch, Christian Luksch, Attila Szabo, Harald Steinlechner, Georg Haaser, Michael Schwärzler, and Stefan Maierhofer. Lens flare prediction based on measurements with real-time visualization. *Visual Computer*, 34(9):1155–1164, 9 2018.

[WMP19] Marek Wernikowski, Radosław Mantiuk, and Rafał Piórkowski. Preferred Model of Adaptation to Dark for Virtual Reality Headsets. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11295 LNCS, pages 118–129. Springer Verlag, 2019.

[WSS13] Chris Wyman, Peter-Pike Sloan, and Peter Shirley. Simple Analytic Approximations to the CIE XYZ Color Matching Functions. Technical Report 2, NVIDIA, 2013.

[YBMS05] Akiko Yoshida, Volker Blanz, Karol Myszkowski, and Hans-Peter Seidel. Perceptual evaluation of tone mapping operators with real-world scenes. In Bernice E Rogowitz, Thrasyvoulos N Pappas, and Scott J Daly, editors, *Human Vision and Electronic Imaging X*, volume 5666, page 192 – 203. SPIE, 2005.

[YIMS08] Akiko Yoshida, Matthias Ihrke, Rafał Mantiuk, and Hans-Peter Seidel. Brightness of the Glare Illusion. In *Proceedings of the 5th Symposium on Applied Perception in Graphics and Visualization*, APGV '08, pages 83–90, New York, NY, USA, 2008. ACM.

[YS20] Myron Yanoff and Joseph W Sassani. 12 - Vitreous. In Myron Yanoff and Joseph W Sassani, editors, *Ocular Pathology (Eighth Edition)*, pages 481 – 493.e2. Elsevier, London, eighth edition edition, 2020.

[ZDRCLG19] Juan F Zapata-Díaz, Hema Radhakrishnan, W Neil Charman, and Norberto López-Gil. Accommodation and age-dependent eye model based on in vivo measurements. *Journal of Optometry*, 12(1):3 – 13, 2019.