Biologically Inspired Ray Tracing

Master's Thesis

Johannes Bohlig

Supervisor
Prof. Dr. Filip Sadlo

Second Supervisor
Dr. Susanne Krömker

Heidelberg, Germany, January 10, 2024

Faculty of Mathematics and Computer Science
Heidelberg University

Declaration of Authorship

I hereby certify that I have written the work myself and that I have not used any sources or aids other than those specified and that I have marked what has been taken over from other people's works, either verbatim or in terms of content, as foreign. I also certify that the electronic version of my thesis transmitted completely corresponds in content and wording to the printed version. I agree that this electronic version is being checked for plagiarism at the university using plagiarism software.

first and last name	city, date and signature

ABSTRACT

Ray tracing has become increasingly popular in recent years and is mainly used for the realistic simulation of light in the field of computer graphics. This work is dedicated to this topic and the question of how ray tracing can be used to recreate the vision of the human eye.
For this purpose, methods are developed that simulate the lens system of the eye, a sampling procedure to realize the ray path within this lens system, as well as the accommodation and thus the focusing of an object in space. On this basis, effects such as chromatic aberration are simulated.
These methods are investigated with an explorative research approach and the basis for implementation is realized by means of literature research using a specially developed ray tracing engine.
The results show that, in principle, it is possible to implement the vision and simulation of the eye using the ray tracing algorithm. However, in order to obtain an all-encompassing simulation for the holistic representation of reality, a much more detailed simulation is necessary, which can be developed in further work and can deal with topics such as the exact simulation of the lens of the eye.

Zusammenfassung

Das Thema Raytracing gewinnt in den letzten Jahren immer mehr an Beliebtheit und wird vor allem zur realistischen Nachbildung des Lichts im Bereich der Computergrafik verwendet. Diese Arbeit widmet sich diesem Thema und der Frage, wie sich mittels Raytracing die Sicht des menschlichen Auges nachempfinden lässt.
Zu diesem Zweck werden Methoden entwickelt, welche das Linsensystem des Auges, ein Samplingverfahren zur Realisierung des Strahlenverlaufs innerhalb dieses Linsensystems, sowie die Akkommodation und damit das Scharfstellen eines Objektes im Raum, simulieren. Auf dieser Basis werden Effekte wie beispielsweise die Chromatische Aberration simuliert.
Diese Methoden werden mit einem explorativen Forschungsansatz untersucht und mittels Literaturrecherche die Grundlage zur Umsetzung mittels einer eigens entwickelten Raytracing-Engine realisiert.
Die Ergebnisse zeigen, dass grundsätzliche eine Umsetzung der Sicht und Simulation des Auges mittels des Raytracing-Algorithmus möglich sind. Um eine allumfassende Simulation zur ganzheitlichen Abbildung der Realität zu erhalten ist allerdings eine sehr viel detailiertere Simulation nötig, welche in weiteren Arbeiten erarbeitet werden kann und hierbei Themen wie die exakte Simulation der Linse des Auges behandeln können.

Contents

1 Introduction

This chapter presents the motivation and aim of this work, the reasons behind the motivation for this work and what is ultimately to be achieved. A brief outline of the work is also presented.

1.1 Motivation

Ray tracing has long been the method of choice in the field of visualization and computer graphics when it comes to displaying realistic images and scenes. More recently, this topic has received increasing attention in computer graphics. This came about through various innovations in hardware acceleration through graphics processors and software support through special ray tracing render pipelines in interfaces such as DirectX and Vulkan. Thanks to these advances, it has become possible to significantly accelerate the calculation of ray-traced images and considerably increase the complexity of displaying realistic images. In context with this newly gained relevance and the possibility of realistically simulating ray optics with this technology, questions such as: "How does the human eye see?", "Can this be visualized with ray tracing?", "What optical effects can be represented on this basis?". These questions, with an open result, will be investigated in this master's thesis.

1.2 Goal of the Thesis

The aim of this master's thesis is to simulate the vision of the human eye. For this purpose, a graphical representation of the human lens system is used for the utilization of ray tracing. By using this technique and the associated simulation of ray optics, a realistic simulation of the optics of the eye should be made possible. Here, the pure image perception of the eye is to be simulated and effects that arise through the processing of optical impressions in the brain, such as optical illusions, are to be omitted. A model of the human eye will be created for the simulation, on

the basis of which optical effects can be implemented and simulated. This model should be able to simulate basic functions of the eye, such as accommodation, i.e. focusing on an object. This model should be realized based on existing, simple eye models.
The implementation of the optical effects is based on an explorative research approach. The model, which serves as the basis, as well as the effects are not predetermined at the beginning and arise during the processing of the work, so that optical effects can be selected as desired. This involves researching how individual effects affect an image or a scene. However, the focus here is on the realization of these effects, i.e. on the realistic recreation of them, rather than on the representation of a completely realistic image, which requires further effects such as realistic lighting, different shadows or others.

1.3 Outline

To answer these questions, they are addressed in five chapters in this Master's thesis. After the introduction, the second chapter, "Related Work", the research that has already been carried out on this or related and similar questions is reviewed and briefly explained. Chapter three, "Fundamentals", provides and explains the theoretical foundations that are necessary for understanding the other topics in this thesis. These topics include the basic ray tracing technique for displaying computer graphics, the eye model and its components, such as the lens and iris, as well as the visual optical effects that are simulated and displayed on the basis of the eye model and ray tracing technique explained. Chapter four of the thesis deals with the research method used and how the topics to be researched were implemented using this method.

The fifth chapter presents the results of the work, which were implemented using the presented method. The underlying eye model and its implementation are discussed, as well as the realization of the optical effects. It is shown how the individual effects and the model are created. The technology and mathematical models are also explained in detail. It is also shown how the various effects can be combined and how different overlays can affect a scene.

In the sixth chapter, the results are discussed and concludet. How was the eye model implemented? Which optical effects could be represented? What problems were encountered? The final chapter suggests ideas, possibilities and improvements for future work and provides a brief outlook.

2 Related Work

The topic of the realistic representation of images using ray tracing and the simulation of the human eye and human vision has already been addressed by several papers of other authors in the past.

In their paper, Greivenkamp et al.[GSMM95] describe how ray tracing and a schematic eye model can be used to simulate human vision and the visual representation of the eye. Furthermore, they attempt to recreate the so-called Stiles-Crawford effect on this basis.

Wei et al.[WPP14] present "Fast ray-tracing of human eye optics on Graphics Processing Units", a technique to simulate the formation of images on the retina of the human eye by simulating the large number of rays passing through the visual apparatus in three dimensions. To do this, they use a simulation consisting of polygonal meshes to enable a precise simulation of the visual apparatus, in contrast to other solutions that use a simplified analytical solution.

With the help of ISET3d software, Lian et al.[Lia+19] develop the method "Ray tracing 3D spectral scenes through human optics models", which contains calculations and measurements for modelling the ray path through the eye. To do this, they use several eye models and validate them. Based on these validations, they apply various physiological parameters to quantify their effect on the three-dimensional image formation.

In his book Wee[Wee22] writes about the representation of the eye as a reduced model. He also writes about the use of this reduced representation in connection with ray tracing.

"Real-time human vision rendering using blur distribution function" is a method, publicated by Tang et al. [TX15] that makes it possible to use a blur distribution function in real time and thus simulate a realistic impression of the view through a human eye. A neural network is used to calculate the "blur size" for each pixel. Based on a schematic eye model, a "depth of field" effect is simulated, which reproduces this characteristic of human vision.

Cholewiak et al. [Cho+17] present in their paper "Chromablur: rendering chromatic eye aberration improves accommodation and realism", a method that incorporates aberrations and thus generates screen images that are much closer to those that occur on the retina than those that occur in natural vision. By incorporating chromatic aberration effects, different effects are created in the retinal image, depending on the distance of an object viewed. The authors combine these effects in the "ChromaBlur" method.

The author Barsky [Bar04] describes the concept of vision realistic rendering in his paper "Vision-realistic rendering: simulation of the scanned foveal image from wavefront data of human subjects". Here, a so-called "object space point spread function" is calculated using real data and then mixed with input images in order to achieve a blurring of the images. The blurred input image is then referred to as vision-realistic.

Kakimoto et al. [KTMN07] describe a method called "Interactive Simulation of the Human Eye Depth of Field and Its Correction by Spectacle Lenses", a fast rendering algorithm for verifying the lens design of spectacles. The method simulates corrections of astigmatisms. This algorithm relies on ray tracing, uses and introduces the concept of "blur fields" to enable real-time calculations using pre-calculated information.

Mostafawy et al. [MKL97] describe in their paper "Virtual Eye: retinal image visualization of the human eye", a method that aims to simulate human vision by combining effects based on existing methods for motion blur, depth of field and a realistic camera model. This is an attempt to better achieve a "retinal image", to simulate the human eye optics.

"Real-time simulation and visualisation of human vision through eyeglasses on the GPU", is a publication by Niener et al.[NSG12] in which they present a method that enables real-time simulation through glasses or lenses. Here, so-called "wavefront tracing" is used for simulation on a GPU. Distributed ray tracing is also used for defocusing, or an approximation is used for real-time simulation. Furthermore, methods are presented to simulate astigmatism with the aim of carrying out quality assessments of special lenses.

3 Fundamentals

This chapter introduces and presents the fundamentals that form the basis for addressing the research question and implementing the methods. Firstly, the basics of how the ray tracing algorithm works, which serves as the information technology basis for the display, are explained. Furthermore, basic functions of how the eye works are explained and, finally, effects that are based on this functionality and can be implemented with it.

3.1 Raytracing

Ray tracing is an algorithm in computer graphics that makes it possible to display objects in a three-dimensional scene on two dimensional display. This method is based on sending rays from a viewpoint, called a camera, into a scene to determine which object in the scene can be seen from this viewpoint. The algorithm allows rays to be traced through a scene, making it suitable for simulating the light in a scene, as it allows a ray of light to be traced from its source, e.g. a lamp, to the screen [Gla07].

3.1.1 Basic Principle

The basic principle of the ray tracing algorithm is to emit a ray, usually determined by origin and direction, in the viewing direction. With these determinations, rays are sent into the scene across the entire sphere of viewing directions. After being sent out, the emitted ray is intersected with the polygons and other primitives such as spheres, which is done by an intersection test. If this intersection test is successful, this means that the ray has hit an object. The pixel with the position of the emitted ray now receives the color of the object that was successfully hit. If there is no object in the direction of the ray, it takes on a predetermined background color. If several objects are hit with one ray, the object with the smallest distance to the camera is selected. This is an inherent occlusion calculation, which has to be calculated separately with other methods, such as rasterization.

This process is carried out for all pixels of the image so that a color value is determined for each pixel of the image after the algorithm has been completed, resulting in an image [Gla07].

3.1.2 Simple Effects

The advantage of the ray tracing algorithm is that the rays behave very similarly to light rays and therefore are traceable in inverted direction. For this reason, it is suitable for simulating light effects that can be realized using the ray properties of light [Gla07].

Lighting

The illumination of objects in a scene is determined by following the path of the ray from the camera through the scene to the light source. Each reflection on the surface of an object, based on its material, absorbs part of the light and therefore less light reaches the camera. This results in a darker pixel and a weaker illuminated object. If a ray cannot reach a light source, it is considered an unlit pixel and is therefore colored black [Gla07].
Alternatively, local lighting methods such as the Phong lighting model can also be used to calculate the lighting. This model has the advantage that it allows a faster calculation of the lighting, but only considers the current object itself in the calculation and does not allow any influence from surrounding objects. [Gla07]

To calculate shadows, there are so-called shadow rays. These rays perform a test on a surface point of an object to check whether a point is hit by a light source or not. This is done by calculating a ray between the light source and the surface point and subjecting it to an intersection test with the objects in the scene. If there is an intersection with an object, the surface point is not hit by the light source and is therefore evaluated as a shadow. If the test is negative, the point is not in the shadow. [Gla07]

Reflection

One of the strengths of the ray tracing algorithm is the ability to calculate reflections. The algorithm inherently offers the possibility of calculating reflections on the surfaces of objects by calculating the path of the light through the scene. This is done by not ending a ray when it hits a surface, but also sending the reflected ray from there. The retransmission of the ray follows the laws of reflection:

$$\vec{R} = \vec{V} - 2(\vec{V} \times \vec{N})\vec{N} \tag{3.1}$$

[Gre06]

- $\vec{R}$ the reflected ray
- $\vec{V}$ the incident ray
- $\vec{N}$ the normal vector at the point of the object surface where the incident ray is reflected.

This law states that the incident ray is reflected at the normal vector in such a way that the angle of the incident ray and the normal is equal to the angle of the reflected ray and the normal.
When reflected at the surface, based on the material properties of the object, the light ray loses intensity as some of the light is absorbed. This means that the first reflection of a ray has a relatively large influence on its color value, but with each further reflection the relative influence of a light ray decreases.
Depending on the material, a light ray can be reflected from a surface in one (specular) or several (diffuse) directions, which can result in a reflective or rather rough surface.

Refraction

Like reflection, the refraction of light is also an effect that can be well represented by the ray tracing algorithm. This effect, which is often found on water surfaces or glass-like materials, is described by the so-called Snell's law of refraction. Refraction occurs when a light ray passes from one medium, e.g. air, into another medium, e.g. glass, which has a different refractive index. During this transition, the light ray is "bent" towards the perpendicular when entering from a optical less dense medium into a optical denser one, and away from the perpendicular in the opposite case.

The Snell's law of refraction that describes this says:

$$\frac{sin\Theta_1}{sin\Theta_2} = \frac{n_2}{n_1}. \tag{3.2}$$

Here, the angle Θ_1 is the angle between the incident light ray and the perpendicular (normal) of the surface of the object, and Θ_2 is the angle between the perpendicular and the incident ray.

To calculate the refraction formula on the surface of an object, it can be divided into a part running perpendicular to the normal and a part running parallel to the normal:

$$\vec{t} = \vec{t_{||}} + \vec{t_{\perp}} \tag{3.3}$$

[Gre06]
The parallel part is calculated by:

$$\vec{t_{||}} = \frac{\eta_1}{\eta_2}(\vec{i} + \cos\theta_i \vec{n}) \tag{3.4}$$

[Gre06]
The vertical part is calculated by:

$$\vec{t_{\perp}} = -\sqrt{1 - |\vec{t_{||}}|^2}\vec{n} \tag{3.5}$$

[Gre06]
Put together, this gives the resulting formula for calculating the refraction:

$$\vec{t} = \frac{\eta_1}{\eta_2}\vec{i} + (\frac{\eta_1}{\eta_2}\cos\theta_i - \sqrt{1 - \sin^2\theta_t})\vec{n} \tag{3.6}$$

[Gre06],
where:

$$\sin^2\theta_t = (\frac{\eta_1}{\eta_2})^2(1 - \cos^2\theta_i) \tag{3.7}$$

[Gre06]

- Here η_1 and η_2 are the refractive indices of the current and transition material

- $\vec{i}$ is the incident ray
- $\vec{n}$ is the normal
- θ_i is the angle between the incoming ray $\vec{i}$ and the normal $\vec{n}$

3.1.3 Problems

In addition to the advantages explained above, the ray tracing algorithm also has some disadvantages. The biggest disadvantage is the computing power required, which is due to the fact that a large number of rays have to be calculated for each pixel, which can sometimes be split into many more rays due to various effects. The complexity of the algorithm is therefore estimated with a complexity of $\mathcal{O}(n)$. In complex scenes, this means that images can take hours or even days to finish rendering, making it difficult or impossible to use ray tracing for real-time applications in particular.

Solution Approaches

There are a variety of solution approaches that address the problem of high complexity and the associated computing time requirements.
Simple methods for reducing computing power include limiting the depth of light, which limits the number of new rays emitted per pixel. Although this reduces the level of detail of reflections and refractions, the significance of these becomes less important in most applications after the first few iterations. Reducing the image resolution can also bring improvements, but even a small reduction leads to a visible reduction in image quality.
Advanced optimization methods include data structures such as Bounding Volume Hierarchies (BVH), which enable intersection tests of a ray with the smallest possible number of objects in a scene. Without optimizations, all objects in a scene must always be checked for an intersection with a ray, which means that the number of tests increases very quickly due to a high resolution and a large number of objects. BVH attempts to minimize the number of intersection tests with the help of a series of data structure optimizations, from simple collision structures such as spheres surrounding a complex object to more complex data structures such as KD-Tress. These are arranged hierarchically, whereby simple but coarse tests are performed first and complex but more precise tests only after a successful intersection with

them.
Another optimization option is sampling. [HH11]

3.1.4 Sampling

Sampling is a method based on the creation of samples of a scene that are taken as representative for the approximation of values of the scene.
This is important for ray tracing, on the one hand to reduce the required computing power by reducing the accuracy, or on the other hand to enable the calculation of some values in the first place, since a discretization of continuous values takes place through sampling. A frequently encountered example of this in ray tracing is the so-called Monte Carlo integration, which is often used in ray tracing to solve the so-called rendering equation, which describes the illumination for a point by an integral that can only be solved numerically. By sampling with random values, this integral can be approximated in finite time on the basis of the "law of large numbers" and thus the illumination for a point can be approximated.
However, sampling also has disadvantages due to approximation instead of exact calculation. These can be very diverse, but in ray tracing they often manifest themselves in the form of artifacts and noise in the image. To reduce these image errors, methods are also being developed, such as ever better sampling methods, e.g. Importance Sampling, or filters that filter or smooth out the errors that occur. Neural networks are also frequently used in very modern sampling methods.

3.2 Eye-Model

The human eye is a complex organ consisting of a multitude of structures, layers, tissue types and much more. This complexity makes correct modeling of the eye very difficult . For this reason, there are schematic models of the eye that attempt to reduce the complexity to the essentials. In this work, the focus is on the modeling of the lens, the retina and the correct accommodation, i.e. the focusing of the image on the retina and the associated adaptation of the lens.

3.2.1 Light Refraction in the Eye

Light passes through several layers in the eye, causing it to be refracted several times on its way to the retina. These layers are primarily the cornea, the lens, the vitreous body and the eye chamber. Each of these layers refracts the light to a greater or lesser extent. Due to the different materials of each layer, they have different refractive indices, which contribute to the overall refraction of light up to the retina.

In the past, this refraction through several layers was converted into a total refractive power, which contributes to the approximation and thus the reduction of the complexity of the refractions through several layers. In his paper, H. Gernet has calculated a fictitious total refractive index of the eye, which can summarize and generalize these layers under certain standardized conditions. This total refractive index amounts to a total refractive power of 61.7 dpt (diopters).[Ger64]

3.2.2 Lens

The lens is mainly responsible for the refraction or the change in the refraction of light in the eye. In contrast to the other layers, the lens has the ability to adjust the refraction. This is done by changing the thickness of the lens by contracting muscles. By changing the thickness and shape of the lens, the eye is able to adjust the path of the light ray through the lens. This is used to perform so-called accommodation, which ensures that the image of an object can be focused on the retina and thus adapted to different situations or viewing distances.

3.2.3 Types of Lenses

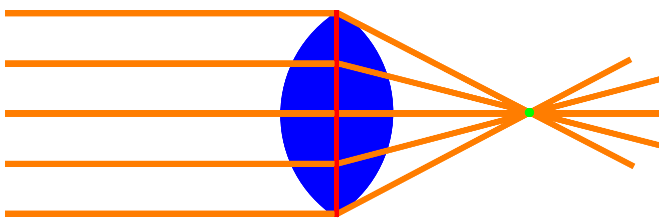

Figure 3.1: This figure shows a convex lens that refracts the light rays in the direction of the perpendicular at the lens plane (red). With this type of lens, the light rays gather at a focal point (green).

There are 2 main types of lenses, convex and concave lenses. Convex lenses, also known as converging lenses, are lenses that have an outwardly curved surface. There are lenses which either have only one convex side and those where both sides are convex. Lenses that are convex on both sides are called biconvex.[Ser98] In a convex lens, the course of the light rays, starting from parallel light, is such that the rays are refracted at the lens plane so that they run towards the perpendicular of the lens. (Figure 3.2)

These lenses have the property that light rays passing through them converge behind the lens. A special case of these lenses are those that have a point where all the rays meet. This point is known as the focal point.

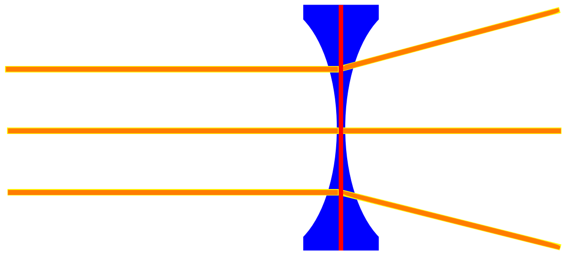

Figure 3.2: This figure shows a concave lens which refracts the light rays away from the perpendicular of the lens at the lens plane (red).

The second type of lenses are concave lenses. The sides of these lenses are curved inwards. As with convex lenses there are planconcave lenses with only one concave side or biconcave lenses with two. With concave lenses, the course of light rays striking the lens in parallel is such that they are refracted at the lens surface in such a way that the rays run away from the perpendicular of the lens, so that no focal point can arise as with a convex lens.

3.2.4 The Human Eye

The shape of the lens of the human eye largely corresponds to the shape of a so-called rotational ellipsoid [source]. However, the lens is not symmetrical; the curvature of the outer side facing away from the retina is relatively pronounced, while the inner side is rather flat.

Due to the curvature of the lens of the human eye on both sides, it is a so-called biconvex or converging lens. A converging lens is characterized by the fact that it refracts the light in such a way that all incoming rays meet or collect in one point behind the lens.

The point at which the rays converge is called the focal point (F). Focal points are found on both sides of the lens, with one focal point located between the so-called object plane (G) and the lens, outside the eye, and another focal point (F') located between the lens and the image plane (B). The object plane can be seen as the plane on which an object to be focused is located. The image plane is the counterpart to

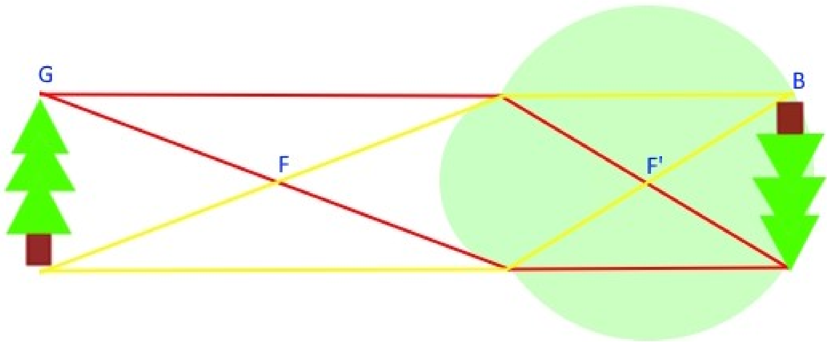

Figure 3.3: Visualization of the ray path from the object plane (G), through the focal point (F) on the left side of the lens and on the right side (F') of the lens and the image plane on the retina (B).

the object plane; it represents the plane at which the ray bundles (Figure 3.2), which create the object plane. This plane formed by the ray bundles has the property that the bundles on it form a sharp image of the object of the object plane [Ser98].

The eye uses the properties of the object and image plane to produce a sharp image on the retina. The adjustable lens allows the eye to focus on objects at different distances.

It is possible to calculate the focal length, object plane and image plane of a lens, if two of three of theses properties are given. This requires information about both the lens and the situation of the scene.

The so-called lens equation is used to calculate these values. This equation describes the relationship between focal length, object plane and image plane.

$$\frac{1}{f} = \frac{1}{g} + \frac{1}{b} \tag{3.8}$$

[Ser98]

The aim of the eye as a lens system is to be able to focus on an object at any distance. This means that the focal length and the image plane must be adapted to any object plane. If the lens system adapts to an object distance, either the lens and thus the focal length or the image distance must be adjusted. As the eye is an unchangeable body and the size and therefore the image width cannot be adjusted, the focal length of the lens must be adjusted. This is done in the eye, as described in the "Lens" chapter, by adjusting the thickness and shape of the lens.

Lens makers equation

The so-called Lensmaker's formula is a formula that can be used to calculate a focal point for a given lens. This formula uses the radii of curvature and the refractive index of the lens material to calculate the focal length of the given lens [GME19].
The complete formula for the calculation is:

$$D = \frac{1}{f} = \frac{n - n_0}{n_0}\left(\frac{1}{R_1} - \frac{1}{R_2} + \frac{(n - n_0)d}{nR_1R_2}\right). \tag{3.9}$$

The variables required to calculate the focal length are:

- R_1 and R_2, which determine the so-called spherical radii of the lens.
- d which determines the thickness of the lens.
- n_0 the refractive index which determines the medium that lies outside the lens.
- n the refractive index of the lens material.
- f f ist the focal length of the lens to be calculated.
- D is the refractive power of the lens, which is measured in units of diopters.

3.2.5 Light Regulation

The light entering the eye must be regulated in order to protect the retina from damage and allow it to function correctly. This function is performed in the eye by the iris, which is a circular body that contracts at high brightness and thus creates the pupil, a smaller opening to allow less light to fall on the retina and thus protect it. At low light levels, when less light falls on the retina and the retina cannot see properly, the iris opens and so enlarges the pupil, to allow more light to enter the eye.

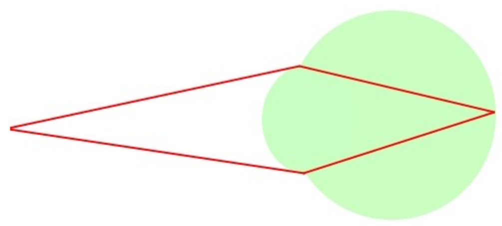

Figure 3.4: Representation of the ray bundle that runs conically across the lens to a focal point on the retina.

Adaption

The process of adapting the eye to the prevailing amount of light is known as adaptation. This takes place on two levels of the eye: the pupil and the retina.

Adaptation processes also take place on the retina. The receptors on the retina have the ability to change their sensitivity depending on the light intensity. Dark adaptation, i.e. the adaptation of the retina to dark light conditions, is a slow process that can take around 10 minutes for cones and up to 30 minutes for rods. Light adaptation, on the other hand, is faster and takes around 6 minutes.

3.3 Visual Effects

There are a variety of visual effects that are caused by the structure, materials or function of the components of the eye. These effects can be both non-critical optical phenomena and more critical ones such as diseases.

3.3.1 Myopia and Hyperopia

Myopia and hyperopia, also known colloquially as short-sightedness and long-sightedness, are forms of defective vision that occur when an object does not focus correctly on the retina. This occurs because the focal length of the lens does not match the distance between lens and retina.

In the case of myopia, this results in the image plane lying in front of the retina despite the minimal curvature of the lens and thus the maximum adaptation distance of the eye, resulting in a blurred image.

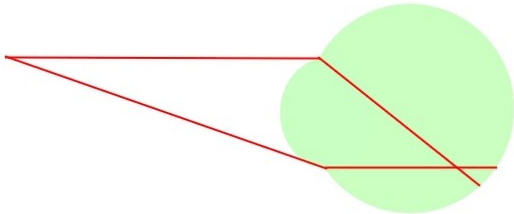

Figure 3.5: Representation of the ray bundle that runs conically across the lens to a focal point on the retina.

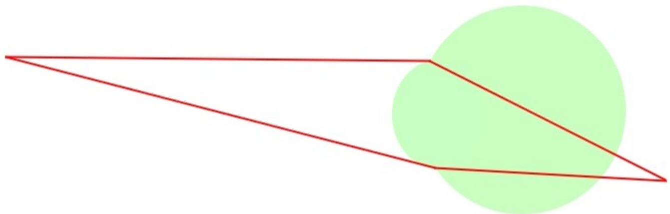

Figure 3.6: Representation of the ray bundle that runs conically across the lens to a focal point on the retina.

The opposite is the case with hyperopia. In this case, the image plane lies behind the retina, resulting in a blurred image.

Correction

There are correction methods for this visual defect in the form of lenses, e.g. glasses or contanct lenses. These lenses correct the path of the rays of light in front of the eye so that the wrong image planes are corrected.
In the case of myopia, a lens is used that places the image distance further away from the lens. This is usually achieved with a concave lens, which refracts the light in front of the eye in such a way that the light rays diverge in front of the lens and thus push the focal point backwards so that it lies on the retina.
In the case of hyperopia, on the other hand, a lens is used that places the image width closer to the lens, which can be achieved with a convex lens. In this case, the light rays are focused by the convex lens in front of the eye lens and the focal point is placed on the retina.

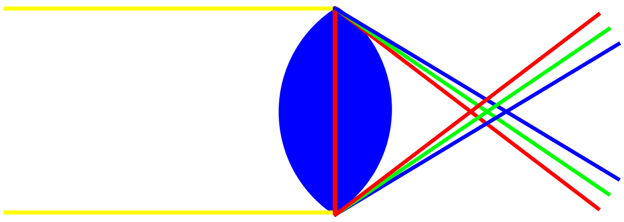

Figure 3.7: This figure shows the division of a white (yellow) light ray at a convex lens into red, green and blue light rays, which are refracted with different strengths.

3.3.2 Chromatic Aberration

Chromatic aberration is an optical defect that occurs in optical lenses or lens systems. This effect is caused by the fact that different wavelengths of light, i.e. colors, are refracted differently in a medium. This results in a chromatic magnification difference, which causes different wavelengths of light to be magnified to different degrees. On the one hand, this can result in the so-called longitudinal chromatic aberration, which leads to a deviation of the colors in front and behind the focal point of the lens. In addition to the longitudinal chromatic aberration, there is also the so-called transverse chromatic aberration, which will leave color edges visible at object edges.

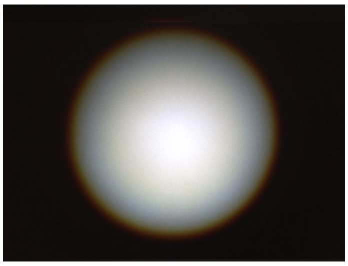

Figure 3.8: Chromatic aberration on flashlight cone[Fer24].

The effect of chromatic aberration is due to so-called dispersion. The greater the dispersion of a material, the greater the difference in the refraction of different wavelengths.

The strength of the chromatic aberration of a material is measured in the so-called Abbe number. The higher the value of this number, the lower the so-called dispersion number, i.e. the strength of the change in the refractive index depending on the wavelength of the light. The Abbe number has been empirically determined for various materials by different organizations and recorded in tables.

3.3.3 Photopic, Scotopic, Mesopic Vision

The eye sees differently at different levels of brightness. A basic distinction is made between so-called day vision, photopic vision, and night vision, scotopic vision. Twilight vision, also known as mesopic vision, lies between day and night vision.

The different brightness levels have an effect on the light receptors located on the retina, which are located on the eye. These receptors are divided into cones, which are responsible for color vision and are mainly active in photopic vision, and rods, which are mainly sensitive to brightness and are active in scotopic vision. In mesopic vision, a combination of rods and cones is active, depending on the level of brightness.

As a result, it is possible to see the colors red, green and blue at full brightness, as this is made possible by the rods. At low brightness, the proportion of red in particular decreases, resulting in a green-blue image. This effect is known as the Purkinje effect. In mesopic vision, the red value slowly decreases with decreasing brightness. [BAR57]

V-Lambda Curve

The behavior of cones and rods in bright light is modeled for photopic vision in the so-called V-Lambda curve, also known as the relative spectral luminosity. This curve was published by the CIE, the International Commission on Illumination, in the "International Standard Observer" in 1983. The V-Lambda curve is most pronounced at wavelength values around 560nm. This means that a resultant to which this curve has been applied has a strong red-green cast [BIP19].

Scotopic vision is modeled in the so-called V'-Lambda curve. These curves model the strength of the respective wavelengths in bright and dark vision. A resulting image from this curve has a strong green-blue cast, as color values around 510nm are weighted the strongest.

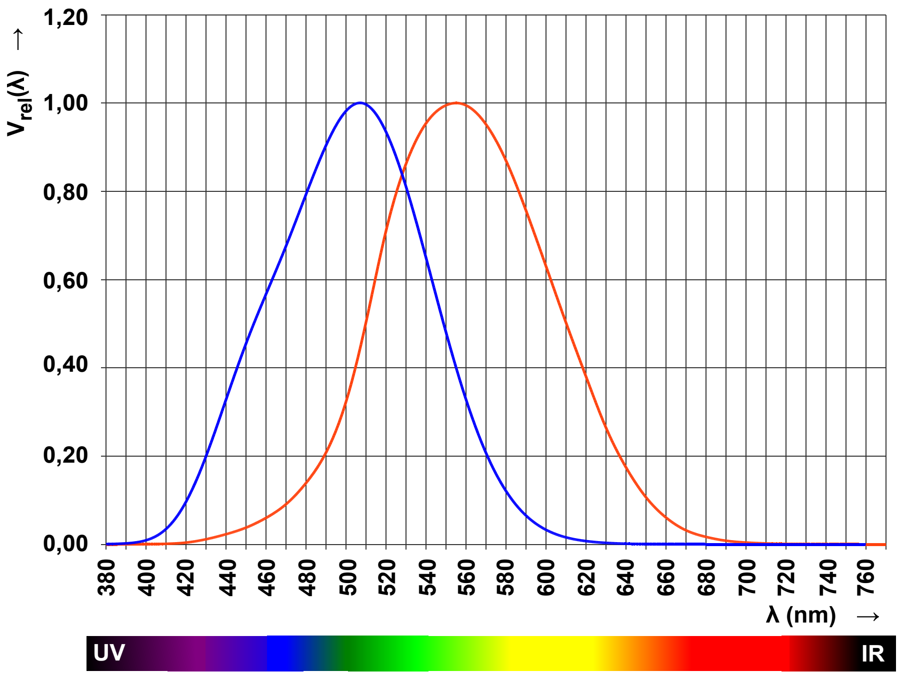

Figure 3.9: This curve shows the V-lambda (red) and V′-lambda(blue) curves. On the X-axis are the respective wavelengths, on the Y-axis the resulting function values as a weighting of the respective wavelength value [HHa10].

The mesopic view is represented by interpolating between the V-lambda and V′-lambda curves. This is achieved with the so-called "mesopic luminosity function":

$$V_m(\lambda) = (1 - x)V'(\lambda) + xV(\lambda). \tag{3.10}$$

[BIP19]

- λ is the wavelength
- x is a luminance value

4 Method

The methods chapter describes the procedures and algorithms for implementing the theoretical principles that have already been developed. Ways of implementation are sought and their solutions described. At the beginning, the problem is defined in concrete terms and the method for researching and validating the results is described. The software environment in which these methods are practically implemented is also briefly explained. Finally, the specific methods are developed and presented.

4.1 Problem Definition

Ray tracing is a very broad field of research, which has also been explored in various ways with a focus on the human eye, as shown in the Related Work chapter. Since research in this area is very diverse and there are countless effects that can be realized and researched on the basis of a virtual eye, it makes sense to design the research method exploratively in order to have a wide range of possible effects available. Furthermore, the underlying eye model was created as part of this work, which means that possible complications and limitations that arise during processing are not known in advance and are therefore difficult to assess. Exploratory research allows the goals to be adapted during processing and is therefore the appropriate method for this work to answer the question "How does the human eye see, can this be visualized with the help of ray tracing and which optical effects can be represented on this basis?".

4.2 Method and Validation

The methodology, which should lead to the processing and answering of the problem, is the development of methods and functions on the basis of literature research. Based on the researched literature and the resulting findings, methods are developed that lead to the solution of the problem. Due to the developmental nature of this work, this research work is classified as qualitative research. Once a function has been developed, it is compared with reference values in the form of real images and thus validated. If no real-world data or other data is available to check the plausibility of the results, an attempt is made to check the results using theoretical validations.

4.3 Application

The methods are applied to a raytracing system that was developed to simulate the eye. The raytracing system is a system that enables development on both CPU and GPU basis using C++ and OpenGL. At the start of development, the system allows the description of scenes using 3D mesh data and other simple geometry using explicit representations such as spheres based on the general spherical equation. The development on the GPU, which is mainly used to accelerate the calculation, does not fully utilize the OpenGL rasterization graphics pipeline, but works mainly in its last stage, the fragment shader, as this is the only stage that works at pixel level and is therefore most suitable for calculating ray tracing scenes. For this reason, the entire scene is described within a fragment shader program. In this system, the developed methods are combined into a scene and tested.

4.4 Method Realization

The functionality of a simulation of the eye and its quality and precision depend heavily on the chosen type of implementation of its principles and methods. It is necessary to weigh up which aspects of the simulation are of central importance and whether and where compromises are permissible, which allow scope for practical aspects such as performance and usability. This chapter explains which methods are implemented in which way and with which precision in this thesis.

4.4.1 Eye Model

The basis of all effects is the simulation of the eye. For this purpose, a schematic model is used as a template, which is transferred in parts to the ray tracing environment and realized in software. The central aspect here is the modeling of a lens that fulfills the properties and criteria that an eye lens requires. The central criteria here are the ability to focus an image, i.e. to form a focal point, so that a clear image plane can be created for an object plane. Furthermore, the thickness and shape of the lens must be adjustable and adaptable so that the simulated eye can react to a changed object plane and adjust the focal point and thus the image plane accordingly. Exceptions must also be made in order to realize the schematic model on the screen, such as adapting the curved retina to a straight screen.

4.4.2 Lens

In addition to the conditions required to simulate the eye, i.e. above all the adjustment of the refractive power by the thickness and shape of the lens, the lens for simulating the eye must fulfill further criteria in order to be suitable for use with ray tracing. Therefore, the central criteria for the lens are:

1. The possibility of adjusting the refractive power by thickness
2. Forming a focal point and an image plane
3. A feasible calculation of the intersection point with a ray
4. Advantageous: Low computational costs when calculating the intersection point

There are several ways to describe objects and surfaces mathematically, e.g. explicitly, implicitly, using a mesh consisting of primitives such as a triangle mesh, spline surfaces and many more. Probably the most common method in 3D graphics is the creation of primitives from triangle meshes. In addition to meshes, ray tracing software can also be used to display explicitly defined objects, as in many cases these allow a comparatively simple calculation of the intersection test. Due to the availability of these two display methods in the existing ray tracing software, this work was limited to these.

4.4.3 Comparison of Visualization Methods

When deciding which type of lens modeling is the most suitable for modeling the eye lens, the strengths and weaknesses of the individual display variants must be discussed and compared. To this end, this chapter compares the possibilities of the "triangle mesh" lens and the "sphere surface lens".

Triangle Mesh

Modeling and displaying a lens with a triangle mesh requires several steps. A way of modeling the lens is needed. A great advantage here is that the modeling possibilities are widely developed by external software, such as CAD or modeling programs, and thus the modeling of different lens shapes and types using triangle mesh is very well possible. The representation of the triangle mesh works by performing intersection tests with each triangle that belongs to this mesh. The resulting representation is a three-dimensional object composed of triangles. However, this composition also creates an object that cannot represent curves. Since curves are a central component of most lenses, this poses a problem. There are two main approaches to solving this problem. One is to increase the resolution by increasing the number of triangles in the mesh, which, with a sufficient number and correct orientation, get closer and closer to a rounding until the difference to the desired curve is sufficiently small. This approach has the advantage that its implementation and application does not require any further adaptation of the program. The problem here is the large increase in the number of triangles, which can result in a considerably longer calculation time.

The second approach is shading. Here, a round surface is simulated by changing the surface normal. An established method here is so-called smooth shading, which is achieved by interpolating the surface normals using the normals at the corner points of the triangle, the so-called vertex normals. This assumes that the vertex normals represent an average value of the surface normals of the adjacent surfaces. If this condition is met, a surface normal can be interpolated between the three vertices for any point x on the surface of a triangle. Barycentric interpolation is used to calculate an interpolation between three points. The interpolation is calculated for a point x as follows:

The barycentric interpolation can be calculated in two dimensions because we know that the point x lies on the surface of the triangle and no depth value is required.

$$\begin{aligned}
\lambda_1 &= \frac{[(y_2 - y_3)(x - x_3) + (x_3 - x_2)(y - y_3)]}{[(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)]} \\
\lambda_2 &= \frac{[(y_3 - y_1)(x - x_3) + (x_1 - x_3)(y - y_3)]}{[(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)]} \\
\lambda_3 &= 1 - \lambda_1 - \lambda_2
\end{aligned} \tag{4.1}$$

- $x_{1,2,3}, y_{1,2,3}$ are the 2D coordinates of the vertices of the triangle
- $\lambda_{1,2,3}$ are the resulting weightings

These formulas calculate the weights of the respective normals of the vertex normals λ_1, λ_2 and λ_3, based on their proximity to point x with their 3D coordinates x, y, z. After the calculation, these weights are applied to the vertex normals, resulting in the new surface normal at point x:

$$\vec{N}_x = \lambda_1 \times v\vec{N}_1 + \lambda_2 \times v\vec{N}_2 + \lambda_3 \times v\vec{N}_3 \tag{4.2}$$

By applying this interpolation to the triangular mesh, a surface is obtained that has a round surface normal despite the "angular" triangular surface. Since only the surface normal and not the geometry itself is required to display the illumination and calculate the refraction at a surface point x, this is a method that makes it possible to create a round lens surface. Depending on the degree of resolution, this method even makes it possible to reduce it. The problem with interpolation is the possible reduction in accuracy, as the actual intersection point, which should lie on the surface of a true rounding, lies on the surface of the triangle, i.e. the true geometry, during interpolation and thus induces an error. This is not a major problem for the pure representation of the lens with a round surface, but can lead to problems with accuracy in subsequent calculations. A central aspect that is of great importance for the lens of the eye is the adaptability of the sizes and shape, especially their thickness. In principle, a triangular mesh offers good possibilities here, as the mesh can be changed and transformed at each vertex. In addition, the thickness of a meshed lens can be transformed and adjusted using transformation matrices, which can be applied to all vertices.

Triangle Mesh Problems

The representation of the lens using a triangular mesh causes some problems. The transformation of the mesh is well feasible. The problem here is that the thickness of the lens with a transformation matrix does not change uniformly when transformed in only one direction. As a result, the shape of the lens also changes when the thickness is changed, which can lead to undesirable consequences such as the loss of the focal point and thus the image plane. It is also not possible to precisely change the thickness simply by changing simple parameters, but must always be converted to the vertices. Another problem, which has already been briefly described, is the error caused by the barycentric interpolation on the surface of the lens. The difference between the collision point of the intersection test and the collision point that the lens should actually have results in an inaccuracy or error. As a result of this error, the ray path through the lens deviates slightly from the expected path, which in turn means that the focal point of the lens can also form, but at a different location to the expected focal point. This makes it very difficult or virtually impossible to calculate.

Display errors at the edges of the lens can occur with triangle meshes due to an interpolation problem with the normals of vertices that have strongly deviating spatial directions. In this case, the interpolation can lead to a deviation in the normal orientation of up to 90° and thus to a strongly deviating refraction calculation. A significant problem for ray tracing software is the relatively high computational requirement of the triangle mesh, which is caused by the relatively computationally intensive triangle intersection test and is also amplified by a high number of triangles. Although a high number of triangles can be reduced by using barycentric interpolation, as explained above, this also increases the previously discussed error.

Sphere Surface Lens

Another way of realizing the lens surface, in addition to the triangular meshes, is the combination of implicit spherical surfaces, linked with further conditions. This type of lens is called a spherical lens. Here, the intersection of two spheres forms the lens surface. The spheres are defined by their centers M_1 and M_2, as well as their radii R_1 and R_2. The spheres form a lens if the sum of their centers is smaller than the distance between the sum of their radii, so that:

$$|M_1 - M_2| < R_1 + R_2. \tag{4.3}$$

This means that for each point within the lens L applies:

$$L = \{P \in \mathbb{R}^3 \mid (P - M_1)^2 \leq R_1^2 \cap (P - M_2)^2 \leq R_2^2\}. \tag{4.4}$$

The lens defined in this way can be intersected with a vector to make it representable for the ray tracing algorithm. This works by intersecting both spheres of the lens with the vector and, if the intersection test is successful, checking the resulting point for the condition of belonging to the set of points of the lens.

The normal is required to use the lens to refract light. The calculation of this for a point x works for a sphere by subtracting the point coordinates from those of the center M in the form $\overrightarrow{x - M}$. The representation by the intersection of two spheres offers some advantages. The intersection test between vector, e.g. light ray, and lens, is comparatively inexpensive to calculate, consisting exclusively of two tests with spheres and the query for the membership of the set, and is therefore suitable for use with a ray tracing algorithm. Due to the direct determination of the surface of the lens using an implicit function, the precision of this is high. The problem with this representation, however, is that it is not always stable in terms of viewing angle and therefore errors can occur during viewing angle transformations, which can be caused by the fact that, depending on the viewing angle, not both spheres are hit in marginal cases, although they should actually be for the correct representation of a lens shape and thus a part of the lens is not displayed.

Alternative Sphere Surface Lens

In order to solve the problems with the viewing angle stability, an alternative representation of the lens is required which, however, retains the advantages of the sphere surface lens as far as possible. Another way of constructing the lens

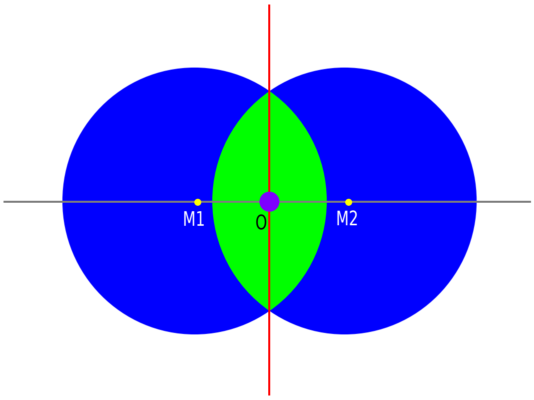

Figure 4.1: This figure shows the construction of the lens using a parting plane. The vertical line represents the dividing plane, the horizontal line, which runs perpendicular to the dividing plane, shows the plane on which both spheres must lie and have their origin. The yellow dots are the origins of the spheres (M_1, M_2), which lie on the horizontal line, the green area in the middle of the two spheres is the resulting lens. The purple point in the center of the green marked area is the origin point (O) of the lens.

other than via the intersection of the spheres is to place a separating plane between the spheres, which defines which part of a sphere should be displayed and which should not.

To form a lens from this, two spheres must lie on the same plane (Figure 4.1). This plane must lie in space in such a way that the normal of the separating plane and a line that could connect the origins of the two spheres are parallel to each other. The plane can be used to determine which part of a sphere should belong to the lens. If the sphere has its origin on the left side of the plane, only right-sided parts are drawn, and vice versa for the right-sided sphere. In this way, a lens can be constructed from two spherical caps that has no problems with viewing angle stability.
To calculate the refraction, the surface normal must also be calculated for this lens. As described for the simple "Sphere Surface Lens", this can be calculated by subtracting the collision point x on the surface from the center M_1 or M_2 of the sphere. The challenge here is to find the correct sphere for a surface point to calculate the normal.
This problem is solved by checking whether the angle between the distance of the point x and the center M_1 or M_2 and the distance of the lens origin O and the sphere centers M_1 or M_2 is less than $\frac{\pi}{2}$:

$$\arccos\left(\frac{\vec{OM}_{1/2}}{\|\vec{OM}_{1/2}\|}\cdot\frac{\vec{Ox}}{\|\vec{Ox}\|}\right) < \frac{\pi}{2}. \tag{4.5}$$

Depending on the side on which the angle is less than $\frac{\pi}{2}$, the normal is used. The position of the lens in space can be changed using the origin O. This is possible because the sphere origins M_1, M_2 are positioned relative to the position of the lens origin by keeping a constant distance of half the lens thickness d from the lens origin. The orientation of the lens in space is determined by the transformation of the parting plane. Changing its orientation also changes the orientation of the distance $\vec{M_1M_2}$, which must always be oriented parallel to the normal of the parting plane in space.

4.4.4 Choice of Lense Construction

The choice of the type of lens construction for modeling the eye lens depends on several factors. The triangle mesh lens has great advantages, especially in the initial design using 3D rendering software. The possibility of modeling using so-called CAD programs is very precise. Subsequent adaptation of a lens to any shape is also quick and easy. On the one hand, there are disadvantages associated with this type of design, especially precision problems due to interpolation and the high computational effort (see chapter Triangle Mesh Problems). On the other hand, there is the sphere-surface lens, which allows precise mapping of a spherical lens through implicit representation and the introduction of a separating plane. Due to the comparatively simple and therefore fast tests, the lens can be calculated very quickly for ray tracing software. The high precision minimizes errors in future calculations based on the lens. A disadvantage is the increased design effort required to change the lens, as although it is easy to adjust the lens thickness, subsequent adjustment of the shape is complex or even impossible without adapting or changing the basic type of lens calculation.

Due to the easy adaptability of the "Triangle Mesh Lens", it was used for the initial modeling and first functional tests of the eye. However, due to the increased speed and precision, the "Alternative Sphere Surface Lens" was finally chosen for the construction of the eye model.

4.5 Modelling the Eye Function

The aim of the eye is to refract incoming light in such a way that it forms an image plane on the retina which can be perceived as a sharp image. In order to model this function, a replica of the eye model is required, which corresponds to it primarily in terms of function. This requires a lens that can perform accommodation, i.e. changing the lens to adjust the refraction of light and thus the focal point and the image plane, as well as a plane that fulfills the function of the retina.

4.5.1 Accommodation

The central function of the lens is the correct refraction of light, i.e. the focusing of a given object plane by adapting the refraction to a given image plane or the retina by adjusting the shape and thickness of the lens. This adaptation of the lens is called accommodation (see chapter 3.2.4 The human eye). As described in chapter 4.4.4, the "Alternative Sphere Surface Lens" is used in this project, which has a parameter that specifies and determines the thickness of the lens. This is relevant because the lens of the eye has to adjust the refraction by changing the thickness due to the fixed size of the eye.

In order to calculate the appropriate thickness of the lens for a targeted object, i.e. any object plane, the required focal length must be calculated. The first step is to determine the appropriate distance between the lens and the object plane. This is done by emitting a central ray from the position of the lens in the direction of the object to be focused. An intersection test is carried out with this ray, as with the calculation of the rays of the ray tracing algorithm. If this test is successful, the object hit is assumed to be an object on the object plane. In order to calculate the value g for this object, the distance between the lens and the object must be calculated. This is achieved by measuring the length of the ray *dist* that was emitted to find an object after the successful test. This length can be used as the value for g.

$$g = \sqrt{dist_x^2 + dist_y^2 + dist_z^2} \tag{4.6}$$

This results from the so-called lens equation:

$$\frac{1}{f} = \frac{1}{g} + \frac{1}{b}. \tag{4.7}$$

This formula can be used to calculate the focal point for the given image plane b and the given object plane g. A second formula is required to calculate the focal point for a lens. This can be achieved with the so-called "Lens-Makers-Equation":

$$\frac{1}{f} = \frac{n - n_0}{n_0}\left(\frac{1}{R_1} - \frac{1}{R_2} + \frac{(n - n_0)d}{nR_1R_2}\right) \tag{4.8}$$

[GME19]

In order to calculate the required thickness of the lens from the two given formulas, both expressions must be equated, which is possible because $\frac{1}{f}$ results in:

$$\frac{1}{g} + \frac{1}{b} = \frac{n - n_0}{n_0}\left(\frac{1}{R_1} - \frac{1}{R_2} + \frac{(n - n_0)d}{nR_1R_2}\right). \tag{4.9}$$

With this formula, it is possible to calculate all lens parameters for a required object or image plane by means of transformation. In this case, the aim is to calculate the thickness of the lens, which is represented by the parameter d in this expression. By transforming to d you get the expression:

As the medium n_0 outside the lens is always air with a refractive index of 1, this value is set to $n_0 = 1$ in this expression.

$$d = \frac{n(bg(n-1)(R_1 - R_2) + bR_1R_2 + gR_1R_2)}{bg(n-1)^2} \tag{4.10}$$

- R_1, R_2 are the radii of the left- and right-sided spheres

- n is the refractive index of the lens

The resulting expression can be used to calculate the required thickness of the lens. This value can be applied to the "Alternative Sphere Surface Lens" and thus the distances of the lens center M_1 and M_2 can be adapted to the new lens thickness and thus accommodation can be achieved.

4.5.2 Retina

The part of the eye that receives light and transmits this information to the brain to display an image is the retina. This function is performed by the screen in an eye simulation, which aims to simulate its function and vision. The screen forms the plane on which the image plane b must be formed during accommodation in order to produce a sharp image of an object on the object plane. On the retina, the light rays hit the retina in the form of bundles of rays, where they are perceived by receptors.

Simulation of the Retina

In the simulation, the retina is mapped and replaced by the screen as the "information receiver", as it also receives the light rays in the scene and ultimately generates the image for further processing.
In order to simulate the behavior of the retina, the behavior of the incidence of light on the retina must be simulated. Here it is relevant to follow the course of the rays from the seen, focused object through the eye to the retina. As described in chapter 3.2.4, this happens in the eye in a conical course. Starting on the object plane at a point, which spreads out towards the lens and then meets at a point in the eye.

The conic section describes the space in which the light rays can move to create a sharp point on the retina. This means that a point on the retina can be hit by a large number of different light rays that move within the limit of the conical section.

In order to simulate the behavior in a ray tracing algorithm, some adjustments are required to make it calculable. A fundamental difference between the ray tracing algorithm and reality is the direction in which the light travels- or is viewed. In reality, light rays always originate from a light source and some of these rays hit the retina in the eye, creating an image. In the ray tracing algorithm, this is reversed in order to reduce the number of light rays and only trace and calculate the relevant light rays that are important for the representation (see chapter 3.1.2). This is possible because the progression of light through a room or scene is an inverse problem. This means that the retina in the simulation ultimately does not "receive" light rays, but "emits" them to calculate the illumination for a point. Since the simulation ultimately has to calculate the pixel color on the screen, the pixels are regarded as the color receptors of the simulated eye. For this purpose, it is relevant to know the position of each pixel in three-dimensional space in order to be able to use it as the origin of a light ray. To determine the position of a pixel in space, it is relevant to know the position of the eye in space. Based on the eye position, the position of a pixel can be calculated as follows:

Each pixel of the screen must be projected into space. For this purpose, a point representing the bottom left corner of the image in three-dimensional space is defined. From this point, the other pixels of the image are calculated by defining the directions of the image horizontally and vertically. These image directions are given with a value $picture_{width}$ ranging between 0 and MaxHeight, or value $picture_{height}$ ranging between 0 and MaxWidth. The pixels of the image move within these value ranges and are scaled along the horizontal and vertical directions using these values.

The directions are calculated as follows:

$$\overrightarrow{horizontal} = picture\_width \times \vec{u} \tag{4.11}$$

$$\overrightarrow{vertical} = picture\_height \times \vec{v} \tag{4.12}$$

$$\overrightarrow{lower\_left\_corner} = \overrightarrow{camera\_position} - \frac{\overrightarrow{horizontal}}{2} - \frac{\overrightarrow{vertical}}{2} - \vec{w}. \tag{4.13}$$

The parameter $\vec{w}$ shows the direction in which the camera is aligned. These values can be used to determine the position of a pixel in space. Using the length and width values s and t of a pixel on the screen, the position can be determined using the position of the "$\overrightarrow{lower\_left\_corner}$":

$$\overrightarrow{ray\_origin} = \left(\overrightarrow{lower\_left\_corner} + s \cdot \overrightarrow{horizontal} + t \cdot \overrightarrow{vertical}\right) \tag{4.14}$$

The origin of the ray is calculated in this way. The second part required for a ray is its spatial direction. This is calculated by subtracting the origin from a point in space. A reference point is required to calculate this direction. For this purpose, a so-called sampling point is used, which is located at a point relevant for the spatial direction calculation, usually on the lens (see chapter Sampling):

$$\overrightarrow{ray\_direction} = sampling\_point - \overrightarrow{ray\_origin} \tag{4.15}$$

By calculating the two values $\overrightarrow{ray\_origin}$ and the vector $\overrightarrow{ray\_direction}$, a light ray is determined for a point on the retina and can be used to calculate the color values in the scene. Many rays fall on a receptor on the retina at the same time. This behavior must be taken into account in the simulation. For this reason, the value of many rays is taken into account for a pixel. This is achieved by emitting several rays from a pixel in different directions into space. The resulting values must be integrated into a value that can be used as the color value for a pixel.

To calculate this integration, a discretization is required in order to be able to perform this in finite time. The resulting color values of the individual rays are added up and averaged into the pixel color value by calculating the mean value.

$$col = \frac{1}{n} \sum_{0}^{n-1} f(r, g, b) \tag{4.16}$$

This value must be calculated for each pixel of an image in order to calculate the image that is formed on the retina. The higher the number of rays, the more accurate the resulting image.

4.5.3 Sampling

A basic function for calculating and approximating ray tracing algorithms is sampling (see section 3.1.4). This involves setting sampling points in the environment that are representative of a region of the environment. The number and distribution of the samples is important for the most accurate representation possible. This means that, for example, to approximate a circle, the sample points should be evenly distributed within this circle to represent its shape; the more of these points there are, well distributed, the more accurately the shape can be approximated.
In the case of the eye simulation, it is primarily the number of rays reaching the retina that requires the support points. Since theoretically an infinite number of rays can hit a single receptor or pixel of the simulation and this is possible for every point of the image, this makes it virtually impossible to calculate the image in finite time without sampling.

Lens Sampling

Sampling to simulate the eye is mainly carried out to approximate the lens shape in the middle of the eye. This is realized with the help of so-called concentric circles. These are circles that are calculated from an origin with different radii. Sampling points are placed at uniform intervals along these circles. These are mainly used to calculate the direction of the light rays that hit the retina (see chapter Simulation of the retina).

With the help of these points, the direction of the rays originating in the retina can be determined through the lens and thus the uniform and thus accurate coverage of the lens can be implemented. This uniform distribution can be seen in Figure 4.2, where the number of points can also be significantly higher than shown and

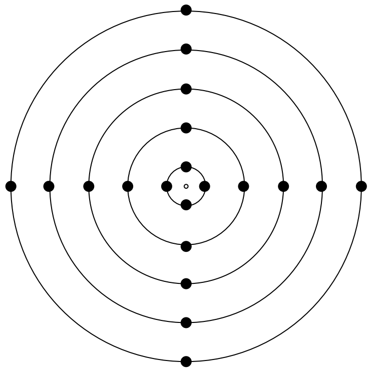

Figure 4.2: This figure shows the even distribution of the sampling points across the lens using concentric circles.

therefore points can also be found in the vertical. One advantage of this sampling method is the decoupling of the sampling points from the actual lens. In principle, a sampling point calculation based on the function for calculating the lens or on the vertices of the triangle mesh of a mesh-based lens would be conceivable, which on the one hand would reduce the effort of the independent sampling points, but on the other hand any changes to the lens function would automatically affect the sampling points. In many cases, this would result in a reduction in the uniformity of the distribution, which is why it is advantageous to separate these functions.

4.6 Visual Effects

Due to the complex image structure in the eye and its various properties, there are many different visual effects. These properties can be caused by the physical properties of the media used in the eye or, for example, by undesirable behavior such as diseases. Some of these effects are realized in this thesis based on the eye model described in chapter 4.4.1.

4.6.1 Light Regulation

The retina of the eye is a sensitive organ that can be damaged if the light intensity is too high. For this reason, it is necessary to protect it. This is done in the eye by the so-called iris. Similar to a camera aperture, this serves to reduce the incident light and thus protects the retina from excessive light intensity. However, as the eye requires a certain light intensity to function optimally, the iris must also be able to let in more light. For this reason, it is variable in size and can therefore be adapted to different light intensities in the environment.

Simulation of the Iris

In order to simulate this behavior, the function of the iris must be simulated. In the eye, this is a ring-shaped body that is located in front of the lens. It is opened or closed by the contraction of muscles. It forms a ring in the middle, which is larger or smaller depending on the degree of opening; the larger it is, the more light can reach the retina. If too much light falls onto the retina, it is closed and the ring becomes smaller. The technical implementation of the iris in this simulation is limited to restricting or expanding the incidence of light by means of a circular diaphragm.

For this purpose, an aperture is introduced in front of the lens of the eye model as a three-dimensional object, which has its origin in the center of the lens. This object has the shape of a disc, i.e. a round disk, which is defined by its origin and radius. Its size can be adjusted by changing its radius. The diaphragm works in such a way that light that hits the disc is transmitted into the scene, light that does not hit the disc hits the iris, which means that this beam does not leave the eye and does not hit a light source, resulting in the color black for this light ray. With a partially open iris, this results in an image that has a translucent part in the center of the image where the scene is displayed, but the edge of the image is black because the light rays are absorbed by the iris.

The described image with a hard edge of the iris shape is created when only a single ray is used per pixel. As described in chapter 4.5.2 Retina, in reality many rays fall on one retinal receptor. This results in some rays hitting the iris as seen from one point and thus being colored black, while others fall through the iris opening and thus receive light and a color value different from black. In the end, these values are averaged, which means that the color values are offset against black values, resulting in the reduction of the color value and thus the reduction of the brightness of a color. Furthermore, the edges of the iris become increasingly blurred with a higher number of rays, as the influence of each individual value decreases with a higher number.

Lightintensity Regulation

In order to be able to decide whether the iris should open or close for a given light intensity the light intensity of an image must be determined. This can be calculated using the average value of the colors p_r, p_g, p_b of an existing image. To do this, the color values of the pixels of the image are added up and divided by their number.

$$brightness = \frac{1}{n} \sum_{0}^{n-1} \frac{p_r + p_g + p_b}{3} \tag{4.17}$$

This brightness value can be used to decide whether the light intensity is too high, too low or within a suitable spectrum. Depending on this, the iris is adjusted. This can be measured via a limit value which must be set; if the brightness value falls below this, the iris can be opened, and closed if it exceeds it.

The adjustment to a target brightness value is performed iteratively. In the first iteration, it is measured whether there is a deviation from the limit value; if this is the case, the radius of the iris must be changed depending on the value of the excess. The radius of the iris must be reduced for darkening and increased for brightening. The radius is adjusted in small steps to enable a gradual approach. After each radius adjustment, the light intensity is measured again; if there are deviations, a further iteration is carried out until the limit value is reached.

4.6.2 Myopia and Hyperopia

Myopia and hyperopia are refractive errors that occur in the eye when the image plane does not match the distance between the lens and the retina. These errors can occur in varying degrees and the focal point can be too far in front of or behind the retina, see chapter 3.3.1. The simulation of this error by the schematic eye works by placing the image plane so that they are in front of the retina (myopia) or behind the retina (hyperopia), resulting in an unfocused image.

To correct these refractive errors, it is possible to use lenses that counteract this refractive error by shifting the focal point so that it forms the image plane on the retina again. This correction is simulated by placing a concave lens in front of the eye in the case of simulated myopia and a convex lens in the case of hyperopia. An attempt is made to achieve compensation by placing another lens in front of the eye lens, which refracts in the opposite direction to the eye's error and thus places the image plane back on the retina.

4.6.3 Chromatic Aberration

Chromatic aberration is an effect that is due to the fact that different wavelengths of light are refracted to different degrees (see chapter 3.3.2 Chromatic aberration). The implementation of this effect based on the simulated eye model requires the introduction of the possibility of using different refractive indices based on the color respectively wavelength. The effect is implemented in such a way that the rendering of an image is divided into three separate render passes: one for the red component, one for the green component and one for the blue component of a light ray. In each of these passes, the scene is rendered individually for the color components, which makes it possible to use different refractive indices for an object. The pixel values calculated from this with the proportions of the three colors are added after the individual render passes, resulting in a colored image with red (pic_r), green (pic_g) and blue (pic_b) components, which were calculated with 3 different refractive indices.

$$pic_{res} = pic_r + pic_g + pic_b \tag{4.18}$$

The resulting image contains objects that show a shift in color at their edges and one object is drawn multiple times in red, green and blue. The stronger the dispersion, the clearer the shift. The different refractive indices depending on the color can be determined in the so-called dispersion table, as these are different depending on the lens material.

4.6.4 Soiled Lens

Dirt or other particles on the lens or cornea of the eye are a common problem or disease. In addition to the eye's possible defensive reaction, these cause dark shadows in front of the eye or can lead to a complete darkening of the image, depending on the size and proximity of the dirt particles to the lens. The simulation of this "effect" or problem can be easily carried out with the existing eye model. For this purpose, one or more particles are placed in front of the existing lens to simulate the contamination. The most direct way to simulate such a particle is to construct it using a triangular mesh. A virtually arbitrary shape can be selected for this. The color of a particle is assumed to be black. The particle is positioned in the scene directly in front of the lens in order to simulate the contamination, which in reality is likely to be found on the cornea of the eye, as closely as possible.

Another effect that can simulate a dirty lens is the simulation of water on the lens. For this purpose, water droplets are placed directly on the lens, similar to contamination by dirt particles. These particles are given a refractive index corresponding to that of water and thus $n = 1.333$ to simulate tears or water droplets.

4.6.5 Photopic, Scotopic and Mesopic Vision

A central function of the eye is to adapt to different light intensities and situations. To regulate the amount of light hitting the retina, the regulation of the incidence of light, which is controlled by the iris in the eye, is relevant (see Simulation of the iris). Another mechanism of the eye is the adaptation of the retina to the brightness conditions of the environment. This means that the retina produces a different perception of the environment and its colors in different situations. This is achieved by the retina's receptors reacting differently to different levels of light intensity (see section 3.3.3). These adaptation levels are called photopic, scotopic and mesopic vision. The realization of these different types of vision in the simulation requires several steps or capabilities and components:

- The realization of the ambient brightness and lighting of the scene
- Information on receptor behavior at different brightness levels
- Application of the information to the color values of a pixel.

Ambient Lighting

To realize the ambient brightness, the so-called "ambient" lighting is introduced. This lighting simulates the brightness of the general ambient reflections of a scene and thus represents the general brightness of this. The so-called Phong lighting model is used for this and its ambient term is used:

$$I = k_a I_a + k_d(I_d \cdot (\hat{L} \cdot \hat{N})) + k_s(I_s \cdot (\hat{R} \cdot \hat{V})^n) \tag{4.19}$$

[Pho75]

$$\rightarrow I = k_a I_a \tag{4.20}$$

Here, k_a stands for the material constant, which describes how much light the material absorbs and thus models material properties. This constant is multiplied by the incident light intensity I_a to calculate the resulting light intensity of the reflected light I. As an alternative to using the Phong model, the ambient lighting could also have been calculated using the ray tracing algorithm and a high ray density, but this was not carried out due to the long calculation time and greater complexity. The resulting light intensity is identical at every point and in every direction, which is why there is no differentiation between different directions in the calculation. The ambient brightness values are used to determine and evaluate which ambient brightness is present in a room or scene.

To be able to simulate differences in illumination from point light sources, the other terms of the Phong model were also implemented:

The diffuse term describes the incidence of light that hits "rough" material and is therefore not reflected in one but several directions:

$$k_d(I_d \cdot (\hat{L} \cdot \hat{N})) \tag{4.21}$$

[Pho75]

As with "ambient" lighting, k_d stands for the material constant and I_d for the light intensity, while $\hat{L}$ stands for the vector in the direction of the light source and $\hat{N}$ for the normal of the surface on which the light ray hits.

The specular term describes the incidence of light that hits "reflective" material and for which the angle of incidence = angle of reflection:

$$k_s(I_s \cdot (\hat{R} \cdot \hat{V})^n). \tag{4.22}$$

[Pho75]

The values k_s and I_s also stand for light intensity and material constant in the specular term. The value R describes the ideal reflection vector, the value V the vector in the direction of the viewer or camera. The value n describes the degree of reflection that a material exhibits and therefore the amount of specular light that a material reflects.

The three terms are added together in the total intensity I, but only the ambient lighting term I_a is used to evaluate the ambient brightness.

Receptor Behaviour

The receptors of the retina, i.e. the rods and cones, react differently to different brightness values. At high brightness, the cones are more active, as a result of which the image has more evenly distributed color values; at low brightness, the rods are more active, as a result of which the color vision becomes more green-blue (see chapter 3.3.3). These states of color vision occur in their entirety when different threshold values are exceeded. At brightness values between these thresholds, so-called twilight vision, or mesopic vision, occurs. This behavior was measured and recorded in data and modeled in the so-called V and V′ lambda curves. The V-lambda curve is used for photopic, i.e. day vision, and the V′-lambda curve for night vision. The values for mesopic vision are calculated by interpolating values between the V and V′-lambda curves.

Application on Pixels

In order to apply the values of the photopic, scotopic and mesopic vision curves to the colors of the pixel values, the brightness of a scene must first be determined. To do this, the light intensity of a pixel is determined by calculating its color value. In this way, the brightness value is also determined using the "ambient" lighting values of the objects encountered. As soon as this value is set, it is determined whether the brightness value is to be assigned to the photopic, scotopic or mesopic view. As soon as this assignment has been made, the colors can be weighted with the corresponding values of the applicable V-lambda curve. Since the weighting is performed on the red, green and blue colors of the pixels and these colors cover an entire wavelength range of the curve, but a discrete value is required to be able to read out and apply the weighting. For this purpose, values were chosen which represent the wavelength spectrum on the V-lambda curve, for red the wavelength value of $600nm$, for green the value $540nm$ and for blue a value of $440nm$.

The calculation with weighting for the photopic view with the V-lambda curve results in:

$$\vec{col}_{res} = [pixCol_r \times 0.631, pixCol_g \times 0.954, pixCol_b \times 0.0230]. \tag{4.23}$$

The calculation with weighting for the scotopic view with the V′-lambda curve results in:

$$\vec{col}_{res} = [pixCol_r \times 0.03315, pixCol_g \times 0.65, pixCol_b \times 0.3281]. \tag{4.24}$$

For the mesopic view, the values of the V-lambda and V′-lambda curves must be interpolated. This works with the so-called "mesopic luminosity function":

$$Vm(\lambda) = (1 - x)V'(\lambda) + xV(\lambda) \tag{4.25}$$

The x-value is an interpolation value between 1 and 0, which is obtained by forming the difference between the limit value of the brightness of the photopic and scotopic view and normalizing it to the range between 1 and 0. The standardized value is then used for interpolation.

After implementing the three prerequisites, the photopic, mesopic and scotopic view can be applied to the final image and realized.

5 Results

In this chapter, the results of the methods developed are presented and discussed. In particular, the resulting images of the methods are examined and checked for plausibility. In some cases, real examples can be used for comparison, but since real-world examples are difficult to find or do not exist when looking through the eye, the results are compared between the expected value and the result.
The results of the eye model, which forms the basis, are discussed first, followed by the results based on the model.
The images were created and rendered using the software developed as part of this thesis.

5.1 Eye Model

This chapter looks at the results of the methods used to simulate the eye model. For this purpose, the renderings for the function of the lens, i.e. the accommodation of the lens for a specific object plane and the retina from which the rays are sent into the scene, are shown.

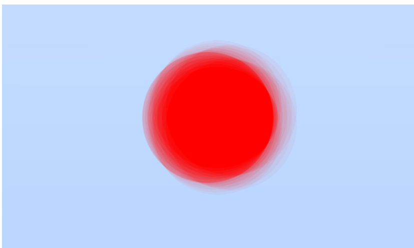

Figure 5.1: This image shows a blurred object at a distance of 5, which corresponds to a relatively short distance.

First, we will look at the function of focusing the lens and the model. Figure 5.1, 5.2 and 5.3, 5.4 show objects, a red sphere in this example, at different distances from the lens. The aim is to focus on these objects. In the initial state, the lens system is not set appropriately for an object, which is why it is displayed out of focus. This can be seen in Figure 5.1. An out-of-focus object can be recognized above all by the fact that the contours of the object do not overlap, especially at its edge. This is why it appears blurred. Colors also overlap and become indistinct. An out-of-focus object also takes up more space in the image than it would if it were in focus.

Figure 5.2: This image shows a focused object at a distance of 5.

Figure 5.2 shows the object from Figure 5.1 after it has been focused. In contrast to the blurred object, the focused object is considerably smaller. This is because the contours of the object are overlapping. This also means that color areas of the surface lie on top of each other and do not overlap. Despite the object being in focus, slight blurring can be seen at the edges. This is due to the fact that focusing using the calculation of the general lens formula for the spherical lens is not 100 percent accurate. A slight blur remains in most cases.

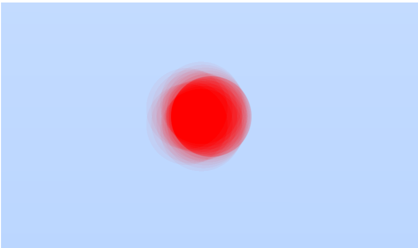

Figure 5.3: This image shows a blurred object at a distance of 10, which corresponds to a recognizably greater distance than 5.

Figures 5.3 and 5.4 show what happens when a newly focused object has a different distance than a previously focused object. The calculated distance for the object planes in Figure 5.3 is still at the distance value 5 of the object from Figure 5.2. However, the new object is now on object plane 10, which can be recognized by the smaller size of the object. For this reason, the representation of the object

Figure 5.4: This image shows a focused object at a distance of 10

is blurred again. The distance to the lens is then recalculated and used as the new object plane. The corrected and focused object can be seen in Figure 5.4, which is smaller due to the distance, but otherwise corresponds to the focused object from Figure 5.2.

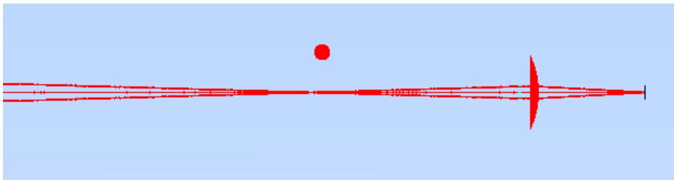

Figure 5.5: Illustration of the formation of a focal point when the ray falls from the retina through the lens. The focal point is at the level of the object plane of the sphere. The sphere is placed above the focal point for demonstration purposes.

Figure 5.5 shows the path of the light rays starting at the retina, through the lens until the focal point of the object plane is formed. This representation corresponds to the side view of the lens system; in the normal simulation, the scene would be simulated from the viewpoint of the surface on the right. Here you can see that a bundle of rays originates in the center of the retina and passes through the central area of the lens. This bundle of rays is only selected as an example; in the simulation, a bundle of rays is emitted from every point of the retina.
The focal point is at the height of the sphere, which means that it would be shown as a sharp object, as the object plane is formed exactly at the object. The sphere is placed over the focal point of the object plane solely for demonstration purposes in order not to cover it. If the focal point were to form far to the left or right of the object, this would correspond to a blurred image.

For better illustration and to demonstrate the formation of a focal point, this is shown again enlarged in Figure 5.6. This focal point corresponds to the one shown

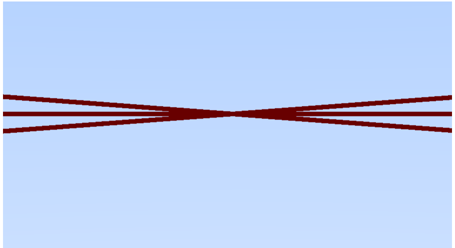

Figure 5.6: Illustration of the formation of a focal point of the ray bundle. This bundle forms the focal point on the object plane.

in Figure 5.5 at the height of the object.

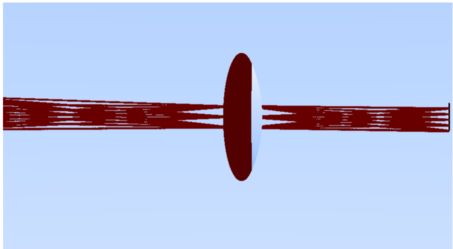

Figure 5.7: This illustration shows the origin of the rays in the form of ray bundles on the retina.

Figure 5.6 shows the retina. Several ray bundles were used here to get closer to the simulation situation. Here you can see that rays are emitted over the entire retina and pass through the lens. For reasons of clarity, not all possible rays are shown here either. The rays mainly pass through the central area of the lens, as this is the relevant area for refraction. The further the rays would be distributed in the height of the lens, the higher the potential errors in the final image would be.

5.2 Visual Effects

In this part, the results of the visual effects based on the eye model are discussed. These effects are light regulation by means of the iris, near and far vision and their correction by means of a lens, chromatic aberration, dirt particles on the lens of the eye, as well as day and night vision and the resulting color changes.

5.2.1 Light regulation

The effect of light regulation by varying the size of the iris is an important effect for the correct functioning of the eye. The Light Regulation method (Chapter 4.6.1) was developed for this purpose and presented using a sphere.

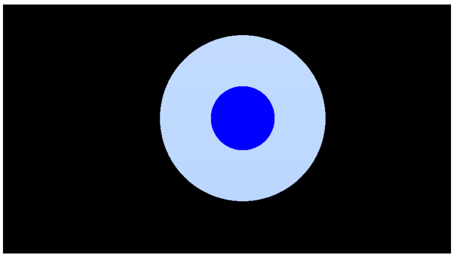

Figure 5.8: Iris with single ray resolution, almost closed.

Figure 5.8 shows the iris in an almost closed state. It can be seen that the areas covered by the iris are shown in black. The area that is not colored black is the area through which light enters the eye and reaches the retina. The object, in this case a blue sphere, can be seen through this area, as can the background of the scene. The resolution, i.e. the number of rays per pixel, is set to one ray per pixel in this image, which corresponds to the lowest possible value (a value of zero would result in a completely black image) in order to be able to display the iris shape.

The single-resolution image shown in Figure 5.8 shows the iris very clearly. As this is not realistic, the resolution per pixel in Figures 5.9, 5.10, 5.11, 5.12 is increased to a value of at least 10x10, i.e. 10 sample points in 10 different circles (see Chapter 4.5.3), which corresponds to a resolution of at least 100 rays per pixel. This increase in resolution leads to a "blurring" of the contours of the iris, so that in Figure 5.9 the circle has virtually disappeared and only a darkening can be seen. The degree of aperture of the iris now determines how bright or dark the resulting image is displayed.

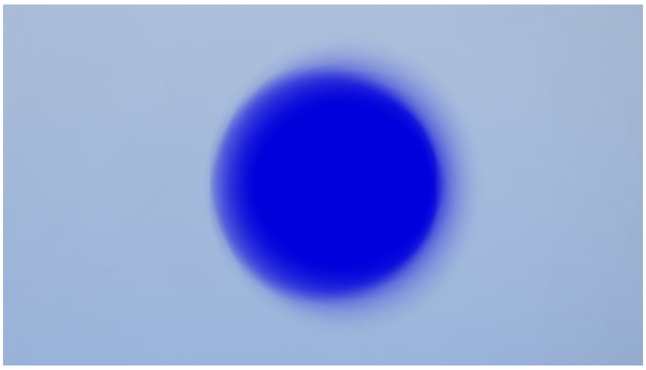

Figure 5.9: Very wide open iris with high resolution

Figure 5.10: Wide open iris with high resolution.

In Figures 5.9 and 5.10, the iris is not recognizable as an object, but only a darkening of the scene. This darkening extends over the entire image, although the iris opening is located in the center of the image and thus leads to the behavior of image darkening due to less light entering the eye, without changing the image itself.

Figure 5.11 shows a widely shut iris, which is why the image is very dark. In this image, it can be seen that when the iris is strongly closed, a slight shape can be recognized again, the edges are already strongly darkened or completely black while some light transmission can still be seen in the middle of the image. This behavior is as expected, since with a strongly closed iris very few rays can fall through the iris, especially rays that lie at the edge of the retina can almost no longer fall through the remaining visual aperture, since only very few sample points remain in the area of the visual aperture. In order to reduce this effect even further, a higher number of rays would probably have to be aimed for, but this would result in a considerably longer calculation time for the image.

Figure 5.11: Almost closed iris with high resolution.

5.2.2 Chromatic Aberration

Chromatic aberration is an effect that occurs due to different refractive indices of different wavelengths.

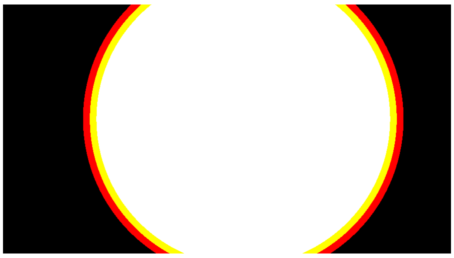

Figure 5.12: Chromatic aberration on white sphere.

Figure 5.12 shows the effect of chromatic aberration with a simple resolution. In this image, a white-colored sphere can be seen at the edges of which, from the inside to the outside, first a yellow and then a red color can be seen. This is due to the fact that the colors are refracted to different degrees. In the area of the white coloration of the sphere, all rays of the colors red, green and blue hit equally and mix evenly, resulting in the white. In the next "ring", the yellow one, only the colors red and green meet, which results in the color yellow in case of additive color mixing. In the outermost ring, only the color red can be seen. This is due to the fact that the color red is refracted the most in this image and therefore takes up the most space in the resulting image. The color green is refracted the second strongest, which is why the areas of green and red overlap and therefore mix. The color blue is refracted the least strongly, which is why the blue area is the smallest and results in a white sphere together with the other colors.

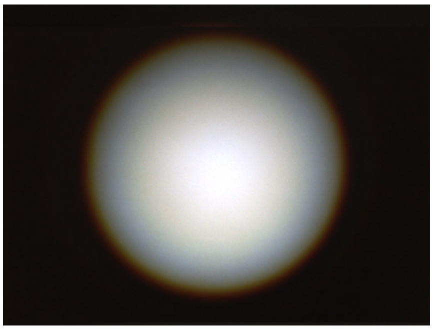

Figure 5.13: Chromatische Aberration an Taschenlampenkegel.[Fer24]

Figure 5.13 shows a real-world image of a flashlight cone, which also exhibits chromatic aberration. In this image it can also be seen that the edges of the sphere first have a yellow-colored edge and then a red-colored edge further out. If you compare Figures 5.12 and 5.13, which show a very similar scene, you can observe a similar effect. Since the image of the simulation is very close to the real world image, it can be concluded that the chromatic aberration in the simulation works correctly.

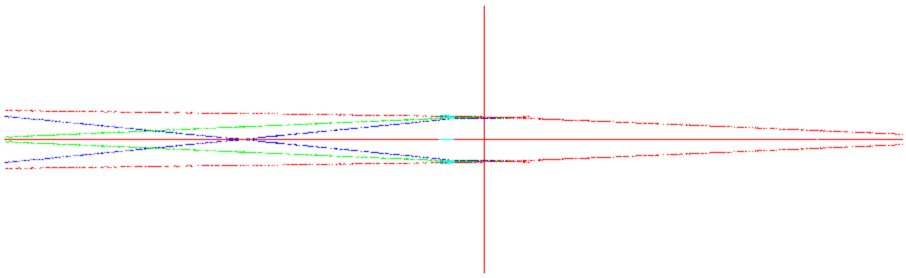

Figure 5.14: Representation of the division of a single light ray into differently refracted light rays with different wavelengths. Lateral view of the scene, on the right the retina with the origin of the rays, in the center the lens (red line).

Figure 5.14 shows the course of the rays from the retina through the lens. Here you can see how the rays, which are still moving to the right of the lens in the form of a single bundle, split into three separate rays after passing through the lens. Red, green and blue rays can be seen, each for the refraction of the corresponding wavelength.

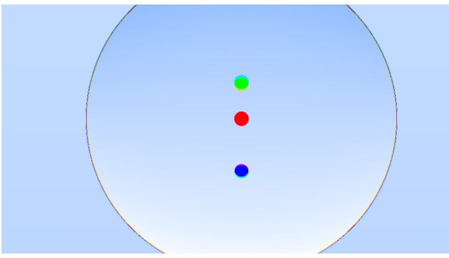

Figure 5.15: View through a glass sphere with visible chromatic aberration. Color gradients visible at the edge of the glass sphere, as well as at the edges of the middle red, green and blue spheres.

Figure 5.15 shows an image of a more complex scene. In this scene, the effect of chromatic aberration can be observed primarily at the edge of the glass sphere (larger, transparent sphere). It can also be seen that the effect only occurs at the edges of the sphere and only has a small influence on the resulting image. The deviation of the colors is so close to each other that they almost "mix". the spheres that can be seen through the glass sphere show a considerably higher shift of the colors. The upper green and lower blue spheres in particular show considerable differences in refraction. On the one hand, this effect could be due to the fact that the objects are represented by a double refraction, on the one hand by the eye lens and on the other by the glass sphere. Secondly, this effect could also be amplified by the slight refractive error of the lens (see 5.1 Eye model).

5.2.3 Soiled Lens

Dirt on the lens often affects the field of vision. However, the dirt is not visible as a clear object on the lens, but causes a haze over the field of vision of the affected eye. This is simulated by placing a particle directly in front of the lens of the eye.

Figure 5.16: Dirty lens with low resolution.

Figure 5.16 shows a particle that was rendered with a very low resolution. Clear edges of the particle can be recognized. It can also be seen that the particle, which is actually an opaque object, becomes transparent directly in front of the eye.

Figure 5.17: Dirty lens with high resolution.

In Figure 5.17, in contrast to Figure 5.16, the resolution has been greatly increased to a value of at least 10x10. Here you can see that the clear edges of the object's representation become very blurred and turn into haze.

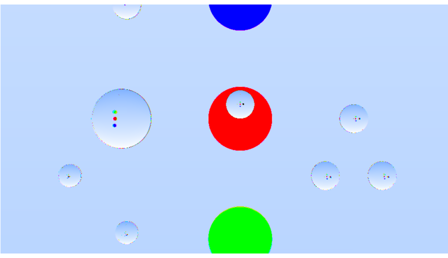

Figure 5.18: Drops of water in front of the lens at low resolution. The colored spheres are displayed upside down in the spheres due to the refraction of light.

Figure 5.18 shows a representation of water droplets on the lens. The chromatic aberration effect is also activated here. The image has been rendered with a resolution of 1x1. Drops can be seen in the foreground. The colored sphere in the background can be seen in these drops, which are turned upside down due to the refraction of the material.

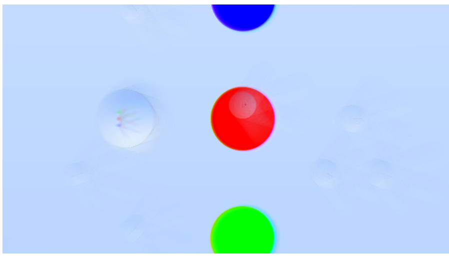

Figure 5.19: Drops of water in front of the lens at high resolution. Due to the high resolution, the image is blurred, but the outlines of the drops are still recognizable.

In Figure 5.19, the image has a much higher resolution than in Figure 5.18. The refraction effects can still be seen in the now blurred water pots, but these are also distorted by the higher resolution.

5.2.4 Photopic, Scotopic and Mesopic Vision

The effect of day and night vision is divided into three different areas: photopic (day), mesopic (twilight) and night vision (scotopic). This phenomenon is demonstrated using an example scene showing a Christmas tree with a present. The scene is designed to emphasize and demonstrate the proportions of blue, green and red in the different representations.

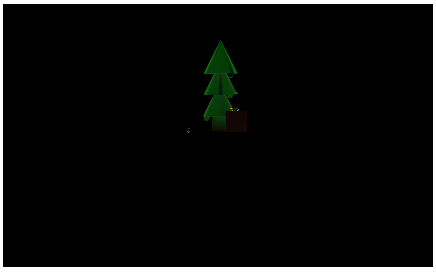

Figure 5.20: Representation of an object in the form of a Christmas tree and gift at low light intensity (scotopic). The colors green and blue are more present than the color red.

Figure 5.20 shows the scotopic view. This can be seen on the one hand from the low brightness, but also from the color distribution of the image. The dominant color value in this representation is green. There are also some blue-colored areas, especially on the strand of the tree, as well as on the gift, which actually has a red hue, but tends towards blue in this image. This color distribution matches the course of the color curve V'-Lambda, which models the scotopic view (see chapter 3.3.3).

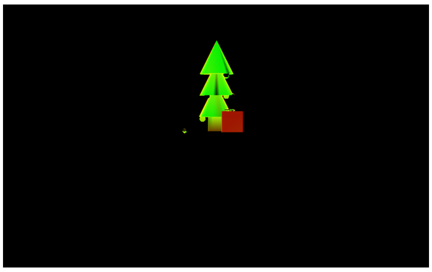

Figure 5.21: Depiction of an object in the shape of a Christmas tree and gift with high light intensity (photopic). Green and red are relatively strong, blue a little less.

Figure 5.21 shows the same scene as in Figure 5.20, but this time with higher brightness values to represent the photopic view. It can be seen that in the photopic view, the red color of the gift in particular is clearly visible. This corresponds to the expectation and the values of the V-lambda curve, which is particularly pronounced for red and green values. This is less pronounced for blue values, which is why they are less visible in the image of the photopic view.

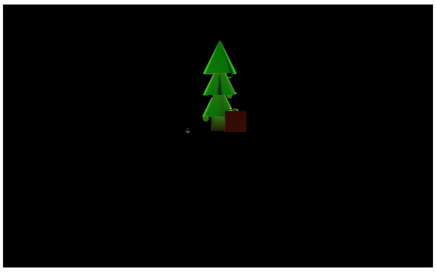

Figure 5.22: Representation of an object in the form of a Christmas tree and a present at medium light intensity (mesopic). Green is strongly present, blue and red less so.

The mesopic view, which is shown in Figure 5.22, is obtained by interpolating the photopic and scotopic views, i.e. values between the V and V' lambda curves. For this reason, color values can be expected in the resulting image that lie between those of the two views shown in Figure 5.20 and 5.21.
This coloring can be seen in the figure. The red color is weaker than in the photopic representation, but more visible than in the scotopic representation. Blue parts can also be seen on the leaves of the tree, which were not visible in the photopic view.

The renderings show that the photopic, scotopic and mesopic views are displayed with the expected color distribution and thus the desired result was achieved.

6 Conclusion

In this work, the questions "How does the human eye see?", "Can this be visualized with ray tracing?", "What optical effects can be represented on this basis?" were addressed. To answer these questions, a simulation was developed to simulate the basic functions of the eye. This system is based on the ray tracing rendering algorithm, which is used to simulate light. This simulation was used to simulate the path of light through the eye to the retina. For this purpose, the components of the eye that are necessary for correct image generation were recreated. This primarily includes the lens and the retina, but also effects on the retina such as the behavior of the receptors. Ultimately, several effects could be implemented on this basic framework. One of the most basic functions, the generation of a focused image, the so-called accommodation, could be simulated. The first step was to develop a process that enables the ray tracing program to emit a large number of rays in different directions for one pixel value. In addition to this method for emitting rays, a sampling method using concentric circles was also developed, which forms the basis for calculating the direction of the rays. By introducing this into the ray tracing program, "blurred" images could be generated that could simulate the almost infinite number of light rays in the real world. A lens was introduced to focus the blurred image, which is also used in the eye for this purpose. Various ways of constructing this lens were compared and ultimately the so-called "alternative sphere surface lens" was chosen due to its viewing angle stability and comparatively inexpensive calculation. Using the lens, the image could be focused by creating an image plane for an object plane through focal point formation, which generates a sharp image at the level of the retina and can thus focus on an object in space. This simulated the basic function of the eye, allowing its effects to be recreated. The first of these effects was the darkening of the image using the iris. A disc was implemented here, which is variable in size and determines how much light is allowed to fall on the retina. The amount of incident light was measured and adjusted by enlarging or reducing the size of the disc and thus adjusting the incidence. In this way, the function of the iris in the eye was imitated.

Another effect was chromatic aberration, which simulates the different refraction of different colors and thus wavelengths and the associated effect of color shifts. For this purpose, several render runs were introduced into the raytracing software, which ultimately calculates three differently refracted images and merges them back into one at the end in order to simulate this effect.
A common problem that can often become visible in everyday life is a dirty lens. This causes haze or cloudiness to appear in the vision, which affects it. An attempt was made to simulate this phenomenon by placing simulated dirt particles directly in front of the artificial lens. This was also carried out with water droplets. As a result, a veiled image was created using high resolution. The last effect that was simulated was that of photopic, mesopic and scotopic vision. For this, the so-called V and V' lambda curves were used to simulate the different color perceptions at different light intensities. For this purpose, data from the CIE organization was used and applied to the simulation. Ambient lighting was also introduced for this purpose. As a result, it was possible to observe how the different curves affect the colors at different brightness levels.
Was it possible to simulate the eye using the ray tracing algorithm in this work? It must be said that a completely realistic simulation was not the aim of this work, as this would go beyond the scope of this thesis. However, it was possible to achieve a simulation of the lens model of the eye using ray tracing, which corresponds in its function to the basic aspects of the human eye. This means that an object can be focused on almost any object plane by changing the thickness of the lens. The generation of a sharp image is also based on the change of light rays. In these aspects, the simulation basically corresponds to the function of the eye. However, there are differences in the type of lens, which does not correspond to that of the eye, but to a form that is easier to calculate and use for ray tracing and this simulation. The many layers that make up the body of the eye have also been combined into one overall refraction in order to reduce complexity and possible errors. Therefore, the questions can be answered in the sense of a simplified simulation with yes, a full simulation of the eye function was not the aim of this work and was not implemented and achieved due to the high complexity.
The second question, which effects could be implemented on the basis of the eye simulation, cannot be answered in full, as there is a very large number of effects and combinations of these. However, some effects could be implemented. In addition to the effects that originate from the eye itself, such as the darkening effect of the iris or photopic, mesopic and scotopic vision, these also include effects that

occur due to physical phenomena, such as chromatic aberration. It can therefore be stated that, in addition to the basis of the eye, various effects can also be simulated on this basis, although there is still plenty of potential for further simulations and effects.

It can therefore be stated that the question has been answered on the whole, but that many properties, functions and effects still need to be simulated and recreated in order to achieve a comprehensive and completely realistic simulation. The software framework used for the processing and implementation of this work, which is not a professional but an in-house development, had advantages in processing, as there were virtually no limitations due to restrictions of the software, such as sampling, but on the other hand it also had disadvantages, some of which manifested themselves in greatly increased effort due to the lack of features. These features, such as the display of the lens system and the course of the rays, which can be seen in Figures 5.5, 5.6 and 5.7, had to be reprogrammed from scratch without being able to fall back on existing features. This took a considerable amount of time in some cases.

7 Outlook and Future Work

Future research and goals that can build on or extend the results of this work can be found above all in the expansion and scope of the simulation. There is great potential for this in functions that simulate the individual components in greater detail, such as other lens shapes using splines or similar display variants, or a lens that is modeled as closely as possible on the lens of the eye. The simulation of all individual components from the cornea of the eye to the retina, as well as other components of the eye such as the eye socket, could also be added. There is also potential in the investigation of methods to improve performance, such as the possibility of using data structures such as Kd-trees or similar structures, or the possible use of the rasterizing algorithm to simulate the eye, which also allows real-time applications, for example. For future work based on the methods of this thesis, the use of professional or established environments or frameworks as a basis is recommended, as these can greatly simplify the work, as they provide the basic functions that do not have to be redeveloped. It can therefore be concluded that the most relevant points and objectives have been achieved, but that there is still great potential for improving and expanding the field of ray tracing in the area of eye simulation in this work, as well as in general.

Bibliography

[Bar04] B. A. Barsky. "Vision-realistic rendering". In: *APGV 2004*. Ed. by V. L. Interrante, A. McNamara, H. H. Bülthoff, and H. E. Rushmeier. ACM Press, New York, NY, 2004, pp. 73–81. ISBN: 1581139144 (cit. on p. 6).

[BAR57] H. B. BARLOW. "Purkinje shift and retinal noise". *Nature* 179:4553, 1957, pp. 255–256. ISSN: 0028-0836 (cit. on p. 20).

[BIP19] BIPM/CIE. "Principles governing photometry - 2019", 2019 (cit. on pp. 20, 21).

[Cho+17] S. A. Cholewiak, G. D. Love, P. P. Srinivasan, R. Ng, and M. S. Banks. "Chromablur". *ACM Transactions on Graphics* 36:6, 2017, pp. 1–12. ISSN: 0730-0301 (cit. on p. 6).

[Fer24] L. Fermum. *Chromatische Aberration*. 8.01.2024. URL: https://www.vision-doctor.com/optische-fehler/chromatische-aberration.html] (cit. on pp. 19, 54).

[Ger64] H. Gernet. "ber Achsenlnge und Brechkraft emmetroper, lebender Augen". *Albrecht von Graefes Archiv für Ophthalmologie* 166:5, 1964, pp. 424–431. ISSN: 1435-702X. URL: https://link.springer.com/article/10.1007/bf00682315 (cit. on p. 13).

[Gla07] A. S. Glassner. *An introduction to ray tracing*. Morgan Kaufman, San Francisco, 2007. ISBN: 9780122861604 (cit. on pp. 7, 8).

[GME19] M. A. GMELCH. *DURCHBLICK IN OPTIK: Mit phnomenen, formeln und fragen zum verstndnis*. 1. Aufl. 2019. Springer, 2019. ISBN: 9783662589397. URL: http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1545330 (cit. on pp. 16, 33).

[Gre06] B. de Greve. *Reflections and refractions in ray tracing*. 2006. URL: https://graphics.stanford.edu/courses/cs148-10-summer/docs/2006--degreve--reflection_refraction.pdf (cit. on pp. 9, 10).

[GSMM95] J. E. Greivenkamp, J. Schwiegerling, J. M. Miller, and M. D. Mellinger. "Visual acuity modeling using optical raytracing of schematic eyes". *American Journal of Ophthalmology* 120:2, 1995, pp. 227–240. ISSN: 0002-9394 (cit. on p. 5).

[HH11] M. Hapala and V. Havran. "Review: Kd–tree Traversal Algorithms for Ray Tracing". *Computer Graphics Forum* 30:1, 2011, pp. 199–213. ISSN: 0167-7055 (cit. on p. 12).

[HHa10] HHahn. *V-lambda-phot-scot.svg*. Lizenz: GNU Free Documentation License, Version 1.2 oder eine spätere Version, veröffentlicht von der Free Software Foundation; mit keinen unveränderlichen Abschnitten, keinen Vorderen und keinen Hinteren Umschlagtexten. Eine Kopie der Lizenz ist im Abschnitt mit dem Titel GNU-Lizenz für freie Dokumentation enthalten. Diese Datei ist unter den Creative-Commons-Lizenzen „Namensnennung – Weitergabe unter gleichen Bedingungen 3.0 nicht portiert", „2.5 generisch", „2.0 generisch" und „1.0 generisch" lizenziert. 2010. URL: https://commons.wikimedia.org/wiki/File:V-lambda-phot-scot.svg (cit. on p. 21).

[KTMN07] M. Kakimoto, T. Tatsukawa, Y. Mukai, and T. Nishita. "Interactive Simulation of the Human Eye Depth of Field and Its Correction by Spectacle Lenses". *Computer Graphics Forum* 26:3, 2007, pp. 627–636. ISSN: 0167-7055 (cit. on p. 6).

[Lia+19] T. Lian, K. J. MacKenzie, D. H. Brainard, N. P. Cottaris, and B. A. Wandell. "Ray tracing 3D spectral scenes through human optics models". *Journal of vision* 19:12, 2019, p. 23 (cit. on p. 5).

[MKL97] S. Mostafawy, O. Kermani, and H. Lubatschowski. "Virtual Eye: retinal image visualization of the human eye". *IEEE Computer Graphics and Applications* 17:1, 1997, pp. 8–12. ISSN: 02721716 (cit. on p. 6).

[NSG12] M. Nießner, R. Sturm, and G. Greiner. "Real-time simulation and visualization of human vision through eyeglasses on the GPU". In: *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*. Ed. by D. Thalmann, E. Wu, Z. Pan, A. El Rhalibi, N. Magnenat-Thalmann, and M. Adcock. ACM, New York, NY, USA, 2012, pp. 195–202. ISBN: 9781450318259 (cit. on p. 6).

[Pho75] B. T. Phong. "Illumination for computer generated pictures". *Communications of the ACM* 18:6, 1975, pp. 311–317. ISSN: 0001-0782 (cit. on pp. 43, 44).

[Ser98] Serges Medien, ed. *Schule 99 Grundstock des Wissens*. Serges Medien, Köln, 1998 (cit. on pp. 13, 15).

[TX15] N. Tang and S. Xiao. "Real-time human vision rendering using blur distribution function". In: *Proceedings of the 14th ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry*. Ed. by I. Fujishiro, Z. Pan, N. Magnenat-Thalmann, S. N. Spencer, M. Oshita, X. Yang, and H. S. Yang. ACM, New York, NY, 2015, pp. 39–42. ISBN: 9781450339407 (cit. on p. 5).

[Wee22] D. Wee. "Schematic Eye". In: *Optics for the New Millennium: An Absolute Review Textbook*. Ed. by K. M. Riaz, G. V. Vicente, and D. Wee. Springer, Cham, 2022, pp. 85–89. ISBN: 978-3-030-95250-1 (cit. on p. 5).

[WPP14] Q. Wei, S. Patkar, and D. K. Pai. "Fast ray-tracing of human eye optics on Graphics Processing Units". *Computer Methods and Programs in Biomedicine* 114:3, 2014, pp. 302–314. ISSN: 0169-2607 (cit. on p. 5).