Visualizing General Relativity: Perception of Schwarzschild Black Holes

Nicole-Christine Popst

München 2021

Visualizing General Relativity: Perception of Schwarzschild Black Holes

Nicole-Christine Popst

Masterarbeit
an der Fakultät für Physik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Nicole-Christine Popst
aus Burghausen

München, den 3. Mai 2021

Erstgutachter: Prof. Dr. Jochen Weller

Die allgemeine Relativitätstheorie Visualisieren: zur Wahrnehmung Schwarzer Löcher mit Schwarzschild Metrik

Nicole-Christine Popst

München 2021

1 Abstract

In this master thesis ways of visualizing curved space around massive objects are explored. In particular a virtual reality (VR) application is described and implemented that visualizes the vicinity around a Schwarzschild black hole. Bending of light, frequency shift, magnification by gravitational lensing and relativistic aberration all take part in shaping the appearance of a black hole in space. The position of the observer relative to the black hole greatly alters the perception of it. A 2d image or video does not completely convey the visual appearance. Therefore virtual reality was used as a way to visualize curved space. VR offers the possibility to observe the object interactively in 3d space and thereby helps to get an intuitive understanding of the visual effects. In the center of visualizing curved space is the numerical integration of ray tracing algorithms. Different approaches will be presented and compared. For the task of programming an interactive real-time VR-application traditional ray tracing methods could not be used. By storing the spacetime information of the curved area around the black hole in a lookup table and by utilizing GPU-programming this barrier could be overcome.

Contents

2 Introduction

The behaviour of light within a gravitational field is nowadays described by General Relativity. However, even Newton, who considered light to consist of particles, already assumed that light particles should theoretically be deflected under the influence of a gravitational field. [1] In his Opticks, published in 1704, he wrote: "Do not Bodies act upon Light at a distance, and by their action bend its Rays; and is not this action (caeteris paribus) strongest at the least distance?" But he did not go further with these thoughts. Years later, in 1783, in a letter exchange between John Michell and Henry Cavendish, Michell remarked "... if the semi-diameter of a sphere of the same density with the Sun were to exceed that of the Sun in the proportion of 500 to 1, a body falling from an infinite height towards it, would have acquired at its surface a greater velocity than that of light, and consequently, supposing light to be attracted by the same force in proportion to its vis inertiae, with other bodies, all light emitted from such body would made to return towards it, by its own proper gravity". He went on by proposing that such invisible "black" objects could be found by revolving bodies around them. Cavendish calculated the deflection angles of light rays inspired by this exchange without publishing the results. [2] [3] In 1796, Pierre Simon Laplace independently proposed that "the attractive force of a heavenly body could be so large that light could not flow out of it. There is an escape velocity which increases with the mass of the object: [4]

$$v_e = \sqrt{\frac{2GM}{R}} \tag{1}$$

It reaches the speed of light at the radius:

$$R_S = \frac{2GM}{c^2} \tag{2}$$

This means that an object with mass M and a radius smaller than R_S would have a gravitational field which would not even allow light to escape and consequently result in a black object. More than a hundred years later, with the theory of general relativity, [20] Einstein proposed a framework to theoretically describe these phenomena of black holes which Michell and Laplace anticipated. In 1916, Karl Schwarzschild found the solution to the Einstein field equation of a point mass [19]. One of the consequences of this solution is that black holes have an Event Horizon as given by the Schwarzschild radius R_S.

But going back to Newtonian mechanics, to calculate the deflection of a light particle passing by a gravitational field one derives the deflection angle α [1] :

$$\tan\left(\frac{\alpha}{2}\right) = \frac{GM}{v^2 r} \tag{3}$$

which for small angles leads to:

$$\alpha \simeq \frac{2GM}{v^2 r} \tag{4}$$

here one can set $v = c$ to consider light. This is the value that should be observable in the case of a flat euclidean space. Using the equations of General Relativity, however, Einstein got to a value of:[5]

$$\alpha = \frac{4GM}{c^2 r} = \frac{2R_S}{r} \tag{5}$$

so twice the Newtonian value. This was affirmed by the solar eclipse of May 29, 1919, during an expedition lead by Arthur Eddington. It was also Eddington who first pointed out that multiple images of the same star may appear when the alignment of the objects is sufficient. [6]

A black hole forms when a star reaches a stage where internal pressure forces are no longer sufficient to hold against the inward pull of its own gravitational force. This can happen when a star which measures several masses of the sun cools down and cannot remain the pressure necessary for equilibrium any longer. When this happens, the collapse becomes inevitable as the gravitational force works relentlessly. Under these high pressure conditions, mass is forced to react. Of particular interest is electron or neutron degeneracy pressure. It is caused by the Pauli principle which states that two or more identical fermions cannot occupy the same quantum state. A white dwarf, for example, is stabilized by its electron degeneracy pressure and is as dense as an object can be. Degeneracy pressure, however, is not enough to prevent gravitational pull when the mass exceeds about two times our solar mass. Subrahmanyan Chandrasekhar established in 1931 the limit of $1.4M_{\odot}$ for white dwarfs. The limit is slightly larger for neutron stars. [7] Every cold object that has a considerably larger mass will fall into a limitless inward pull.

In certain cases of high masses however, the infall of the outer parts can trigger an explosion called a supernova. Some of them might result in a white dwarf or a neutron star. In other cases, a black hole will form due to the unstoppable inward fall and will reach the limit where the escape velocity equals the speed of light and an event horizon occurs. Figure 1 depicts this scenario graphically.

So far, these are theoretical assumptions. It took a couple of decades until empirical evidence was found concerning the objects in consideration. In the 1960's, multiple examples where found of "strange" double-star systems where only one star was visible. [12]

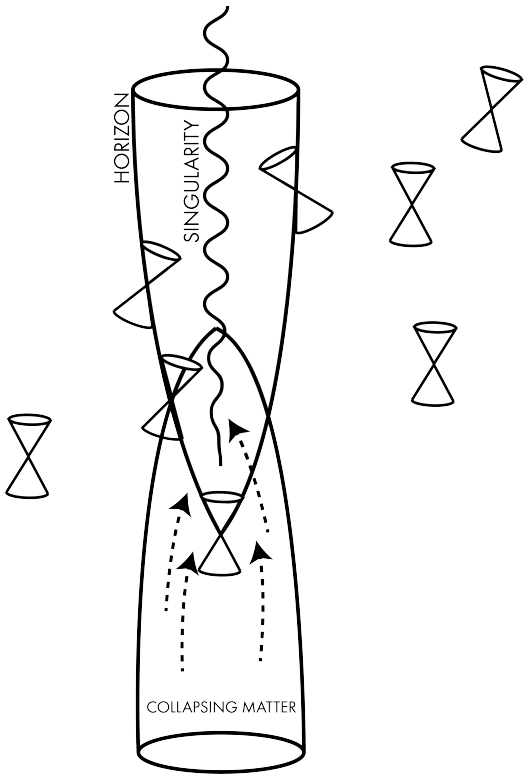

Figure 1: A space-time diagram of a black hole with one spacial dimension suppressed. The null cones are tangent to the horizon. Signals can move inwards but not outwards.

The movement and luminosity of one star was obviously influenced by a partner star. The X-ray signals coming from this region also clearly showed that this invisible object was a massive object reminiscent of a white dwarf or neutron star. These X-ray emissions where consistent with the radiation of an accretion disk surrounding a black hole. A accretion disk is made up of gas and dust and spirals closer and closer towards the horizon, becoming extremely heated as the distance diminishes. Before the material enters the hole, it will radiate as X-rays. One of these cases was the X-ray source Cygnus X-1 whose invisible "partner" was having a mass of about $7M_{\odot}$. It was verified as a black hole 2011. [9] [10] [11] Years later, it was also shown that even more massive black holes reside in our universe. [13] [14] An extreme mass and the absence of any surface of certain objects within galaxies pointed to the discovery of supermassive black holes which are located in many galaxies and may be found in all of them – they might even take on an important part in their development. In the Milky Way seemingly resides a massive $30M_{\odot}$ black hole - Saggitarius A*. [8] Stellar black holes and super-massive black holes continue to be an active area of research for experimental and theoretical physicists alike. In the theoretical regime, black hole research is of supreme importance to understand the extreme conditions around the singularity where our standard model breaks down and a theory combining quantum mechanics and general relativity needs to be developed. Any candidate for this job, e.g. string theory, will ultimately be tested as to how well it can describe those extreme spacetime regions which are still enigmas. The most staggering experimental breakthrough is the picture taken in 2019 of the super-massive black hole located in the elliptical galaxy Messier 87 by the event horizon telescope. [15] This followed the 2016 success of observing gravitational waves which also indirectly observed black hole

mergers. It was the result of 10 years of research and the combined effort of 60 international institutions. It is seen as yet another prove for general relativity. The numerical magneto-hydrodynamic simulations that accompanied the observations [16] pointed to a strong correlation between actual observation and simulation. This proves the current proximity of our theoretical understanding to the real observational nature of black holes.

In the following visualizations of the regions close to black holes are explored. Chapter 3 and 4 give a theoretical and mathematical background that underlies the practical implementations. Chapter 5 describes in detail the different visual effects that are to be observed around a black hole, which is the distortion of the environment and the shift in color frequencies induced by general and special relativity. Chapter 6 outlines the computational formalism of graphical visualization which is primarily governed by ray tracing algorithms. Different works in this field are presented and compared. In chapter 7, the implementation accompanying this work is explained, which is a visualization in virtual reality (VR). The equations describing the trajectories of the light beams are quite cumbersome to calculate and require a lot of computational resources. Real-time rendering is therefore quite challenging, because straight forward ray tracing is simply not possible. Non the less, real-time visualization can be achieved by storing the necessary data that has been pre-calculated in lookup tables that are then read by the GPU in real-time. Interactive, real-time rendering, as it is necessary for virtual reality, is thereby possible. VR gives the possibility to interact with the visual distortions that appear around a black hole. The user can move in a 3d environment in which the black hole is placed and can therefore observe it freely from varying directions and angles. Thereby, the user can witness the change of the environment that is induced by the curved space and the relative position of the user. The black hole in this application is placed in front of a star background (the milky-way given from the Hipparcos catalogue), so by moving around, the observer can see the ever changing position of the stars near the horizon.

3 General Relativity

A black hole is a region in spacetime in which a massive star has collapsed and whose gravitational pull becomes so strong that even light particles can no longer escape. The region of no-return is the event horizon. Schwarzschild calculated that the event horizon is located at the radial distance $r = 2MG/c^2$. The event horizon is not made out of anything; it is a hypersurface in spacetime that describes the place from which all information is consumed by the black hole. Within the center of the black hole is a singularity, a spacetime region where the curvature of space reaches infinity. The theoretical description of black holes and curved space is given by General Relativity. General Relativity tells us that the curvature of spacetime is directly related to the energy and momentum of the local distribution of matter. This relation is expressed in the Einstein tensor G (a generalization of the Riemann Curvature Tensor which more generally describes curvature). [17] The central relation is the Einstein field equation

$$G_{\mu\nu} = 8\pi T_{\mu\nu} \tag{6}$$

with T being the Stress-Energy tensor. The equation is independent of coordinates and encodes many important consequences. This framework allows us to describe the necessary objects to calculate the light trajectories within curved spacetime which then allows us to visualize what these curved spaces would look like. In the following, a more detailed look will be provided, highlighting this purpose.

3.1 Gravitational Principles and Mathematics

Within general relativity, spacetime is described as a curved manifold. Differential geometry is the mathematical framework to describe these manifolds. [17] A manifold is described by an atlas of coordinate charts. One coordinate chart maps an open set on the manifold to an open set on the coordinate charts. Multiple charts are necessary to cover the whole manifold. These maps must be diffeomorph. This means that the transformation function and its inverse are smooth. In addition, the transition function from one chart to other charts in the atlas are smooth as well. Lengths on a manifold are given by an expression that takes into account the the metric tensor $g_{\mu\nu}$ of the manifold. As the metric of space-ime is not positively definite, spacetime is a pseudo-Riemannian manifold. For every point in spacetime there is a coordinate transformation which brings the metric tensor at that point to the flat Minkowski metric. This concludes that spacetime is a Lorentzian manifold. The physical interpretation of that is the equivalence principle: one can always find a local inertial frame of reference.

In the following, there are some mathematical outlines of the more important objects for the calculations that underlie the implementation and numerical calculation of the visual distortion within curved space.

Metric Tensor the metric g defines distances and angels on a manifold. It is a tensor of rank (0,2).

$$g = g_{\mu\nu} dx^\mu \otimes dx^\nu \tag{7}$$

The flat Minkowski space has, for example, a metric of $g_{\mu\nu} = diag(-c^2, 1, 1, 1)$. In n-dimensional Minkowski coordinates $g_{\mu\nu}$ becomes $\eta_{\mu\nu}$. For the infinitesimal segment ds^2 we have

$$ds^2 = \eta_{\mu\nu} dx^\mu dx^\nu \tag{8}$$

where dx^μ is the infinitesimal distance in the μ direction of the coordinate system.

Local Tetrad To describe the reference frame of an observer inside a manifold with metric g at a point P, we need a set of four vectors called tetrad. These vectors are orthonormal to each other. This means the vectors $\mathbf{e}_i = e^\mu_{(i)} \partial_\mu$ fulfill the relation:

$$\langle \mathbf{e}_i, \mathbf{e}_j \rangle = \eta_{ij} \tag{9}$$

with $\eta_{ij} = diag(-1, 1, 1, 1)$. With this orthonormal basis we can set up a vector that describes the direction of any incoming or outgoing ray:

$$\mathbf{y} = y^{(0)} \mathbf{e}_{(0)} + \Psi(\sin(\eta)\cos(\xi)\mathbf{e}_{(1)} + \sin(\eta)\sin(\xi)\mathbf{e}_{(2)} + \cos(\xi)\mathbf{e}_{(3)}) \tag{10}$$

with $y^{(0)} = -1$ and $\Phi = 1$ for a ray that goes back in time. It reduces in a rotational symmetric space-time to:

$$\mathbf{y} = y^{(0)} \mathbf{e}_{(0)} + \psi(\cos(\xi)\mathbf{e}_{(1)} + \sin(\xi)\mathbf{e}_{(2)}) \tag{11}$$

Euler-Lagrange-formalism the coordinate independent Lagrange is given by: [18]

$$\mathcal{L} = g_{\mu\nu} \frac{dx^\mu}{d\lambda} \frac{dx^\nu}{d\lambda} = g_{\mu\nu} \dot{x}^\mu \dot{x}^\nu = \kappa c^2 \tag{12}$$

λ is an affine parameter and κ is a dimensionless parameter which is -1 for time-like, 0 for light-like and 1 for space-like geodesics. The Euler-Lagrange equation with affine parameter is:

$$\frac{d}{d\lambda} \frac{\partial \mathcal{L}}{\partial \dot{x}^\mu} - \frac{\partial \mathcal{L}}{\partial x^\mu} = 0 \tag{13}$$

Geodesics Geodesics describe the shortest and therefore "straight" lines in curved space. They are the solutions to a set of ordinary second-order differential equations, the geodesic equations:

$$\frac{d^2x^\mu}{d\lambda^2} + \Gamma^\mu_{\nu\rho}\frac{dx^\nu}{d\lambda}\frac{dx^\rho}{d\lambda} = 0 \tag{14}$$

λ is an affine parameter. $\Gamma^\mu_{\nu\rho}$ are the Christoffel symbols which are calculated with the metric tensor. There are three different kinds of geodesics: time-like, space-like and light-like. Massive objects move along time-like geodesics, photons move along light-like geodesics also called null geodesics.

The trajectories of light-like geodesics described by the geodesic equation can be simplified to the form:

$$g_{\mu\nu}\frac{dx^\mu}{d\lambda}\frac{dx^\nu}{d\lambda} = 0 \tag{15}$$

where the metric tensor g can be the Schwarzschild metric tensor, for example, and thereby this equation describes the movement of light around Schwarzschild black holes.

4 Black Holes

Black holes are the center of our interest. They arise from the gravitational collapse of a star whose mass gets compressed into a small region of spacetime. The curvature reaches infinity at its center called the singularity, which gets screened to outside observers by the (non-physical) coordinate singularity at a finite radial distance. This coordinate singularity at $r = 2M$ is called the event horizon.

Outside of the horizon the non-rotating and non-charged black hole is described by the Schwarzschild metric with a Schwarzschild coordinate (t, r, θ, ϕ). It is the simplest solution to the Einstein equation and is rotational symmetric and describes the gravitational field of any spherical mass distribution. If the mass distribution is confined to the boundaries of the Schwarzschild radius, it is a black hole. [18]

The Schwarzschild Metric is given by:

$$ds^2 = -\left(1 - \frac{R_S}{r}\right) c^2 dt^2 + \frac{1}{(1 - \frac{R_S}{r})} dr^2 + r^2 d\theta^2 + r^2 sin^2\theta d\phi^2 \tag{16}$$

It only describes the outside space of R_S and has a coordinate singularity at the Schwarzschild radius. This is just a mathematical singularity and can be removed by changing the coordinate system. For this to show, one can use the Eddington-Finklstein coordinates (v, r, θ, ϕ). It is given by

$$ds^2 = -\left(-\frac{r_s}{r}\right) c^2 dv^2 + 2cdvdr + r^2 d\Omega^2 \tag{17}$$

It does not have a singularity at the horizon. This formulation will be useful for describing trajectories which cross the horizon. This metric now only exhibits a singularity at $r = 0$. This remaining singularity is often called a physical singularity. Physical singularities are generally defined as points in spacetime where a scalar quantity that measures the gravitational field blows up. For this purpose, the Ricci scalar is used, yielding $R = R^j_j = \frac{48M^2}{r^6}$ for the Schwarzschild metric with $r = 0$ a "real" singularity.

We have shown that the radius $r = 2M$ does not define a physical singularity, but does in turn define the event horizon. The event horizon is a 2-dimensional surface at $r = 2M$ that divides $r > 2M$ and $r < 2M$ where $g_{00} = 0$ and arriving at the singularity is inevitable. It is interesting to note that an observer cannot see anything crossing the horizon as the light cone closes up as r approaches $2M$:

$$\text{radial light rays} \rightarrow ds^2 = 0 \rightarrow \frac{dr}{dt} = \pm\left(1 - \frac{2M}{r}\right) \tag{18}$$

with $c = 1$. But even though the light cone closes up and the event cannot be observed whatever falls into the black hole will reach $r = 2M$ at finite time. When an observer reaches the event horizon she will not experience anything strange although her destiny is sealed as there is no way of return anymore. [28] As it is our goal to describe the trajectories of light and how they hit a camera which is located in the spacetime area around the black hole, we need in addition to the actual metric a way to express the position and orientation of objects within the given metric. For this we use local tetrads as they were discussed prior in 3.1. A local reference frame of a static observer is described by the natural local tetrad given by:

$$\vec{e}_\nu = \frac{1}{\sqrt{1 - r_s/r}} \frac{\partial_\nu}{c} \tag{19}$$

$$\vec{e}_r = \frac{1}{\sqrt{1 - r_s/r}} \frac{\partial_\nu}{c} + \sqrt{1 - \frac{r_s}{r}} \partial_r \tag{20}$$

$$\vec{e}_\theta = \frac{1}{r} \partial_\theta \tag{21}$$

$$\vec{e}_\phi = \frac{1}{r \sin\theta} \partial_\phi \tag{22}$$

An initial direction $\mathbf{y} = y^i \vec{\mathbf{e}}_i = \pm \vec{\mathbf{e}}_t - \cos\xi \vec{\mathbf{e}}_r + \sin\xi \vec{\mathbf{e}}_\phi$. The sign of $\vec{\mathbf{e}}_t$ decides whether the four-directions point to the future (plus) or the past (minus). As we consider only spherical symmetry, $\mathbf{e}_\theta$ can be neglected and the angle ϕ will be constrained between the region $[0, \pi]$. In practice, these tetrads are used in ray tracing techniques where the integrated movement is often used as the starting and end point of the integrations as cameras, and the objects they capture can be "placed" in the "scene" with the tetrads.

4.1 Diagram Techniques

Penrose-Carter diagrams is a way to visualize and understand the causal relation between points within a spacetime. The spacetime of black holes are asymptotic and therefore difficult to understand in an intuitive way. Penrose diagrams achieve this through a conformal mapping of the infinite structure of space and time to finite coordinates. The vertical axis represents time and the horizontal represents one-spacial dimension. Lines with the angle 45° represent the movement of light ($c = 1$). Diagonal lines perpendicular to light-rays correspond to infinities of spacetime.

To achieve the Penrose-Carter diagram for a Schwarzschild metric, we need the Kruskal metric:

$$ds^2 = -\frac{4r_S^3}{r} e^{r_S/r} \left(dv^2 - du^2\right) + r^2 d\Omega^2 \tag{23}$$

and transform it with conform compactifying coordinates

$$v = \frac{1}{2}\tan\left(\frac{\phi+\xi}{2}\right) + \frac{1}{2}\tan\left(\frac{\phi-\xi}{2}\right) \tag{24}$$

$$u = \frac{1}{2}\tan\left(\frac{\phi+\xi}{2}\right) - \frac{1}{2}\tan\left(\frac{\phi-\xi}{2}\right) \tag{25}$$

here $-\pi < \phi + \eta < \pi$ and $-\pi < \phi - \eta < \pi$. This leads to a new line element:

$$ds^2 = -\left(\cos(\phi) + \cos(\xi)\right)^{-2} \frac{4s_S^3}{r} e^{r_s/r}(d\phi^2 - d\xi^2) + r^2 d\Omega^2 \tag{26}$$

$$\left(\frac{r}{r_S} - 1\right) e^{r/r_S} = u^2 - v^2 = \frac{\cos(\phi) - \cos(\xi)}{cos(\phi) + \cos(\xi)} \tag{27}$$

the singularity that was at $r = 0$ is now at $\phi = \pi/2$ and the horizon that was at $r = r_S$ is now at $\phi = \pm\xi$.

With this set, what happens when approaching a black hole can be better understood and explained. Light-like trajectories are always depicted with a 45° angle. Time-like objects will be depicted with a higher angle than 45°; they cannot move on trajectories smaller than 45° because then they would move faster than light. In Figure 2, the spacetime region is divided into region I and II. Region I is outside the event horizon and region II inside of the event horizon. When an object is within the region I, there is always the possibility to move with a speed smaller than c on a trajectory to escape the black hole and move to infinity. If, however, something crosses the horizon, a time-like trajectory cannot escape the singularity. In Figure 2,

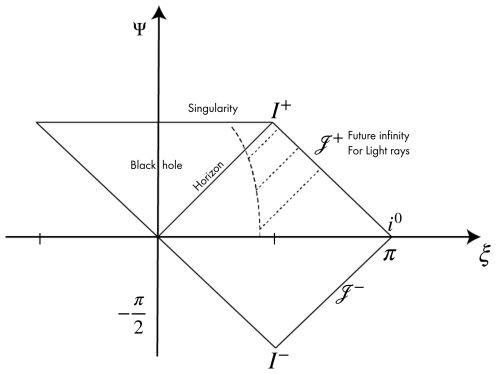

Figure 2: Diagrammatic representation of the Schwarzschild metric. When an observer falls into a black hole and sends out light signals along the way, they will arrive in increasingly larger intervals. After crossing the horizon no signal can escape this spacetime region and the observer will inevitably reach the singularity.

a dashed trajectory is representing an
object, maybe a spacecraft, falling into the black hole. Every now and then, it emits a signal making up the dotted line which escapes infinitely far into space and time. Note however that a light-ray emitted close to the horizon will end up at I^+, which is an end-point of time-like world-lines reached only by observers who have experienced an infinite amount of proper time. So whatever happens at the horizon cannot be observed within finite time. Therefore, from an observer's outside perspective, it appears that something which falls into the black hole slows down infinitesimally and never crosses the horizon. This is another way of looking at the event horizon crossing event from the prior chapter. [28]

5 Visual Effects in a High Gravity Environment

The extreme environment around a black hole or generally an area in spacetime with high curvature results in many visual effects. Our eyes are used to see in flat spacetime. The appearance of a highly curved spacetime will appear to be distorted as the way light-rays hit our eyes do not make sense as to how our brain processes visual information. The results are the perception of multiple images of the stellar sky in a gravitational lens with amplification of the stars, caustic effects or "Einstein Rings" and extreme distortion of the area. The light spectrum also undergoes changes which result in other visible colors. This is due to special relativistic effects and gravitational effects. Special relativistic effects happen due to the motion of the observer, which will reach high velocities due to the gravitational pull. Those different effects will be explained and described in the following. Yet in a more realistic setting, black holes within our universe will also have an accretion disk, made up of matter that falls into the black hole and, they will also more than likely rotate. Rotating black holes are described by the so-called Kerr metric which leads to its own visual implications. Even though these cases were not the object of interest in the practical implementation of this thesis, they will still be briefly discussed due to their physical relevance.

5.1 Gravitational Redshift

When the photons climb out of the gravitational field they loose energy. This loss of energy is visible by a shift of wavelength. The spectrum shifts towards red. Those photons falling into the gravitational field become more energetic and become blueshifted. The equation for redshift states for the energy [40]:

$$E_{observed} = \frac{\sqrt{1 - R_S/r_{emitted}}}{\sqrt{1 - R_S/r_{observed}}} E_{emitted} \tag{28}$$

and the frequency shift is given by:

$$f_{obs} = f_{source} \left(1 - \frac{2GM}{rc^2}\right)^{1/2} \tag{29}$$

This means that everything which surrounds a black hole and which we can see had to climb out of the gravitational potential and is therefore redshifted. Stars which orbit black holes, for example, appear redshifted as well as accretion disks. In reality, details can vary, and in the case that the emitted light is in the ultra-violet spectrum, it would shift the frequency to violet and so the observer would perceive a violet blueish object which was caused by redshift. But this is just one special case that shows that appearances can deceive.

5.2 Relativistic Doppler Effect

When the observer falls into the black hole or is pulled into it in a radial direction (radial symmetry of a Schwarzschild black hole) the relativistic longitudinal Doppler effect will change the frequency of the light hitting the observers eye.

$$f_{obs} = \sqrt{\frac{1-\beta}{1+\beta}} f_{source} \tag{30}$$

where $\beta = v/c$. The higher the velocity becomes, the smaller the prefactor of f_{source}. The observed frequency becomes smaller, it is blue-shifted. So falling into a Schwarzschild black hole will change the color of the surrounding stars into higher frequencies.

As we observed in the former chapter, the gravitational potential leads to a redshift in perception. What the observer actually sees will depend on his or her speed. If the observer is static, the gravitational redshift will dominate, whereas if the observer is in free fall, the relativistic Doppler effect will dominate more with the increase of velocity.

In another case where the observer moves transversely in relationship to the position of the black hole, the frequency shift would appear differently. Also for a rotating black hole the frequency shift would appear differently, as the spin can achieve extremely high velocities. [31] [32]

5.3 Relativistic Aberration

Besides Lorentz contraction and time dilation, another relativistic effect which occurs at high velocities is relativistic aberration. This causes the sight of field of the observer, who moves at high velocities, to expand. The direction of the source changes according to:

$$cos\left(\phi_{obs}\right) = \frac{cos\left(\phi_{emit} - v/c\right)}{1 - cos\left(\phi_{emit} - v/c\right)} \tag{31}$$

The higher the speed, the more objects the observer sees; these are actually behind him (in this case stars). The object in front, in our case the black hole, seems to be contracted. So an observer who falls freely into a black hole will see the black hole shadow smaller than it actually is (for shadow size, see chapter 5.7). At the event horizon, the black hole actually encompasses half of the observers sky, but if the observer falls freely and is close to the speed of light, it still appears to be far away and the shadow will never encompass the whole sky. [33]

5.4 Multiple Images and Amplification

A strong gravitational field will cause a distortion of the stellar sky, so one light source will be seen multiple times. In a spherical field, this image will be located on the plane defined by the observers position, source and the center of the lens (in this case, the position of the black hole). Due to the principle of conservation of angular momentum, the image cannot be seen outside of the defined plane. (see Figure 11)

An extended source will not only appear multiplied on the sky around the center, but it will also be hugely distorted. The surface brightness, however, will be maintained after red- or blueshift corrections. The conserved entity is the "correct surface brightness" $B_c = B_r(1 - R_s/r)^2$ with B_r being the surface brightness at distance r. B_c is the surface brightness at the observer far away from the source, summed over all possible energies. The observer might see several images of an extended source surrounding the center of the black hole, and every one of them - even though they might be hugely distorted - has the same surface brightness after the red- or blueshift was corrected.

The flux, however, (luminosity) is affected by an amplification A and can result in a larger or smaller flux as the original source. This means a source which is influenced by a gravitational field can appear to have a higher luminosity than it would have without the gravitational field. There are two factors which cause amplification. One is the time distortion induced amplification A_{time}. This is related to the slowing of time in the given gravitational potential which results in a change of the photons energy (blue- or redshift) as well as the perceived arrival time (which in turn results in a changed power when integrated over all wavelengths). The other amplification is due to the apparent angular size of the source $A_angular$. The total amplification is therefore $A_{total} = A_{time} \cdot A_{angular}$. The time induced amplification is calculated with [40]:

$$A_{time} = \frac{\left(1 - \frac{R_S}{r_{emitted}}\right)^2}{1 - \frac{R_S}{r_{observed}}} \tag{32}$$

The angular amplification typically out-weighs the time amplification and depends on the ratio between the observed source with and without gravitational field:

$$A_{angular} = \frac{\sin\alpha}{\sin\beta}\left(\frac{d\alpha}{d\beta}\right) \tag{33}$$

β represents the angle between the source and the observer without the lens (black hole) and α represents the angle with the lens (black hole).

On the whole, the number of photons hitting the observer does not change, they are just redistributed and red/blue-shifted. Some sources (at a certain

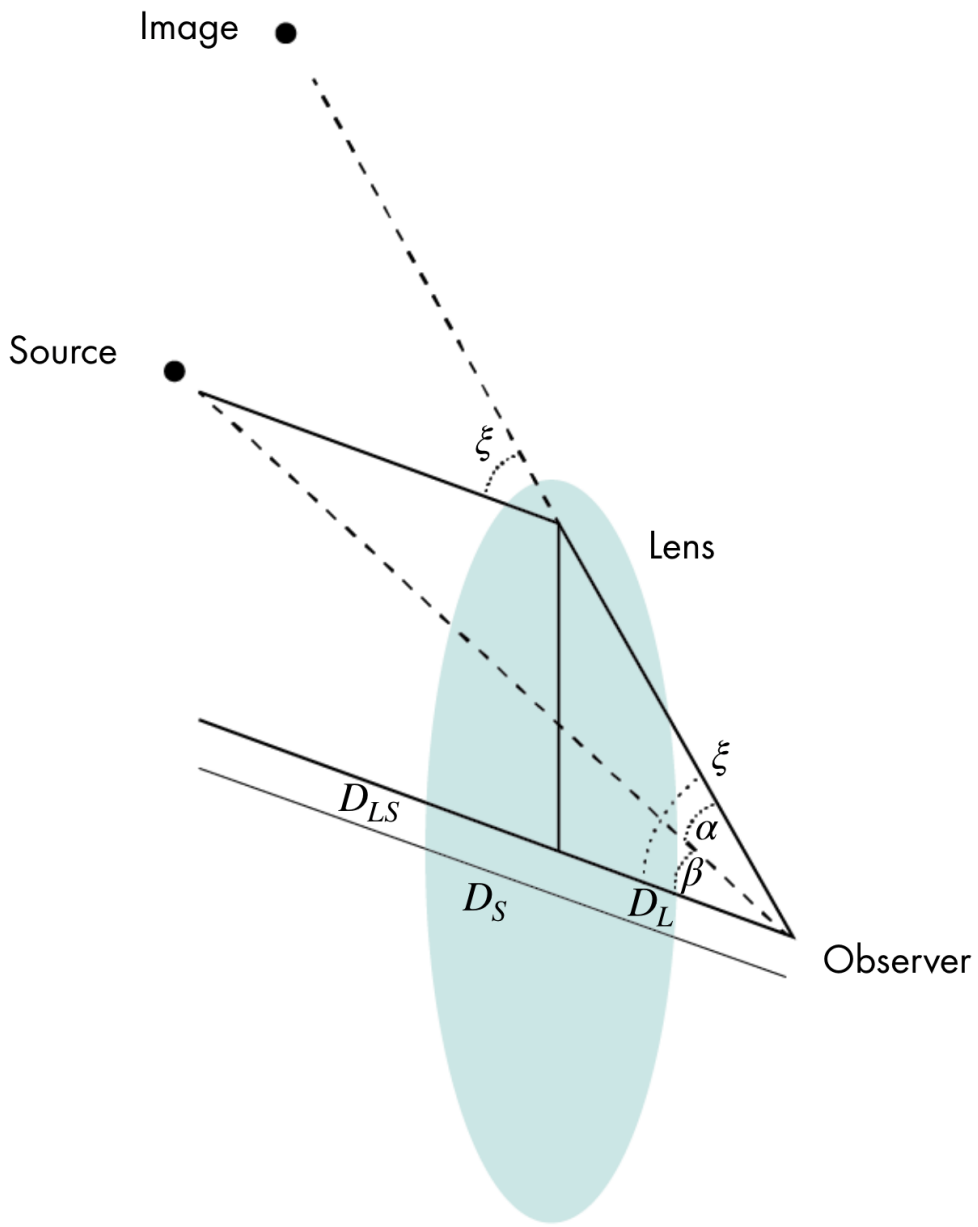

Figure 3: A black hole is a gravitational lens. Light rays which connect an observer with a light emitting source take different paths with and without the presence of a (gravitational) lens. As with classical lenses in optics, physical quantities to describe the system are the distance between emitter and observer D_S, observer to lens D_L and lens to emitter D_{LS} and the two angles between lens axis and image with and without the lens.

angle) have an amplification that is $A > 1$, others have $A < 1$. [23] The total amplification, however, is $A_{total} > 1$, with A_{total} increasing the closer the observer is to the black hole.

5.5 Einstein Rings

Einstein rings are a special case of the source distortion discussed above. We defined the outcome of the distortion dependent on a plane constructed by source, observer and lens center. In the case that the source, the observer and the center of the lens are all placed on one line, an Einstein ring will be formed. [24] [25] [26] If the source is a single star, its light will be visible as one fine circular line around the center. Multiple Einstein rings can be formed, and they divide the distorted images that were explained in the former chapter [62]. They are visible if a star is located at the required place. More likely, however, it will not be visible, simply because a star being located at exactly that spot is unlikely. The angle at which the Einstein ring would be located is:

$$\alpha = \sqrt{\frac{4GM}{c^2}\frac{D_{LS}}{D_s D_L}} \tag{34}$$

This is the first Einstein ring. However, there are several. A photon can circle around the center multiple times before it reaches the observer. These are the odd numbered Einstein rings. The higher the number, the dimmer the ring because the likelihood of a photon to actually arrive to the observer decreases with every turn.

The other set of Einstein rings (composed of even numbers) are observable if observer and source are on the same side of the lens. The second ring is a light source from behind the observer, but taken on a trajectory around the center of the black hole, and it reaches the observer from that direction visible as an Einstein ring. Higher order rings are again trajectories of photons which turn around the center multiple times. [40]

This phenomenon does not only occur in high gravity environments but also in low gravity environments such as in the area around galaxies and clusters of galaxies. [29] In analogy to optics and massive astronomical objects working as gravitational lenses, Einstein rings are also referred to as "caustics". [64]

5.6 Complete Sky and Surface Visibility

Between two Einstein rings there is an image of the complete sky. [40] If a photon sphere exists, there will be infinitely many of these images. Between the zeroth and first Einstein ring is the primary image. Between the first and the

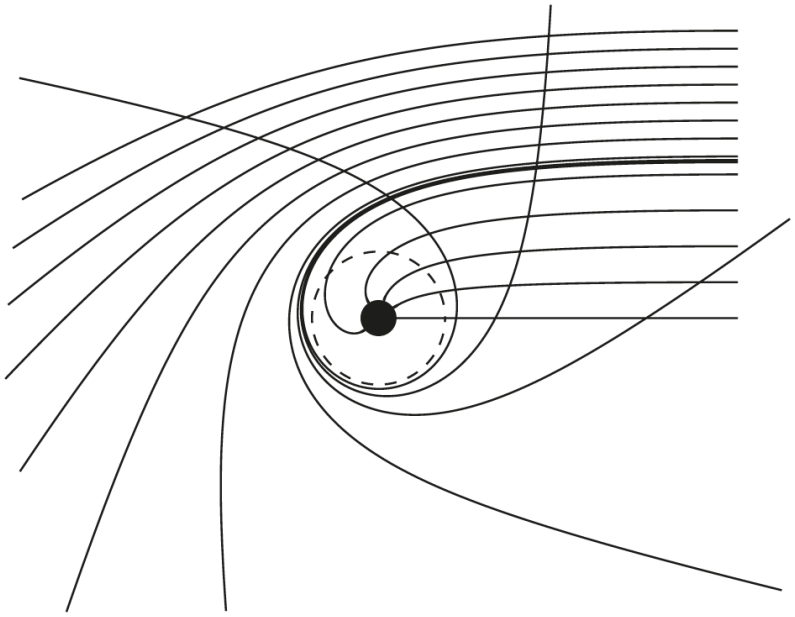

Figure 4: The trajectory of light rays are bent around black holes. When they pass the black hole too close, they cannot escape and will fall into the black hole (dashed line, event horizon). Light rays that hit an observer nearby can come from angles which are completely different than those from where they were emitted. In extreme cases, they can return to the point of emission or they might take multiple turns around the black hole before arriving at a destination. The light rays follow null-geodesics in Schwarzschild metric and these images can be achieved by visualization algorithms like [41]. Taken from Müller T., 2011 [49].

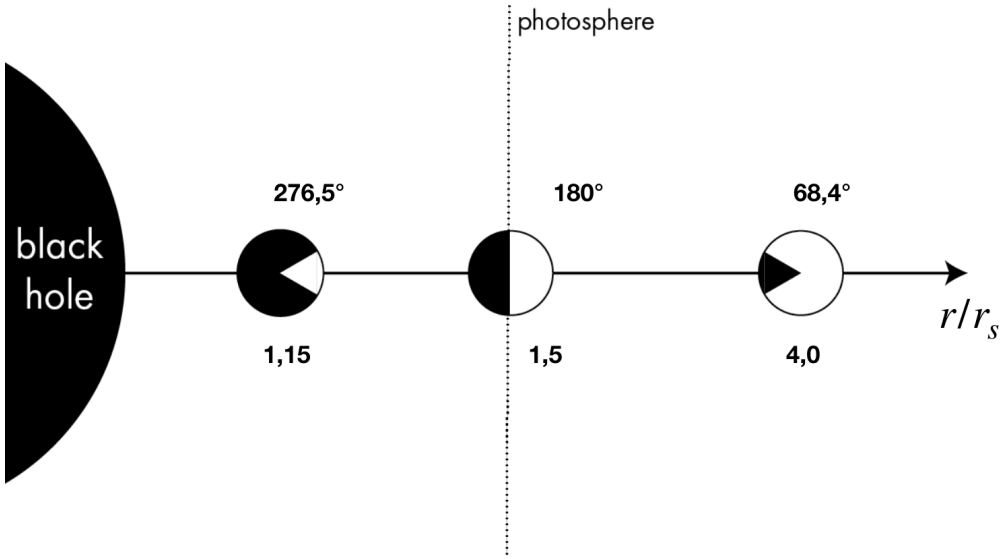

Figure 5: The closer the observer gets to the horizon, the bigger the field of view completely filled with the black hole will be. At the photon sphere, half of the observer's field of view will be black, and by crossing the horizon, the rest of the star sky will be shrunk to a point. Values from Boblest S., 2016 [18].

second Einstein ring, the second image is located and is mirrored by a 180deg. The next image is, in turn, not mirrored. All successive Einstein rings appear in this pattern of being mirrored and not mirrored. The third image has circled around the black hole once before it reaches the observer. The forth image circled around the black hole in the opposite direction. The next image circled twice around, again in the opposite direction, and so on. Theoretically, there is an infinite number of Einstein rings.[27] This results in a parallax effect: if the observer moves around the black hole in one way, it will appear as if the successive image moves in the other directions. (This effect is very well observable in the accompanying VR application.)

In fact, every star has an Einstein ring. Only if the star is massive enough like a neutron star or a black hole a second appears. But light is bent around stars like around our sun. Second Einstein rings can only exist for high mass objects like neutron stars and black holes. An infinite number Einstein rings are possible only for those objects that have a photon sphere.

5.7 Shadow of a Black Hole

The apparent size of the black hole is often called the black hole's shadow. [18] The visual diameter 2ξ of the black hole can be calculated as follows:

$$\xi = \arcsin\left(\sqrt{\frac{27}{4}\frac{r_s^2}{r_{obs}^2}\left(1-\frac{r_s}{r_{obs}}\right)}\right) \tag{35}$$

It is dependent on the position of the observer r_{obs} and on the Schwarzschild radius r_s. The closer the observer gets to the black hole, the bigger it appears. At the photon sphere ($r_p = 3r_s/2$) 50 percent of the sky are black. The closer the observer gets, the more of the visible celestial sky shrinks to a small tiny point behind the observer. The apparent size will change, though, dependent on whether the observer is static or falling freely.

5.8 Static vs Falling Observer

We can ask the question as to how a black hole appears from two perspectives - from the perspective of an observer who is falling freely and from the perspective of a static observer. Depending on that, the black hole has a different apparent size. Both observers will also perceive time differently. In both cases, general and special relativistic effects compete with one another. The apparent size results from the bending of light around the black hole and from the aberration of light of the fast moving observer. For a static observer the size of the black hole or the shadow of the black hole will reach the size as outlined in the previous chapter. If, however, the observer is falling freely, the aberration due to special relativity will increase with the increase of speed which makes the apparent size of the black hole smaller. If no other effects were present, it would have the size of a point at the speed of light. Yet as the two effects (general and special relativistic) compete, there is a finite size of the black hole which is around $2\xi \approx 84.2$ at the event horizon and $2\xi \approx 180$ at the singularity. [48]

The relationship of proper time between a static and a fast moving observer depends on time dilation due to the curved spacetime and on the motion of the observer itself. As already pointed out in 4.1, an object falling into the black hole never really reaches the horizon from an outside perspective. Supposing that every black hole arises due to the collapse of star material, a distant observer will not be able to see a complete black hole as at least the surface of the star does not appear to ever reach the event horizon. This is also what an observer reaching the black hole will perceive: only when she crosses the horizon can she see that the material has crossed the horizon right after her (as objects distant to her move slower from her perspective). From the perspective of the observer

close to the black hole, it appears as if the time for the distant observer would move faster. This is true for the static observer for whom the time dilation slows down time for objects outside of the gravitational field due to the curved spacetime. For a freely falling observer, this is also true, but it even increases the effect, as special relativistic time dilation also slows down the time of the objects outside of the fast moving reference frame. Exact relations are given e.g. in [48] and for more details see [39].

5.9 Accretion Disk

When matter falls into a black hole, it will distribute in the shape of an accretion disk. This happens due to the conservation of angular momentum from the infalling material. Inside the disk, the particles experience friction which causes the angular momentum of the system to move outwards and the material will thus fall inwards, thereby releasing potential energy which, in turn, heats up the gas. The energy that is released bursts as electromagnetic radiation in the X-ray regime. It is thought that these processes occur in active galactic nuclei and in quasars. They have first been described with general relativity by Thorne and Page [45]. Due to the electromagnetic nature of these accretion disks, they are best simulated and visualized by ray traced general relativistic magneto-hydrodynamic methods which are an extensive field of research on their own. They take in consideration the dynamic flow of electro-emissive plasma. None the less, in order to give a rough idea as to how a accretion disk would visually behave around a Schwarzschild black hole, it can be simplified as a flat disk that lies around the black hole. This approximation as a thin disk is also how the first visualizations were produced [36] [35]. They did not incorporate electromagnetic flow dynamics yet. For these visualizations, one needs to consider an infinitesimal small disk around a black hole. The inner radius is the last stable orbit $r_min = 3r_s$ and the outer radius has an arbitrary size r_{max}. The disk would be slightly inclined towards the observer. Because of the light bending around the black hole, the disk is hugely distorted and multiplied. The part of the disk which should supposedly be behind the black hole is bent above it as an arc. The bottom of the disk, which should not be visible either since the observer looks at the top of the disk, is visible as a second arc under the black hole.

5.10 Rotating Black Holes and Complex Caustic

In general, black holes are to be expected to have angular momentum due to their formation by the collapse of rotating stars. As angular momentum needs to be preserved, the material's angular momentum forming the black hole will add up to the angular momentum of the black hole. Those spinning black holes are

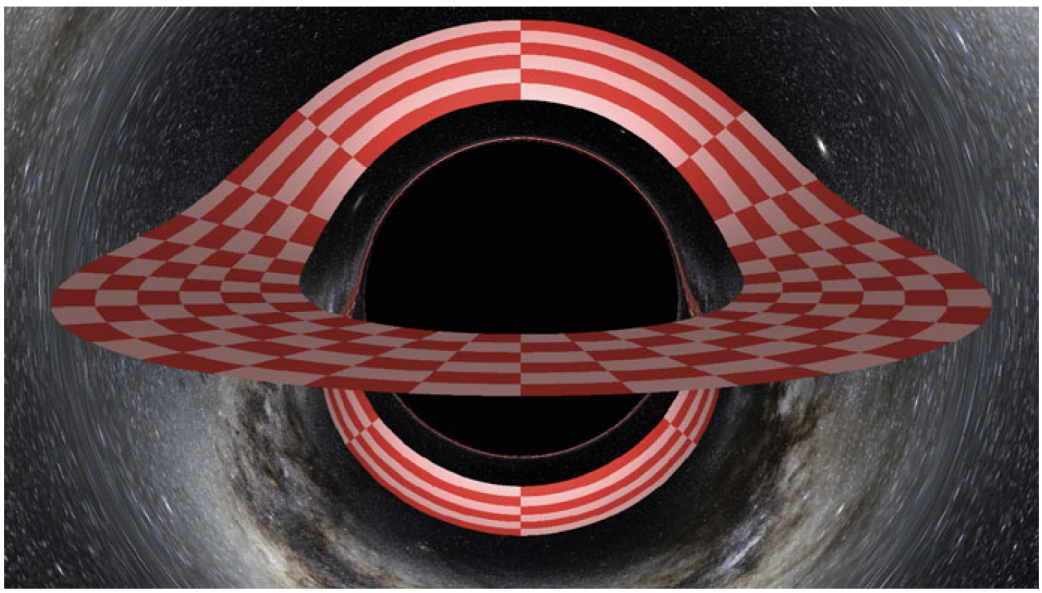

Figure 6: Checker board accretion disk in the equatorial plane of a Schwarzschild black hole. The observer sees the disk slightly from above. Taken from Boblest S., 2016 [18]

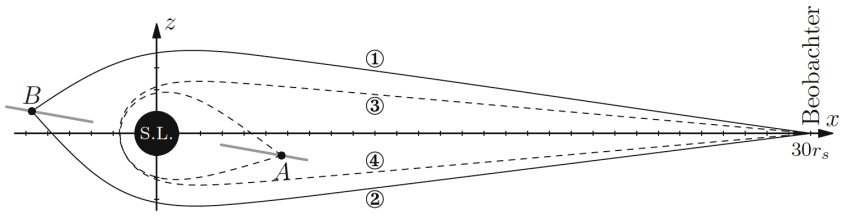

Figure 7: Four light rays emitted by the accretion disk at different places. Every light ray emitted by the disk reaches the observer. Taken from Boblest S., 2016 [18]

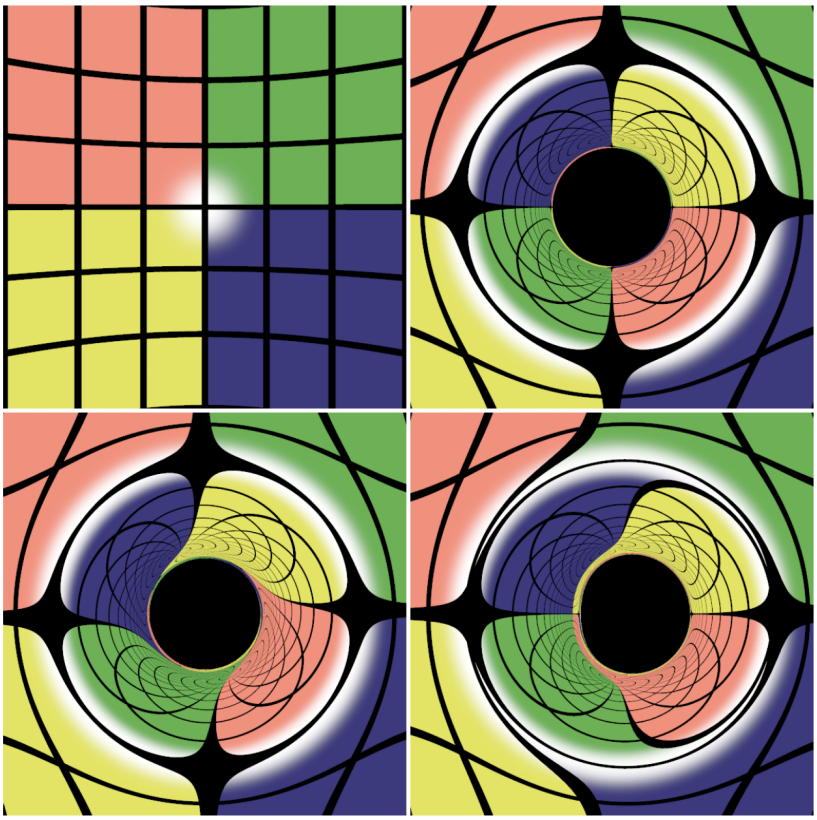

Figure 8: Lensing effect caused in different spacetimes. The upper-left image is a Minkowsky image. There is a slight distortion visible due to the fact that there is already a camera with 60% field of view applied. On the upper right is the metric of a Schwarzschild black hole. It is completely rotationally symmetric. In the lower row are two renderings of the metric of a Kerr black hole. Depending on which direction the camera points to, the center of the curvature produces different visual distortions. On the bottom left, the camera points to the spin axis of the black hole, and on the bottom right the camera view axis is perpendicular to the spin axis. Taken from Bohn A., 2015 [78].

called Kerr-Black holes and are described with the Kerr metric.[64] Consequently, this metric (or its solution to the Einstein equation) has no rotational symmetry which makes it harder to compute and to numerically analyse. The Schwarzschild black hole appears the same from every direction. The appearance of the Kerr black hole, on the other hand, differs according to the perspective from which the viewer looks at it. Looking at the Kerr black hole in the direction of the spin axis it seems round, but looking perpendicularly at the spin axis, the black hole does not appear to be perfectly round: It is bulged to one side (see Figure 8).

Due to the orbits of photons that turn around the black hole before they arrive at the observers eye, complex caustic effects appear. The caustics which were visible at the Schwarzschild black hole were the Einstein rings of differing order. At Kerr black holes, there are caustic pheonomena with higher complexity due to the broken nature of rotational symmetry. They are defined in the Kerr metric case as 2-dimensional surfaces separating regions of space in which a source would give rise to a different number of images [64] (see analog discussion of multiple images separated by Einstein rings in 5.4, 5.5). It was shown that the first order caustic is a tube with the cross-section having the shape of an astroid (closed curve with four cusps) bended around the black hole [63]. For details see [64] and also [56].

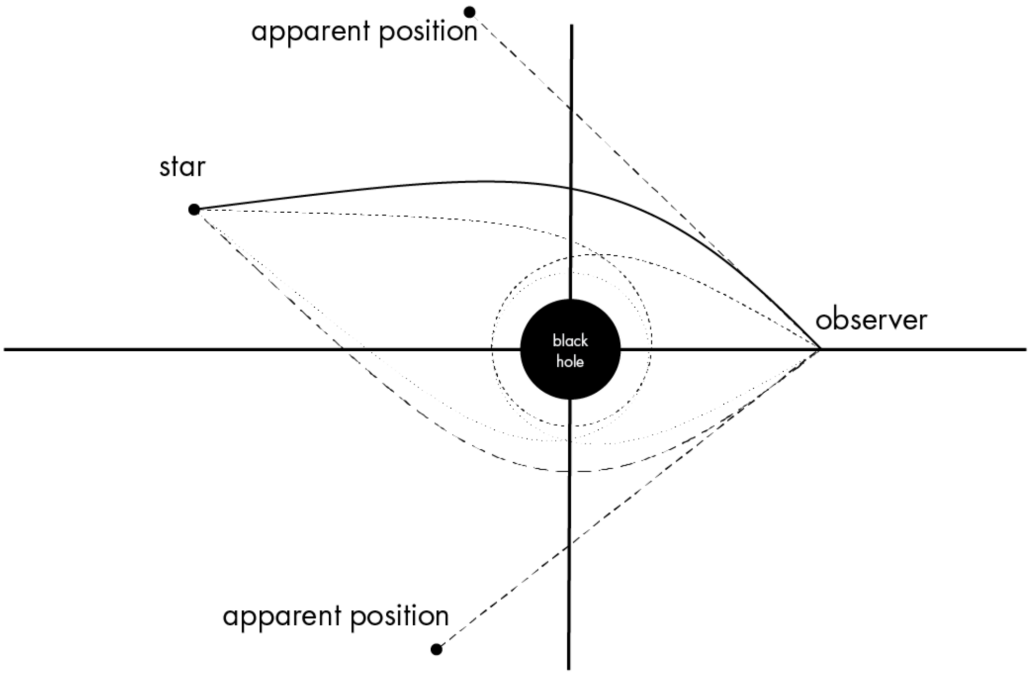

Figure 9: A point-like star is connected to the observer by multiple trajectories. In this case, there are two; in principle, there are infinite many. Depending on which path the light travels, it will hit the observers eye from different directions who interpolates the star light on a linear trajectory which gives the illusion that there are multiple stars at different positions than where it is actually located.

6 Ray Tracing

For understanding how light behaves in curved space and what we consequently would see in a general relativistic setting light-rays need to be traced in curved space. Their movement is governed by the geodesic equation. More exactly in ray tracing the light trajectory from a source e.g. a star is traced back in time to a virtual camera (the observer). In the following examples of ray tracing algorithms and achieved results are presented and compared, which is followed by a more detailed approach and an explanation of the method applied in the thesis' accompanied implementation.

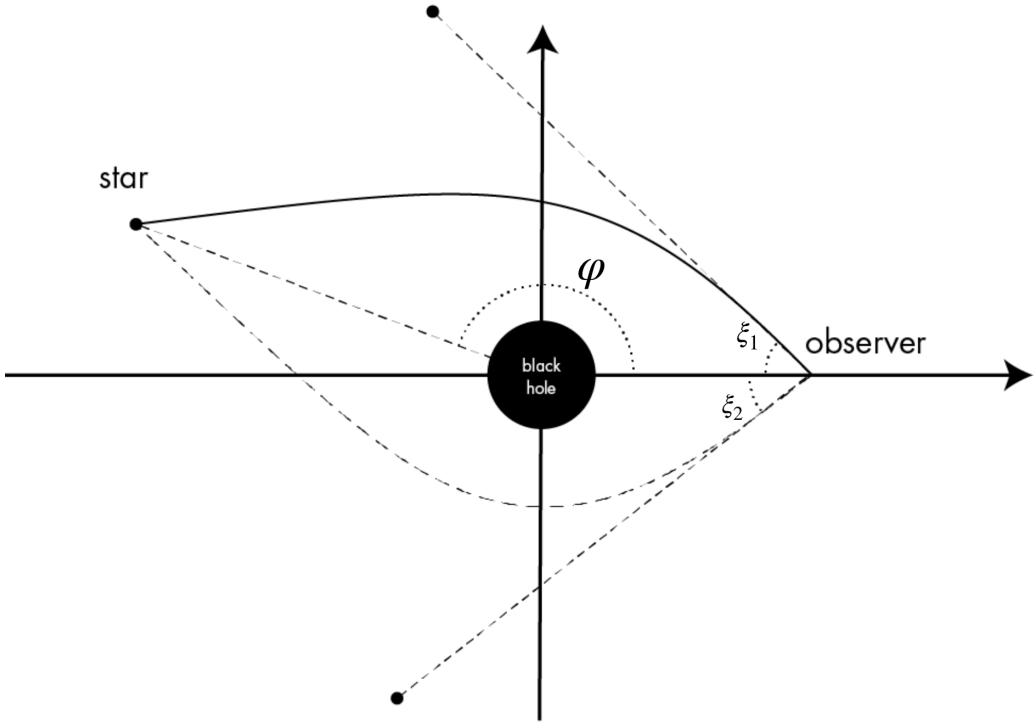

Figure 10: To describe the position of the original and the new star position the angles φ and ξ are used. φ is the angle between the direction angle to the original star position and the direction to the observer with the black hole in the center. ξ is the angle between the direction vector to the black hole and the vector to the new star position with the observer in the center.

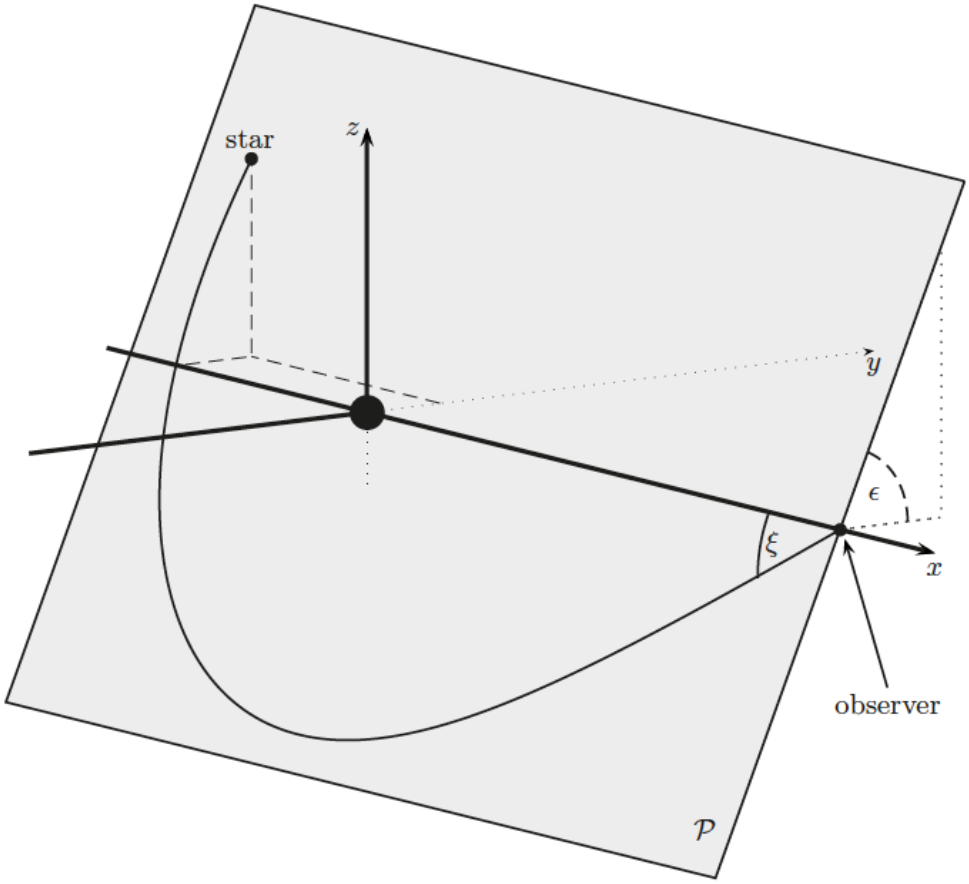

Figure 11: As the Schwarzschild spacetime is rotationally symmetric, the problem of tracing the light trajectory can be reduced to a 2d problem. The nullgeodesics (light trajectories) which connects the star and the observer lies on a plane on which the star, the observer and the black hole are located. Taken from Müller T., 2010 [46].

6.1 Comparing Ray Tracing Algorithms and Applications

Some of the earlier works which describe general relativistic ray tracing are by Cunningham [62] and Bardeen [22] that calculate the optical appearance of a star orbiting a Kerr black hole. Luminet [36] studied the appearance of an accretion disk around a Schwarzschild black hole.

Some other early general relativistic visualizations are by Nollert [34], who describes the appearance of a neutron star, and Weiskopf [53], who calculated phenomena like the general relativistic appearance of a rigidly rotating disk, a wormhole, and an Alcubierre warp metric.

Today there are a couple of comprehensive (open-source) ray tracing code libraries which implement numeric and analytic calculations with many different metrics. E.g. GYOTO [59] is a open-source ray tracing code written in C++. It calculates the appearance of massive objects in a general relativistic setting as well as the trajectories of massive objects within these settings. It is a fully numeric approach using the Hamiltonian formalism for its equations of motion to then integrate the equations with a Runge-Kutta algorithm. It is originally based on the Kerr metric, but due to its numeric nature, it can calculate all kinds of metrics within the 3+1 formalism of general relativity. In addition, it also takes in consideration the emission and absorption coefficients of the objects at hand and thereby gives a description of the radiative transfer and spectra which would hit the observer. Motion4d [50] is another C++ library integrating geodesic equations numerically and uses parallel transport to transport objects in a relativistic setting. It has a huge collection of different metrics to choose from. GeodesicViewer [41] is a UI application using the Motion4d library to visualize geodesics in different metrics. The geodesics can be explored in 2 and 3 dimensions and parameters can be changed interactively.

Besides the more comprehensive libraries, there are studies which focus on particular aspects of general relativistic visualization. A more recent paper [78] studied the appearance of a black hole merger which overcomes the challenge of solving for a highly complex and dynamic spacetime.

Another recent study is by Riazuelo [71]. They produced an exact rendering of the Schwarzschild metric by incorporating every special (aberration, amplification, Doppler) and general (deflection of light, lensing, gravitational redshift) relativistic physical effect and also provided an exact rendering of the sky image in the background. It aimed at visualizing the maximal analytical extension of the Schwarzschild metric and let the camera go deep inside the ergosphere, crossing the horizon and even venturing into the white hole region and illustrated the non-intuitive shape-shifts of these spacetime regions. In two other papers [72] [73] they explored the visual appearance of the (physically irrelevant) Reissner-Nordström black hole and the physically relevant Kerr black hole and

again explored the maximal extensions of these metrics.

Another interesting case are the research papers around the movie Interstellar [56] [77]. Kip Thorne consulted Christopher Nolan on the physical aspects of the movie. The team at the London based CGI studio Double Negative worked with the researchers from the California Institute of Technology to develop a general relativistic render engine (DNGR) which can perform the visualization of a Kerr-Black hole with an accretion disk and a wormhole in IMAX-quality. As such it was the first movie to show these objects backed up by physical reasoning to a consumer-oriented audience. In contrast to the other visualizations discussed above, it tracked ray-bundles instead of single light-rays back in time to achieve the high resolution and smoothness in camera motion.

Nearly all of these numerical schemes rely on Runge-Kutta algorithm. The downside to this is that it does not conserve energy and thereby does not provide highly accurate results at critical points (e.g. close at the horizon). In [54], different numerical schemes are compared and an energy conserving Hamiltonian integration is proposed.

In addition to the visualizations outlined here, there is a huge field of research concerning magneto-hydrodynamics (MHD) which aims to simulate high-energy accretion disk formation and ray-jet formation, but presenting them surpasses the aims of this thesis.

6.2 General Ray Tracing Approach

Visualizations help us understand how curved space appears from the point of view of a first-person observer in general relativity. A first-person view shows the image that a virtual camera would produce if it was set in a general relativistic space. For this case we need to extend standard 3d ray tracing methods to ray tracing in 4d spacetime. All visualizations follow a similar pipeline which we will call general relativistic ray tracing. In standard 3d ray tracing used in computer graphics, light rays are straight lines and have "infinite" speed. For each pixel on the screen of the virtual camera a light ray is cast back into the scenery and traced back as to when it hits an object or leaves the region of interest. At an intersection point with an object, material properties are taken into account as to calculate emission, absorption, reflection and refraction. This establishes the object's appearance on the 2d screen. In 4d general relativistic ray tracing, however, the speed of light is finite and the light does not follow straight lines but is curved following the curvature of space. [49] Light rays follow null geodesics which can be calculated numerically starting with the geodesic equation. In case of an inertial massive particle that is small enough that its extension can be neglected (which would otherwise also lead to curvature changes in its neighborhood), this object can be described as a 3d object in space being located at a local tetrad

that is transported with parallel-transport along a geodesic. Light-ray intersections with these particles can also be calculated, but the method to achieve this is very expensive computationally. In our case we are only interested in light trajectories and their movement in curved space. Other effects that also have to be taken into account in a special or general relativistic regime are Doppler-shift, gravitational frequency shift, gravitational lensing or geodesic precession.

To calculate the propagation of light in a general relativistic setting, we need to integrate the geodesic equation for the light ray within the given spacetime (light follows null-geodesics) for every pixel of the observers virtual screen (i.e. our camera). The integration starts at the observer and goes backwards in time until it reaches the emitter. The integration can be stopped if: the light ray hits an object, the light ray leaves the region of interest, the numerical integration becomes invalid or the defined time period becomes too long (at photosphere e.g.). The equation for the propagation of light is given by the geodesic equation (15). The local reference frame of a camera is given by the local tetrad (11). This gives us an initial direction from which we can start the integration of the geodesic equation. This approach is used in most cases where an arbitrary metric is given. In the case of the Schwarzschild metric, simpler approaches are possible which are used in the application of this work and were performed by [46].

6.3 Numerical Approach in this Application

In the case at hand - the Schwarzschild black hole - we can solve for the movement of the photons by an analytical approach. As Figure 11 illustrates, every trajectory lies on a plane as the metric is rotationally symmetric. This means we can suppress one angle and search for the equation of motion $r = r(\varphi)$. Using the geodesic equation for light geodesics (15), the Euler-Lagrange formalism with (13) and the Schwarzschild metric (16) (Eddington-Finklstein coordinates yield the same result) lead to the orbital differential equation of the system:[46]

$$\left(\frac{dx}{d\phi}\right)^2 = a^2 - (1-x)\,x^2 \tag{36}$$

with $a^2 = r_s^2/b^2 = r_s^2 k^2/(c^2 h^2)$ with b being the apparent impact parameter. The constants of motion are:

$$k = \left(1 - \frac{r_s}{r}c^2\dot{v} - c\dot{r}\right) \tag{37}$$

and

$$h = r^2\dot{\phi} \tag{38}$$

A dot refers to differentiation with respect to an affine parameter. $x = r_s/r$ is the scaled coordinate. An example is the case $x = x_0 = 2/3$ which is the photon

sphere. Here the null geodesic follows a circular orbit. Equation (36) is difficult to solve analytically, therefore, for the means of a computational visualization, we can approximate the result and solve for ϕ by numerically integrating:

$$\phi = \int_{x_{obs}}^{x_{star}} \frac{dx}{\sqrt{a^2 - (1-x)\,x^2}} \tag{39}$$

The ray tracing method is now to trace the null geodesics back in time. The stars are put at infinity $x_{star} = 0$. We solve for ξ with the following constants of motion which are derived from (37) and (38):

$$\frac{k}{c} = \sqrt{x - x_0}\cos(\xi) - \sqrt{1 - x_0} \tag{40}$$

$$h = \frac{r_s}{x}\sin(\xi) \tag{41}$$

In the final application the lookup table contains the inverse relation ξ as a function of the current position x_{obs} and the azimuthal angle φ_∞. See figure 13. Integral (39) is calculated for every position and (40) and (41) are used to calculate the apparent angle for every position and azimuthal angle.

7 Implementation

For the implementation of a black hole in virtual reality, the 2d application by Mueller [46] was used as a starting point. It was programmed in the C++ graphics API OpenGL [66]. For the virtual reality functionality OpenVR by Steam [67] was used. In the following, the basic graphics pipeline which computer graphics use in general and OpenGL uses in particular are discussed. Using this fundamental pipeline the black hole code is implemented. The original 2d implementation will be described followed by a detailed description of the changes and new functionalities in the 3d VR application.

7.1 Standard Rendering Pipeline

Computer graphics follow an established way of converting digital objects and their coordinates to pixels on the screen. This pipeline is conducted by the graphics card and can be divided into several steps. [70] One step relies on the output of the former step as its input. Every step is highly specialized and can be executed in parallel. Modern GPUs have therefore thousands of small processing cores to quickly process the data in parallel. The processing cores run small programs for each step of the pipeline, these programs are called shaders. The two main shader types are vertex and fragment shader. The vertex shader uses the vertices (3d points in 3d space) to project the primitives that they form on a 2d screen. This information is then used by the fragment shader to rasterize the image and give a corresponding color to every single pixel on the screen.

In the vertex shader the projection of the 3d points operate in different coordinate systems.

There are four different coordinate systems:

World coordinates This global coordinate system contains all objects and cameras

Model coordinates Every model has its own local coordinate system. The center of the coordinate system is typically in the center of the object.

Camera coordinates The camera has its own coordinate system and sits directly at the center of it, it looks at the negative z direction (OpenGL).

Projection coordinates In front of the camera is the view frustum, a "cut-off pyramid" with far and near clipping plane restricting its expansion. Its content

gets projected on the the final image using the projection coordinates. The x and y axis give the horizontal and vertical content of the image the z axis its depth.

Every coordinate system can be considered as a separate space. Transformations between these spaces allow to describe the digital content using different coordinates. These transformations are represented and calculated by 4d matrices. Every 3d point (x,y,z) is replaced by a 4d point (x,y,z,w), giving an extra coordinate labeled "w". This form is called homogeneous coordinates. The geometric interpretation is that the 3d space is represented as a slice in 4d at w $= 1$. This method allows all transformations (scaling, translation, rotation) to be linear and to be described by 4d matrices and is necessary for calculating projections between different spaces. A coordinate transformation by rotation and translation would look like:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} rot_{xx} & rot_{xy} & rot_{xz} & trans_x \\ rot_{yx} & rot_{yy} & rot_{yz} & trans_y \\ rot_{zx} & rot_{zy} & rot_{zz} & trans_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{42}$$

In camera space every point maps from (x,y,z,1) to the 4d projection space (x',y',z',w'). Therefore, to get to the actual coordinate, rescaling is necessary to (x'/w',y'/w',z'/w',1). This now gives the image of the view frustum as a box in 3d projection space.

In this shader pipeline outlined above, it is possible to use "textures"; these are typically 2d images used to change the appearance of vertices and pixels. Normally in computer graphics this is used to change the look of digital objects, to achieve the illusion of high details but still having a small vertex count or to add light without real-time illumination. Textures can be used to store all kind of information to be used to change the appearance of vertices. In the case of the black hole multiple textures are used to store information which would be computationally too heavy to calculate in real-time. So instead of using textures as images, we use textures to store information about vertices. This information can be read in real-time. Calculating this information in real-time would be too calculative expensive. In this case, the old and new angles of the star position before and after the distortion, which takes place due to the black hole lensing, are getting stored on a texture and are read by the shader. In addition, textures about the color space of the temperature of the stars are used.

Quaternions Complicated rotations in computer graphics are often solved by using Quaternions. They are also used in this application, and because of this they will be outlined shortly. The typical form of describing rotations are Euler

Angles. These are three angles per axis, and they describe how much rotation takes place around each axis. Although it is fairly intuitive, it is rather hard to smoothly interpolate between two differently oriented vectors in space. Generally, many calculations are difficult using Euler Angles. Quaternions are a way to describe angles and rotations which is far less intuitive but does not have those problems. Quaternions store the axis orientation around which the rotation takes place as well as the angle of rotation. The form is as follows:

```
// RotationAngle is in radians
x = RotationAxis.x * sin(RotationAngle / 2)
y = RotationAxis.y * sin(RotationAngle / 2)
z = RotationAxis.z * sin(RotationAngle / 2)
w = cos(RotationAngle / 2)
```

Accordingly, combining different rotations in space is far easier, and they will therefore be used in the application to achieve a parallax effect in the scene.

7.2 Rendering in Curved Space

The section about ray tracing already explored as to how the visualization in curved spaces is calculated, but it is worth noting that the standard rendering pipeline is already suitable for curved spaces and non-euclidean geometry. The transformation matrices in graphic programs are generally euclidean, but due to them being 4-dimensional, homogeneous coordinates, graphic cards have no problem with calculating curved spaces when new transformation matrices are provided which are based on different metrics. This has been explored in multiple works [70] [44] [84]. In these papers, ways of visualizing hyperbolic and spherical spaces are presented. There already are implementations of curved spaces within Virtual Reality [81] [82] [83], as well. These projects enable the user to visit hyperbolic H^3, spherical S^3 and two-dimensional hyperbolic, one-dimensional euclidean $H^2 \times E$ worlds by allowing the user to move freely within them.

7.3 OpenGL

To implement the application, OpenGL (Open Graphics Library) is used[66]. OpenGL is a library and interface (API) for rendering 2d and 3d graphics on the screen. It is designed in a way to interact with the GPU to achieve hardware accelerated rendering. The theoretical ideas outlined in 7.1 are used in a specific process in the OpenGL pipeline and will be explained in the following. The GPU needs vertices and fragments to render anything. The fragments are provided by

the GPU itself. At first, however, the vertices need to be defined on the CPU with C++ code before they are stored on the GPU. This happens by defining an array for the vertex data: Multiple sets of vertex arrays can be defined. For the vertices, we need to make room on the graphics card in the form of memory. This memory is managed by the so called "vertex buffer object" (VBO). We also need to declare what the GPU should do with these vertices. This is done by attributes. The attributes are stored in the Vertex Array (VAO). With VBO and VAO we can call our data in the code at any time. Finally, to process this data we define and compile a shader. This is the code which runs on the GPU and gets the vertices and their corresponding attributes as input. Another way of storing values on the GPU is using uniforms. These have some other use cases as attributes and enable us to bring floats, vectors and matrices to the shader and to use and change them in real-time. On top of that, we can also use textures which store data in a 2d image which is processed in a similar way and brought to the shader (higher dimensional textures are also possible, theoretically).

When we want to use and start a certain render pipeline, we bind our VBO, VAO and bind them to the GPU and link them to the shader program.

To sum up the basic workflow step by step:

- Loading data or defining our vertices so that we have an array of vertices.
- Then we create a vertex buffer (VBO) and fill it with our vertices, these get stored on the graphics card.
- Then we create a vertex array, this stores our attributes and tells the graphics card what to do with our vertices.
- Now we activate the shader.
- Now we set attributes (configuration gets stored in VAO) and then enable the attributes.

In OpenGL this basic workflow would look like [79] [80]:

```
// ..:: Initialization code :: ..
// 1. bind Vertex Array Object
glBindVertexArray(VAO);
// 2. copy our vertices array in a buffer
glBindBuffer(GL_ARRAY_BUFFER, VBO);
glBufferData(GL_ARRAY_BUFFER, sizeof(vertices),
    vertices, GL_STATIC_DRAW);
// 3. set vertex attributes pointers
```

```
glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE,
    3 * sizeof(float), (void*)0);
glEnableVertexAttribArray(0);


[...]

// ..:: Drawing code (in render loop) :: ..
// 4. draw the object
glUseProgram(shaderProgram);
glBindVertexArray(VAO);
someOpenGLFunctionToDraw();
```

7.4 Original 2d Implementation

The original implementation on which the VR implementation is based on is a 2d QT-Application by T. Müller 2010 [46]. It has a visual interface where the user can adjust the relative position to the black hole as well as adjustment setting whether the observer is static or freely falling. It is a real-time application which allows the user to see immediate changes of the black hole environment when he/she changes some settings. Therefore, it was an ideal start for a Virtual Reality application as real-time rendering is imperative for this technology. Other visualization tools are mostly using ray tracing (see chapter 6.1) and are therefore not directly applicable for VR visualizations.

To achieve real-time speed, the application uses shaders which store the information of the distortion effect as well as star magnification, the Planck spectra for the stars, apparent magnitude of the stars and Fraunhofer diffraction. The distortion effect was pre-calculated by storing the original star position and the new position for every angle in a lookup table. The necessary information was achieved by ray tracing the geodesic equation as described in chapter 6.3. The running application is depicted in Figure 12.

The background of the black-hole is the Hipparcos catalogue which stores the information of 118 000 stars which make up the part of the milky way which is visible from earth. It contains the right ascension for every star and declination angle as well as the apparent visual magnitude and temperature.

The catalogue is given as a binary file with 5 rows for every star corresponding to an index, right ascension, declination, parallax, visual magnitude and

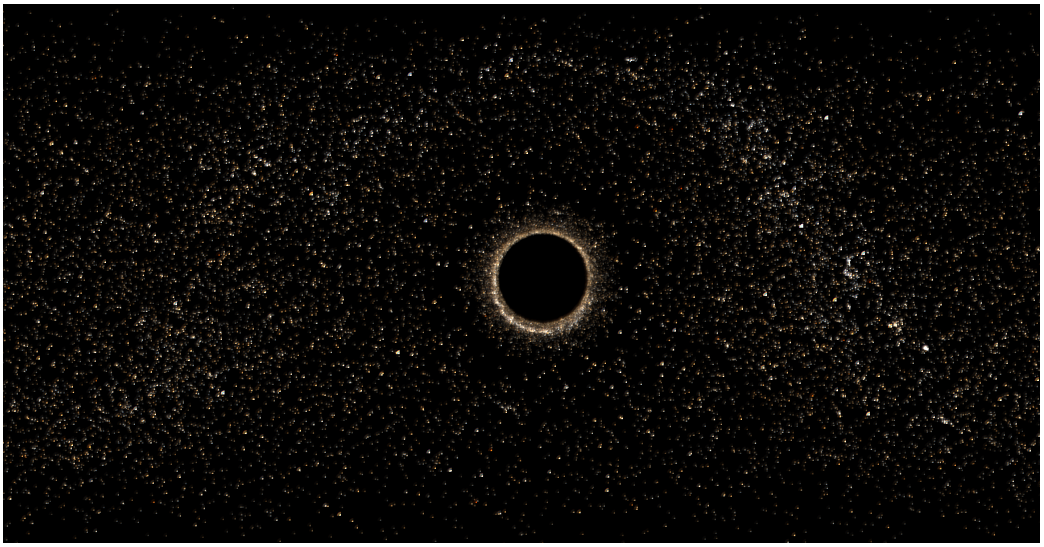

Figure 12: Screenshot from the original 2d application. The Hipparcos star catalogue showing the sky from the earth. It is projected as a mercator map on the screen. The application calculates the re-positioning of the stars and frequency and aberration changes to visualize the appearance of a Schwarzschild black hole in front of the stellar sky. The distance to the black hole can be manually changed as well as the the position of the background star map and whether the observer is static or falling freely. Taken from Müller T., 2010 [46].

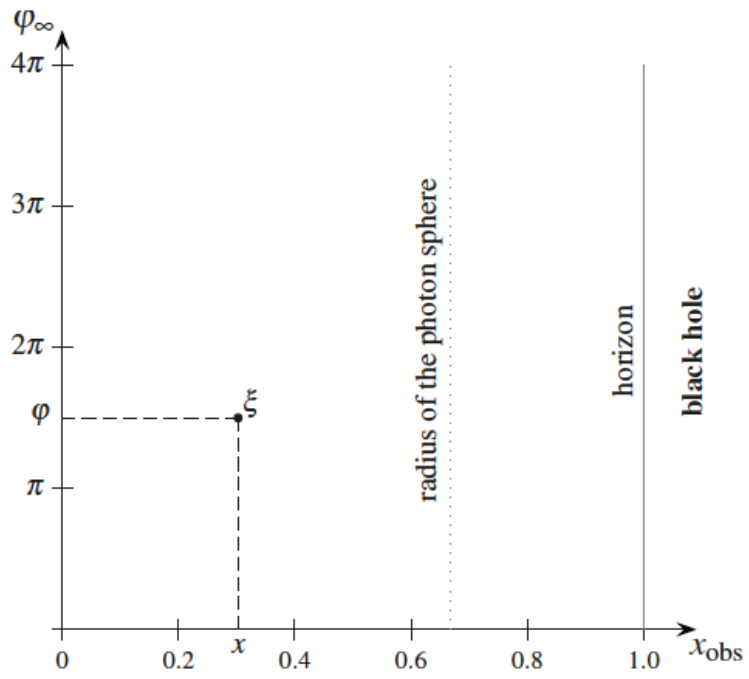

Figure 13: The lookup table for a static observer. For a given position x of the observer and an azimuth angle ϕ of a star gives the observed angle ξ. At $x = 2/3$ and $x = 1$ there is the photon sphere and the horizon. This lookup table is used by the shader as a texture file to read off the right star position for every observer position in every direction. Taken from Müller T., 2010 [46].

temperature. Within the code low magnitudes are sorted out. Temperature and magnitude are later used to calculate the stars' color on the screen.

The ascension and declination angle was used to calculate a 3d position of every star in space (by converting spherical coordinates into cartesian). These are the vertex points in the application which are used by the shader to calculate the varying effects. The vertex points are stored as arrays on the GPU as described in 7.3. In total, we have three sets of vertices: position, apparent magnitude and effective color temperature.

The lensing effect or distortion effect is the most prominent visual effect and was achieved through the numerical approach outlined in 6.3 The values that were found for ϕ and ξ in relation to the distance of the observer r_{obs} were stored in 2d lookup tables as depicted in Figure 13. For an observer position $r_{obs} \epsilon [r_s, r_{max}]$ and each star position $\phi_{star} \epsilon [0, 2\pi)$ the corresponding angular direction ξ_1 and ξ_2 for the two new apparent positions can be read off. To adjust this for the magnification effect outlined in 5.4, the lookup table also contained a factor for the magnification. This lookup table describes the situation $r_{obs} > r_r$. Another lookup table was provided which stored the necessary values after crossing the horizon. This is necessary as the observer cannot be static after crossing it and the magnification needs to be calculated with special relativity in mind (see 5.3). Depending on the position which is appointed by the user in the application, the shader will read the necessary values from either lookup table.

In addition to lensing and magnification, other visual effects that were explained in the previous chapters are calculated in the application. The relativistic Doppler effect and relativistic aberration are taken into consideration for a falling observer. For both cases the gravitational red-shift is also calculated and the color temperature changes accordingly. After all of these steps have taken place in the vertex shader, the vertices get processed further in the fragment shader where the color temperature is mapped to colors according to a predefined color table and it also smears all the stars to a Fraunhofer diffraction pattern caused by the finite aperture of a telescope which makes the stars appear more real.

7.5 Hardware and Setup

Due to the implementation in OpenVR, (see next chapter) any VR Headset can be used that supports 6 Dof (Degrees of freedom). We tested the application with the HTC Vive, Windows Mixed Reality, Oculus Rift and Oculus Quest (with link cable). Other high-end HMDs like the Valve Index should also be compatible because of the support of all common headsets. For the application to run, SteamVR has to be installed as the communication with the headset driver runs through it. As to the GPU, a more high-end version is required which is due to the nature of VR itself. The black-hole application is not more demanding than any

other standard VR applications. I would recommend a Nividia graphics card like gtx 960 or higher. I cannot assure that the application works with AMD Radeon GPUs. Testing it on one of them it unfortunately failed to render the stellar sky. But it was not tested on more Radeon GPUs, so it remains unclear whether this was an anomaly or they generally fail to render the application correctly.

7.6 OpenVR

OpenVR is an API developed by Valve to support any VR Headset[67]. It is the default system for the HTC Vive Headset, but it supports any vendor, which makes it extremely beneficial and flexible to use. The API is implemented as a set of C++ interface classes with virtual functions. It serves as the interface between VR hardware and software. The runtime, though, is SteamVR, which is not open-source but is necessary for the communication between driver, headset and executable.

OpenVR features support for OpenGL, DirectX as well as Vulcan which are the APIs to interact with a graphics processing unit (GPU) to achieve hardware-accelerated rendering. In our case OpenGL is used. For handling output, window and input interfaces, OpenVR uses SDL[68]. For algebra calculations the library GLM [69] is used. It offers many handy functions for typical vector calculations that are common in computer graphics.

7.7 Implementation in 3d and Virtual Reality

For the application we used the example code "hellovr_opengl_main" from OpenVR as a boilerplate template. This scene renders a standard scene which consists out of a grid of cubes in space. Within the code, there already exists the possibility to call for the position and rotation of the headset and the controllers. The cubes were deleted and a blank new scene ready to be filled remained. We translated and adjusted the functions necessary for the black hole rendering which were used in the original 2d implementation within this sample code and implemented other functionality which was necessary to achieve a 3 dimensional scene.

We introduced a camera position which used the OpenVR provided information about the position of the user. In the shader the projection matrix (see projection coordinated in 7.1) was changed from orthogonal to perspective. This change provides a sense of depth and perspective to a 3d environment.

In the original implementation, the position of the observer was defined only by the radial direction in relation to the black hole. The star background could be re-positioned horizontally by dragging the background with the mouse. By using the x, y, z position of the camera and Pythagoras' theorem, the radial distance r_{obs} was calculated for every frame to get the distance to the black hole, thereby

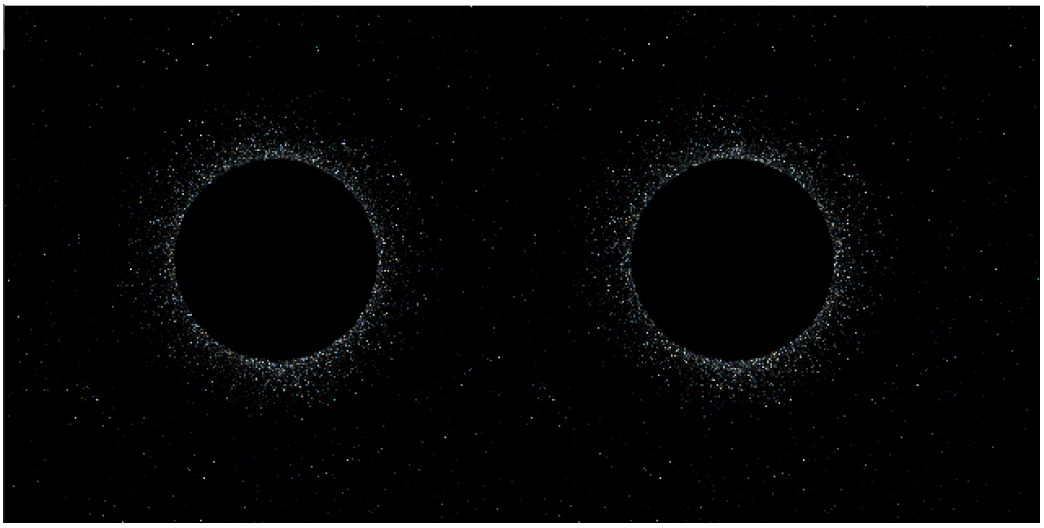

Figure 14: A screen capture of the application window on the desktop screen when the application is running on the VR device. Both images on the left and right are combined in the headset to a stereoscopic image leading to the perception of depth for the observer who is wearing the headset. The observer can move freely around the black hole like a 3d object in space with the star changing position in alignment with the position of the observer according to the lens distortion effect.

adjusting the perceived size of the black hole for every frame and movement. The position of the star map in the background was kept fixed at first, but its position will be changed indirectly in the shader for a parallax effect. To the user, though, it will appear static at every time.

The star map was generated with the Hipparcos catalog. They were originally mapped with a Mercator projection on the screen which is not necessary in a 3d environment. The ascension and declination angles were kept and projected on a sphere with large enough radius to feel like a skybox for the user. To achieve the feeling of a skybox with stars which are infinitely far away, the star positions were constantly (every frame) added to the camera's position so they would not change their position in relationship to the user. This introduced a major problem. The black hole is "only" the redistribution of the stars, it is not an object in space, it is a visual distortion of the sky (in reality as well as in the application). This led to the fact that the black hole would now be arbitrarily far away. To accommodate and change that, we had to introduce and implement a parallax effect that would change the black hole's position in relation to the skybox when the player moves around the black hole. Thereby the illusion is created that the black hole is an object in 3d space around which the player can move.

In the following I will go over the functions that were used, added and changed in the process in more detail. For clarity see Figure 15 as a roadmap.

Setting up the Scene and the Vertices First, `BInitGL()` calls all the setup functions for the scene which again is called in the `main()` function of the code. There are a couple of setup functions that come with the OpenVR scene in general; I will only go through those that were added to the original ones.

`SetupBlackHole()` is called in `BInitGL()` alongside other setup functions (like for the camera or the companion window). In this method, the Hipparcos catalogue is loaded, and all the textures necessary for the shader are getting initialized. The Hipparcos catalogue file yields three different variables: mVerts, mMagnitudes, mTemperatures, which are stored as OpenGL specific GLfloat arrays. Then there are the textures storing the lookup tables for the lensing effect. The information gets stored as type GLuint arrays. Those functions which read the files are `readHipBin( string )` and `initTextures()`.

Still within the `SetupBlackHole()` function, variables for the Hipparcos vbos are created and the buffer gets filled with the corresponding vertices. Vaos are created following the pipeline outlined in chapter 7.2. We have now three sets of vertices: star position, star temperature and star magnitude.

Rendering the Scene After the setup of the scene is finished we enter the main render loop. OpenVR calls for some specific functions that are necessary for virtual reality in general (like `RenderSteroeTarget()` to render the scene to both eyes). Custom code is found in `RenderScene()` and `RenderBlackHole()`.

Setting black hole specific uniforms for the shader happens in the method `RenderBlackHole()`. Here, the other variables that are provided by the lookup table will be bound as uniforms as they are meant to change the vertices as attributes. These lookup uniforms are ksiArray, ksiInsideArray, textSigma and textPsiTemp.

The other uniforms that will be necessary for shader calculations include: obsDist, x0, a, b, ksiCrit, beta, gamma, defgamma, deff0, defmaxpointsize, order. obsDist is the distance from the observer to the coordinate axis, which is the position to the black hole and is given by the calculated radial distance to the center of the coordinate system (location of the black hole). defmaxpointsize gives the size of the star position vertices which is defined once and kept. order is important for the lensing effect. The star will appear at several new positions (see figure 9). Order $= 0$ gives the first position and order $= 1$ the second. In reality there would be many more orders (see 5.4), but higher orders would be too close to the black hole to be visible in good resolution and are therefore rejected. The other uniform variables are used to calculate frequency shifts and relativistic aberration.

Generally values that change with every frame will be calculated in the render functions. Important values changing every frame are of course the HMD position and rotation. These are given by OpenVR as entries in matrices. Binding them happens in the method `RenderScene()`. They are getting used by other shaders besides the black hole, therefore they are not included in the `RenderBlackHole()` method. In `RenderScene()`, shaders are getting activated and the necessary variables are getting bound. The current position of the HMD gets called via `GetCurrentPosition(nEye)`. This method takes the 4x4 matrix elements 12,13,14 from `GetHMDMatrixPoseEye(nEye)`. The other matrix elements describe the orientation of the headset; these are not important in our case. The information is translated to a 4-vector and stored as a uniform called campos. The other necessary real-time variable is a rotation matrix helping to facilitate the parallax effect. This matrix is calculated by the function `calcParallaxRotation()`. To achieve the parallax effect, two vectors were established: one of them a forward vector, which has only one component in the z direction which always points to the position of the black hole on the skybox; the other vector spans from the camera position to the center of the coordinate system (where the black hole will be located). What we want to obtain is a transformation matrix that rotates

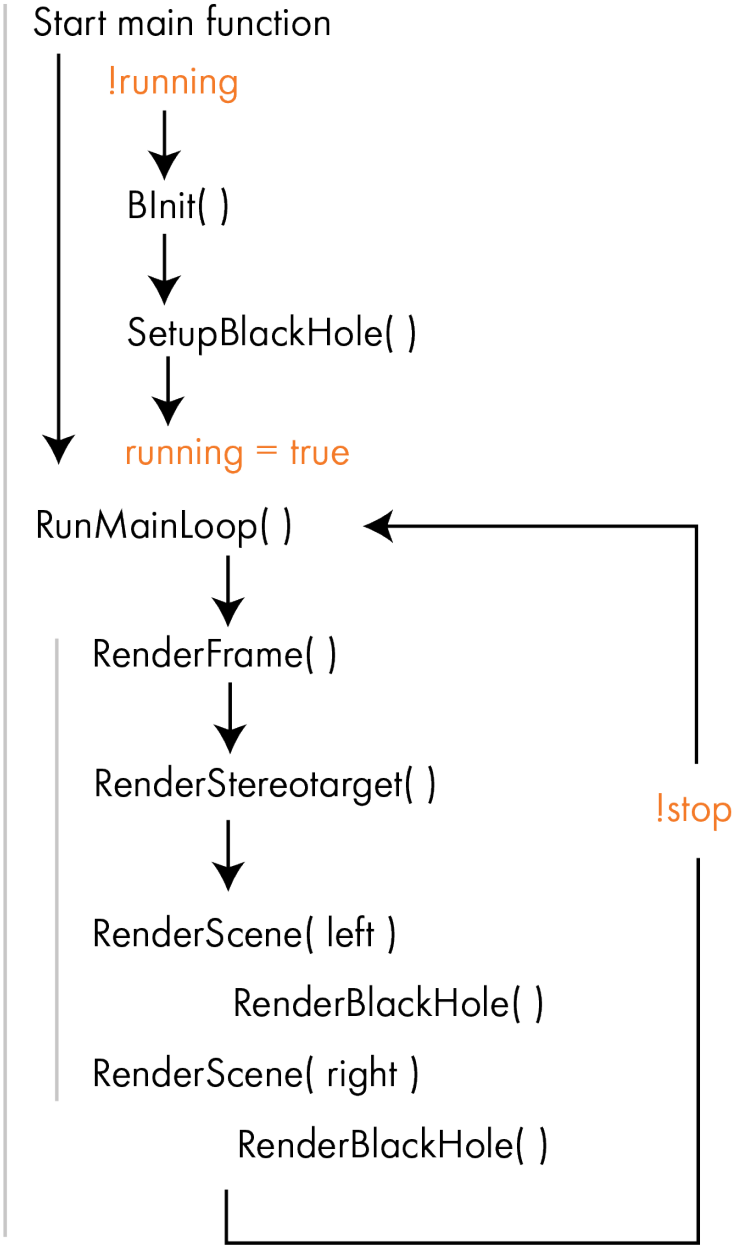
Start main function
!running
BInit( )
SetupBlackHole( )
running = true
RunMainLoop( )
RenderFrame( )
RenderStereotarget( )
RenderScene(left)
RenderBlackHole( )
RenderScene(right)
RenderBlackHole( )
!stop

Figure 15

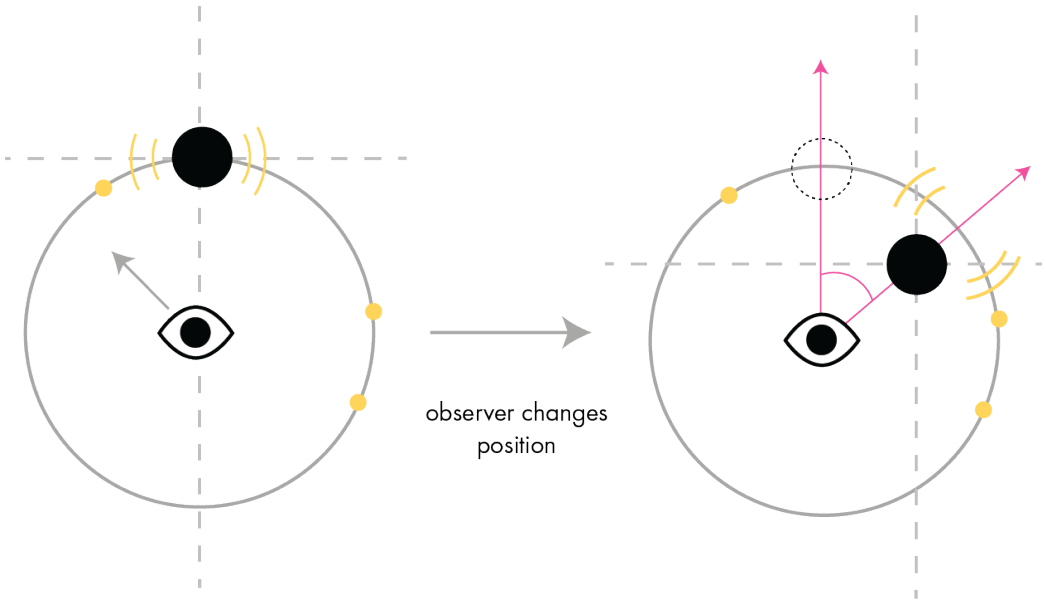

Figure 16: The black hole distortion was originally fixed on the skybox. To achieve the illusion that the black hole is an object in space we need a parallax effect. To obtain a parallax effect two vectors and the rotational transformation between those are necessary. Those vectors are depicted in pink. The skybox is the grey sphere around the observer in the middle, the stars are the yellow images on the skybox, those need to stay at the same position but the distortion of the skybox changes with the change of position by the observer. The black hole resides in the center of the global coordinate system.

between these two vectors. This transformation matrix will then rotate the stars background back and forth with the black hole changing its position but the stars remaining at their position. This transformation matrix is calculated by using Quaternions. The GLM library has a function `rotation(vector1, vector2)` which gives a matrix that facilitates the rotation between two given vectors. This matrix gets inverted. Those two matrices - the rotation and the inverted rotation are called parallax and invparallax (which get updated for every frame with every new position) – are brought into the shader during rendering, where they are now used as follows: Firstly, the undistorted skybox is rotated with parallax, then the star vertices run through the function that calculates the black hole distortion, and then those "distorted" stars are rotated back with invparallax to their original position. This is how the stars stay at their position while the distortion effect changes its position, which leads to the illusion that the black hole changes its position, meaning it stays in the center of the coordinate system when the player walks around. See Figure 16 for a graphical illustration of the parallax effect.

Lastly, there is the projection matrix that gets bound. This matrix is also provided by OpenVR via `GetCurrentViewProjectionMatrix(nEye)` and is called mvpmatrix.

To sum up the different functionality of `RenderScene()` and `RenderBlackHole()` is therefore: In `RenderScene()` more global variables are calculated, such as position and rotation of the headset, and it also calls for `RenderBlackHole()`. In `RenderBlackHole()`, only variables concerning the black hole physics are calculated and bound to the shader (see Figure 15).

Shader programming Most of the physically important calculations happen within the vertex shader itself; of these the most important steps are:

```
// ..:: Initialization code :: ..

[...]

float ra = aPos.y; //star right ascension
float de = aPos.z; // star declination
vec4 dir = vec4(cos(de)*cos(ra),cos(de)*sin(ra),sin(de),0.0);

dir = parallax * dir;
```

```
float newRa, newDe, phi, newKsi;
//gives new_dir:
calcNewRaDec(dir.xyz, newRa, newDe, phi, newKsi);
new_dir = invparallax * new_dir;
// skybox changes position with cam
vert.xyz = new_dir.xyz * radius + camPos;
//set the final vertex position:
gl_Position = mvp_matrix * vec4(vert.xyz, 1.0);

[...]

// ..:: calcNewRaDec gives lensing distortion :: ..

vec3 ex, ey, ez;
ex = vec3(1.0,0.0,0.0);
ez = normalize(cross(ex, dir));
ey = normalize(cross(ez, ex));

 phi = acos(dot(dir, -ez));

newKsi = texture(ksiInsideArray, vec2(obsDist, phi)).x;
newKsi = abberration(newKsi);
r = cos(newKsi)*ex + sin(newKsi)*-ey;
new_dir = vec4(r.xyz, 0.0);
```

This snippet contains the steps for the lensing distortion. The star positions are allocated as the angles which make up the skybox and are arranged on a sphere. Then the displace takes place by first making a parallax rotation as described earlier, and then the vertex positions are given to a function that reads out the new angle for every star. Firstly, a tetrad is established at the position of the observer. By taking the $\cos^{-1}$ of the dot product between observer position vector and direction of the star, we acquire the angle ϕ. See Figure 10 for the relationship of these angles. With ϕ, the new angle and thereby the new position of the star can be read from the texture (see Figure 13). A small aberration is calculated for every position, and by the aid of the vector tetrad the new angles are brought in a 4-vector that contains new information of the star position in cartesian coordinates. Now the skybox gets rotated again by an inverse rotation of the parallax matrix. The star positions are then scaled by a radius value to make them appear further away and are added to the camera position to make sure they stay in place, working as an actual skybox. Lastly the vertices

are multiplied with the projection matrix which ensures that the scene has the perspective of a proper 3d scene. There are a couple of other calculations that take place in the vertex and fragment shader taking care of the other visual effects discussed before. They follow the calculations already introduced earlier. The vertex positions are then forwarded to the fragment shader where they are mapped to a color table. This step is identical to the original 2d implementation.

8 Conclusion

The visualization of the close environment of a Schwarzschild black hole was brought into virtual reality. An original implementation in 2d was used as a starting point. The ray tracing of the light geodesics is not performed in the application, but was indirectly accessed with lookup tables. Those could be read by the GPU in real-time which enables the visualization of the black hole in real-time, which would otherwise have been not possible due to the complexity of geodesic equations in curved space. The finished application conveys the feeling of a 3d space in which the user can move around the black hole and observe the visual effects. The lensing and distortion effect are clearly visible. Frequency shifts are hardly visible in the application, which is probably due to the rendering abilities of the headset (the frequency changes are only visible as a slight change in the color of the stars and thus not very dominant). Further steps of development could include a physically realistic accretion disk. This would help to visualize the change in color frequency better and how particles behave in an accretion disk around a black hole.

The application relies on pre-existing lookup tables which where taken from another applications. This is something which could be improved by generating own lookup tables during a next stage of development.

Another interesting idea would be to change the environment from the star background to an every-day scene so that one could observe how a black hole would distort the environment we live in. Thinking this further, even an Augemented Reality (AR) or Mixed Reality (MR) application could be possible where one wears AR/MR glasses and looks at ones environment in real-time with a black hole right in front of the observer. Other metrics might also be introduced. The lookup table would need to be filled by other results of ray tracing calculations. The Kerr-metric would be challenging as it has no rotational symmetry. A metric like the Morris-Thorne wormhole, on the other hand, could be implemented in a similar fashion due to its symmetric nature. The distortion and redistribution of vertices would act on two skymaps in that case. Generally, porting the application into a game engine might also be reasonable as opposed to bare OpenGL code because that could enable to create a more flexible application which is

easier to augment with more functionality.

Besides curved space in general relativity another use case could be the visualization of special relativistic movement and how the environment would change by velocities near the speed of light. The movement of a VR user could be mapped to velocities near the speed of light. This would enable a far more intuitive sense of the change of physics at very high velocities. Because, again, it would be interactive and therefore more immersive and direct to the user. This really shows that physically extreme environments that are impossible for humans to observe can be visualized in VR and be observed from a first-hand-perspective. It helps us to visit and experience "strange" worlds. So on the whole, virtual reality offers innovative ways of depicting abstract mathematical and physical ideas and theories. It is helpful for scientist and non-specialists alike to achieve an intuitive understanding of spacetime regions which otherwise could not be perceivable in our "normal" reality.

References

[1] P. Schneider, J. Ehlers, E.E. Falco. *Gravitational Lenses*. Astronomy and astrophysics Library, Springer, 1999.

[2] Michell, J. 1784, "On the means of discovering the distance, magnitude, &c. of the fixed stars...," Trans. R. Sooc. London, 74, 35, reprinted in: Black holes: Selected reprints, S. Detweiler (ed.), Am. Assoc. of Physics Teachers, Stony Brook, NY, 1982.

[3] Will, C.M. *Henry Cavendish, Johann von Soldner, and the deflection of light*. Am. J. Phys., 56, 413., 1988

[4] Laplace, P.S. *Exposition du système du monde*. 1795.

[5] Einstein, A. *Erklärung der Perihelbewegung des Merkur aus der allgemeinen Relativitätstheorie*. Sitzungber. Preuß. Akad. Wissensch., erster Halbband, p.831, 1915.

[6] Eddington, A.S. *Space, time and gravitation* Cambridge University Press, Cambridge, 1920.

[7] M. H. van Kerkwijk, D. L. Kaplan *Isolated neutron stars: magnetic fields, distances,and spectra*. Astrophysics and Space Science, 2007.

[8] Penrose, R. *The Road to Reality. A Complete Guide to the Laws of the Universe.* Jonathan Cape (1st edition), 2004

[9] Reid et al. *The Trigonometric Parallax of Cygnus X-1.* Astrophysical Journal, 2011.

[10] Orosz et al. *The Mass of the Black Hole in Cygnus X-1.* Astrophysical Journal, 2011.

[11] Gou et al. *The Extreme Spin of the Black Hole in Cygnus X-1.* Astrophysical Journal, 2011.

[12] S. Bowyer et al. *Cosmic X-ray Sources.* Science, 1965.

[13] R. Narayan *Evidence for the Black Hole Event Horizon.* arXiv, 2003.

[14] R. Narayan, J. E. McClintock *Observational Evidence for Black Holes.* arXiv, 2013.

[15] The Event Horizon Telescope Collaboration *First M87 Event Horizon Telescope Results. I. The Shadow of the Supermassive Black Hole.* The Astrophysical Journal Letters, 2019.

[16] The Event Horizon Telescope Collaboration *First M87 Event Horizon Telescope Results. V. Physical Origin of the Asymmetric Ring.* The Astrophysical Journal Letters, 2019.

[17] C.W. Misner, K.S. Thorne, J.A. Wheeler *Gravitation.* Freeman and Company

[18] Boblest, S., Müller, T., Günter Wunner *Spezielle und allgemeine Relativitätstheorie.* Springer Spektrum, 2016

[19] K. Schwarzschild *Ueber das Gravitationsfeld eines Massenpunktes nach der Einsteinschen Theorie.* dtsch. Akad. Wiss. Berlin, 1916.

[20] A. Einstein *Die Grundlagen der allgemeinen Relativitätstheorie.* Ann. Phys, 1916.

[21] S. Chandrasekhar *The Mathematical Theory of Black Holes.* Clarendon, Oxford, 1983.

[22] Cunningham, C. T., Bardeen, James M. *The Optical Appearance of a Star Orbiting an Extreme Kerr Black Hole.* Astrophysical Journal, 1973.

[23] Y. Avni, I.Shulami *Flux Conservation by a Schwarzschild Graviational Lens.* Astrophys. J., 1998.

[24] S. Liebes Jr. *Gravitational Lenses.* Phys. Rev. B., 1964.

[25] S. Refsdal *The Graviational Lens Effect.* Mon. Not. Roy. Astron. Soc., 1964

[26] A. Einstein *Lens-like Action of a Star by the Deviation of Light in the Gravitational Field.* Science, 1936.

[27] H. C. Ohanian *The Black Holee as a Gravitational "Lens".* Am. J. Phys., 1987.

[28] R. Borsato, S. Vandoren *Student Book on Black Holes* http://www.staff.science.uu.nl/ vando101/BH.pdf

[29] J. N. Hewitt et al. *Unusual Radio Source MG1131+0456 - A Possible Einstein Ring.* Nature, 1988.

[30] Don N.Page and Kip S.Thorne *Disk-Accretion onto a Black Hole. Time-Averaged Structure of Accretion Disk.* American Astronomical Society, 1974.

[31] John Castor *Lectures on radiation hydrodynamics.* 1999.

[32] Mihalas & Mihalas *Foundations of radiation hydrodynamics.* 1984, 1999, Dover Publications.

[33] Albert Einstein *On the Electrodynamics of Moving Bodies.* 1905, revised and translated in The Principle of Relativity (Dover, NY, 1923).

[34] Nollert, H.-P., Ruder, H., Herold, H., & Kraus, U. *The relativistic 'looks' of a neutron star.* Astronomy and Astrophysics, 1989

[35] Jun Fukue, Takushi Yokoyama. *Color Photographs of an Accretion Disk around a Black Hole*. Publ. Astron. Soc. Japan, 1988.

[36] J.-P. Luminet. *Image of a Spherical Black Hole with Thin Accretion Disk*. Astron. Astrophys., 1979.

[37] R.D. Blandform, R. Narayan. *Cosmological applications of gravitational lensing.*

[38] Andrew J. S. Hamilton, Gavin Polhemus. *Stereoscopic visualization in curved spacetime: seeing deep inside a black hole.* New Journal of Physics, 2010.

[39] S.Krasnikov. *Falling into the Schwarzschild black hole. Important details.* arXiv, 2008.

[40] Robert J. Nemiroff. *Visual Distortions near a Neutron Star and Black Hole.* American Journal of Physics, 1993.

[41] Daniel Kuchelmeister, Thomas Müller, Marco Ament, Günter Wunner, Daniel Weiskopf. *GPU-based four-dimensional general-relativistic ray tracing.* Computer Physics Communications, 2012.

[42] Dimitrios Psaltis, Tim Johannsen. *A Ray-Tracing algorithm for spinning compact object spacetimes with arbitrary quadrupole Moments. I. Quasi-Kerr Black Hole*. The Astrophysical Journal, 2010.

[43] Jason Dexter, Eric Agol. *A Fast New Public Code For Computing Photon Orbits In a Kerr Spacetime.*. APJ, 2005.

[44] Charles Gunn. *Discrete Groups and Visualization of Three-Dimensional Manifolds*. 1993.

[45] Kip S. Thorne. *Disk-Accretion onto a Black Hole II. Evolution of the Hole*. The Astrophysical Journal, 1974.

[46] Thomas Müller, Daniel Weiskopf. *Distortion of the Stellar Sky by a Schwarzschild Black Hole*. American Association of Physics Teachers, 2009.

[47] Thomas Müller. *Exact geometric Optics in a Morris-Thorne Wormhole Spacetime*. Physical Review, 2008.

[48] Thomas Müller *Falling into a Schwarzschild Black Hole.* Gen Relativ Gravit, 2008.

[49] Thomas Müller and Daniel Weiskopf *General-Relativistic Visualization.* Computing in Science & Engineering, 2011.

[50] Thomas Müller, Frank Grave *Motion4D - A library for lightrays and timelike worldlines in the theory of relativity*. Computer Physics Communications, 2009.

[51] Thomas Müller, Frank Grave *GeodesicViewer - A tool for exploring geodesics in the theory of relativity*. Computer Physics Communications, 2010.

[52] W.L. Smith, R.B.Mann *Formation of Topological Black Holes from Gravitational Collapse*. 2018.

[53] Daniel Weiskopf *Four-Dimensional Non-Linear Ray Tracing as a Visualization Tool for Gravitational Physics*.

[54] F.Bacchini, B. Ripperda, A.Y. Chen, L. Sironi *Generalized, energy-conserving numerical simulations of partivles in general relativity. I. Time-Like and Null Geodesics*. 2018.

[55] P.Chris Fragile, Omer M. Blaes, Peter Anninos, Jay D. Salmonson *Global General Relativistic Magnetohydrodynamic Simulation of a Tilted Black HOle Accretion Disk*. The Astrophysical Journal, 2007.

[56] Oliver James, Eugénie von Tunzelmann, Paul Franklin, Kip S. Thorne. *Gravitational Lensing by Spinning Black Holes in Astrophysics, and in the Movie Interstellar*. Classical and Quantum Gravity, 2015.

[57] Oliver James, Eugénie von Tunzelmann, Paul Franklin, Kip S. Thorne. *Visualizing Interstellar's Wormhole.*. Am. J. Phys., 2015.

[58] Chi-Kwan Chan, Dimitrios Psaltis, Feryal Özel. *GRay: A Massively Parallel GPU-Based Code for Ray Tracing in Relativistic Spacetimes.*. 2013.

[59] F. H. Vincent, T. Paumard, E. Gourgoulhon, G. Perrin. *GYOTO: a New General Relativistic Ray-Tracing Code*. 2011.

[60] U.Kraus *Light Deflection Near Neutron Stars*.

[61] Eduard Gröller *Nonlinear Ray Tracing: Visualizing Strange Worlds*. The Visual Computer, 1995.

[62] C. T. Cunningham *Optical Appearance of Distant Objects to Observers Near and Inside a Schwarzschild Black Hole*. Physical Review, 1975.

[63] Kevin P. Rauch, Roger D. Blandford. *Optical Caustics in a Kerr Spacetime and the Origin of Rapid X-Ray Variability in Active Galactic Nuclei.* The Atrophysical Journal, 1994.

[64] V. Bozza. *Optical Caustics of Kerr Spacetime: The Full Structure*. 2008.

[65] Andrey Cadez, Uros Kostic. *Optics in the Schwarzschild Space-Time*.

[66] Khronos Group. Official OpenGL Documentation: https://www.khronos.org/registry/OpenGL-Refpages/gl2.1/xhtml/

[67] OpenVR SDK/SteamVR by Valve Corporation: https://www.steamvr.com/de/ Documentation: https://github.com/ValveSoftware/openvr/wiki/API-Documentation

[68] SDL by Valve Corporation https://libsdl.org/

[69] GLM library by G-truc https://glm.g-truc.net/0.9.9/

[70] Jeff Weeks. *Real-Time Rendering in Curved Spaces*. 2002.

[71] Alain Riazuelo. *Seeing Relativity - I. Ray Tracing in a Schwarzschild Metric to Explore the Maximal Analytic Extension of the Metric and Making a Proper Rendering of the Stars*. 2018.

[72] Alain Riazuelo. *Seeing Relativity - II. Revisiting and visualizing the Reissner-Nordström metric*. International Journal of Modern Physics, 2019.

[73] Alain Riazuelo. *Seeing Relativity - III. Journeying within the Kerr metric toward the negative gravity region*. 2020.

[74] Joachim Schastok, Michael Soffel, Hanns Ruder, Manfred Schneider. *Stellar Sky as Seen from the Vicinity of a Black Hole*. American Journal of Physics, 1987.

[75] Steve Bryson *Virtual Spacetime: An Environment for the Visualization of Curved Spacetimes via Geodesic Flows*.

[76] Thomas Müller *Visual Appearance of a Morris-Thorne-Wormhole*. American Journal of Physics, 2004.

[77] Oliver James, Eugénie von Tunzelmann, Paul Franklin, Kip S. Thorne *Visualizing Interstellar's Wormhole*. American Association of Physics Teachers, 2015.

[78] Andy Bohn, William Throwe, Francois Hebert, Katherine Henriksson, Darius Bunandar, Mark A. Scheel, Nicholas W. Taylor. *What does a Binary Black Hole Merger Look Like?*. 2015.

[79] learning resource for OpenGL: https://open.gl/introduction

[80] learning resource for OpenGL: https://learnopengl.com/Introduction

[81] Vi Hart, Andrea Hawksley, Henry Segerman. *Hyperenom: Mapping VR Headset Orientation to S^3*. Proceedings of Bridges, 2015.

[82] Vi Hart, Andrea Hawksley, Elisabetta A. Matsumoto, Henry Segerman. *Non-eulcidean virtual reality I: explorations of H^3*. arXiv, 2017.

[83] Vi Hart, Andrea Hawksley, Elisabetta A. Matsumoto, Henry Segerman. *Non-eulcidean virtual reality I: explorations of H^2xE*. arXiv, 2017.

[84] Mark Phillips, Charlie Gunn. *Visualizing Hyperbolic Space: Unusual Uses of 4x4 Matrices*. 1991.

Erklärung:

Hiermit erkläre ich, die vorliegende Arbeit selbstständig verfasst zu haben und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt zu haben.

Ort, Datum

Unterschrift