Vision, Modeling, and Visualization (2022)
J. Bender, M. Botsch, and D. Keim (Eds.)

# An Overview of Techniques for Egocentric Black Hole Visualization and Their Suitability for Planetarium Applications

A. Hissbach[1,2], C. Dick[1], and K. Lawonn[2]

[1]Carl Zeiss Jena GmbH, Germany
[2]Friedrich Schiller University Jena, Germany

**Figure 1:** *A black hole rendered in realtime on a planetarium dome with the ZEISS planetarium software Uniview™.*

**Abstract**

*The visualization of black holes is used in science communication to educate people about our universe and concepts of general relativity. Recent visualizations aim to depict black holes in realtime, overcoming the challenge of efficient general relativistic ray tracing. In this state-of-the-art report, we provide the first overview of existing works about egocentric black hole visualization that generate images targeted at general audiences. We focus on Schwarzschild and Kerr black holes and discuss current methods to depict the distortion of background panoramas, point-shaped stars, nearby objects, and accretion disks. Approaches to realtime visualizations are highlighted. Furthermore, we present the implementation of a state-of-the-art black hole visualization in the planetarium software Uniview.*

**CCS Concepts**

• ***Computing methodologies*** → ***Ray tracing;***

## 1. Introduction

Black holes are a popular item of science communication. In recent years, researchers have collected substantial evidence for their existence: gravitational wave detection [AAA*16], motion of stars in the center of the Milky Way [GET*09], and two images of supermassive black holes in galactic nuclei [EHT19, EHT22]. These discoveries have further increased the public interest in black holes and astronomy.

Educational visualizations are especially relevant for institutions like museums and planetariums. Exocentric visualizations like diagrams and charts (Figure 2) depict spacetime and light rays from a third-person perspective. In contrast, egocentric visualizations (Figure 1) focus on rendering what an observer would see close to a black hole. The first-person perspective can provide an intuitive understanding of special and general relativistic effects and helps to understand unfamiliar phenomena. Using such visualiza-

tions in realtime is particularly beneficial for planetariums because they enable the presenter to interact with their audience.

The main challenge of black hole visualization is the efficient computation of light rays in a curved spacetime. To do so, *general relativistic ray tracing* (GRRT) is employed, which requires the integration of differential equations of varying complexity. Today, modern GPUs and new optimization techniques allow for the visualization of black holes in realtime.

In this state-of-the-art report, we present the first overview of existing works on egocentric black hole visualization targeted at non-scientists. Focusing on Schwarzschild and Kerr black holes, different rendering methods for the following aspects are discussed: distortion of background panoramas, point-shaped stars, nearby objects, and accretion disks. Our goal is to provide an overview of existing approaches and a starting point for future works on black hole visualization for general audiences. To this end, we highlight existing realtime visualizations and provide a case study, showing that realtime black hole visualizations can be brought to planetariums.

## 2. Related Reports on Relativistic and Black Hole Visualizations

Black hole visualizations have been reviewed in reports of science communication and new developments in astronomy. However, none of these reports provide an overview or comparison of the existing methods of egocentric black hole visualization.

Lan et al. [LYA*21] classify state-of-the-art visualizations in astrophysics, including educational visualizations and planetariums in their discussion. Yoon et al. [YPL20] discuss both simulation-based and observation-based black hole visualizations in the context of science education, but do not go into mathematical nor implementation details. Luminet [Lum18] provides an exhaustive historical review of black hole imaging and black hole visualizations from 1979 to 2018, but does not focus on realtime nor egocentric visualizations.

The visualization of black holes is also often addressed in reviews of relativistic visualizations. Müller and Weiskopf [MW11] present and explain various types of general relativistic visualizations, including first-person visualizations of black holes, diagram visualizations, and field-based data visualizations. More visualization techniques for special and general relativity are discussed by Ruder et al. [RWNM08], together with a simulation of a thin accretion disk around a Schwarzschild black hole. Finally, Weiskopf et al. [WBE*06] present methods for explanatory and illustrative visualizations of various relativistic effects. They focus on egocentric visualizations and give an overview of the mathematical concepts of general relativistic ray tracing.

## 3. Physical Background

In this section, we first provide a brief overview of the essential mathematical concepts of black hole visualization. Then, we introduce features of black holes that are part of most visualizations. For a more general introduction into general relativity, black holes, and the mathematics involved, readers may refer to Chapter 17 of Carroll and Ostlie [CO14].

### 3.1. Metrics and the Geodesic Equation

In the theory of *special relativity*, space and time are united into 4-dimensional spacetime. It describes effects that arise at relativistic speed close to the speed of light. The theory of *general relativity* considers the curvature of spacetime, which is caused by mass and energy. This curvature dictates how matter moves within the spacetime.

A *metric tensor* (short *metric*) describes the geometry of a manifold (i. e., a space). For a 4-dimensional spacetime, the metric is a four-by-four matrix. Two metrics commonly used in black hole visualizations are the *Schwarzschild* metric and the *Kerr* metric. The Schwarzschild metric describes spacetime around non-rotating and uncharged black holes, i. e., *Schwarzschild black holes*:

$$g(\mathbf{x}) = \begin{pmatrix} -(1-2M/r) & 0 & 0 & 0 \\ 0 & 1/(1-2M/r) & 0 & 0 \\ 0 & 0 & r^2 & 0 \\ 0 & 0 & 0 & r^2\sin^2\theta \end{pmatrix} \tag{1}$$

where

$M$ : Black Hole Mass

$\mathbf{x} = (t, r, \theta, \phi)$ : Position in Spacetime

The Kerr metric describes spacetime around rotating and uncharged black holes, i. e., *Kerr black holes*. It is a generalization of the Schwarzschild metric and is more complex because of its dependency on the black hole's spin:

$$g(\mathbf{x}) = \begin{pmatrix} -\alpha^2 + \omega^2\varpi^2 & 0 & 0 & -\omega\varpi^2 \\ 0 & \rho^2/\Delta & 0 & 0 \\ 0 & 0 & \rho^2 & 0 \\ -\omega\varpi^2 & 0 & 0 & \varpi^2 \end{pmatrix}$$

where

$$\begin{aligned} \alpha &= \frac{\rho\sqrt{\Delta}}{\Sigma}, \Delta = r^2 - 2Mr + a^2 \\ \varpi &= \frac{\Sigma\sin\theta}{\rho}, \omega = \frac{2aMr}{\Sigma^2} \\ \Sigma &= \sqrt{(r^2+a^2)^2 - a^2\Delta\sin^2\theta} \\ \rho &= \sqrt{r^2 + a^2\cos^2\theta} \end{aligned} \tag{2}$$

$a$ : Black Hole Spin

The Schwarzschild metric is given in *Schwarzschild coordinates*, in which a position is given by three spatial $(r, \theta, \phi)$ and one time coordinate $(t)$. The Kerr metric is given in *Boyer-Lindquist coordinates*, which are a generalization of Schwarzschild coordinates for the Kerr metric. Both metrics are given in *natural units*, which assign unit values to the speed of light ($c = 1$) and the gravitational constant ($G = 1$).

Free-falling objects follow *geodesics*, which are straight lines in a flat spacetime and curves if spacetime is distorted. Because light is mass-less, it follows *null-geodesics*, which represent the shortest possible paths between two points. Geodesics are computed by solving the geodesic equation (using the Einstein summation convention and indices $\mu, \nu, \rho, \sigma$ running from 0 to 1):

$$\frac{d^2x^\mu}{d\lambda^2} + \Gamma^\mu_{\rho\sigma}\frac{dx^\rho}{d\lambda}\frac{dx^\sigma}{d\lambda} = 0, \tag{3}$$

where $x$ is a point on the geodesic, $\lambda$ denotes an affine parameter of the geodesic, and $\Gamma^\mu_{\rho\sigma}$ are the Christoffel symbols, which depend on the metric $g$:

$$\Gamma^\mu_{\rho\sigma} = \frac{1}{2}g^{\mu\nu}\left(\frac{\partial g_{\sigma\nu}}{\partial x^\rho} + \frac{\partial g_{\rho\nu}}{\partial x^\sigma} - \frac{\partial g_{\rho\sigma}}{\partial x^\nu}\right).$$

Requiring the computation of higher derivatives, Equation 3 can not generally be solved explicitly and is often integrated numerically.

### 3.2. Black Holes

Black holes are extremely dense objects which concentrate high mass in a theoretically infinitely small point, a *singularity*. Therefore, spacetime is strongly curved in their vicinity.

Every black hole is characterized by three physical properties: mass, spin, and charge. Since no black hole can exist without mass, there are four types of black holes with respect to these properties: *Schwarzschild* without spin and charge, *Kerr* with spin but no charge, *Reissner-Nordström* with charge but no spin, and *Kerr-Newman* with spin and charge.

Only Kerr black holes are expected to exist in reality and are, therefore, most relevant for realistic black hole visualizations. However, visualizations also often depict Schwarzschild black holes because they are easier to simulate. They are spherically symmetric because they have no spin and, consequently, they have a less complex mathematical description (Equation 1) than Kerr black holes (Equation 2).

### 3.3. Visual Features of Black Holes

*Gravitational lensing* is the most prominent feature of a black hole visualization. The black hole curves spacetime and light rays are bent before reaching an observer. As a result, the light source appears to be at a different position and can be visible infinitely many times. These *higher order images* appear because light rays can wrap around the black hole multiple times before reaching the observer (Figure 2).

We cannot see black holes directly because they create an *event horizon*. The Schwarzschild radius $r_s = 2GM/c^2$ defines the distance to a black hole where the escape velocity equals the speed of light, meaning that not even light can escape at this distance. This radius $r_s$ defines the event horizon, which is not a physical boundary, but marks a point of no return. No light is emitted from the event horizon and it appears as a black region in space, the *black hole shadow*.

Black holes are often depicted being surrounded by an *accretion disk*. Accretion disks form through the process of matter being attracted and collected by a massive object. Because the material in accretion disks moves at relativistic speed, their appearance is subject to special relativistic effects such as the Doppler shift. *General relativistic magnetohydrodynamical* (GRMHD) simulations are used to study the accretion process based on several different accretion models [Las16]. One such model is that of a geometrically thin, but optically thick (i. e., opaque) accretion disk.

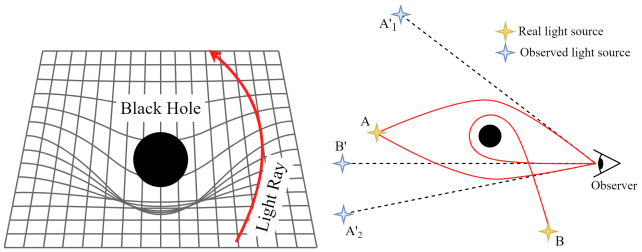


**Figure 2:** *Illustrations of the principal behind gravitational lensing in curved spacetime. Left: The mass of a black hole curves spacetime, leading to the bending of light rays. Right: Curved light rays around a black hole, leading to an apparent change in position $(A'_1, A'_2, B')$ of the light sources $A, B$.*

Furthermore, black holes can cause the formation of *jets*, which consist of ejected matter from the accretion disk. They can extend to thousands of light years and are observed in *active galactic nuclei* (AGN), where super-massive black holes reside.

## 4. Aspects of Black Hole Visualization

In this section, we first discuss challenges of black hole visualization in general and with respect to educational visualization. Then, we present existing approaches of egocentric visualization of Schwarzschild and Kerr black holes. We focus on four aspects: distortion of background panoramas, point-shaped stars, nearby objects, and accretion disks. For each aspect, we describe existing works and group them based on the methods they employ for their visualizations.

### 4.1. Challenges of Black Hole Visualization

Because black holes bend spacetime to an extent such that light cannot escape their gravitational pull, they remain hidden behind their event horizon. Consequently, they cannot be visualized directly, but only through the effects they have on their surroundings.

*General relativistic ray tracing* (GRRT) simulates light in a curved spacetime. This requires integrating the geodesic equation (Equation 3), which is often done numerically. This is a challenge when depicting a black hole in realtime, since curved light rays have to be traced for every frame and every pixel of the viewport at runtime.

Other challenges come with the physically accurate visualization of a black hole's surrounding. To render nearby objects, intersections of light ray segments with the content of the scene (e. g., accretion disks or other objects) have to be determined. But also a physically accurate depiction of accretion disks is difficult because they are not limited to flat disks in reality. An accurate representation requires GRMHD simulations and volume rendering in combination with GRRT.

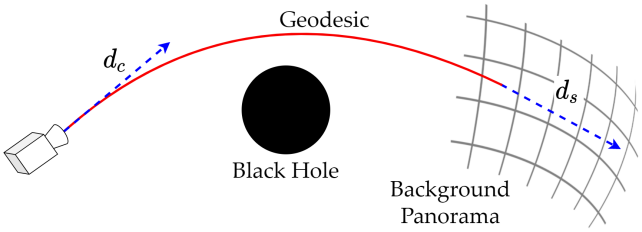


**Figure 3:** *The distortion of a background panorama due to gravitational lensing is computed by determining a light ray's original direction $-d_s$.*

There are also requirements that should be considered when visualizing black holes for general audiences. A detailed discussion of such requirements is provided by Weiskopf et al. [WBE*06]. Firstly, the visualization should be correct, and therefore, based on the appropriate mathematical description of the black hole to be visualized. At the same time, it should be of high visual quality and appealing in order to encourage its audience to interact with it. The visualization should not be cluttered, but focus on a few key points it aims to convey. Finally, an educational black hole visualization benefits from being interactive or animated to aid an intuitive understanding of the effects of general relativity and curved spacetime.

The four visualization aspects we focus on address these requirements in the following way. In an interactive visualization, the distortion of a background panorama, stars, and the accretion disk allow users to get a feeling for the curving of light rays. Visualizing the distortion of everyday objects and familiar star constellations further improves the understanding of this effects and motivates users to play around with the application. Depicting an accretion disk helps to convey the extreme conditions in the vicinity of black holes, and show special and general relativistic effects like the Doppler effect. Finally, stars and an accretion disk rendered in high quality can be used to create stunning scenes and increase the audience's interest in the visualization.

### 4.2. Background Panoramas

To depict the distortion of a background panorama due to gravitational lensing, one has to determine where a light ray came from. A light ray arriving at the camera from the direction $-d_c$ is traced back in time to a sufficiently large distance from the black hole, giving its original direction $-d_s$ (Figure 3).

Schwarzschild black holes are spherically symmetric because they have no spin. As a result, the geodesic equation of the Schwarzschild metric is simple enough to be integrated in a fragment shader in realtime. There are several visualizations of Schwarzschild black holes on GitHub (e.g. [Ose17, Ros20]) that employ the Euler method to integrate geodesics at runtime.

For more accurate results or more complex metrics, Runge-Kutta methods are used to integrate geodesics. The *Black Hole Flight Simulator* by Hamilton [Ham04] creates relativistic visualizations of observers and objects near black holes. It is used by Hamilton and Polhemus [HP10] to study the stereoscopic visualization in curved spacetime. Bohn et al. [BTH*15] present a method to calculate the gravitational lensing of analytic and numerical spacetimes. Using time-dependent, numerical simulations of binary black holes generated for gravitational-wave experiments, they create images of the gravitational lensing of such systems. James et al. [JvTFT15] present a visualization of a Kerr black hole for the movie *Interstellar*. Being created on a render-farm, this visualization focuses on rendering high-quality, physically accurate IMAX images of black holes. Riazuelo [Ria19, Ria20] presents ray tracing code to create egocentric images of views close to and inside the event horizons of black holes. The Schwarzschild and the Kerr metric are investigated, producing fulldome images which are suitable for planetariums.

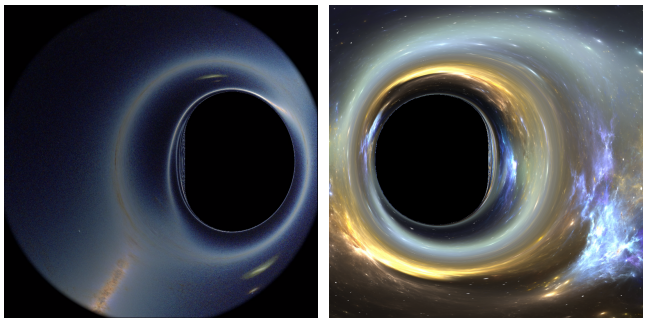

**Figure 4:** *Comparison of gravitational lensing visualizations of Kerr black holes. Left: Kerr black hole rendered using offline GRRT [Ria20]. Right: Kerr black hole visualization rendered in realtime using GRRT and ray interpolation [VE20].*

GRRT can be improved by using lookup tables to avoid the integration of geodesics at runtime. Because of the spherical symmetry of Schwarzschild black holes, lookup tables for the Schwarzschild metric can be kept small and need only two dimensions. Müller and Weiskopf [MW10] have presented a realtime application that visualizes the distortion of a stellar sky and Müller [Mül15] visualizes a distorted background panorama in realtime. For both visualizations, 2-dimensional lookup tables were generated, which are used to map the real position of an object on the celestial sky to its apparent position. Bruneton [Bru20b] also uses precomputed lookup tables for all possible geodesics around a Schwarzschild black hole to compute where on the celestial sphere a light ray originated from in realtime. Verbraeck and Eisemann [VE20] present a realtime visualization of a Kerr black hole by reducing the number of light rays to be integrated. For a predefined camera configuration, they construct a spherical grid centered around the camera. They integrate the geodesic equation numerically in parallel on the CPU for light rays passing through the grid vertices. The origin of a light ray passing through an image pixel is then interpolated from the precomputed rays with CUDA kernels on the GPU. To move the camera in realtime, they precompute a small number of grids lying on a predefined path and generate an image by interpolating between two grids at a time. Figure 4 compares the visual results of the offline Kerr visualization by Riazuelo [Ria20] and the realtime visualization by Verbraeck and Eisemann [VE20].

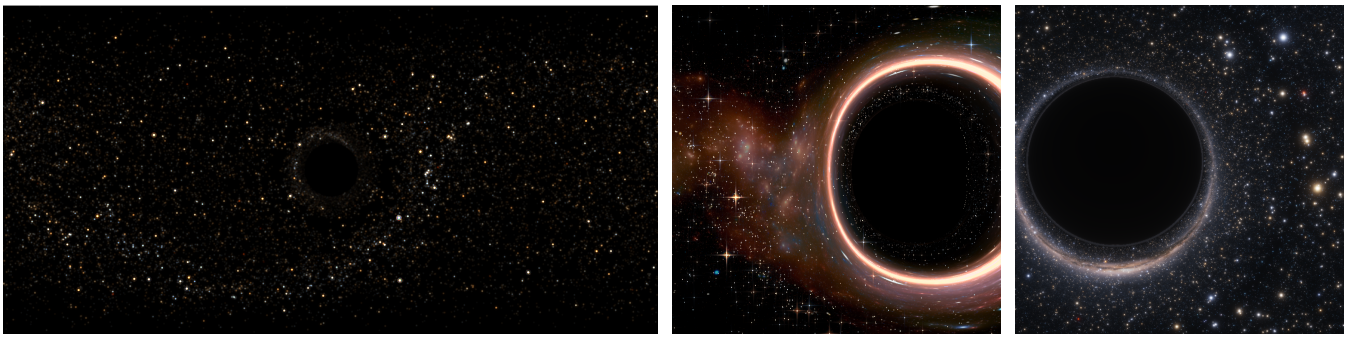

**Figure 5:** *Visualizations of stars as point light sources based on different techniques. Left: Point-based rasterization [MW10]. Middle: k-d tree and diffraction spikes [VE20]. Right: Cubemap and blurring [Bru20b].*

### 4.3. Point-Shaped Stars

Stars have to be handled differently than an overall background texture when the gravitational lensing effect is rendered. This is necessary since stars being baked into the background texture would appear stretched because of texture interpolation. However, since stars are point light sources, gravitational lensing only changes their apparent position, brightness, and color, but not their size. Figure 5 depicts visualizations of point-shaped stars based on three different techniques.

Point-based approaches consider each star separately during rendering. Müller and Weiskopf [MW10] use preprocessed data in lookup tables to compute the apparent position, brightness, and change in color of a star in the vertex stage of the rasterization pipeline. In the fragment stage, the diffraction and final color of a star are computed. Each star is rendered twice in order to depict its first and second order images. Riazuelo [Ria19] computes how gravitational lensing apparently changes the position, brightness, and color of each star separately and renders them accordingly in a separate pass on top of the distorted background panorama.

An alternative approach is to sum up the light of all stars contributing to a pixel's color after the ray tracing step. Verbraeck and Eisemann [VE20] employ a k-d tree that stores position, magnitude, and the color-index of stars to collect all stars within a pixel's footprint. Furthermore, they render diffraction spikes and star trails to simulate visual results of telescopes. Bruneton [Bru20b] renders stars using a cubemap whose texels store color and position of the background stars. A mip-map of this cubemap is generated by summing the colors of texels instead of averaging them. A custom filtering method is then used to select mip-map levels of the star map and sum the colors of texels within the pixel's footprint. In a final step, a bloom effect increases the impact of bright stars.

### 4.4. Nearby Objects

Visualizing the gravitational lensing effect on objects close to the black hole further increases the complexity of GRRT. In contrast to classical ray tracing, intersection tests with scene objects have to be performed with ray segments during the numerical integration, instead of a single intersection with a straight ray.

Zahn [Zah91] describes GRRT and how objects are rendered in curved spacetime. Furthermore, a sequence of images is rendered that show a ring around Schwarzschild black holes of varying mass. Müller and Boblest [MB11] visualize the circular motion around a Schwarzschild black hole and encapsulate the observer in a torus surrounding the black hole. Using GRRT, they precompute two lookup tables that store the first and second intersection of light rays with the torus, respectively. These lookup tables are used in the fragment shader to determine whether a light ray has passed through the torus.

*Relativistic ray tracers* implement GRRT to perform ray tracing and ray-object intersections in curved spacetimes. They allow users to define scenes and objects to be rendered, together with the properties of the spacetime the scene resides in. Weiskopf et al. [WSE04] implement a nonlinear ray tracer running on GPUs using pixel shaders. Their implementation can compute ray-object intersections for spheres and triangles and is able to render scenes in a Schwarzschild spacetime. Kuchelmeister et al. [KMA*12] present an interactive relativistic ray tracer that is capable of rendering spheres around Schwarzschild and Kerr black holes. Its ray tracing code runs on GPUs and is implemented in CUDA. Müller [Mül14] implements *Geovis*, a relativistic ray tracer for special and general relativistic visualization that is parallelized on a computer cluster using MPI. This ray tracer employs GRRT using the *Motion4D* library by Müller [MG09] and tests each light ray for intersection with the scene objects. Müller et al. [MBW15] show visualizations of objects nearby single and binary black holes created with Geovis in a visualization showcase. Mandal et al. [MAC21] implement a Monte-Carlo approach for GRRT and depict what everyday scenes would look like if they contained a black hole or wormhole.

An alternative approach to GRRT is to use rasterization-based rendering [Yam16, MSW21a]. First, lookup tables are generated in a preprocessing step, storing a mapping from 3D positions in the region around the black hole to where an observer at a given position would really see them. The mesh vertices are translated in the vertex stage based on the precomputed data, mapping positions in space to an apparent position according to gravitational lensing and light travel time. Figure 7 demonstrates this method, depicting a distorted mesh of a ring around a Schwarzschild black hole. This approach makes fast rendering of meshes possible, but requires

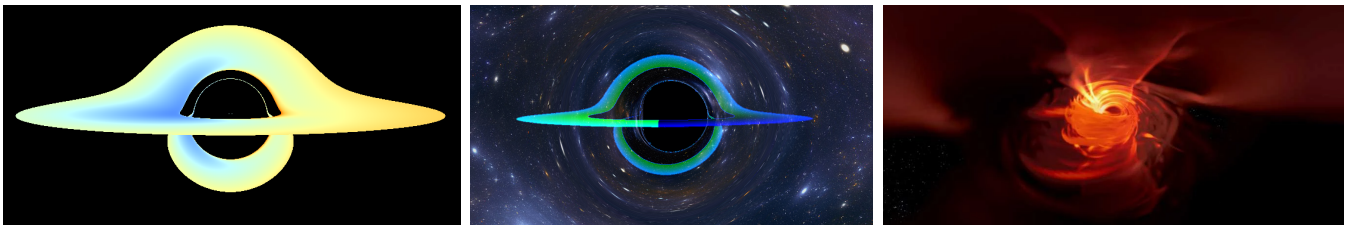

**Figure 6:** *Accretion visualizations based on three different techniques. Left: Analytic intersection test with a thin disk [MF12]. Middle: Thin disk around a Kerr black hole based on ray-disk intersection during ray tracing [VE20]. Right: Volumetric accretion flow based on GRMHD data [DBK*18].*

large lookup tables for free camera movement. Also, despite tessellation, a mesh is heavily distorted when close to the event horizon and strongly degenerated triangles must be removed.

### 4.5. Accretion Disks

Many visualizations depict an optically thick, flat disk with zero thickness, based on a thin disk model. Figure 6 shows three different visualizations of accretion around black holes.

Müller and Frauendiener [MF12] present a realtime approach to render a thin accretion disk around a Schwarzschild black hole. They evaluate an analytic expression in constant time, giving the intersection point of a geodesic with a plane. Verbraeck and Eisemann [VE20] and James et al. [JvTFT15] determine the intersection of geodesics and the disk during ray tracing. Bruneton [Bru20b], in contrast, determines disk intersections in realtime using a lookup table that stores intersection points for any given geodesic and plane around a Schwarzschild black hole.

After determining whether a geodesic intersects the accretion disk, the light emitted from the intersection point has to be determined. Müller and Frauendiener [MF12] provide visualizations of the Doppler shift and of the temperature color of the accretion disk. Bruneton [Bru20b] renders the color of the accretion disk considering the Doppler shift based on a thin disk model and a user-defined peak temperature. Furthermore, the structure of the disk is modeled with regions of higher and lower density, moving along orbits around the black hole.

Visualizing volumetric accretion disks is more challenging since GRRT has to be combined with volume rendering. Also, realistic volume data have to be created with GRMHD simulations. Hamilton [Ham04] makes use of such data in the Black Hole Flight Simulator and Davelaar et al. [DBK*18] use GRMHD data to render a 360° film for virtual reality of a black hole and its accretion flow.

## 5. Using Black Hole Visualizations in Planetariums—A Case Study

In this section, we investigate the realtime visualization of black holes for planetariums. We discuss performance requirements of planetariums and which approaches fulfill them. Finally, we describe a case study in which we implement a black hole visualization in a state-of-the-art planetarium software.

Visualizations for planetariums have to meet high requirements

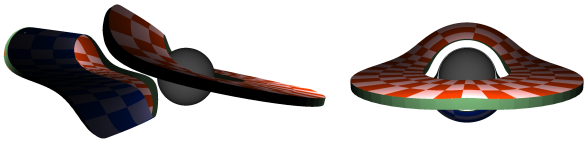

**Figure 7:** *Rasterization-based rendering of a ring around a Schwarzschild black hole [MSW21a]. Left: Seen from the side to demonstrate the mesh deformation. Right: Seen from the simulated observer's point of view.*

on quality and performance. Planetarium domes cover a 180° field-of-view angle and require resolutions of at least 4k pixels along the dome meridian. Frame rates of 60Hz are the industry standard and must be continuously delivered to provide a smooth and stutter free visual experience.

The planetarium software renders the digital content that is projected onto the dome surface. It is used to generate animated 3D renderings in realtime, allowing for interactive control and live presentations as well as playback of prerecorded fulldome videos for non-interactive planetarium shows.

In our case study, we investigated how a state-of-the-art black hole visualization can be integrated into a planetarium application. To this end, we used the ZEISS planetarium software *Uniview*™ [Car22,KHE*10], which allows for the realtime visualization of astronomical objects, datasets, and simulations on both desktop PCs and planetarium domes. The software offers free navigation within a model of the observable universe and it is possible to add custom content through an OpenGL interface.

For our visualization, we considered the works of Bruneton [Bru20b], Verbraeck and Eisemann [VE20], and Müller et al. [MSW21a] because they are performant enough to be rendered at realtime frame rates, in a sufficient quality for planetariums. However, we found that integrating the visualization of Bruneton would be most viable because it is mostly based on shaders. This way, the visualization could be implemented using the OpenGL interface of Uniview. Integrating the other two visualizations into the software would be more involved because they depend on realtime ray tracing or would require changes within the rendering pipeline.

Our visualization consists of the distortion of a background panorama, an accretion disk, a jet, and point-shaped stars. It is pos-

**Table 1:** *Frame rates in Hz of our Schwarzschild black hole visualization in Uniview.*

| | Base | AD | J | S | AD & J & S |
|---|---|---|---|---|---|
| PC | 630 | 429 | 558 | 351 | 275 |
| Cluster | 137 | 98 | 127 | 65 | 55 |

*AD* = accretion disk, *J* = jet, *S* = stars

sible to rotate the camera freely around the black hole and operators have interactive control over the size and temperature of the accretion disk as well as the size of the jet cones.

The performance of the black hole visualization was inspected on a PC and on a rendering cluster driving a fulldome projection system, which consists of six digital projectors. The PC was equipped with an Intel Xeon W-2275 @ 3.30 GHz CPU, 64 GB of RAM, and an NVIDIA RTX A4000 graphics card with 16 GB of VRAM. The rendering cluster consists of six render nodes, of which each renders and projects a part of the fulldome image onto the dome. Each node was equipped with an Intel Xeon E5-2643v3 @ 3.40 GHz CPU, 16 GB of RAM, and an NVIDIA Quadro K5200 graphics card with 8 GB of VRAM. The cluster nodes render at a viewport resolution of 3072x3072, but use different view frustums to accommodate for the distinct parts of the dome they cover. To compare the PC and the cluster nodes, we tested the performance on the PC for every view configuration of the six nodes.

In our experiment, a base setup of the visualization was rendered, consisting of the black hole shadow and a distorted panorama background. Additionally, the performance impact of three visualization features was tested: the accretion disk (*AD*), the jet (*J*), and point-shaped stars (*S*). Table 1 summarizes the performance tests on the PC and on the fulldome rendering cluster. Each column represents a feature that was added to the base setup and we always report the minimum frame rate over all six cluster node view configurations.

Uniview rendered the black hole with a frame rate of over 60Hz on the PC with all features enabled (last column in Table 1). Point-shaped stars (second to last column in Table 1) had the largest impact on performance, because of the sampling and custom filtering of the star cubemap. On the cluster, which was equipped with older hardware, the frame rate behaved similarly, but was always lower than on the PC and below 60Hz when rendering all features of the visualization.

With this case study, we showed that black hole visualizations can be used in planetariums in realtime. Our visualization created high quality results on both test systems at realtime frame rates. The reduced performance on the rendering cluster was expected, since it is equipped with older GPUs. Planetariums with more recent hardware will be able to use all of the visualization's features in realtime, while others can omit the rendering of point-shaped stars or render them at a reduced quality.

Rendering rotating black holes and objects close to a black hole remains future work. For more complex metrics, like the Kerr metric, we expect that performance will still be a bottleneck for high quality visualizations. Rendering objects close to the black hole is currently not possible with our implementation and would reduce its performance because of additional intersection tests. However, combining the ray tracing-based approaches with the rendering of nearby objects using rasterization would be a promising direction and a valuable addition to planetarium applications.

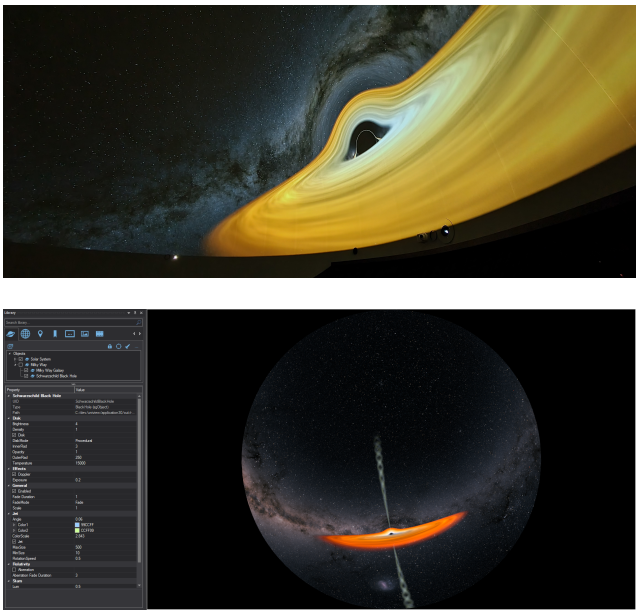

**Figure 8:** *A Schwarzschild black hole visualization based on the work of Bruneton [Bru20b] rendered in Uniview. Top: Projection onto a planetarium dome. Bottom: Rendered in the desktop mode of Uniview.*

A standalone prototype of our visualization is available in a GitHub repository [His22]. Some of the works discussed in this report provide source code or applications of their visualizations as well. Many visualizations of Müller et al. are available on the website *VisAstro* [Mü22]. Their black hole visualization based on rasterization [MSW21a] can be downloaded from a GitHub repository [MSW21b], and the implementation of Bruneton [Bru20b] is available as an online demo and source code [Bru20a].

## 6. Conclusions and Future Work

In this work, we have provided an overview of the state-of-the-art in egocentric visualization of Schwarzschild and Kerr black holes. We focused on four aspects of black hole visualization: distortion of background panoramas, point-shaped stars, nearby objects, and accretion disks. Promising candidates for realtime visualizations in planetariums were highlighted and an implementation of a state-of-the-art Schwarzschild black hole visualization in the planetarium software Uniview was shown.

The key result of our case study is that state-of-the-art Schwarzschild black hole visualizations can be used in planetariums in realtime. Considering, however, that all black holes have a non-zero spin in reality, even more efficient visualizations of Kerr black holes than available today are desirable, along with accretion disks and jets. This would allow for even more realistic realtime visualizations of black holes for planetarium audiences.

## References


[AAA*16] ABBOTT B. P., ABBOTT R., ABBOTT T., ET AL.: Observation of gravitational waves from a binary black hole merger. *Physical review letters 116*, 6 (2016), 061102. 1

[Bru20a] BRUNETON E.: A real-time high-quality black hole shader, 2020. URL: https://github.com/ebruneton/black_hole_shader. 7

[Bru20b] BRUNETON E.: Real-time high-quality rendering of non-rotating black holes. *arXiv preprint arXiv:2010.08735* (2020). 4, 5, 6, 7

[BTH*15] BOHN A., THROWE W., HÉBERT F., ET AL.: What does a binary black hole merger look like? *Classical and Quantum Gravity 32*, 6 (2015), 065002. 4

[Car22] CARL ZEISS JENA GMBH: Uniview, 2022. URL: https://www.zeiss.com/planetariums/int/products/software-and-applications/uniview.html. 6

[CO14] CARROLL B. W., OSTLIE D. A.: General relativity and black holes. In *An Introduction to Modern Astrophysics*, 2 ed. Pearson, 2014, ch. 17, pp. 671–716. 2

[DBK*18] DAVELAAR J., BRONZWAER T., KOK D., ET AL.: Observing supermassive black holes in virtual reality. *Computational Astrophysics and Cosmology 5*, 1 (2018), 1–17. 6

[EHT19] EHT COLLABORATION: First M87 event horizon telescope results. I. The shadow of the supermassive black hole. *The Astrophysical Journal Letters 875*, 1 (2019), L1. 1

[EHT22] EHT COLLABORATION: First sagittarius A* event horizon telescope results. I. The shadow of the supermassive black hole in the center of the Milky Way. *The Astrophysical Journal Letters 930*, L12 (2022). 1

[GET*09] GILLESSEN S., EISENHAUER F., TRIPPE S., ET AL.: Monitoring stellar orbits around the massive black hole in the galactic center. *The Astrophysical Journal 692*, 2 (2009), 1075. 1

[Ham04] HAMILTON A. J.: Black hole flight simulator. In *American Astronomical Society Meeting Abstracts # 204* (2004), vol. 204, pp. 78–06. URL: https://jila.colorado.edu/~ajsh/insidebh/bhfs.html. 4, 6

[His22] HISSBACH A.: BlackHoleVis, 2022. URL: https://github.com/brosefine/BlackHoleVis. 7

[HP10] HAMILTON A. J., POLHEMUS G.: Stereoscopic visualization in curved spacetime: seeing deep inside a black hole. *New Journal of Physics 12*, 12 (2010), 123027. 4

[JvTFT15] JAMES O., VON TUNZELMANN E., FRANKLIN P., THORNE K. S.: Gravitational lensing by spinning black holes in astrophysics, and in the movie Interstellar. *Classical and Quantum Gravity 32*, 6 (2015), 065001. 4, 6

[KHE*10] KLASHED S., HEMINGSSON P., EMMART C., ET AL.: Uniview—visualizing the Universe. In *Eurographics (Areas Papers)* (2010), pp. 37–43. 6

[KMA*12] KUCHELMEISTER D., MÜLLER T., AMENT M., ET AL.: GPU-based four-dimensional general-relativistic ray tracing. *Computer Physics Communications 183*, 10 (2012), 2282–2290. 5

[Las16] LASOTA J.-P.: Black hole accretion discs. In *Astrophysics of Black Holes*. Springer, 2016, pp. 1–60. 3

[Lum18] LUMINET J.-P.: Seeing black holes: From the computer to the telescope. *Universe 4*, 8 (2018), 86. 2

[LYA*21] LAN F., YOUNG M., ANDERSON L., ET AL.: Visualization in astrophysics: developing new methods, discovering our universe, and educating the Earth. In *Computer Graphics Forum* (2021), vol. 40, Wiley Online Library, pp. 635–663. 2

[MAC21] MANDAL A., AYUSH K., CHAUDHURI P.: Non-linear Monte Carlo ray tracing for visualizing warped spacetime. In *VISIGRAPP (3: IVAPP)* (2021), pp. 76–87. 5

[MB11] MÜLLER T., BOBLEST S.: Visualizing circular motion around a Schwarzschild black hole. *American Journal of Physics 79*, 1 (2011), 63–73. 5

[MBW15] MÜLLER T., BOBLEST S., WEISKOPF D.: Visualization showcase: General-relativistic black hole visualization. In *EGPGV@ EuroVis* (2015), pp. 29–32. 5

[MF12] MÜLLER T., FRAUENDIENER J.: Interactive visualization of a thin disc around a Schwarzschild black hole. *European journal of physics 33*, 4 (2012), 955. 6

[MG09] MÜLLER T., GRAVE F.: Motion4D—a library for lightrays and timelike worldlines in the theory of relativity. *Computer Physics Communications 180*, 11 (2009), 2355–2360. 5

[MSW21a] MÜLLER T., SCHULZ C., WEISKOPF D.: Adaptive polygon rendering for interactive visualization in the Schwarzschild spacetime. *European Journal of Physics 43*, 1 (2021), 015601. 5, 6, 7

[MSW21b] MÜLLER T., SCHULZ C., WEISKOPF D.: GRPolyRen, 2021. URL: https://github.com/tauzero7/GRPolyRen. 7

[Mül14] MÜLLER T.: GeoViS—relativistic ray tracing in four-dimensional spacetimes. *Computer Physics Communications 185*, 8 (2014), 2301–2308. 5

[Mül15] MÜLLER T.: Image-based general-relativistic visualization. *European Journal of Physics 36*, 6 (2015), 065019. 4

[MW10] MÜLLER T., WEISKOPF D.: Distortion of the stellar sky by a Schwarzschild black hole. *American Journal of Physics 78*, 2 (2010), 204–214. 4, 5

[MW11] MÜLLER T., WEISKOPF D.: General-relativistic visualization. *Computing in Science & Engineering 13*, 06 (2011), 64–71. 2

[Mü22] MÜLLER T.: VisAstro, 2022. URL: https://www2.mpia-hd.mpg.de/homes/tmueller/index.php. 7

[Ose17] OSEISKAR: Ray-traced simulation of a black hole, 2017. URL: https://github.com/oseiskar/black-hole. 4

[Ria19] RIAZUELO A.: 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. *International Journal of Modern Physics D 28*, 02 (2019), 1950042. 4, 5

[Ria20] RIAZUELO A.: Seeing relativity—III. Journeying within the Kerr metric toward the negative gravity region. *International Journal of Modern Physics D 29*, 16 (2020), 2050109. 4

[Ros20] ROSSNING92: Real-time black hole rendering in OpenGL, 2020. URL: https://github.com/rossning92/Blackhole. 4

[RWNM08] RUDER H., WEISKOPF D., NOLLERT H.-P., MÜLLER T.: How computers can help us in creating an intuitive access to relativity. *New Journal of Physics 10*, 12 (2008), 125014. 2

[VE20] VERBRAECK A., EISEMANN E.: Interactive black-hole visualization. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2020), 796–805. 4, 5, 6

[WBE*06] WEISKOPF D., BORCHERS M., ERTL T., ET AL.: Explanatory and illustrative visualization of special and general relativity. *IEEE Transactions on Visualization and Computer Graphics 12*, 4 (2006), 522–534. 2, 4

[WSE04] WEISKOPF D., SCHAFHITZEL T., ERTL T.: GPU-based nonlinear ray tracing. In *Computer Graphics Forum* (2004), vol. 23, Wiley Online Library, pp. 625–633. 5

[Yam16] YAMASHITA Y.: Implementing a rasterization framework for a black hole spacetime. *Journal of Information Processing 24*, 4 (2016), 690–699. 5

[YPL20] YOON H.-G., PARK J., LEE I.: Significance of black hole visualization and its implication for science education focusing on the event horizon telescope project. *Universe 6*, 5 (2020), 70. 2

[Zah91] ZAHN C.: *Vierdimensionales Ray-Tracing in einer gekrümmten Raumzeit*. Dissertation, Universität Stuttgart, 1991. 5