Eurographics Symposium on Parallel Graphics and Visualization (2015)
C. Dachsbacher, P. Navrátil (Editors)

Visualization Showcase: General-Relativistic Black Hole Visualization

T. Müller, S. Boblest, and D. Weiskopf

Visualization Research Center, University of Stuttgart, Germany

Abstract

Black holes are among the most fascinating and weird objects in the universe. They distort space and time in their close neighborhood in a way that is far beyond our every day experience. We demonstrate the visual effects of this curved spacetime by means of four-dimensional nonlinear ray tracing applied to an accretion disk around a spinning black hole and a sphere oscillating between two static, charged black holes. We discuss how visualization helps predict and communicate the interesting effects of general relativity, in particular, its geometric effects on light propagation. The nonlinear behavior of light propagation leads to a compute-intensive rendering process; we report on our experiences with highly parallel rendering in this context.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

In 1905, Albert Einstein showed that space and time are not two distinct qualities but one single entity called *spacetime*. Special relativity, based on this fact, describes relative motion near the speed of light and the resulting effects. Ten years later, Einstein generalized the idea of spacetime to incorporate gravitation. From that time on, gravitation is no longer considered as a force like in Newton's theory, but as a consequence of curved spacetime mathematically subsumed in the general theory of relativity.

The most extreme object described by general relativity is a black hole. Even though the existence of black holes lacks a direct proof, astrophysicists are convinced that a black hole exists at the center of each galaxy. Probably, in the next ten years it will become feasible to directly observe the close neighborhood of a black hole. Hence, it is important to understand what will probably be seen in such an observation.

We follow the basic approach of *egocentric* visualization [We06]: such first-person visualization aims to depict the image a virtual camera or telescope would actually produce in a general relativistic setting. This approach is conceptually simple and intuitive because it resembles a visual experiment in which we—as viewers of the visualization—are virtually put into a scene governed by general relativity.

Technically, the rendering process for such visualization is based on *four-dimensional nonlinear ray tracing*, where the propagation of light is reversed by starting light rays from the observer and tracing them back in time [Wei00]. Compared to standard three-dimensional ray tracing, the finite speed of light, the motion of objects, and the influence of the curved spacetime on light ray trajectories resulting in gravitational lensing and frequency shifts have to be taken into consideration. These lead to a rendering process with very high compute costs. Therefore, we have to employ parallelization strategies on cluster computers to make the rendering problem tractable.

Not only since the motion picture "Interstellar" by Christopher Nolan and Kip Thorne [JvTFT15], general relativistic visualization has become attractive for astrophysical modeling and, in particular, for educational and communication purposes. For example, the Einstein year 2005—the 100^{th} anniversary of Special Relativity and Einstein's annus mirabilis—led to numerous visualization activities centered around Einstein's theory of relativity and corresponding museum exhibitions for the general public [We06].

Over the following ten years, we have been making much progress in dealing with more complex scenarios and types of spacetimes. In this showcase paper, we demonstrate that such advanced visualization can help better understand such relativistic scenes and the curious bending of light due to curved spacetime. In particular, we show the influence of

DOI: 10.2312/pgv.20151152

the angular momentum and the mass of a black hole on the visual appearance of an accretion disk that might also be interesting for real observations. We also present a scenario of a dihole metric that serves as a first step to study multi black hole systems—a scenario not previously covered in motion pictures or advanced animation productions.

2. Four-Dimensional Ray Tracing

The transition from 3D to 4D ray tracing bears several challenges. First of all, light rays no longer follow straight lines but lightlike geodesics bent due to curved spacetime. The underlying geodesic equation reads

$$\frac{\mathrm{d}^2 x^\mu}{\mathrm{d}\lambda^2} + \Gamma^\mu_{\rho\sigma} \frac{\mathrm{d}x^\rho}{\mathrm{d}\lambda} \frac{\mathrm{d}x^\sigma}{\mathrm{d}\lambda} = 0, \tag{1}$$

where the Christoffel symbols $\Gamma^\mu_{\rho\sigma}$ depend on the metric (i.e., a mathematical description of the geometry of the spacetime) and its partial derivatives. The 4D positions (space and time) along the light rays are described by x^μ, and λ serves to parametrize the 4D curve of the light ray. The Greek indices run from 0 to 3, indicating the 4D spacetime coordinates. The geodesic equation is a second-order nonlinear ordinary differential equation that has to be integrated from the observer into the scene in order to determine the primary rays for ray tracing. Implicitly, the finite speed of light becomes crucial while solving the geodesic equation. A detailed discussion of relativistic visualization can be found elsewhere [Wei01].

With the above approach, we can compute the propagation of light. However, we also need a scene description to produce contents for the image. Unfortunately, modeling a relativistic scene is much more demanding than a 3D nonrelativistic scene because the geometry of the underlying space is curved already, i.e., the deformation of the spacetime affects the physical model of the scene object.

As one solution to this problem, we adopt the approach of local linearization: a small (static or moving) object can be defined with respect to its local reference frame, i.e., a local coordinate system centered around the object. Here, the term *small* has to be related to the curvature of spacetime. Within the reference frame, the object can be described as in the 3D linear and static case. The object's motion is relegated to the *local tetrad* that represents the object's reference frame. If there is no propulsion, the tetrad follows the path of a timelike geodesic. Therefore, we have to integrate the geodesic equation (1) again, but now for the motion of the object. We also have to take into account that the local tetrad changes its orientation while it is moving along its timelike geodesic: we additionally solve the equation for the parallel-transport of the tetrad vectors to determine the geodesic precession of the local tetrad [MG09]. For example, an object on the last stable circular orbit around a Schwarzschild black hole (i.e., a black hole that is not spinning and has no charge) undergoes a 105° rotation due to precession.

Another solution to the issue of scene modeling uses *coordinate objects* to represent scene objects. Even extended objects can be used, as long as there is a mapping from some appropriate internal coordinate system of the object to the overall coordinate system in which ray tracing is computed. The internal coordinates should resemble Cartesian coordinates so that the object description can be understood intuitively. Since the internal coordinates are often not perpendicular in the Euclidean sense, they are called *pseudo-Cartesian* coordinates. For example, the special, yet relevant cases of a background image (at infinity) or a simple accretion disk model can be handled as coordinate objects.

In our visualization showcase, we use and combine both types of scene modeling; the rendering process is based on 4D nonlinear ray tracing. Furthermore, we deliberately "switch off" relativistic effects on illumination like modifications of color (e.g., gravitational red shift) or luminance (e.g., from gravitational lensing). In this way, we let the viewer focus on the geometric effects of general relativity, which are the main purpose of our video showcase.

3. Parallel Rendering and Implementation

From the previous section, it is clear that rendering has to deal with high computational costs due to 1) the integration of curved light rays and 2) the computation of the intersection between curved rays and the complex scene description. As a result, straightforward and nonparallel rendering would lead to extremely long production timelines that would make long or complex videos infeasible.

Therefore, parallel computation is indispensable. Since rays are traced from the camera independently from each other, we use image decomposition for parallelization. This leads to a simple variant of sort-first rendering [MCEF94]: The scene descriptions tend to be quite compact and, thus, can be replicated on each of the parallel compute nodes. The only purpose of parallelization is to scale rendering speed, but not memory. The parallelization works by splitting the image space into stripes (for all images of an animated film), leading to a queue of *working sets* to be processed by the nodes. Once finished with one working set, a node fetches the next working set from the queue. We organize the working sets at an appropriate granularity in order to achieve excellent load balancing and scalability. The intermediate results are collected and combined on the master node.

Parallelization by image-space decomposition addresses the problem of high computational costs for solving the geodesic equation and light tracing. The other compute-intensive part is the calculation of intersections between light rays and objects. In a naive implementation, each ray segment (of the polygonal approximation of the ray) has to be tested against each motion segment of an object and all its components (e.g., triangles). In other words, the single intersection computation that would be needed for a single

ray–object pair (i.e., $O(1)$) now leads to an $O(n^2)$ computation, where n represents the number of discretization steps along geodesics. One acceleration strategy could use space partitioning such as octrees. However, such partitioning techniques are based on Euclidean distances and partially on Cartesian coordinates, but general relativity is based on semi-Riemannian manifolds and arbitrary coordinate systems. Instead, we adopt another acceleration strategy: the use of bounding volumes. We employ a spacetime tube that acts as a bounding box around the scene object but has an additional time extent, thus largely reducing the number of intersection computations between a ray segment and the object.

For the implementation, parallel computation on GPUs would be one option [KMA*12, WSE04]. Unfortunately, GPU architectures are less suited for high-quality general relativistic visualization than standard 3D rendering (but rather for interactive rendering). One issue is that floating-point precision is not sufficient because of the high accuracy needed for solving the geodesic equations. Another issue is the stepsize control for the geodesic integration which is inevitable because of the possibly large differences in curvature along a light ray. Furthermore, we need an implementation of the spacetime tube as an acceleration strategy.

Instead, we use parallelization on a CPU cluster computer. We use the 4D ray tracing code *GeoViS* [Mül14], which is based on the *Motion4D-library* [MG09] and has an educational focus. The source code is freely available from `http://go.visus.uni-stuttgart.de/geovis`. Alternative ray tracing codes—albeit with different focal points—include GYOTO [VPGP11] or GRay [CPÖ13].

Our showcase video was rendered on a Linux cluster with 64 Intel Xeon E5620 (2.4 GHz) CPUs (with a total of 512 cores), 1.5 TB of combined main memory, and Infiniband interconnect. *GeoViS* is parallelized using OpenMPI and scenes are implemented using a description language based on the Scheme programming language. The Runge-Kutta Fehlberg integrator from the GNU Scientific Library (GSL) with stepsize control was used integrate the lightlike geodesics. The images were rendered at 1280×720 resolution with 2×2 supersampling. To obtain nearly optimal load balancing and for quick previews, we split each image into 512 stripes. For the dihole sequence consisting of 2100 images, this yields a total of about 1 075 200 working sets. Table 1 documents the overall rendering times and the number of cores used. The scene names refer to the scenarios described in the following section.

Table 1: *Calculation times and number of compute cores.*

Scene	# Images	Compute time	# Cores
kerr_accrDisk	900	4 h 15 min	288
kerr_passing	1700	7 h 20 min	288
dihole	2100	48 h 11 min	232

4. Video

We now provide a brief description of the scenes shown in the video, along with the physical story behind them.

4.1. Accretion Disk around Kerr Black Hole

The Kerr spacetime can describe a rotating black hole—a typical scenario of a massive object in the center of a galaxy or a massive remnant of a star. Matter cannot directly collapse into a black hole because it first has to get rid of angular momentum. Hence, a star that approaches a black hole will first be torn apart and its remnants will build an accretion disk, see Fig. 1. The inner rim is determined by the last stable orbit where the velocity is just high enough to compensate for the gravitational attraction. For the sake of convenience, we use the last stable orbit $r = 6\,M$ for a black hole with zero angular momentum, $a = 0$, and we neglect the actual velocity of the disk particles by using a fixed checkered texture. We use geometric units, where distances are given in multiples of the black hole mass M.

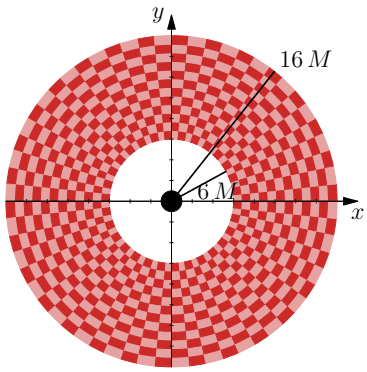

Figure 1: *The infinitely thin accretion disk around the Kerr black hole is modeled using pseudo-Cartesian coordinates.*

The video starts with the view above the accretion disk ("kerr_accrDisk" in Table 1). While increasing the inclination angle from zero to 80°, the back side of the disk appears to bend like an arch above and below the black hole. The occurrence of these apparent structures can be understood by tracing some responsible light rays, see Fig. 2. The asymmetry of the shadow of the black hole (black region) decreases with decreasing angular momentum a of the Kerr black hole. For $a = 0$, the spacetime simplifies to a static Schwarzschild black hole. Reducing the mass of the black hole finally yields flat spacetime and an undistorted ring.

The second part of the video ("kerr_passing") shows the free-fall of an observer along a timelike geodesic that penetrates the plane of the accretion disk between the black hole horizon and the inner rim of the disk. The observer's reference frame undergoes a geodesic precession with a rotation along the trajectory. Hence, the special-relativistic aberration effect due to the high velocity at the point of closest approach to the black hole is nearly unnoticeable.

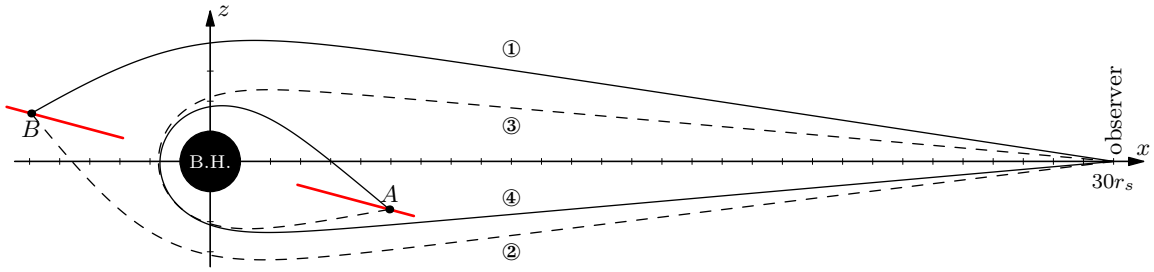

Figure 2: *Geodesics responsible for the apparent distortion of the accretion disk for $M = 1$ and $a = 0$. Dashed lines represent geodesics coming from the bottom side of the disk while solid lines start from the top side.*

4.2. Multi Black Hole System

A simple model of a multiple black hole system is described by the extreme Reissner-Nordstrøm dihole metric, where two maximally charged black holes with identical masses are fixed on the z-axis [WMWW13]. Compared to the disk scene, we use a moving, checkered sphere that starts from rest at $(x = 5, y = z = 0)$, in the symmetry plane between the two black holes. The curved spacetime then lets the sphere oscillate along the x-axis with a period of $T \approx 110.59$.

The third part of the video ("dihole") begins with the sphere at its initial position in flat space. While increasing the masses of the black holes from zero to their maximum value, multiple shadows of the black holes appear, and also the sphere appears multiple times, see Fig. 3. The observer is located at $(x = 0, y = 80, z = 0)$ when the sphere is released and begins to oscillate along the x-axis. After one period, the observer pans along a circular line to $(x = 80, y = 0, z = 0)$ while the sphere is still moving along its trajectory. After another full period, the observer pans to $(x = 0, y = 0, z = 80)$ and then waits until the sphere has finished its fifth oscillation. In the end, the sphere is fixed again at its initial position while the masses of the black holes are decreased to zero.

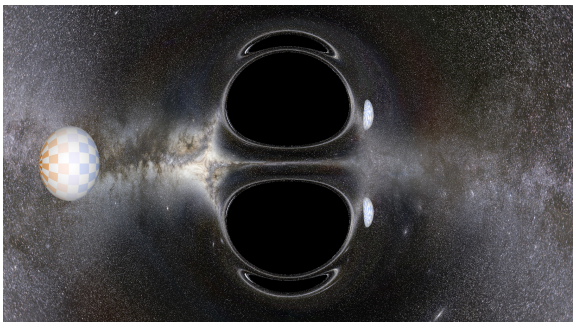

Figure 3: *Sphere at rest in the dihole metric. The four dark regions represent the shadows of the two black holes. To the right, the sphere has two apparent images.*

Acknowledgments

This work was partially funded by Deutsche Forschungsgemeinschaft (DFG) within project WE 2836/2-2.

References

[CPÖ13] CHAN C.-K., PSALTIS D., ÖZEL F.: GRay: a massively parallel GPU-based code for ray tracing in relativistic spacetimes. *Astrophys. J. 777*, 1 (2013), 13. 3

[JvTFT15] JAMES O., VON TUNZELMANN E., FRANKLIN P., THORNE K. S.: Gravitational lensing by spinning black holes in astrophysics, and in the movie Interstellar. *Class. Quantum Grav. 32*, 6 (2015), 065001. 1

[KMA*12] KUCHELMEISTER D., MÜLLER T., AMENT M., WUNNER G., WEISKOPF D.: GPU-based four-dimensional general-relativistic ray tracing. *Comput. Phys. Commun. 183*, 10 (2012), 2282–2290. 3

[MCEF94] MOLNAR S., COX M., ELLSWORTH D., FUCHS H.: A sorting classification of parallel rendering. *IEEE Comput. Graph. Appl. 14*, 4 (1994), 23–32. 2

[MG09] MÜLLER T., GRAVE F.: Motion4D – a library for lightrays and timelike worldlines in the theory of relativity. *Comput. Phys. Commun. 180* (2009), 2355–2360. 2, 3

[Mül14] MÜLLER T.: GeoViS – relativistic ray tracing in four-dimensional spacetimes. *Comput. Phys. Commun. 185*, 8 (2014), 2301–2308. 3

[VPGP11] VINCENT F. H., PAUMARD T., GOURGOULHON E., PERRIN G.: GYOTO: a new general relativistic ray-tracing code. *Class. Quantum Grav. 28*, 22 (2011), 225011. 3

[We06] WEISKOPF D., ET AL.: Explanatory and illustrative visualization of special and general relativity. *IEEE Trans. Vis. Comput. Graphics 12*, 4 (2006), 522–534. 1

[Wei00] WEISKOPF D.: Four-dimensional non-linear ray tracing as a visualization tool for gravitational physics. In *Proc. IEEE Visualization Conf.* (2000), pp. 445–448. 1

[Wei01] WEISKOPF D.: *Visualization of Four-Dimensional Spacetimes*. PhD thesis, University of Tübingen, 2001. 2

[WMWW13] WÜNSCH A., MÜLLER T., WEISKOPF D., WUNNER G.: Circular orbits in the extreme Reissner-Nordstrøm dihole metric. *Phys. Rev. D 87* (2013), 024007. 4

[WSE04] WEISKOPF D., SCHAFHITZEL T., ERTL T.: GPU-based nonlinear ray tracing. *Comput. Graph. Forum 23*, 3 (2004), 625–634. 3