Review of Black Hole Ray Tracing Code

Introduction

Black holes represent regions in spacetime where gravitational forces are so intense that nothing, including light, can escape once it crosses the boundary known as the event horizon. These enigmatic objects, predicted by Albert Einstein's theory of general relativity, profoundly warp the fabric of spacetime in their vicinity. One of the most striking consequences of this extreme curvature is gravitational lensing, a phenomenon where the paths of light rays are bent as they pass through the strong gravitational field of a black hole. This bending can distort the appearance of objects located behind or around the black hole, creating fascinating visual effects.

Ray tracing is a powerful computer graphics technique employed to generate images of three-dimensional scenes by simulating the paths of light rays as they travel from a light source to an observer. By meticulously tracing the trajectory of each ray and accounting for its interactions with objects in the scene, ray tracing can produce highly realistic and detailed renderings. This technique is particularly well-suited for visualizing the complex effects of gravity around black holes, as it allows for the simulation of the curved paths of light rays in the warped spacetime.

The Schwarzschild metric, an exact solution to Einstein's field equations, describes the spacetime surrounding a non-rotating, spherically symmetric black hole. This metric is fundamental to understanding the gravitational effects of black holes and serves as the basis for many theoretical studies and numerical simulations. This report aims to provide a detailed expert review of the user's C++ code designed for black hole ray tracing. The analysis will focus on the accuracy of the implemented physical model, the efficiency of the computational approach, and potential areas for enhancement, drawing upon established research in the field of black hole visualization.

Code Structure and Overview

The provided C++ code implements a ray tracer designed to render a black hole and its accretion disk. The code is organized into several key functions that collectively simulate the path of light rays and determine the final image. The core functionality is distributed across the following functions: trace_rayStep, which calculates the gravitational deflection of a single ray segment; check_accretion_disk_intersection, which determines if a ray intersects the simulated accretion disk; cal_accretion_disk_colour, which computes the color of the accretion disk at the point of intersection; trace_black_hole_ray, which integrates the path of a ray through the spacetime, checks for intersections with the black hole and accretion disk, and determines the final color of the ray; and raytrace_blackhole, which sets up the virtual camera and iterates through each pixel of the output image to trace a corresponding ray into the scene.

The code includes several standard C libraries for basic input/output, memory allocation, and boolean data types. It also utilizes a custom math library, blackholemath.h, which presumably provides vector and other mathematical operations necessary for the ray tracing calculations. The SDL2 library is employed for managing the image surface and rendering pixels to the screen. Additionally, the code includes custom header files, raytracer.h and camera.h, which likely define structures and functions related to the overall ray tracing framework and the setup of the virtual camera used to view the scene.

Analysis of the Schwarzschild Metric Implementation

The trace_rayStep function is central to the simulation as it handles the crucial aspect of gravitational lensing by attempting to model the deflection of light rays as they pass near the black hole. The function begins by calculating the distance, denoted as r, of the current ray position from the center of the black hole. This distance is determined using the vec3_length function on a vector pointing from the origin to the ray's position. Following this, the code checks if the calculated distance r is less than or equal to the Schwarzschild radius, which is a parameter defining the event horizon of the black hole. If the ray's position falls within this radius, the function returns false, indicating that the ray has crossed the event horizon and is effectively absorbed by the black hole.[1]

The code then proceeds to calculate a deflection_factor using the formula 2.5 * params.schwarzschild_radius / (r * r). This formula represents a simplified approximation of the gravitational lensing effect. It is important to note that this formula is not directly derived from the geodesic equation of the Schwarzschild metric in general relativity. Instead, it bears resemblance to a Newtonian approximation, where the deflection angle is inversely proportional to the square of the distance.[1] The factor of 2.5 is likely included to approximate the apparent size of the black hole's shadow, which research suggests appears about 2.5 times larger than the actual Schwarzschild radius.[1] This suggests an approach focused on achieving a visually plausible result rather than a strict adherence to the complex mathematical framework of general relativity in this particular step.

To apply the deflection, the code first determines the direction from the current ray position to the black hole's center, termed radial_dir, by normalizing the negated position vector. The current ray direction is then decomposed into two components: one parallel to the radial_dir and the other perpendicular to it. This decomposition is achieved by calculating the dot product of the ray direction and the radial_dir to find the magnitude of the parallel component, which is then used to create the parallel_vector. The perpendicular_vector is obtained by subtracting the parallel_vector from the original ray direction. The gravitational deflection is then applied by scaling the perpendicular_vector by a factor of (1.0 - deflection_factor) to obtain deflection_perpendicular. The new ray direction is calculated by adding the parallel_vector and the deflection_perpendicular and then normalizing the resulting vector. Finally, the ray's position is updated by moving it forward along the new direction by a distance equal to the step_size.

The approach of applying the deflection solely to the perpendicular component of the ray's velocity is intuitively reasonable for simulating the bending of light towards a massive object. However, the specific mathematical form of the deflection_factor used in the code is a significant simplification compared to the predictions of general relativity. The geodesic equation in the Schwarzschild metric, which describes the paths of light rays, involves more intricate relationships between the inverse radius and conserved quantities such as angular momentum.[2] These complexities are not captured by the code's current approximation. More accurate simulations often rely on numerical integration of the geodesic equation or employ more sophisticated analytical approximations to determine the precise deflection of light.[4] Furthermore, the trace_rayStep function uses a fixed step_size provided as an argument. While the trace_black_hole_ray function, which calls trace_rayStep, does implement an adaptive step size, the core deflection calculation within trace_rayStep itself does not inherently adjust the step size based on the local gravitational field strength. This could potentially impact the accuracy of the ray's trajectory, especially in regions of rapidly changing gravitational fields close to the black hole.

Evaluation of the Ray Tracing Algorithm

The trace_black_hole_ray function orchestrates the overall process of tracing a single light ray from its origin to its termination. The function takes the initial ray origin and direction, along with the black hole parameters, as input. It initializes the current ray position and direction with these values and sets a default background color for the ray. The function also defines several parameters to control the ray tracing process, including a base_step_size, a maximum number of steps (max_steps), a maximum distance the ray is allowed to travel (max_distance), and a variable to track the total distance travelled by the ray.

The ray tracing is performed within a loop that iterates up to max_steps. In each iteration, the function first checks if the current ray intersects the accretion disk by calling the check_accretion_disk_intersection function. If an intersection occurs, the color of the ray is determined by calling the cal_accretion_disk_colour function at the intersection point, and the loop is terminated. If no intersection with the accretion disk is detected, the code proceeds to calculate an adaptive step_size. The formula used for this calculation is base_step_size * (1.0 + (Params.schwarzschild_radius / (r + epsilon)) * (Params.schwarzschild_radius / (r + epsilon))), where r is the current distance of the ray from the black hole, and epsilon is a small value to prevent division by zero. This adaptive step size mechanism is a crucial aspect of the ray tracing algorithm. As the ray approaches the black hole (i.e., r becomes smaller), the step size decreases. This is a sound strategy because the gravitational field is stronger and changes more rapidly closer to the black hole, requiring smaller steps to maintain accuracy in the ray's trajectory.[1] Conversely, as the ray moves further away from the black hole, the step size increases, which improves the efficiency of the simulation by allowing the ray to cover larger distances with fewer computational steps.

After calculating the adaptive step size, the function calls trace_rayStep to move the ray forward by this step and apply the gravitational lensing effect. The trace_rayStep function returns false if the ray crosses the event horizon. In this case, the trace_black_hole_ray function sets the color of the ray to black, representing the absorption of light by the black hole, and terminates the loop. The function also checks if the total distance_travelled by the ray exceeds the max_distance. If it does, a simple starfield effect is applied to the ray's color based on its final direction, and the loop is terminated. This provides a basic visual representation for rays that do not interact with the black hole or the accretion disk within the simulation's defined boundaries. It is worth noting that a commented-out section of the code includes a check based on the total distance from the original ray origin, vec3_length(vec3_sub(pos, ray_origin)) > max_distance. The current implementation uses the accumulated distance_travelled, which is the sum of the individual step sizes. Both approaches serve as a mechanism to prevent rays from being traced indefinitely, but the choice between them might depend on the specific requirements of the simulation. For instance, using distance_travelled might be more appropriate when the curvature of spacetime causes the ray to take a longer path than its direct distance from the origin would suggest. A more sophisticated approach to rendering the background would involve sampling a background star map based on the final direction of the ray, which could enhance the visual realism of the simulation.[8]

Review of the Accretion Disk Model

The code includes two functions dedicated to modeling the accretion disk: check_accretion_disk_intersection and cal_accretion_disk_colour. The check_accretion_disk_intersection function determines whether a given ray intersects the accretion disk, which is modeled as a flat disk lying in the XY plane. The function first checks if the Z component of the ray's direction is negligibly small. If it is, the ray is considered to be traveling parallel to the disk and will not intersect it, so the function returns false. If the Z component is significant, the function calculates the distance t along the ray to the point where it intersects the XY plane using the formula -pos.z / dir.z. If this distance t is negative, it means the intersection point is behind the ray's origin, so no intersection occurs, and the function returns false. Otherwise, the function calculates the coordinates of the intersection point in 3D space and then computes the distance of this point from the center of the black hole in the XY plane. Finally, it checks if this distance from the center falls within the defined inner and outer radii of the accretion disk and if the absolute value of the Z coordinate of the intersection point is within the specified thickness of the disk. If all these conditions are met, the function returns true, indicating an intersection, and also provides the distance to the intersection and the coordinates of the intersection point through the provided pointer arguments. This is a standard geometric approach to checking for intersection between a ray and a finite flat disk.

The cal_accretion_disk_colour function is responsible for determining the color of the accretion disk at a given intersection point. The color calculation is based on the distance r of the intersection point from the center of the disk in the XY plane and a simplified model of the relativistic Doppler effect. The code first calculates the angle of the intersection point in the XY plane using the atan2 function. This angle is then used to determine a doppler_factor. The calculation of this factor is highly simplified: if the sine of the angle is negative, the Doppler factor is set to 1.5; otherwise, it is set to 0.7. An additional term, (1.0 + 0.2 * cos(angle)), is then added to the Doppler factor. This model of the relativistic Doppler effect is a significant oversimplification of the actual physical phenomenon in an accretion disk. In reality, the relativistic Doppler effect depends on the orbital velocity of the material in the disk, which varies with the distance from the black hole, as well as the angle of observation and relativistic beaming effects.[10] The code's approximation, which depends only on the angle in the XY plane and uses arbitrary scaling factors, does not capture these complexities.

The function then calculates a base_intensity using the formula 3.0 * params.schwarzschild_radius / r. This intensity decreases linearly with the distance from the black hole's center. The factor of 3.0 might be empirically chosen, possibly related to the photon sphere, which for a non-rotating black hole is located at 1.5 times the Schwarzschild radius.[7] The final intensity is obtained by multiplying the base_intensity with the calculated doppler_factor. The intensity is then clamped to a maximum value of 1.0. Finally, this intensity value is used to determine the RGB color components of the accretion disk by scaling it and mapping it to specific red, green, and blue values using the SDL_MapRGB function. The use of a temperature profile for the accretion disk, where the temperature varies with radius, and modeling its emission as blackbody radiation would provide a more physically accurate way to determine the color.[24] Including the effect of gravitational redshift, which causes light escaping the black hole's strong gravitational field to lose energy and shift towards the red end of the spectrum, would also enhance the realism of the accretion disk's appearance.

Event Horizon Handling

The code correctly handles the event horizon by checking if the ray's distance from the black hole's center is less than or equal to the Schwarzschild radius within the trace_rayStep function. When this condition is met, the function returns false, signaling that the ray has been absorbed. Subsequently, the trace_black_hole_ray function, upon receiving this false value, sets the color of the corresponding pixel to black. This accurately reflects the physical property of a black hole where light cannot escape from within the event horizon.

Suggestions for Improvement

To enhance the accuracy and realism of the black hole ray tracing simulation, several improvements could be considered:

- **Gravitational Lensing:** The current simplified deflection model could be replaced with a more accurate approach based on the geodesic equation of the Schwarzschild metric. This could involve using analytical approximations for the deflection angle, which offer a balance between accuracy and computational cost.[2] Alternatively, numerical integration of the geodesic equation could be implemented for higher accuracy, although this would likely be more computationally intensive.[1] Implementing a more accurate lensing model would allow for the visualization of phenomena such as the photon sphere and higher-order lensing effects.
- **Accretion Disk Model:** The color model for the accretion disk could be significantly improved by incorporating a more physically accurate relativistic Doppler effect that takes into account the orbital velocity of the disk material as a function of radius. Including the effect of gravitational redshift on the light

emitted from the disk would also enhance realism. Furthermore, considering relativistic beaming, which affects the apparent brightness of the rapidly moving disk material, could add to the visual fidelity.[10] Exploring the use of a temperature profile for the accretion disk and modeling its emission as blackbody radiation would provide a more physically grounded approach to color determination.[24]
- **Ray Tracing Algorithm:** More sophisticated adaptive step size control methods could be investigated. Techniques based on error estimation could potentially offer better accuracy and efficiency compared to the current heuristic approach.[34] Implementing a mechanism to detect when a ray becomes trapped in orbit around the black hole, particularly within the photon sphere, could prevent infinite loops and improve the robustness of the simulation.
- **Additional Features:** Adding the option to include background stars by tracing rays to infinity and sampling a star map based on the final ray direction would enhance the visual context of the black hole.[8] Implementing more detailed accretion disk textures or using procedural generation techniques could create more visually interesting and dynamic disks.[10] For a more advanced simulation, allowing the user to specify the black hole's spin (Kerr metric) could be considered, although this would significantly increase the complexity of the physical model and the required computations.[1]

Conclusion

The provided C++ code represents a commendable effort towards implementing a black hole ray tracer. The inclusion of an adaptive step size mechanism and a basic model for the accretion disk are notable strengths. However, the gravitational lensing and Doppler effect calculations rely on significant simplifications that deviate from the more complex predictions of general relativity. The empirical factors used in these calculations suggest an approach aimed at visual plausibility. For future development, prioritizing a more accurate model for gravitational lensing and enhancing the accretion disk's color model to include relativistic effects like Doppler shift and gravitational redshift would significantly improve the physical accuracy and visual realism of the simulation. Depending on the user's specific goals for accuracy, realism, and performance, a prioritized approach to implementing these and other suggested improvements would be beneficial.

Valuable Tables to Include:

Feature	User's Code	More Accurate Models	Relevant Snippets
Gravitational Lensing	Simplified Newtonian approximation with empirical factor	Numerical integration of geodesic equation or analytical approximations of Schwarzschild metric	[1]
Accretion Disk Doppler Effect	Simplified model based on angle in XY plane with arbitrary factors	Relativistic Doppler shift based on orbital velocity and viewing angle	[10]
Accretion Disk Color	Based on distance and simplified Doppler factor, mapped to RGB	Blackbody radiation based on temperature profile, considering gravitational redshift and relativistic beaming	[24]
Step Size Control	Adaptive based on distance from black hole	More sophisticated adaptive methods based on error estimation	[34]

Works cited

1. Approximate Black Hole Renderings with Ray Tracing - Rensselaer Polytechnic Institute, accessed on May 5, 2025, https://www.cs.rpi.edu/~cutler/classes/advancedgraphics/S21/final_projects/qianm.pdf
2. Schwarzschild geodesics - Wikipedia, accessed on May 5, 2025, https://en.wikipedia.org/wiki/Schwarzschild_geodesics
3. Integrating the Schwarzschild geodesics equations - Physics Stack Exchange, accessed on May 5, 2025, https://physics.stackexchange.com/questions/314230/integrating-the-schwarzschild-geodesics-equations
4. Ray Tracing Schwarzschild & Kerr Black Holes - Undergraduate Lab @ Berkeley (ULAB), accessed on May 5, 2025, https://ulab.studentorg.berkeley.edu/static/doc/posters/s234.pdf
5. Approximating light rays in the Schwarzschild field - Inspire HEP, accessed on May 5, 2025, https://inspirehep.net/literature/1334876
6. arxiv.org, accessed on May 5, 2025, https://arxiv.org/abs/1909.05732

7. Ray Tracing Black Holes – cs184 - (Eliot) Han, accessed on May 5, 2025, https://eliot1019.github.io/Black-Hole-Raytracer/
8. Interactive Black-Hole Visualization - TU Delft Research Portal, accessed on May 5, 2025, https://pure.tudelft.nl/ws/files/84313235/blacktracer_IEEEVis_final_version.pdf
9. ahiser1117/Blackhole-Renderer: 3D Rendering Engine - GitHub, accessed on May 5, 2025, https://github.com/ahiser1117/Blackhole-Renderer
10. Black Hole Accretion Disk Visualization - NASA SVS, accessed on May 5, 2025, https://svs.gsfc.nasa.gov/13326/
11. Relativistic accretion disks in active galactic nuclei - Imagine the Universe! - NASA, accessed on May 5, 2025, https://imagine.gsfc.nasa.gov/observatories/satellite/asca/agndisk.html
12. Black Hole Anatomy - NASA Science, accessed on May 5, 2025, https://science.nasa.gov/universe/black-holes/anatomy/
13. Relativistic beaming - Wikipedia, accessed on May 5, 2025, https://en.wikipedia.org/wiki/Relativistic_beaming
14. Broad iron lines in AGN - Fabian, Iwasawa, Reynolds, Young, accessed on May 5, 2025, https://ned.ipac.caltech.edu/level5/Fabian4/Fab2_2.html
15. Line shifts in accretion disks - the case of Fe Kα - arXiv, accessed on May 5, 2025, https://arxiv.org/pdf/1601.02389
16. Class 17 : Jets!, accessed on May 5, 2025, https://www.astro.umd.edu/~miller/teaching/astr350/lecture17.pdf
17. Relativistic Doppler Effect - Physics Bootcamp, accessed on May 5, 2025, https://www.physicsbootcamp.org/sec-RelativisticDopplerEffect.html
18. Doppler Tomography of Relativistic Accretion Disks, accessed on May 5, 2025, https://academic.oup.com/pasj/article-pdf/48/5/771/9714126/pasj48-0771.pdf
19. Relativistic Doppler effect - Wikipedia, accessed on May 5, 2025, https://en.wikipedia.org/wiki/Relativistic_Doppler_effect
20. Accretion disk - Wikipedia, accessed on May 5, 2025, https://en.wikipedia.org/wiki/Accretion_disk
21. Relativistic iron lines in accretion disks: the contribution of higher order images in the strong deflection limit - arXiv, accessed on May 5, 2025, https://arxiv.org/pdf/1607.05365
22. General relativistic spectra of accretion discs around rapidly rotating neutron stars: effect of light bending | Monthly Notices of the Royal Astronomical Society | Oxford Academic, accessed on May 5, 2025, https://academic.oup.com/mnras/article/325/3/989/961777
23. Raytracing a Black Hole, accessed on May 5, 2025, https://rantonels.github.io/starless/
24. Accretion discs - Scholarpedia, accessed on May 5, 2025, http://www.scholarpedia.org/article/Accretion_discs
25. The true colour of black hole disks - Max-Planck-Gesellschaft, accessed on May 5, 2025, https://www.mpg.de/569206/pressRelease200807231
26. Exploring Color Magnitude Diagrams for AGN Accretion Disks - Quinnipiac University, accessed on May 5, 2025,

https://iq.qu.edu/experiential-learning/course-projects-and-capstones/student-projects/exploring-color-magnitude-diagrams-for-agn-accretion-disks/
27. Modeling the accretion disk X-ray continuum of black hole candidates - Purdue e-Pubs, accessed on May 5, 2025, https://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=2440&context=physics_articles
28. Planck Curves from Multi-Color Accretion Disk Model : r/astrophysics - Reddit, accessed on May 5, 2025, https://www.reddit.com/r/astrophysics/comments/mvv3vp/planck_curves_from_multicolor_accretion_disk_model/
29. Home - Google Sites, accessed on May 5, 2025, https://sites.google.com/berkeley.edu/cs184-final-project-final-repo/home
30. Accretion disk | Astronomy & Formation Processes - Britannica, accessed on May 5, 2025, https://www.britannica.com/science/accretion-disk
31. ACCRETION DISKS Max-Planck-Institut für Astrophysik Postfach 1523, D-85740 Garching, Germany Abstract. In this lecture the basi, accessed on May 5, 2025, https://wwwmpa.mpa-garching.mpg.de/~henk/pub/disksn.pdf
32. Stone's Accretion Disk Research, accessed on May 5, 2025, https://www.astro.princeton.edu/~jstone/disks.html
33. Someone made a simulation of a black hole more accurate than Interstellar with the relativistic doppler effect : r/Physics - Reddit, accessed on May 5, 2025, https://www.reddit.com/r/Physics/comments/k33yb1/someone_made_a_simulation_of_a_black_hole_more/
34. Magrathea-Pathfinder: A 3D adaptive-mesh code for geodesic ray tracing in N-body simulations - arXiv, accessed on May 5, 2025, https://arxiv.org/html/2111.08744v4
35. Adaptive Step Size Selection for Homotopy Methods to Solve Polynomial Equations† - Michael Shub, accessed on May 5, 2025, https://shub.ccny.cuny.edu/articles/2012-Adaptive_Step_Size.pdf
36. Adaptive mesh and geodesically sliced Schwarzschild spacetime in 3+1 dimensions - MPG.PuRe, accessed on May 5, 2025, https://pure.mpg.de/rest/items/item_153043/component/file_153042/content
37. Explicit symplectic integrators with adaptive time steps in curved spacetimes - arXiv, accessed on May 5, 2025, https://arxiv.org/html/2412.01045v1
38. arXiv:1903.00990v1 [gr-qc] 3 Mar 2019, accessed on May 5, 2025, https://arxiv.org/pdf/1903.00990
39. 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 - World Scientific Publishing, accessed on May 5, 2025, https://www.worldscientific.com/doi/10.1142/S0218271819500421
40. [1511.06025] 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 - arXiv, accessed on May 5, 2025, https://arxiv.org/abs/1511.06025
41. 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 -

Semantic Scholar, accessed on May 5, 2025, https://www.semanticscholar.org/paper/Seeing-relativity-I%3A-Ray-tracing-in-a-Schwarzschild-Riazuelo/00707a9183a5f24a2b3411dcc413795d6b97a38b
42. Real-time High-Quality Rendering of Non-Rotating Black Holes - arXiv, accessed on May 5, 2025, https://arxiv.org/pdf/2010.08735
43. Black Hole - Matt Ebb, accessed on May 5, 2025, https://mattebb.cargo.site/Black-Hole
44. Black Hole Accretion Disk Visualization - NASA SVS, accessed on May 5, 2025, https://svs.gsfc.nasa.gov/13326
45. Black Hole with Accretion Disk Visualization - NASA SVS, accessed on May 5, 2025, https://svs.gsfc.nasa.gov/14619/
46. NASA creates stunning new black hole visualization | Space - EarthSky, accessed on May 5, 2025, https://earthsky.org/space/black-hole-visualization-from-side-schnittman-goddard/
47. Black Hole Accretion Disk Visualization | Credit: NASA's Goddard Space Flight Center, Jeremy Schnittman : r/spaceporn - Reddit, accessed on May 5, 2025, https://www.reddit.com/r/spaceporn/comments/n7p7zk/black_hole_accretion_disk_visualization_credit/
48. Realistic rendering of a supermassive black hole & accretion disk for Christopher Nolan's "Interstellar" [1800x1200] : r/spaceporn - Reddit, accessed on May 5, 2025, https://www.reddit.com/r/spaceporn/comments/2k10px/realistic_rendering_of_a_supermassive_black_hole/
49. Artist's rendering of a black hole - unawe.org, accessed on May 5, 2025, https://www.unawe.org/resources/images/Black_Hole/
50. General Relativity 3: Volumetric Accretion Disks and Live Event! - SpaceEngine, accessed on May 5, 2025, https://spaceengine.org/news/blog220830/
51. Final Deliverable: Black Hole Raymarcher - Lawrence Zhao, accessed on May 5, 2025, https://celticspwn.github.io/CS184FinalProject/final.html
52. Blender 3D Mastering Black Holes Episode 3: Accretion Disk Part 2 - YouTube, accessed on May 5, 2025, https://www.youtube.com/watch?v=jWPW1sMc3P0
53. Black Hole Raytracer | CS 184, accessed on May 5, 2025, https://cyang2020.github.io/BlackHoleRayTracer/
54. Making cool animations of an accretion disk/planetary disk - Mathematica Stack Exchange, accessed on May 5, 2025, https://mathematica.stackexchange.com/questions/215416/making-cool-animations-of-an-accretion-disk-planetary-disk
55. Rendering a Black Hole in Unreal Engine 5 - YouTube, accessed on May 5, 2025, https://www.youtube.com/watch?v=B2vPGUi7wq8
56. general relativity raytracing, accessed on May 5, 2025, https://www.techfreaq.de/grraytracing.html
57. Kerr-Newman black hole null geodesic ray tracing (part 3) - YouTube, accessed on May 5, 2025, https://www.youtube.com/watch?v=yuim-5-nb_8
58. Astray: A Performance-Portable Geodesic Ray Tracer - VR, RWTH Aachen,

accessed on May 5, 2025, https://vr.rwth-aachen.de/media/papers/242/Astray_A_Performance-Portable_Geodesic_Ray_Tracer.pdf
59. Ray Tracing: - Black Hole PIRE, accessed on May 5, 2025, http://bhpire.arizona.edu/wp-content/uploads/2020/10/2020-10-27_Ray-Tracing_Reduced.pdf
60. Spacetime curvature due to a black hole : r/GraphicsProgramming - Reddit, accessed on May 5, 2025, https://www.reddit.com/r/GraphicsProgramming/comments/1j2mq61/spacetime_curvature_due_to_a_black_hole/
61. Black Hole Images as Tests of General Relativity: Effects of Spacetime Geometry - UCL Discovery, accessed on May 5, 2025, https://discovery.ucl.ac.uk/10163636/1/Younsi_Black%20Hole%20Images%20as%20Tests%20of%20General%20Relativity_VoR.pdf
62. Topics Geometry and Topology – MATH 599 Visualization of Kerr black holes, accessed on May 5, 2025, https://www.math.mcgill.ca/gantumur/math599w17/project-kerr.pdf
63. Geodesic structure of a rotating regular black hole - ResearchGate, accessed on May 5, 2025, https://www.researchgate.net/publication/370840144_Geodesic_structure_of_a_rotating_regular_black_hole
64. Parallel Computing in High Energy Astrophysics: Combining ... - SC13, accessed on May 5, 2025, http://sc13.supercomputing.org/sites/default/files/PostersArchive/tech_posters/post129s2-file2.pdf
65. Orbits Around a Spinning Black Hole, accessed on May 5, 2025, https://wittman.physics.ucdavis.edu/Animations/Kerr/index.html
66. Coding Challenge 144: 2D Black Hole Visualization - YouTube, accessed on May 5, 2025, https://www.youtube.com/watch?v=Iaz9TqYWUmA
67. Researchers create 3-D simulations of the collision of two supermassive black holes | College of Liberal Arts & Sciences at Illinois, accessed on May 5, 2025, https://las.illinois.edu/news/2016-07-05/researchers-create-3-d-simulations-collision-two-supermassive-black-holes
68. MALBEC: a new CUDA-C ray-tracer in General Relativity - arXiv, accessed on May 5, 2025, https://arxiv.org/pdf/1803.08320