

Graviational Lensing for a Schwarzschild Black Hole using Backwards Ray Tracing

General relativity says that matter tells space how to bend, and space tells matter how to move. Assuming a cosmological constant of zero, this relation is governed by the Einstein field equations:

$$R_{\mu\nu} - \frac{1}{2}R = 8\pi GT_{\mu\nu} \quad (1)$$

The Schwarzschild solution is the simplest case; a completely stationary mass centered at the origin, whose metric (using $- , + , + , +$) in spherical coordinates is well known to be

$$ds^2 = -\left(1 - \frac{r_s}{r}\right) dt^2 + \left(1 - \frac{r_s}{r}\right)^{-1} dr^2 + r^2 d\theta^2 + r^2 \sin^2 \theta d\phi^2 \quad (2)$$

The spherical symmetry of the problem allows us to calculate the geodesics in a plane ($\theta = \pi/2$). We can write a pair of differential equations for r and ϕ in terms of the conserved quantities energy E and angular momentum L :

$$\ddot{\phi} = \frac{L}{r^2}, \quad \ddot{r} = -\frac{R_s}{2r^2} \left(\frac{L^2}{r^2} + \epsilon \right) + \frac{L^2}{r^3} \left(1 - \frac{R_s}{r} \right) \quad (3)$$

Where the quantity $\epsilon = 0$ since light travels on null geodesics. These equations can be solved numerically on a computer to generate the path that light takes in the presence of a Schwarzschild black hole.

Looking towards a background galaxy with a black hole in front of it will caused the resulting image for the observer to be distorted, as the mass of the black hole bends light around it in such a way that it is possible to see two (or more) images of the same light source. This effect is normally done computationally using a thin lens approximation to derive formulas for the angle of deflection which can then be used to distort an image (which is used as the background galaxy) as if it emitted parallel light rays towards black hole.

The lensing effect should be reproducible using ray tracing methods as well. By considering each pixel of an image (as the background galaxy) and casting rays out toward the black hole and hit some target plane (observer) to map the pixels of the image their corresponding locations in the target plane where the trajectories can be obtained using the light geodesics discussed above.

Approach

Figuring out the photon trajectories comes down to integrating the equations of motion 3 with the either the [Richardson extrapolation](#) (along with the usual Runge-Kutta method) or the [Runge-Kutta-Fehlberg](#) method (also known as RK45) for dynamic time stepping and increased rate of convergence, whichever ends up working. The energy E and angular momentum L are both determined by initial conditions.

The ray tracing method used will be backwards ray tracing. The reasoning is as follows: ray tracing involves casting out several rays, then letting hit surfaces, bounce around, and so on until they reach the position of some observer. This process is described as forward ray tracing, and is generally very inefficient. This is because most of the rays cast from the light source will never actually reach the camera, perhaps being reflected to somewhere far off or obstructed by some black body. In turn this uses a lot of unnecessary computation and so to remedy this, we take rays cast from the observer and track what they hit instead. These rays are guaranteed to have “hit” the camera, since they started there to begin with, but more formally: this solution works because running time backwards ($t \rightarrow -t$) will give the effect of reversing the direction the ray took to get to the object. In other words, it makes it look like the object emitted the rays which then bounced around and hit the camera.

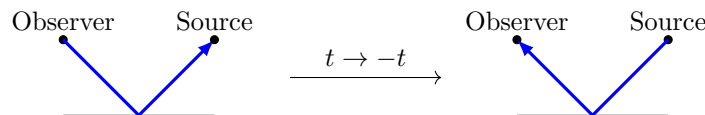


Figure 1: Backward Ray Tracing

In particular, time reversal works for light geodesics, as the they are (with valid initial conditions) unique for the Schwarchild metric. We will make an image by sending out rays with initial conditions backwards from the start plane toward the black hole, let it do whatever it does and eventually one of four outcomes can occur.

1. It enters the black hole, which we set that pixel to black and terminate the calculation, since once inside the black hole it can never leave.

2. It hits the image, in which case we take the color of the pixel and map it back to where the ray started.
3. It shoots off to infinity (out of bounds) in which we also set to black. (Note: While debugging, we may set the pixel to a different color depending on whether it enters the black hole or goes out of bounds.)
4. It happens to be a special trajectory (or near one) whose solutions end up orbiting the black hole indefinitely, or for a long period of time, which can be solved by enforcing a limit to the number of iterations in the integrator.

In theory, should allow us to get the same distortion effects that are obtained using lensing equations and angles of deflection (which seems to be the most common way to do this).

Objectives

1. Simulate and render a 3D trajectory for a single photon.
2. Check simulation by using known solutions (i.e. ICs where light orbits the black hole are known, so setting those in the simulation should match).
3. Generate an image displaying the lensing effect using a “background galaxy” (which will probably be an image off of Google). Or if this is too computationally expensive, use a lower resolution image. Using a checkerboard pattern as the image will allow us to clearly see the lensing effect.
4. Compare the results with other (physically correct) lensing simulations (like this one from [NASA](#), [Harvard](#), or some of the examples on [Wikipedia](#)) and see how well it matches.

Stretch Goals

1. Translate or tilt the image it at an angle to simulate changing the perspective of the lensing and make a short video displaying the lensing effect.
2. Lens with a different metric (Kerr, Kerr-Newmann, etc.) for a rotating or charged (or a combination of the two) black hole.

Approval

- Ethan approves of this proposal!
-