Accelerated Ray Tracing in Schwarzschild Spacetime

Christian Daley
UC Santa Barbara
2019

Abstract

Einstein's theory of general relativity predicts that the presence of matter and energy curves spacetime. The Schwarzschild metric is an exact solution to the Einstein field equations that describes the curvature of spacetime around a spherically symmetric, non-rotating, non-charged mass. In order to accurately render images of massive gravitational objects such as black holes the effects of curved spacetime on light rays must be taken into account. The path that a light ray takes through spacetime is called a null geodesic, and typically these null geodesics are computed numerically. A method for accelerating the computation of null geodesics by exploiting the symmetries of Schwarzschild spacetime is presented. The ability to quickly compute null geodesics is used to reduce the work necessary to perform intersection tests between curved light rays and objects in the scene being rendered.

Contents

1. Introduction

1.1. Traditional ray tracing

Traditional ray tracing is a rendering technique that seeks to accurately simulate lighting effects by tracing the paths that light rays take through a scene. This includes simulating the interactions of the light rays with any objects in the scene that are encountered along the way. As its name implies, ray tracing considers the path that a photon follows to be a mathematical ray: a half line determined by a starting location and a direction. The formula for a light ray in a Cartesian coordinate system is given by

$$x(t) = O + Dt$$

where $O \in \mathbb{R}^3$ is the initial position of the light, $D \in \mathbb{R}^3$ is a unit vector representing the direction/velocity of the light, $t \in [0, \infty)$ is a "time" parameter, and $x(t)$ is the location of the light at time t. This equation is a simple closed form expression for the trajectory of a single photon. It is easy to see that $x(t)$ can be computed in constant time. Additionally, it is possible to compute the intersection of $x(t)$ and a primitive object in the scene, such as a triangle or sphere, in constant time. This means that given an arbitrary light ray and some object in the scene, we can always quickly determine if and where the light ray intersects the object, regardless of how far apart the object and origin of the light ray are.

1.2. Curvature of light rays due to gravity

Einsten's theory of general relativity describes spacetime as a four dimensional differentiable manifold equipped with a Lorentzian metric and torsion free metric connection. In the absence of an outside force, the path of any massive or massless particle through spacetime must satisfy the geodesic equation:

$$\frac{d^2x^\lambda}{ds^2} + {\Gamma^\lambda}_{\gamma\rho}\frac{dx^\gamma}{ds}\frac{dx^\rho}{ds} = 0$$

where x^λ is position in spacetime according to the chosen coordinate system, s is an affine parameter, and ${\Gamma^\lambda}_{\gamma\rho}$ are the unique connection coefficients of the Levi-Civita connection, also called the Christoffel symbols. This equation uses Einstein summation notation, where repeated indices in the same term are implicitly summed over. In flat spacetime with a Cartesian coordinate system Γ vanish everywhere, leaving the equation $\frac{d^2x^\lambda}{ds^2} = 0$. This is the

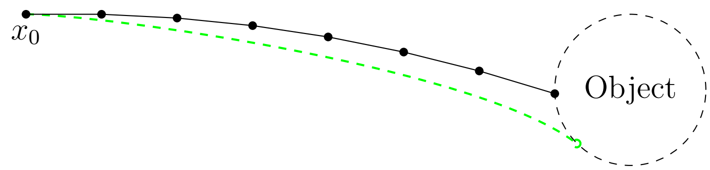

Figure 1: Intersection between a curved ray and an object. The dashed green line is the true trajectory of the light ray. The black line segments are a numerical approximation of the light ray obtained by continually marching the ray forward by a fixed timestep. As each of these line segments are computed they are checked for intersection with the object.

familiar equation of a "straight line", a curve with a second derivative of zero everywhere. According to the Einstein field equations the presence of matter and energy creates intrinsic curvature in the spacetime manifold. In such curved spacetime it is not possible to choose a coordinate system such that Γ vanish everywhere.

1.3. Tracing curved light rays

Traditional ray tracing relies on the assumption that the path of a photon can be accurately described by the equation $x(t) = O+Dt$. This equation has a second derivative of zero everywhere, indicating that this equation is valid only when the photon is traveling through flat spacetime. To accurately trace the path of light through curved spacetime the full geodesic equation must be solved. Unfortunately, in most cases it is not possible to obtain a closed form expression for $x(t)$ given arbitrary initial conditions. Furthermore, it is not clear how we can perform an intersection test between $x(t)$ and an object in the scene if we do not even know what form the curve $x(t)$ takes.

In practice we can use numerical methods to solve for $x(t)$ given initial conditions (x_0, x'_0). The basic process is to continually march the ray forward by a chosen timestep of Δt, using the geodesic equation to estimate changes to the ray's velocity over time. If Δt is chosen to be small enough then the computed value of $x(t)$ should be close to its true value. This process also provides for a simple way to perform intersection tests between curved rays and objects in the scene. Given the numerically computed values $x(t)$ and $x(t+\Delta t)$, we consider the line segment with endpoints $x(t)$ and $x(t+\Delta t)$ and check for intersections between this line segment and objects in the scene. As the ray is continuously marched forward we check for intersections along each of the line segments that are created.

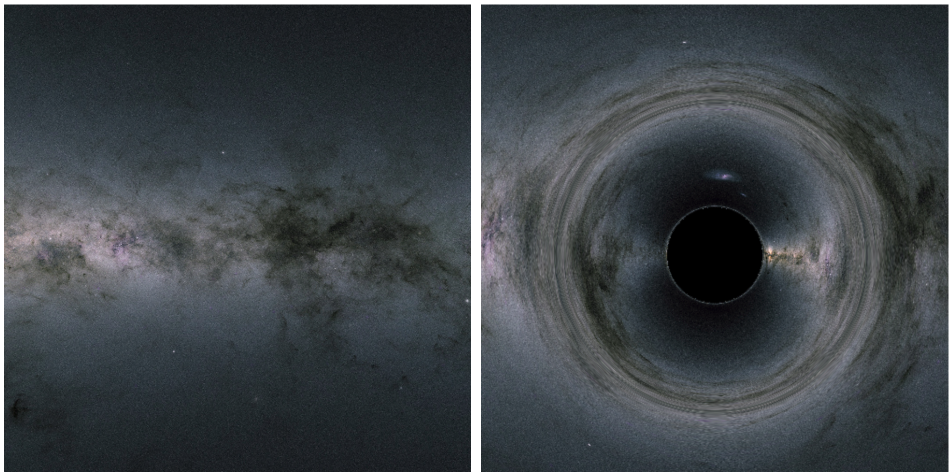

Figure 2: Two 500x500 images of stars both rendered with 1 ray/pixel. The image on the left took 0.05 seconds to render. The image on the right depicts a Schwarzschild black hole in front of the stars. It took 2.47 seconds to render using the ray marching technique with an adaptive step size.

1.4. Practical aspects of tracing curved rays

Numerical integration of the geodesic equation is straightforward to implement, but it does suffer from practical problems. In the case of flat spacetime $x(t)$ can be computed in constant time. However, when using numerical integration $x(t)$ can only be computed in time that is linear in t. This also means that the time it takes to determine if $x(t)$ intersects any particular object also requires time that is linear in t. For example, consider the case of a ray that intersects some particular object at a parameter distance of $t = 1$. Now suppose we move that object so that the ray now intersects it at a parameter distance of $t = 100$. Using our numerical technique it will now take 100 times longer to determine where the ray intersects the object, a problem that does not occur when tracing straight rays. This greatly increases the amount of time it takes to render scenes in curved spacetime compared to flat spacetime.

2. Schwarzschild spacetime

2.1. The Schwarzschild metric

The Schwarszchild metric was discovered by Karl Schwarzschild shortly after Einstein published his thoery of general relativity. It was the first exact nontrivial solution to the Einstein field equations. It describes the geometry of spacetime around a spherically symmetric mass that is not rotating and not electrically charged. The Schwarzschild metric is static, meaning that the geometry does not change over time, and it is also spherically symmetric. The metric uses coordinates $x^\lambda = (t, r, \theta, \varphi)$, where t is time, r is distance from the center of the mass, θ is the azimuthal angle, and φ is the polar angle, all measured by a stationary observer infinitely far from the massive object (time and distance will be measured differently by observers close to the object). The metric g for a gravitating mass M is given as

$$g_{\mu\nu} dx^\mu dx^\nu = -c^2 d\tau^2 = -\left(1-\frac{r_s}{r}\right)c^2 dt^2 + \left(1-\frac{r_s}{r}\right)^{-1} dr^2 + r^2\left(d\theta^2 + \sin^2(\theta) d\varphi^2\right)$$

where $-c^2 d\tau$ is the spacetime interval and $r_s = \frac{2GM}{c^2}$ is the "Schwarzschild radius" of the massive object, with G being Newton's gravitational constant and c being the speed of light in a vacuum. Notice that the metric becomes undefined at $r = r_s$. Any object with a radius smaller than its Schwarszchild radius is called a black hole. The spherical region of space around a Schwarzschild black hole where $r = r_s$ is called the event horizon.

It can be shown that all geodesics through Schwarzschild spacetime lie in a plane. When considering a specific geodesic we can always choose coordinates such that $\theta = \frac{\pi}{2}$ and $d\theta = 0$. The metric simplifies to

$$-c^2 d\tau^2 = -\left(1-\frac{r_s}{r}\right)c^2 dt^2 + \left(1-\frac{r_s}{r}\right)^{-1} dr^2 + r^2 d\varphi^2$$

2.2. Numerically computing null geodesics

The path of a ray of light though Schwarzschild spacetime is determined by the initial conditions of the light ray and the Schwarzschild radius of the gravitating mass. Such a path can be represented by the function μ.

$$\mu(x_0, x'_0, r_s, t) \to (x_t, x'_t)$$

The function μ gives the position and velocity at time t of a light ray with an initial position and velocity of (x_0, x'_0). μ can be computed numerically by iterating the STEP function, defined below.

function STEP$(x_0, x_0', r_s, \Delta t)$

$\hat{r} := \frac{x_0}{||x_0||}$

$x'_{\Delta t} := x_0' + \Delta t \cdot F(x_0, x_0', r_s) \cdot \hat{r}$

$x_{\Delta t} := x_0 + \Delta t \cdot \frac{x_0' + x'_{\Delta t}}{||x_0' + x'_{\Delta t}||}$

return $(x_{\Delta t}, x'_{\Delta t})$

end function

STEP computes $(x_{\Delta t}, x'_{\Delta t})$ given $(x_0, x_0', r_s, \Delta t)$. It works by first computing $x'_{\Delta t}$, the velocity at the end of the timestep, by adding the initial velocity vector to a radial "force vector". Then it advances the position according to the normalized sum of the initial and final velocity vector. STEP relies on the function $F(x_0, x_0', r_s)$ which is the instantaneous radial acceleration due to gravity of a particle with position x_0 and velocity x_0'. This function must be calculated using the Schwarzschild metric and the geodesic equation. More details on this function are provided in section 5.

A ray tracer can by implemented by using STEP to numerically compute μ. In order to trace a ray we repeatedly call the STEP function. Before each call to STEP we check to see if the ray has fallen below the event horizon, and if it has we stop tracing the ray (the resulting pixel will be black). After each call to STEP we check to see if any objects in the scene have been intersected by the ray and we handle the collisions.

2.3. The gravitational sphere of influence

Curved rays cannot be traced to infinity like straight rays. When tracing a curved ray we pick a certain radius r_{max} beyond which the ray is no longer considered curved. As long as the ray is further from the origin than r_{max} it is treated as a straight ray. A ray with an initial radius $r > r_{max}$ is traced until it intersects with the "gravitational sphere of influence", after which it is traced as a curved ray until it intersects an object, falls past the event horizon, or leaves the gravitational sphere of influence. If the ray never intersects the sphere of influence then it is just traced as a completely straight ray.

3. Accelerating the computation of light paths

3.1. The basic idea

Using STEP to numerically compute μ has a running time of $O(t)$. We now describe a method for estimating μ in constant time. The basic idea is

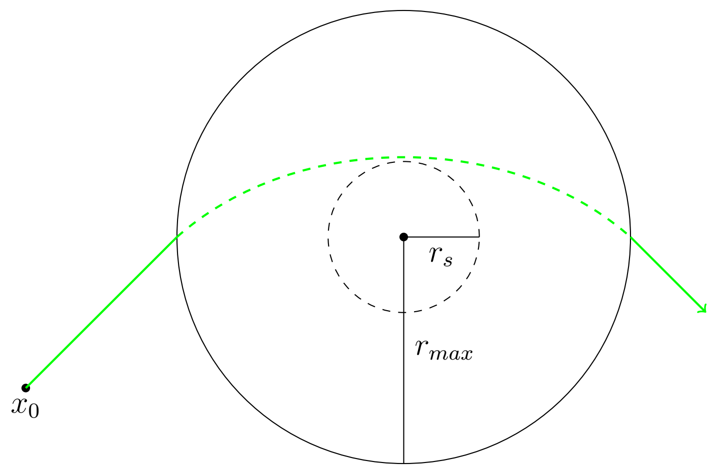

Figure 3: Tracing a light ray with initial radius $||x_0|| > r_{max}$. The ray is considered to be straight until it intersects with the gravitational sphere of influence. Then it is traced as a curved ray until it leaves the sphere of influence.

to sample μ at a discrete number of points and save the results. Then μ can be estimated in constant time by simply interpolating between known values of μ. If our samples are dense enough the resulting error should be within an acceptable range. However, it is not feasible to sample μ directly because μ has too many parameters. Both x_0 and x'_0 are elements of $\mathbb{R}^3$, so specifying x_0 and x'_0 requires six parameters. Additionally we must specify r_s and t, bringing the total number of parameters to eight. Densely sampling between these eight parameters is not feasible. Fortunately, the highly symmetrical nature of the Schwarzschild metric reduces the effective dimensionalty of μ to three.

3.2. Using the symmetries of the Schwarzschild metric

As mentioned earlier, all geodesics through Schwarzschild spacetime lie in a plane. When computing the path of a specific light ray we can confine ourselves to working within this orbital plane. We create a polar coordinate system (r, φ) on the plane. Due to the spherical symmetry of the Schwarzschild metric we can always choose this coordinate system in such a way that the φ component of x_0 is zero. Additionally, we scale the radial coordinate r such that a point on the event horizon has a radial component of 1, i.e r_s is considered to be the radial unit length. Now we perform a coordinate transformation on x_0. We find its components in this new coordinate system:

$$x_0 \equiv (r_0, 0)$$

where $r_0 = \frac{||x_0||}{r_s}$. The vector x_0' is a unit vector and can be described by a single angle α, measured relative to the vector pointing from the origin to x_0. For example, $\alpha = 0$ corresponds to a light ray moving directly away from the origin and $\alpha = \pi$ corresponds to a light ray moving directly towards the origin. The orientation of the φ coordinate is always chosen such that $0 \leq \alpha \leq \pi$. The values r_0 and α uniquely determine x_0 and x_0' within this coordinate system.

The components of x_t are

$$x_t \equiv (r_t, \beta)$$

where $r_t = \frac{||x_t||}{r_s}$. The values r_t and $\beta \in [0, 2\pi)$ uniquely determine x_t. x_t' can also be described by a single angle $\gamma \in [0, 2\pi)$ in the same way that x_0' is given by α. The last step of our coordinate transformation involves the time coordinate. Time must be scaled by a factor of $\frac{1}{r_s}$ because we also scaled distances by this factor. We get a new time coordinate $t_s = \frac{t}{r_s}$. Thus, instead of expressing μ as

$$\mu(x_0, x_0', r_s, t) \rightarrow (x_t, x_t')$$

we can instead express it as a new function ψ.

$$\psi(r_0, \alpha, t_s) \rightarrow (r_t, \beta, \gamma)$$

ψ can be thought of as a version of μ where redundant degrees of freedom have been eliminated. Rotating x_0 and x_0' about the origin, reflecting the coordinates about an axis, or scaling distance and time by a constant factor leaves r_0, α, t_s, r_t, β and γ unchanged.

Being able to compute ψ is sufficient to be able to compute μ. Given x_0, x_0', r_s and t we simply perform a coordinate transformation

$$(x_0, x_0', r_s, t) \rightarrow (r_0, \alpha, t_s)$$

Then we evaluate $\psi(r_0, \alpha, t_s)$ to get (r_t, β, γ) and perform an inverse coordinate transformation

$$(r_t, \beta, \gamma) \rightarrow (x_t, x_t')$$

to obtain the final result.

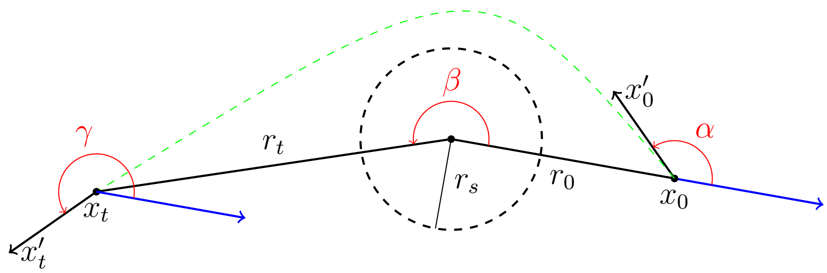

Figure 4: visualization of ψ. Distances are normalized such that $r_s = 1$. The dotted green line represents the trajectory of the light ray over a period of time t. The blue vectors have a φ component of zero. The angle α defines the initial velocity x'_0 and the angle γ defines the final velocity x'_t. The distance r_0 defines the initial position x_0. The distance r_t and the angle β define the final position x_t. From this image it can be seen that rotating x_0 and x'_0 about the origin or reflecting across an axis will leave the angles unchanged because all angles are measured relative to x_0. Additionally, scaling distance and time by a constant factor will leave all values unchanged because distance and time are measured relative to r_s.

3.3. Sampling the ψ function

ψ is a function with three parameters. Sampling ψ is feasible and is equivalent to sampling μ. To sample ψ we fix a specific value of r_s and also fix a polar coordinate system (r, φ) on a plane. Then we pick three sets S_r, S_α, and S_t which contain all of the values of r_0, α and t_s that we wish to sample over. For each possible $(r_0, \alpha, t_s) \in S_r \times S_\alpha \times S_t$ we perform a coordinate transformation $(r_0, \alpha, t_s) \to (x_0, x'_0, r_s, t)$ and obtain (x_t, x'_t) by numerically computing μ. (x_t, x'_t) is transformed to (r_t, β, γ) and the values $(r_0, \alpha, t_s, r_t, \beta, \gamma)$ are saved in a file to be used later.

In the provided code, r_0 is sampled over values ranging from 1.05 to 30, with more dense sampling for smaller numbers. α is sampled in increments of $1°$, from $0°$ to $180°$. t_s is sampled in increments of $\frac{1}{20}$, from 0 up to a maximum of 30. The value of r_s is fixed at $r_s = 100$ and the coordinate system is chosen such that the point $(1, 0, 0)$ has a φ component of zero and the point $(0, 1, 0)$ has a φ component of $\frac{\pi}{2}$.

The samples of ψ can be organized into a table that allows for fast lookup. To estimate $\mu(x_0, x'_0, r_s, t)$ we transform $(x_0, x'_0, r_s, t) \to (r_0, \alpha, t_s)$. Then we linearly interpolate over all three parameters using the closest known values of ψ. This requires a total of eight lookups and gives us an estimate for

```
function SAMPLE(S_r, S_α, S_t)
    r_s := 100
    for all r_0 ∈ S_r do
        for all α ∈ S_α do
            for all t_s ∈ S_t do
                x_0 := (r_0 · r_s, 0, 0)
                x'_0 := (cos α, sin α, 0)
                t := t_s · r_s
                (x_t, x'_t) := μ(x_0, x'_0, r_s, t)
                r_t := ||x_t|| / r_s
                β := arctan(x_t^1 / x_t^0)
                γ := arctan(x_t^1' / x_t^0')
                Record (r_0, α, t_s, r_t, β, γ) for later use
            end for
        end for
    end for
end function
```

(r_t, β, γ) which is transformed into (x_t, x'_t). Sampling took about 15 seconds. The total size of the model on disk is 60 MB.

3.4. Results

The model was tested by fixing a value of $r_s = 100$ and randomly selecting $1,000,000$ starting positions, velocities, and travel distances. For each randomly chosen $(x_0,\ x'_0,\ t)$, $\mu(x_0, x'_0, r_s, t)$ was computed in three different ways. First, μ was computed using a very small step size of $\frac{1}{2000}r_s = 0.05$. This result was considered the ground truth. Next, μ was computed using a step size of $\Delta t = \frac{1}{10}r_s = 10$. This larger step size would be feasible to use to render images. Last, μ was computed using the model.

method	mean error/distance	max error/distance	total time
$\Delta t = 10$	2.0×10^{-4}	5.9×10^{-3}	4857ms
model	3.3×10^{-4}	8.4×10^{-3}	327ms

On average, using the model produced an error 1.6 times larger but was also 14.8 times faster than the ray marching method. Interestingly, the largest error value for the model was smaller than the largest error value for the ray marching method. The model took an average of 327 ns/query, while the numerical method took an average of 66 ns/step. This means that using the model is faster than ray marching if the travel distance is more than about $5\Delta t$.

4. Improved ray-object intersection tests

4.1. Using our model to aid with intersection tests

The ability to estimate μ in constant time can be used to reduce the work necessary to detect collisions between curved rays and objects in the scene. Consider a point x_0 and an object Ω. We define the distance ℓ_{min} as the length of the shortest line segment with one endpoint at x_0 and another endpoint on the surface of Ω. ℓ_{min} represents the minimum distance that a ray of light must travel from x_0 before possibly intersecting Ω. Therefore, when determining if a light ray with initial conditions (x_0, x_0') intersects Ω we can safely advance the ray by a timestep of $t = \ell_{min}$ without having to perform any intersection tests. Using our model we compute $(x_{\ell_{min}}, x'_{\ell_{min}}) = \mu(x_0, x_0', r_s, \ell_{min})$. If ℓ_{min} is much larger than Δt then we have saved ourselves from having to perform many intersection tests and iterations of the STEP function.

After making the jump of distance ℓ_{min} we must continue to trace the ray. We can now simply consider $(x_{\ell_{min}}, x'_{\ell_{min}})$ to be the initial conditions of the ray. Setting $(x_0, x_0') = (x_{\ell_{min}}, x'_{\ell_{min}})$, we re-compute ℓ_{min} and make another jump. This process can be iterated until the ray leaves the gravitational sphere of influence.

4.2. Finding intersections

If the ray does not intersect the object or the event horizon, repeating the jumping process will suffice to fully trace the ray.

If the ray does intersect the object, repeatedly jumping will not yield good results. The jump distance will become smaller and smaller as the ray approaches the object and the intersection may take a large number of iterations to be detected, or the intersection may not ever be detected at all. This problem is solved by fixing a minimum jump distance of ϵ_ℓ. In the case that $\ell_{min} < \epsilon_\ell$ we do not use the model. Instead, we must advance the ray using the STEP function while performing intersection tests. This suffices to detect intersections between rays and objects.

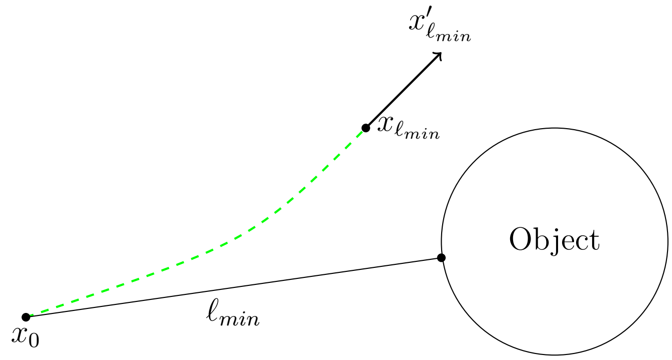

Figure 5: The minimum distance between the object (in this case a sphere) and x_0 is ℓ_{min}. The ray is advanced along its trajectory by a distance of ℓ_{min} by querying the model. If $\ell_{min} >> \Delta t$ then this is significantly faster than iterating the STEP function.

A similar problem occurs for rays that fall through the event horizon. When ψ is sampled rays are not traced past the event horizon so using the model to compute μ will never allow us to detect if the ray has crossed the event horizon. Additionally, the path of a light ray very near the event horizon can be significantly curved, leading to unacceptably large errors when interpolating. We fix a factor ϵ_r. If the distance between x_0 and the origin is less than $\epsilon_r \cdot r_s$ then we use the STEP function rather than querying the model. In the provided code, $\epsilon_r = 1.05$ and $\epsilon_\ell = \Delta t$.

4.3. Results

The performance gain from using the model and improved intersection testing vs the ray marching method depends on the type of scene being rendered. Scenes with few, small objects allow for the best performance, with the accelerated method giving speedups of more than a factor of 10. Objects that are large, such as an accretion disk, impact the performance of our accelerated method more than the performance of the ray marching method. But even in these cases, the accelerated method still outperforms ray marching by a factor of around 5.

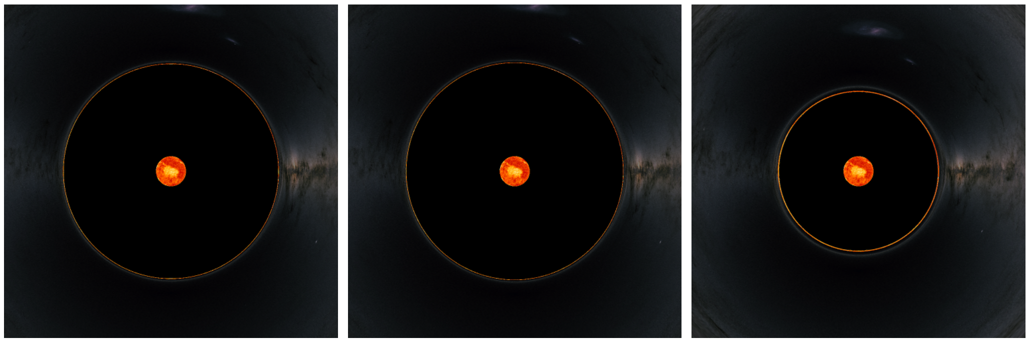

Figure 6: Three 500x500 images rendered with 4 rays/pixel. These images depict a star directly in front of a black hole. The black hole has Schwarzschild radius of 5.0. The star has a radius of 1.5 and is a distance $2r_s$ from the origin. The circle of light is a result of light rays curving all the way around the black hole and then intersecting with the star. The leftmost image was rendered in 19.9 seconds using ray marching. The middle image was rendered in 0.96 seconds using the model. Both the left and middle images used an adaptive step size with a minimum step of 0.05. The rightmost image was rendered in 1.08 seconds using ray marching with a minimum step of 1.5. In this image the black hole appears noticeably smaller due to error caused by the large step size.

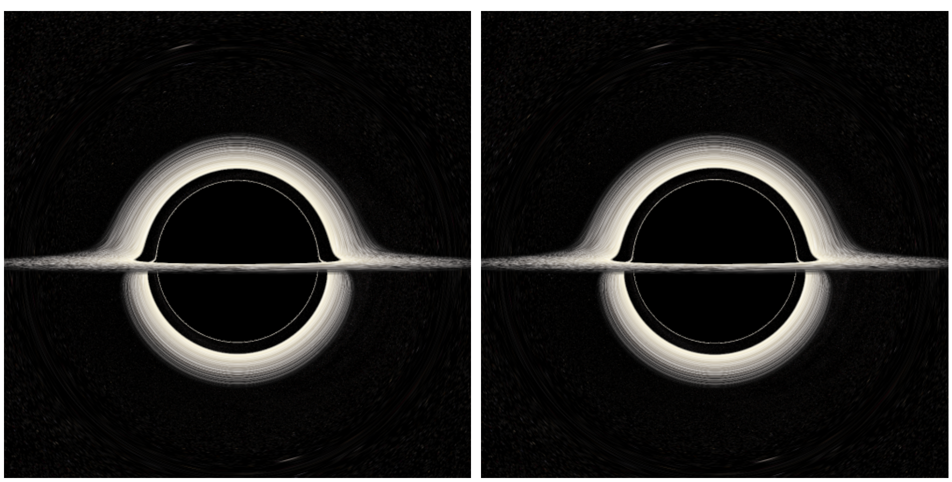

Figure 7: Two 500x500 images rendered with 4 rays/pixel. These images depict a black hole surrounded by an accretion disk. The black hole has Schwarzschild radius of 5.0. The left image was rendered in 16.9 seconds using ray marching. The right image was rendered in 2.37 seconds using the model. For both images, an adaptive step size with a minimum step of 0.05 was used.

5. Deriving the radial acceleration of a light ray

5.1. Finding Schwarzschild geodesics

The paths of particles through Schwarzschild spacetime can be found using the geodesic equation. It is useful to re-parameterize the geodesic equation in terms of coordinate time $x^0 = t$. This can be done using the chain rule. The geodesic equation becomes

$$\frac{d^2x^\lambda}{dt^2} = -\Gamma^\lambda{}_{\gamma\rho}\frac{dx^\gamma}{dt}\frac{dx^\rho}{dt} + \Gamma^0{}_{\gamma\rho}\frac{dx^\gamma}{dt}\frac{dx^\rho}{dt}\frac{dx^\lambda}{dt}$$

This equation gives us the coordinate acceleration of a particle according to an observer far away from the gravitating mass. For the purpose of raytracing we only need to find $\frac{d^2r}{dt^2} = \frac{d^2x^1}{dt^2}$ and $\frac{d^2\varphi}{dt^2} = \frac{d^2x^3}{dt^2}$. The Christoffel symbols $\Gamma^\lambda{}_{\gamma\rho}$ are calculated from the metric.

$$\Gamma^\lambda{}_{\gamma\rho} = \frac{1}{2}g^{\beta\lambda}\left(\partial_\rho g_{\gamma\beta} + \partial_\gamma g_{\beta\rho} - \partial_\beta g_{\rho\gamma}\right)$$

where

$$g_{\mu\nu} = \begin{bmatrix} -\left(1-\frac{r_s}{r}\right)c^2 & 0 & 0 & 0 \\ 0 & \left(1-\frac{r_s}{r}\right)^{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r^2 \end{bmatrix}$$

$$g^{\mu\nu} = \begin{bmatrix} -\left(1-\frac{r_s}{r}\right)^{-1}c^{-2} & 0 & 0 & 0 \\ 0 & \left(1-\frac{r_s}{r}\right) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r^{-2} \end{bmatrix}$$

Here we have used the simplified metric where $\theta = \frac{\pi}{2}$. The relevant Christoffel symbols are:

$$\Gamma^0{}_{\gamma\rho} = \begin{bmatrix} 0 & \frac{r_s}{2r^2}\left(1-\frac{r_s}{r}\right)^{-1} & 0 & 0 \\ \frac{r_s}{2r^2}\left(1-\frac{r_s}{r}\right)^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\Gamma^1{}_{\gamma\rho} = \begin{bmatrix} \frac{c^2 r_s}{2r^2}\left(1 - \frac{r_s}{r}\right) & 0 & 0 & 0 \\ 0 & -\frac{r_s}{2r^2}\left(1 - \frac{r_s}{r}\right)^{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_s - r \end{bmatrix}$$

$$\Gamma^3{}_{\gamma\rho} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r^{-1} \\ 0 & 0 & 0 & 0 \\ 0 & r^{-1} & 0 & 0 \end{bmatrix}$$

Plugging the Christoffel symbols into the geodesic equation results in

$$\frac{d^2 r}{dt^2} = -\frac{c^2 r_s}{2r^2}\left(1 - \frac{r_s}{r}\right) + \frac{3r_s}{2r^2}\left(1 - \frac{r_s}{r}\right)^{-1}\left(\frac{dr}{dt}\right)^2 + (r - r_s)\left(\frac{d\varphi}{dt}\right)^2$$

$$\frac{d^2\varphi}{dt^2} = -\frac{2}{r}\frac{d\varphi}{dt}\frac{dr}{dt} + \frac{r_s}{r^2}\left(1 - \frac{r_s}{r}\right)^{-1}\frac{d\varphi}{dt}\frac{dr}{dt}$$

or, using dot notation:

$$\ddot{r} = -\frac{c^2 r_s}{2r^2}\left(1 - \frac{r_s}{r}\right) + \frac{3r_s}{2r^2}\left(1 - \frac{r_s}{r}\right)^{-1}\dot{r}^2 + (r - r_s)\dot{\varphi}^2$$

$$\ddot{\varphi} = -\frac{2}{r}\dot{r}\dot{\varphi} + \frac{r_s}{r^2}\left(1 - \frac{r_s}{r}\right)^{-1}\dot{r}\dot{\varphi}$$

5.2. Challenges with tracing Schwarzschild geodesics

The path of a light ray can be numerically computed by using the above formulas for $\ddot{r}$ and $\ddot{\varphi}$. However, the above formulas are problematic because they are parameterized according to time experienced by an outside observer. Due to gravitational time dilation the speed of a photon through Schwarzschild spacetime will appear slower the closer the photon is to the gravitating mass. Advancing the light ray by a timestep of Δt will move the lightray by a variable distance that depends on the light ray's location and direction of motion. This is a problem because for the purpose of ray tracing it is assumed that the speed of the light ray is a constant, chosen to be 1. This assumption greatly simplifies the ray tracing code as it makes distance and time equivalent to each other. Unfortunately, numerically integrating the geodesic equation will produce a curve with a speed that varies over time.

An additional problem is that, to an outside observer, any particle that falls past the event horizon will appear to take an infinite amount of time to do so. If we directly integrate the geodesic equation then we will never be able to detect if the ray falls past the event horizon. The ray may approach the event horizon very closely but never cross it.

5.3. Re-parameterizing the null geodesics

These problems can be fixed by re-parameterizing the light ray. Instead of using coordinate time t as the parameter we instead use coordinate distance ℓ. This will produce a curve with the exact same shape as the original geodesic but with a constant speed of 1 everywhere. To numerically compute this curve we use the chain rule for second derivatives.

$$\begin{aligned} \frac{d^2r}{d\ell^2} &= \frac{d^2r}{dt^2}\Big(\frac{dt}{d\ell}\Big)^2 + \frac{dr}{dt}\frac{d^2t}{d\ell^2} \\ &= \ddot{r}\Big(\frac{dt}{d\ell}\Big)^2 + \dot{r}\frac{d^2t}{d\ell^2} \end{aligned}$$

To find $\frac{dt}{d\ell}$ and $\frac{d^2t}{d\ell^2}$ it is easier to first find $\dot{\ell} = \frac{d\ell}{dt}$ and $\ddot{\ell} = \frac{d^2\ell}{dt^2}$. The value $\dot{\ell}$ is the speed of the light ray according to a far away observer. It is determined by $\dot{r}$ and $\dot{\varphi}$, the radial and angular velocities of the light ray according to the observer.

$$\frac{d\ell}{dt} = \Big(\dot{r}^2 + r^2\dot{\varphi}^2\Big)^{1/2}$$

Deriving with respect to t yields

$$\frac{d^2\ell}{dt^2} = \frac{\dot{r}\ddot{r} + r\dot{r}\dot{\varphi}^2 + r^2\dot{\varphi}\ddot{\varphi}}{\Big(\dot{r}^2 + r^2\dot{\varphi}^2\Big)^{1/2}} = \frac{\dot{r}\ddot{r} + r\dot{r}\dot{\varphi}^2 + r^2\dot{\varphi}\ddot{\varphi}}{\dot{\ell}}$$

From this we find:

$$\Big(\frac{dt}{d\ell}\Big)^2 = \frac{1}{\dot{\ell}^2}$$

and, using the second derivative rule for inverse functions:

$$\frac{d^2t}{d\ell^2} = -\frac{\frac{d^2\ell}{dt^2}}{\Big(\frac{d\ell}{dt}\Big)^3} = -\frac{\dot{r}\ddot{r} + r\dot{r}\dot{\varphi}^2 + r^2\dot{\varphi}\ddot{\varphi}}{\dot{\ell}^4}$$

Therefore

$$\frac{d^2r}{d\ell^2} = \frac{\ddot{r}}{\dot{\ell}^2} - \frac{\dot{r}^2\ddot{r} + r\dot{r}^2\dot{\varphi}^2 + r^2\dot{r}\dot{\varphi}\ddot{\varphi}}{\dot{\ell}^4}$$

This is the total radial acceleration of the curve. We are interested only in the portion of radial acceleration resulting from the radially inward "force" due to gravity. To find this we must subtract the radial acceleration due to the light ray's own motion. The centripetal acceleration is given by

$$r\left(\frac{d\varphi}{d\ell}\right)^2 = \frac{r\dot{\varphi}^2}{\dot{\ell}^2}$$

The total "force" F due to gravity is

$$F = \frac{\ddot{r} - r\dot{\varphi}^2}{\dot{\ell}^2} - \frac{\dot{r}^2\ddot{r} + r\dot{r}^2\dot{\varphi}^2 + r^2\dot{r}\dot{\varphi}\ddot{\varphi}}{\dot{\ell}^4}$$

This is the F function that appears in STEP. $\ddot{r}, \ddot{\varphi}$ and $\dot{\ell}$ are determined by $r_s, r, \dot{r}$ and $\dot{\varphi}$. To find F we must know $\dot{r}$ and $\dot{\varphi}$. Recall that the direction of the light ray in its orbital plane at any time can be represented as an angle α. This angle determines $\dot{r}$ and $\dot{\varphi}$.

$$\dot{r} = \frac{dr}{dt} = \cos(\alpha)\left(1 - \frac{r_s}{r}\right)c$$

$$\dot{\varphi} = \frac{d\varphi}{dt} = \frac{\sin(\alpha)}{r}\left(1 - \frac{r_s}{r}\right)^{1/2}c$$

We verify that this is consistent with the metric.

$$-c^2d\tau^2 = -\left(1 - \frac{r_s}{r}\right)c^2dt^2 + \left(1 - \frac{r_s}{r}\right)^{-1}dr^2 + r^2d\varphi^2$$

Substituting

$$dr^2 = \cos^2(\alpha)\left(1 - \frac{r_s}{r}\right)^2c^2dt^2$$

and

$$d\varphi^2 = \frac{\sin^2(\alpha)}{r^2}\left(1 - \frac{r_s}{r}\right)c^2dt^2$$

the metric gives

$$-c^2d\tau^2 = -\left(1 - \frac{r_s}{r}\right)c^2dt^2 + \left(1 - \frac{r_s}{r}\right)^{-1}\left[\cos^2(\alpha)\left(1 - \frac{r_s}{r}\right)^2c^2dt^2\right]$$

$$+r^2\Big[\frac{\sin^2(\alpha)}{r^2}\Big(1-\frac{r_s}{r}\Big)c^2dt^2\Big]$$
$$= -\Big(1-\frac{r_s}{r}\Big)c^2dt^2 + \Big(\cos^2(\alpha)+\sin^2(\alpha)\Big)\Big[\Big(1-\frac{r_s}{r}\Big)c^2dt^2\Big]$$
$$= -\Big(1-\frac{r_s}{r}\Big)c^2dt^2 + \Big(1-\frac{r_s}{r}\Big)c^2dt^2$$
$$= 0$$

Consistent with a null geodesic. Plugging the definitions for $\dot{r}$ and $\dot{\varphi}$ into the formula for F and then simplifying gives

$$F = \frac{-3r_s}{2\Big((r-r_s)\cot^2(\alpha)+r\Big)^2}$$

We can also verify that the curve has a velocity with a constant magnitude of 1. The magnitude of the velocity $||v||$ of the curve is given by:

$$||v||^2 = \Big(\frac{dr}{d\ell}\Big)^2 + r^2\Big(\frac{d\varphi}{d\ell}\Big)^2$$
$$= \Big(\frac{dr}{dt}\frac{dt}{d\ell}\Big)^2 + r^2\Big(\frac{d\varphi}{dt}\frac{dt}{d\ell}\Big)^2$$
$$= \frac{\dot{r}^2}{\dot{\ell}^2} + \frac{r^2\dot{\varphi}^2}{\dot{\ell}^2}$$
$$= \frac{\dot{r}^2 + r^2\dot{\varphi}^2}{\dot{\ell}^2}$$
$$= \frac{\dot{\ell}^2}{\dot{\ell}^2} = 1$$