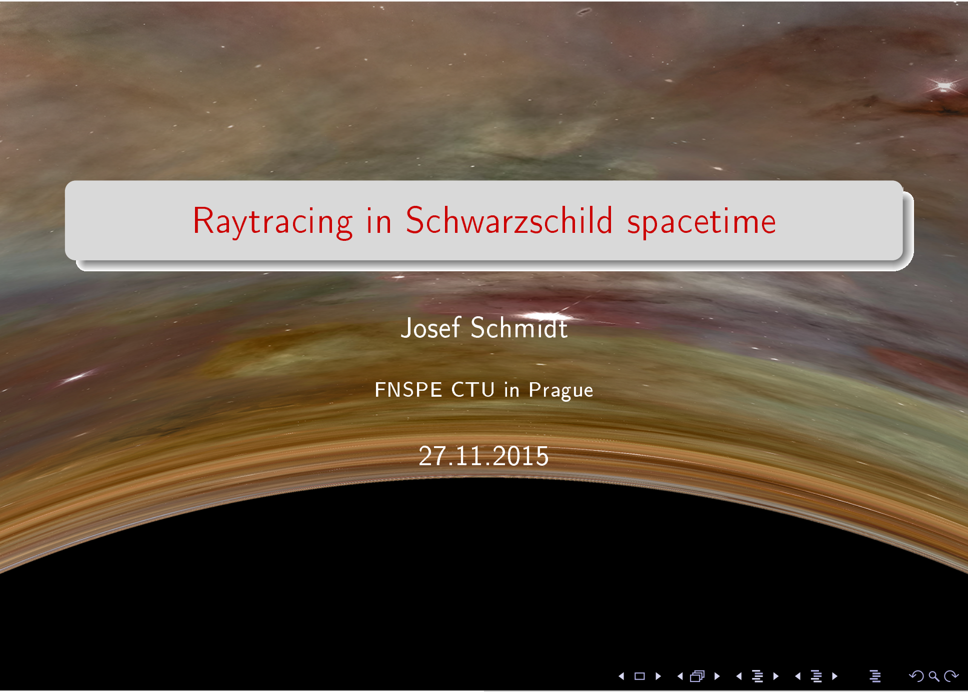
Raytracing in Schwarzschild spacetime

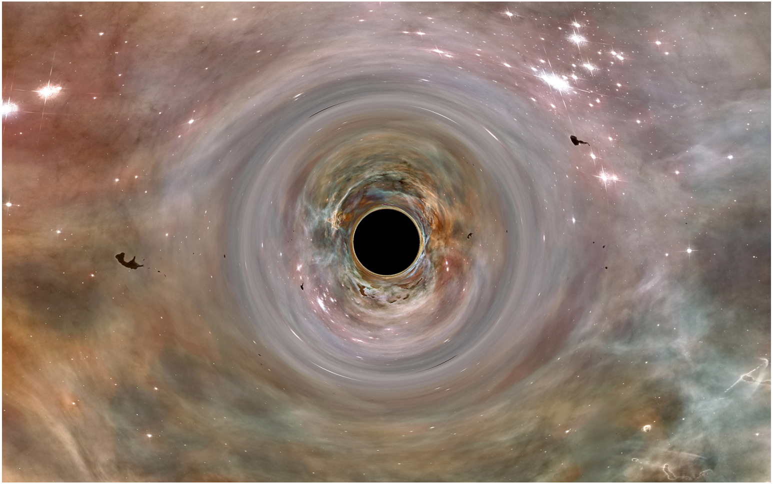
Josef Schmidt

FNSPE CTU in Prague

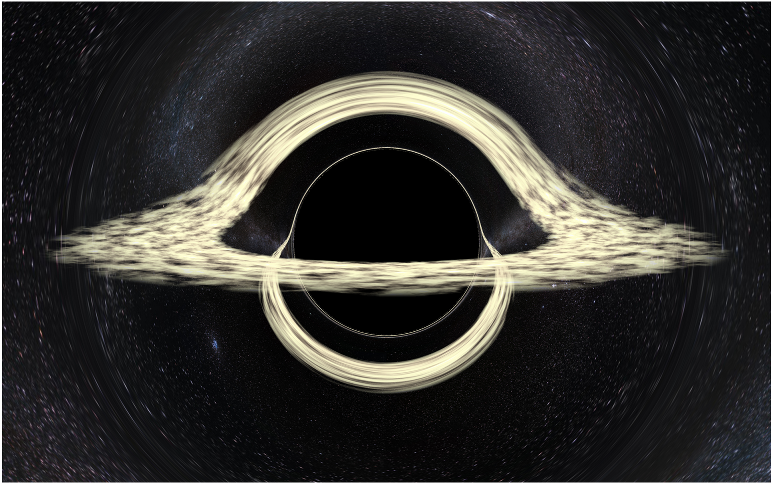
27.11.2015



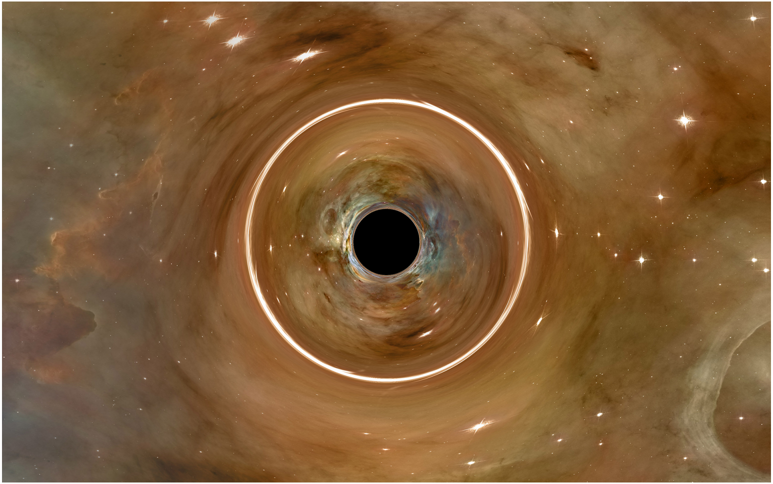
Showcase – Schwarzschild black hole



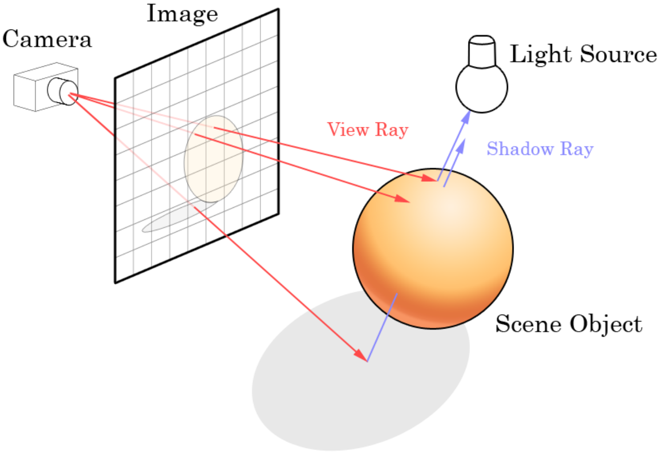
Showcase – accretion disc



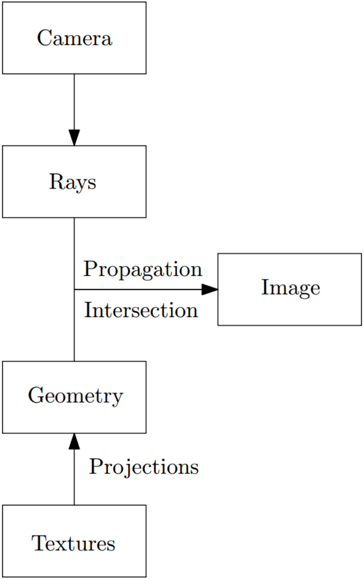
Showcase – Einstein's ring



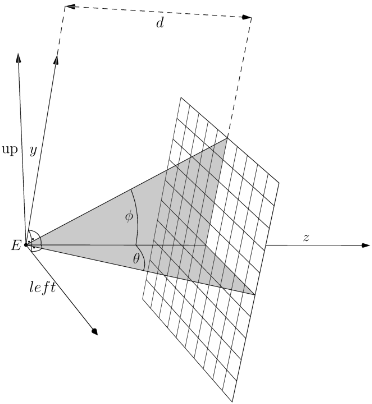
Raytracing method



Raytracing method



- Camera generates rays for each pixel of image.
- Rays are propagated through spacetime.
- Intersections with object in the scene (spacetime) are calculated.
- Resulting pixel color is obtained by projecting textures on objects.

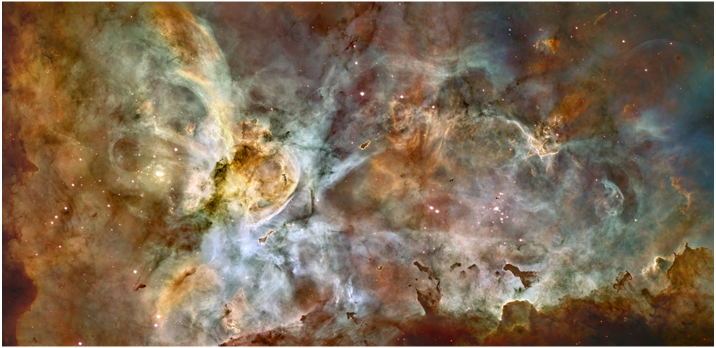


- Camera parameters: image resolution (width x height), size of projection plane and distance from focal point
- Camera located at point P with coordinates $(r_P, \theta_P, \varphi_P)$
- Lorentz tetrad $(e_i^\mu)_{i=0}^3$ at P
  - i.e. $g_{\mu\nu} e_i^\mu e_j^\nu = \eta_{ij}$
  - e.g. $(e^\mu_{(t)}, e^\mu_{(r)}, e^\mu_{(\theta)}, e^\mu_{(\varphi)})$
- Tetrad can be rotated with $R \in SO(3)$ (rotating camera) or boosted (camera moving with velocity $\vec{V}$)
- Direction $\vec{n} = (n_x, n_y, n_z)$ leading to null vector $u^\mu = \alpha e^\mu_{(t)} + n_x e^\mu_{(r)} + n_y e^\mu_{(\theta)} + n_z e^{(\mu)}_{(\varphi)}$, α determined by $g_{\mu\nu} u^\mu u^\nu = 0$

- Sphere of infinite radius – representing stars very far from black hole
- Plane intersecting singularity – representing plane of accretion disc around black hole
- Not yet implemented:
  - "Centered" cylinder – simulating non-zero width of accretion disc
  - "Centered" spheres with finite radius – e.g. surface of neutron star
  - General surface $f(\vec{r}) = 0$, or even non-static $f(\vec{r}, t) = 0$

Textures

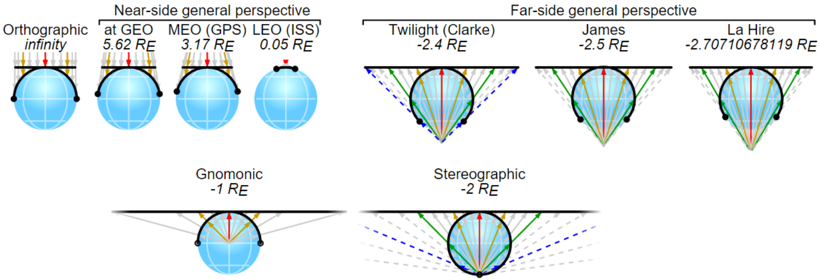
- Image files
- Textures from Hubble Space Telescope – hubblesite.org
  - e.g. Carina nebula
  - Size 29566x14321px $\approx$ 432Mpx (1211 MB RAM)



- Procedural textures: pixel color determined by function $f(u, v)$

Projections

- 2D textures has to be projected on objects in the scene
- Examples:
  - Plane: affine transformation
  - Sphere: azimuthal or cylindrical projections; practically: celestial sphere is covered by cylindrically projected texture around equator and two azimuthally projected textures on poles



Ray propagation

- Schwarzschild metric in equatorial plane ($\theta = \pi/2$)

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

- Let's denote $x^\mu(\lambda) = (t(\lambda), r(\lambda), \theta(\lambda) = \frac{\pi}{2}, \phi(\lambda))$
- Denoting $\frac{dx^\mu}{d\lambda} = \dot{x}^\mu = u^\mu$ we get the normalization condition

$$g_{\mu\nu} u^\mu u^\nu = 0 = -\left(1 - \frac{r_S}{r}\right) \dot{t}^2 - \frac{1}{1 - \frac{r_S}{r}} \dot{r}^2 - r^2 \dot{\phi}^2.$$

Killing vectors and constants of motion

- If ξ^μ is Killing vector, the quantity $\xi^\mu u_\mu$ is a constant of geodesic motion.
- For Killing vector ∂_t and ∂_ϕ we get the following expressions

$$u_t = g_{tt} u^t = -\left(1 - \frac{r_S}{r}\right) \dot{t} \equiv -E, \qquad u_\phi = g_{\phi\phi} u^\phi = r^2 \dot{\phi} \equiv L$$

- Substituting back into normalization condition we get the radial equation

$$\dot{r}^2 = E^2 - \left(1 - \frac{r_S}{r}\right) \frac{L^2}{r^2}$$

- Reparametrizing $\frac{dr}{d\lambda} = \frac{dr}{dt}\frac{dt}{d\lambda} = \frac{dr}{dt}\frac{E}{1-\frac{r_S}{r}}$ and introducing dimensionless variables:
    - inverse radial coordinate: $\zeta = \frac{r_S}{r}$,
    - impact parameter: $l = \frac{L}{Er_S}$,
    - dimeonsionless parametrization: $\sigma = \frac{E\lambda}{r_S}$;

  we get

$$\zeta' = \pm\zeta^2\sqrt{1-(1-\zeta)l^2\zeta^2} \quad \text{and} \quad \phi' = l\zeta^2,$$

  where prime denotes differentiation w.r.t. σ
- Combining the above equations we obtain first order differential equation for function $\phi(\zeta)$

$$\frac{d\zeta}{d\phi} = \pm\sqrt{q^2 - \zeta^2(1-\zeta)},$$

where inverse impact parameter $q = 1/l$ has been introduced.

Equation for $\phi(\zeta)$

$$\frac{d\zeta}{d\phi} = \pm\sqrt{q^2 - \zeta^2(1-\zeta)} = \pm\sqrt{P(\zeta)}$$

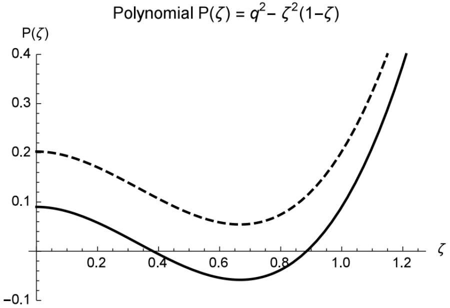
- Solutions are symmetric around radial turning points given by $P(\zeta) = 0$
- Solution can be written as

$$\boxed{\phi(\zeta_a, \zeta_b) = \int_{\zeta_a}^{\zeta_b} \frac{d\zeta}{\sqrt{q^2 - \zeta^2(1-\zeta)}},}$$

which can be expressed using incomplete elliptic integrals or simply evaluated numerically.

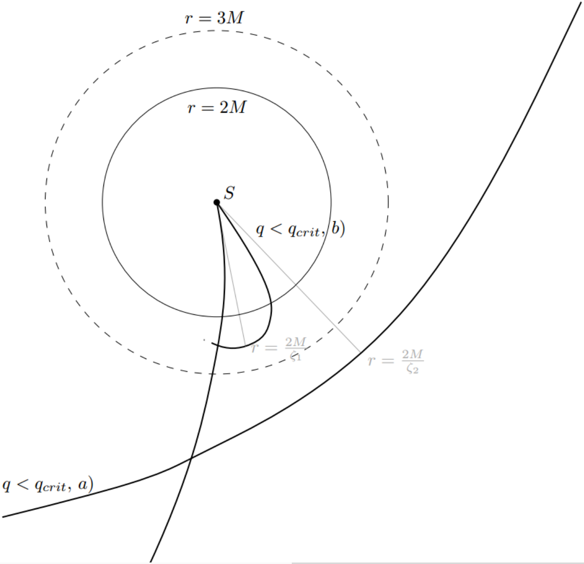
Cubic polynomial $P(\zeta) = q^2 - \zeta^2(1-\zeta)$

- Ray behaviour depends on properties of $P(\zeta)$ (and value of q).
- Minimum located at $\zeta_{min} = \frac{2}{3}$, i.e. at $r = \frac{3}{2} r_S = 3M$: photon sphere
- Depending on $q <=> q_{crit} = \frac{2}{3\sqrt{3}}$ we have 2, 1 or 0 roots for $\zeta > 0$.



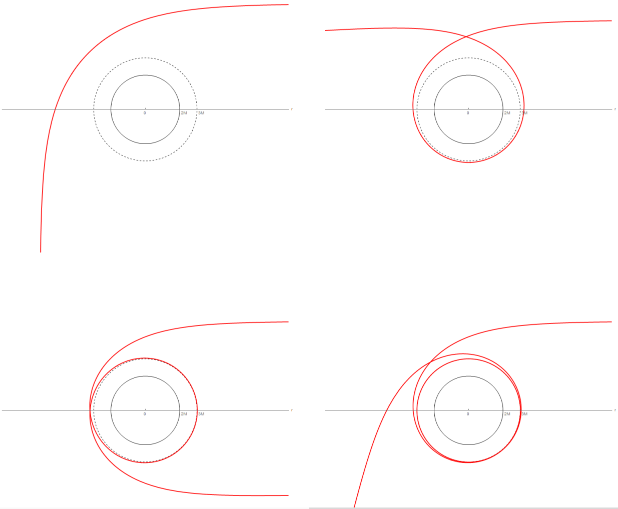
Types of rays

Different types of rays depending on $r <> 3M$, $q <> q_{crit}$ and $u^r <> 0$.



Bending of light rays

Bending of light rays for $Q = 1.85$, $Q = 6.75$, $Q = 9.85$ and $Q = 14.3$, where $q = q_{crit}(1 - e^{-Q})$.

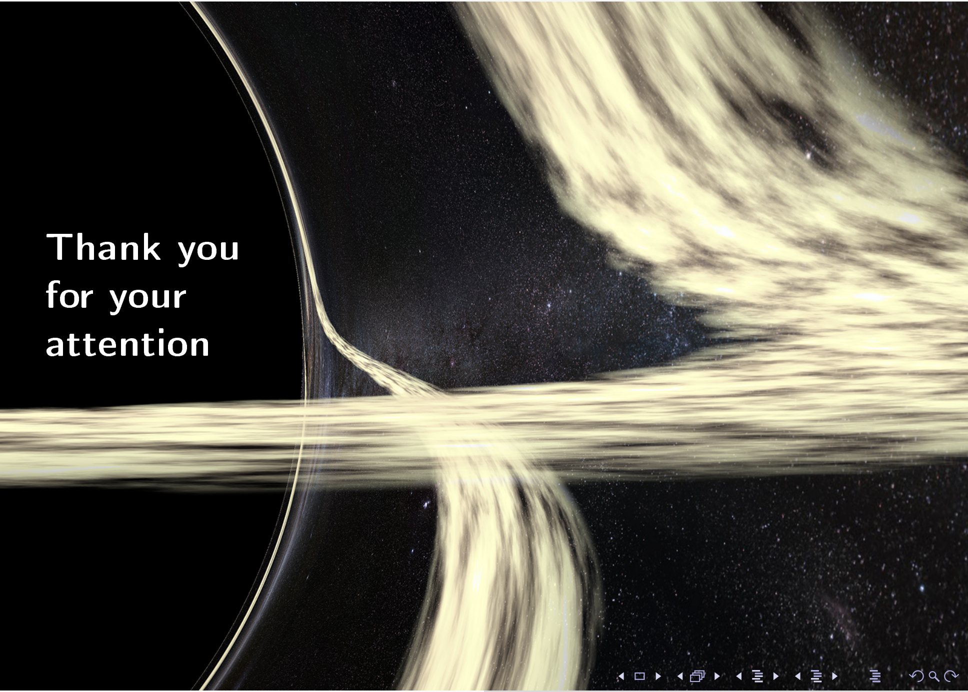


Technical info

- Implemented in C++ – 2700 lines of code
- Parallelized with OpenMP
- Performance on Intel Quad Core 3.3GHz – rendering time:
  - 4K resolution: $\approx$ 6 s
  - FullHD resolution: $\approx$ 1.5 s
  - (g++ compiler with -O3 flag)

What is next to be implemented?

- Doppler shift
- Brightness of images
- Subhorizon Lorentz camera tetrads
- Horizon crossing coordinates
- Retarded time, Shapiro delay
- Point stars
- "Full" raytracer (Minkowski space)
- More geometries
  - Reissner-Nordström spacetime (charged black hole)
  - Kerr spacetime (rotating black hole)
  - wormhole spacetime
- Postprocessing effects
- GPU acceleration


Thank you
for your
attention