CATEGORY: REAL-TIME GRAPHICS - RG02
POSTER P5271
CONTACT NAME Gerhard Zumbusch: gerhard.zumbusch@uni-jena.de
GPU TECHNOLOGY CONFERENCE

A Real-Time Ray Tracer for General Relativity

Gerhard Zumbusch
Friedrich-Schiller-Universität Jena

Abstract

Visualization of astrophysical data according to the theory of general relativity is discussed. Space is curved by energy and momentum, i.e. stars and black holes. Light rays take the shortest path (geodesic) through curved space-time. They look like they are bended like optical lenses bend light rays. Given some computed solutions to Einstein's equations of general relativity, an observer would see an image of pixels, each representing a curved light ray from the scene to the observer.
Computational ray tracing has to deal with the standard problem of ray-object intersection. Additionally the curved rays have to be computed as solutions to the geodesic differential equation. Some techniques to accelerate the ray tracing to real-time on GPUs are presented.

Curved Space-Time

Einstein's theory of general relativity describes several large scale phenomena by the notion of curved space-time. This includes classical Newtonian mechanics and the motion of stars and planets, but also less intuitive effects like black holes, light bending, red-shift of the spectra of distant stars and predictions like gravitational waves.

Geodesic Equation

Position and velocity (x, v) in 4-space with respect to an intrinsic time

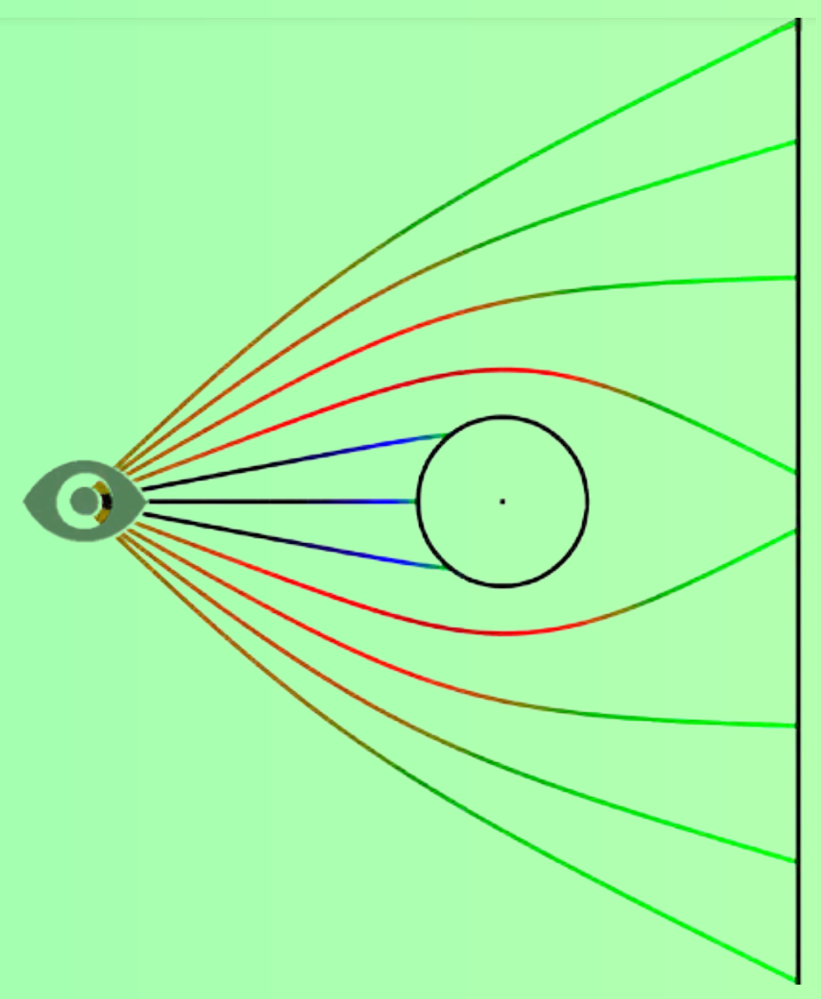
$$\begin{aligned} x^i(\tau)' &= v^i(\tau) \\ v^i(\tau)' &= \textstyle\sum_{jk} \Gamma^i_{jk}(x(\tau)) v^j(t) v^k(\tau) \end{aligned}$$

with Christoffel symbol $\Gamma^i_{jk} = \frac{1}{2}\sum_l g^{il}(d_j g_{kl} + d_k g_{jl} - d_l g_{jk})$, g^{ij} inverse metric to g_{ij} and the light ray condition

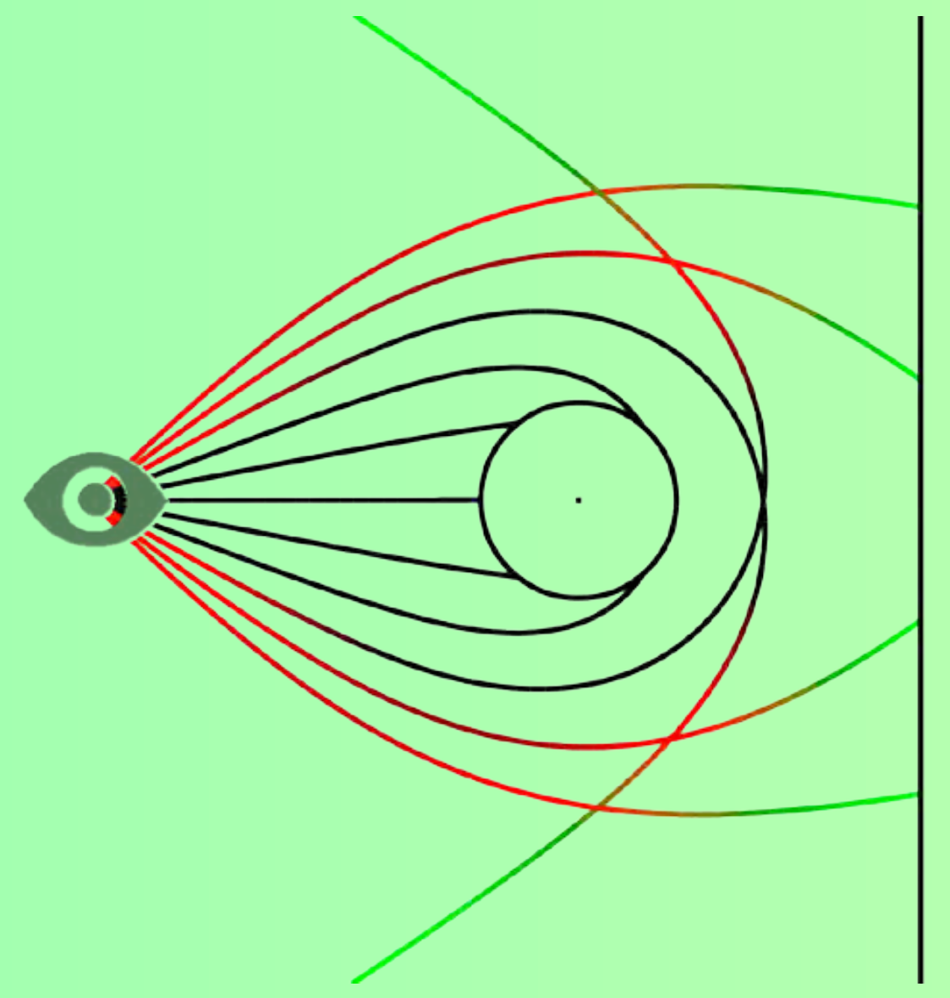
$$\sum_{jk} g_{jk}(x(\tau)) v^j(\tau) v^k(\tau) = 0$$

Phenomena

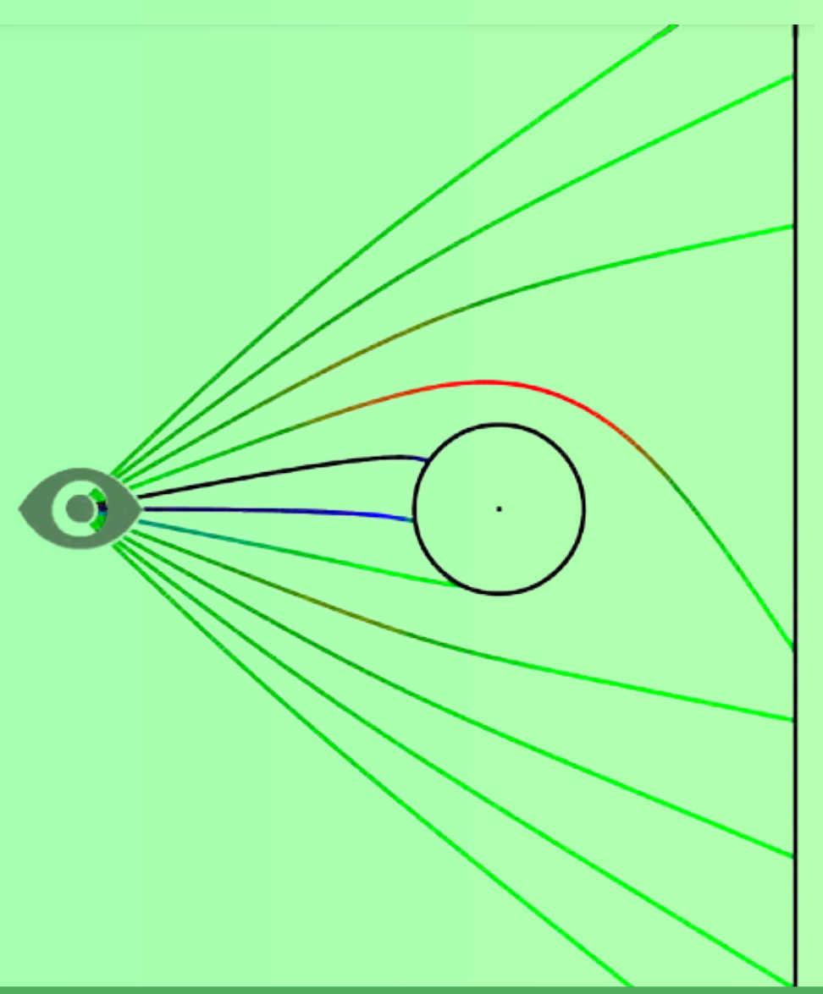
mass bends light



a black hole with event horizon

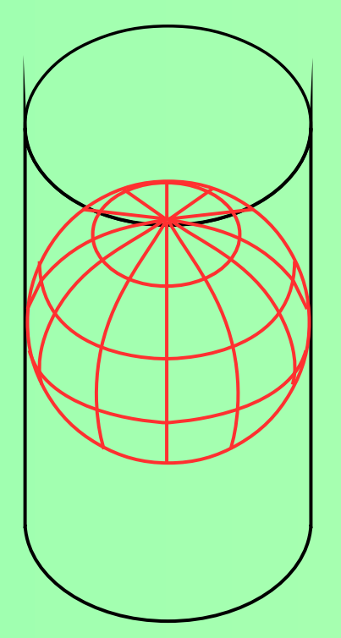


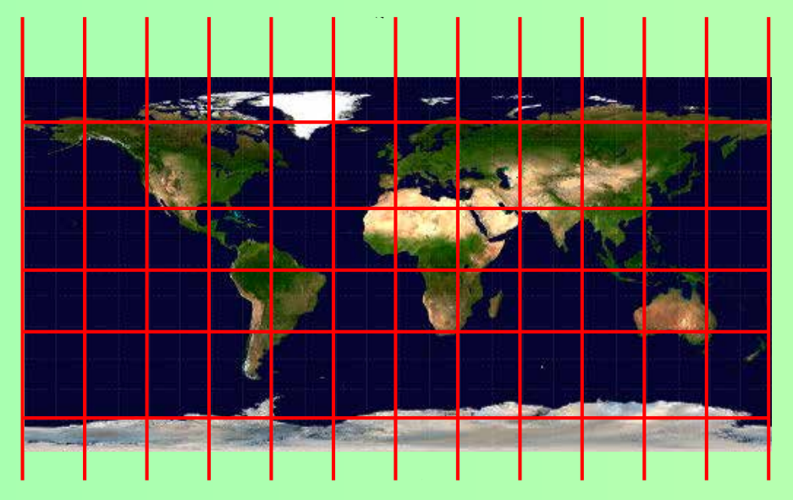
a rotating mass



Schematic View

Given a solution to Einstein's equation, we have a 4D space-time and a metric (distance function). How do visualize a 4D space properly?
Any projection to flat space is an approximation: See the different ways to chart the earth (conserve area, or distances, or shape,...)





Solve an Initial Value Problem

Use explicit time-stepping scheme. Low order scheme means that ray-object intersection can be computed exactly. Use adaptive time steps, small steps close to the horizon of the black holes, large steps far apart. Euler scheme, piece-wise linear solution, straight rays:

$$\begin{aligned} x(\tau_{k+1}) &= x(\tau_k) + \Delta t v(\tau_k) \\ v(\tau_{k+1}) &= v(\tau_k) + \Delta t \textstyle\sum_{jk} \Gamma^i_{jk}(x(\tau_k)) \\ &\quad \cdot v^j(\tau_k) v^k(\tau_k) \end{aligned}$$

Metric and Fast Derivatives

The metric $g_{ij}(x)$ is given analytically or numerically:

- Schwarzschild solution: static black hole, static star
- Kerr solution: rotating black hole
- various Neutron star models
- numeric field, output of some HPC codes

Avoid computation or storage of derivative terms $d_j g_{kl}$, use numeric differentiation:

$$d_j g_{kl} \approx \frac{1}{h}(g_{kl}(x + he_j) - g_{kl}(x))$$

Saves texture memory and time.

Fast Linear Algebra

Avoid inverse g^{ij}, solve linear equation system with $g_{ij} \in R^{4\times4\,\mathrm{sym}}$ instead: Symmetric (Cholesky) factorization $L \cdot D \cdot L^t$

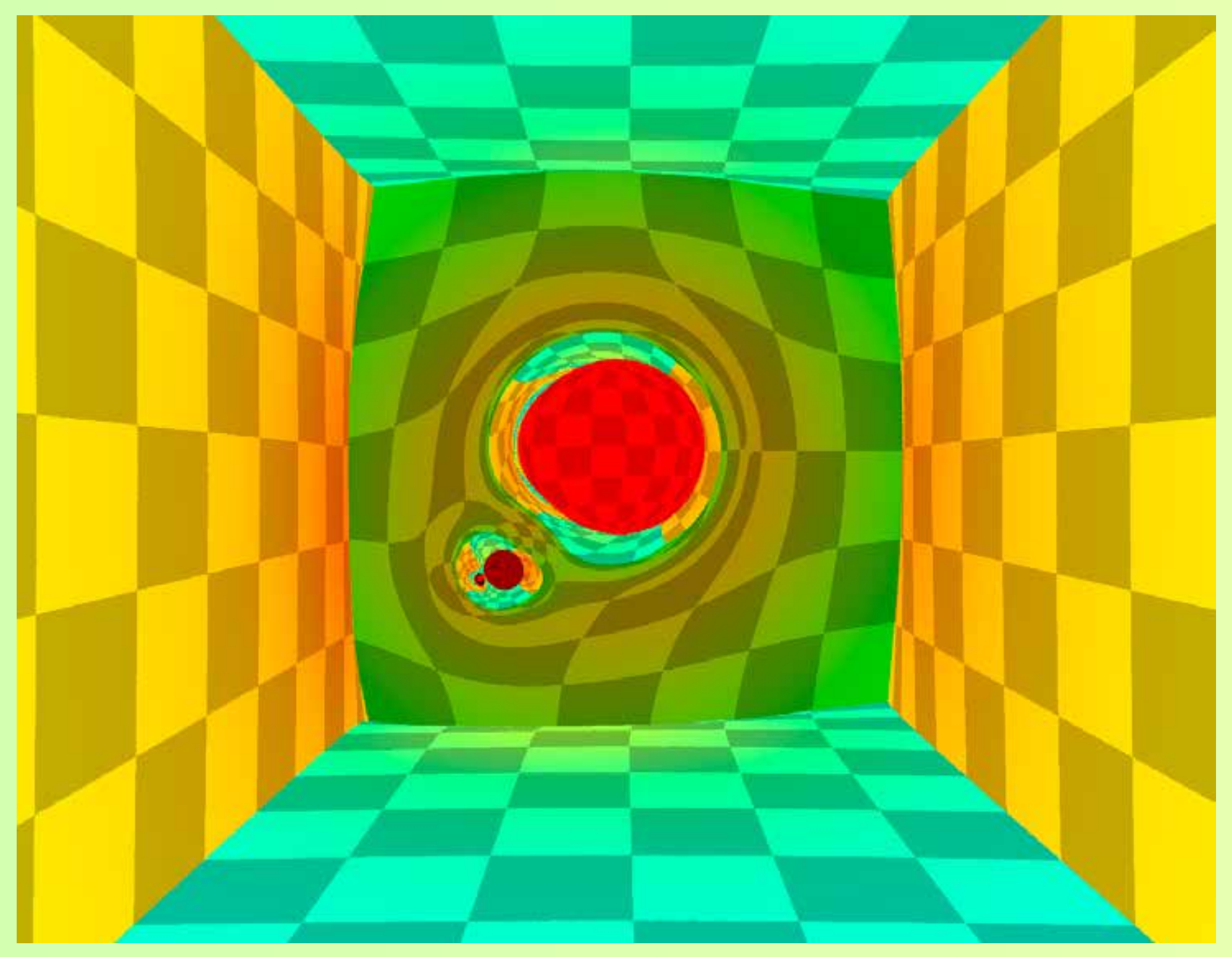
$$L = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & l_{32} & 1 & \\ l_{41} & l_{42} & l_{43} & 1 \end{pmatrix}$$

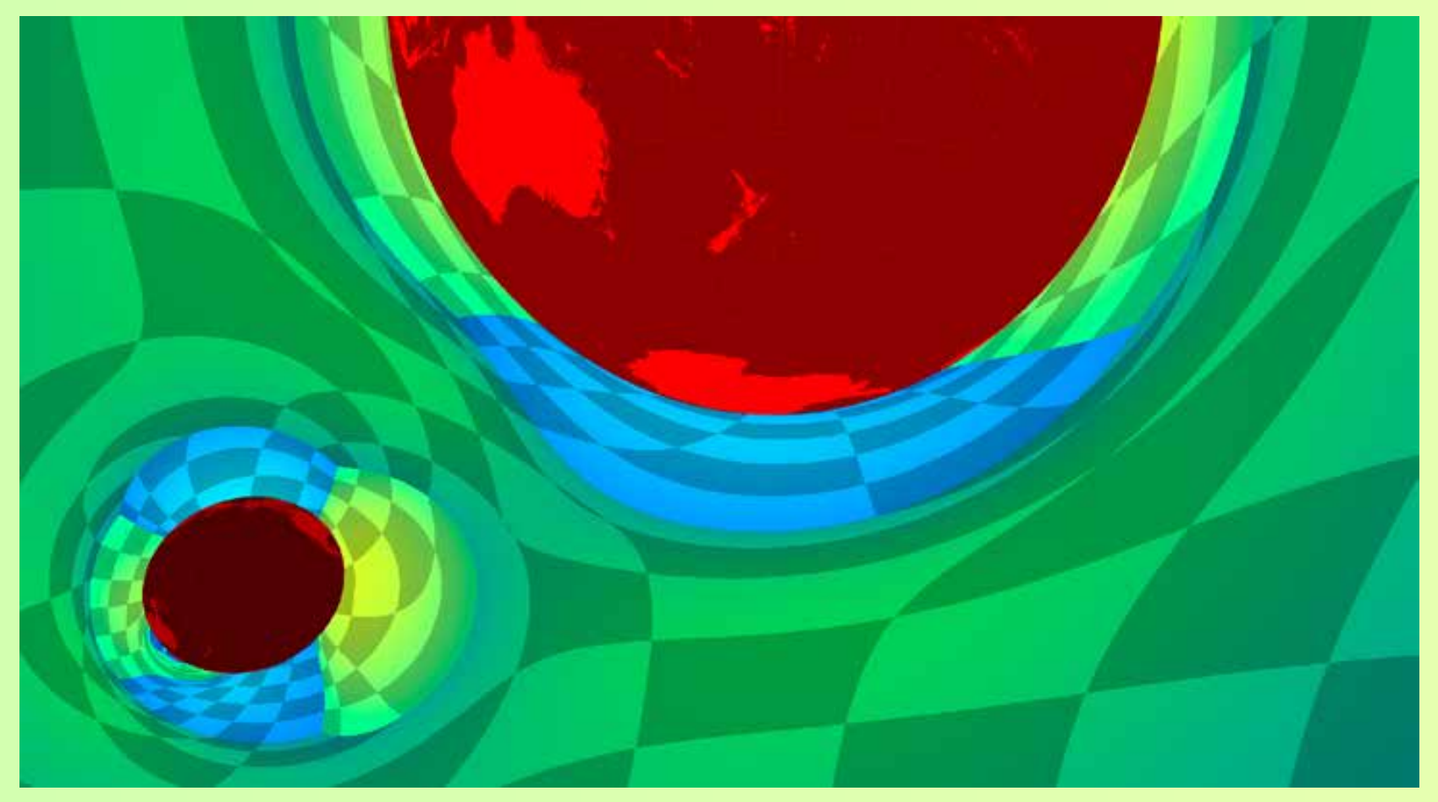
Avoid loops and arrays, manually unroll the code, perform computation in registers, store triangular matrix part only.

```
for (int j=0; j<4; j++) {
  float s = a[j][j];
  for (int k=0; k<j; k++) {
    float t = a[j][k];
    s -= a[k][k] * (t * t);
  }
  a[j][j] = s;
  for (int i=j+1; i<4; i++) {
    float s = a[i][j];
    for (int k=0; k<j; k++) {
      s -= a[k][k] * a[i][k] * a[j][k];
    }
    a[i][j] = s / a[j][j];
  }
}
```

Geometric Patterns

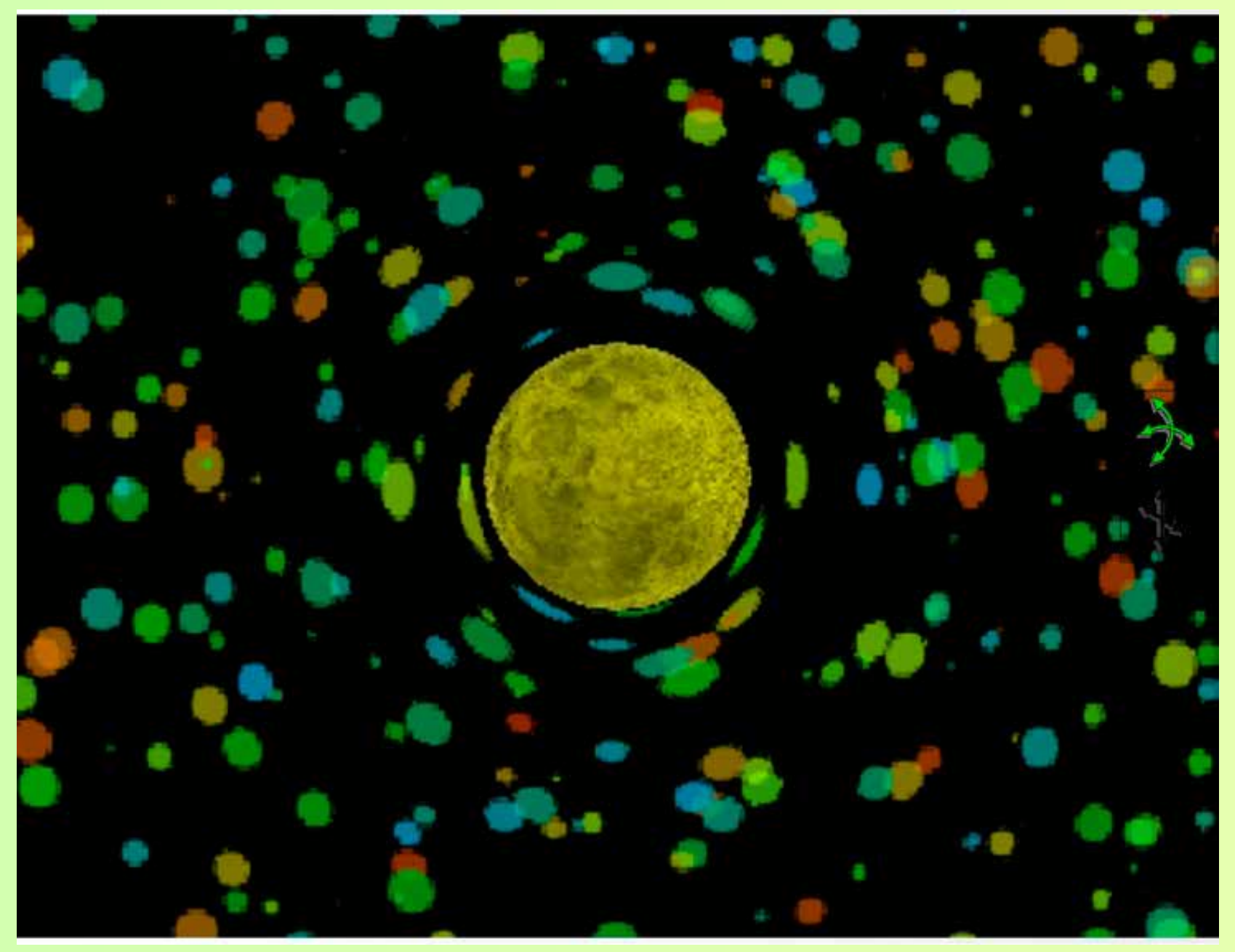
two rotating masses



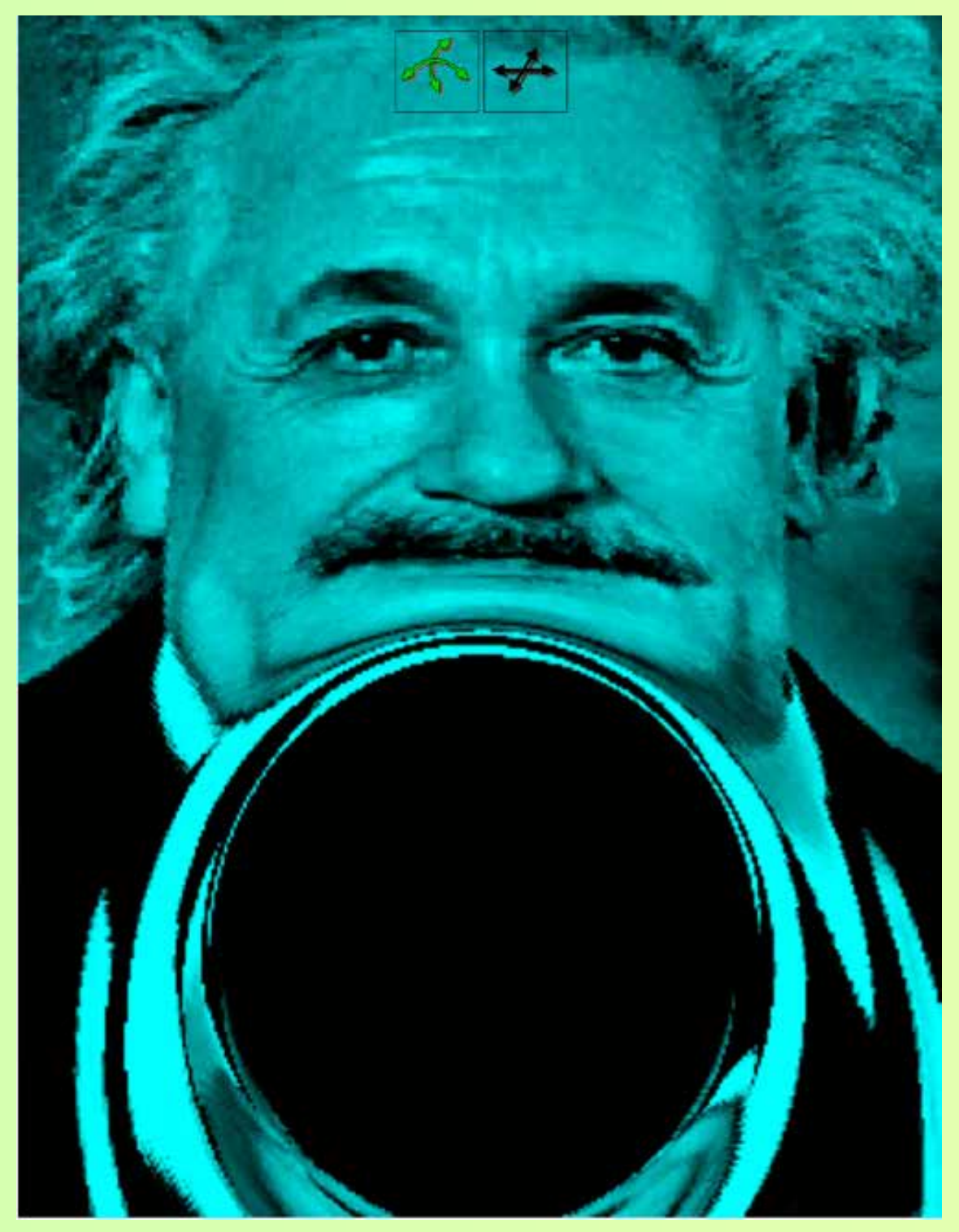


Textures

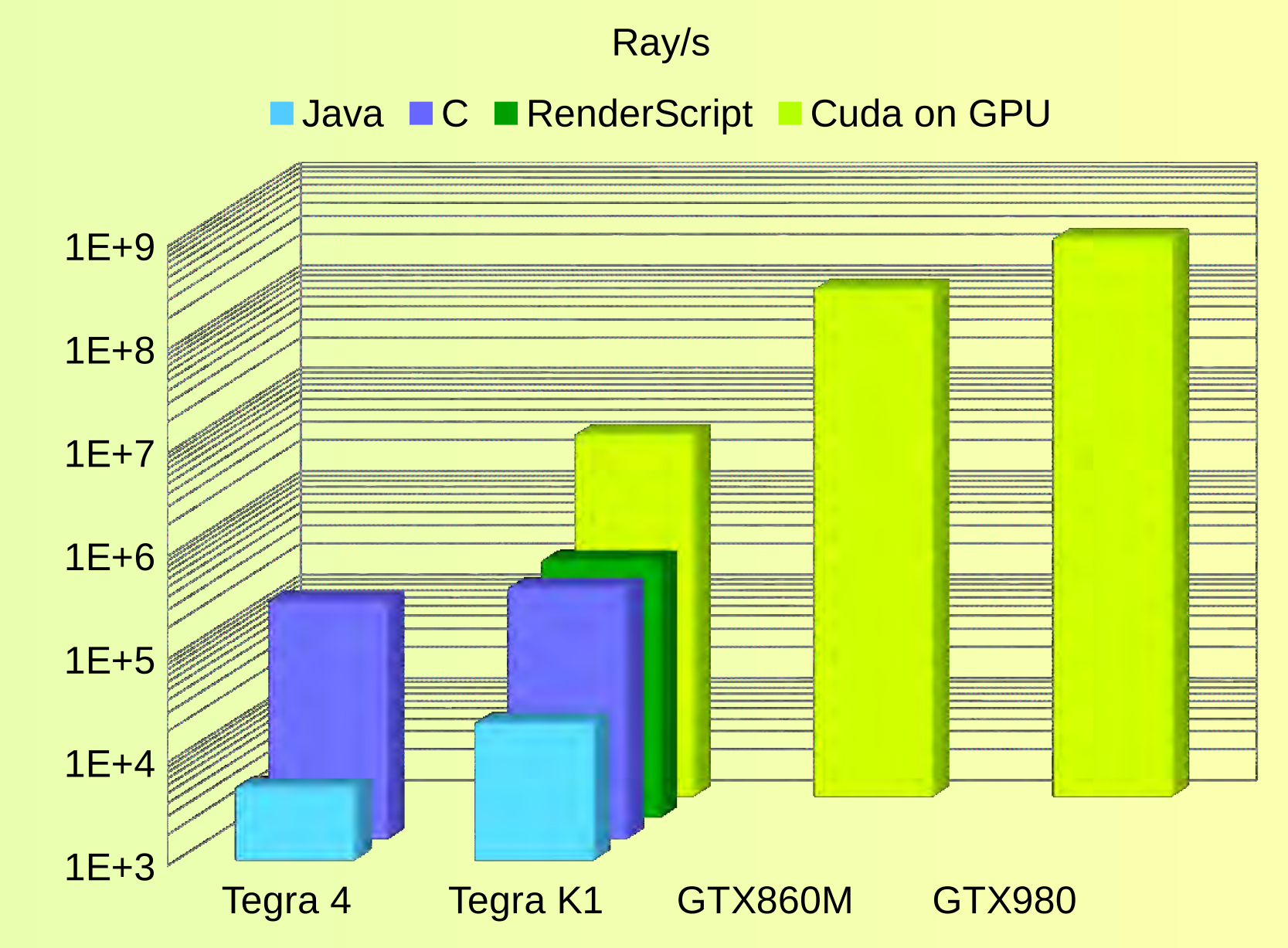
gravitational lensing



and a photo with gravitational lensing of a black hole



Performance



relativistic ray tracing, pixel/s

- Tablet (Tegra K1): 1280*720 @ 3 frames/s
- Laptop (GM107): 1920*1080 @ 40 frames/s
- PC (GM204): 3840*2160 @ 30 frames/s

Outlook

The ray tracer may serve as a scientific visualization tool for numerical solution to Einstein's equations, binary systems, Neutron stars, gravitational waves.

References

[1] A. Bohn, W. Throwe, F. Hebert, K. Henriksson, D. Bunandar, N. W. Taylor, and M. A. Scheel, *What would a binary black hole merger look like?* arXiv:1410.7775, Nov 2014. [gr-qc].
[2] R. Cowen, *Black-hole mergers cast kaleidoscope of shadows*, Nature News, (2014). doi:10.1038/nature.2014.16283.
[3] Google Play: GRRay.
[4] http://github.com/zumbusch/gr-ray.
[5] http://www.interstellar-movie.com.
[6] D. Kuchelmeister, T. Müller, M. Ament, G. Wunner, and D. Weiskopf, *GPU-based four-dimensional general-relativistic ray tracing*, Computer Physics Communications, 183 (2012), pp. 2282–2290.
[7] T. Müller, *Visualization in the theory of relativity*, PhD thesis, Eberhard-Karls-Universität Tübingen, 2006.
[8] H.-P. Nollert, U. Kraus, and H. Ruder, *Visualization in curved spacetimes: I. Visualization of objects via fourdimensional ray-tracing*, in Relativity and Scientific Computing, F. W. Hehl, R. A. Puntigam, and H. Ruder, eds., Springer, 1996, pp. 314–329.
[9] H.-P. Nollert, H. Ruder, H. Herold, and U. Kraus, *The relativistic "looks" of a neutron star*, Astronomy and Astrophysics, 208 (1989), p. 153.
[10] R. Richter, *Gravitational lensing behind black holes*, B.Sc. thesis, Friedrich-Schiller-Universität Jena, 2010.
[11] D. Weiskopf, *Visualization of Four-Dimensional Spacetimes*, PhD thesis, Eberhard-Karls-Universität Tübingen, 2001.
[12] A. Weyhausen, *Ray tracing in general relativity*, B.Sc. thesis, Friedrich-Schiller-Universität Jena, 2009.