JRelStarFlight – technical notes

Thomas Müller

formerly, Visualisierungsinstitut der Universität Stuttgart (VISUS),
Allmandring 19, 70569 Stuttgart, Germany

Abstract. In contrast to the usual graphical representation of the visual appearance of a star field in science fiction movies with its elongated stars, the actual view of such a star field as seen from within a warp bubble is completely different. To demonstrate what could be really seen, we have developed a Java application with which the user can travel through the stars of the Hipparcos catalogue either with relativistic speeds in the Minkowskian spacetime or with warp speed in the Alcubierre metric.

1. Graphical user interface

The graphical user interface of our relativistic flight simulator `JRelStarFlight` is shown in Fig. 1. When the application is launched, the observer is at rest in a Minkowskian spacetime and uses a full-sky panorama camera. To toggle between special relativistic and warp flight, use the drop-down menu in the lower left corner. There are two camera modes: the full-sky camera maps the whole sky from spherical coordinates onto a rectilinear like grid; the pinhole camera acts as a normal camera with 50° vertical field of view.

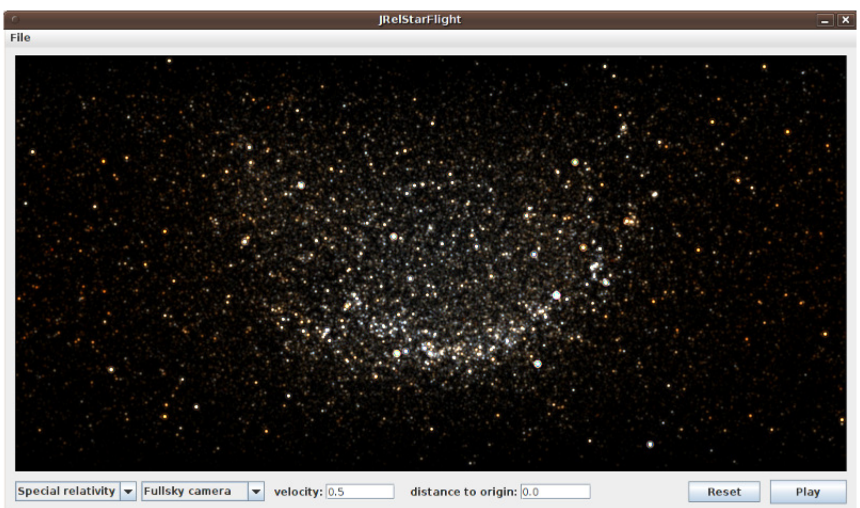

Figure 1. Screenshot of the Java application *JRelStarFlight*.

Moving the mouse in the OpenGL window with the left mouse button pressed, the viewing direction and, thus, the direction of motion can be changed. The current velocity $\beta = v/c$ can be modified by means of the input field ‘velocity’. For special relativistic flights, the velocity is restricted to $\beta = [-0.99c, 0.99c]$. The Warp metric only accepts velocities between $\beta = 0c$ and $\beta = 9c$. Toggling the ‘play’ button lets the observer move or stop. The current position can also be set with the input field ‘distance to origin’. Please note that when the observer has moved away from the origin, the mouse rotation acts as if the observer has moved in the newly selected direction ab initio. After editing the ‘velocity’ or the ‘distance’ field, please press *Enter*.

2. Technical details

The apparent position of a point-like star as seen within the warp bubble depends on how a light ray of this star is influenced by the curved spacetime. For the visual appearance, we also have to take the frequency shift and the lensing effect into account.

For the actual visualization of the stars we adopt the method by Müller and Weiskopf [1]. As database we use the Hipparcos star catalogue [2] and assign a Planck spectrum to each star with temperature T_{star} calculated by Reed's empirical law [3]. The frequency-shifted spectrum is again a Planck spectrum but at the different temperature $T_{\text{obs}} = T_{\text{star}}/(1+z_f)$. The resulting apparent visual magnitude of a star is determined by the magnification factor μ_{mag} and the integral over the Planck spectrum in the visual wavelength domain. A finite eye or telescope aperture yields a Fraunhofer diffraction pattern.

Since the star distances are in terms of light-years, we use years as unit of time. With 25 frames per second and a step size of $\Delta t = 0.4y$ per frame, one second of visualization time corresponds to 10 years of simulation time.

3. Implementation details

Before running the Java application we have to generate the two-dimensional lookup table that stores the observation angle ξ, the frequency shift z_f, and the magnification factor μ_{mag} for each asymptotic light direction φ and velocity v of the warp bubble. For that, we integrate the geodesic equation, the parallel transport of the Sachs basis vectors, and the Jacobian equations from the observer back in time until the geodesic reaches the sphere with radius $r_{\text{max}} = 5 \times 10^4$. Hence, we obtain a relation between observation angles ξ and intersection angles φ with the sphere, which must be inverted at the end of the calculations.

For our simulator, we use the warp parameters $R = 2, \sigma = 1$ and a lookup table with resolution 4096×248. While the angle φ is sampled linearly from 0 to π, the velocity in the row number 'row' is given by

$$v = 10^{(\text{row}+1)/248} - 1. \tag{1}$$

Thus, the minimum velocity $v_{\text{min}} \approx 0.0093$ is stored in row $= 0$ and the maximum velocity $v_{\text{max}} = 9$ in row $= 247$.

The rendering of the stars is realized using a Java implementation of the open graphics library OpenGL and the shading language GLSL [4]. In the vertex shader, we use the above explained lookup table to calculate the apparent positions and frequency-shifted Planck temperatures of the stars. In the fragment shader, the Planck temperatures are mapped to the precalculated color and Fraunhofer diffraction pattern.

References

[1] T. Müller and D. Weiskopf, "Distortion of the stellar sky by a Schwarzschild black hole," Am. J. Phys. **78**, 204–214 (2010).

[2] The Hipparcos catalogue can be found on `cdsarc.u-strasbg.fr/viz-bin/Cat?I/239` 1997HIP. . . C . . . . . . 0E - European Space Agency SP-1200 (1997), catalogue ID I/239.

[3] B. Cameron Reed, "The composite observational-theoretical HR diagram," J. Roy. Astron. Soc. Can. **92**, 36–37 (1998).

[4] Information about OpenGL and the OpenGL Shading Language (GLSL) can be found on `www.opengl.org.`