Special Relativity in Virtual Reality

René T. Rau[1], Daniel Weiskopf[2], and Hanns Ruder[2]

[1] WSI/GRIS, University of Tübingen, Auf der Morgenstelle 10,
D-72076 Tübingen, Germany
[2] Theoretical Astrophysics, University of Tübingen, Auf der Morgenstelle 10,
D-72076 Tübingen, Germany

Abstract. The appearance of fast moving objects can be calculated according to the Theory of Special Relativity. In addition to the Lorentz contraction the effects of finite light speed and aberration play an important role. These phenomena were first discovered and described correctly in 1959 by Penrose and Terrel. Concerning the visualization of the phenomena there already exist systems with relativistic ray tracing and polygon rendering. Investigating these approaches in detail we found a reformulation of the problem which allows the treatment of acceleration in real-time. Therefore, user interaction could be integrated and a virtual reality for special relativity was possible.

1 Introduction

The Theory of Relativity is a fascinating topic in astrophysics. Most of the phenomena are beyond our normal experience and seem strange for a newcomer in the field. Especially for educational purposes, the visualization of many phenomena can give faster and better understanding of the transformations and effects. Visualization can create new virtual realities which make the effects part of our experience.

One aspect of the Theory of Special Relativity is the rendering of objects moving with relativistic velocities, i.e., velocities near the speed of light. The appearance of fast moving objects was already discussed in the beginning of this century with the first formulation of the theory. However, the early descriptions have been wrong, e.g. Einstein's statements in [4]. Even after the theory had been accepted the problem was not treated correctly, aside from a generally ignored article about the invisibility of the Lorentz contraction by Lampa in 1924. In 1959 Penrose [11] and Terrel [17] recognized the problem for the first time and gave correct solutions. Detailed descriptions were given later by Boas 1961 in [2], Scott and Viner 1965 in [15], and Scott and van Driel 1970 in [14].

We describe the relativistic transformations necessary for the computation of the correct appearance of relativistic objects and restrict our considerations to the correct display of the objects' shapes. Other effects, such as the Doppler effect, will not be considered but can be integrated as well.

The rendering of the objects can be considered as a generalization of already known rendering techniques used in computer graphics to the special relativistic situation. Mainly two approaches exist to obtain relativistic images: ray tracing and polygon rendering. We investigate these approaches with respect to efficiency and image quality.

Most of the literature does not treat the accelerated case or the authors suggest a method which is not suitable for real-time rendering on a workstation (cf. [6]). In this paper we show how the problem can efficiently be reformulated for the situation with acceleration and how a virtual reality environment can be built.

We proceed as follows: In the next section we give a brief introduction to the Theory of Special Relativity. Here we restrict ourselves to the situation with constant relative velocity. In Section 3 we investigate relativistic rendering as described in the literature. The reformulation in the accelerated situation in Section 4 allows the implementation of a virtual environment, which is described in Section 5. In the last section we discuss the results and the future work.

2 Special Relativistic Transformation

The Theory of Special Relativity usually considers inertial systems moving with constant relative velocity. However, accelerations can be treated within the same framework (cf. [8]).

For the moment, let us restrict ourselves to constant relative velocity. Figure 1 shows the situation for two given coordinate systems K and K', where K' is separating from K at a constant velocity v. On the assumption that space is homogeneous and isotropic we can choose a special coordinate system without loss of generality. At time $t = t' = 0$ the origins of the coordinate systems coincide, i.e., $x = x' = 0$, $y = y' = 0$, and $z = z' = 0$. In addition, the axes x and x' are parallel to the relative velocity v, the x-y-plane coincides with the x'-y'-plane, and the x-z-plane coincides with the x'-z'-plane.

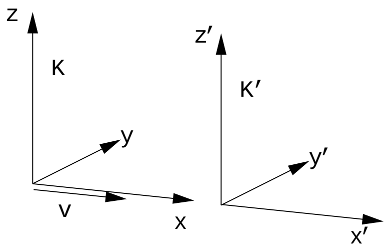

Fig. 1. Two coordinate frames moving with constant relative velocity.

In classical mechanics the coordinate transformations are given by

$$x' = x + vt, \quad y' = y, \quad z' = z, \quad t' = t, \tag{1}$$

the so-called Galilean transformations. The time t remains invariant under these transformations and is in this sense “absolute”. The laws of motion remain invariant as well.

The principle of special relativity states that the laws of nature are invariant under a particular group of transformations, the so-called Lorentz transformations. These transformations are given by

$$x' = \gamma(x + vt), \quad y' = y, \quad z' = z, \quad t' = \gamma(t + \frac{v}{c^2}x) \tag{2}$$

$$x = \gamma(x' - vt'), \quad y = y', \quad z = z', \quad t = \gamma(t' - \frac{v}{c^2}x'), \tag{3}$$

where $\gamma = \frac{1}{\sqrt{1-\beta^2}}$, $\beta = \frac{v}{c}$, and c denotes the velocity of light.

For a derivation of the Lorentz transformations from the Einstein Postulates see [13]. In contrast to the Galilean transformation the Lorentz transformation leaves the speed of light and Maxwell’s equations invariant. For a more detailed presentation of the theory we refer, e.g., to [13] and [8].

3 Special Relativistic Rendering

The process of creating images from three-dimensional models is called rendering. Special relativistic rendering means the process of generating images of fast moving objects or image generation with a fast moving camera according to the equations given in the previous section. In addition, it means the visualization of relations given by these equations in order to obtain a better understanding of the image generation process.

Since the rendering of a static or slowly moving object can be considered as a special case in the relativistic situation, relativistic rendering should be a generalization of techniques already known in the field of computer graphics.

Many renderers for solid three-dimensional objects are based on a polygonal representation of the 3D objects and their surfaces. In many cases this representation is an approximation of the real object by a triangle mesh. In the rendering pipeline the vertices of the triangles are projected onto the image plane. The method is very fast and is supported by modern graphics hardware. For a detailed description of the pipeline we refer to [5].

Another well-known rendering technique is ray tracing. Here, for each pixel a ray is traced from the viewpoint into the scene. The rays are usually assumed to be infinitely thin and infinitely fast, and scattering is left out of consideration. The disadvantage of this technique are high computational costs.

In the relativistic situation Hsiung and Dunn (cf. [7]) formulated relativistic ray tracing. Gekelman et al. described the polygon rendering of a relativistically moving cube (cf. [6]). They also discussed the accelerated situation but their solution was not performable in real-time. In the most recent publication of Chang et al. [3] polygon rendering was used, but no acceleration was considered. Let us describe the two approaches in more detail.

3.1 Polygon Rendering

Let us first consider the situation with a single point light-source, which is at rest in K'. For the explanation of the effects we use spacetime coordinates (ct, x, y, z). In this framework the correct visualization can be derived by purely geometric operations. This geometric interpretation will allow us to treat the accelerated observer (cf. Section 4).

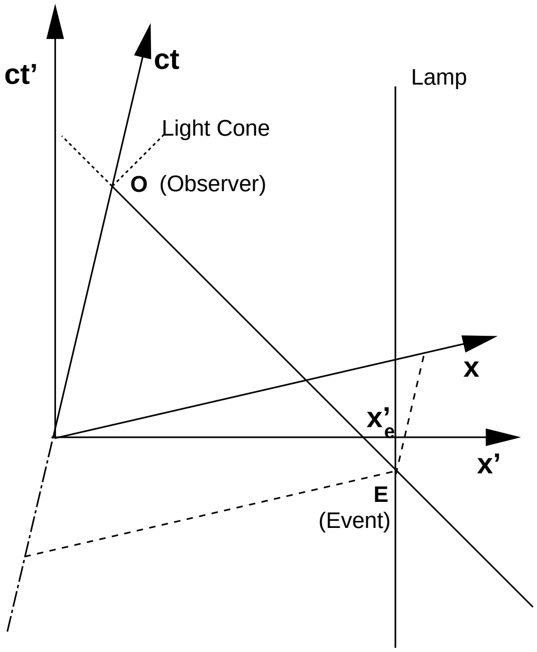

Fig. 2. Minkowski diagram with constant relative velocity.

In Figure 2 we show the Minkowski diagram, which is a spacetime diagram without the coordinates y and z. We use two coordinate systems. In the frame K' the object is at rest whereas the frame K is the observer's rest frame. Let (ct'_o, x'_o) denote the coordinates of the observer in K'. The light propagates on light cones. The light cone at the observer is outlined by the dotted lines. The line $\{(ct', x'_e)|t'\}$ denotes the world line of the point light-source. The intersection of the backward light cone with the world line of the light source is denoted by E, which is the emission position of the light reaching the observer at point O. Once this position is determined, we only have to compute the coordinates of E with respect to the frame K. In the diagram this can be done graphically as shown by the dashed lines, which corresponds to the Lorentz transformation from K' to K.

In four dimensions the coordinates of E can be computed by

$$(ct'_o - ct'_e) = \sqrt{(x'_e - x'_o)^2 + (y'_e - y'_o)^2 + (z'_e - z'_o)^2},$$

where $(ct'_e, x'_e, y'_e, z'_e)$ denote the coordinates of E and $(ct'_o, x'_o, y'_o, z'_o)$ the coordinates of O in K'. With the Lorentz transformation the coordinates of the emission event in K can then be determined. The space coordinates (x_e, y_e, z_e) determine the direction of the incoming light in the image plane and the emission time is not relevant. Light travels along straight lines in 4D space relative to every coordinate system and so does it with respect to a restriction to the three space coordinates. Therefore, standard computer graphics processing can be used for the correct projection onto the image plane.

In the polygonal representation the vertices hold the information such as color, material properties, surface normal etc., and, therefore, can be considered — after the evaluation of a suitable lighting model — as single light sources. We can then apply the transformation described above to all vertices and obtain a completely new object which approximates the emission positions of the objects surface in the coordinate frame of the camera. With Gekelman et al. [6] we call this virtual 3D object *photosurface*. This new object can be projected through a normal 3D projection with correct hidden surface elimination.

In addition to realistic rendering, this method offers another visualization of relativistic effects. The three-dimensional photosurface can be viewed from arbitrary view-points. To understand how the relativistic image was produced viewing the shape of the photo object from different view positions is helpful.

Figure 3 and 4 show the photosurface of a sphere and a cube. The observer is marked as a small sphere. The idea to transform only the vertices of a polygonal representation of a 3D object and to render the resulting object as shaded polygons is quite simple and seems to be very efficient. But due to the highly nonlinear transformation in the proximity of the observer for high velocities, new problems appear with the polygonalization.

The artifacts, which appear mainly at the boundaries, can be reduced by a fine triangulation of the objects. Although the fine triangulation of the moving object can slow down the rendering process the method remains still very fast for velocities up to $\beta = 0.99$.

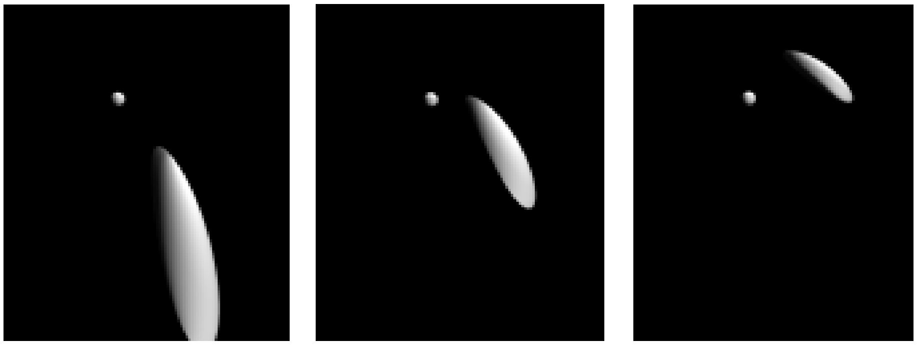

Fig. 3. Photosurfaces of a sphere with $\beta = 0.95$ at different time steps. The observer is marked as a small sphere. The view is perpendicular to the direction of motion.

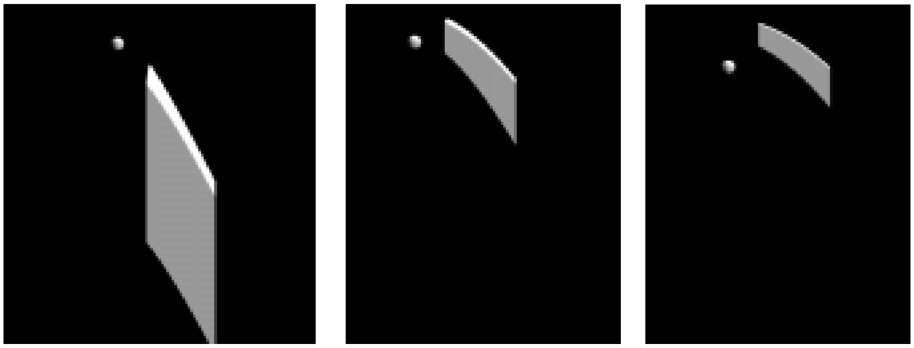

Fig. 4. Photosurfaces of a cube with $\beta = 0.9$ at different time steps. The observer is marked as a small sphere. The view is perpendicular to the direction of motion.

3.2 Ray Tracing

Ray tracing in computer graphics is normally performed in three-dimensional space and one may think of relativistic ray tracing as ray tracing in four-dimensional spacetime. This might be true for a general approach, but in many situations, e.g., the computation of the shape of the object, we can perform the task with "ordinary" ray tracing in three dimensions. This approach was proposed by Hsiung and Dunn in [7].

To explain this method we consider an eye-ray which is traced from the viewpoint through a pixel into the scene. The viewpoint, time, and direction of the ray are given in the coordinate system of the observer. This ray can be described by a four-dimensional start position (ct_0, x_0, y_0, z_0) and a direction $\boldsymbol{d} = (d_x, d_y, d_z)$ obtained from the camera parameters.

The transformation of the point (ct_0, x_0, y_0, z_0) is given by the Lorentz transformation described in Section 2 and results in a point $(ct'_0, x'_0, y'_0, z'_0)$. The direction $\boldsymbol{d}$ is transformed like a velocity and the transformed direction is obtained by

$$\boldsymbol{d}' = (\gamma(d_x + \beta), d_y, d_z)/\gamma(1 + \beta d_x). \tag{4}$$

For a fixed time t_0 we compute the direction for each pixel according to the viewpoint. The pixel coordinates in the observer frame and the time t_0 determine the start position of the ray. We transform the ray according to Equation (4) and send this ray through the scene performing classical ray tracing.

Since the transformation of the ray by Equation (4) is given analytically, a straight line which is bent due to the transformation, is displayed as a bent line in the raytraced image. Therefore, the fine triangulation or, in general, the subdivision of patches and lines is not necessary with ray tracing.

One disadvantage of ray tracing is the fact that the transformation given by Equation (4) is not very intuitive and the resulting image is frequently surprising. The method is well-suited for the generation of high quality images and less suited for educational purposes.

4 Virtual Reality for Relativistic Flights

User interaction being part of a virtual reality requires the consideration of the accelerated observer.

Again, as in the previous section, we use a Minkowski diagram for a fixed object coordinate system to reduce the visualization problem to a geometric one. Figure 5 shows the accelerated situation. The world line of the observer is no longer a straight line. For a given observer position O in K' the current relative velocity can be computed by the tangent to the world line. Once this velocity is known we can proceed as in the case of constant velocity. We obtain the coordinates of the emission event by computing the intersection of the backward light cone with the straight world line of the object. After a coordinate transformation into the current frame K of the observer we can again project the space coordinates onto the image plane.

The computation of the coordinates of the observer's world line can be done successively. Since the world lines of the light sources are not effected by the observer, we are able to render accelerated scenes with almost no delay.

4.1 Transformation Details

For a correct implementation of the accelerated observer it is necessary to parameterize its world line by the so-called *proper time*, which is defined as the time measured by a co-moving clock.

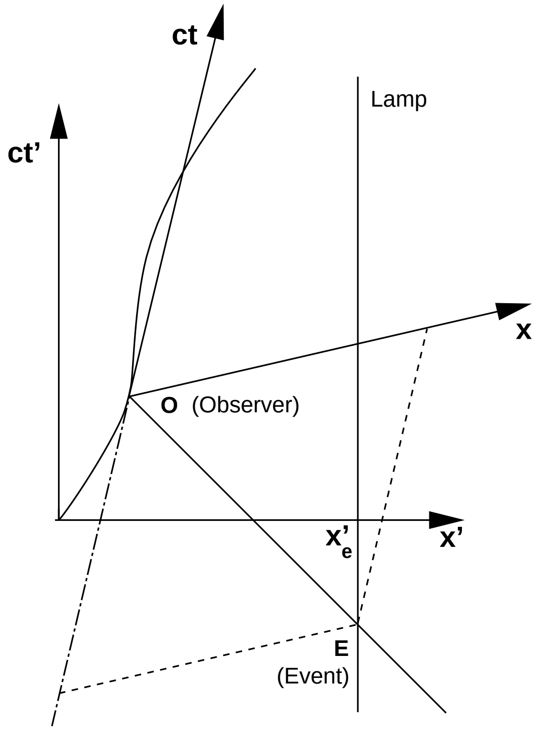

Fig. 5. Minkowski diagram with accelerated observer.

In order to show how this parameterization can be obtained we have to introduce the notion of spacetime and four-vectors and refer to, e.g., [8,9,12] for a detailed presentation. The time coordinate t and the three spatial coordinates (x, y, z) describe a point in spacetime and can be combined to form the position four-vector

$$x^{\mu} = (ct, x, y, z) = (x^0, x^1, x^2, x^3) \qquad , \quad \mu = 0, 1, 2, 3.$$

In this framework a Lorentz transformation is a change of coordinate systems. A *four-vector* is defined as a quantity which has four components (b^0, b^1, b^2, b^3) relative to every coordinate system and which are transformed in the same way as the coordinates (x^0, x^1, x^2, x^3) (cf. [9]).

The *proper time* τ is a Lorentz scalar, i.e., it is independent of the reference frame. The differential proper time is given by

$$d\tau = \sqrt{1 - \beta^2}\, dt = \frac{dt}{\gamma},$$

where β and γ are defined by

$$\beta = \frac{v}{c} \quad , \qquad \gamma = \frac{1}{\sqrt{1-\beta^2}}.$$

Classical quantities such as velocity and acceleration can be extended to corresponding four-vectors.

The four-velocity is defined by

$$u^\mu = \frac{dx^\mu}{d\tau} \tag{5}$$

and the components of the four-velocity are obtained by

$$u^0 = \gamma\, c \quad , \quad u^1 = \gamma\, v_x \quad , \quad u^2 = \gamma\, v_y \quad , \quad u^3 = \gamma\, v_z .$$

The four-acceleration is given by

$$a^\mu = \frac{du^\mu}{d\tau} = \frac{d^2 x^\mu}{d\tau^2}. \tag{6}$$

The effect of the user interaction on the path of the observer, i.e., the observer's world line, is computed according to the equations for the four-velocity and four-acceleration. The user interaction determines the acceleration in the observer frame. From a given three-acceleration (a_x, a_y, a_z) we obtain a four-acceleration

$$a^\mu = (0, a_x, a_y, a_z)$$

in the observer frame, which is Lorentz transformed into the frame K' (see Equation (7)) and yields a coupled system of ordinary differential equations according to Equations (6). The system is numerically solved for the following time step using Euler's method. This way, we obtain the path of the observer in spacetime parameterized by the proper time.

The transformation of the photon emission event from the objects frame K' to the observer's frame K is performed according to the description in the previous section. This transformation can be divided into three parts.

- Translation of the origin of the K' frame to the current position of the observer.
- Lorentz transformation without rotation (Lorentz boost).
- Rotation according to the direction of motion.

The general Lorentz boost (cf. [8, p.69]) is given by

$$\begin{pmatrix} \gamma & -\beta\,\gamma\, n_x & -\beta\,\gamma\, n_y & -\beta\,\gamma\, n_z \\ -\beta\,\gamma\, n_x & (\gamma-1)\, {n_x}^2 + 1 & (\gamma-1)\, n_x\, n_y & (\gamma-1)\, n_x\, n_z \\ -\beta\,\gamma\, n_y & (\gamma-1)\, n_x\, n_y & (\gamma-1)\, {n_y}^2 + 1 & (\gamma-1)\, n_y\, n_z \\ -\beta\,\gamma\, n_z & (\gamma-1)\, n_x\, n_z & (\gamma-1)\, n_y\, n_z & (\gamma-1)\, {n_z}^2 + 1 \end{pmatrix} \tag{7}$$

where $\boldsymbol{n} = (n_x, n_y, n_z)$ is the normalized direction of motion and γ and β denote the velocity parameters defined above.

5 Description of the System

Our user interactive environment is approximately an airplane environment. We provide controls to accelerate and to decelerate in the current flight direction. Moreover, the user can change the direction by accelerating to the left or right side. Finally, a rotation around the current direction of motion is possible. We use keyboard, space-ball, and space-mouse as input devices.

We use textures, which are extremely useful for visualizing the apparent distortion of large scale objects due to relativistic effects. Our implementation accepts arbitrary scenes and we support VRML 1.0 file format (cf. [1]). In a preprocessing step we perform the fine triangulation of the surface patches. The meshing can be controlled interactively and individually for each surface in a radiosity program called RadioLab (cf. [16]). The rendering supports level-of-detail and is based on OpenInventor and OpenGL (cf. [18] and [10]).

An impression of the system can be obtained from Figures 6-9. In Figure 6 the test scene with some geometric primitives is at rest. Besides the objects the parameters of the camera can be seen. In Figures 7-9 images are displayed which have been taken during a flight with acceleration. The flight direction coincides with the viewing direction. Concerning the acceleration the most interesting effect is aberration of the incoming photons (cf. [8, p.68]). Some objects seem to move away from the observer although the movement is towards the objects.

This test scene with about 9,000 triangles was investigated on a SGI O2, R10000, workstation. With our current implementation we obtain a performance of about 25 frames per second for the non-relativistic and about 20 frames per second for the relativistic movement.

6 Conclusions and Further Work

According to the Theory of Special Relativity we described the transformation necessary for rendering. We described the two classical rendering techniques which are able to generate realistic images of fast moving objects. Polygon rendering is able to generate images very quickly and the image quality for lower velocities is sufficient for real-time applications. The ray tracing technique is much slower, but since the transformation is analytic, artifacts at the silhouettes do not appear.

Through the consideration of the spacetime the rendering equations can be obtained from intersection calculations. Our approach can be extended to the accelerated situation and allows rendering in real-time. Therefore, it was possible to build a virtual relativistic reality. The implemented system reveals surprising effects even for a user familiar with the theory.

We will integrate the Doppler effect and more realistic illumination into our future work. Finally, we plan to provide a freely distributed version of our environment based on OpenGL.

Acknowledgments

We would like to thank the referees for their constructive and detailed suggestions.

This work was supported by the German Science Foundation (DFG) and is part of the projects D1 and D4 within the Sonderforschungsbereich 382.

References

1. G. Bell, A. Parisi, and M. Pesce, *VRML: Version 1.0 specification*, http://webspace.sgi.com/Archive/Spec1.0/index.html, 1995.
2. M. Boas, *Apparent shape of large objects at relativistic speeds*, Amer. J. Phys. **29** (1961), 283.
3. M.-C. Chang, F. Lai, and W.-C. Chen, *Image shading taking into account relativistic effects*, ACM Trans. Graphics **15** (1996), 265–300.
4. A. Einstein, *Zur Elektrodynamik bewegter Körper*, Ann. d. Phys. **17** (1905), 891.
5. J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Fundamentals of interactive computer graphics*, second ed., Addison-Wesley Publishing Company, 1990.
6. W. Gekelman, J. Maggs, and L. Xu, *Real-time relativity*, Computers in Physics (1991), 372–385.
7. P.-K. Hsiung and R. H. P. Dunn, *Visualizing relativistic effects in spacetime*, Proceedings of Supercomputing '89 Conference, 1989, pp. 597–606.
8. C. Misner, K. Thorne, and J. Wheeler, *Gravitation*, Freeman and Company, 1973.
9. C. Møller, *The Theory of Relativity*, second ed., Clarendon Press, 1972.
10. J. Neider, *OpenGL programming guide*, Addison-Wesley, 1993.
11. R. Penrose, *The apparent shape of a relativistically moving sphere*, Proc. Cambr. Phil. Soc. **55** (1959), 137.
12. W. Rindler, *Introduction to Special Relativity*, second ed., Clarendon Press, Oxford, 1991.
13. H. Ruder and M. Ruder, *Die Spezielle Relativitätstheorie*, Vieweg, 1993.
14. G. Scott and M. Driel, *Geometrical appearances at relativistic speeds*, Amer. J. Phys. **38** (1970), 971.
15. G. Scott and R. Viner, *The geometrical appearance of large objects moving at relativistic speeds*, Amer. J. Phys. **33** (1965), 534.
16. R. Sonntag, *RadioLab: An object oriented global illumination system*, Dissertation, Universität Tübingen, Tübingen, Germany, 1997, Forthcoming.
17. J. Terrel, *Invisibility of the Lorentz contraction*, Phys. Rev. **116** (1959), 1041.
18. J. Wernecke, *The Inventor Mentor*, Addison-Wesley, 1995.

Appendix: Color Plates

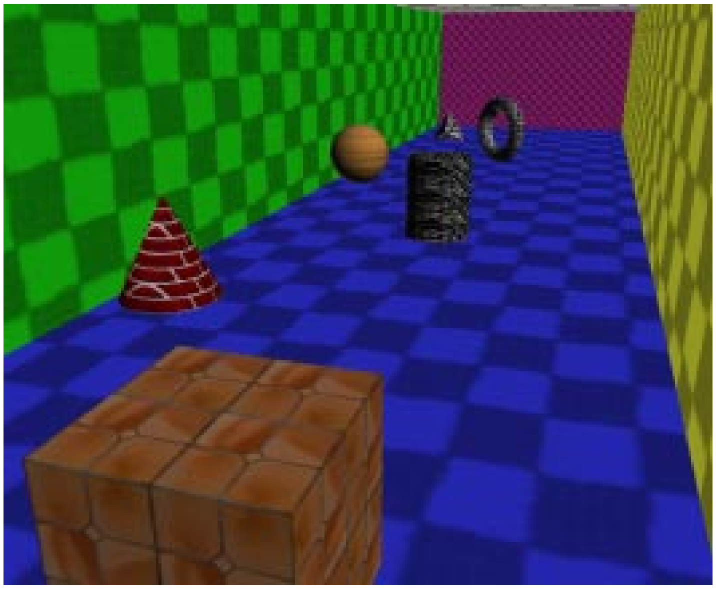

Fig. 6. A test scene for the virtual relativistic reality at low speed

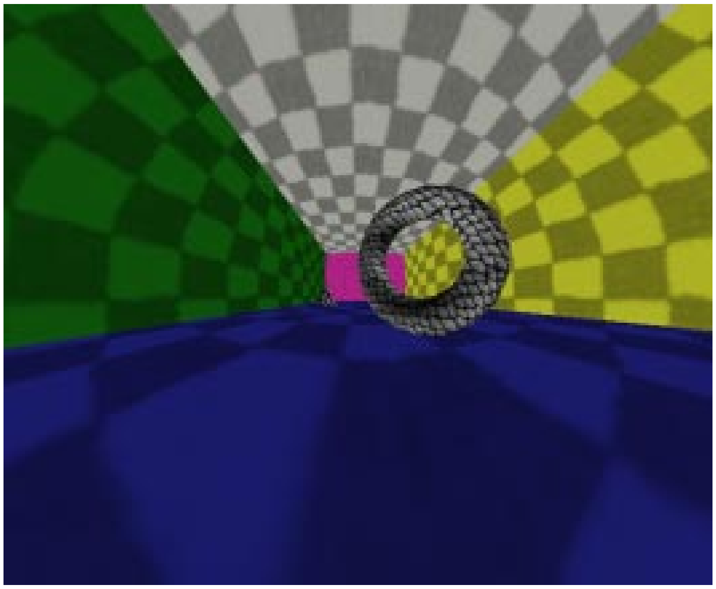

Fig. 7. The same scene as above with $\beta = 0.95$ at time $\tau = 0$

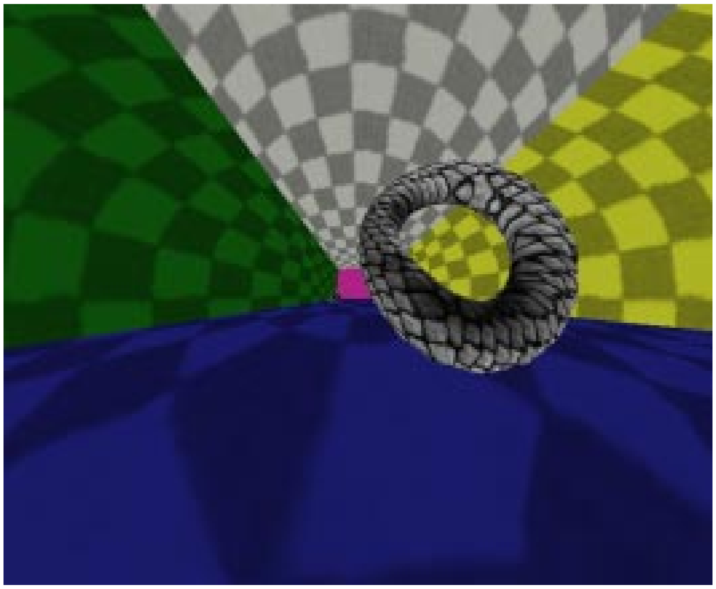

Fig. 8. The scene after acceleration ($\beta = 0.983$, $\tau = 2.1$)

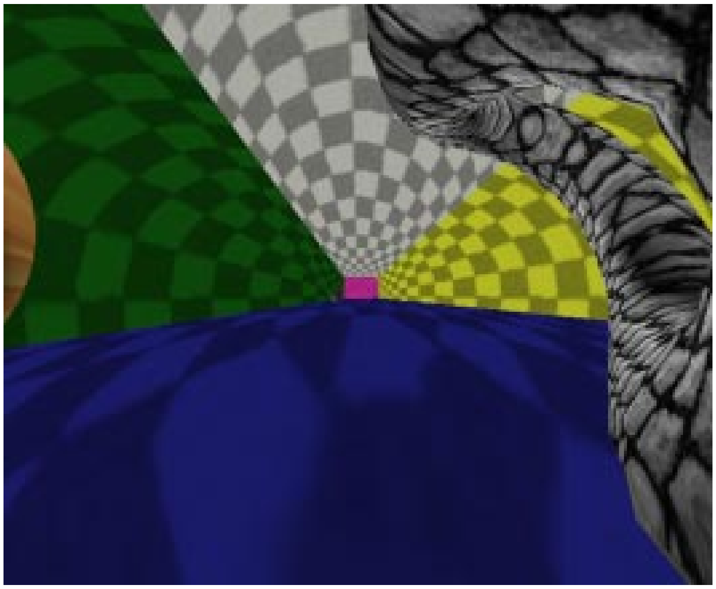

Fig. 9. The scene at $\tau = 3.7$ with $\beta = 0.994$