

# An Immersive Virtual Environment for Special Relativity

Daniel Weiskopf

Theoretical Astrophysics, University of Tübingen, Tübingen, Germany  
e-mail: weiskopf@tat.physik.uni-tuebingen.de

---

## Abstract

*This paper presents an immersive virtual environment which allows the user to virtually explore and experience special relativity, including phenomena such as Lorentz contraction, time dilation, aberration, and finite speed of light.*

*The relativistic-vehicle-control metaphor is introduced for navigating at high velocities. This is a physically based camera control technique. Accelerations of the relativistic observer are investigated. Furthermore, tracking of both the position and velocity of the user is considered.*

*A geometric approach to relativistic polygon rendering is described. The rendering pipeline is extended to accomplish the relativistic transformations in parallel. The implementation supports multiprocessing and multipipe systems for fast rendering and the same frame rates can be achieved for relativistic visualization as for non-relativistic rendering.*

*The new interaction techniques and a well performing implementation make possible a virtual environment that provides an intuitive approach to special relativity.*

---

## 1. Introduction

Einstein's theory of special relativity is widely regarded as a difficult and hardly comprehensible theory, even today—more than ninety years after its publication in 1905<sup>7</sup>. One important reason for this is that the properties of space, time, and light in relativistic physics are totally different from those in classical, Newtonian physics. In many respects, they are contrary to human experience and everyday perception, which is based on low velocities.

In the real world, mankind is limited to very small velocities compared to the speed of light. For example, the speed of light is a million times faster than the speed of an airplane and 40,000 times faster than the speed at which the Space Shuttle orbits the Earth. Even in the long term future, there is no hope of achieving velocities comparable to the speed of light.

Therefore, computer simulations are the only means of visually exploring the realm of special relativity. The intent of this paper is to present an immersive virtual environment which allows the user to submerge into the world of special relativity. An egocentric frame of reference is adopted. This will enable us to experience special relativity right in front of our eyes, including phenomena such as Lorentz contraction,

time dilation, aberration, and finite speed of light. Newcomers and students will benefit from this intuitive approach, but even people familiar with the theory will gain some interesting new insight which is normally hidden by the abstract formalism, cf. Section 8.

In addition to the issues related to normal non-relativistic immersive virtual environments, the relativistic environment essentially has two extra demands.

First, a computational model for relativistic rendering is needed. Here, relativistic polygon rendering, which is based on the apparent shapes of objects as seen by a relativistic observer, is used. An additional step is introduced into the rendering pipeline to account for relativistic effects. A parallel implementation of the relativistic transformations will be presented. Therefore, on a multiprocessor system, relativistic visualization is achieved at the same frame rates as non-relativistic rendering.

Secondly, a new means of interaction with the virtual environment has to be established. The focus is on an appropriate camera control mechanism. The *relativistic-vehicle-control* metaphor is introduced for navigating at high velocities. The intent of the virtual environment is to be as close as possible to an important part of today's physics. Therefore,

the camera control should be based on physical laws. The *relativistic-vehicle-control* metaphor consists of both active and passive locomotion. Passive locomotion is implemented in the form of travelling in a fast vehicle. The user controls the movements of the vehicle by supplying information about the current acceleration. Furthermore, the user can walk within the virtual environment, which accounts for direct, active locomotion. Image generation in the non-relativistic situation is influenced by viewpoint and viewing direction. Relativistic rendering, however, depends on the current velocity of the viewer as well. Therefore, tracking of both position and velocity in the virtual environment has to be considered.

The contributions of this paper are:

- A geometric approach to relativistic polygon rendering.
- Treatment of the accelerated observer.
- The *relativistic-vehicle-control* metaphor, a physically based metaphor for navigating at high velocities. Due to the nature of special relativity, tracking of user's position and velocity is considered.
- Extension of the normal rendering pipeline for relativistic rendering in parallel. With this technique, the same frame rates can be achieved for relativistic rendering as for non-relativistic rendering.
- An immersive virtual environment that provides a natural and intuitive approach to special relativity.

The paper is organized as follows. The next section gives a brief survey on previous and related work. The first part is related to special relativistic rendering. The second part addresses issues of virtual environments in general, and of navigation and travelling within a virtual world in particular. In Section 3, the mathematical foundations of the theory of special relativity are introduced in so far as needed for relativistic visualization. In Section 4, a geometric approach to relativistic polygon rendering is presented. Section 5 gives a treatment of the accelerated relativistic observer. The equations of motion are presented, and it is shown how they can be solved numerically. In Section 6, a virtual environment for special relativity is developed. In Section 7, the implementation is described with focus on high performance rendering and parallel algorithms. Section 8 shows results of relativistic visualization and performance measurements. In Section 9, the results are discussed and an outlook on future work is given.

## 2. Previous and Related Work

Remarkably, the issue of visual perception in special relativity was ignored for a long time, or wrong interpretations were given. Apart from a previously disregarded article by Lampa<sup>18</sup> in 1924 about the invisibility of the Lorentz contraction, it was only in 1959 that the first solutions to this problem were described by Penrose<sup>22</sup> and Terrell<sup>30</sup>. Later, more detailed descriptions of the geometrical appearance of

fast moving objects were given by Weiskopf<sup>33</sup>, Boas<sup>1</sup>, Scott and Viner<sup>28</sup>, and Scott and van Driel<sup>27</sup>.

Direct visualization of special relativity on a computer was first presented by Hsiung and Dunn<sup>13</sup>. They proposed an extension of normal three-dimensional ray tracing for image shading of fast moving objects. This technique accounts for relativistic effects on the apparent geometry as seen by the observer. Hsiung et al.<sup>15</sup> investigated relativistic ray tracing in more detail and included the visualization of the Doppler effect. Based on relativistic ray tracing, Hsiung et al.<sup>14</sup> used the temporal variation of the color of objects in the scene for the visualization of relativistic time dilation.

Hsiung et al.<sup>16</sup> introduced the *time-buffer* for fast visualization of relativistic effects. The *time-buffer* technique resembles the normal z-buffer and can be mapped onto it. The *time-buffer* technique allows for relativistic polygon rendering. It makes use of contemporary computer graphics hardware and achieves interactive frame rates. Gekelman et al.<sup>8</sup> described the polygon rendering of a relativistically moving cube. They mentioned the trajectory of an accelerated cube. In their description, however, an integral along the trajectory of every point on the object needs to be solved numerically, which is time consuming and leads to a polygonal approximation of the trajectory. Furthermore, for this calculation, the velocity has to be known for all relevant times, but Gekelman et al. gave no description of how this velocity could be determined. Chang et al.<sup>2</sup> investigated the polygon rendering approach in detail and gave a comprehensive presentation. However, they did not consider acceleration at all.

A lot of research has been conducted on issues related to virtual environments. An overview can be found, for example, in <sup>12, 5</sup>. Cruz-Neira et al.<sup>4</sup> presented the CAVE, a surround-screen, projection-based, immersive virtual environment, which is also a model for the virtual environment of this paper.

A number of researchers have addressed issues related to user interaction in general. A survey is given in <sup>9, 19</sup>. Robnett and Holloway<sup>25</sup> described the main actions in virtual worlds under control of a manual input device. Hinckley et al.<sup>11</sup> focused on interaction techniques relying on free-space three-dimensional input devices. Jacoby and Ellis<sup>17</sup> gave a summary of menus in virtual environments.

This paper is focused on the issues of navigation and locomotion in a virtual environment. Ware and Osborne<sup>32</sup> described three interaction metaphors which they called *eyeball-in-hand*, *scene-in-hand*, and *flying-vehicle-control*. The eyeball-in-hand technique maps the position and orientation data of a spatial input device directly onto the viewpoint, whereas the scene-in-hand technique maps the input onto the movement of the virtual world. For the flying-vehicle-control, the velocity is controlled by the displacement of the input device. They noticed that, with the flying-vehicle-control, subjects adopted different metaphorical analogies depending on the scene. For example, percep-

tion could be similar to flying or to walking. The flying-vehicle metaphor will serve as the basis for the development of the *relativistic-vehicle-control* metaphor presented in this paper.

Turner et al.<sup>31</sup> introduced the *virtual-camera* metaphor. This is a physically based metaphor for controlling the camera motion. They described an abstract physical model of a camera object with parameters such as mass, moment of inertia, and friction coefficients. The trajectory of the camera is determined by classical, Newtonian mechanics. Some of these ideas will be adopted for the relativistic situation in Sections 5 and 6.

Chapman and Ware<sup>3</sup> proposed a predictor-based feedback aid for navigating through virtual environments. Their method provides the user with feedback on his current velocity by showing future position and orientation of the viewpoint. The predictor is a rectangular frame placed in front of the viewpoint and represents the extrapolated position of the viewpoint. In the relativistic simulation, the feedback to the current velocity will be naturally induced by laws of physics which change the perceived geometry, cf. Sections 6 and 8.

The world-in-miniature (WIM) technique<sup>29</sup> allows a rapid change of the viewpoint by manipulating a small virtual model of the scene which is held in the hand. This is an exocentric frame of reference which gives a feeling of looking in from the outside, whereas the other navigation techniques described above establish an egocentric frame of reference. In the relativistic simulation, the egocentric frame of reference dominates. An additional outside view, however, might be useful for orientation and for the understanding of the virtual world.

### 3. Lorentz Transformation

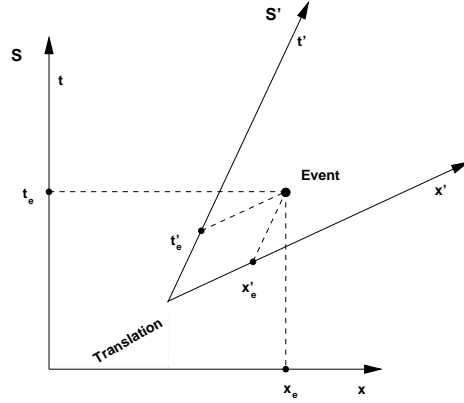
For a thorough investigation of relativistic visualization the notion of spacetime and four-vectors has to be introduced. A detailed presentation can be found in<sup>20, 21, 24</sup>. The time coordinate  $t$  and the three spatial coordinates  $(x, y, z)$  describe a point in spacetime and can be combined to form the position four-vector

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

where  $c$  is the speed of light. The Lorentz transformation connects two inertial frames of reference and leaves the speed of light invariant. The Lorentz transformation is just a change of coordinate systems between the associated inertial frames. A general *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 position coordinates  $(x^0, x^1, x^2, x^3)$ .

By including translations of the frames of reference, the Lorentz transformation is extended to the Poincaré transformation. Figure 1 shows a geometric approach to the Poincaré transformation in the form of the Minkowski diagram.

Minkowski diagram is a spacetime diagram without the coordinates  $y$  and  $z$ .



**Figure 1:** Minkowski diagram showing two frames of reference,  $S$  and  $S'$  respectively. The translation of the origins of the two coordinate systems is explicitly marked. The different velocities of the reference frames cause the rotation of the axes of  $S'$ . Spatial rotation is not considered because the Minkowski diagram shows only one spatial dimension. The projection of the Event onto the axes yields the coordinates  $(t_e, x_e)$  and  $(t'_e, x'_e)$  of the event in the two respective frames of references, which provides a geometric interpretation of the Poincaré transformation.

The Poincaré transformation can be divided into three consecutive parts: translation, Lorentz boost (Lorentz transformation without rotation), and rotation. The general Lorentz boost, cf. <sup>20</sup>p.69, is given by

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

where  $\vec{n} = (n_x, n_y, n_z)$  is the normalized direction of motion. The velocity parameters  $\gamma$  and  $\beta$  are defined by

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

with the velocity  $v$ .

Rotations can be computed as in the non-relativistic three-dimensional case. Translation in spacetime is analogous to translation in normal three-dimensional space and is based on component-wise addition.

### 4. Relativistic Polygon Rendering

An egocentric frame of reference is adopted for relativistic rendering. The scenario is as follows. The user, i.e. the observer, is moving freely through a virtual world. The objects of the scene are at rest relative to each other. For the explanation of relativistic polygon rendering spacetime coordinates

are used. In this framework, the correct visualization can be derived by purely geometric operations. This geometric interpretation allows for the treatment of the accelerated observer.

Let us investigate the generation of a snapshot taken by the relativistic observer. At the time of image production, the observer is moving with velocity  $\vec{v} = (v_x, v_y, v_z)$  with respect to reference frame  $S_{\text{world}}$ .  $S_{\text{world}}$  is the frame of reference of the outside world. Frame  $S_{\text{obs}}$  is a co-moving reference frame with respect to the observer, i.e., at the moment of image generation, it has the same velocity and direction of motion as the observer. First, only a single point light-source which is at rest in  $S_{\text{world}}$  is considered. In Figure 2, the associated Minkowski diagram is shown. 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  $E$ .  $E$  is the emission four-position of the light reaching the observer at point  $O$ . Once this position in spacetime is determined, the coordinates of  $E$  with respect to frame  $S_{\text{obs}}$  have to be computed. In the diagram this can be done graphically as shown by the dotted lines, which corresponds to the Poincaré transformation from  $S_{\text{world}}$  to  $S_{\text{obs}}$ .

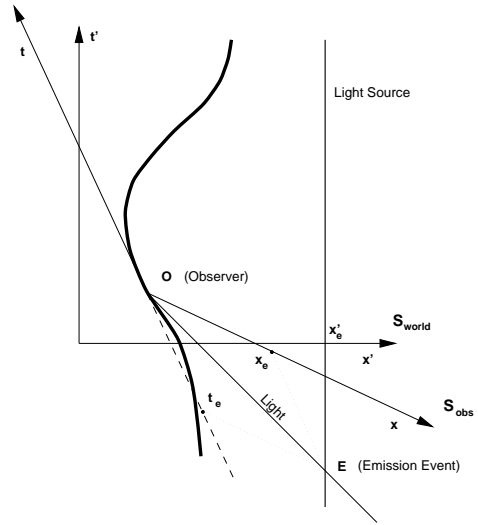
In the following, the primed coordinates are with respect to  $S_{\text{world}}$  and the unprimed coordinates are with respect to  $S_{\text{obs}}$ . In frame  $S_{\text{world}}$ , the time coordinate of  $E$  can be computed by

$$(x_o^{0'} - x_e^{0'}) = \sqrt{(x_e^{1'} - x_o^{1'})^2 + (x_e^{2'} - x_o^{2'})^2 + (x_e^{3'} - x_o^{3'})^2}, \quad (2)$$

where  $x_e^{u'}$  denote the coordinates of  $E$  and  $x_o^{u'}$  the coordinates of  $O$  in  $S_{\text{world}}$ . With the Poincaré transformation described in the previous section, the coordinates of the emission event in  $S_{\text{obs}}$  can then be calculated.

Light travels along straight lines in four-dimensional spacetime relative to every coordinate system and so does it with respect to the restriction to the three space coordinates. Therefore, the space coordinates  $(x_e^1, x_e^2, x_e^3)$  determine the direction of the incoming light in the reference frame of the observer and represent the apparent position of the light source as seen by the observer.

For polygon rendering, solid three-dimensional objects can be represented by an approximation using a triangle mesh. The vertices of the triangles hold information such as color, material properties, surface normal etc., and can be treated like the point light-source above. After the relativistic transformation of the vertices, standard computer graphics processing can be used for hidden-surface removal and the projection onto the image plane. Standard hidden-surface removal such as z-buffer gives correct results, since the spatial distance of an emission event is directly related to its temporal distance, cf. Equation 2. In frame  $S_{\text{obs}}$ , the vertices associated with “newer” events will correctly hide vertices associated with “older” ones.



**Figure 2:** Minkowski diagram showing the world line of the accelerated observer with the co-moving reference frame  $S_{\text{obs}}$ . The light source is at rest in reference frame  $S_{\text{world}}$ . The coordinates  $(t_e, x_e)$  of the emission event  $E$  are determined by projection onto the axes of  $S_{\text{obs}}$ .

When the geometrical transformation of a vertex is performed relativistic effects on the illumination could also be included. These are the Doppler and searchlight effects. The Doppler effect accounts for a shift in wavelength of the incoming light and thus for a change of color. The searchlight effect accounts for a transformation of intensity. However, in the current implementation, only the relativistic effects on the geometry are considered.

## 5. Accelerated Observer

The correct implementation of the accelerated observer needs the world line of the observer to be parameterized by the proper time. The proper time is defined as the time measured by a co-moving clock and is thus the appropriate measure of time for an egocentric point of view.

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

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

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

The four-velocity is defined by

$$u^\mu = \frac{dx^\mu}{d\tau}, \quad (3)$$

and the components of the four-velocity are obtained by

$$u^0 = \gamma c, \quad u^1 = \gamma v_x, \quad u^2 = \gamma v_y, \quad u^3 = \gamma v_z, \quad (4)$$

where  $\vec{v} = (v_x, v_y, v_z)$  is the normal three-velocity.

The four-acceleration is given by

$$a^\mu = \frac{du^\mu}{d\tau} = \frac{d^2x^\mu}{d\tau^2}. \quad (5)$$

The user interaction determines the acceleration in the observer frame because an egocentric point of view is adopted. From a given three-acceleration  $(a_x, a_y, a_z)$  the four-acceleration in the observer frame is obtained by

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

The four-acceleration is transformed into frame  $S_{\text{world}}$  by the associated Lorentz transformation, cf. Section 3. The equations of motion, Equation (5), resemble those of classical physics. The quantities  $a^\mu$ ,  $u^\mu$ ,  $x^\mu$ , and  $\beta$  are functions of  $\tau$ . The Lorentz transformation of the four-acceleration into frame  $S_{\text{world}}$  causes a mixture of space and time components, resulting in a coupled system of ordinary differential equations. The initial value problem is numerically solved using Euler's method, i.e. the four-velocity and four-position of the observer are obtained for the following time step. Finally, the trajectory of the observer in spacetime parameterized by the proper time is obtained.

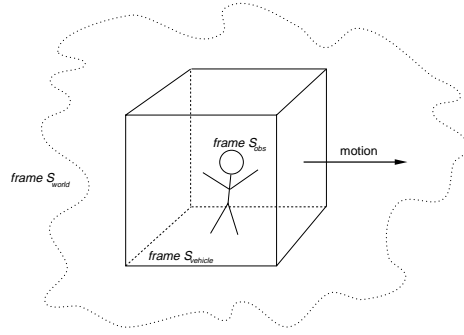
Note that special relativity is capable of describing accelerations as long as gravitation is ignored. (Gravitation is the domain of general relativity). This seems awkward, since Lorentz transformations are restricted to inertial frames of reference. However, in every moment of time a co-moving inertial frame can be found and results known from inertial frames of reference can be used. The concept of co-moving frames of reference enables us to deal with acceleration in special relativity.

## 6. Relativistic Virtual Environment

Since relativistic visualization is inherently connected with motion, a relativistic virtual environment has to provide some kind of fly or drive mode. This paper introduces the *relativistic-vehicle-control* metaphor for interacting with the virtual world. In the following, the elements of this metaphor are presented.

The following scenario is proposed. The user resides in a fast vehicle which enables him to explore a static virtual world. The motion of the vehicle is controlled by user input. The spatial input device offers three translational degrees of freedom, which determine the acceleration of the vehicle in the respective directions. The user can turn the vehicle around the axis of motion by rotating the mouse around this axis. The front screen always points into the direction of motion, i.e. the orientation of the vehicle is changed by lateral acceleration in a way that the front window is turned towards the direction of motion. This way, the vehicle behaves very similarly to a car or plane. The other important element of interaction is that the user can freely walk within this vehicle. The movement of the vehicle yields passive locomotion,

the movement of the person yields active, direct locomotion. The situation is depicted in Figure 3.



**Figure 3:** Sketch of the virtual environment. The user can walk within a moving, cube-shaped vehicle, whose motion is controlled by user input. Both position and velocity of the head of the user are tracked and, together with the underlying motion of the vehicle, account for the relativistic transformation. Reference frame  $S_{\text{obs}}$  is associated with the user; reference frame  $S_{\text{vehicle}}$  with the vehicle, and reference frame  $S_{\text{world}}$  with the outside objects.

This scenario introduces an additional frame of reference. In the following, the reference frame of the objects is denoted as  $S_{\text{world}}$ , the reference frame of the vehicle as  $S_{\text{vehicle}}$ , and the reference frame of the user as  $S_{\text{obs}}$ .

Input from the three-dimensional mouse determines the acceleration of the surrounding vehicle. The spatial midpoint of the cube-shaped vehicle is chosen as the reference point of frame  $S_{\text{vehicle}}$ . The acceleration due to user input has effect on the trajectory of this reference point. The treatment of the accelerated observer in Section 5 is adopted for the reference point. The path of the reference point through spacetime can be calculated with the method in Section 5. It is assumed that the coordinate system  $S_{\text{vehicle}}$  is a co-moving frame with respect to the reference point. Thus, for every moment in time, the frame  $S_{\text{vehicle}}$  of the vehicle is known.

The normal three-position of the user within the vehicle is measured by a head tracking device. The three-velocity is, in principal, calculated by taking the difference of consecutive positions. With Eqs. (1) and (4), the respective four-position and four-velocity can be determined in frame  $S_{\text{vehicle}}$ . The Lorentz transformation from  $S_{\text{vehicle}}$  to  $S_{\text{world}}$  yields the four-position and four-velocity of the user with respect to the coordinate system of the objects. The Lorentz transformation accounts for the relativistically correct addition of velocities and for the Lorentz contraction of the measured distances.

Finally, the four-position and four-velocity of the user's head are used to accomplish relativistic polygon rendering from Section 4. This way, the images are generated as seen by the user.

In order to notice relativistic effects on the apparent geom-

entry one has to change the “speed of light” or use different time and length scales in the virtual world. For example, the dimensions of the vehicle could be scaled to the range of light seconds. Eventually, the “speed of light” is comparable to the user’s walking speed. Therefore, the measured velocity of the user has to be limited to the speed of light by a filter.

In addition to six degrees of freedom for motion, the three-dimensional mouse provides three buttons. Acceleration along the current direction of motion is accomplished by moving the mouse while the left button is pressed. Accordingly, acceleration perpendicular to the current direction of motion needs the right button to be pressed. Pressing the middle button enables the rotation of the vehicle. This clutching mechanism allows the user to separately control the various kinds of motion. It also overcomes the problem of fatiguing which could occur when the user would be permanently holding the physical mouse.

To sum up, the *relativistic-vehicle-control* metaphor consists of two main parts. The first element is a fast virtual vehicle whose motion is governed by the relativistic equations of motion. The second part is the consideration of the user’s movement within the vehicle. In the non-relativistic limit, the *relativistic-vehicle-control* metaphor resembles the *flying-vehicle-control* metaphor. At low speed, the main extension of the *relativistic-vehicle-control* metaphor is a physically motivated velocity control which is based on the acceleration supplied by the user. The relativistic situation requires the following extensions: solving the relativistic equations of motion, tracking of the user’s speed, and relativistic transformation of the user’s position and velocity.

Input from magnetic tracking devices is affected by a lot of noise. Therefore, the velocity calculated from consecutive positions of the head is quite inaccurate. Since relativistic image generation is very sensitive to velocity changes, unacceptable fluctuations in the rendered images would be caused by this method. To overcome this problem, the mean value of the velocity in the last  $n$  frames is used instead, where  $n$  is usually about four. The drawback is a higher latency with respect to direct velocity tracking.

Application control, i.e. the communication between user and system which is not part of the virtual environment, is implemented in the form of a three-dimensional menu floating in the virtual world. The menu is an object which is at rest in the vehicle and which is not subject to relativistic transformations. Menu choices are selected with a virtual laser pointer. Figures 6 and 7 show the menu during a typical simulation.

The menus offer several options. For example, various tracking models can be chosen. First, the physically correct position and velocity tracking can be used, which was described above. Second, tracking of only the position is supported. This option resembles normal tracking techniques. Third, tracking can be disabled, which leaves the user with

passive locomotion which is induced by the motion of the vehicle. The miscellaneous tracking techniques can be useful to make step-by-step acquaintance with the relativistic world. In addition to a fly mode with acceleration in all spatial dimensions, a drive mode is implemented which suppresses vertical accelerations. This constraint helps to navigate through flat scenes and can give the impression of walking. The user can hold the simulation at any time to take a three-dimensional “snapshot”.

## 7. Implementation

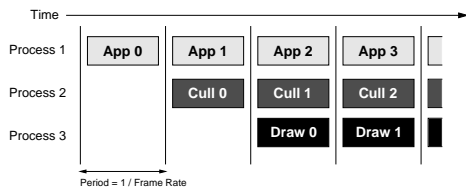
The implementation of the relativistic virtual world runs in the *CUBE*<sup>10</sup>, which is similar to the *CAVE*<sup>4</sup> immersive virtual environment. The *CUBE* is a classical four side back-projection system, consisting of three walls and a floor. Stereo shutter glasses are used to achieve three-dimensional imagery. A magnetic tracking device mounted on the glasses provides information about the position and orientation of the user. A three-dimensional mouse is used as spatial input device with six degrees of freedom. The *CUBE* is driven by a Silicon Graphics Onyx2 system with 14 R10000 CPUs and three InfiniteReality pipes.

The implementation is an extension of the non-relativistic COVER renderer<sup>23</sup>. Rendering is performed using IRIS Performer<sup>26,6</sup>. The geometry nodes of Performer are extended and made aware of special relativity by subclassing. The derived geometry nodes hold both the original vertices and the transformed vertices which are actually rendered.

For optimal performance, Performer-based multiprocessor and multipipe support is used. The flow through the rendering pipeline is modelled into application, culling, and draw stages, which are partitioned into separate processes. This enables the three stages to work in parallel. Figure 4 shows a diagram of the multiprocessor model. The application stage handles user input and carries out the relativistic transformation, the culling stage traverses the visual database, accomplishes view frustum culling, and creates Performer-based display lists, the draw stage generates the actual image. The number of processes for the culling and the draw stages depends on the number of graphics subsystems. The application stage, however, is always handled by a single process.

The relativistic transformations are performed in the application process for two reasons. First, in non-relativistic simulations, the draw process takes much more time than the application and culling processes. Therefore, no further work should be done in the draw process. Secondly, the relativistic transformation is independent of the orientation of the camera, i.e. several culling and draw processes in a multipipe system can use the same transformed vertices which are computed only once per frame.

As shown in Figure 4, the application, culling, and draw processes work on three different frames at the same time.



**Figure 4:** Multiprocessing model of IRIS Performer. App  $n$  is the application process for the  $n$ th frame, Cull  $n$  and Draw  $n$  are the respective culling and draw processes. The vertical lines are frame boundaries, which are used to synchronize the various stages in the rendering pipeline. In a multipipe system, the culling stage and the draw stage, respectively, are actually running in several processes, depending on the number of pipes.

In order to separate the effects of these processes the transformed vertices are held in so-called flux buffers. This means that each process works on its own copy of the vertex buffer. The flux buffers are cycled through as frames change.

Depending on the kind of scene and the graphics performance, the relativistic transformations in the application stage might delay the culling and draw stages. Therefore, an additional multithreading model which computes the relativistic transformation synchronously to the application process is implemented. The application process creates extra processes which are synchronized at frame boundaries by using semaphores. Since computation time for each vertex is constant, load balancing is based on the number of vertices. Furthermore, the transformation of a vertex is independent of the other vertices and allows parallelism on a per vertex basis.

This way, the rendering pipeline is extended by an additional transformation of the vertices which is processed concurrently to the other stages. Therefore, rendering performance is limited by the draw process and the graphics hardware only, and the same frame rates are achieved for relativistic rendering as for non-relativistic rendering.

## 8. Results

Figures 5 through 10 show snapshots taken during typical simulations.

Very interesting effects occur when the velocity is being changed due to acceleration. Objects seem to move away from the user when he is increasing speed towards the objects. Conversely, the objects are apparently coming closer when the speed is being decreased. The situation is depicted in Figures 8 through 10. The three snapshots were taken when the user was accelerating.

The effects of acceleration can be explored in two ways. First, there is an indirect approach by controlling the motion of the surrounding vehicle. Secondly, the user can directly

change the velocity by walking through the virtual environment. Here, the interaction takes full advantage of velocity tracking.

Several students of physics and other sciences had the opportunity to use the relativistic virtual environment. Some of them were first time users of a CAVE-like environment who took a little time to get used to stereo viewing and handling the three-dimensional mouse. Afterwards, they needed only a few minutes to become familiar with the relativistic interaction techniques. The restriction to acceleration in only two dimensions proved to be very helpful for the beginners. Direct locomotion was reported to be a very intuitive means of interaction.

Performance measurements for the three test scenes are documented in Table 1. They show that the same frame rates are achieved for relativistic rendering as for non-relativistic rendering. The frame rates are upheld during the whole simulation. The big test scene 2 is included to show the performance of relativistic rendering under heavy load.

	scene 1	scene 2	scene 3
number of vertices	21,198	135,907	20,786
number of triangles	18,770	113,397	19,864
non-relativistic rendering	14.2 fps	7.1 fps	19.0 fps
relativistic rendering, single-threaded	14.2 fps	2.6 fps	14.2 fps
relativistic rendering, multi-threaded	14.2 fps	7.1 fps	19.0 fps
number of additional threads	0	2	1

**Table 1:** Rendering performance on an Onyx2 system with 14 R10000/195MHz CPUs. Two InfiniteReality pipes are used for rendering onto four screens in stereo. The size of a single screen is 1024\*915 pixels. All frame rates are measured in frames per second. The frame rates for normal non-relativistic rendering, for relativistic rendering with no extra threads for the relativistic transformations, and for relativistic rendering with multithreading are compared. The last row shows the number of the additional threads needed for the multithreaded relativistic transformations. The test scenes are depicted in Figure 5 (scene 1), Figure 6 (scene 2), and Figure 7 (scene 3).

## 9. Conclusion and Future Work

In this paper an immersive virtual environment for special relativity has been presented. This virtual environment allows an intuitive approach to special relativity.

The *relativistic-vehicle-control* metaphor has been introduced as an interaction technique for the relativistic virtual

environment. The *relativistic-vehicle-control* metaphor is a physically based camera control technique that supports both passive and active locomotion. The treatment of the relativistic observer has been described. Furthermore, relativistic tracking of the position and velocity of the user has been introduced.

The geometric approach to relativistic polygon rendering allows the integration of the accelerated observer. Polygon rendering is supported by contemporary computer graphics hardware and is suitable for rendering in a virtual environment. It has been shown how the rendering pipeline can be extended to perform the relativistic transformations concurrently to the other rendering processes. This way, the same frame rates can be achieved for relativistic rendering as for non-relativistic rendering.

Future work will cover several aspects. To improve navigation and enrich the physical understanding, a second, exocentric view on the virtual world could be helpful. Similar to the world-in-miniature (WIM) technique, an outside view, which will be displayed in addition to the egocentric view, will be investigated. In addition, space exclusion, i.e. collision detection, could serve as a visual feedback technique to improve navigation. More sophisticated filters for velocity tracking will be examined to reduce the latency in the direct velocity control.

Furthermore, relativistic effects on the illumination by the Doppler and searchlight effects will be included.

Finally, improvements of the rendering performance will be pursued. Techniques known from standard non-relativistic rendering will be adopted, such as occlusion culling and adaptive multi-resolution methods.

### Acknowledgements

The author would like to thank the numerous CO-VISE/COVER developers, in particular Paul Benölken, Daniela Rainer, and Uwe Wössner. Special thanks to Michael Doggett, Stefan Gumhold, Bettina Salzer, and Nikolaus Weiskopf for their help in preparing this paper. The test scene shown in Figures 6 and 8 through 10 was originally designed by Ralf Sonntag and PYTHA Lab.

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) and is part of the project D4 within the Sonderforschungsbereich 382.

### References

1. M. L. Boas. Apparent Shape of Large Objects at Relativistic Speeds. *American Journal of Physics*, 29(5):283–286, May 1961.
2. M.-C. Chang, F. Lai, and W.-C. Chen. Image Shading Taking into Account Relativistic Effects. *ACM Transactions on Graphics*, 15(4):265–300, Oct. 1996.
3. D. Chapman and C. Ware. Manipulating the Future: Predictor Based Feedback for Velocity Control in Virtual Environment Navigation. *Computer Graphics (Proceedings of the 1992 Symposium on Interactive 3D Graphics)*, 25(2):63–66, Mar. 1992.
4. C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE. In *SIGGRAPH 93 Conference Proceedings*, pages 135–142, Aug. 1993.
5. R. Earnshaw, M. Gigante, and H. Jones, editors. *Virtual Reality Systems*. Academic Press, London, 1993.
6. G. Eckel and S. Hiatt. IRIS Performer Programmer's Guide. Silicon Graphics Document Number 007-1680-040, 1998.
7. A. Einstein. Zur Elektrodynamik bewegter Körper. *Annalen der Physik*, 17:891–921, 1905. In German.
8. W. Gekelman, J. Maggs, and L. Xu. Real-Time Relativity. *Computers in Physics*, pages 372–385, July/Aug. 1991.
9. C. Hand. Survey of 3D Interaction Techniques. *Computer Graphics Forum*, 16(5):269–281, Dec. 1997.
10. *Virtual Environments Lab*. In High-Performance Computing-Center Stuttgart Web Site: <http://www.hlrs.de/structure/organisation/vis/velab>.
11. K. Hinckley, R. Pausch, J. C. Goble, and N. F. Kassell. A Survey of Design Issues in Spatial Input. In *Symposium on User Interface Software and Technology '94*, pages 213–222, Nov. 1994.
12. R. Holloway and A. Lastra. Virtual Environments: A Survey of the Technology. Technical Report TR93-033, Department of Computer Science, University of North Carolina at Chapel Hill, Sept. 1993.
13. P.-K. Hsiung and R. H. P. Dunn. Visualizing Relativistic Effects in Spacetime. In *Proceedings of Supercomputing '89 Conference*, pages 597–606, Nov. 1989.
14. P.-K. Hsiung, R. H. Thibadeau, C. B. Cox, and R. H. P. Dunn. Time Dilation Visualization in Relativity. In *Proceedings of the Supercomputing '90 Conference*, pages 835–844, 1990.
15. P.-K. Hsiung, R. H. Thibadeau, C. B. Cox, R. H. P. Dunn, M. Wu, and P. A. Olbrich. Wide-Band Relativistic Doppler Effect Visualization. In *Proceedings of the Visualization '90 Conference*, pages 83–92, Oct. 1990.
16. P.-K. Hsiung, R. H. Thibadeau, and M. Wu. T-Buffer: Fast Visualization of Relativistic Effects in Spacetime. *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, 24(2):83–88, Mar. 1990.
17. R. H. Jacoby and S. R. Ellis. Using Virtual Menus in a Virtual Environment. In *SIGGRAPH '93 Implementing*



- Virtual Reality course notes*, pages 7–1 to 7–8, Aug. 1993.
18. A. Lampa. Wie erscheint nach der Relativitätstheorie ein bewegter Stab einem ruhenden Beobachter? *Zeitschrift für Physik*, 27:138–148, Aug./Sept. 1924. In German.
  19. M. R. Mine. Virtual Environment Interaction Techniques. Technical Report TR95-018, Department of Computer Science, University of North Carolina at Chapel Hill, May 1995.
  20. C. W. Misner, K. S. Thorne, and J. A. Wheeler. *Gravitation*. Freeman and Company, New York, 1973.
  21. C. Møller. *The Theory of Relativity*. Clarendon Press, Oxford, second edition, 1972.
  22. R. Penrose. The Apparent Shape of a Relativistically Moving Sphere. *Proceedings of the Cambridge Philosophical Society*, 55:137–139, 1959.
  23. D. Rantau and U. Lang. A Scalable Virtual Environment for Large Scale Scientific Data Analysis. *Future Generation Computer Systems*, 14:215–222, 1998.
  24. W. Rindler. *Introduction to Special Relativity*. Clarendon Press, Oxford, second edition, 1991.
  25. W. Robinett and R. Holloway. Implementation of Flying, Scaling, and Grabbing in Virtual Worlds. *Computer Graphics (Proceedings of the 1992 Symposium on Interactive 3D Graphics)*, 25(2):189–192, Mar. 1992.
  26. J. Rohlf and J. Helman. IRIS Performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics. In *SIGGRAPH 84 Conference Proceedings*, pages 381–395, July 1994.
  27. G. D. Scott and H. J. van Driel. Geometrical Appearances at Relativistic Speeds. *American Journal of Physics*, 38(8):971–977, Aug. 1970.
  28. G. D. Scott and M. R. Viner. The Geometrical Appearance of Large Objects Moving at Relativistic Speeds. *American Journal of Physics*, 33(7):534–536, July 1965.
  29. R. Stoakley, M. J. Conway, and R. Pausch. Virtual Reality on a WIM: Interactive Worlds in Miniature. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'95)*, pages 265–272, May 1995.
  30. J. Terrell. Invisibility of the Lorentz contraction. *Physical Review*, 116(4):1041–1045, Nov. 1959.
  31. R. Turner, F. Balaguer, E. Gobbetti, and D. Thalmann. Physically-Based Interactive Camera Motion Control Using 3D Input Devices. In N. M. Patrikalakis, editor, *Scientific Visualization of Physical Phenomena (Proceedings of CG International '91)*, pages 135–145. Springer, 1991.
  32. C. Ware and S. Osborne. Exploration and Virtual Camera Control in Virtual Three Dimensional Environments. *Computer Graphics (Proceedings of the 1990 Symposium on Interactive 3D Graphics)*, 24(2):175–183, Mar. 1990.
  33. V. F. Weiskopf. The Visual Appearance of Rapidly Moving Objects. *Physics Today*, 13(9):24–27, Sept. 1960.

---

Submitted to:  
Eurographics '99

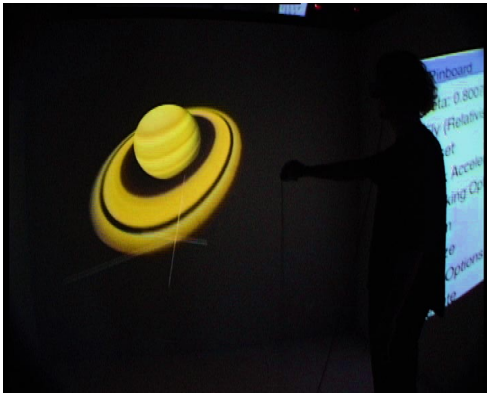


Figure 5: Saturn.

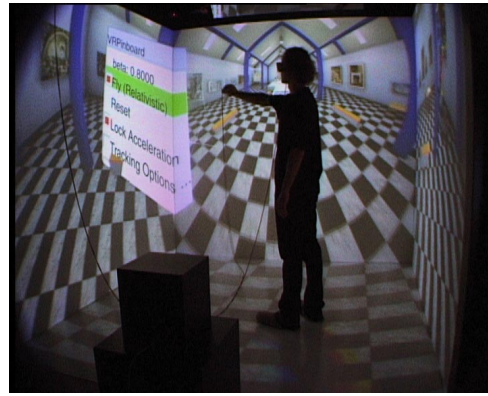


Figure 6: Gallery scene with standard menu.



Figure 7: Simple box-shaped test scene with tracking options menu.

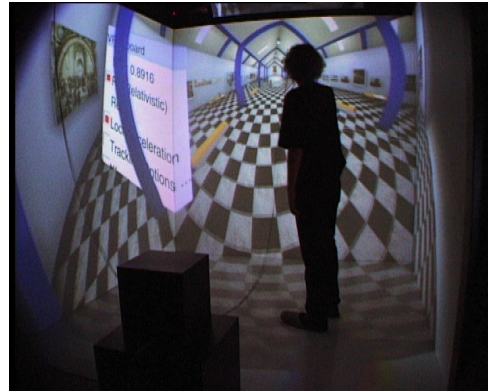


Figure 8: Acceleration of the vehicle. First snapshot with velocity  $\beta = 0.892$ .

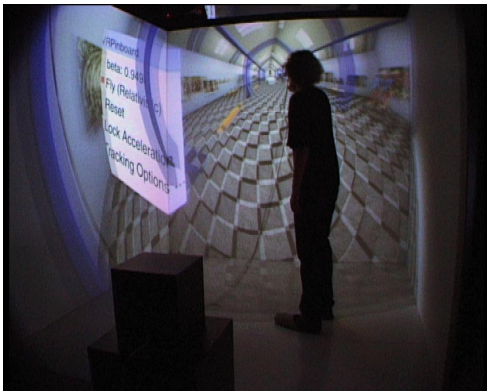


Figure 9: Acceleration of the vehicle. Second snapshot with  $\beta = 0.949$ .

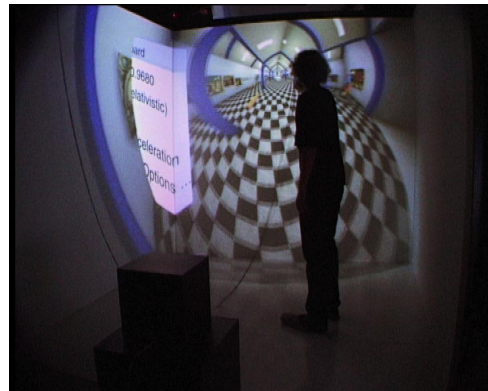


Figure 10: Acceleration of the vehicle. Third snapshot with  $\beta = 0.968$ .