

An Immersive Virtual Environment for Special Relativity

Daniel Weiskopf

Institute for Astronomy and Astrophysics
University of Tübingen
Auf der Morgenstelle 10, D-72076 Tübingen, Germany
weiskopf@tat.physik.uni-tuebingen.de

ABSTRACT

In this paper, an immersive virtual environment is presented 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—a physically based camera control technique—is introduced for navigating at high velocities. Acceleration of the relativistic observer is 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 multiprocessor and multipipe systems for fast rendering and the same frame rates can be achieved for relativistic visualization as for non-relativistic rendering.

Keywords: special relativity, visualization, immersive virtual environment

1 INTRODUCTION

Einstein's Special Theory of Relativity is widely regarded as a difficult and hardly comprehensible theory, even today—more than ninety years after its publication in 1905 [Einst05]. 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 ego-centric frame of reference is adopted. This will enable us to experience special relativity right in front of our eyes, including phenomena such as Lorentz con-

traction, 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.

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 physics-based *relativistic-vehicle-control* metaphor is introduced for navigating at high velocities, consist-

ing 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. In this context, the calculation of the trajectory of an accelerated object is indispensable. It will be shown how this calculation can be implemented. 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.

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 [Lampa24] 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 [Penro59] and Terrell [Terre59]. Later, more detailed descriptions of the geometrical appearance of fast moving objects were given by Weisskopf [Weiss60], Boas [Boas61], Scott and Viner [Scott65], and Scott and van Driel [Scott70].

Direct visualization of special relativity on a computer was first presented by Hsiung and Dunn [Hsiun89]. 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. [Hsiun90a] investigated relativistic ray tracing in more detail and included the visualization of the Doppler effect. Hsiung et al. [Hsiun90b] introduced the *time-buffer* for fast visualization of relativistic effects. The *time-buffer* technique allows for relativistic polygon rendering. It makes use of computer graphics hardware and achieves interactive frame rates. Gekelman et al. [Gekel91] described the polygon rendering of a relativistically moving cube. They mentioned the trajectory of an accelerated cube. However, they did not describe in detail how to treat acceleration in special relativity. Chang et al. [Chang96] investigated the polygon rendering approach in detail and gave a comprehensive presentation, but they neglected acceleration. In our previous work [Rau98], we considered acceleration in the visualization of special relativity.

A lot of research has been conducted on issues related to virtual environments. An overview can be found, for example, in [Hollo93; Earns93; Vince98].

Cruz-Neira et al. [Cruz-93] presented the CAVE, a surround-screen, projection-based, immersive virtual environment, which is also a model for the virtual environment of this paper. The *flying-vehicle metaphor* by Ware and Osborne [Ware90] will serve as the basis for the development of the *relativistic-vehicle-control metaphor*. Turner et al. [Turne91] introduced the physically based *virtual-camera metaphor* for controlling the camera motion. They described an abstract physical model of a camera object whose trajectory is determined by classical, Newtonian mechanics. Some of these ideas will be adopted for the relativistic situation in Sects. 5 and 6.

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 [Misne73; Mølle72; Rindl91]. 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

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

where c is the speed of light and $\mu \in \{0, 1, 2, 3\}$. 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. Fig. 1 shows a geometric approach to the Poincaré transformation in the form of the Minkowski diagram, a spacetime diagram without the coordinates y and z . The translation of the origins of the two coordinate systems is explicitly marked. The different velocities of the reference frames cause the relative rotation of the axes of the coordinate systems. The projection of the *Event* onto the axes yields its coordinates (t_e, x_e) and (t'_e, x'_e) 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. [Misne73, p.69], is described by the symmetric 4×4 matrix $\Lambda^{\mu'}_{\nu}$, whose

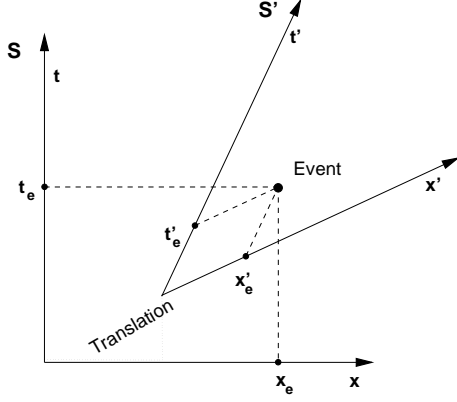


Figure 1: Minkowski diagram showing two frames of reference, S and S' respectively.

components are

$$\begin{aligned}\Lambda^{0'}_0 &= \gamma, \\ \Lambda^{0'}_j &= \Lambda^{j'}_0 = -\beta\gamma n^j, \\ \Lambda^{j'}_k &= \Lambda^{k'}_j = (\gamma - 1)n^j n^k + \delta^{jk},\end{aligned}$$

where $\vec{n} = (n^1, n^2, n^3) = (n^x, n^y, n^z)$ is the normalized direction of motion, δ^{jk} is the Kronecker symbol, and $j, k \in \{1, 2, 3\}$. The velocity parameters γ and β 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 RENDERING

Relativistic polygon rendering makes use of graphics hardware and thus interactive frame rates can be achieved. The other known rendering technique, relativistic ray-tracing, has high computational costs and is not suitable for a virtual environment. A more detailed comparison of both methods can be found in [Rau98].

In the following, a geometric approach to relativistic polygon rendering will be presented, which allows for the treatment of the accelerated observer. An ego-centric frame of reference is adopted for relativistic rendering. The scenario is as follows. The user, i.e. the observer, is freely moving through a virtual world. The objects of the scene are at rest relative to each other.

Let us investigate the generation of a snapshot taken by the relativistic observer. At the time of image pro-

duction, the observer is moving with velocity $\vec{v} = (v_x, v_y, v_z)$ with respect to reference frame S_{world} . S_{world} is the frame of reference of the outside world. Frame S_{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. The acceleration of the observer does not affect rendering. The accelerated observer and the co-moving inertial observer take exactly the same snapshot.

First, only a single point light-source which is at rest in S_{world} is considered. In Fig. 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_{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_{world} to S_{obs} .

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

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

where $x_e^{\mu'}$ denote the coordinates of E and $x_o^{\mu'}$ the coordinates of O in S_{world} . With the Poincaré transformation described in the previous section, the coordinates of the emission event in S_{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

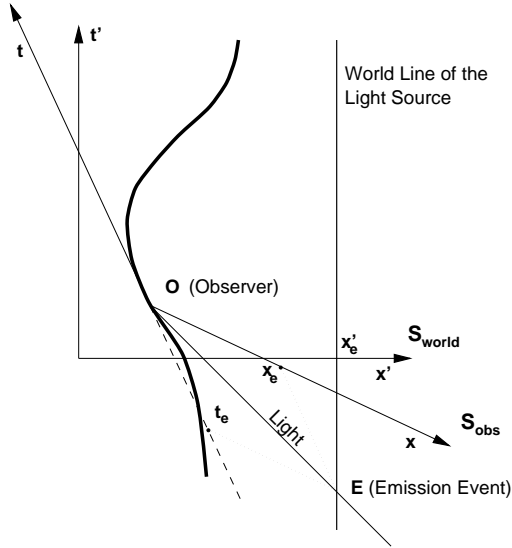


Figure 2: Minkowski diagram showing the world line of the accelerated observer with the co-moving reference frame S_{obs} . The light source is at rest in reference frame S_{world} . The coordinates (t_e, x_e) of the emission event E are determined by projection onto the axes of S_{obs} .

the spatial distance of an emission event is directly related to its temporal distance, cf. Eq. 2. In frame S_{obs} , the vertices associated with “newer” events will correctly hide vertices associated with “older” ones. Due to the non-linear transformation of the vertices artifacts are induced by the triangle mesh representation. These artifacts, which appear mainly at the boundaries of the objects, can be reduced by fine triangulation of the objects. When the geometrical transformation of a vertex is performed relativistic effects on the illumination could also be included, such as the transformation of wavelength and radiance. 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 τ 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

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

and the components of the four-velocity are obtained by

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

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

$$a^\mu = \frac{du^\mu}{d\tau} = \frac{d^2 x^\mu}{d\tau^2}. \quad (4)$$

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_{world} by the associated Lorentz transformation, cf. Sect. 3. The equations of motion (4) resemble those of classical physics. The quantities a^μ , u^μ , x^μ , and β are functions of τ . The Lorentz transformation of the four-acceleration into frame S_{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 the Runge-Kutta 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 space-time parameterized by the proper time is obtained.

Note that special relativity is capable of describing acceleration 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. The development of this metaphor is guided by the properties and requirements of an immersive virtual environment. Furthermore, the metaphor is directly based on and motivated by relativistic physics because the aim of the

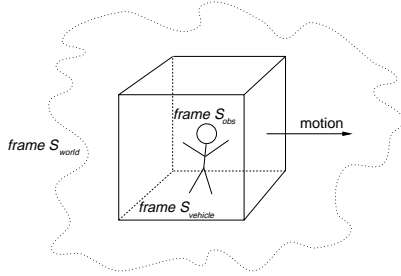


Figure 3: Sketch of the virtual environment.

virtual environment exactly is to visualize this part of modern physics. In this section, the elements of the metaphor are presented.

The following scenario is proposed. The user resides in a fast vehicle which enables him or her to explore a static virtual world. The motion of the vehicle is controlled by user input. The spatial input device, a three-dimensional mouse, 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 Fig. 3.

This scenario introduces an additional frame of reference. In the following, the frame of the objects is denoted as S_{world} , the frame of the vehicle as S_{vehicle} , and the frame of the user as S_{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_{vehicle} . The acceleration due to user input affects the trajectory of this reference point. The treatment of the accelerated observer in Sect. 5 is adopted for the reference point, and the trajectory of the reference point can be computed accordingly. It is assumed that the coordinate system S_{vehicle} is a co-moving frame with respect to the reference point. Thus, for every moment in time, the frame S_{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 calculated by taking the difference of consecutive positions. With Eqs. (1) and (3), the

respective four-position and four-velocity can be determined in frame S_{vehicle} . The Lorentz transformation from S_{vehicle} to S_{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, cf. Sect. 4. This way, the images are generated as seen by the user.

In order to notice relativistic effects on the apparent geometry 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.

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. The user's motion and varying velocity has a high impact on image generation, which provides direct visual feedback. 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.

More generally, the *relativistic-vehicle-control* metaphor can be used in any relativistic simulation where the motion of the virtual camera is superimposed onto another, underlying motion.

7 IMPLEMENTATION

The implementation of the relativistic virtual world runs in the *CUBE* [Hig99], which is similar to the *CAVE* [Cruz-93] immersive virtual environment. The *CUBE* is a classical four side back-projection system, consisting of three walls and a floor. The implementation is an extension of the non-relativistic *COVER* renderer [Rantz98]. Rendering is performed using *IRIS Performer* [Rohlf94]. The geometry nodes of *Performer* are extended and made aware of special relativity by subclassing. The derived geometry nodes

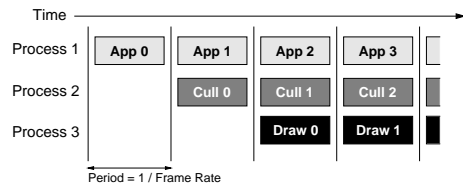


Figure 4: Multiprocessing model of IRIS Performer. *App n* is the application process for the *n*th frame, *Cull n* and *Draw n* are the respective culling and draw processes. The vertical lines are frame boundaries, which are used to synchronize the various stages in the rendering pipeline.

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. Fig. 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 Fig. 4, the application, culling, and draw processes work on three different frames at the same time. 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 transforma-

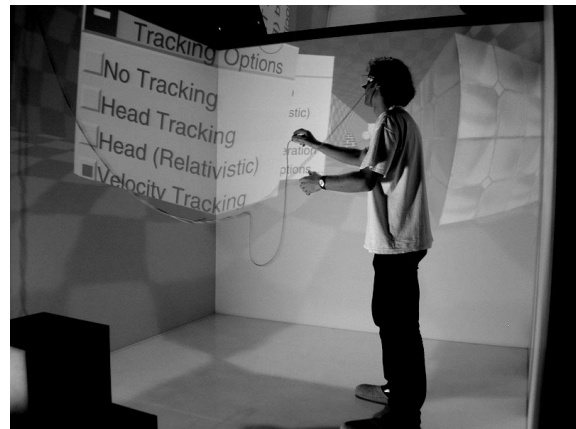


Figure 5: Simple box-shaped test scene with menu.

tion 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 by domain-decomposition 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

Figs. 5 through 8 show snapshots taken during typical simulations. Very interesting effects occur when the velocity is being changed due to acceleration. Objects which are not too close to the user seem to move away when he or she is approaching the objects with increasing speed. Conversely, the objects are apparently coming closer when the speed is being decreased. The situation is depicted in Figs. 6 through 8. 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 be-



Figure 6: Acceleration of the vehicle. First snapshot with velocity $\beta = 0.8$.



Figure 7: Acceleration of the vehicle. Second snapshot with $\beta = 0.9188$.



Figure 8: Acceleration of the vehicle. Third snapshot with $\beta = 0.9832$.

come familiar with the relativistic interaction techniques. Direct locomotion was reported to be a very intuitive means of interaction.

Performance measurements on an Onyx2 system with R10000/195MHz CPUs are documented in Table 1. Two InfiniteReality pipes are used for rendering onto

four screens in stereo. The size of a single screen is 1024*915 pixels. The third row of the table shows the frame rates for normal non-relativistic rendering. The fourth row gives the frame rates for the hitherto used relativistic polygon rendering (no multithreading). The fifth row documents the rendering performance for the new, multithreaded implementation. The last row shows the number of the additional threads needed for the multithreaded relativistic transformations. In the multiprocessor mode, the same frame rates are achieved for relativistic rendering as for non-relativistic rendering. The frame rates are upheld during the whole simulation.

The accompanying video (also available on our web page [Weisk99]) gives a good impression of the relativistic virtual environment. The first sequence shows a simulation with Saturn. Acceleration in all three dimensions is used, there are no constraints on the movement of the viewpoint. The second scene is a box-shaped room which contains simple objects such as a cube, sphere, torus etc., see Fig. 5. Textures are attached to all objects to visualize relativistic effects on the surface. Acceleration is restricted to the horizontal plane. The focus is on how the user can control the movement of the vehicle. No tracking is used. The third sequence also uses the box-shaped room. Here, the difference between tracking of position only and tracking of both position and velocity is presented. The last simulation takes place in a virtual gallery, cf. Fig. 6.

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 in-

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

Table 1: Rendering performance. The test scenes are depicted in Fig. 5 (scene 1), and Fig. 6 (scene 2).

troduced 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 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. Space exclusion, i.e. collision detection, could serve as a visual feedback technique to improve navigation. Furthermore, relativistic effects on the illumination will be included, for example the Doppler effect. Finally, improvements of the rendering performance will be pursued. Techniques known from standard non-relativistic rendering will be adopted, such as occlusion culling and multi-resolution methods. Moreover, an adaptive subdivision scheme could reduce the artifacts which are caused by the non-linear, relativistic transformation of the vertices.

ACKNOWLEDGEMENTS

The author would like to thank the numerous COVISE/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 Figs. 6 through 8 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

- [Boas61] M. L. Boas. Apparent Shape of Large Objects at Relativistic Speeds. *American Journal of Physics*, 29(5):283–286, 1961.
- [Chang96] M.-C. Chang, F. Lai, and W.-C. Chen. Image Shading Taking into Account Relativistic Effects. *ACM Transactions on Graphics*, 15(4):265–300, 1996.
- [Cruz-93] 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, 1993.

- [Earns93] R. Earnshaw, M. Gigante, and H. Jones, editors. *Virtual Reality Systems*. Academic Press, London, 1993.
- [Einst05] A. Einstein. Zur Elektrodynamik bewegter Körper. *Annalen der Physik*, 17:891–921, 1905. In German.
- [Gekel91] W. Gekelman, J. Maggs, and L. Xu. Real-Time Relativity. *Computers in Physics*, pages 372–385, 1991.
- [Hig99] *Virtual Environments Lab*. In High-Performance Computing-Center Stuttgart Web Site: <http://www.hlrs.de/structure/organisation/vis/velab>, 1999.
- [Hollo93] 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, 1993.
- [Hsiun89] P.-K. Hsiung and R. H. P. Dunn. Visualizing Relativistic Effects in Spacetime. In *Supercomputing '89 Proceedings*, pages 597–606, 1989.
- [Hsiun90a] 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 *Visualization '90 Proceedings*, pages 83–92, 1990.
- [Hsiun90b] P.-K. Hsiung, R. H. Thibadeau, and M. Wu. T-Buffer: Fast Visualization of Relativistic Effects in Spacetime. *Computer Graphics*, 24(2):83–88, 1990.
- [Lampa24] A. Lampa. Wie erscheint nach der Relativitätstheorie ein bewegter Stab einem ruhenden Beobachter? *Zeitschrift für Physik*, 27:138–148, 1924. In German.
- [Misne73] C. W. Misner, K. S. Thorne, and J. A. Wheeler. *Gravitation*. Freeman and Company, New York, 1973.
- [Mølle72] C. Møller. *The Theory of Relativity*. Clarendon Press, Oxford, second edition, 1972.
- [Penro59] R. Penrose. The Apparent Shape of a Relativistically Moving Sphere. *Proceedings of the Cambridge Philosophical Society*, 55:137–139, 1959.
- [Rantz98] D. Rantzau and U. Lang. A Scalable Virtual Environment for Large Scale Scientific Data Analysis. *Future Generation Computer Systems*, 14:215–222, 1998.
- [Rau98] R. T. Rau, D. Weiskopf, and H. Ruder. Special Relativity in Virtual Reality. In H.-C. Hege and K. Polthier, editors, *Mathematical Visualization*, pages 269–279. Springer, Heidelberg, 1998.
- [Rindl91] W. Rindler. *Introduction to Special Relativity*. Clarendon Press, Oxford, second edition, 1991.
- [Rohlf94] J. Rohlf and J. Helman. IRIS Performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics. In *SIGGRAPH 94 Conference Proceedings*, pages 381–395, 1994.
- [Scott65] 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, 1965.
- [Scott70] G. D. Scott and H. J. van Driel. Geometrical Appearances at Relativistic Speeds. *American Journal of Physics*, 38(8):971–977, 1970.
- [Terre59] J. Terrell. Invisibility of the Lorentz contraction. *Physical Review*, 116(4):1041–1045, 1959.
- [Turne91] 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*, pages 135–145. Springer, 1991.
- [Vince98] J. Vince. *Essential Virtual Reality Fast*. Springer, London, 1998.
- [Ware90] C. Ware and S. Osborne. Exploration and Virtual Camera Control in Virtual Three Dimensional Environments. *Computer Graphics*, 24(2):175–183, 1990.
- [Weisk99] D. Weiskopf. *SFB 382 – Project D4*. Web Site: <http://www.tat.physik.uni-tuebingen.de/~weiskopf>, 1999.
- [Weiss60] V. F. Weisskopf. The Visual Appearance of Rapidly Moving Objects. *Physics Today*, 13(9):24–27, 1960.