

# Interactive Visualization of Relativistic Effects with the Hardware Acceleration

Radoslaw Mantiuk, Karina Murawko-Wisniewska, and Dorota Zdrojewska

Technical University of Szczecin, Zolnierska 49, 71-210 Szczecin, Poland,  
phone +48(91)4495661,  
rmantiuk@wi.ps.pl, <http://zgk.wi.ps.pl>

**Abstract.** This paper describes an interactive rendering technique for the special relativistic visualization of fast moving objects. The method takes into consideration relativistic aberration of light and Doppler effect. Programmable graphics hardware is used to achieve interactive visualization. Standard texture mapping pipeline is modified on per-pixels basis and relativistic effects are rendered. The basis of optical relativistic phenomena and a survey of relativistic real time rendering techniques are also presented in the paper.

**Keywords:** rendering of relativistic effects, vertex and pixel shaders, interactive image synthesis, computer graphics.

## 1 Introduction

Mankind is limited to very low velocities compared to the speed of light. Therefore, computer simulations are the only means of seeing the world from the perspective of fast moving vehicle. Properties of space, time, and light in relativistic physics are totally different from those in classical, Newtonian physics. They are contrary to human experience. Until now, people could only imagine how such effects like relativistic aberration, Doppler shift, Terrell rotation look like. Computer visualization of relativistic effects could help the intuition of people and make the relativistic phenomena more understandable. Interactive visualization of relativistic phenomena brings additional value. The observer can change speed or direction of motion and see the results immediately.

Real time rendering methods need to be used to achieve interactive visualization of relativistic effects. Moreover the computation should be aided by the 3D graphics hardware to generate smooth animation. This paper describes such interactive rendering technique. The method is based on environment texture mapping technique. Standard texture mapping pipeline is modified on per-pixels basis and relativistic effects are rendered. Shader programs, running on graphics board, are used to speed up calculations.

The next section describes the previous works on interactive visualization of relativistic effects. In Sect. 3, the basis of relativistic physic are presented. Section 4 gives details of our relativistic rendering algorithm based on hardware acceleration. In Sect. 5 implementation is described and results are shown. The paper ends with conclusion and an outlook on future work.

## 2 Previous Works

The issue of visual perception in special relativity was ignored for a long time. Hsiung and Dunn [1] were the first to use visualization techniques for image shading of fast moving objects. They proposed the extension of ray tracing for generation of relativistic effects. The first real time approach for relativistic rendering was presented by Hsiung et al. in [2]. In the last few years two different approaches, which allow rendering relativistic phenomena in real time were discussed.

The first one extends the normal rendering pipeline by an additional transformation of vertices and is an extension of Hsiung T-buffer. This approach, called polygon rendering technique, was investigated by Gekelman et al. in [3]. In [4] and [5] Gekelman's method was generalized to any object built with polygon mesh. The goal of polygon rendering technique is to create a new copies of 3D objects by relativistic transformation of existing objects. Objects' vertices are considered as point lights. The light from these light sources runs to observer not along the straight lines and causes relativistic deformations. The main drawback of polygon rendering technique is the necessity of polygon tessellation. Relativistic transformation causes bending of polygon edges and polygons need to be tessellated to achieve acceptable image quality.

The second method of relativistic effects rendering is based on texture mapping. This method was proposed by Weiskopf in [6] and developed in [14]. The main goal of the method is to use extended texture mapping technique to transform standard image of the environment to the coordinate system of very fast moving observer. Objects are deformed by the relativistic aberration of light and finite light speed effects. These deformations can be achieved by mapping of an environment texture on the sphere surrounding the observer. During the mapping the relativistic calculation are done and colour and position of texels in environment map are changed. Interior of the sphere with mapped texture is projected on the plane to achieved final image with relativistic phenomena. In [7] above method was extended to allow visualization of Doppler effect (colour shift) and radiance fluctuation (brightness shift). The texture mapping technique does not require any changes in 3D scene structure, only pixel transformations (changes of pixel colour and position) are done. The main drawback of the technique is limitation to the walkthrough environments (the environment textures need to be recalculated when the position of objects or lights is changed). Also the achieved rendering speed (5-7 fps) doesn't allow for smooth animation.

The relativistic phenomena visualization method presented in our paper is an extension of Weiskopf method. We propose the modification of texture mapping technique and the usage of reprogrammable hardware to speed up rendering.

## 3 Relativistic Transformations

Take two coordinate systems  $S$  and  $S'$  moving relatively to each other with velocity  $v$ . When velocity  $v$  is close to the speed of light an observer in system  $S$

experiences events differently than the observer in system  $S'$ . These differences, explained by Einstein in his Theory of Special Relativity, can be computed using Lorentz transformation [9]. Lorentz transformation of point  $(P, t_p)$  measured in system  $S$  to system  $S'$  is given by

$$P' = P + \left[ \frac{(\gamma - 1)}{\beta_l^2} * P \cdot \beta + \gamma * c * t_p \right] * \beta, \quad t'_p = \gamma * t_p + \gamma * \frac{\beta \cdot P}{c} \quad (1)$$

where  $c$  - speed of light,  $\beta = v/c$ ,  $\beta_l = |\beta|$ ,  $\gamma = \frac{1}{\sqrt{1-\beta_l}}$ .

The direction of a light beam depends on the velocity of the light source relative to the observer, this effect is called the optic aberration of light (relativistic aberration). Consider a particle moving with velocity  $v$  (along the x-axis) through the  $S$  coordinate system [13], it emits the light beam at an angle  $f'$  to the x'-axis (measured in  $S'$  system, in which particle is at rest), in the  $S$  system the light beam is pushed to the new angle  $f$ . The relation between the angles (in spherical coordinates  $(\Theta, \phi)$ ) is given by

$$\cos \Theta' = \frac{\cos \Theta - \beta}{1 - \beta * \cos \Theta}, \quad \phi' = \phi. \quad (2)$$

The shift of light wave's frequency ( $\nu$ ) during the relative motion of a light source and an observer is called the Doppler effect [11]. When the source and the observer are approaching each other colors are shifted toward the high end of the visible spectrum (the blue shift). The Doppler shift of light waves in vacuum which arrive with an angle  $f$  with respect to direction of travel is:

$$\nu' = \nu * \gamma * (1 - R_d \cdot \beta). \quad (3)$$

More detailed description of special relativity can be found in [9][12][10].

## 4 Real Time Visualization of Relativistic Aberration and Doppler Effects

The algorithm for relativistic image synthesis is designed upon the Weiskopf's texture-based relativistic rendering method [6]. Hardware acceleration is accomplished by modifying the method in order to use the vertex and pixel shaders (an extension of modern graphic cards).

In preprocessing a set of images representing the environment must be generated. The single image is not sufficient, because the aberration of light changes the field of view (it could widen the field of view beyond the image). The images allow looking from the observation point in any direction, similar as in environment mapping techniques (we use cube mapping technique). The new set of textures must be prepared when the scene objects' or lights' position change and in other specific situations (when the information in textures is not enough for calculations).

In the first step of algorithm the texture coordinates are transformed with the relativistic aberration equation, in our system it is achieved with vertex shader. Then the prepared textures are mapped on the sphere, which surrounds the observer. The last step is the Doppler effect calculation, the change of textures' pixel colour is done by pixel shader. For calculating of colour shift in RGB the algorithm from ray tracer Backlight [8] was used. It allows calculating every component separately by using a simplified modeling of spectrum. The RGB components are shifted according to equation 3.

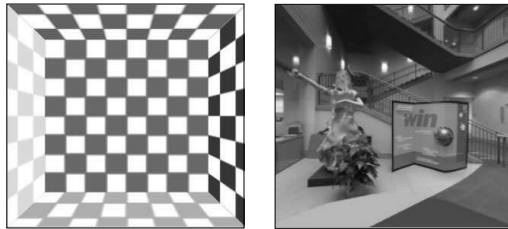
There is also possibility for interactive change of speed. The elements of the scene (scene objects and light sources) are at rest relative to each other. The relative motion of the scene and the observer is limited to the direction along the z-axis, but can be expanded by using additional rotations of the coordinate system.

## 5 Implementation and Results

Our implementation of the relativistic visualization algorithm is based on hardware acceleration in form of vertex and pixel shaders. The algorithm is implemented in Direct3D 9.0 with NVIDIA's Cg language.

The most intuitive and fastest method of projecting the environment onto the sphere is the use of cube texture. Its' faces represent different parts of the scene viewed from the six orientations of the camera positioned at the coordinate system origin.

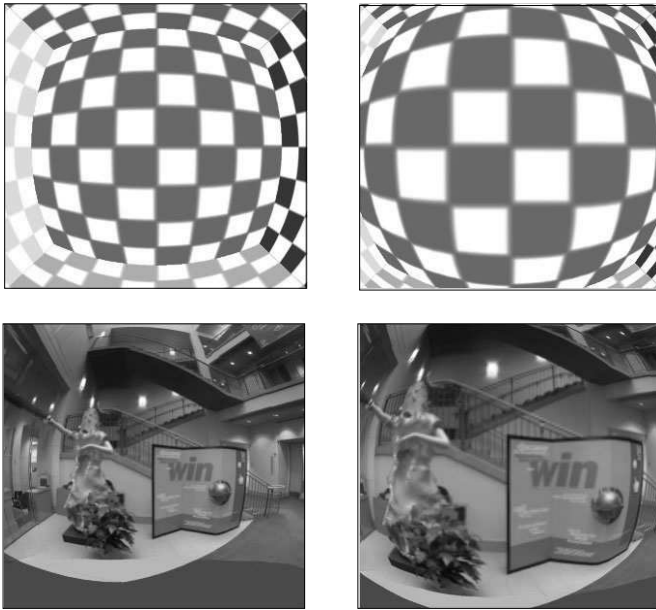
In our implementation, cube mapping and aberration of light are calculated in the vertex shader program (for each vertex of the sphere surrounding the observer). All essential data is gathered in vertex shaders registers. Vertex position and normal are passed from the vertex buffer into input registers (POSITION, NORMAL).  $\beta$  coefficient, necessary to calculate the light aberration, is placed in one of the constant registers. Vertices coordinates are transformed to spherical coordinates and relativistic light aberration is calculated (according to equation 2). The results are passed in output registers (POSITION, COLOR0 and TEXCOORD0).



**Fig. 1.** Test scenes: left - cube with the checkered textures, right - inside the hall (the texture from DirectX samples)

Pixel shader does not have POSITION register, so we use COLOR0 register to store coordinates of each vertex of the sphere (calculated by vertex shader). These coordinates are used in pixel shader to compute Doppler effect. Pixel shader program computes final environment texture coordinates, and then gets colour of the corresponding texel. Afterwards the Doppler effect is applied - colour of each pixel is changed with the equation 3, and passed in the output COLOR register. After colour computation, primitives are drawn on the screen.

Figure 1 presents two example scenes to which relativistic effects are going to be applied. The camera is motionless.



**Fig. 2.** Relativistic light aberration: top/left - velocity  $v=0.6c$ , top/right - velocity  $v=0.9c$ , bottom/left - velocity  $v=0.6c$ , bottom/right - velocity  $v=0.9c$

Relativistic light aberration effect applied to example scenes can be seen in the fig.2. The camera is moving into the scene with  $0.6c$  or  $0.9c$  velocity.

As seen in the pictures above, change of the movement velocity causes impression of disturbance in the scene geometry. When the camera is moving with a large speed, close to the speed of light, straight lines seem to be curved.

The Doppler effect causes shift of colour into high frequencies when the observer is moving into the scene. It appears in change of colour into blue.

The rendering speeds 30 fps without Doppler effect and 15 fps with Doppler effect were achieved (for images of resolution  $320 \times 320$  pixels). The colour shift algorithm plays a dominant role for the rendering performance.

## 6 Conclusion and Future Work

In this paper a hardware aided texture mapping approach to relativistic visualization has been presented. The physical basis is the relativistic aberration of light and Doppler effect. Interactive visualization of relativistic phenomena was achieved. The most important parts of the rendering algorithm are calculated in shaders, programs activated in hardware. Our method doesn't require any special hardware, it can be activated in a standard graphics card which supports vertex and pixel shaders.

In future work, the implementation of searchlight effect is considered. Also we plan to increase the performance of the rendering. Faster algorithms of colour shifting and new features of shaders will be explored. The implementation of hybrid method of rendering which uses texture mapping and Image Based Rendering approach together is also considered. It would allow for real time visualization of such relativistic effects as Terrell rotation.

## References

1. Hsiung P.K., Dunn R.H.P.: Visualizing relativistic effects in spacetime. In Proceedings of Supercomputing '89 Conference, pp. 597-606, 1989.
2. Hsiung P.K., Thibadeau R.H., Wu M.: T-buffer: Fast visualization of relativistic effects in spacetime. *Computer Graphics*, 24(2), pp. 83-88, March 1990.
3. Gekelman W., Maggs J., Xu L.: Real-time relativity. *Computers in Physics*, pp. 372-385, 1991.
4. Rau R.T., Weiskopf D., Ruder H.: Special Relativity in Virtual Reality. *Mathematical Visualization*, H.-C. Hege, K. Polthier, Springer, pp. 269-279, 1998.
5. Weiskopf Daniel: An Immersive Virtual Environment for Special Relativity. WSCG 2000 Conference, pp. 337-344, 2000.
6. Weiskopf Daniel: A Texture Mapping Approach for the Visualization of Special Relativity. *IEEE Visualization '99 Late Breaking Hot Topics Proceedings*, A. Vars-hney, C.M. Wittenbrink, H. Hagen, ACM Press, pp. 41-44, 1999.
7. Weiskopf Daniel: Fast Visualization of Special Relativistic Effects on Geometry and Illumination. W. de Leeuw, R. van Liere, *Data Visualization 2000 (Proceedings of the EG/IEEE TCVG Symposium on Visualization)*, Springer 2000, Wien, pp 219-228, 2000.
8. Backlight documentation: [Http://www.anu.edu.au/Physics/Searle/](http://www.anu.edu.au/Physics/Searle/)
9. Bobrowski Czeslaw: *Fizyka - krotki kurs*. Wydawnictwa Naukowo-Techniczne, Warszawa, 3rd edition, 1993 (in Polish).
10. Orear Jay: *Fizyka*. Vol. 1. Wydawnictwa Naukowo-Techniczne, Warszawa, 5th edition, 1998 (in Polish).
11. Savage C.M., Searle A.C.: *Visualising special relativity*. [Http://www.anu.edu.au](http://www.anu.edu.au), 1997.
12. Schutz Bernard F.: *Wstep do ogolnej teorii wzglednosci*. PWN, Warszawa 2002, ISBN 83-01-11666-8 (in Polish).
13. Chang M-C., Lai F., Chen W-C.: Image Shading Taking into Account Relativistic Effects. *ACM Transactions on Graphics* 15, 4 (Oct. 1996), 265-300.
14. Weiskopf D., Kraus U., and Ruder H.: Searchlight and Doppler effects in the visualization of special relativity: A corrected derivation of the transformation of radiance. *ACM Transactions on Graphics*, 18(3), July 1999.