

Daniel Weiskopf

# **Visualization of Four-Dimensional Spacetimes**

Dissertation

der Fakultät für Physik  
der Eberhard-Karls-Universität zu Tübingen  
zur Erlangung des Grades eines Doktors  
der Naturwissenschaften

vorgelegt von  
Daniel Weiskopf aus Reutlingen

---

2001

Tag der mündlichen Prüfung: 16. Februar 2001

Dekan: Professor Dr. Gerhard J. Wagner

1. Berichterstatter: Professor Dr. Hanns Ruder

2. Berichterstatter: Professor Dr. Wolfgang Straßer

3. Berichterstatter: Professor Dr. Thomas Ertl

## Abstract

In this thesis, new and improved methods for the visualization of four-dimensional spacetimes are presented.

The first part of this thesis deals with the flat spacetime of special relativity. A unified physical basis for special relativistic visualization is established. Issues of illumination, color vision, transformation of properties of light, and the kinematics of accelerating bodies are discussed. In particular, a derivation of the transformation of radiance is included.

Rendering techniques for special relativistic visualization are presented. Previously known techniques—special relativistic polygon rendering and special relativistic ray tracing—are described in a unified framework. It is shown how relativistic effects on illumination can be incorporated in these techniques and it is demonstrated that visual perception is dominated by the searchlight and Doppler effects. Relativistic radiosity, texture-based relativistic rendering, and image-based relativistic rendering are proposed as new rendering methods. Relativistic radiosity can visualize effects on illumination up to arbitrary accuracy for scenes made of diffuse materials. Radiosity is well suited for interactive walk-throughs, but also for high-quality images. Texture-based relativistic rendering utilizes the texture-mapping hardware to implement the relativistic transformations. It is most appropriate for interactive applications which visualize special relativistic effects on both geometry and illumination. Image-based relativistic rendering closes the gap between well-known non-relativistic image-based techniques and relativistic visualization. Image-based rendering does not require laborious three-dimensional modeling and achieves photo-realism at high rendering speeds. Image-based relativistic rendering allows to generate photo-realistic images of rapidly moving real-world objects with great ease and is a powerful tool to produce movies and snapshots for both entertainment and educational purposes.

Interactive virtual environments for the exploration of special relativity are introduced. The first environment is a simple “relativistic flight simulator” which runs on a standard PC or graphics workstation. The second system is a sophisticated immersive virtual environment which exploits multi-pipe and multi-processor architectures. Parallelization of the relativistic transformation results in the same frame rates for relativistic rendering as for standard non-relativistic rendering. The *relativistic-vehicle-control* metaphor is introduced for navigating at high velocities. This metaphor contains a physics-based camera control and provides both active and passive locomotion.

The second part of the thesis deals with curved four-dimensional spacetimes of general relativity. Direct visualization of what an observer would see in a general relativistic setting is achieved by means of non-linear ray tracing. A generic system is presented for ray tracing in spacetimes described by a single chart. The suitability of ray tracing as a visualization tool is demonstrated by means of two examples—the rigidly rotating disk of dust and the warp metric. Extensions to single-chart ray tracing are proposed to incorporate the differential-geometric concept of an atlas. In this way, spacetimes of complex topologies can be considered. An example is included, showing the visualization of a wormhole.

Ray tracing is applied to the field of gravitational lensing. It is shown how the visualization of standard lensing can be included in a ray tracing system. Furthermore, ray tracing allows to investigate deflecting objects beyond the approximations of standard lensing. For example, large angles of deflections can be considered. The *caustic finder* is proposed as a numerical method to identify two-dimensional caustic structures induced by a gravitational lens.

The inner geometry of two-dimensional spatial hypersurfaces can be visualized by isometric embedding in three-dimensional Euclidean space. A method is described which can embed surfaces of spherical topology. This embedding scheme supports sampled metric data which may originate from numerical simulations.

Finally, a specific application in classical visualization is described. Classical visualization means the visual representation of data from relativistic simulations without taking into account the curvature of spacetime. An algorithm for the adaptive triangulation of height fields is developed in order to achieve a good mesh quality, even in areas where the underlying function has high gradients. Height field visualization is exemplarily applied to data from neutron star simulations.



Imagination is more important than knowledge.  
Knowledge is limited. Imagination encircles the world.

*“What Life Means to Einstein: An Interview by George  
Sylvester Viereck,” for the October 26, 1929 issue of  
The Saturday Evening Post.*



# Contents

<b>1 Introduction</b>	<b>1</b>	<b>9 General Relativistic Ray Tracing</b>	<b>55</b>
1.1 Principles, Methods, and Techniques . . . . .	1	9.1 Theoretical Background . . . . .	55
1.2 Goals . . . . .	2	9.2 Ray Tracing in a Single Chart Spacetime . . . . .	57
1.3 Outline . . . . .	2	9.3 Visualization of the Rigidly Rotating Disk of Dust . . . . .	58
		9.4 Visualization of the Warp Metric . . . . .	60
		9.5 Spacetimes of Non-Trivial Topology . . . . .	62
<b>I Special Relativistic Visualization</b>	<b>3</b>	<b>10 Gravitational Lensing</b>	<b>65</b>
<b>2 Introduction to the Visualization of Special Relativity</b>	<b>5</b>	10.1 Theoretical Background . . . . .	65
2.1 Historical Remarks . . . . .	5	10.2 Visualization in Standard Lensing Theory . . . . .	66
2.2 Previous Work . . . . .	6	10.3 Direct Application of Ray Tracing . . . . .	67
2.3 Outline . . . . .	6	10.4 Caustic Finder . . . . .	68
<b>3 Color Vision and Illumination</b>	<b>7</b>	<b>11 Visualization of Inner Geometry by Embedding</b>	<b>71</b>
3.1 Radiometry . . . . .	7	11.1 Embedding of a Triangulated Two-Surface . . . . .	71
3.2 Colorimetry . . . . .	7	11.2 Conditions on Triangle Mesh . . . . .	71
3.3 Reconstruction of the Power Spectrum . . . . .	11	11.3 Implementation and Results . . . . .	72
3.4 Local Illumination . . . . .	12	11.4 Discussion . . . . .	73
<b>4 Physical Basis of Special Relativistic Visualization</b>	<b>15</b>	<b>12 Classical Visualization</b>	<b>75</b>
4.1 Lorentz Transformation . . . . .	15	12.1 Triangulation of Height Fields . . . . .	75
4.2 Aberration of Light and Doppler Effect . . . . .	16	12.2 Implementation and Results . . . . .	76
4.3 Searchlight Effect . . . . .	17	<b>13 General Relativistic Visualization: Summary and Open Questions</b>	<b>77</b>
4.4 Transformation of Photon Field . . . . .	17	13.1 Ray Tracing and Gravitational Lensing . . . . .	77
4.5 Inter-Frame Relationships for Events . . . . .	18	13.2 Embedding . . . . .	78
4.6 Equivalence of Exocentric and Egocentric View . . . . .	18	13.3 Classical Visualization . . . . .	78
4.7 Acceleration of a Point Particle . . . . .	20	<b>14 Conclusion</b>	<b>79</b>
<b>5 Rendering Techniques for Special Relativity</b>	<b>23</b>	<b>A Special Relativistic Transformation of Radiance and Irradiance</b>	<b>81</b>
5.1 Special Relativistic Polygon Rendering . . . . .	23	A.1 Derivation of the Transformation of Radiance . . . . .	81
5.2 Special Relativistic Ray Tracing . . . . .	27	A.2 Incident Irradiance . . . . .	82
5.3 Special Relativistic Radiosity . . . . .	29	<b>B Implemented Gravitational Lenses</b>	<b>85</b>
5.4 Texture-Based Special Relativistic Rendering . . . . .	31	B.1 Axially Symmetric Lenses . . . . .	85
5.5 Image-Based Special Relativistic Rendering . . . . .	35	B.2 Quadrupole Lenses . . . . .	86
<b>6 Virtual Environments for Special Relativity</b>	<b>41</b>	<b>C Program Documentation</b>	<b>87</b>
6.1 Previous and Related Work . . . . .	41	C.1 Special Relativistic Polygon Rendering . . . . .	87
6.2 Relativistic Flight Simulator . . . . .	41	C.2 Texture-Based Special Relativistic Rendering . . . . .	88
6.3 Relativistic-Vehicle-Control Metaphor . . . . .	42	C.3 Extensions to RayViS . . . . .	89
6.4 Immersive Virtual Environment . . . . .	43	C.4 Isometric Embedding . . . . .	92
		C.5 Adaptive Subdivision of Height Fields . . . . .	92
<b>7 Special Relativistic Visualization: Summary and Open Questions</b>	<b>47</b>	<b>D Supplementary Video Material</b>	<b>93</b>
7.1 Comparison of Rendering Techniques . . . . .	47	D.1 Detailed Description of the Videos . . . . .	93
7.2 Future Work . . . . .	49	<b>E Conventions and Notation</b>	<b>95</b>
		E.1 Variable Names . . . . .	95
<b>II General Relativistic Visualization</b>	<b>51</b>	<b>Bibliography</b>	<b>97</b>
<b>8 Introduction to the Visualization of General Relativity</b>	<b>53</b>		
8.1 Previous Work . . . . .	53		
8.2 Outline . . . . .	53		





## **Visualization of Four-Dimensional Spacetimes**



# Chapter 1

## Introduction

We perceive our world by means of our senses. Interaction with our surroundings is based on material and energy transport, for example, by sound or light waves or by scents. The senses are our “detectors” for these transferring “substances”. Visual perception is the most pronounced of these senses. We receive light radiated by objects of the surrounding scene and extract a wide range of information, such as shape of objects, color, emotions from facial expressions, etc. The eye accomplishes the physical detection of the incoming light, but the more complex and more interesting processing of the raw measurements is performed in the brain. Only the cognitive ability of our brain allows us to comprehend our environment.

The neurophysiological processing of visual information is a topic far beyond the scope of this thesis. For more information on this subject I refer to Mallot[96] and to a comprehensive overview on neurophysiology of visual perception by Spillmann and Werner[149]. A related field is computer vision, on which numerous textbooks[49, 69, 75, 98] are available.

Visualization has the task to graphically represent information in a way that a human being can grasp this information. It is a highly interdisciplinary field of research. First, the context in which the information is gathered has to be taken into account. In this thesis, the underlying models are based on the physics of relativity and the mathematics of differential geometry. Other fields of application require different models; typical applications concern engineering, medical imaging, molecular biology, or social sciences. Secondly, visualization makes use of computer graphics to actually render the visual representation. Thirdly, the way the human user processes and understands information has to be considered, a fact that relates visualization to the previously mentioned neurophysiology of visual perception.

In this thesis, I focus on the first two points. In particular, I investigate how the physical properties and the mathematical structure of four-dimensional spacetime, which is the basis for special and general relativity, can be taken into account in the visualization process and how the corresponding images can be efficiently rendered on a graphics system.

### 1.1 Principles, Methods, and Techniques

Visualization has a long tradition, starting well before the advent of the computer era. To understand scientific and technical data graphical tools were already used back in the 18th and 19th centuries. For example, contour plots, height fields,

or color coding allowed to visualize scalar data in maps and atlases. Collections of arrows displayed vector fields, such as the magnetic field. The visualization techniques were essentially restricted to two-dimensional fields. With the invention of the computer and its graphical abilities, more sophisticated visualization techniques have become possible, in particular, regarding the presentation of data of higher dimension and of higher-order tensor fields. Moreover, time-dependent phenomena can now be taken into account as well.

The brain builds a model of real objects seen by a human being. The visualization principle is to render objects which cannot be observed directly in order to enable us to grasp their meaning. The reason why the object cannot be directly perceived can be multifaceted. It might be too small to be seen, for example, a molecule or other microscopic objects; or it might be too far away, such as a neutron star; it might be too dangerous or even lethal, as it is the case for imaging diagnosis of the human body; it might be hidden by an obstacle, such as a deposit of oil and gas; it might be hardly visible or invisible, such as a flow in a fluid; and so on. The goal of visualization is to map these “objects” to types of items which are familiar to the viewer.<sup>1</sup> So far, only the geometry of three-dimensional objects has been implied. Visualization can be extended to time-dependent, dynamic, or higher-dimensional objects as well, thus expanding the class of presented data.

The topic of this thesis is the visualization of spacetimes and information from relativistic physics. In our everyday life we do not deal with relativistic effects. We are able to measure only very restricted kinds of information from physical relativistic systems. Most of the processes which are governed by relativistic effects take place either in the microscopic scales of elementary particles or in the macroscopic scales of objects in remote areas of the universe. The data we register from astrophysical objects is essentially restricted to the information carried to the earth by photons. Therefore, we obtain two-dimensional images—the direction of the incoming photons and their spectral energy distribution. The time intervals of our measurements are tiny compared to the time scales of most cosmological or astrophysical processes. To sum up, we are restricted to two-dimensional data as opposed to an observation of the full four-dimensional, spatial and temporal information about the physical system.

Therefore, it is essential to make use of a mathematical and physical model to interpret the data. Very often, we do not even visualize data from measurements, but rather from nu-

---

<sup>1</sup>“Objects” is put within quotation marks, since they might cover a wide range of abstractness, reaching from real materialized objects to abstract mathematical models.

merical simulations.

In relativistic physics, visualization has to deal with abstract and non-intuitive concepts. Relativity is based on four-dimensional spacetime. Spacetime is a curved Lorentzian manifold. To put it another way, we have to address the following fundamental problems. First, we live in a three-dimensional world. How can we understand a four-dimensional manifold? Secondly, we are used to flat geometry. How can we grasp a curved spacetime? Thirdly, how can we develop an intuitive understanding of a manifold without a positive definite metric? (A Lorentzian manifold does not have a positive definite metric; it allows distances of positive, negative, or null square length). Fourthly, spacetime allows complex topologies. How can we represent these topologies?

The problem is to find familiar visual analogies to support comprehension. One approach is to visualize what an observer would actually see when he or she is watching a relativistic object. This procedure is in tradition of the *gedanken experiments*, which were frequently used by Einstein. This direct visual approach is a key concept in this thesis. Nevertheless, I show more abstract visualization techniques as well, such as isometric embedding or classical visualization.

## 1.2 Goals

This thesis intends to present new or improved and extended methods for visualizing phenomena in four-dimensional spacetimes. These techniques are developed for the specific needs of the theory of relativity. In particular, the following goals are aspired.

There already exist techniques for relativistic visualization. However, the supported types of physical systems and phenomena are restricted to specific classes. I intend to take into account wider classes of physical processes by improving existing visualization techniques and proposing new ones.

I specifically address issues of time-dependency, dynamics, and interactivity. Previous work is focused on static systems or uniformly moving, precalculated objects. In this way, additional means of information transfer offered by time-dependent visualization are wasted. I propose that interactive systems should be employed wherever possible. Interactivity means that the user can interact with the system, i.e., that she or he can steer the system, change parameters, or move objects with prompt feedback. In this context, the issue of human-computer interfaces is addressed and new interaction techniques specific to relativity are proposed. Furthermore, I focus on performance in general and rendering performance in particular to allow for real-time applications.

Another goal is the production of high-quality images, especially for publication in print media, TV, video, and movies. This goal is achieved by various means, such as adaptive refinement, adequate sampling, or use of image-based modeling.

Moreover, the software design of the developed programs is based on an object-oriented framework. Several layers of abstraction in the design allow for easy extensibility and high usability. The development of generic visualization methods takes priority over specialized implementation of single physical systems. In the same context, platform-independency is pursued. Moreover, standard file formats are supported for input and output wherever applicable. All these steps aim at usability of the software even in a longer term.

## 1.3 Outline

The thesis is split into two major parts. The visualization in special relativity is investigated in the first part. The second part addresses curved spacetimes of general relativity.

Spacetime is flat within special relativity. Therefore, visualization is mathematically less elaborate and computationally less expensive than in general relativity. With the computer power available today, many special relativistic visualization techniques can be run in real-time and thus allow interactivity. In general, the visualization of general relativity is much more costly and prohibits real-time applications. Part One and Part Two of this work are connected by the fact that a locally Lorentzian frame of reference can always be found, even in curved spacetime. Therefore, many results from special relativity can be extended to general relativity.

Part One extends over Chapters 2–7. First, it deals with basics of color vision, radiometry, illumination, and special relativity. It is followed by an extensive discussion of special relativistic rendering methods. Then, interaction techniques and virtual environments for special relativity are proposed. The first part closes with a summary and an outlook on possible future work in the field of special relativistic visualization.

Part Two contains Chapters 8–13. Three fundamentally different types of visualization are discussed. The first visualization method is ray tracing in curved spacetime. Analogously to the visualization idea of Part One, ray tracing simulates the visual perception within curved spacetime. Here, I specifically address the issue of gravitational lensing. The second technique gives an abstract account of a curved manifold by isometric embedding of a two-dimensional hypersurface. The third method uses classical visualization to show data from simulations or measurements. Part Two ends with a brief summary and a description of future work in the field of general relativistic visualization.

Chapter 14 concludes both parts of the thesis.

The Appendix contains some detailed information. An explicit derivation of the transformation of radiance between two inertial frames of reference is presented. Then, the implemented gravitational lenses are described. A documentation of the developed software and the accompanying video follows. Variable name conventions and notations are included as well.

## **Part I**

# **Special Relativistic Visualization**



## Chapter 2

# Introduction to the Visualization of Special Relativity

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[44] in 1905. 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, there is no hope of achieving velocities comparable to the speed of light for extended macroscopic objects. Mankind is only able to accelerate microscopic particles up to velocities comparable to the speed of light, for example, electrons, protons, etc. in high-energy accelerators. In nature, there exist some extended relativistic objects such as QSOs (quasi-stellar objects), jets, or accretion phenomena. However, most of them are too far away to be spatially resolved by contemporary telescopes and thus do not provide illustrations of special relativity.

Therefore, computer simulations are the only means of visually exploring the realm of special relativity. The first part of this work intends to present different kinds of visualization techniques which help to improve the comprehension of special relativity. These techniques will enable us to experience special relativity, including phenomena such as Lorentz contraction, time dilation, aberration, finite speed of light, the Doppler effect, and the searchlight effect. This approach should help to improve the intuition of physicists. It gives a good motivation for newcomers and students to learn more about the theory. Moreover, it is an ideal means for disseminating special relativity to the public and thus supports popularizing physics.

### 2.1 Historical Remarks

Remarkably, the issue of visual perception in special relativity was ignored for a long time, or wrong interpretations were given. Einstein writes in his original work on the special theory of relativity[44]:

“Ein starrer Körper, welcher in ruhendem Zustand ausgemessen die Gestalt einer Kugel hat, hat also in bewegtem Zustande—vom ruhenden System aus betrachtet—die Gestalt eines Rotationsellipsoides

mit den Achsen

$$R\sqrt{1 - \left(\frac{v}{V}\right)^2}, R, R.$$

Während also die  $Y$ - und  $Z$ -Dimension der Kugel (also auch jedes starren Körpers von beliebiger Gestalt) durch die Bewegung nicht modifiziert erscheinen, erscheint die  $X$ -Dimension im Verhältnis  $1 : \sqrt{1 - (v/V)^2}$  verkürzt, also um so stärker, je größer  $v$  ist.”

According to Miller[105], the translation into English is:

“A rigid body which, measured in a state of rest, has the form of a sphere, therefore has in the state of motion—viewed from the resting system—the form of an ellipsoid of revolution with the axes

$$R\sqrt{1 - \left(\frac{v}{V}\right)^2}, R, R.$$

[remark:  $V$  is the speed of light,  $v$  the velocity of the sphere, and  $R$  the radius of the sphere]. Thus, whereas the  $Y$ - and  $Z$ -dimensions of the sphere (and therefore of every rigid body of arbitrary form) do not appear modified by the motion, the  $X$  dimension appears shortened in the ratio  $1 : \sqrt{1 - (v/V)^2}$ , i.e., the greater the value  $v$ , the greater the shortening.”

The difference between *seeing* and *measuring* is crucial. Measurements are performed at all relevant sample points simultaneously with respect to the reference frame of the observer. Seeing, however, is based on the photons which arrive simultaneously at the observer's camera. These photons are usually not emitted simultaneously (with respect to the observer's frame of reference) because of the finite speed of light.

Einstein uses the words “betrachtet” (view, regard, or consider) and “erscheinen” (appear) in his paper and hence does not clearly express the difference between measuring and seeing. However, there are other publications which definitely give a wrong account of visual perception within relativity. For example, Gamow equates the Lorentz contraction and the visual perception of moving bodies in his book “Mr Tompkins in Wonderland”[55, 56].

Apart from a previously disregarded article by Lampa[91] in 1924 about the invisibility of the Lorentz contraction, it

was only in 1959 that the first correct solutions to the issue of visual perception within special relativity were presented by Penrose[126] and Terrell[159]. Later, more detailed descriptions of the geometrical appearance of fast moving objects were given by Weisskopf[177], Boas[17], Scott and Viner[144], and Scott and van Driel[143]. These papers derive and explain the following basic properties of the appearance of rapidly moving objects. Usually, these objects do not appear to be contracted but rather rotated and distorted. Straight lines perpendicular to the direction of motion appear to be hyperbolae, straight lines parallel to the direction of motion remain straight lines.

## 2.2 Previous Work

Hsiung and Dunn[76] were the first to use advanced visualization techniques for image shading of fast moving objects. They proposed an extension of normal three-dimensional ray tracing. This technique accounts for relativistic effects on the apparent geometry as seen by the observer. Hsiung et al.[78] investigated relativistic ray tracing in more detail and included the visualization of the Doppler effect. Hsiung et al.[77] used the temporal variation of the color of objects in the scene for the visualization of relativistic time dilation.

Hsiung et al.[79] introduced the *time-buffer* for fast visualization of relativistic effects. The time-buffer method resembles the normal z-buffer and can be mapped onto it. The time-buffer technique allows for relativistic polygon rendering, a scan-line method. It is based on the apparent shapes of objects as seen by a relativistic observer. Gekelman et al.[57], Chang et al.[23], and Betts[14] investigated the polygon rendering approach in detail and gave a comprehensive presentation. The kinematics of special relativity was described, for example, by Ruder et al.[138].

Relativistic effects on illumination is a subject that is often ignored in previous work. Most authors concentrate their efforts on geometrical appearance and, apart from the Doppler effect, neglect relativistic effects on the lighting model. Chang et al.[23] and Betts[14] presented effects on illumination, but they included some misinterpretations of the special theory of relativity which lead to incorrect images. Respective corrections can be found in our papers[175, 176].

Acceleration in the visualization of special relativity is another subject widely neglected. The article by Gekelman et al.[57] is the only other work known to me which addresses the issue of acceleration in special relativistic rendering. However, Gekelman et al. did not describe acceleration in detail. They mentioned the trajectory of an accelerated cube. In their description, 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.

## 2.3 Outline

In Chapter 3, the physical and psychophysical basis of color vision is briefly described. Definitions of radiometric terms are summarized. The psychophysics of colorimetry and typical and useful spectral power distributions are presented. I dis-

cuss local illumination as the basis for all shading algorithms. In this way, a physically sound foundation for the treatment of color in the subsequent presentation of rendering techniques is established.

In Chapter 4, the physical basis of special relativistic visualization is established. The focus is on the transformation of properties of light, such as the aberration of light, the Doppler effect, and the searchlight effect. These effects determine the extensions required for special relativistic image generation. Moreover, the kinematics of a point particle within special relativity is described.

In Chapter 5, rendering techniques for special relativity are presented. In the first two sections, the two methods previously available—relativistic polygon rendering and relativistic ray tracing—are described. The following methods have been developed during the work for this thesis. The relativistic extension of radiosity allows rendering of diffusely reflecting scenes. Texture-based relativistic rendering utilizes contemporary computer graphics hardware in order to visualize relativistic effects on geometry and illumination in real-time. Image-based relativistic rendering generates panoramas and movies based on data acquired by standard cameras, which allows to produce photo-realistic images with great ease.

In Chapter 6, interaction techniques in special relativistic visualization are investigated. Acceleration of the relativistic observer is used as the basis for a new navigation technique, the *relativistic-vehicle-control* metaphor. An implementation of an immersive virtual environment for special relativity is described.

Chapter 7 concludes the first part of the thesis and gives an outlook on possible future work.



## Chapter 3

# Color Vision and Illumination

### 3.1 Radiometry

In this section, the basic terms of radiometry are defined to establish a firm foundation for the discussion of illumination in the remaining parts of this work. The respective SI units are provided as well. A more detailed presentation can be found, for example, in Glassner's book[59]. The units are defined by Le Système International d'Unités (the International System of Units) [37].

The unit of *radiant energy*  $Q$  is Joule. *Radiant power* or *radiant flux* is

$$\Phi := \frac{dQ}{dt} \quad [\text{W}],$$

with the unit Watt. *Radiant intensity* is defined as

$$I := \frac{d\Phi}{d\Omega} \quad \left[ \frac{\text{W}}{\text{sr}} \right],$$

where  $d\Omega$  is the element of solid angle as seen by the observer. Intensity also is the measure for radiant power leaving a point light source. Accordingly, the measure for radiant power leaving a point light source in an element of solid angle  $d\Omega$  and in a wavelength interval  $d\lambda$  is called wavelength-dependent intensity  $I_\lambda$ :

$$I_\lambda := \frac{d\Phi}{d\Omega d\lambda} \quad \left[ \frac{\text{W}}{\text{sr m}} \right]. \quad (3.1)$$

The energy arriving at a surface element  $dA$  is called *irradiance*  $E^i$ ,

$$E^i := \frac{d\Phi}{dA} \quad \left[ \frac{\text{W}}{\text{m}^2} \right].$$

Wavelength-dependent irradiance  $E_\lambda^i$  is the radiant power per unit area in a wavelength interval:

$$E_\lambda^i := \frac{d\Phi}{dA d\lambda} \quad \left[ \frac{\text{W}}{\text{m}^3} \right].$$

The measure of flux leaving a surface is called *radiant exitance*  $M$  or *radiosity*  $B$ ,

$$M \equiv B := \frac{d\Phi}{dA} \quad \left[ \frac{\text{W}}{\text{m}^2} \right].$$

A quadratic law of distance applies to a point-like light source, i.e.,  $M \propto 1/r^2$  with the distance  $r$ . Even for a finite size of the source, this law is a valid approximation in most applications.

Radiance  $L$  is defined as

$$L := \frac{d\Phi}{dA_\perp d\Omega} = \frac{d\Phi}{dA d\Omega_\perp} \quad \left[ \frac{\text{W}}{\text{sr m}^2} \right].$$

Either the area projected along the direction of light,  $dA_\perp$ , is used or the projected element of solid angle,  $d\Omega_\perp$ . Radiance can be expressed in terms of intensity, irradiance, or exitance:

$$L = \frac{dI}{dA_\perp} = \frac{dE^i}{d\Omega_\perp} = \frac{dM}{d\Omega_\perp} \quad \left[ \frac{\text{W}}{\text{sr m}^2} \right].$$

Radiation transport in vacuo leaves radiance invariant. If the interaction between radiation and matter is minimal—if absorption, scattering, and luminescence can be neglected—radiance is constant as well, for example for visible light in air.

The Poynting vector  $\vec{S}$  is connected with radiance by

$$\|\vec{S}\| = \int L d\Omega.$$

In the remaining parts of this work, wavelength-dependent radiance,

$$L_\lambda := \frac{d\Phi}{dA_\perp d\Omega d\lambda},$$

is of great importance, since the dependency on wavelength has to be taken into account in many calculations. Accordingly, frequency-dependent radiance is defined

$$L_\nu := \frac{d\Phi}{dA_\perp d\Omega d\nu},$$

with the frequency  $\nu$ . Energy-dependent radiance is

$$L_E := \frac{d\Phi}{dA_\perp d\Omega dE}.$$

Please note that  $E$  denotes energy, not irradiance.

### 3.2 Colorimetry

This thesis deals with the visualization of physical phenomena. Therefore, the presentation of these effects in a way that can be perceived by the human eye is an important goal. One essential part is the perception of color, i.e., the power spectrum of the incoming light has to be connected to the human perception.

*Colorimetry* deals exactly with the problem of measurement of colors. Colorimetry is a part of psychophysics, which, in general, investigates the sensual feedback to underlying stimuli. A comprehensive introduction to colorimetry can be found in Judd and Wyszecki[83] and Wyszecki and Stiles[184].

### 3.2.1 Additive Mixture of Colors

Grassmann summarizes the basics of color vision in the following three laws, cf. Judd and Wyszecki[83, page 45]:

- The human eye can distinguish only three kinds of difference or variation.
- If, of a three-stimulus mixture, one stimulus is steadily changed, the color of the mixture steadily changes.
- Stimuli of the same color produce identical effects in mixtures regardless of their spectral composition.

These laws are based on additive mixture of colors, which adds up the responses to stimuli for each component. There exist three different basis colors. Red, green, and blue are the most prominent basis colors. The relative radiances of the three components are called the *tristimulus values*  $R, G, B$ . The tristimulus values are normalized so that white has  $R = G = B = 1$ .

The tristimulus values can be combined to form the corresponding color. The basic colors are called *primaries*  $\vec{R}, \vec{G}, \vec{B}$ . To put it another way, tristimulus values are components of an element in a three-dimensional tristimulus-color space whose basis functions are the three primaries. The additive mixture of a color  $\vec{C}$  is

$$\vec{C}(R, G, B) = R\vec{R} + G\vec{G} + B\vec{B}.$$

Negative tristimulus values cannot be generated by additive mixture, since stimuli cannot be subtracted. However, negative components can be eliminated by addition. For example,

$$\vec{C} = R\vec{R} + G\vec{G} - B\vec{B}$$

can be re-formulated to

$$B\vec{B} + \vec{C} = R\vec{R} + G\vec{G}.$$

Here,  $B\vec{B}$  and  $\vec{C}$  are added and then compared to  $R\vec{R} + G\vec{G}$ . In this sense, the tristimulus-color space can be regarded as a vector space with three basis vectors  $\vec{R}, \vec{G}, \vec{B}$ .

### 3.2.2 Transformation of Primaries

So far, only the primaries red, green, and blue have been used. Since the primaries are basis vectors of a three-dimensional vector space, a change to another basis  $\{\vec{P}_i | i = 1, 2, 3\}$  is possible. The vectors of the new basis have to be linearly independent, i.e., a primary must not be reproduced by a mixture of the two remaining primaries. If the old primaries are represented by the new primaries according to

$$\begin{aligned}\vec{R} &= \sum_{i=1}^3 P_{i,R} \vec{P}_i, \\ \vec{G} &= \sum_{i=1}^3 P_{i,G} \vec{P}_i, \\ \vec{B} &= \sum_{i=1}^3 P_{i,B} \vec{P}_i,\end{aligned}$$

then the transformation from old coordinates  $(R, G, B)$  to new coordinates  $(C_1, C_2, C_3)$  of an arbitrary color  $\vec{C}$  is

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} P_{1,R} & P_{1,G} & P_{1,B} \\ P_{2,R} & P_{2,G} & P_{2,B} \\ P_{3,R} & P_{3,G} & P_{3,B} \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

The change of basis is described by the matrix

$$M = \begin{pmatrix} P_{1,R} & P_{1,G} & P_{1,B} \\ P_{2,R} & P_{2,G} & P_{2,B} \\ P_{3,R} & P_{3,G} & P_{3,B} \end{pmatrix}.$$

The transformation from new to old coordinates is accomplished by the inverse matrix  $M^{-1}$ .

In 1931, the CIE (Commission Internationale de l'Éclairage) introduced a reference system for color measurements. This system uses the norm primaries  $\vec{X}, \vec{Y}, \vec{Z}$ . The norm primaries cannot be produced by physical power spectra and thus are called virtual primaries. An advantage of the XYZ system is that all physical colors are represented by an additive mixture of the norm primaries, i.e., that the tristimulus values are positive-definite. A motivation for the specific choice of norm primaries is given in Sect. 3.2.4.

### 3.2.3 Chromaticity

The terms

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B},$$

are called *chromaticities* of a color with respect to the RGB basis. Accordingly, the chromaticities with respect to a generic basis  $\{\vec{P}_i | i = 1, 2, 3\}$  are

$$c_i = \frac{C_i}{\sum_{j=1}^3 C_j}, \quad i = 1, 2, 3,$$

where  $C_i$  are the corresponding tristimulus values. They describe the type of the corresponding color on a chromaticity diagram, which is a hypersurface containing the three primaries in the tristimulus-color space. Figure 3.1 shows the standard CIE 1931 chromaticity diagram for the system of XYZ norm primaries.

### 3.2.4 Color-Matching Functions

By color-matching experiments with men and women with normal color vision, the tristimulus values of the spectral colors were measured with respect to a fixed set of primaries. These measurements can be used to determine the tristimulus values from the spectral power distribution in order to generate the final output on the monitor.

The red primary  $\vec{R}$  at 700.0 nm wavelength, the green primary  $\vec{G}$  at 546.1 nm wavelength, and the blue primary  $\vec{B}$  at 435.8 nm wavelength are the standard choice of basis vectors. The *color-matching functions* yield the spectral values in the range of visible light. In general, they are denoted  $\vec{f}_i(\lambda)$  for  $i = 1, 2, 3$ . The spectral color of wavelength  $\lambda$  is

$$\vec{C}(\lambda) = \sum_{i=1}^3 \vec{f}_i(\lambda) \vec{P}_i.$$

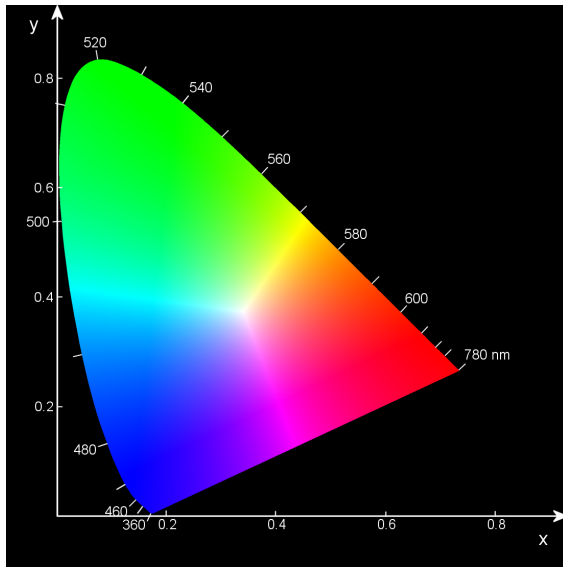


Figure 3.1: CIE 1931 chromaticity diagram.

In the case of the RGB system, the color-matching functions are denoted  $\bar{r}(\lambda)$ ,  $\bar{g}(\lambda)$ ,  $\bar{b}(\lambda)$ . The graphs of the color-matching functions for the primaries red, green, and blue are displayed in Fig. 3.2. The respective values are tabulated in Wyszecki and Stiles[184, Table I(3.3.3)].

An arbitrary stimulus in the form of the wavelength-dependent radiance  $L_\lambda$  generates the tristimulus values  $C_i$ ,

$$C_i = \int_{\text{supp}(\bar{f}_i)} L_\lambda(\lambda) \bar{f}_i(\lambda) d\lambda, \quad i = 1, 2, 3. \quad (3.2)$$

For the color-matching functions of the CIE, the support is [380 nm, 780 nm]. In the RGB system, the color-matching function for red is negative for an extended interval in the domain. Therefore, the respective spectral colors cannot be reproduced by an additive mixture of the  $\bar{R}$ ,  $\bar{G}$ ,  $\bar{B}$  primaries. Figure 3.3 shows the *spectrum locus* and the *purple line* in a chromaticity diagram. The spectrum locus is the set of all spectral colors. The short and long wavelength regions are connected by the purple line. Figure 3.3 shows that the spectrum locus is outside the gamut of the primaries  $\bar{R}$ ,  $\bar{G}$ ,  $\bar{B}$  for wavelengths between 380 nm and 550 nm.

As mentioned before, the CIE (Commission Internationale de l'Éclairage) introduced a reference system for color measurements. This system uses the norm primaries  $\bar{X}$ ,  $\bar{Y}$ ,  $\bar{Z}$  and the corresponding color-matching functions  $\bar{x}$ ,  $\bar{y}$ ,  $\bar{z}$ . Since  $\bar{y}$  equals the luminous-efficiency function,  $Y$  reflects the luminance of the light stimulus. The primaries  $\bar{X}$ ,  $\bar{Y}$ ,  $\bar{Z}$  lie outside the spectrum locus, cf. Fig. 3.3. They cannot be produced by physical power spectra and thus are called virtual primaries.

The standard color-matching functions of the CIE XYZ system are the basis for many color-processing applications, including all implementations for this thesis. Nevertheless, I would like to point out that there exist more recent and improved measurements of color vision, for example, by Stockman et al.[153]. An up-to-date investigation of the neurophysiology of color perception can be found in Teufel[160].

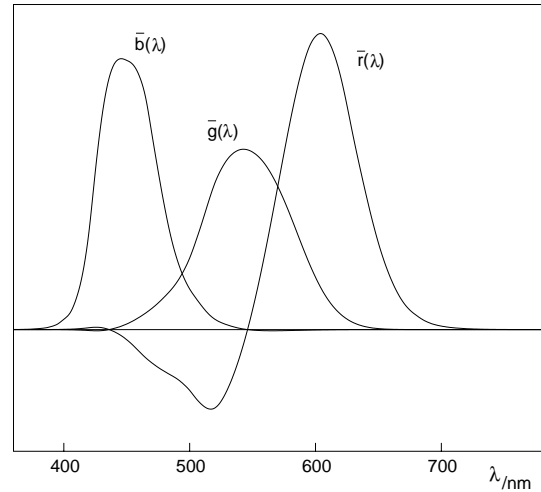


Figure 3.2: Color-matching functions for the primary stimuli red (700.0 nm), green (546.1 nm), and blue (435.8 nm). The units of the primary stimuli are of radiant power ratios 72.1:1.4:1.

### 3.2.5 Mappings in Tristimulus-Color Space

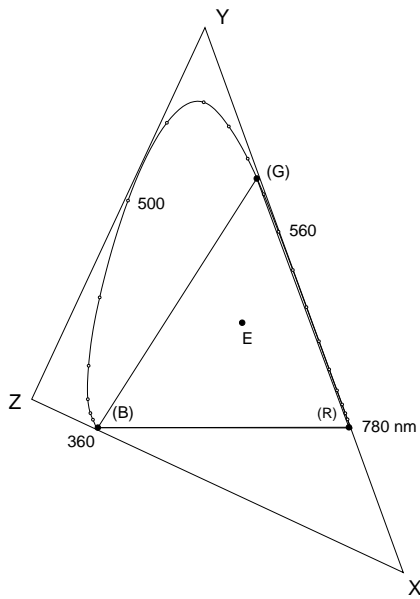
In the previous section, it is shown that, in many cases, spectral colors cannot be reproduced by an additive mixture of the  $\bar{R}$ ,  $\bar{G}$ ,  $\bar{B}$  primaries. Figure 3.4 visualizes the problem. The colors which can be physically produced lie inside the spectrum locus. Contemporary computer monitors can thus generate only a small part of these colors.

In addition to the chromaticity, the luminosity can lie outside the displayable range. In general, one needs to investigate how an arbitrary color  $\vec{C}$  can be mapped into the set of displayable colors. If the color is given with respect to the RGB system, one has to find a mapping of the color  $\vec{C}$  into the RGB unit cube. A continuous function  $G$  for this (*color*) *gamut mapping* has to be determined in a way that color perception comes close to the original values:

$$G: \mathbb{R}^3 \longrightarrow [0, 1]^3, \quad \vec{C} \longmapsto \vec{C}' = G(\vec{C}).$$

Stone et al.[154] investigate gamut mapping for reproduction on paper. They split  $G$  in two steps. The first step represents a transformation of tristimulus-color space in a way that most of the colors can be reproduced. Possible types of transformation are translation, scaling, rotation around the gray axis (axis from black point to white point), and desaturation by adding white. The parameters are interactively controlled by the user, whose color perception is essential for adjusting these parameters. In the second step, the colors which still cannot be reproduced are clipped. Unfortunately, this technique cannot be used in most relativistic visualization methods because this type of user interaction is undesirable in an automated visualization process and is too time consuming. Furthermore, the method needs the complete image to be known and thus does not allow progressive image generation.

Therefore, only simpler gamut mapping procedures are used in the relativistic visualization techniques. They are based on standard methods, cf. the description in Hall[66] or Glassner[59]. The fastest method is color clamping, which clamps the tristimulus values to zero or one, respectively. An-



**Figure 3.3:** Chromaticity diagram with the primaries  $\bar{X}$ ,  $\bar{Y}$ ,  $\bar{Z}$  and the spectral colors for red, green, and blue.

other method is desaturation by adding white. If the resulting color is too bright, the tristimulus values are uniformly scaled so that the maximum value equals one. These two methods change brightness. The third method is another type of desaturation which keeps the brightness constant as long as the maximum brightness is not exceeded.

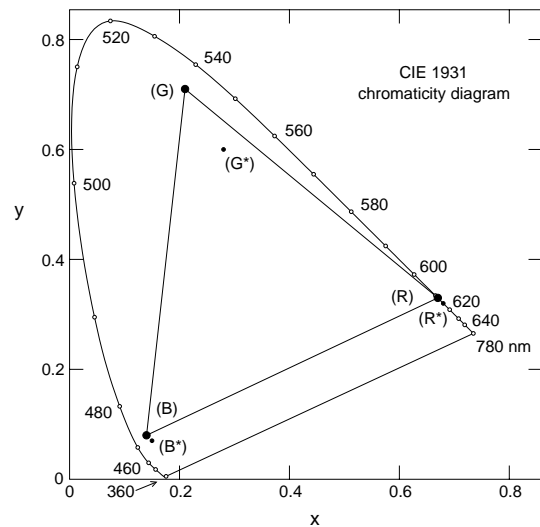
These methods are compared in Fig. 3.5. Unfortunately, the color reproduction on paper is also subject to gamut mapping problems. Therefore, the reproduced colors are not completely correct.

### 3.2.6 Curved Color Spaces

From the previous discussion, there is overwhelming evidence that the manifold of perceived colors is three-dimensional. The Grassmann's laws imply three different types of stimuli. Psychological specifications of color use three parameters, such as hue, chroma, and lightness. The structure of the cones in our eyes also indicates that the color manifold is three-dimensional.

However, the internal structure of perceptual color space is not determined by the fact that the space is three-dimensional. What are the intrinsic properties of this manifold? In particular, in which way do we perceive color differences? Color space has to be equipped with an internal structure which makes it a curved manifold. Please note that this internal structure is independent of the law of additive color mixture. The addition of colors is caused by adding physical stimuli, which is a linear operation under normal conditions of vision and thus yields the properties of the vector space mentioned in previous sections. The internal structure, however, is based on perceptible color differences: How do we decide that two colors are different, what is the measure for that difference?

Silberstein[147] concludes that the internal structure of color space is described by a metric of Riemannian type. Ashtekar et al.[6] present an abstract derivation of the Rie-



**Figure 3.4:** Chromaticity diagram with the gamut of the NTSC monitor primaries  $\bar{R}$ ,  $\bar{G}$ ,  $\bar{B}$  and the primaries  $\bar{R}^*$ ,  $\bar{G}^*$ ,  $\bar{B}^*$  which can be produced by usual phosphors. Major parts of the physically possible colors cannot be displayed on computer monitors because the spectrum locus lies outside the supported color gamut.

mannian structure of color space, based only on the assumption that the essential features of perceived colors are completely captured in three absorption coefficients. Therefore, it is widely accepted that perceptual color space is equipped with a Riemannian metric.

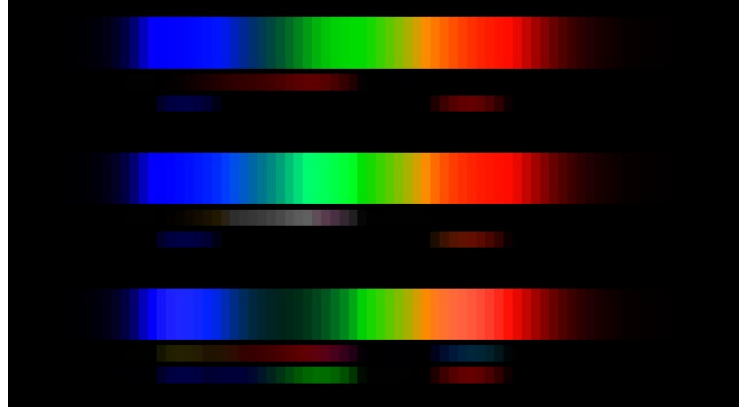
The perceptible color difference is expressed in terms of the line element  $ds$ , with

$$ds^2 = \sum_{i,j=1}^3 g^{ij} dC_i dC_j,$$

where  $g^{ij}$  is the metric tensor and  $dC_i$  is the differential of the color coordinate  $C_i$ . An introduction to the mathematics of curved manifolds can be found in Sect. 9.1.

The difficult task is to actually measure this metric by means of investigating normal color vision of human beings. First efforts to formulate an appropriate line element go back to Helmholtz[70]. Other metrics were proposed by Schrödinger[142] and Stiles[151]. More details and references to further metrics can be found in Judd and Wyszecki[83].

A more pragmatic approach does not explicitly specify the metric of color space, but introduces non-linear mappings from standard color coordinates, such as  $RGB$  or  $XYZ$ , to new coordinates. The new coordinate system is intended to be a perceptually uniform color space. However, a completely linear color space has not been developed yet. The unsuccessful attempts have, on the contrary, supported the suspicion that a strictly linear color space cannot possibly be developed. Nevertheless, some good approximations have been found. Prominent ones are the  $L^*u^*v$  and the  $L^*a^*b$  systems, cf. Judd and Wyszecki[83].



**Figure 3.5:** Gamut mapping methods for the reproduction of the spectrum of visible light. In addition to the spectrum itself, the figure displays the colors which have to be added or subtracted in order to achieve the respective reproduction. The first block shows color clamping at zero and one. In the middle block, the negative RGB values are removed by adding white. Bright colors are displayed by uniform scaling of the tristimulus values. The last block is reproduced by using desaturation at constant brightness.

### 3.3 Reconstruction of the Power Spectrum

In many applications of relativistic visualization, it is not sufficient to consider only three tristimulus values. In fact, often the spectral power distribution of the light has to be taken into account, for example, when color changes due to the Doppler effect or gravitational redshift have to be calculated.

Unfortunately, common input data does not comprise the full power spectrum, but only the tristimulus values because the materials in geometric modeling are normally specified by tristimulus values and image-based modeling makes use of cameras which provide RGB information only. The full power spectrum has to be reconstructed from the tristimulus values in these cases.

In this section, prominent spectral power distributions and their representation with respect to basis functions are described. The basis function representation is used to build a reconstruction scheme which computes a spectral power distribution from given tristimulus values.

#### 3.3.1 Spectral Power Distributions

Percy[125] describes in detail how a finite set of orthonormal basis functions can be used to represent radiance. The orthonormality condition can be omitted and an arbitrary set of basis functions can be used if no projection onto the basis functions is needed. In this representation, wavelength-dependent radiance is

$$L_\lambda(\lambda) = \sum_{j=1}^{n_L} l_j L_{\lambda,j}(\lambda), \quad (3.3)$$

with the set of basis functions,  $\{L_{\lambda,j}(\lambda) | j \in \mathbb{N}, 1 \leq j \leq n_L\}$ , and the coefficients of the vector representation,  $l_j$ .

The Planck distribution is the first example of a basis function. It describes the blackbody radiation and thus has a direct physical meaning and a widespread occurrence in nature. The

Planck distribution is

$$L_\lambda(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{\exp(hc/kT\lambda) - 1}, \quad (3.4)$$

where  $T$  is the temperature in Kelvin,  $k$  is the Boltzmann constant, and  $h$  is the Planck constant. The color values for blackbody radiation lie inside the RGB color gamut for all temperatures, cf. Fig. 3.6. A Planck spectrum is changed by the Doppler effect or gravitational redshift to another Planck spectrum of a different temperature, cf. Peebles and Wilkinson[124]. Therefore, the blackbody radiation can be reproduced on a computer monitor for all relevant physical systems and scenarios.

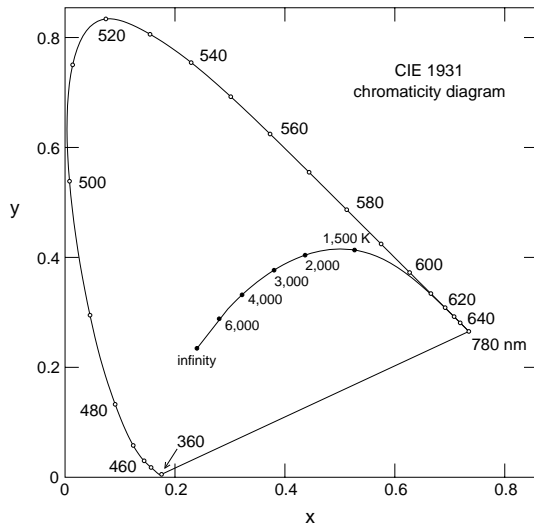
Various other sets of basis functions are described in the literature, such as box functions[67], Fourier functions[58], Gaussian functions with adapted width[156], and delta functions[58].

#### 3.3.2 Reconstruction Process

Unfortunately, the mapping from the space of all power distributions into tristimulus space is not injective. There are infinitely many spectra associated with a single color. This phenomenon is called *metamerism*. However, a possible spectrum can always be determined and metamerism gives considerable freedom of doing so.

Two different reconstruction schemes are presented in this section. The first makes use of the representation via basis functions. First, a set of three new basis functions is chosen. Second, the tristimulus values corresponding to each single basis function are determined, i.e., the representation of the new basis functions with respect to the original basis. Third, the matrix representing the transformation from the original basis to the new basis is constructed according to Sect. 3.2.2. This matrix allows to compute the weights for the three basis functions from given tristimulus values. In this way, the power spectrum can be calculated from the color values.

A simple approach models three RGB values by the line spectrum consisting of the corresponding primaries at 700.0 nm, 546.1 nm, and 435.8 nm. The peaks are weighted



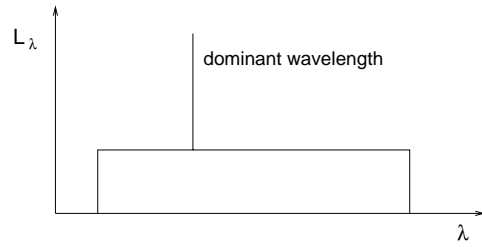
**Figure 3.6:** Chromaticity diagram with the Planck spectrum locus. The Planck spectrum locus is inside the displayable gamut for all temperatures.

by the respective tristimulus values. Unfortunately, discontinuous changes in color can appear if a primary is shifted out of the visible spectrum. Noticeable changes might appear even for a small frequency shift. For example, the blue primary can be shifted out of the visible spectrum by the Doppler effect for a velocity as low as 13 percent of the speed of light. Therefore, a set of wide-spread basis functions can provide a smoother transition of colors.

The other reconstruction scheme is based on the dominant wavelength model[51] and does not use a representation via basis functions. The dominant wavelength model is useful because it provides a smooth change of color and brightness for a wide range of frequency-shifting. Figure 3.7 shows an example of a spectral power distribution with a dominant wavelength. The corresponding spectral power distribution consists of a spike at the dominant wavelength and of a uniform distribution, i.e., white light. The luminance and excitation purity determine the levels of the two parts of the spectrum. The excitation purity corresponds to the saturation of the color, the luminance to the brightness. The location of the dominant wavelength spike depends on the hue of the color. The parameters for the dominant wavelength model can be computed from RGB values according to [51]. In contrast to the original implementation of the dominant wavelength model, the uniform part of the spectrum is not restricted to the range of visible wavelengths, but comprises a larger interval. In this way, the spectrum is still present after frequency-shifting.

### 3.4 Local Illumination

Local illumination describes the reflectance properties of a small patch of a surface. It is always possible to find a local Lorentz frame. Therefore, the local illumination known from standard computer graphics can be adopted even for relativistic rendering, as long as all physical properties are transformed to the local reference frame before the illumination calculation.



**Figure 3.7:** Spectral power distribution illustrating dominant wavelength.

The reflection of a material is described by the spectral *bidi-rectional reflectance distribution function* (BRDF),

$$\begin{aligned} f_{\lambda,r}(\lambda, \phi_r, \theta_r, \phi_i, \theta_i) &= \frac{dL_{\lambda}^r(\lambda, \phi_r, \theta_r, \phi_i, \theta_i)}{dE_{\lambda}^i(\lambda, \phi_i, \theta_i)} \\ &= \frac{dL_{\lambda}^r(\lambda, \phi_r, \theta_r, \phi_i, \theta_i)}{L_{\lambda}^i(\lambda, \phi_i, \theta_i) \cos \theta_i d\Omega_i}, \end{aligned}$$

where  $L_{\lambda}^i$  is the incident wavelength-dependent radiance,  $E_{\lambda}^i$  is the wavelength-dependent irradiance, and  $L_{\lambda}^r$  is the reflected wavelength-dependent radiance. Figure 3.8 illustrates the geometry of the quantities needed for the computation of radiance.

By integrating over all directions of the incident radiation, the *reflectance equation* is obtained:

$$\begin{aligned} L_{\lambda}^r(\lambda, \phi_r, \theta_r) \\ = \int_{\Omega_{\text{incident}}} f_{\lambda,r}(\lambda, \phi_r, \theta_r, \phi_i, \theta_i) L_{\lambda}^i(\lambda, \phi_i, \theta_i) \cos \theta_i d\Omega_i, \end{aligned} \quad (3.5)$$

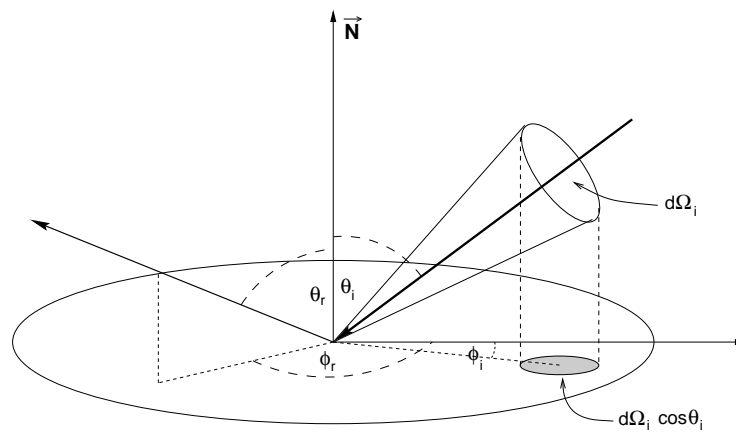
where  $\Omega_{\text{incident}}$  is the incident hemisphere. The BRDF and the reflectance equation are the basis of local illumination. Further information on this topic can be found, for example, in Glassner's book[59].

The three simplest and often used shading models are the *Lambert* model, the *Phong* model, and the *Blinn-Phong* model.

The Lambert model applies to a perfectly diffuse surface. The BRDF is constant for all wavelengths and all directions. The value of the BRDF is  $f_{\lambda,r} = 1/\pi$ .

The Phong model[128] represents highlights by an exponentiated cosine law. The reflected light is modeled as a cone centered around the reflection direction with exponentially decreasing intensity. Suppose that the angle between reflection vector and viewing vector is given by  $\alpha$ . Then, the reflected light is modulated by  $(\cos \alpha)^{k_e}$ ;  $k_e$  is the roughness or shininess exponent. Blinn[16] proposes a modification of Phong shading which avoids the computation of the reflection vector and thus is slightly faster. Neither the Phong nor the Blinn-Phong model is normalized, even the energy-conservation condition is ignored.

The *Cook-Torrance* shading model[28] is based on a physical model of a surface. The surface is assumed to consist of small mirrors, the microfacets, which are randomly oriented. The distribution of the microfacets is parameterized by a roughness term. Reflectance itself is based on Fresnel's formulas and hence depends on wavelength. The Cook-Torrance model is a physical shading model. Descriptions and references to more advanced local illumination models can be found, for example, in Glassner[59].



**Figure 3.8:** Geometry for the BRDF. The normal vector  $\vec{N}$  points upwards. The direction of the incident light is described by  $(\phi_i, \theta_i)$ , the direction of the reflected light is described by  $(\phi_r, \theta_r)$ . The projected solid angle is denoted  $d\Omega_i \cos \theta_i$ .





## Physical Basis of Special Relativistic Visualization

In this chapter, the mathematical and physical basis of special relativity is introduced in so far as needed for the understanding of special relativistic visualization. A detailed presentation can be found in numerous text books, for example [35, 107, 108, 135].

### 4.1 Lorentz Transformation

The Lorentz transformation relates two inertial frames of reference. It is a basic element of the special theory of relativity.

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*

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

Throughout this thesis, units are used in which the speed of light  $c$  equals 1.

The line element is

$$\begin{aligned} ds^2 &= dt^2 - dx^2 - dy^2 - dz^2 \\ &= \eta_{\mu\nu} dx^\mu dx^\nu. \end{aligned}$$

The Greek indices are from the set  $\{0, 1, 2, 3\}$ . The Einsteinian sum convention is used. The *Minkowski metric* is

$$\eta_{\mu\nu} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

The *Lorentz transformation* is a linear and homogeneous transformation of Minkowski coordinates,

$$x^\mu \mapsto x^{\mu'} = \Lambda^\mu_{\nu} x^\nu,$$

which leaves the metric  $\eta_{\mu\nu}$  invariant. From the previous equation follows

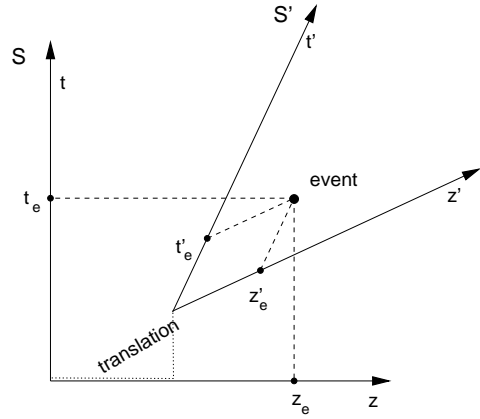
$$\Lambda^\mu_{\nu} = \frac{\partial x^{\mu'}}{\partial x^\nu}.$$

The inverse Lorentz transformation is

$$x^{\mu'} \mapsto x^\mu = \Lambda_\nu^{\mu'} x^{\nu'},$$

with

$$\Lambda_\nu^{\mu'} = \frac{\partial x^\mu}{\partial x^{\nu'}}.$$



**Figure 4.1:** 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 rotations of the axes of S'. Spatial rotation is not considered because the Minkowski diagram shows only one spatial dimension. The projection of an event onto the axes yields the coordinates  $(t_e, z_e)$  and  $(t'_e, z'_e)$  of the event in the two respective frames of reference. This provides a geometric interpretation of the Poincaré transformation.

The Lorentz transformations form the Lorentz group  $L$ .  $L$  is partitioned in four separate pieces which are characterized by  $\det \Lambda^\mu_{\nu}$  and  $\Lambda^0_0$ . In the remaining parts of this work, only proper ( $\det \Lambda^\mu_{\nu} = 1$ ) and orthochronous ( $\Lambda^0_0 \geq 1$ ) Lorentz transformations  $L^\uparrow_+$ , which conserve spatial and temporal orientation, are considered. The Lorentz transformation connects two inertial frames of reference and leaves the speed of light invariant. The Lorentz transformation can be regarded as a change of coordinate systems between the associated inertial frames.

The Poincaré group consists of the linear inhomogeneous transformations which leave  $\eta_{\mu\nu}$  invariant. The Lorentz transformation is extended to the Poincaré transformation by including translations of frames of reference,

$$x^\mu \mapsto x^{\mu'} = \Lambda^\mu_{\nu} x^\nu + a^\mu.$$

Figure 4.1 shows a geometric approach to the Poincaré transformation in the form of a Minkowski diagram. Here, the Minkowski diagram is a spacetime diagram without the coordinates  $x$  and  $y$ .

In general, a *four-vector* is defined according to its properties under Lorentz transformations. A contravariant 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 whose components are transformed in the same way as the position coordinates  $(x^0, x^1, x^2, x^3)$ , i.e.,

$$b^{\mu'} = \Lambda^{\mu'}_{\nu} b^{\nu}.$$

A covariant four-vector  $a_{\mu}$  is transformed according to

$$b_{\mu'} = \Lambda_{\mu'}^{\nu} a_{\nu}.$$

The inner product between two four-vectors,

$$a^{\mu} b_{\mu} = \eta_{\mu\nu} a^{\mu} b^{\nu} = \eta^{\mu\nu} a_{\nu} b_{\mu} = a_{\mu} b^{\mu},$$

is a *Lorentz scalar*, i.e., it is invariant under Lorentz transformations. A four-vector is characterized by its square length,  $a^{\mu} a_{\mu}$ . The four-vector is *spacelike* for a negative square length, *timelike* for a positive square length, and *lightlike* if the square length is zero.

Proper time is defined as the time measured by a co-moving clock. Differential proper time,

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

is a Lorentz scalar. The velocity parameters are defined as

$$\beta = v,$$

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

with  $v$  being the absolute value of the velocity. The classical, three-dimensional velocity can be generalized to the four-velocity,

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

The four-velocity is connected to the classical velocity,  $\vec{v} = (v_x, v_y, v_z)$ , by

$$u^{\mu} = (u^0, u^1, u^2, u^3) = \gamma(1, v_x, v_y, v_z). \quad (4.1)$$

Accordingly, the four-acceleration is

$$a^{\mu} = \frac{du^{\mu}}{d\tau} = \frac{d^2 x^{\mu}}{d\tau^2}. \quad (4.2)$$

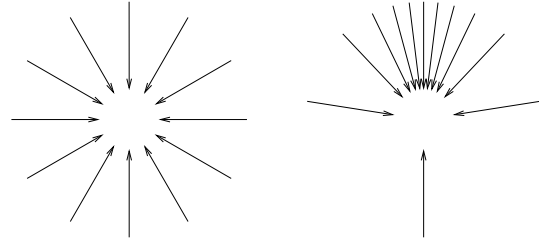
The Lorentz transformation depends on parameters for the separating velocity and the rotation of the two frames of reference. The Poincaré transformation possesses additional parameters for translation. The Lorentz transformation due to a separating velocity is called *Lorentz boost*. The general Lorentz boost, cf. [107, page 69], is described by the symmetric  $4 \times 4$  matrix  $\Lambda^{\mu}_{\nu}$  whose components are

$$\Lambda^0_0 = \gamma,$$

$$\Lambda^0_j = \Lambda^j_0 = -\beta\gamma n^j,$$

$$\Lambda^j_k = \Lambda^k_j = (\gamma - 1)n^j n^k + \delta^{jk},$$

where  $\vec{n} = (n^1, n^2, n^3)$  is the normalized direction of motion,  $\delta^{jk}$  is the Kronecker symbol, and  $j, k \in \{1, 2, 3\}$ . 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.



**Figure 4.2:** Relativistic aberration of light. The left image shows some of the light rays registered by an observer at rest. The right image shows the same light rays with the observer moving upwards at  $\beta = 0.9$ .

## 4.2 Aberration of Light and Doppler Effect

In this section, the Lorentz transformation is applied to the direction and wavelength of a photon. This yields the relativistic aberration of light and the Doppler effect.

The relativistic aberration of light causes a rotation of the direction of light when one is changing from one inertial frame of reference to another. The aberration of light is sufficient to completely describe the apparent geometry seen by a fast moving camera. Figure 4.2 illustrates the aberration of light.

The Doppler effect accounts for the transformation of wavelength from one inertial frame of reference to another and causes a change in color.

The frequency and direction of propagation of a photon are described by the *four-wave vector*

$$k^{\mu} = (\omega, \vec{k}).$$

The normal three-dimensional wave vector  $\vec{k}$  points into the direction of propagation and has length  $k = 2\pi\lambda^{-1}$ , with  $\lambda$  being the wavelength of the photon. The circular frequency  $\omega$  is related to the frequency  $\nu$  of the photon by  $\omega = 2\pi\nu$ . Wavelength and frequency are related by  $\lambda = \nu^{-1}$ . Only light propagation in vacuo is regarded.

Let us consider two inertial frames of reference  $S$  and  $S'$ , with  $S'$  moving with velocity  $v$  along the  $z$  axis of  $S$ . The Lorentz transformation along the  $z$  axis connects frames  $S$  and  $S'$ .

In reference frame  $S$ , consider a photon with circular frequency  $\omega$  and the direction determined by spherical coordinates  $(\theta, \phi)$ , as shown in Fig. 4.3. The respective four-wave vector is

$$k^{\mu} = (\omega, \omega \cos \phi \sin \theta, \omega \sin \phi \sin \theta, \omega \cos \theta).$$

Via the Lorentz transformation, the four-wave vector in frame  $S'$  can be obtained. Moreover, in  $S'$ , the photon is described by the circular frequency  $\omega'$  and the direction  $(\theta', \phi')$ . By comparing the two formulations, one can read off the expressions for the relativistic aberration of light,

$$\cos \theta' = \frac{\cos \theta - \beta}{1 - \beta \cos \theta}, \quad \phi' = \phi, \quad (4.3)$$

and for the Doppler effect,

$$\omega' = \frac{\omega}{D}. \quad (4.4)$$

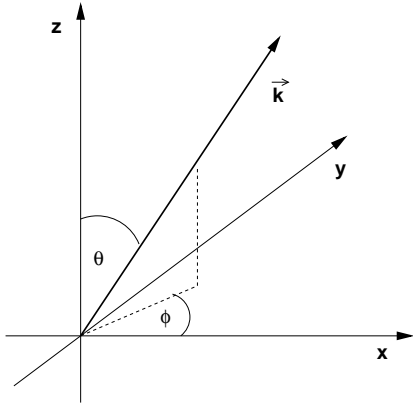


Figure 4.3: A photon with wave vector  $\vec{k}$ .

The Doppler factor  $D$  is defined as

$$D = \frac{1}{\gamma(1 - \beta \cos \theta)} = \gamma(1 + \beta \cos \theta'). \quad (4.5)$$

Expressed in wavelengths, the Doppler effect is

$$\lambda' = D\lambda. \quad (4.6)$$

### 4.3 Searchlight Effect

The searchlight effect is based on the transformation of radiance from one inertial frame of reference to another. The transformation of radiance increases the brightness of objects ahead when the observer is approaching these objects at high velocity.

#### 4.3.1 Transformation of Radiance

Frequency-dependent radiance is transformed from one frame of reference to another according to

$$L'_\nu(\nu', \theta', \phi') = \frac{1}{D^3} L_\nu(\nu, \theta, \phi). \quad (4.7)$$

Similarly, wavelength-dependent radiance is transformed according to

$$L'_\lambda(\lambda', \theta', \phi') = \frac{1}{D^5} L_\lambda(\lambda, \theta, \phi). \quad (4.8)$$

By integrating over all wavelengths, the transformation of radiance is obtained:

$$L'(\theta', \phi') = \frac{1}{D^4} L(\theta, \phi). \quad (4.9)$$

A derivation of these relations can be found in our paper[175] or, alternatively, in Appendix A.

Effects of relativistic illumination in general and the searchlight effect in particular are discussed in the context of special relativistic visualization in our papers[175, 176].

#### 4.3.2 Incident Irradiance from a Point Light Source

For a surface patch on the object, the wavelength-dependent irradiance  $E'_\lambda(\lambda')$  coming from a moving point light source is

$$E'_\lambda(\lambda') = \frac{1}{D^5} \frac{\cos \alpha'}{r'^2} I_\lambda(\lambda), \quad (4.10)$$

with the angle  $\alpha'$  between the vector normal to the surface and the direction of the incident photons, and with the apparent distance  $r'$  of the light source from the surface patch. These quantities are measured in the reference frame of the moving observer, whereas the wavelength-dependent intensity  $I_\lambda$  is measured in the reference frame of the light source. Accordingly, the frequency-dependent irradiance is

$$E'_\nu(\nu') = \frac{1}{D^3} \frac{\cos \alpha'}{r'^2} I_\nu(\nu), \quad (4.11)$$

and the integrated, wavelength-independent irradiance is

$$E^i = \frac{1}{D^4} \frac{\cos \alpha'}{r'^2} I. \quad (4.12)$$

A derivation of these equations is presented in Appendix A or in our paper[175]. Note that, for an isotropic point source in one frame of reference, we get an anisotropic source in the other frame of reference due to the implicit angular dependency in the Doppler factor  $D$ .

### 4.4 Transformation of Photon Field

Because of the relativity principle all inertial frames of reference are equally valid to describe a physical scenario. Therefore, one can choose an inertial frame which is best suited for the respective application. When calculating special relativistic images, one could consider a static scene with a fast moving observer, or a static camera with rapidly moving objects. In this section, I investigate the first approach in more detail. I denote this approach as *exocentric* view because a reference frame outside the camera is adopted. The other approach is considered in Sect. 4.6.

The basic idea of the exocentric approach is as follows. In the first step, all illumination calculations are performed with respect to the outside frame. In the second step, the relevant information about the light reaching the observer is transformed into the frame of the observer. This information is represented by the so-called plenoptic function[1]. The full plenoptic function  $P(x, y, z, t, \theta, \phi, \lambda)$  is the radiance of the light depending on the direction  $(\theta, \phi)$  in spherical coordinates, the spatial position  $(x, y, z)$ , the time  $t$ , and the wavelength  $\lambda$ . Polarization is usually neglected. The plenoptic function is a compact representation of the relevant information about the photon field.

The reduced three-parameter plenoptic function  $\tilde{P}(\theta, \phi, \lambda)$  is defined as the full plenoptic function evaluated at the position  $(x, y, z)$  and time  $t$ .

#### 4.4.1 Transformation of the Plenoptic Function

The transformation of the reduced plenoptic function is determined by the relativistic aberration of light, the Doppler effect,

and the searchlight effect. These effects can be combined to form the transformation of the plenoptic function from  $S$  to  $S'$ :

$$\begin{aligned}\tilde{P}'(\theta', \phi', \lambda') &= \frac{1}{D^5} \tilde{P}(\theta, \phi, \lambda) \\ &= \frac{1}{D^5} \tilde{P}\left(\arccos \frac{\cos \theta + \beta}{1 + \beta \cos \theta}, \phi', \frac{\lambda'}{D}\right)\end{aligned}\quad (4.13)$$

By inverting Eqs. (4.3) and (4.4), the parameters  $\theta$ ,  $\phi$ , and  $\lambda$  have been substituted by terms containing  $\theta'$ ,  $\phi'$ , and  $\lambda'$ .

Usually, the direction of motion is not identical to the  $z$  axis. Therefore, additional rotations of the coordinate system have to be considered before and after the aberration transformation. These rotations are identical to the standard rotations in three-dimensional Euclidean space. By including these rotations, the complete Lorentz transformation of the reduced plenoptic function is obtained. Note that the transformed plenoptic function  $\tilde{P}'$  depends only on the original plenoptic function  $\tilde{P}$  and on the observer's velocity and direction of motion.

For the Lorentz transformation of the complete plenoptic function  $P$ , the respective four-positions at which the plenoptic function is evaluated have to be related by the Poincaré transformation.

#### 4.5 Inter-Frame Relationships for Events

An alternative way of describing a special relativistic image is based on the transformation of emission events between two frames of reference. It makes use of *inter-frame relationships*.

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  $S$ .  $S$  is the frame of the outside world. Frame  $S'$  is the rest frame of the observer. A single point light source which is at rest in  $S$  is considered. Figure 4.4 shows the associated Minkowski diagram. The line  $\{(t, x_e, y_e, z_e) | t\}$  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  $O$ . The emission event  $E$  and the absorption event  $O$  are related by a lightlike connection. Notated with four-vectors, this fact is expressed by

$$(x_o^\mu - x_e^\mu)(x_{o\mu} - x_{e\mu}) = 0, \quad (4.14)$$

where  $x_e^\mu$  denotes the coordinates of  $E$  and  $x_o^\mu$  the coordinates of  $O$  in  $S$ . Once the intersection of the backward light cone with the world line of the light source is determined, the coordinates of  $E$  with respect to frame  $S'$  can be computed. In the diagram this can be done graphically as shown by the dotted lines. This corresponds to the Poincaré transformation from  $S$  to  $S'$ .

In frame  $S$ , the time coordinate of  $E$  can be computed with the use of Eq. (4.14),

$$(x_o^0 - x_e^0) = \sqrt{(x_e^1 - x_o^1)^2 + (x_e^2 - x_o^2)^2 + (x_e^3 - x_o^3)^2}. \quad (4.15)$$

Causality demands to choose only the positive sign of the root, i.e., the intersection with the backward light cone is considered

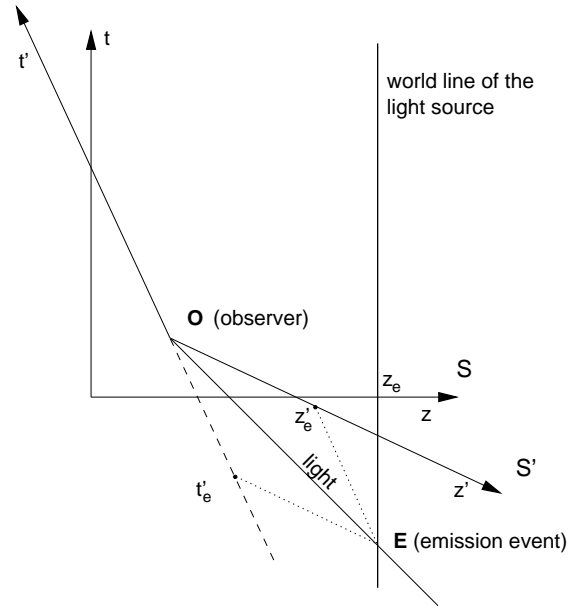


Figure 4.4: Minkowski diagram showing the frame of the observer  $S'$  and the frame of the outside world  $S$ . The light source is at rest in frame  $S$ . The coordinates  $(t_e, z_e)$  of the emission event  $E$  are determined by projection onto the axes of  $S'$ .

and the forward light cone is neglected. With the Poincaré transformation, the coordinates of the emission event in  $S'$  can then be calculated.

Light travels along straight lines in four-dimensional space-time 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.

Here, the direction of the incoming light is identical to the one obtained by the relativistic aberration of light. In the following section, the results of both calculations are matched explicitly. In contrast to the transformation of the plenoptic function in the previous section, the transformation of the emission event provides additional depth information. This difference plays a decisive role in the comparison of the various rendering methods in Chapter 5.

Special relativistic effects on illumination are determined by the Doppler and searchlight effects. These are applied to the spectral power distribution of the incoming light after the transformation of the position of the emission event.

#### 4.6 Equivalence of Exocentric and Egocentric View

One basic feature of special relativity is the absence of a single universal frame of reference and of a universal time. Any inertial frame is equally valid to describe the physical world. Therefore, an egocentric point of view (the camera is at rest and the objects are moving) and an exocentric point of view (the objects are at rest and the camera is moving) are totally

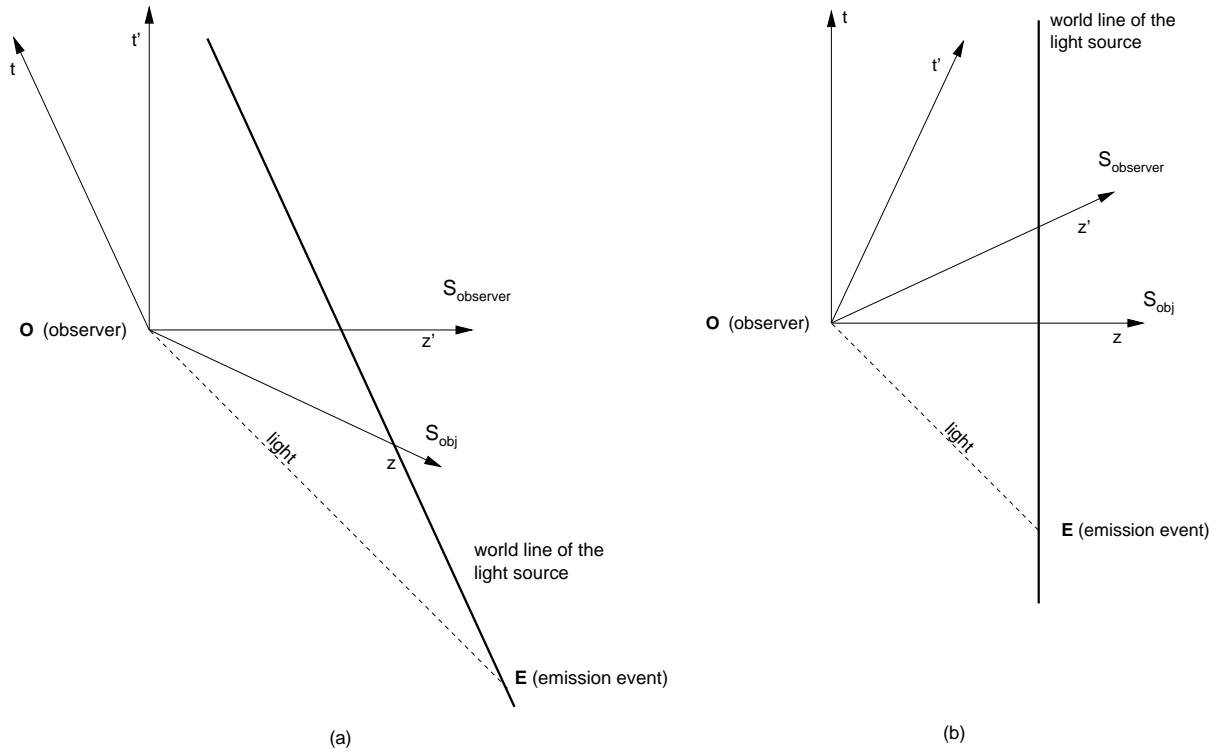


Figure 4.5: Minkowski diagrams for egocentric view (a) and exocentric view (b).

equivalent.

Nevertheless, I would like to explicitly show how both points of view can be matched. Only the issues related to the geometrical appearance are discussed. The Doppler and searchlight effects are neglected because they are usually presented in only one way, equivalent to the exocentric point of view.

The following considerations are based on relations between the observer's rest frame  $S'$  and the objects' rest frame  $S$ . Without loss of generality let the origins of the two frames coincide at the event of image production, the axes be aligned, and the observer be moving with  $\beta$  along the positive  $z$  axis of  $S$ . Then, the two frames of reference are related to each other by the Lorentz boost along the  $z$  axis.

#### 4.6.1 Direction of Incoming Light

First, it will be shown that the direction of the incoming light is identical for both points of view. The light emitted by a single point light source is considered. Figure 4.5 shows the respective Minkowski diagrams.

According to Eq. (4.15), the event of light emission is

$$x_E^\mu = (-\sqrt{x_E^2 + y_E^2 + z_E^2}, x_E, y_E, z_E),$$

in  $S$ , if the light source is located at the spatial position  $(x_E, y_E, z_E)$ . The component  $x_E^0$  reflects the time of flight from the emission event to the absorption event at the camera. Alternatively, the emission event can be expressed in spherical

coordinates,

$$x_E^\mu = (-r_E, r_E \cos \phi_E \sin \theta_E, r_E \sin \phi_E \sin \theta_E, r_E \cos \theta_E), \quad (4.16)$$

with  $r_E = \sqrt{x_E^2 + y_E^2 + z_E^2}$ .

In  $S'$ , the emission event is obtained by the Lorentz transformation,

$$x_E^{\mu'} = (-\gamma(r_E + \beta z_E), x_E, y_E, \gamma(z_E + \beta r_E)).$$

The comparison to the analog of Eq. (4.16) in the observer's frame of reference yields the transformed angles:

$$\cos \theta' = \frac{\cos \theta - \beta}{1 - \beta \cos \theta}, \quad \phi' = \phi,$$

for the direction of the corresponding outgoing light. Therefore, the direction of the light rays is identical for the egocentric and the exocentric point of view, which is based on the aberration equation (4.3).

#### 4.6.2 Visibility

The next concern is the issue of visibility. If one object is hidden by another object in one frame of reference, is it as well hidden in all other frames? Is it allowed to compute visibility in  $S$  and then use the result in  $S'$ ?

Light travels along straight lines in four-dimensional spacetime with respect to every frame. The order of emission events along a light ray is independent of the chosen frame of reference, so the visibility property is not changed by Lorentz transformations.

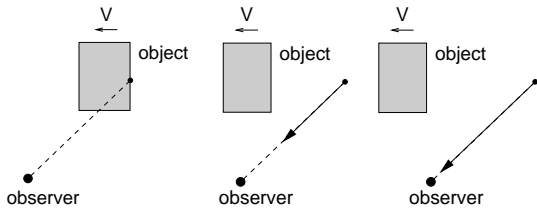


Figure 4.6: Egocentric view, three snapshots for  $\beta = 0.7$ .

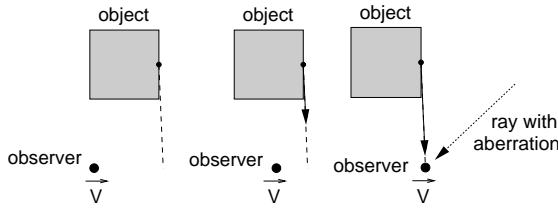


Figure 4.7: Exocentric view, three snapshots for  $\beta = 0.7$ .

The explicit calculation is as follows. Let us consider two emission events  $E_1$  and  $E_2$ . In  $S$ , let  $E_2$  be hidden by  $E_1$ . The respective coordinates are related by

$$x_{E_2}^\mu = C x_{E_1}^\mu, \quad (4.17)$$

with a constant  $C > 1$ . With the use of the Lorentz transformation, the coordinates of the emission events in  $S'$  follow:

$$x_{E_2}^{\mu'} = C x_{E_1}^{\mu'}.$$

Combined with the fact that the aberration formula (4.3) is invertible, the invariance of visibility under Lorentz transformation is proven. Since the inverse Lorentz transformation is just a Lorentz transformation with opposite direction of motion, the invariance of invisibility is valid for the transformation from  $S'$  to  $S$  as well.

After all these abstract calculations, how can it be explained that we are able to look “around” relativistic objects and see their back? Figures 4.6 and 4.7 clarify the situation for the example of a moving cube. In the egocentric view, Fig. 4.6, the camera is at rest and the cube is moving with  $\beta = 0.7$  to the left. The cube is Lorentz-contracted along the direction of motion. Here, the back of the cube becomes visible because the cube outruns the light rays emitted from its back. In the exocentric view, Fig. 4.7, the cube is at rest and the camera is moving with  $\beta = 0.7$  to the right. Here, the back of the cube becomes visible because the observer is behind the cube when she or he is registering the incoming light. The rightmost image in Fig. 4.7 also illustrates the direction of the light ray based on aberration in order to allow a direct comparison to the egocentric view.

### 4.6.3 Apparent Rotation

The apparent rotation of fast moving objects is closely related to the explanation in the previous subsection. In the egocentric view, an object seems to be rotated because light emitted from the normally invisible back of the object is outrun by the object and can thus reach the observer. In the exocentric view, the

observer is already behind the object and can thus see its back. However, from the observer’s point of view the object seems to still be ahead because of the aberration of the incoming light. Seeing the back side of an object is interpreted as an apparent rotation of the object.

### 4.6.4 Summary

It has been shown that the egocentric and exocentric views are equivalent with respect to the apparent geometry in special relativistic rendering. The main difficulty in matching both views is to transform all physical components of the system, especially the position of the observer and the objects.

Usually, the egocentric view is regarded more natural and hence is a widely used model of explanation. In fact, I deem the exocentric view to be more appropriate for the following reasons. First, the exocentric view allows for the transformation of all relevant information about the light field in one operation. Secondly, an accelerated motion of the observer can be incorporated in the exocentric view without any modification, as described in Chapters 5 and 6. Thirdly, the exocentric view better reflects the physical reality. There is no direct interaction between the observer and remote objects. All the information about the surrounding environment is carried to the observer via photons. The generation of a snapshot is based on a local interaction between the photons and the detector (camera). Therefore, it is closer to physics to transform the photons reaching the observer than to transform emission events far away from the camera.

## 4.7 Acceleration of a Point Particle

In this section, the kinematics of a point particle within special relativity is described. Acceleration has to be taken into account, for example, in interaction techniques or in the animation of scene objects.

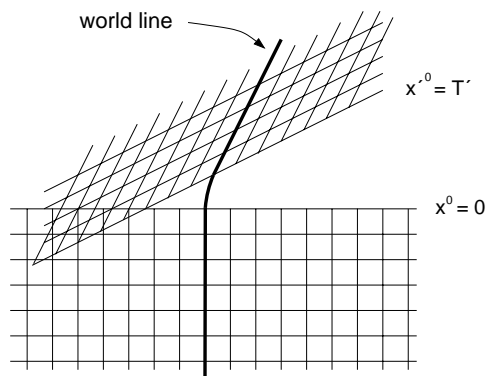
The kinematics of a particle is described by the equations of motion, Eq. (4.2),

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

Frame  $S$  is the inertial frame of the outside world, frame  $S'$  is a co-moving frame with respect to the accelerated particle. Let us assume that the acceleration in the frame of the particle is given. From the three-acceleration  $(a_x', a_y', a_z')$  the four-acceleration in frame  $S'$  is obtained by

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

The four-acceleration is transformed into frame  $S$  by the associated Lorentz transformation. The equations of motion 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  $S$  causes a mixture of space and time components, resulting in a coupled system of ordinary differential equations. There exist well-known numerical methods for solving this initial value problem, for example, the Runge-Kutta method [130]. Finally, the four-velocities and four-positions along the trajectory of the particle are obtained with respect to  $S$ .



**Figure 4.8:** World line of a particle with a short phase of acceleration. The Minkowski diagram displays the coordinate grids of the reference frames before and after the phase of acceleration.

Please note that special relativity is perfectly capable of describing acceleration as long as gravitation is ignored. (Gravitation is the domain of general relativity.) The fact that the kinematics of particles in high energy particle accelerators can be described by special relativity is only one experimental evidence for this statement. In the storage ring HERA (Hadron-Electron Ring Accelerator) at DESY[38], for example, electrons and positrons are accelerated up to an energy of 30 GeV, which yields a velocity of approximately  $\beta = 0.99999999985$ . The circular motion of these particles in the storage ring is caused by acceleration by electromagnetic fields. The calculation of this acceleration and of the collision experiments is based on special relativity and is in perfect agreement with the experiment.

Lorentz transformations are restricted to inertial frames of reference. However, a co-moving inertial frame can be found at every point in spacetime even for an accelerating object. In this way, expressions 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. Figure 4.8 illustrates co-moving frames for a particle before and after a short phase of acceleration.





## Rendering Techniques for Special Relativity

Rendering a single image is a basic part in special relativistic visualization. In this chapter, the generation of a snapshot taken by a relativistic observer is investigated.

In the first two sections, well-known relativistic polygon rendering and relativistic ray tracing are described. Then, the relativistic extension of the radiosity method, texture-based rendering, and image-based relativistic rendering are presented. These three techniques have been developed during the work for this thesis.

### 5.1 Special Relativistic Polygon Rendering

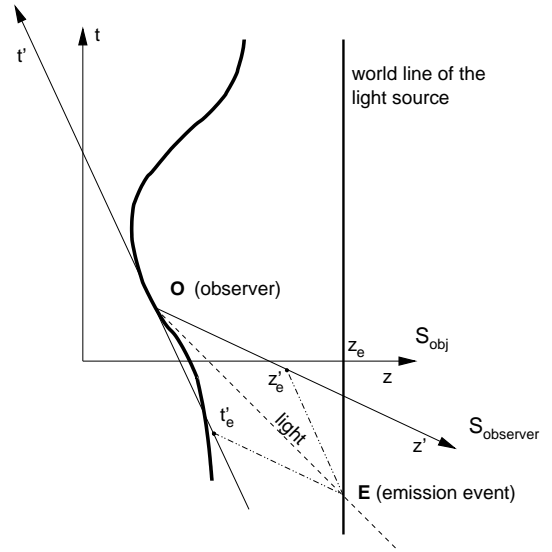
Special relativistic polygon rendering was introduced by Hsiung et al.[79]. They proposed the so-called *time-buffer* for fast visualization of relativistic effects. The time-buffer technique resembles the normal z-buffer and can be mapped onto it. Relativistic polygon rendering is a scan-line method which is based on the apparent shapes of objects as seen by a relativistic observer. Gekelman et al.[57] and Chang et al.[23] investigated the polygon rendering approach in detail and gave a comprehensive presentation. Relativistic polygon rendering utilizes the transformation of emission events according to inter-frame relationships, as described in Sect. 4.5.

#### 5.1.1 Static Scene

First, a simple scenario in which the objects of the scene are at rest relative to each other is considered. The user, i.e., the observer, is moving freely through this virtual world.

Let us investigate the creation 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  $S_{obj}$ .  $S_{obj}$  is the frame of reference of the objects. Frame  $S_{observer}$  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. In Sect. 4.5,  $S_{obj}$  is denoted  $S$  and  $S_{observer}$  is denoted  $S'$ .

First, only a single point light source which is at rest in  $S_{obj}$  is considered. In Fig. 5.1, the associated Minkowski diagram is shown. The line  $\{(t, z_e)|t\}$  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



**Figure 5.1:** Minkowski diagram showing the world line of an accelerated observer with the co-moving reference frame  $S_{observer}$ . The light source is at rest in reference frame  $S_{obj}$ .  $E$  is the emission position of the light reaching the observer in  $O$ . The coordinates  $(t'_e, z'_e)$  of the emission event  $E$  are determined by projection onto the axes of  $S_{observer}$ .

coordinates of  $E$  with respect to frame  $S_{observer}$  are computed by the Poincaré transformation from  $S_{obj}$  to  $S_{observer}$ .

In the following, the primed coordinates are with respect to  $S_{observer}$  and the unprimed coordinates are with respect to  $S_{obj}$ . According to Eq. (4.15), the time coordinate of  $E$  in frame  $S_{obj}$  is given by

$$(x_o^0 - x_e^0) = \sqrt{(x_e^1 - x_o^1)^2 + (x_e^2 - x_o^2)^2 + (x_e^3 - x_o^3)^2}, \quad (5.1)$$

where  $x_e^\mu$  denotes the coordinates of  $E$  and  $x_o^\mu$  the coordinates of  $O$  with respect to  $S_{obj}$ . According to the Poincaré transformation described in Sect. 4.1, the coordinates of the emission event in  $S_{obs}$  can then be calculated. 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, a solid three-dimensional object 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. The relativistic transformation can be applied to all vertices. This yields a completely new object which approximates the emission positions of the surface of the object in the reference frame of the observer. According to Gekelman et al.[57], this virtual surface is denoted *photo-surface*.

Standard computer graphics processing can be used for hidden-surface removal and the projection of the photo-surfaces onto the image plane. Hidden-surface removal techniques such as z-buffer give correct results, since the spatial distance of an emission event is directly related to its temporal distance, cf. Eq. 5.1. In frame  $S_{\text{observer}}$ , the vertices associated with “newer” events will correctly hide vertices associated with “older” ones.

After this geometrical transformation of a vertex the relativistic effects on illumination can also be included by applying the Doppler and searchlight effects. The spectral power distribution associated with the vertex is transformed from  $S_{\text{obj}}$  to  $S_{\text{observer}}$  with the use of Eqs. (4.6) and (4.8). The final RGB colors are determined by Eq. (3.2).

### 5.1.2 Moving Scene Objects and Accelerated Observer

Relativistic polygon rendering can easily be extended to incorporate moving objects. Figure 5.2 shows the Minkowski diagram for a typical situation where two scene objects are moving relative to each other. As long as all scene objects are uniformly moving, the respective world lines are straight. Then, the intersection between the backward light cone starting at the observer and the world lines of the objects can be calculated by a scheme similar to the one described in the previous subsection.

The rendering process is extended as follows. The scene is partitioned into groups of objects which are moving at the same velocity. For each group, the relative velocity with respect to the observer determines the creation of the photo-surfaces. These photo-surfaces are rendered as in the previous subsection. As mentioned before, the depth information of the photo-surface also carries the information about the emission time with respect to the frame of the observer. Therefore, standard hidden-surface removal by the z-buffer gives rise to correct visibility calculations. In this way, arbitrary inertial motions of objects cause no extra computational costs.

Relativistic polygon rendering could be further extended to take accelerating objects into account. However, this implicates two major problems. First, the world line of an accelerating object is bent. The world line has to be represented in a more complex way, for example, by a list of key positions in spacetime. Therefore, intersection computation can no longer be performed according to Eq. (5.1) and becomes much more time-consuming. Secondly, a reasonable model for describing the acceleration of extended objects has to be developed. In Sect. 4.7, only the kinematics of an accelerating point particle is described. The physically correct description of an accelerating extended object is more complicated because the internal forces which hold the object together have to be considered. For extremely high acceleration, the internal forces are too weak to keep the structure of the object stable. Therefore, it might be difficult to develop a model which is based on the

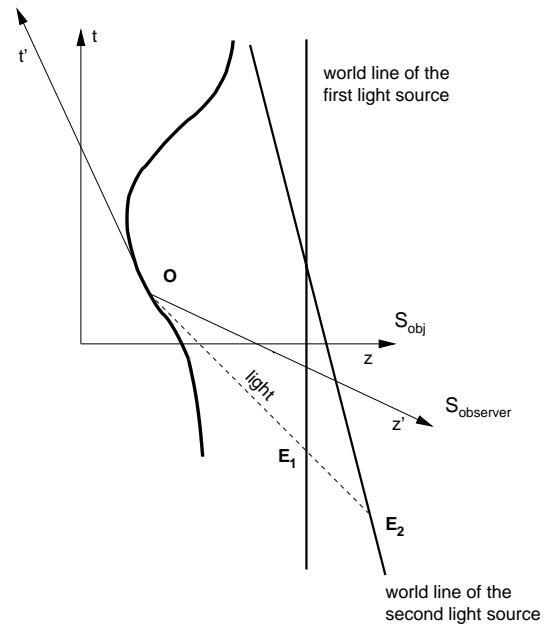


Figure 5.2: Minkowski diagram showing the world lines of two scene objects moving relative to each other.  $E_1$  denotes the emission event on the first object,  $E_2$  the emission event on the second object.

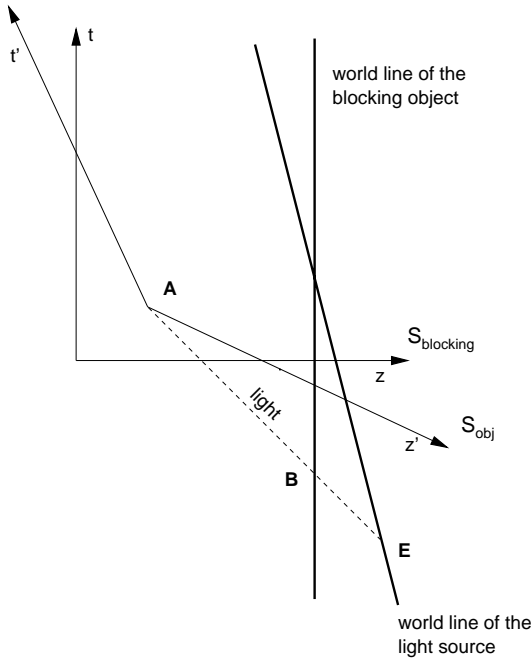
equations of state for standard materials.

Nevertheless, the treatment of an accelerated camera can be included into the relativistic polygon rendering algorithm without any modification because the computation of the trajectory of the camera and relativistic rendering are completely separate processes. In fact, relativistic rendering is not sensitive to acceleration of the camera. The production of a snapshot is only determined by the position, velocity, and direction of motion of the observer and by the standard camera parameters. The rendered image is identical to the image seen by a co-moving observer. However, acceleration affects the motion of the camera and thus its position and speed. Although the generation of a single snapshot is not altered by acceleration, the appearance of a film sequence is heavily influenced by the changing velocity and position of the camera due to acceleration. The acceleration of the observer plays an important role in the discussion of navigation techniques and virtual environments in Chapter 6.

### 5.1.3 Shadowing and Moving Light Sources

In traditional computer graphics, there exist a couple of algorithms to determine shadowing, for example, scan-line generation[5, 18], shadow volumes[32], shadow polygons[7], or shadow z-buffer[181]. However, these algorithms cannot be used in special relativistic rendering because they neglect the propagation delay of light signals and the Lorentz contraction.

Within special relativity, shadows can be correctly calculated by further utilizing inter-frame relationships of spacetime events. This scheme also allows for moving light sources. Figure 5.3 shows the Minkowski diagram for a typical case of shadow casting. The problem is to determine the incoming light for a point on the illuminated surface of an object. The



**Figure 5.3:** Minkowski diagram showing the world lines of a light source and a blocking object. E denotes the emission event at the light source, A the absorption event at the illuminated object. B is the blocking event, i.e., the absorption event at the intercepting object.

respective four-position is called A, as for absorption. Its coordinates are known from the intersection calculations from Sects. 5.1.1 and 5.1.2. The emission event at the light source is denoted E. The object is shadowed at A if an intercepting object is able to block the light from E at a blocking event B. The issue of shadow casting is reduced to the question whether there exists such a blocking event or not. As shown in Fig. 5.3, the illuminated object, the blocking object, and the light source can move at different velocities. They have a finite relative velocity.

The shadow casting algorithm is split in two steps. In the first step, the apparent positions of the light source and the possibly blocking object as seen by the illuminated object are determined. This calculation is analogous to Sect. 5.1.1. The light source and the blocking object may move at different velocities. This is considered in the computation of their respective apparent positions. In the second step, it is checked whether the connecting line  $\overline{AE}$  between the illuminated object and the light source is intersected by the blocking object. This can be done by explicit calculation of the intersection. Another option utilizes a depth buffer (z-buffer) in order to detect whether the apparent position of the light source lies behind the apparent position of the blocking object. The z-buffer method resembles the visibility test as described for moving objects in Sect. 5.1.2.

#### 5.1.4 Local Illumination

In standard computer graphics, the local reflectance properties of a surface are described by the BRDF, cf. Sect. 3.4. The BRDF measures reflection with respect to the rest frame of

the surface. Therefore, standard methods for local illumination can be adopted for special relativistic rendering, as long as all relevant parameters are transformed into the rest frame of the surface. These parameters are the direction of the incident light, the viewing direction (the outgoing light), and the wavelength-dependent radiance.

This transformation has to be performed for every point on the illuminated surface and hence is very time-consuming. Moreover, this transformation is not supported by graphics hardware, so rendering has to be done in software.

However, there are two important special cases which do not require further relativistic lighting transformation. In both cases, the objects of the scene and the light sources are at rest relative to each other. The first case is ambient lighting. Since illumination is independent of the viewing vector and the lighting vector, it does not need a transformation of these vectors. The second case is diffuse reflection of light from infinitely far light sources (directional light). Diffuse reflection depends on the angle between the lighting vector and the normal vector, but is independent of the viewing vector. As long as the light source and the scene object are at rest relative to each other, the angle between the lighting vector and the normal vector is not changed by relativistic transformation.

#### 5.1.5 Visualization of Photo-Surface

In Sect. 5.1.1, the calculation of the apparent position of an object as seen by a moving observer is described. This calculation can be regarded as a map  $R_{\vec{x}_o, \vec{v}_o}$  from the three-dimensional Euclidean space of the scene objects to the three-dimensional space of the photo-objects (photo-surfaces),

$$R_{\vec{x}_o, \vec{v}_o} : \mathbb{R}^3 \longrightarrow \mathbb{R}^3.$$

The map  $R_{\vec{x}_o, \vec{v}_o}$  depends on the position  $\vec{x}_o$  and the velocity  $\vec{v}_o$  of the observer.

The image of a surface under the map  $R_{\vec{x}_o, \vec{v}_o}$  yields the photo-surface. This photo-surface can be used to visualize the generation of relativistic pictures, in addition to the realistic, physical rendering. The photo-surface can be viewed from positions other than the physical position  $\vec{x}_o$ . For correct rendering of the photo-surface, the respective normal vectors are required. Since the map  $R_{\vec{x}_o, \vec{v}_o}$  is differentiable on  $\mathbb{R}^3 \setminus \{0\}$ , the transformation of a normal vector  $\vec{n}$  at the point  $\vec{p}$  is obtained by

$$N_{\vec{x}_o, \vec{v}_o}(\vec{p}) : \mathbb{R}^3 \longrightarrow \mathbb{R}^3, \quad \vec{n} \longmapsto (DR_{\vec{x}_o, \vec{v}_o}(\vec{p})^{-1})^T \vec{n},$$

where  $DR_{\vec{x}_o, \vec{v}_o}(\vec{p})$  is the differential of  $R_{\vec{x}_o, \vec{v}_o}$  at the point  $\vec{p}$ . In general, the transformation of a normal vector is given by the transposed inverse matrix which describes the transformation of the coordinates of the points, cf., for example, Encarnação et al.[46].

The visualization of the photo-surface by adopting an outside view can also be found in our paper[134].

#### 5.1.6 Rendering Pipeline

In the relativistic polygon rendering approach, photo-surfaces are computed and then rendered with standard computer graphics algorithms. This rendering technique can be mapped onto the rendering pipeline used by contemporary graphics hardware. A description of rendering pipelines can be found,

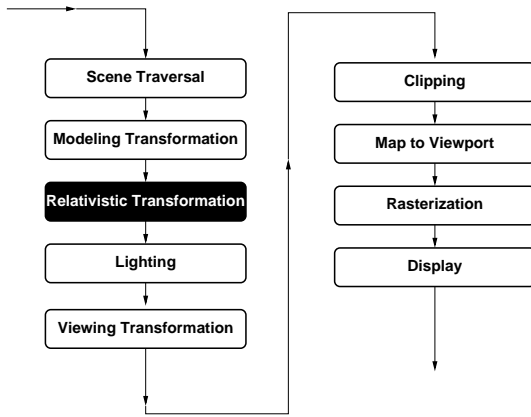


Figure 5.4: Relativistic rendering pipeline for z-buffer and Gouraud shading.

for example, in Foley et al.[51]. Figure 5.4 shows the modified, relativistic rendering pipeline. The black box contains the relativistic extension, the white boxes represent the non-relativistic rendering pipeline. Only the relativistic transformation of the vertex coordinates is added to the original rendering pipeline. Visibility testing is based on the z-buffer, illumination is based on Gouraud shading.

The three-dimensional scene is normally organized in a scene graph. The scene graph is an acyclic directed graph which contains geometry, transform, property, and group nodes. The structure of the scene graph is very well suited for the needs of standard non-relativistic graphics.

However, the relativistic transformation does not commute with standard, affine transformations. Hence, the scene graph has to be “flattened” by expanding it to a tree and applying all modeling transformations. Shared instancing of scene objects is lifted by adding copies of these objects. (Shared instancing occurs for objects having more than one parent in the scene graph). By using a flattened scene tree, the rendering pipeline is simplified and does not require any modeling transformation. Therefore, the relativistic transformation can be computed before the standard rendering pipeline is applied.

Unfortunately, lighting calculation in the hardware-supported rendering pipeline cannot be changed and the full local illumination scheme presented in Sect. 5.1.4 cannot be applied. Therefore, only the two special cases of ambient lighting and diffuse reflection of directional light can be used, cf. Sect. 5.1.4.

### 5.1.7 Discussion

Relativistic polygon rendering can be mapped onto the normal rendering pipeline. It makes use of graphics hardware and allows rendering at interactive frame rates. This makes relativistic polygon rendering most suitable for real-time applications, cf. the implementation of a virtual environment for special relativity in Chapter 6.

However, relativistic polygon rendering suffers from reduced rendering quality compared to the other rendering techniques. The relativistic transformation of the vertices,  $R_{\vec{x}_o, \vec{v}_o}$ , is non-linear. Therefore, artifacts are introduced by the linear connection between the transformed vertices through straight

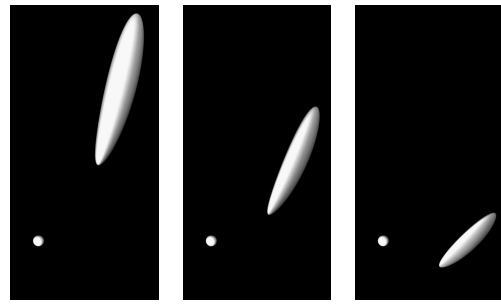


Figure 5.5: Photo-surface of a sphere with  $\beta = 0.95$  at different times. The observer is moving upwards and is symbolized by a small sphere.

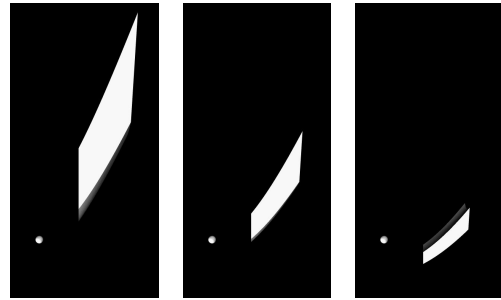


Figure 5.6: Photo-surface of a cube with  $\beta = 0.95$  at different times. The observer is moving upwards and is symbolized by a small sphere.

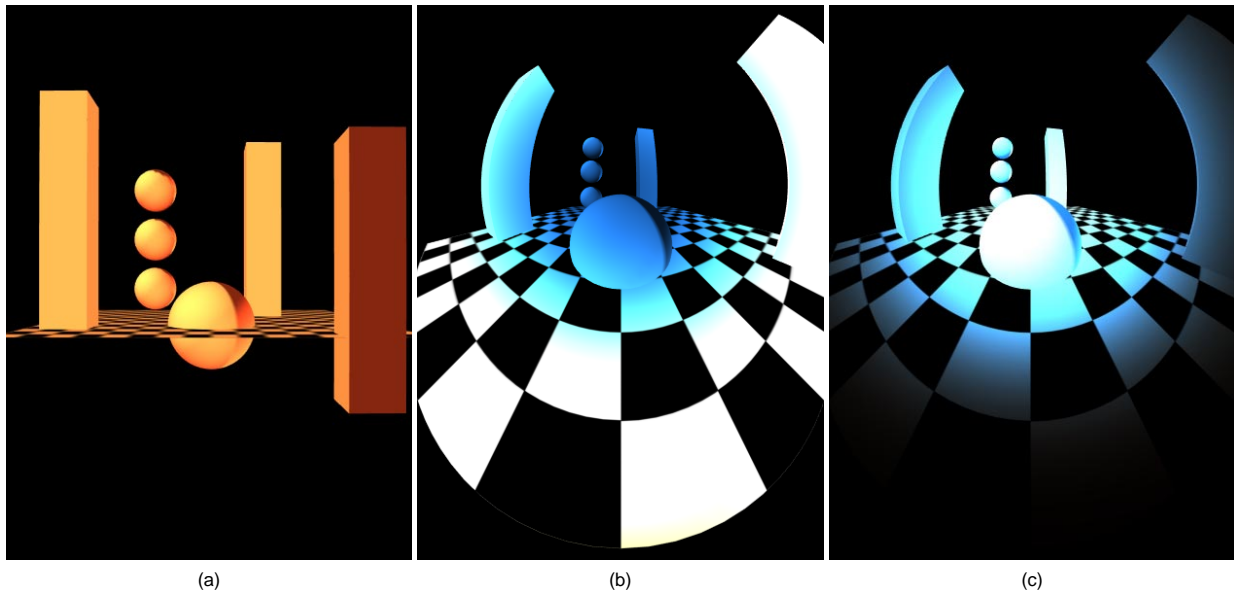
edges. These artifacts, which appear mainly at the boundaries of the objects, can be reduced by a fine remeshing of the original meshes in a preprocessing step or by an adaptive subdivision scheme during runtime. Another shortcoming of relativistic polygon rendering is the restriction to a limited set of illumination models.

### 5.1.8 Implementation and Results

In this thesis, special relativistic polygon rendering is implemented as an object-oriented C++ program. Rendering is based on standard OpenGL 1.1[183]. A detailed description of the system can be found in Appendix C.1. An extension of this implementation which is described in Chapter 6 utilizes multi-processor and multi-pipe architectures and is based on IRIS Performer[137, 42].

Figure 5.7 gives an example of special relativistic rendering. Here, the scene is illuminated by Planck radiation and the objects are colorless Lambertian reflectors. Figure 5.7(b) visualizes the apparent geometry due to the relativistic aberration of light and shows the blueshift due to the Doppler effect. A dominant geometric effect is the increased apparent field of view—the objects seem to move away. Furthermore, straight lines which are perpendicular to the direction of motion become distorted to hyperbolae. It can be seen that the outline of a sphere remains spherical, a fact proven by Penrose[126].

In Fig. 5.7(c), for the transformation of radiance to be displayed, the overall illumination has to be reduced to one thousandth compared to that in Figs. 5.7(a) and (b). Due to the transformation of radiance the objects ahead are extremely bright. This demonstrates that the visual perception of fast



**Figure 5.7:** Visualization of the Doppler effect and the transformation of radiance for a scene which is illuminated by Planck radiation with a temperature of 2800 K. The left image shows a non-relativistic view on the test scene. Picture (b) illustrates the Doppler effect. The viewer is moving with  $\beta = 0.9$  into the test scene. The apparent geometry of the objects is changed according to the relativistic aberration of light. Image (c) shows the same situation with the completely relativistic transformation of wavelength-dependent radiance being included. Here, the overall illumination is reduced to one thousandth of that in (a) and (b).

moving objects is substantially determined by the relativistic effects on illumination.

In addition to realistic, physical rendering, the photo-surface can be used to visualize the generation of relativistic images. The photo-surface can be viewed from positions other than the physical one. Figures 5.5 and 5.6 show the photo-surface of a sphere and a cube. The observer is represented by a small sphere. The photo-objects are stretched along the direction of motion because the effects of the finite speed of light supersede the Lorentz contraction. This outside point of view clarifies the rendering of the relativistic images. Please note that the photo-surface of a sphere is not spherical, although the projection of this photo-surface yields a spherical outline on the rendered images.

## 5.2 Special Relativistic Ray Tracing

Standard non-relativistic ray tracing is performed in three-dimensional space and one may think of relativistic ray tracing as ray tracing in four-dimensional spacetime. This is true for the general approach explained in Sect. 5.2.2.

However, in the special case of static scenes, we can perform the task with ray tracing in three dimensions. This approach to special relativistic ray tracing was proposed by Hsiung and Dunn[76]. Their technique accounts for relativistic effects on the apparent geometry as seen by the observer. Hsiung et al.[78] investigated relativistic ray tracing in more detail and included the visualization of the Doppler effect.

### 5.2.1 Static Scene

First, we restrict ourselves to a simple scenario in which the objects of the scene are at rest relative to each other. The user may move at high velocity through this virtual world. Let us investigate the creation of a snapshot taken by the relativistic observer.

Standard non-relativistic ray tracing is based on a sampling of the image plane. Light rays are traced from the observer through the pixels on the image plane into the scene. The principle is illustrated in Fig. 5.8. Reflection and refraction rays—secondary rays—are recursively examined when rays intersect objects. Various local illumination models are used in order to compute the light intensity on the object surface at every ray-object intersection point. The specific intensities along the rays contribute to the final pixel intensity and color. The visualization process can be split in two main parts: intersection calculation and local illumination.

An object-oriented concept is supposed for the standard three-dimensional ray tracing program in which all relevant parts of the visualization system are represented by objects. The relativistic extension of the original functionality can be included by subclassing.

Figure 5.9 shows the basic structure of such an idealized ray tracer. The image plane is sampled by the `Sample Manager`, which uses the `Projector` to generate a `Ray` corresponding to the considered pixel. The `Ray` communicates with the `Scene` in order to find intersections with scene objects, calculate secondary rays and shadow rays, and determine illumination. Finally, the resulting color is stored in the image by the `Sample Manager`.

The relativistic extension of this ray tracing system is based on the exocentric view, as described in Sect. 4.4. In the first

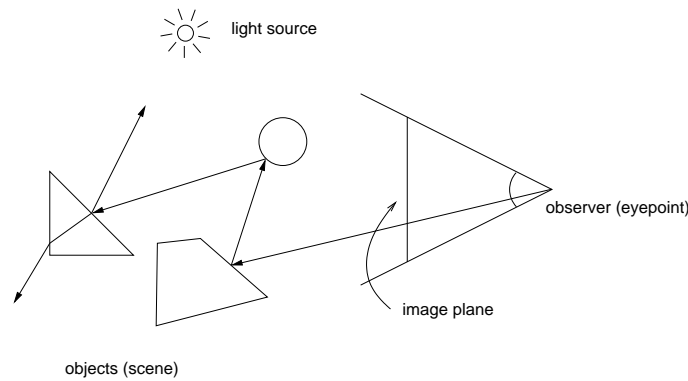


Figure 5.8: Ray tracing principle.

step, the direction of the incoming light ray is determined with respect to the frame of the observer,  $S_{\text{observer}}$ , by the `Sample Manager` and the `Projector`. This calculation is identical to the non-relativistic situation because the observer is at rest in  $S_{\text{observer}}$ . In the second step, the direction of the light ray is transformed from the frame of the observer to the frame of the objects,  $S_{\text{obj}}$ , by applying the aberration of light, Eq. (4.3). The starting point for the ray in  $S_{\text{obj}}$  results from the Poincaré transformation of the original eyepoint according to Sect. 4.1. These extensions are included in the modified `Projector`. In the third step, standard ray tracing is performed with the transformed light ray in  $S_{\text{obj}}$ . In the fourth step, the searchlight and Doppler effects, Eqs. (4.6) and (4.8), are applied to the intermediate result from step three. This transformation is implemented in the `Projector`. In this way, the image as seen by the relativistically moving observer is obtained.

In contrast to relativistic polygon rendering, the relativistic transformation affects the direction of the light ray and not the apparent position of the scene objects. Therefore, the artifacts induced by the non-linear transformation in the polygon rendering approach are avoided in the ray tracing approach. Hence, relativistic ray tracing is very well suited for the production of high-quality images and movies.

## 5.2.2 Four-Dimensional Ray Tracing

So far, only static scenes are supported. The full spatial and temporal information about the scene objects has to be considered in order to include animated and moving objects. This

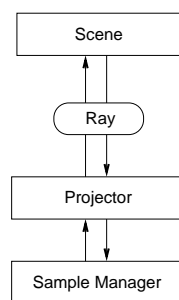


Figure 5.9: Structure of the ray tracing system.

leads to a completely relativistic approach in which ray tracing is performed in flat four-dimensional spacetime. To the author's knowledge, this four-dimensional approach to special relativistic ray tracing has not yet been considered in the literature.

In the object-oriented framework from the previous section, four-dimensional special relativistic rendering requires the following extensions.

First, a standard `Ray` in three dimensions is replaced by a straight `Ray` in four dimensions, i.e. the starting point and the direction of the ray possess one temporal and three spatial coordinates.

Secondly, the ray `Projector` has to generate four-dimensional rays. The four-dimensional starting position (event) and the spatial direction of the ray are known from the camera parameters. The four-direction of a light ray has to be lightlike, thus requiring zero length of the vector. In this way, the temporal coordinate of the direction is fixed.

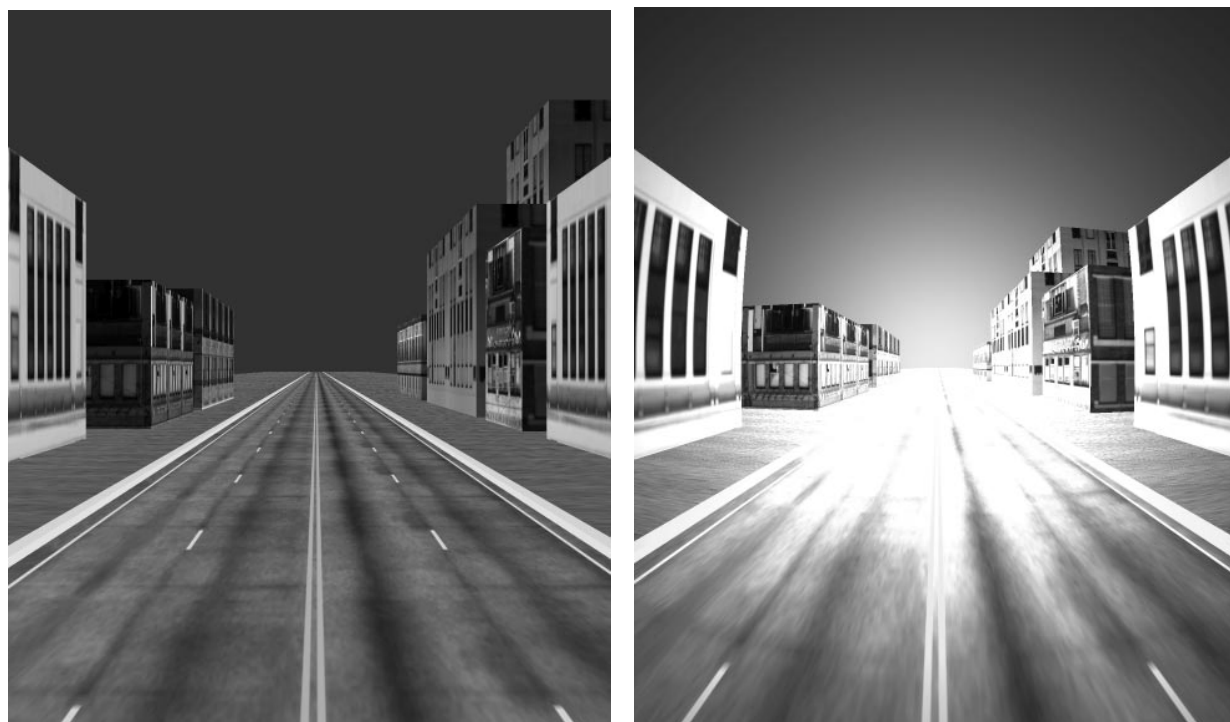
Thirdly, the objects in the `Scene` have to be aware of their motion through spacetime, i.e., they have to know their spatial and temporal positions. The intersection between light rays and objects is computed in four-dimensional spacetime. The intersections correspond to emission, absorption, or transmission events.

In order to calculate local illumination for an intersection event at an object, the direction of the light ray and the spectral power distribution which is transported along the light ray have to be transformed in the reference frame of the respective object. The system has to be aware of the current reference frame in which properties of light are measured.

The proposed concept of four-dimensional special relativistic ray tracing allows for moving and animated objects. In contrast to standard ray tracing, the spatial shape and the motion of an object is not separated, but stored together in the scene graph. Since flat spacetime is a special case of curved spacetime, special relativistic ray tracing is a special case of general relativistic ray tracing. Ray tracing in curved spacetime is investigated in detail in Chapter 9.

## 5.2.3 Implementation and Results

Special relativistic ray tracing is implemented as an object-oriented C++ program. It is based on the non-relativistic ray tracing system *RayViS*[64]. *RayViS* contains a standard ray projection unit which generates light rays corresponding to a



**Figure 5.10:** Special relativistic ray tracing. The left image gives a non-relativistic view on the scene. The right image shows the same scene as seen by an observer moving at  $\beta = 0.8$ , the searchlight effect being included.

given pixel on the image plane. The ray projection unit is subclassed to incorporate the relativistic transformation of light from the reference frame of the camera to the frame of the objects, which allows for static scenes.

The implementation of four-dimensional special relativistic ray tracing is subsumed in the general case of ray tracing in curved four-dimensional spacetime, cf. Chapter 9. In general relativistic ray tracing, four-dimensional descriptions for the light rays, the objects, and the ray projectors are included, covering the extensions for special relativistic ray tracing as well.

A detailed documentation of the implementation of three-dimensional and four-dimensional special relativistic ray tracing can be found in Appendix C.3.

Figure 5.10 gives an example of special relativistic ray tracing. The left image depicts a non-relativistic view on the scene. In the right image, the scene is reproduced as seen by an observer moving at  $\beta = 0.8$ . The aberration of light and the searchlight effect are visualized. The observer is supposed to detect incoming radiation of all wavelengths with equal efficiency. The displayed images result from the measured overall radiance. Hence, the searchlight effect is based on the transformation of wavelength-independent radiance according to Eq. (4.9). The Doppler effect is not visualized in the gray-scale image.

### 5.3 Special Relativistic Radiosity

Non-relativistic radiosity is a global illumination technique which is based on energy conservation or energy equilibrium

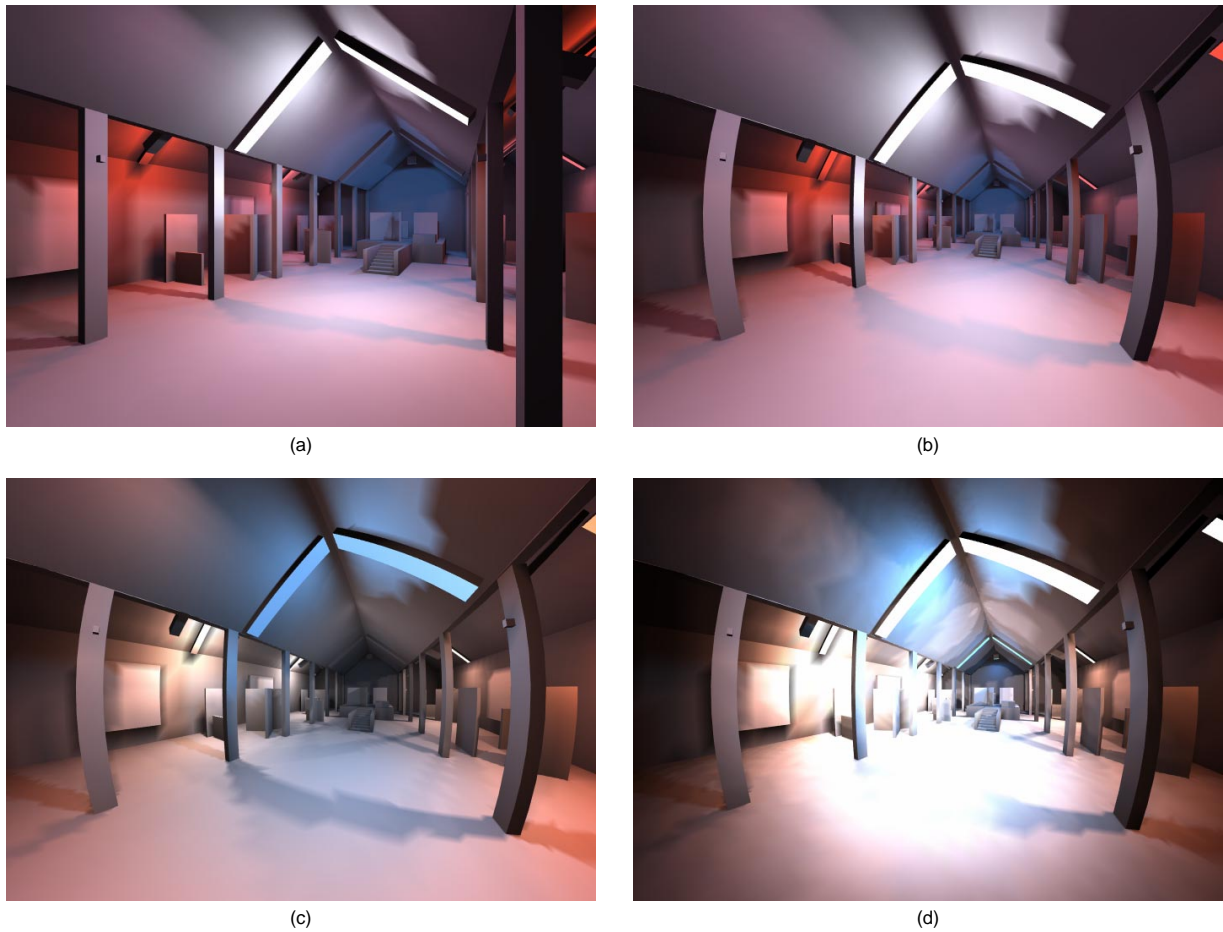
and accounts for the interaction of diffuse light between objects in the scene. The radiosity method was originally developed for image synthesis by Goral et al.[61] and Nishita and Nakamae[118]. An introduction to radiosity can be found, for example, in Cohen and Wallace[27].

In this section, radiosity is extended to incorporate special relativistic rendering. The aspects of illumination are emphasized in order to achieve photo-realism in special relativistic visualization. Here, photo-realism is regarded as the extension of the normal meaning of photo-realism in the sense that a fast moving camera would actually record pictures very similar to the computer-generated pictures.

As mentioned before, computer simulations are the only possible way to visually explore special relativity. In contrast to non-relativistic image synthesis, there is no way of verifying and comparing computer generated relativistic images with reality. Radiosity, however, is an excellent and reliable basis for physically correct visualization of special relativity because radiosity can determine diffuse illumination up to arbitrary accuracy.

#### 5.3.1 Rendering Algorithm

The radiosity method is split in two distinct phases. The first phase employs an object space algorithm, solving for the radiosity at discrete surface patches, and is independent of the viewer position. In the second phase, a renderer computes a particular view from the complete solution. Both a projection method, such as polygon rendering, and ray tracing are suitable for rendering. This splitting in two phases makes radiosity a method most useful for walk-throughs because the



**Figure 5.11:** Special relativistic radiosity. The objects are illuminated by blackbody radiation at temperatures of 2600 K, 5800 K, and 15000 K. Image (a) shows the test scene at low speed, image (b) visualizes apparent geometry at  $\beta = 0.6$ , image (c) adds the visualization of the Doppler effect, and image (d) shows completely special relativistic rendering. In image (d), the overall brightness is reduced to ten percent of that in the other images.

second phase can be supported by graphics hardware.

Only stationary scenes are considered in the relativistic rendering process. Hence, the photon field with respect to the frame of reference of the objects is calculated in the standard non-relativistic radiosity computation. The relativistic extension is restricted to the second phase only, i.e., the view-dependent rendering process. The properties of light are transformed into the frame of reference of the fast moving observer as described in the previous sections. Relativistic effects are computed either by relativistic polygon rendering or relativistic ray tracing.

The radiosity method is based on the assumption that all surfaces are perfectly Lambertian. For this situation, radiance  $L$  and radiosity  $B$  are related by the following equation:

$$\begin{aligned} B &= \frac{d\Phi}{dA} = \int_{\text{hemisphere}} \frac{d\Phi}{dA d\Omega_{\perp}} d\Omega_{\perp} \\ &= \int_{\text{hemisphere}} L \cos \theta d\Omega = \pi L, \end{aligned}$$

with the quantities defined in Chapter 3. In this way, radiosity  $B$ , which is calculated in the first phase of the rendering

process, is directly related to radiance. Hence, the transformation of wavelength-dependent radiance can be applied in the view-dependent rendering phase as described in the previous sections.

### 5.3.2 Implementation and Results

The implementation of the relativistic radiosity approach is based on the non-relativistic radiosity program *RadioLab* [148]. Ray tracing is used for the calculation of form factors, radiosity matrices are solved by Southwell iteration or shooting.

Both ray tracing and polygon rendering are implemented to generate a particular view. Polygon rendering is based on standard OpenGL 1.1[183], which is supported by modern computer graphics hardware. Both Planck spectra with arbitrary temperatures and line spectra at the RGB primary wavelengths can be used. The computation of RGB tristimulus values from blackbody radiation is highly accelerated by calculating and storing tristimulus values for blackbody radiation in a range of temperatures in advance.

Results can be found in Fig. 5.11. The scene objects are il-



**Table 5.1:** Rendering performance for polygon rendering (PR) and ray tracing (RT). All measurements are in seconds per frame.

	PR	RT
non-relativistic	0.25	5.8
relativistic geometry	0.30	5.9
fully relativistic	56.70	169.0
fully relativistic, precalculated tristimulus values	0.42	6.1

luminated by blackbody radiation at temperatures of 2600 K, 5800 K, and 15000 K. Figure 5.11(a) shows a non-relativistic view of the test scene, Fig. 5.11(b) presents the apparent geometry at  $\beta = 0.6$ , and Fig. 5.11(c) adds the visualization of the Doppler effect. The Doppler effect introduces noticeable blueshift. Figure 5.11(d) shows completely relativistic rendering, including the searchlight and Doppler effects. I emphasize that the overall brightness of the last image had to be reduced for reproduction by ninety percent compared to the other images, since the searchlight effect causes an enormous increase of radiance in the direction of motion. This demonstrates the importance of relativistic effects on the illumination model.

Table 5.1 compares rendering performance for polygon rendering (PR) and ray tracing (RT) on a Windows NT workstation with a Pentium II processor (350 MHz) and a Permedia 2 graphics board. Rendering resolution is  $700 \times 700$  pixels. The test scene is depicted in Fig. 5.11 and contains 70,114 patches and 50,939 vertices. Normal non-relativistic rendering is compared to the following relativistic visualizations: geometry only, completely relativistic rendering (including searchlight and Doppler effects), and completely relativistic rendering with precalculated tristimulus values.

The performance measurements show that relativistic rendering causes only minimal extra computational costs if pre-computed tristimulus values are used. Hence, interactive exploration of special relativity via radiosity is possible.

## 5.4 Texture-Based Special Relativistic Rendering

In this section, a texture mapping approach is described as a new means of visualizing the special relativistic effects on both geometry and illumination. Texture-based visualization of apparent geometry was proposed in my paper[169], the extension for relativistic illumination in my paper[170].

The physical basis of this rendering technique consists of the relativistic aberration of light, the Doppler effect, and the searchlight effect, which can be combined in the transformation of the plenoptic function, cf. Sect. 4.4. The rendering method makes extensive use of the functionality available on high-end graphics workstations[109], such as texture objects, pixel textures, and three-dimensional textures. These features are exploited through OpenGL and its extensions.

Unlike the relativistic polygon rendering approach, texture-based relativistic rendering does not transform the coordinates or the color information of the vertices, but transforms the images which are rendered in the normal, non-relativistic way. Therefore, the standard rendering pipeline is not changed and

only an additional step is added at the end of the rendering process. The relativistic transformation is performed on the image plane by texture mapping. This transformation is split in two phases. The first phase determines the geometrical effects by using standard two-dimensional texture mapping. The second phase implements the Doppler and searchlight effects by using pixel textures and three-dimensional texture mapping. These texture operations are supported by modern computer graphics hardware and hence interactive frame rates can be achieved.

### 5.4.1 Idea

The physical basis for texture-based relativistic rendering is the Lorentz transformation of properties of light, cf. Sect. 4.4. Suppose that the photon field is known at one point in spacetime, i.e., at a point in three-dimensional space and at an arbitrary but fixed time. This photon field is measured in one frame of reference which is denoted  $S_{\text{obj}}$ .

Now consider an observer, i.e., a camera, at this point in spacetime. In the following, the generation of a snapshot taken by this camera is investigated. The observer is not at rest relative to  $S_{\text{obj}}$ , but is moving at arbitrary speed relative to  $S_{\text{obj}}$ . However, the observer is at rest in another frame of reference,  $S_{\text{observer}}$ . The photon field can then be calculated with respect to  $S_{\text{observer}}$  by the Lorentz transformation from  $S_{\text{obj}}$  to  $S_{\text{observer}}$ . Finally, the transformed photon field is used to generate the picture taken by the observer's camera.

Now we restrict ourselves to static scenes, i.e., all scene objects and light sources are at rest relative to each other and relative to  $S_{\text{obj}}$ . Here, all relevant information about the photon field—namely direction and wavelength-dependent radiance of the incoming light—can be determined by standard computer graphics algorithms, since the finite speed of light can be neglected in all calculations for the static situation. With this information and with the use of the equations for the relativistic aberration of light and for the Doppler and searchlight effects, the picture seen by the relativistic observer can be generated. In this way, the relativistic effects on geometry, color, and brightness are taken into account.

### 5.4.2 Representation of the Photon Field

The relevant information about the photon field is the direction of the light and the spectral power distribution in the form of the wavelength-dependent radiance. This is the information about the full plenoptic function  $P(\theta, \phi, x, y, z, t, \lambda)$  at a single point  $(x, y, z, t)$  in spacetime, effectively leaving a three-parameter function  $\tilde{P}(\theta, \phi, \lambda)$ . The polarization of light is neglected.

The function  $\tilde{P}$  can be sampled and stored in the form of an image projected onto the unit sphere, with the camera being at the midpoint of the sphere. This image is called *radiance map*. According to Sect. 4.4, the direction of an incoming light ray is determined by the spherical coordinates  $(\theta, \phi)$  of the corresponding point on the sphere.

As described in Sect. 3.3.1, the wavelength-dependent radiance can be represented by a vector with respect to a fixed set of  $n_L$  basis functions. Therefore, each point of the radiance map holds  $n_L$  components for the wavelength-dependent radiance. Here, no projection onto the basis functions is needed. Therefore, the orthonormality condition can be omitted and an

arbitrary set of basis functions can be used. In this representation, the wavelength-dependent radiance is

$$L_\lambda(\lambda) = \sum_{j=1}^{n_L} l_j L_j(\lambda), \quad (5.2)$$

with the set of basis functions,  $\{L_j(\lambda) | j \in \mathbb{N}, 1 \leq j \leq n_L\}$ , and the coefficients of the vector representation,  $l_j$ .

For the display on the screen, three tristimulus values such as RGB are calculated from the wavelength-dependent radiance, cf. Sect. 3.2.4. For example, the RGB values ( $C_R, C_G, C_B$ ) can be obtained by

$$C_i = \int L_\lambda(\lambda) \bar{f}_i(\lambda) d\lambda, \quad i = R, G, B, \quad (5.3)$$

where  $\bar{f}_i(\lambda)$  are the respective color-matching functions for RGB.

How can the radiance map be generated for the non-relativistic situation, i.e., a camera at rest in the frame of the objects,  $S_{\text{obj}}$ ? Since perspective projection is restricted to a field of view of less than  $180^\circ$ , the complete radiance map cannot be created in a single step by using standard computer graphics hardware. Hence, the covering of the whole sphere is accomplished by projecting several images which are taken with differing orientations of the camera. A similar method is used for reflection and environment mapping.

Usually, images do not provide an arbitrary number of channels in order to store the  $n_L$  components for the photon field. However, several radiance maps which contain three standard RGB channels each can be combined to represent the complete photon field.

### 5.4.3 Apparent Geometry

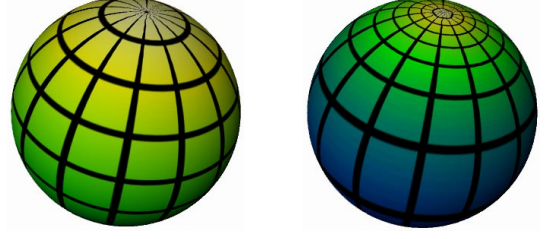
Let us assume that the photon field is known for an observer at rest in the frame  $S_{\text{obj}}$ . Now consider the same situation with the moving observer. The photon field has to be changed by the Lorentz transformation from  $S_{\text{obj}}$  to  $S_{\text{observer}}$ . This transformation is split in two parts—in the relativistic aberration of light and in the transformation of wavelength and radiance. Accordingly, the relativistic part of the rendering algorithm consists of two phases which determine apparent geometry and relativistic illumination.

The relativistic aberration of light yields a transformation of  $(\theta, \phi)$  to  $(\theta', \phi')$  according to Eq. (4.3). The resulting distortion of the image mapped onto the sphere is illustrated in Fig. 5.12.

Usually, the direction of motion is not identical to the  $z$  axis. Therefore, additional rotations of the coordinate system have to be considered. The complete mapping of the coordinates of a point  $(u, v)$  in the original image to the spherical coordinates  $(\theta', \phi')$  of the corresponding point seen by the moving observer is

$$\begin{aligned} T_{\text{mapping}} &= T_{\text{rot, b}} \circ T_{\text{aberration}} \circ T_{\text{rot, a}} \circ T_{\text{proj}} \\ &: [0, 1] \times [0, 1] \longrightarrow S^2. \end{aligned} \quad (5.4)$$

$S^2$  is the unit sphere. The map  $T_{\text{proj}}$  projects the rendered image onto the sphere and determines the corresponding coordinates on the sphere. The coordinates of the pixels are in the interval  $[0, 1]$ . The rotation  $T_{\text{rot, a}}$  considers that the direction of motion differs from the  $z$  axis of the sphere. The actual



**Figure 5.12:** Effect of the relativistic aberration of light on the texture mapped onto the unit sphere. The left sphere shows the mapping without distortion, the right sphere illustrates the distortion for  $\beta = 0.9$ . The direction of motion points towards the north pole of the sphere.

relativistic transformation is accomplished by  $T_{\text{aberration}}$  with the use of Eq. (4.3). Finally,  $T_{\text{rot, b}}$  describes the rotation of the observer's coordinate system relative to the direction of motion.

The relativistic rendering process has to generate the texture coordinates for the mapping of the non-relativistic images onto the unit sphere surrounding the moving observer. With the inverse map  $T_{\text{mapping}}^{-1}$ , the texture coordinates can be calculated from the spherical coordinates  $(\theta', \phi')$  in the coordinate system of the observer. The rotations  $T_{\text{rot, a}}$  and  $T_{\text{rot, b}}$  need not explicitly be calculated, but can be absorbed into the view matrix of the renderer.

### 5.4.4 Relativistic Illumination

The Doppler and searchlight effects account for the relativistic effects on illumination. According to (4.6) and (4.8), both effects depend on the Doppler factor  $D$ . Therefore, in addition to the photon field, the information about the Doppler factors has to be known.

Here, I introduce the term *Doppler factor map*. Analogously to the radiance map, the Doppler factor map holds the Doppler factors for various directions. The Doppler factor map is a one-parameter map because, for a given velocity, the Doppler factor depends only on the angle  $\theta'$ .

With Eqs. (5.2) and (5.3), the tristimulus values  $C'_i$  for each pixel in the frame  $S_{\text{observer}}$  can be calculated by

$$\begin{aligned} C'_i &= \int \bar{f}_i(\lambda') L'_\lambda(\lambda') d\lambda' = \int \bar{f}_i(\lambda') \sum_{j=1}^{n_L} l_j L'_{\lambda, j}(\lambda') d\lambda' \\ &= \sum_{j=1}^{n_L} X_{i, j}(l_j, D), \end{aligned} \quad (5.5)$$

with

$$X_{i, j}(l_j, D) = \int \bar{f}_i(\lambda') l_j L'_{\lambda, j}(\lambda') d\lambda', \quad i = R, G, B, \quad j = 1 \dots n_L. \quad (5.6)$$

The transformed  $L'_j(\lambda')$  are computed from the original  $L_j(\lambda)$  according to (4.6) and (4.8). The  $C'_i$  and  $X_{i, j}$  can be combined to form the three-component vectors  $\vec{C}'$  and  $\vec{X}_j$ .

The function  $\vec{X}_j(l_j, D)$  can be represented by a three-component look-up table (LUT) depending on the two variables  $l_j$  and  $D$ . This LUT can efficiently be implemented by

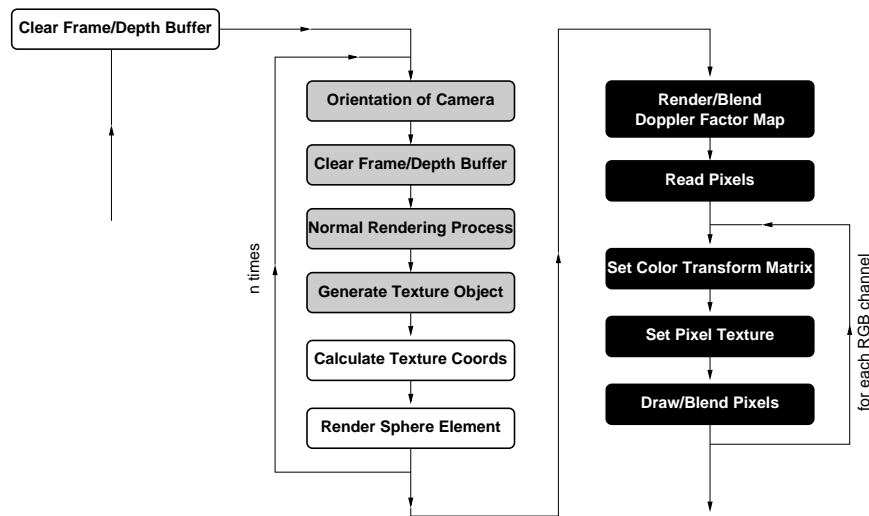


Figure 5.13: Structure of the relativistic rendering process.

using pixel textures. Pixel textures assign texture coordinates on a per-pixel basis instead of a per-vertex basis. Pixel textures are specified in the same way as normal three-dimensional textures. With a pixel texture being activated, all pixels which are drawn from main memory into the frame buffer are interpreted as texture coordinates, i.e., each RGB color triple is mapped into the texture and then the interpolated texture values are actually drawn.

The LUT for each function  $\vec{X}_j(l_j, D)$  is stored in a three-dimensional RGB texture. A two-dimensional texture would be sufficient for a two-parameter function. OpenGL, however, does not support two-dimensional pixel textures. Therefore, a third—a dummy—dimension which is set to one is included. The LUTs do not change for a fixed set of basis functions  $L_j$ . Therefore, the respective pixel textures can be built in a pre-processing step, thus not impairing rendering performance.

Finally, the relativistic transformation of wavelength and radiance is implemented as follows. Another sphere, now textured by the Doppler factor map, is rendered in addition to the sphere resulting from Sect. 5.4.3. The Doppler factors are stored in a previously unused, separate channel, such as the  $\alpha$  channel. The final RGB values are evaluated according to Eq. (5.5) by iterating over the  $n_L$  channels which contain the vectors  $l_j$ , by applying the corresponding pixel textures, and by adding up the results.

If  $n_L$  is greater than three, more than one radiance map is used to hold the  $n_L$  channels, and the whole process above has to be performed several times depending on the number of different radiance maps.

### 5.4.5 Rendering Process

Figure 5.13 shows the structure of the relativistic rendering process. In addition to normal non-relativistic rendering, two phases for the relativistic transformation are appended to the rendering pipeline.

The geometric effects are taken into account in the first phase represented by the white and gray boxes in the diagram. This phase resembles the implementation of reflection or en-

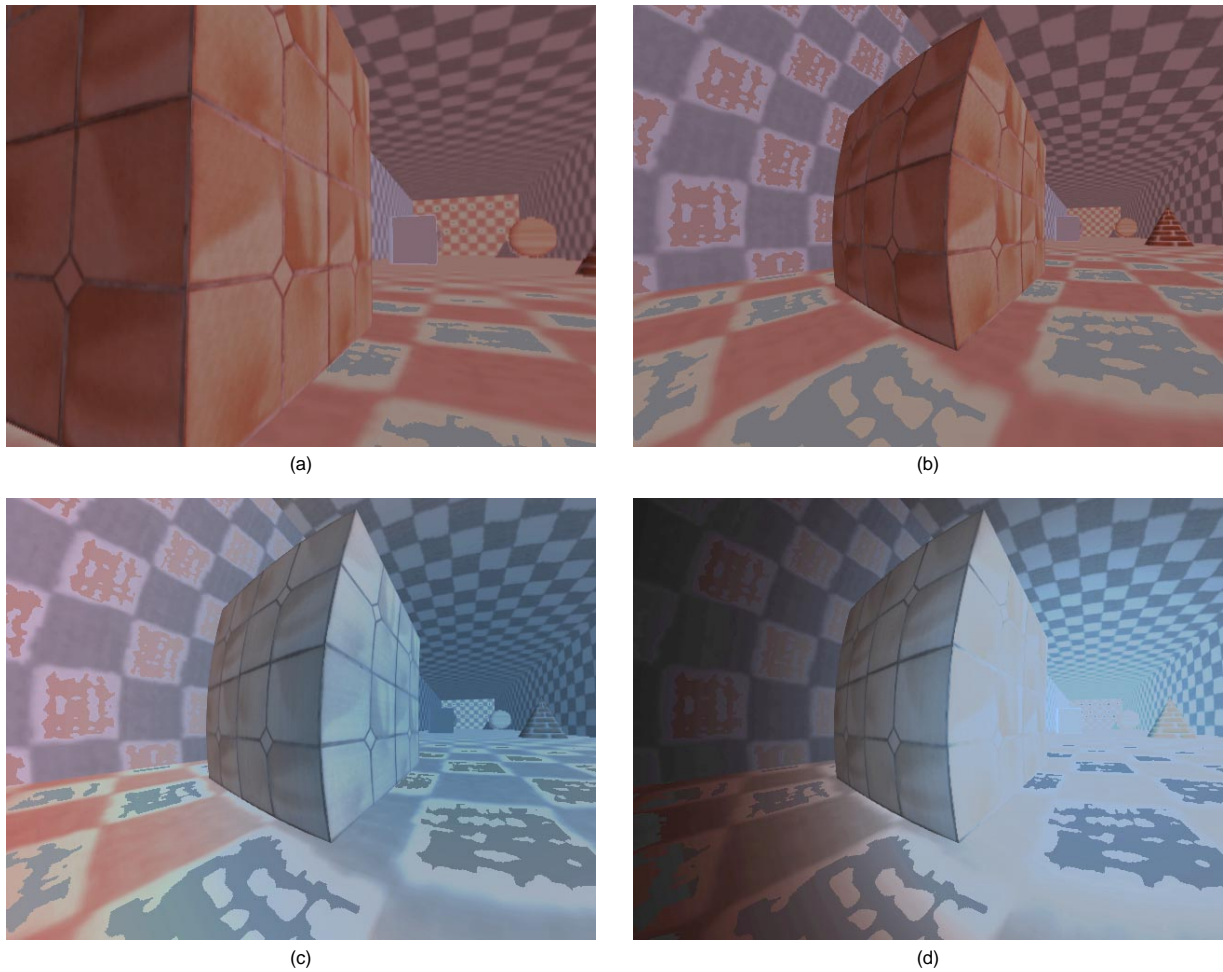
vironment mapping onto a sphere. The main difference is localized in the calculation of the texture coordinates. The operations represented by white boxes work on the back buffer of the normal frame buffer. The operations shown as gray boxes work on an additional, off-screen frame buffer. For example, puffers (SGIX\_pbuffer extension) can be used as off-screen frame buffers.

In the gray part, the textures for a spherical mapping are generated. Here, the standard non-relativistic rendering process is performed  $n$  times;  $n$  is the number of textures mapped onto the sphere and depends on the viewing angle and the orientation that are used for the rendering of the texture images. The OpenGL command `glCopyTexImage2D` transfers the results of non-relativistic rendering from the frame buffer to texture memory. Texture objects (`glBindTexture`) allow fast access to the stored textures.

In the white part, the texture coordinates are calculated with the use of  $T_{\text{mapping}}^{-1}$ . Then, the textured sphere is actually drawn. The relativistic transformation is absorbed into the calculation of the texture coordinates. For rendering the intermediate image which contains the visualization of apparent geometry, a picture is taken from inside the sphere. Note that the viewpoint has to be at the midpoint of the sphere, whereas the orientation is not restricted and allows for viewing into arbitrary directions.

The relativistic effects on illumination are taken into account in the second phase represented by the black boxes. This phase works on the back buffer of the standard frame buffer. Another sphere, now textured by the Doppler factor map, is rendered into the  $\alpha$  channel and blended with the result of the previous rendering steps. Then, the complete RGB $\alpha$  buffer is read into main memory.

The last part is iterated three times, for the respective three color channels which hold the corresponding values  $l_j$ . A color matrix (SGI\_color\_matrix extension) is set, which shifts the current color channel to the red channel and the  $\alpha$  channel to the green channel. Now, the current values of  $l_j$  are stored in the red channel and the corresponding Doppler factors in the green channel. Then, the pixel texture which



**Figure 5.14:** Example of texture-based special relativistic rendering. Image (a) shows a non-relativistic view of the tunnel-like test scene. The scene emits blackbody radiation at 3500 K, 5900 K, and 10000 K. In (b)–(d), the observer is moving towards the end of the tunnel with  $\beta = 0.6$ . Picture (b) shows the visualization of apparent geometry, (c) illustrates the Doppler effect, and (d) presents the complete transformation of illumination. In (d), the overall intensity is reduced to 10 percent of that in (a)–(c) in order to keep the intensities in the displayable range.

transforms  $(I_j, D)$  to displayable RGB values is applied. Finally, the operations in the rasterization stage are performed by drawing the image back into the frame buffer. These results are added up by a blending operation in order to obtain the final image.

So far, the rendering process supports up to three different basis functions  $L_j$ . A larger number of  $n_L$  can be implemented by iterating the whole rendering process several times with varying sets of basis functions and by adding up the results.

#### 5.4.6 Implementation and Results

The relativistic rendering algorithm is implemented in C++ and runs on top of a standard OpenGL renderer. OpenGL version 1.1 with `pbuffer` (SGIX\_pbuffer), pixel texture (`glPixelTexGenSGIX`), and color matrix (SGI\_color\_matrix) extensions is used. The implementation runs on an SGI Octane with Maximum Impact graphics and 250MHz R10000 processor.

A restricted implementation makes use of standard Open-

GL 1.1 only. This version runs on any graphics board with standard OpenGL 1.1. It allows the visualization of apparent geometry only, but not of the Doppler and searchlight effects. Both implementations are described in Appendix C.2.

An example of texture-based relativistic rendering can be found in Fig. 5.14. It illustrates the aberration of light, the Doppler effect, and the searchlight effect. The latter is very prominent and causes an extreme change of brightness.

In the current implementation, a constant number of six textures is used to cover the whole sphere. The six textures originate from the projection of the six sides of a cube onto the sphere. The texture coordinates will only be recalculated if the velocity changes. Three arbitrary basis functions  $L_j$  are supported. The following performance measurements are based on the test scene from Fig. 5.14 and a window size of  $800 \times 600$  pixels.

On an SGI Octane with Maximum Impact graphics and 250MHz R10000 processor, a frame rate of 7.6 fps is achieved for the relativistic visualization of apparent geometry and of 4 fps for the visualization of geometry and illumination. Ap-

**Table 5.2:** Rendering performance on an SGI Onyx2 and on an SGI Visual Workstation 320.

image size	Onyx2		Visual WS	
	512 <sup>2</sup>	1024 <sup>2</sup>	512 <sup>2</sup>	1024 <sup>2</sup>
frame rate (in fps)	7.1	4.9	10.0	5.9
time portion for:				
normal rendering process	50%	38%	59%	52%
transfer frame/tex buffer	21%	41%	11%	26%
others	29%	21%	30%	22%

proximately 30% of the total rendering time is used for rendering the non-relativistic images, 10% for transferring these images from the frame buffer to texture memory, 40% for the pixel operations, and 20% for the remaining tasks.

The restricted implementation, which visualizes apparent geometry only, runs on any graphics board with OpenGL 1.1. Hence, a comparison between different architectures is possible. Table 5.2 shows performance measurements on an SGI Onyx2 system with an InfiniteReality graphics board and a MIPS R10000/195MHz processor, and on an SGI Visual Workstation 320 with a Cobalt graphics board and an Intel Pentium III/500 MHz processor. The time portions for the non-relativistic rendering process and for the data transfer from the frame buffer to texture memory are given. The image and texture sizes are 512×512 or 1024×1024 pixels, respectively.

For the visualization of apparent geometry, rendering speed is limited by the time for generating the non-relativistic images and transferring the non-relativistic images from the frame buffer to texture memory. For typical scenes, the first factor is determined by the pixel fill-rate. The second factor favors the UMA (unified memory architecture) of the Visual Workstation, which holds both texture memory and frame buffer in one universal memory. Texture-based relativistic rendering benefits from a high bandwidth between frame buffer and texture memory.

#### 5.4.7 Issues and Limitations

In the texture mapping approach, the sphere surrounding the observer is represented by a triangle mesh. The texture coordinates which are computed for the pixels inside each triangle by the usual perspective correction scheme differ from the true values. However, these errors do not impair the quality of the relativistic image as long as the angular span under which each single triangle is viewed is not too wide. The errors are independent of the geometry of the scene objects and can be controlled by choosing an appropriate size for the triangles representing the sphere, which is an important advantage over the relativistic polygon rendering approach. In the example depicted in Fig. 5.14, the whole sphere is tessellated by 5120 triangles, guaranteeing a good image quality for velocities as high as  $\beta = 0.99$ .

One problem with texture-based relativistic rendering arises because of the properties of the aberration of light. The aberration equation does not conserve the element of solid angle. Therefore, the relativistic mapping does not conserve the area of an element on the sphere. The image is scaled down in the direction of motion, whereas the image gets magnified

in the opposite direction. This magnification can reveal an inappropriate resolution of the texture. Conversely, aliasing effects can occur in the direction of motion because the texture images are not filtered in the current implementation. These problems could be reduced by adapting the texture size to the relativistic distortion which depends on the observer's velocity and direction of motion. The best solution is anti-aliasing by means of texture filtering. However, it is not clear how filtering could be efficiently implemented on available hardware.

Another problem could be the limited resolution of the RGB $\alpha$  channels, which might cause color aliasing effects whose extent depends on the chosen basis functions  $L_j$  and the interval of the used Doppler factors. These color aliasing effects are usually not very prominent for a depth of eight bits per channel, which is available on current hardware. They should completely disappear on future hardware supporting ten or twelve bits per channel.

## 5.5 Image-Based Special Relativistic Rendering

The special relativistic rendering methods presented so far use a standard geometry-based representation of three-dimensional scenes and thus require cumbersome geometrical modeling and costly rendering.

In this section, an image-based approach to special relativistic rendering is proposed[174]. This approach overcomes problems of geometry-based rendering and has the following important advantages: No three-dimensional geometric modeling is needed, rendering costs are negligible, and photo-realism is easily achieved. Photo-realism and easy modeling make image-based relativistic rendering a method very well suited for video and movie productions.

In Sect. 4.4, it is shown how the plenoptic function is affected by changing frames of reference. All relativistic effects on image generation can be covered by a modification of the plenoptic function. Therefore, the full three-dimensional information about the scene is not required for relativistic rendering. In this framework, only one additional step is appended to the normal non-relativistic rendering pipeline, which is otherwise left unchanged. Therefore, the relativistic transformation can easily be incorporated in all known image-based rendering methods.

### 5.5.1 Previous Work on Image-Based Rendering

A lot of research has been conducted in the field of non-relativistic image-based rendering. *QuickTime VR*[24] is a well-known method for image-based rendering which uses panorama pictures. Other image-based techniques utilize view interpolation[25] or view morphing[145]. Debevec et al.[36] present hybrid geometry and image-based rendering.

Most research on image-based rendering investigates clever methods to acquire, store, retrieve, and/or interpolate the plenoptic function. McMillan and Bishop[101] define plenoptic modeling as generating a continuous five-dimensional plenoptic function from a set of discrete samples. The lumigraph[62] and light fields[93] propose a smart four-dimensional parameterization of the plenoptic function if the scene is constrained to a bounding box. Shum and He[146]

present a three-dimensional plenoptic function which is called concentric mosaics. For a fixed viewpoint, the plenoptic function reduces to a two-dimensional panorama, for example, a cylindrical panorama[24] or a spherical panorama[157].

### 5.5.2 Overview

One basic feature of special relativity is the absence of a single universal frame of reference and of a universal time. Any inertial frame is equally valid to describe the physical world.

Often an egocentric point of view is adopted to derive the properties of relativistic rendering, i.e., the camera is at rest and the objects are moving. Here, I rather take an exocentric point of view. The objects are considered to be at rest and the observer—the camera—is moving at high speed.

The essence of all image-based rendering methods is the evaluation of the plenoptic function, cf. Sect. 4.4. Let us restrict ourselves to a static world in which all objects and light sources are at rest relative to each other and relative to the objects' frame denoted  $S_{\text{obj}}$ . In  $S_{\text{obj}}$ , the plenoptic function can be determined by standard image-based rendering algorithms, since the finite speed of light can be neglected in this static situation.

First, consider the generation of a snapshot taken by a camera at rest in  $S_{\text{obj}}$ . The spatial position of the camera is  $(x, y, z)$  and the time is  $t$ . All the information needed for this snapshot is contained in the reduced three-parameter plenoptic function  $\tilde{P}(\theta, \phi, \lambda)$ , which is evaluated at the respective position and time.

Then, let us bring special relativity back into the game. Consider another observer that is moving relative to the objects. His or her rest frame is denoted  $S_{\text{observer}}$ . This observer is taking a snapshot at the same position and time as the first observer that is at rest in  $S_{\text{obj}}$ . In Sect. 4.4, the plenoptic function for this moving observer is derived.

Once the plenoptic function  $\tilde{P}(\theta, \phi, \lambda)$  with respect to  $S_{\text{obj}}$  is transformed to  $\tilde{P}'(\theta', \phi', \lambda')$  with respect to  $S_{\text{observer}}$ , the normal rendering process can generate the image seen by the fast moving camera because  $\tilde{P}'(\theta', \phi', \lambda')$  is the plenoptic function at rest relative to this camera. (The primed quantities are with respect to  $S_{\text{observer}}$ .) Therefore, all relativistic effects are isolated in the form of the Lorentz transformation of the plenoptic function. The locality property of this transformation allows us to generate relativistic images without knowledge of the depth, or three-dimensional, information about the surrounding scene. Due to the relativity principle the transformation of the plenoptic function can account for both a fast camera and rapidly moving objects.

### 5.5.3 Relativistic Rendering

Image-based relativistic rendering extends the standard non-relativistic techniques by a transformation of the plenoptic function according to the previous section. This extension is located at the end of the rendering pipeline, just before the final image is generated. All other parts of the rendering pipeline are unaffected.

In the following, some variations of relativistic rendering are described. In particular, they address the issue of missing data, since the wavelength dependency of the plenoptic function can usually not be measured. In most cases, data for image-based rendering is acquired by cameras which are

not sensitive to the full power spectrum of the incoming light; standard cameras register light in the visible range only. Furthermore, the information about the recorded spectrum is reduced to tristimulus values such as RGB.

### Completely Relativistic Rendering

If the wavelength-dependent plenoptic function  $\tilde{P}(\theta, \phi, \lambda)$  is provided in the non-relativistic situation, the transformed plenoptic function  $\tilde{P}'(\theta', \phi', \lambda')$  can be computed according to Sect. 4.1. It is important that  $\tilde{P}(\theta, \phi, \lambda)$  is known for an extended range of wavelengths, so that  $\tilde{P}'(\theta', \phi', \lambda')$  can be evaluated for wavelengths in the visible range after Doppler-shifting.

Each pixel on the image plane has corresponding spherical coordinates  $(\theta', \phi')$ , which are transformed to  $(\theta, \phi)$  in the objects' frame. Therefore, a wavelength-dependent radiance  $L'_\lambda(\lambda')$  can be associated with each pixel. For the final display on the screen, three tristimulus values such as RGB have to be calculated from this wavelength-dependent radiance. The RGB values  $(C_R, C_G, C_B)$  can be obtained by

$$C_i = \int L'_\lambda(\lambda') \tilde{f}_i(\lambda') d\lambda', \quad i = R, G, B,$$

cf. Sect. 3.2.4.

### Apparent Geometry

The relativistic effects on the apparent geometry can be visualized by using only a partial transformation of the plenoptic function. Here, only the effects of the aberration of light are taken into account and the searchlight and Doppler effects are neglected, i.e., only the direction  $(\theta, \phi)$  of the incoming light is transformed and all other effects are ignored.

This visualization technique is useful when the full spectral information of the plenoptic function is not available, since this information is not needed for the visualization of apparent geometry. Nevertheless, even this restricted relativistic rendering provides some insight into the special theory of relativity and creates impressive visual effects, as shown in Sect. 5.5.5.

### Reconstruction of the Power Spectrum

In most cases, data for image-based rendering does not comprise the full power spectrum, but only three RGB values. The power spectrum has to be reconstructed from RGB values in order to include the relativistic effects on geometry and illumination, cf. Sect. 3.3. With the reconstructed wavelength-dependent plenoptic function, the fully relativistic image can be generated, as described above.

### Rendering of a Film Sequence

So far, the generation of just a single snapshot has been investigated. But how can a film sequence with a fast camera be produced?

In principle, it works the same way as in the non-relativistic situation. The path of the camera is discretized into a finite set of positions. For every element of this set the plenoptic function is evaluated. Therefore, the plenoptic function has to be known at these positions. Then, the relativistic transformation

is computed and the corresponding image is generated. Finally, a list of snapshots which represent the film sequence is obtained.

For the film to be physically sound, not just the generation of each single snapshot has to be correct, but also the path of the camera itself. As long as the camera is moving uniformly—at constant speed and with a constant direction of motion—the camera is trivially placed at equidistant positions. However, even an accelerated camera can be described by special relativity. In Sect. 4.7, it is shown how the trajectory of an accelerating observer can be computed. Therefore, the positions and velocities of the camera for each snapshot can be calculated, and image-based relativistic rendering can be performed. This method is valid because the generation of a single image is only determined by the position and velocity of the viewer and by the standard camera parameters, but not by the “history” of the trajectory or the acceleration of the observer.

Our everyday experience is based on length scales in the range of meters, time scales in the range of seconds, and velocity scales in the range of meters per second, i.e., the velocities we are used to are approximately eight orders of magnitude smaller than the speed of light. Length  $l$ , time  $t$ , and velocity  $v$  are related by  $v = dl/dt$ . Therefore, one has to change the length, time, or velocity scales in order to notice relativistic effects. For example, the time scales could be reduced to the orders of  $10^{-8}$  seconds. We can think of playing a respective recording at super slow-motion, so that we are able to watch processes which involve time spans of only  $10^{-8}$  seconds. Another option is to artificially reduce the speed of light in vacuo, for example, to walking speed. An instructive illustration of reduced speed of light can be found in Mr Tompkins’ world by Gamow[56]<sup>1</sup>. In the third approach, length scales are increased to the range of light seconds.

The change of scales is the reason why only static scenes are supported. The real-world camera image is recorded using the true values for the length, time, and velocity scales. In particular, the true speed of light is effectively infinite in all practical situations. The relativistic simulation of a dynamic scene would use images which are instantaneously transported from the object to the camera, instead of the correct, retarded images which take into account the reduced speed of light.

#### 5.5.4 Magnification and Anti-Aliasing

The aberration of light does not conserve the element of solid angle. In fact, the infinitesimal solid angle is transformed according to

$$\begin{aligned} \frac{d\Omega'}{d\Omega} &= \frac{\sin\theta' d\theta' d\phi'}{\sin\theta d\theta d\phi} = \frac{d(\cos\theta')}{d(\cos\theta)} \frac{d\phi'}{d\phi} \\ &= \frac{d(\cos\theta')}{d(\cos\theta)} = D^2, \end{aligned} \quad (5.7)$$

with the use of Eq. (4.3) for the aberration of light and Eq. (4.5) for the Doppler factor.

Therefore, the transformation of the plenoptic function causes a magnification opposite to the direction of motion, whereas objects ahead are scaled down. The demand for a

<sup>1</sup>Please note that the illustrations of Mr Tompkins’ world do not show visual perception within special relativity, but only the measurements of Lorentz-contracted lengths.

higher resolution towards the back has to be taken into account when the original data for the plenoptic function is acquired. In the rendering process, the sampled data is accessed by bilinear interpolation.

The image contraction for the front view might cause aliasing effects, especially for extremely high velocities. These effects can be reduced by standard supersampling and postfiltering on the image plane.

Since the sampled plenoptic function can be stored in the form of a two-dimensional texture for the coordinates  $\theta$  and  $\phi$ , anti-aliasing can alternatively be based on texture filtering techniques. Texture mapping can be considered as the process of calculating the projection of a screen pixel onto the texture image—which is called *footprint*—and computing an average value which best approximates the correct pixel color. There exist a couple of filtering methods, of which MIPmapping[182] is the most prominent. This standard technique supports only a quadratic footprint. Hence, it is not very well suited for our application. The mapping by the aberration equation can generate prolate, anisotropic, and distorted footprints because it substantially changes the angle  $\theta$ , whereas it leaves the angle  $\phi$  invariant.

Therefore, techniques which support more complex footprints are required. Summed-area tables[33] (SAT), for example, allow prolate rectangular footprints. The idea of rectangular axes-aligned footprints has been adopted for the relativistic situation. In contrast to SAT, the computation of prefiltered data tables is left out, since each texture is used only once in the rendering process. Filtering provides fair image-quality, even for velocities as high as  $\beta = 0.99$  and for images with high spatial frequencies. Respective examples are shown in the following section. The main advantage of a rectangular footprint over more complex footprints is faster computation and rendering.

Fast footprint MIPmapping[81] is based on quadrilateral footprints and makes use of precomputed MIPmaps and weighting tables. Quadrilateral footprints are an improved approximation compared to rectangular axes-aligned footprints. They support anisotropic, rotated, and distorted footprints. Despite the associated complexity, fast footprint MIPmapping should be able to achieve good rendering performance.

#### 5.5.5 Implementation and Results

The relativistic panorama viewer *Imagine* (IMAge-based special relativistic rendering enGINE) can read panoramas in the *LivePicture* format[94]. This format is similar to *QuickTime VR*, but uses a spherical projection instead of a cylindrical projection. Therefore, a complete  $4\pi$  sterad view is supported.

The interactive viewer is written in C++ and is based on standard OpenGL 1.1[183] and *QGLViewer*[131]. The implementation is due to Kobras[85]. The virtual camera is surrounded by a sphere onto which the panorama texture is mapped. Texture mapping hardware is used to achieve high rendering performance. The relativistic effects on the apparent geometry are implemented by transforming the texture coordinates according to the relativistic aberration of light. The non-interactive part of the viewer uses software rendering to implement completely relativistic visualization by reconstructing the spectrum according to Sect. 3.3. Texture filtering as described in the previous section is not implemented yet.

Another implementation is *Off-Terdingen*, which is an off-screen, batch job-oriented relativistic movie renderer. The im-



**Figure 5.16:** Image (a) shows the non-relativistic view, image (b) the apparent geometry for  $\beta = 0.99$ , and image (c) completely relativistic rendering with  $\beta = 0.3$ . The overall intensity in (c) is reduced to one half of that in (a) and (b) to avoid extreme clamping of the final RGB values.



**Figure 5.15:** Digital video camera mounted on fork arm.

plementation is also due to Kobras[85]. It is able to produce movies of relativistic flights through real-world scenes. It is a C++-based software renderer which stitches and blends series of views taken by different cameras in order to generate a sequence of images for a relativistic flight. The parameters and orientations of the cameras are supplied manually. *Off-Terdingen* provides anti-aliasing by means of texture filtering

with rectangular footprint. Additionally, standard supersampling on the image plane can be applied. The data of the original images is accessed by bilinear interpolation.

Adequate data acquisition for the non-relativistic panorama is an issue, since relativistic image-based rendering demands higher quality of the initial data than standard panorama techniques. First, the resolution of the original images has to suffice the magnification by the aberration formula, Eq. (5.7), when the observer looks into the backward direction. Secondly, a complete  $4\pi$  sterad panorama should be recorded. Most commercial panorama systems, however, are based on cylindrical projection, for example, *QuickTime VR*.

Therefore, a camera system which can automatically film a  $4\pi$  sterad field of view was built. A standard digital video camera is mounted on a fork arm which was originally designed for a telescope. Figure 5.15 shows the fork arm with camera. The fork arm is controlled by a mobile computer. Due to the specific geometry of the fork arm the camera can be placed in a way that avoids parallax artifacts when the camera is turned in different directions. The camera system is DV-based. Images are transferred to the mobile computer via an IEEE 1394 (Firewire) link. The calibrated image data is stitched by *Off-Terdingen* to render spherical panoramas or relativistic views.

Figures 5.16–5.19 provide examples of image-based relativistic rendering. Figure 5.16 shows a long corridor. Figure 5.16(a) provides the non-relativistic view of the scene. Figure 5.16(b) illustrates the effects on apparent geometry when the viewer is rushing into the scene with  $\beta = 0.99$ . A dominant effect is the increased apparent field of view—the objects seem to move away. Furthermore, straight lines which are perpendicular to the direction of motion become distorted to hyperbolae.

Figure 5.16(c) shows completely relativistic rendering with  $\beta = 0.3$ . Here, the power spectrum is reconstructed using the dominant wavelength model. Changes in brightness due to the



searchlight effect and color changes due to the Doppler effect are noticeable. The searchlight effect heavily brightens the image, so the overall intensity has to be reduced to one half of that in Figs. 5.16(a) and 5.16(b) in order to avoid extreme clamping of the final RGB values.

Figure 5.17 shows the apparent geometry for a snapshot of Yosemite Valley at  $\beta = 0.95$ . It exhibits the same effects as in Fig. 5.16, for example the distortion of straight lines to hyperbolae.

Figure 5.18 compares non-relativistic view, apparent geometry, and completely relativistic visualization of an outside scene, analogously to Fig. 5.16. Figure 5.18(c) shows noticeable blueshift due to the Doppler effect and increased brightness due to the searchlight effect.

Figure 5.19 compares filtering and supersampling techniques. This example shows the visualization of apparent geometry at  $\beta = 0.99$ . Figure 5.19(a) is rendered without filtering and supersampling. Aliasing effects are noticeable, especially on the ceiling. Figure 5.19(b) illustrates filtering with a rectangular footprint, as described in Sect. 5.5.4. Aliasing artifacts are attenuated by texture filtering. In Fig. 5.19(c), filtering and  $2 \times 2$  supersampling are combined, yielding a better image quality than mere filtering. This indicates that more sophisticated footprints might improve filtering quality.

The accompanying video presents further examples of relativistic visualization. Parts of the video were shown on TV in a broadcast on Einstein and special relativity[132]. A detailed description of the video can be found in Appendix D.1.3.

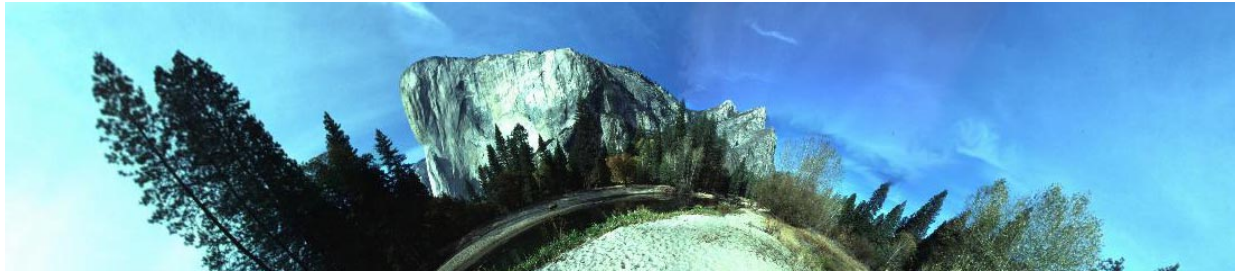


Figure 5.17: Flight through Yosemite Valley with  $\beta = 0.95$ .



Figure 5.18: Image (a) shows the non-relativistic view, image (b) the apparent geometry for  $\beta = 0.9$ , and image (c) completely relativistic rendering with  $\beta = 0.2$ . The overall intensity in (c) is reduced to 10 percent of that in (a) and (b) to avoid extreme clamping of the final RGB values.

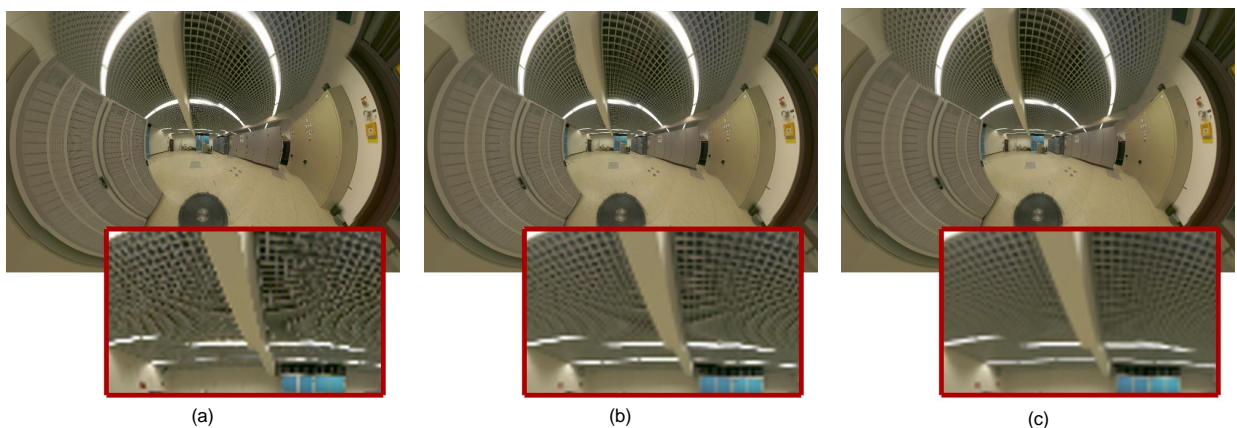


Figure 5.19: Comparison of filtering and supersampling techniques for the visualization of apparent geometry at  $\beta = 0.99$ . Image (a) is rendered without filtering and supersampling, image (b) illustrates filtering with rectangular footprint, and image (c) documents the combination of filtering and  $2 \times 2$  supersampling.

# Virtual Environments for Special Relativity

Computer simulations are the only means of visually exploring special relativity. The intent of this chapter is to present virtual environments which allow the user to submerge into the world of special relativity. They 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.

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

First, a computational model for relativistic rendering is needed. Here, relativistic polygon rendering from Sect. 5.1 is used. An additional step is introduced into the rendering pipeline to account for relativistic effects. A parallel implementation of the relativistic transformation is presented. Therefore, on a multi-processor system, relativistic visualization is achieved at the same frame rates as non-relativistic rendering.

Secondly, a new means of interacting 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 traveling 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 has to be considered in the virtual environment.

The *relativistic-vehicle-control* metaphor and the sophisticated parallelized rendering process are major parts of an immersive virtual environment for special relativity. In Sect. 6.2, a simplified virtual environment is presented. This "relativistic flight simulator" can be regarded as a preliminary stage of the fully evolved immersive virtual environment.

## 6.1 Previous and Related Work

A lot of research has been conducted on issues related to virtual environments. An overview can be found, for example, in

Holloway and Lastra[74] and Earnshaw et al.[41]. Cruz-Neira et al.[34] present the CAVE, a surround-screen, projection-based, immersive virtual environment, which is also a model for the virtual environment in Sect. 6.4.

A number of researchers have addressed issues related to user interaction in general. A survey is given by Hand[68] and Mine[106]. Robinett and Holloway[136] describe the main actions in virtual worlds under control of a manual input device. Hinckley et al.[72] focus on interaction techniques relying on free-space three-dimensional input devices. Jacoby and Ellis[82] give a summary of menus in virtual environments.

This chapter is focused on the issues of navigation and locomotion in virtual environments. Ware and Osborne[164] describe three interaction metaphors which they call *eyeball-in-hand*, *scene-in-hand*, and *flying-vehicle-control*. For the flying-vehicle-control, the velocity of the vehicle is controlled by the displacement of the input device. The flying-vehicle metaphor serves as the basis for the development of the *relativistic-vehicle-control* metaphor presented in this chapter.

Mackinlay et al.[95] present a targeted movement technique which moves the user towards a point of interest with a speed logarithmically related to the distance from the target.

Turner et al.[161] describe the physically based *virtual-camera* metaphor for controlling the camera motion. They describe 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 are adopted for the relativistic situation.

## 6.2 Relativistic Flight Simulator

### 6.2.1 System Design

Since relativistic visualization is inherently connected with motion, a relativistic virtual environment has to provide some kind of fly or drive mode. In this section, a straightforward extension of well-known interaction techniques to special relativity is described. The interactive environment approximates an airplane environment. The aim is to develop a simple "relativistic flight simulator" which can be implemented on a standard graphics workstation or PC.

This flight simulator is a preliminary stage of the more sophisticated immersive virtual environment from Sect. 6.4. The flight simulator is a fishtank-type virtual environment. The user sits in front of the computer monitor and rather looks from the outside onto the virtual world than from the inside of

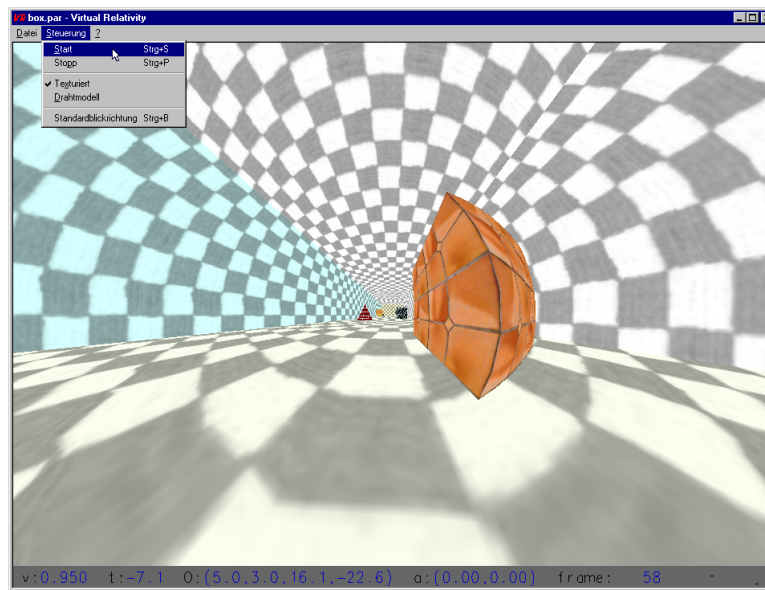


Figure 6.1: Relativistic flight simulator.

an immersive environment such as a CAVE or a head-mounted display.

The following scenario is proposed. The user resides in a fast spaceship which enables him or her to explore a static virtual world. The motion of the ship is controlled by user input. Controls to accelerate and decelerate in the current flight direction are provided. Moreover, the user can change the direction by accelerating to the left or right. Finally, a rotation around the current direction of motion is possible. The screen always points into the direction of motion, i.e., the orientation of the ship is changed by lateral acceleration in a way that the front window is turned towards the direction of motion.

Perception in the relativistic world is very different from our normal experience and thus needs the user's full attention. Navigation is hence based on the well-known principle of flight simulators to make steering as easy as possible.

User interaction, being an essential part of the virtual environment, requires the consideration of the accelerated observer. Since the spaceship is assumed to be small compared to the length scales of the virtual world and the scales of acceleration, it can be regarded as a point particle. Therefore, the description of an accelerating point particle can be adopted from Sect. 4.7. User input provides the three-acceleration of the spaceship with respect to the co-moving reference frame  $S_{\text{observer}}$ , thus the equation of motion (4.2) yields an initial value problem for a system of ordinary differential equations. The initial values are the starting position and velocity of the observer. This system is solved numerically, cf. Sect. 4.7, and provides the trajectory of the accelerated observer parameterized by its proper time.

Relativistic polygon rendering is supported by graphics hardware and achieves interactive frame rates required by real-time simulations. Therefore, this rendering technique is employed in the relativistic flight simulator.

## 6.2.2 Implementation and Results

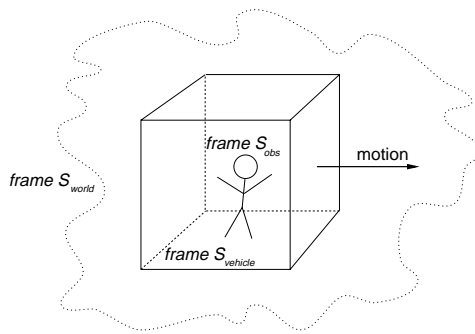
The system is implemented as an object-oriented C++ program. Rendering is based on standard OpenGL 1.1[183] and supports level-of-detail. Arbitrary scenes in the VRML 1.0 file format[12] are accepted. In a preprocessing step, the fine triangulation of the surface patches is performed. The meshing can be controlled interactively and individually for each surface in a radiosity program called *RadioLab*[148]. A detailed program description can be found in Appendix C.1. A previous version of the relativistic flight simulator is described in [134, 167].

Various input devices are supported and were tested. The first kind of input device is the keyboard. On one hand, the steering via keyboard is not very intuitive because it allows input only at discrete steps. On the other hand, keyboards are commonly available. A standard mouse offers more intuitive steering, but is restricted to two degrees of freedom. Spaceball and spacemouse offer three translational degrees of freedom, which can perfectly be mapped to the three directions of acceleration. However, both input devices are not widely available.

In Fig. 6.1, an example of a relativistic simulation in the flight simulator is shown.

## 6.3 Relativistic-Vehicle-Control Metaphor

An immersive virtual environment provides a variety of interaction devices. The interaction techniques developed for the relativistic flight simulator in the previous section can be extended and improved to exploit the advanced technical facilities available in an immersive virtual environment such as the CAVE. Since relativistic visualization is inherently connected with motion, a relativistic virtual environment has to provide



**Figure 6.2:** 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, determine relativistic rendering. 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.

some kind of fly or drive mode. I introduce the *relativistic-vehicle-control* metaphor[171] for interacting with the virtual world. Further on, the elements of this metaphor are presented.

The following scenario is proposed. The user resides in a fast vehicle which enables her or him to explore a static virtual world. Similarly to the previous section, 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. In 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. 6.2.

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  $S_{\text{obs}}$ .

Data from the three-dimensional input device determines the acceleration of the surrounding vehicle. The spatial mid-point of the cube-shaped vehicle is chosen as the reference point of frame  $S_{\text{vehicle}}$ . The acceleration due to user input affects the trajectory of this reference point. The treatment of the accelerated point particle in Sect. 4.7 is adopted for the reference point. The path of the reference point through spacetime can be calculated with the method in Sect. 4.7. It is assumed that the coordinate system  $S_{\text{vehicle}}$  is a co-moving frame with respect to the reference point. In this way, the parameters for the frame of the vehicle,  $S_{\text{vehicle}}$ , are known at every moment in time.

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 Eq. (4.1), the measured three-position and three-velocity determine the four-position and four-velocity with respect to 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 represent the parameters to accomplish relativistic polygon rendering from Sect. 5.1. In 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, cf. Sect. 5.5.3. 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.

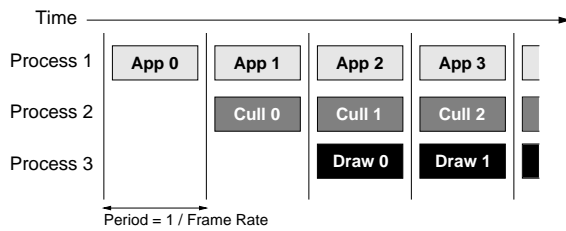
To sum up, the *relativistic-vehicle-control* metaphor consists of two main parts. The first element is a fast virtual vehicle whose motion is governed by the relativistic equations of motion. The second part is the consideration of the user's movement within the vehicle. In the non-relativistic limit, 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.

## 6.4 Immersive Virtual Environment

The CAVE[34] and head-mounted displays are typical examples of immersive virtual environments. The *relativistic-vehicle-control* metaphor directly maps to these immersive environments. Therefore, navigation in the special relativistic virtual environment is based on the *relativistic-vehicle-control* metaphor. In the following, I will particularly consider CAVE-like environments, although the developed interaction elements are applicable to head-mounted displays as well.

Besides camera navigation control, application control plays an important role in a user-interactive environment. Application control is the communication between user and system which is not part of the actual virtual environment. In the proposed environment, application control 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 transformation. Menu choices are selected with a virtual laser pointer. Figures 6.4 and 6.5 show the menu during a typical simulation. In order to save space in the virtual world and avoid unnecessary covering of scene objects, less frequently used options are put into separate submenus which are only displayed when needed.

The menus offer several options. For example, various tracking models can be chosen. First, the physically correct position and velocity tracking can be used, as described above. Secondly, tracking of only the position is supported, which re-



**Figure 6.3:** Multi-processing 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.

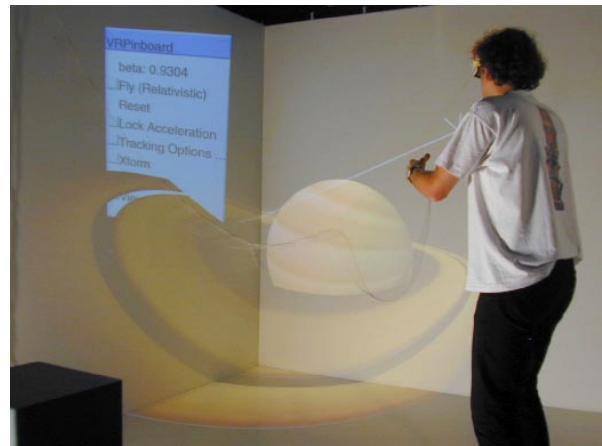
sembles normal tracking techniques. Thirdly, tracking can be disabled, leaving the user with passive locomotion induced by the motion of the vehicle. The miscellaneous tracking techniques can be useful to make step-by-step acquaintance with the relativistic world. In addition to a fly mode with acceleration in all spatial dimensions, a drive mode which suppresses vertical acceleration is implemented. 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”. Moreover, the user can adopt an unphysical outside position in order to visualize the photo-objects, cf. Sect. 5.1.

#### 6.4.1 Implementation and Results

The implementation of the relativistic virtual world runs in the CUBE[71], which is similar to the CAVE 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 an SGI Onyx2 system with 14 R10000 CPUs and three InfiniteReality pipes.

The implementation is an extension of the non-relativistic COVER renderer[133]. Rendering is based on IRIS Performer [42, 137]. Relativistic polygon rendering provides the relativistic extensions. It can be mapped onto the Performer-based rendering pipeline and thus is supported by graphics hardware. 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 multi-processor and multi-pipe support is used. The flow through the rendering pipeline is modeled into application, culling, and draw stages, which are partitioned into separate processes. This enables the three stages to work in parallel. Figure 6.3 shows a diagram of the multi-processor 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



**Figure 6.4:** Immersive virtual environment for special relativity. In this simulation, the user is approaching a model of Saturn at high speed. A virtual menu is located on the left wall.

always handled by a single process.

The relativistic transformation is 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, thus several culling and draw processes in a multi-pipe system can use the same transformed vertices, which are computed only once per frame.

As shown in Fig. 6.3, 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, i.e., each process works on its own copy of the vertex buffer. The flux buffers are cycled through during frame changes.

Depending on the kind of scene and the graphics performance, the relativistic transformation in the application stage might delay the culling and draw stages. Therefore, an additional multi-threading 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 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.

In 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.

Scene data is supplied as a polygonal representation of surfaces. All common three-dimensional file formats are supported. As mentioned in Sect. 5.1.1, a fine remeshing of the surfaces should be employed in a preprocessing step in order to improve rendering quality.

As described in the previous section, the three-velocity of the user is basically calculated by taking the difference of con-

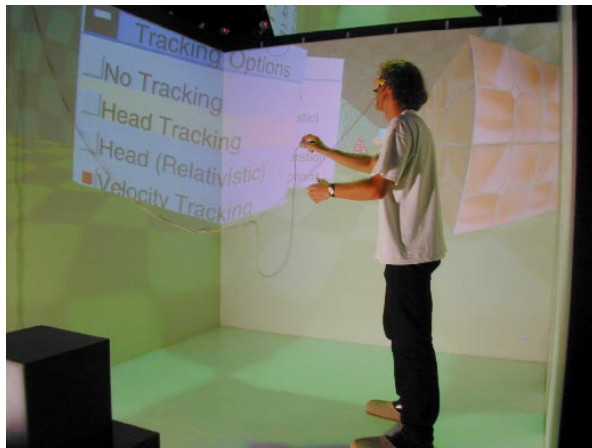


Figure 6.5: Simple box-shaped test scene with menu.

secutive positions measured by a head tracking device. Unfortunately, input from magnetic tracking devices is affected by much 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. The default value for  $n$  is four. The drawback of this filter is a higher latency with respect to direct velocity tracking.

Figures 6.4–6.6 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 or she is increasing the speed towards the objects. Conversely, the objects are apparently coming closer when the speed is being decreased. An example is depicted in Fig. 6.6. The top image in Fig. 6.6 shows the beginning of a phase of acceleration. Using the three-dimensional mouse, the user applies a forward acceleration. The middle and the bottom snapshots are taken at later points in time. Due to the increased aberration of light the background objects become more distorted and seem to move away.

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 some 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 an intuitive means of interaction. The spatial limitations of the CUBE and the missing haptic feedback were found disturbing. Unfortunately, these are technical limitations that cannot be overcome by software design.

Performance measurements for an Onyx2 system with 14

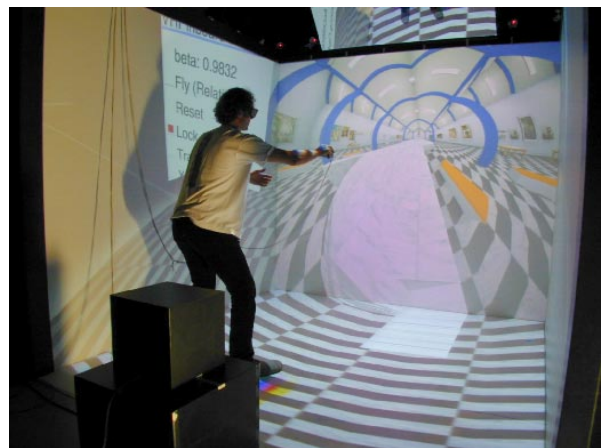
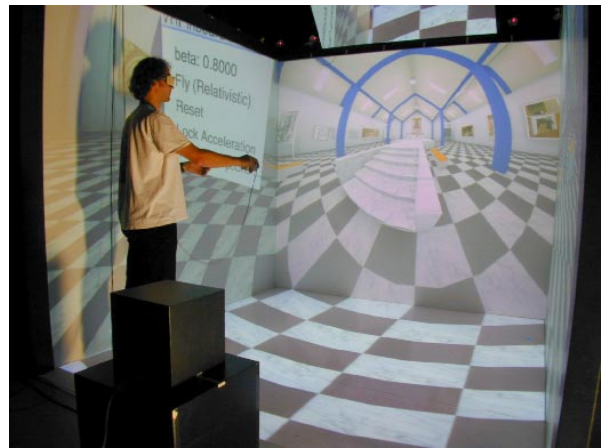


Figure 6.6: Acceleration of the vehicle. The three snapshots are taken during a phase of acceleration. The respective velocities are  $\beta = 0.8$ ,  $0.9188$ , and  $0.9832$ .

R10000/195MHz CPUs are documented in Table 6.1. Two InfiniteReality pipes are used for rendering onto four screens in stereo. The size of a single screen is  $1024 \times 915$  pixels. All frame rates are measured in frames per second. The frame

**Table 6.1:** Rendering performance on an Onyx2 system.

	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

rates for normal non-relativistic rendering, for relativistic rendering with no extra threads for the relativistic transformation, and for relativistic rendering with multi-threading are compared. The last row shows the number of the additional threads needed for the multi-threaded relativistic transformation. The test scenes are depicted in Fig. 6.4 (scene 1), Fig. 6.6 (scene 2), and Fig. 6.5 (scene 3).

The performance measurements 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 large test scene 2 is included to show the performance of relativistic rendering under heavy load. Normally, a minimal frame rate of 15 frames per second should be reached in order to achieve a continuous and smooth progress of images and to avoid dizziness due to high latency.

Appendix D.1.1 describes a video that was recorded in the CUBE during a typical simulation. It gives a good impression of the relativistic virtual environment and, in particular, of the navigation techniques.



## Chapter 7

# Special Relativistic Visualization: Summary and Open Questions

In the first part of this thesis, the visualization of special relativity has been discussed in full detail. The physical and psychophysical basis of color vision has been briefly described in order to establish a firm foundation for the treatment of color in the relativistic rendering methods. A unified physical basis of special relativistic visualization has been presented, with the focus being on the transformation of properties of light, such as the aberration of light, the Doppler effect, and the searchlight effect.

Different types of rendering techniques have been described and implemented. Relativistic ray tracing and relativistic polygon rendering had been known before, whereas the four-dimensional extension of ray tracing, relativistic radiosity, texture-based relativistic rendering, and image-based relativistic rendering have been developed during the work for this thesis. In the next section, the properties of these different rendering techniques are summarized and their advantages and disadvantages are compared.

Previous work was essentially limited to the geometrical appearance of fast moving objects. Except for a few papers related to the Doppler effect, relativistic effects on illumination were completely ignored or treated incorrectly. I have derived all relevant transformation properties of light which are needed for a correct implementation of relativistic shading. It has been shown how all rendering techniques can be enriched to incorporate the visualization of illumination effects. In fact, it has been demonstrated that the visual appearance of fast moving objects is essentially dominated by the Doppler and searchlight effects.

Interaction is the second major topic of the first part of this thesis. Previous work lacked any reference to interaction methods within special relativity. Only still images or movies with uniformly moving observers were presented.

An accelerated observer has been described, based on a solution of the relativistic equation of motion. Acceleration of the observer is a prerequisite for user navigation. I have proposed a simple relativistic flight simulator and an immersive virtual environment for special relativity, both allowing an intuitive approach to special relativity.

The *relativistic-vehicle-control* metaphor has been developed as an interaction technique for immersive virtual environments. The *relativistic-vehicle-control* metaphor is a physically based camera control technique that supports both passive and active locomotion. Relativistic tracking of the position and velocity of the user has been introduced. Furthermore, it has been shown how the rendering pipeline can be ex-

tended to perform the relativistic transformation concurrently to the other parts of the rendering process. In this way, the same frame rates can be achieved for relativistic rendering as for non-relativistic rendering, permitting real-time applications in virtual environments.

In conclusion, I have developed rendering techniques which are improved with respect to performance, image quality, and simulated physical phenomena. Furthermore, I have proposed interaction and navigation techniques adequate for special relativity. The combination of both aspects allows an intuitive, yet physically correct approach to special relativity.

Special relativistic visualization is not only qualified to improve the intuition of physicists and to support education, but has proven to be a most successful means for communicating scientific ideas to the public; the implemented visualization techniques were used to produce images for a number of popular-science publications [19, 20, 22, 97, 112, 113, 116, 129, 162].

### 7.1 Comparison of Rendering Techniques

Relativistic polygon rendering extends the normal rendering pipeline by a relativistic transformation of the vertex coordinates for the triangle meshes representing the scene objects. The relativistic transformation is added to the beginning of the rendering pipeline. Therefore, computer graphics hardware can be utilized for rendering in real-time. The frame rate is usually limited by the floating point performance of the processor due to the transformation of the vertices and by the polygon rate of the graphics board due to a high number of polygons for the finely remeshed surface representations. The relativistic transformation can be parallelized on a per-vertex basis, so the relativistic extensions do not impair performance. The same rendering speeds are achieved for relativistic rendering as for standard non-relativistic rendering on a typical multi-processor and multi-pipe architecture. Since the full three-dimensional structure of the scene objects is preserved, moving and animated objects are possible. Furthermore, an unphysical outside view can be adopted to display the photo-surface, allowing for another type of visualization.

Image artifacts are a disadvantage of relativistic polygon rendering. They are introduced by the linear connection between the transformed vertices through straight edges. The error depends on the angular span under which each single

triangle is viewed and might become very large for objects closely passing by. The artifacts can be reduced by a fine remeshing of the original objects in a preprocessing step or by an adaptive subdivision scheme during runtime. Another problem is the inappropriate handling of relativistic illumination. Only ambient lighting and special cases of diffuse reflection are correctly modeled. Advanced shading techniques cannot be mapped onto the hardware-assisted relativistic polygon rendering pipeline. Furthermore, the Doppler and searchlight effects have to be computed in software and are not supported by graphics hardware. Therefore, the performance advantage is lost for the visualization of relativistic illumination.

In conclusion, the main application of relativistic polygon rendering is real-time simulation of apparent geometry, particularly in virtual environments.

Special relativistic ray tracing can be organized in two subclasses. The first subclass is just an enriched three-dimensional ray tracer taking into account the motion of the camera. The light sources and the scene objects have to be at rest relative to each other. A relativistic three-dimensional ray tracer is easily implemented. Existing non-relativistic systems need only minimal changes in the ray projection unit to allow for the aberration of light and the Doppler and searchlight effects at the position of the observer. The second subclass is a four-dimensional ray tracer taking into account the complete spacetime information for all objects, light rays, and intersection calculations. The implementation is more complex, since all data structures for rays and intersections have to include spacetime coordinates. Four-dimensional ray tracing allows to visualize moving objects.

Both ray tracing approaches generate images of high quality. They do not introduce any relativistic artifacts, as opposed to relativistic polygon rendering. All relativistic effects on illumination can be correctly visualized. The major deficiency of ray tracing are high computational costs, which do not permit real-time applications. However, ray tracing allows easy parallelization, thus reducing the overall rendering times.

To sum up, ray tracing is best suited for the production of high-quality images, in particular, for high-resolution illustrations and movies.

Relativistic radiosity is split in two steps. The first step solves for the radiosity at discrete surface patches in object space and is independent of the viewer position and velocity. In the second step, a renderer computes a particular view from the complete solution. Both polygon rendering and ray tracing are suitable for rendering. Therefore, the areas of application depend on the chosen rendering system.

Radiosity with polygon rendering is ideal for walk-throughs at high velocities. It does not introduce the illumination artifacts that might be present in relativistic polygon rendering, since radiosity is restricted to diffuse materials. In addition, the geometric artifacts are less prominent because a fine meshing of the surfaces is already required for the radiosity computations. Therefore, the main application of radiosity with polygon rendering are interactive walk-throughs and, to some extent, production of illustrations and movies.

Ray-traced radiosity has the burden of high computational costs. Hence, it is essentially limited to the production of high-quality images and movies showing scenes made of diffuse materials.

Texture-based relativistic rendering utilizes the texture-mapping hardware to implement the aberration of light and the Doppler and searchlight effects. In contrast to relativistic

polygon rendering, the relativistic transformation is appended at the end of the rendering pipeline. In this way, every pixel in the image plane is affected by the relativistic transformation.

Texture-based relativistic rendering does not need any modification of the scene or the core rendering method. It does not increase the number of triangles of the scene objects to be rendered. It has no extra computational costs per vertex. It does not introduce the geometric artifacts present in relativistic polygon rendering. Most importantly, a physically correct model for the calculation of illumination is implemented.

The frame rate for the texture approach is limited by the pixel fill-rate, by the operations in the rasterization stage, and by the bandwidth between frame buffer and texture memory and between frame buffer and main memory. Since an increased pixel fill-rate is important for other computer graphics applications such as volume rendering, there will be a continuous effort of hardware developers to achieve higher pixel fill-rates. On the tested hardware, both the texture-mapping and the polygon rendering approaches achieve comparable frame rates, allowing interactive applications.

Relativistic texture mapping, however, requires high-performance graphics hardware. Particularly demanding is the implementation of the searchlight and Doppler effects, which needs pixel texture hardware. Texture-based rendering is limited to the visualization of high-speed motion through static scenes. Moving scene objects cannot be considered. Another drawback are sampling and aliasing artifacts, both for color and geometry. Color aliasing will be significantly reduced by frame buffers with increased color channels depths. Geometry aliasing could be avoided by texture filtering. However, it is not clear how filtering could be efficiently implemented on available hardware. Finally, relativistic texture-mapping cannot be parallelized by using multiple processors, since it is purely based on graphics hardware.

Therefore, texture-based relativistic rendering is most appropriate for interactive applications which visualize special relativistic effects on both geometry and illumination.

Image-based relativistic rendering closes the gap between well-known non-relativistic image-based techniques and relativistic visualization. I have shown how all relativistic effects on image generation can be covered by the Lorentz transformation of the plenoptic function. Therefore, only slight modifications of existing rendering techniques are required to incorporate the physically correct rendering of super-fast objects. Existing methods can be extended for relativistic visualization by adding the Lorentz transformation of the plenoptic function to their rendering pipelines.

The advantages of image-based relativistic rendering are essentially those of standard image-based rendering. Laborious three-dimensional modeling is dispensable; data is acquired by standard cameras. Photo-realism is easily achieved. Rendering performance is high. Image-based relativistic rendering can be applied to both real-world and computer-generated images. In this way, even non-relativistic programs which do not allow modifications of their rendering engine can be used to generate relativistic images.

One limitation of image-based relativistic rendering is the restriction to static scenes. Another issue is the acquisition of image data for large viewing angles, especially, when producing movies.

Image-based relativistic rendering allows to generate photo-realistic images of rapidly moving real-world objects with great ease. Therefore, it is a powerful tool to produce movies

**Table 7.1:** Comparison of rendering techniques.

	RT3	RT4	PR	TBR	IBR
performance	-	-	++	+	+
relativistic illumination	+	+	-	+	+
moving objects	-	+	+	-	-
image quality	+	+	-	o	++
movie production	+	+	-	-	o
interactivity	-	-	+	+	+

and snapshots for both entertainment and educational purposes. Due to high rendering performance, even interactive applications are possible.

The comparison of the special relativistic rendering methods is summarized in Table 7.1. The names of the techniques are abbreviated as follows: RT3 (three-dimensional ray tracing), RT4 (four-dimensional ray tracing), PR (polygon rendering), TBR (texture-based rendering), and IBR (image-based rendering). The qualities of the respective methods are graded from perfect (++), via good (+) and satisfactory (o) to insufficient (-). Relativistic radiosity is missing in the table. It is subsumed in the properties for ray tracing or polygon rendering, depending on the chosen rendering method.

## 7.2 Future Work

Future work might cover further development and improvement of rendering techniques. The performance of relativistic polygon rendering could be increased by adopting methods from non-relativistic rendering, such as occlusion culling and multi-resolution algorithms. Moreover, the implementation of an adaptive subdivision scheme would reduce the artifacts which are caused by the non-linear relativistic transformation of the vertices.

In the texture-mapping approach, the issue of rendering performance and image quality could be addressed as well. A limiting part in the rendering process is the pixel fill-rate for the generation of the radiance map. The number of textures can be reduced if only that part of the sphere actually viewed by the relativistic observer is covered by textures. In addition, the resolution of the textures could be adapted to the magnification by the aberration of light. This will increase rendering speed and enhance image quality. Texture filtering is highly desirable, but it is not clear how filtering could be supported by existing hardware. The situation might change with the development of improved graphics boards.

Image-based relativistic rendering lacks a practical system for data acquisition in movie productions, which also is the case for most other image-based rendering methods. The developed robot-based camera system is too slow to be useful for sampling panoramas along a camera path. The problem can be approached twofold. The first approach targets at an improvement of the data-collecting apparatus. Either a system of several miniature CCD cameras could be coupled to cover a large field of view, or an optic which widens the field of view to an appropriate extent could be employed. The second approach utilizes sophisticated reconstruction methods which allow to interpolate between panoramas taken at widely separated camera positions.

The following ideas could be pursued to improve the virtual environments and the interaction techniques. A second, exocentric view on the virtual world could be helpful to improve navigation and enrich the physical understanding. Similarly to the world-in-miniature (WIM) technique by Stoakley et al.[152], an outside view which is displayed in addition to the egocentric view could be investigated. In addition, space exclusion, i.e., collision detection, could serve as a visual feedback technique to improve navigation.

A challenging task is the investigation of the acceleration of extended bodies. A physical model for the accelerated motions of such objects would lead to a physics-based implementation of animation within special relativity.



## **Part II**

# **General Relativistic Visualization**



# Introduction to the Visualization of General Relativity

In Einstein's general theory of relativity, the geometrical properties of the four-dimensional manifold of space and time are determined by gravitation. A common problem in general relativity is that many terms depend on the chosen coordinate system and do not have a direct physical interpretation. For example, this is the case for the spatial and temporal coordinates or the metric itself. Therefore, I specifically focus on visualization techniques which represent geometrical properties and are independent of coordinate systems.

The first part of this thesis has exclusively dealt with the visualization of special relativity, i.e., flat spacetimes, which cannot describe gravitation. This second part extends visualization techniques to curved spacetimes of general relativity, which include gravitational phenomena.

In this thesis, the visualization of special relativity is restricted to the display of visual perception by a fast moving observer. The visualization of general relativity covers a wider field of principal visualization techniques because the geometry of curved spacetimes is much more complex than the geometry of the flat Minkowski space. The basic visualization techniques are the following.

First, the direct visual appearance of objects under the influence of a gravitational field is investigated. This is in the tradition of special relativistic visualization. Ray tracing in four-dimensional curved spacetime is the state-of-the-art technique to visualize the appearance of objects within a gravitational field. It reverses the way of the light rays by tracing them from the observer back in time into a scene of objects. In this context, I specifically address gravitational lensing effects and caustics caused by gravitational fields.

Secondly, the geometry of curved spacetime itself is visualized. Here, not the full four-dimensional spacetime is considered, but only a two-dimensional spatial hypersurface. The geometry of such a two-dimensional surface is displayed by embedding in flat three-dimensional space.

Thirdly, classical visualization is pursued. In the context of this thesis, classical visualization means the visual representation of data from relativistic simulations without taking into account the curvature of spacetime. Techniques known from other fields of scientific visualization can be adapted in order to illustrate quantities specific to relativistic systems.

## 8.1 Previous Work

In the physics and computer graphics literature, there are some articles concerned with the appearance of objects under the influence of gravitational light deflection. Usually,

only well-known metrics are investigated, which are provided in closed form. The first example is the Schwarzschild solution for a spherically symmetric static distribution of matter. For example, Nollert et al.[121] and Ertl et al.[48], investigate the appearance of a neutron star under orthographic projection. A graphically more sophisticated visualization of the Schwarzschild metric is presented by Nemiroff[114]. Kraus[88] also considers light deflection around a neutron star by means of ray tracing. Nollert et al.[120] give a comprehensive presentation of general relativistic ray tracing. Gröller[63] proposes a generic approach to non-linear ray tracing as a visualization technique.

Bryson[21] presents a virtual environment for the visualization of geodesics in general relativity. He shows examples of the Schwarzschild and Kerr solutions. This approach is different from ray tracing because geodesics are not used to provide color information for pixels on the image plane, but the whole trajectory of a photon in spacetime is displayed. This technique has the disadvantage of depending on a chosen coordinate system.

Classical visualization of relativistic data is investigated by Lehle[92] in detail. Since classical visualization makes use of standard techniques from other fields of scientific visualization, there exists ample previous and related work, cf., for example, the textbook by Nielson et al.[117].

## 8.2 Outline

In Chapter 9, general relativistic ray tracing is described, including the basic physical and mathematical aspects of light propagation in curved spacetimes. I present a generic object-oriented relativistic extension to standard ray tracing in three-dimensional flat space. The focus is on the suitability of ray tracing as a visualization technique in gravitational research; two explicit examples are shown in detail—namely the visualization of the rigidly rotating disk of dust and the warp drive. Finally, an extension to the visualization of non-trivial topologies is proposed. The spacetime of a simple wormhole is implemented as an example.

The following chapter specifically deals with gravitational lensing. I present a brief overview of standard lensing theory and some examples of visualization of standard lensing. An extension to ray tracing is developed to incorporate visualization of standard lensing.

Then, two alternative means of visualizing gravitational lenses are proposed. The first technique directly utilizes ray tracing in order to show how a gravitating body affects the ap-

pearance of background objects. Images and results are compared to standard methods from image synthesis in gravitational lensing. A second visualization technique is proposed which allows to find caustic surfaces and thus some characteristic properties of light propagation under the influence of gravitation.

In Chapter 11, the visualization of the inner geometry of two-dimensional surfaces is investigated. A numerical method for the isometric embedding of surfaces with spherical topology is described. The application of this visualization technique to data from numerical relativistic simulations is demonstrated.

Chapter 12 deals with classical visualization in general relativity. A well-known visualization technique—height fields—is improved to achieve better image quality for a specific application in the simulation of neutron stars.

Chapter 13 summarizes the second part of the thesis and gives an outlook on possible future work.



## Chapter 9

# General Relativistic Ray Tracing

Gravitation determines the geometrical properties of the four-dimensional manifold of space and time. As a consequence of the influence of the gravitational field created by a massive object, for example a black hole, the propagation of light is considerably affected. Conversely, the study of the properties of light rays conveys information about the underlying geometrical structures.

In this chapter, ray tracing in four-dimensional curved spacetime is presented as a technique to visualize the appearance of objects under the influence of gravitation. Large sets of light rays are considered and represented in a very compact form in order to give an intuitive and geometric approach. Ray tracing is based on geometric optics. It reverses the way of the light rays by tracing them from the observer (camera) back in time into a scene of objects, where the bending of light rays has to be considered.

A common problem in general relativity is that many terms depend on the chosen coordinate system and do not have a direct physical interpretation. Although an abstract geometric and coordinate-free formulation of curved spacetime is feasible, most calculations and numerical simulations are based on coordinates. A major advantage of ray tracing is its independence of the coordinate system, even though the underlying data is given in specific coordinates. The final images are results of a *gedanken experiment*: what would an observer see, what would a camera measure? Hence, the images have an immediate physical meaning and are coordinate-free. The resulting pictures are observables and independent of the chosen coordinate system. This is an important feature and advantage in the realm of general relativity.

General relativistic ray tracing is presented as a tool in gravitational physics in my paper[172].

## 9.1 Theoretical Background

In this section, only a very brief overview on the mathematical and physical background of light propagation within general relativity can be given. A detailed presentation of general relativity can be found, for example, in Misner et al.[107] or Weinberg[166].

### 9.1.1 Spacetime as a Manifold

General relativity describes spacetime as a curved manifold. The mathematics of differential geometry provides a framework for such curved manifolds.

A manifold is described by a collection of coordinate *charts*. A coordinate chart is a map from an open set in the manifold to an open set of the coordinate system, which provides a way of expressing the points of a small neighborhood on a manifold as coordinates. The map must be one-to-one and, in fact, must be a diffeomorphism on a smooth manifold. In general, a single coordinate chart does not suffice to cover the whole manifold. For example, a manifold of non-trivial topology cannot be described by a single coordinate system. Therefore, the concept of an *atlas* is introduced. An atlas is a set of consistent coordinate charts on a manifold, where consistent means that the transition functions of the charts are smooth. To sum up, an atlas corresponds to a collection of maps, each of which shows a piece of the manifold.

Length measures on the manifold are expressed by the *first fundamental form* or the *metric tensor*. The metric tensor,  $g_{\mu\nu}$ , at a point in the manifold gives rise to the infinitesimal distance  $ds$ , with the line element

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu,$$

where  $dx^\mu$  is an infinitesimal distance in the  $\mu$  direction of the coordinate system. Here and in the following, the Einsteinian sum convention is applied, i.e., summation over double index variables is implied.

Spacetime is a *pseudo-Riemannian* manifold; its metric is not positive definite. In fact, for any single point in spacetime, there exists a coordinate transformation which brings the metric tensor at that point to the form of the flat Minkowski metric: spacetime is a *Lorentzian* manifold. Physically speaking, this means exactly the equivalence principle, i.e., that one can always find a local inertial frame of reference.

### 9.1.2 Geodesics

Paths of objects under the influence of gravitation are identical to geodesics. Geodesics are the “straightest” lines in curved spacetime and have extreme lengths. Geodesics are solutions to a set of ordinary second-order differential equations, the geodesic equations,

$$\frac{d^2 x^\mu}{d\lambda^2} + \Gamma^\mu_{\nu\rho} \frac{dx^\nu}{d\lambda} \frac{dx^\rho}{d\lambda} = 0, \quad (9.1)$$

where  $\lambda$  is an affine parameter for the geodesic line and the coordinates  $x^\mu$  describe the location of the points on the geodesic line. The Christoffel symbols  $\Gamma^\mu_{\nu\rho}$  are determined by the met-

ric according to

$$\Gamma^\mu_{\nu\rho} = \frac{1}{2}g^{\mu\alpha} \left( \frac{dg_{\alpha\nu}}{dx^\rho} + \frac{dg_{\alpha\rho}}{dx^\nu} - \frac{dg_{\nu\rho}}{dx^\alpha} \right),$$

with  $g^{\mu\alpha}$  being the inverse of  $g_{\mu\alpha}$ .

Geodesics can be partitioned into three classes. Depending on their length, they are denoted timelike, spacelike, or lightlike geodesics. Massive objects move along timelike geodesics, whereas photons move along lightlike geodesics.

### 9.1.3 Light Propagation

Ray tracing is focused on the propagation of light through spacetime. It is based on geometric optics, wave effects are neglected. In this framework, the propagation of electromagnetic waves is represented by light rays. Light rays are identical to lightlike (or null) geodesics and obey the null condition,

$$g_{\mu\nu} \frac{dx^\mu}{d\lambda} \frac{dx^\nu}{d\lambda} = 0. \quad (9.2)$$

The fact that light rays are null geodesics can be explicitly derived from Maxwell's equations within general relativity. The following presentation follows Schneider et al.[141, Chapter 3]

In general, Maxwell's equations cannot be solved explicitly in curved spacetimes. The ray tracing application, however, is focused on a specific kind of electromagnetic waves which is nearly plane and monochromatic on a scale that is large compared to a typical wavelength, but small compared to the typical radius of curvature of spacetime. The light investigated in ray tracing is a superposition of the electromagnetic "locally plane" waves. These "locally plane" waves can be represented by an approximate solution of Maxwell's equations in a short-wave approximation:

$$F_{\mu\nu} \approx \Re \left\{ e^{i\varepsilon S} \left( A_{\mu\nu} + \frac{\varepsilon}{i} B_{\mu\nu} \right) + O(\varepsilon^2) \right\}, \quad (9.3)$$

where  $\varepsilon^{-1}S$  is the phase, i.e., a real scalar field, and  $A_{\mu\nu}$  and  $B_{\mu\nu}$  are skew-symmetric, complex tensor fields. The book-keeping parameter  $\varepsilon$  serves to identify orders of magnitude and can be absorbed into  $S$  and  $B_{\mu\nu}$  at the end of the calculations.  $F_{\mu\nu}$  is the electromagnetic field tensor.

For an observer moving along the world line  $x^\mu(\tau)$  with proper time  $\tau$  and four-velocity  $u^\mu = dx^\mu/d\tau$ , the circular frequency  $\omega$  of the wave is defined as

$$\omega = -\frac{dS}{d\tau} = -S_{,\mu}u^\mu = k_\mu u^\mu,$$

with  $\varepsilon = 1$  being supposed. The four-wave vector is

$$k_\mu = -S_{,\mu}.$$

Maxwell's equations are solved by employing Eq. (9.3) in the asymptotic limit  $\varepsilon \rightarrow 0$ . By inspecting terms of order  $\varepsilon$  and  $\varepsilon^{-1}$ , we obtain that

$$k_\mu k^\mu = 0, \quad (9.4)$$

i.e., that the wave vector is a null vector, cf. Eq. (9.2), and that the phase obeys the eikonal equation,

$$g^{\mu\nu} S_{,\mu} S_{,\nu} = 0.$$

Based on the wave vector, the light ray  $x^\mu(\lambda)$  is defined by

$$\frac{dx^\mu}{d\lambda} = k^\mu,$$

where  $\lambda$  is an affine parameter. By differentiating the tangent vector  $k^\mu$  of a light ray covariantly along the light ray, we obtain from Eq. (9.4)

$$k^\mu k^\nu_{;\mu} = g^{\nu\eta} k^\mu k_{\eta;\mu} = g^{\nu\eta} k^\mu k_{\mu;\eta} = \frac{1}{2} g^{\nu\eta} (k^\mu k_\mu)_{;\eta} = 0.$$

$k^\mu$  is a gradient of  $S$ , therefore  $k_{\eta;\mu} = k_{\mu;\eta}$ . The final result,

$$k^\nu_{;\mu} k^\mu = 0, \quad (9.5)$$

is the geodesic equation for light rays. When expressed in terms of Christoffel symbols, the above equation is identical to the alternative form of the geodesic equation (9.1).

The null geodesics are of great importance because they determine the causal structure of spacetime, i.e., they separate regions which cannot have any causal influence on a given point in spacetime.

### 9.1.4 Frequency Shift and Transformation of Specific Intensity

An observer moving with four-velocity  $u^\mu$  registers a circular frequency of an electromagnetic wave,

$$\omega = k_\mu u^\mu. \quad (9.6)$$

The wave vector  $k_\mu$  undergoes a parallel transport along the light ray according to the geodesic equation (9.5). For example, the geodesic equation has to be solved for  $k^\mu$  to find the frequency shift of a photon traveling through the spacetime of a gravitational field.

Usually, three types of frequency shift are distinguished: Doppler shift, cosmological redshift, and gravitational redshift. This distinction is somewhat artificial, since the change of frequency can be treated in the unified framework of general relativity, as described above. Nevertheless, the differentiation can help to understand the physical origins of redshift. Doppler shift is attributed to a motion of the observer or the light source, cosmological redshift to the expansion of the universe. Gravitational redshift is caused by the influence of a gravitational source. In the case of a time-independent metric with  $g_{0i} = 0$ , the quantity

$$\omega_{\text{local}} |g_{00}|^{1/2} = \text{const.}$$

is conserved. In this way, the change of frequency can conveniently be calculated.

The transformation of specific intensity can be derived on the grounds of kinetic theory and Liouville's theorem in curved spacetime. A presentation of kinetic theory can be found, for example, in work by Ehlers[43] and Misner et al.[107, Chapter 22.6]. Specific intensity is identical to frequency-dependent radiance used in Part One. This change of notation takes into account the different usage in the literature in the fields of general relativity and computer graphics.

Let us consider a swarm of particles (massive or massless) which move through spacetime on geodesic world lines, without colliding. A local Lorentz observer measures a volume in

phase space,  $V = V_x V_p$ , covered by the particles. The particles are located in the three-dimensional spatial volume  $V_x$  and have momenta in the range  $V_p$ . Liouville's theorem in curved spacetime states that the phase-space volume  $V$  occupied by a given set of particles is independent of the location along the world line of the particles. Moreover, the volume  $V$  is independent of the local Lorentz frame in which it is measured.

The number density in phase space is defined as

$$N = N/V,$$

where  $N$  denotes the number of particles. Liouville's theorem implies that  $N$  is conserved along a particle's path through spacetime,

$$\frac{dN(x(\lambda), p(\lambda))}{d\lambda} = 0,$$

where  $x(\lambda)$  is the four-position and  $p(\lambda)$  the four-momentum of the particle. For the application to light propagation, the number density in the local Lorentz frame is expressed in terms of the specific intensity  $I_\nu$  and the frequency  $\nu$  of the light,

$$N = h^{-4} \frac{I_\nu}{\nu^3},$$

where  $h$  is Planck's constant.

This means that the ratio  $I_\nu/\nu^3$  is invariant from observer to observer and from event to event along the photon's world line:

$$I_\nu/\nu^3 = \text{const.} \quad (9.7)$$

The frequency shift and the transformation of specific intensity can therefore be treated simultaneously in the ray tracing implementation in order to model shift in color and brightness. Observe that this result is identical to the searchlight effect in flat spacetime. It is the generalization of the transformation of specific intensity (frequency-dependent radiance) to curved spacetime.

## 9.2 Ray Tracing in a Single Chart Spacetime

In this section, ray tracing in a restricted class of spacetimes is presented. The considered spacetimes have to be described by a single chart, i.e., a single coordinate system. In this way, a standard three-dimensional Euclidean ray tracing system can be easily extended to incorporate general relativistic rendering.

An object-oriented concept is supposed for the standard three-dimensional ray tracing program in which all relevant parts of the visualization system are represented by objects, cf. Sect. 5.2. The extension of the original functionality can be included by subclassing.

Figure 9.1 shows the basic structure of such an idealized ray tracer. The image plane is sampled by the `Sample Manager`, which uses the `Projector` to generate a `Ray` corresponding to the pixel under consideration. The `Ray` communicates with the `Scene` in order to find intersections with scene objects, calculate secondary rays and shadow rays, and determine illumination. Finally, the resulting color is stored in the image by the `Sample Manager`.

In this object-oriented framework, relativistic rendering requires the following four extensions.

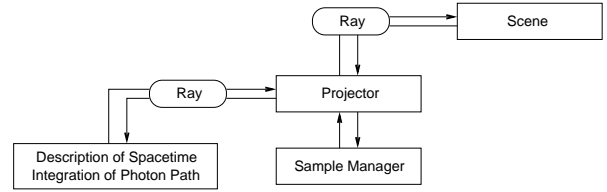


Figure 9.1: Structure of the ray tracing system.

First, the standard `Ray` class representing a straight light ray in three dimensions is replaced by a class representing a bent light ray in four dimensions. This bent ray is approximated by a polygonal line whose points possess one temporal and three spatial coordinates.

Secondly, the description of the `Scene` objects has to be extended from three spatial coordinates to four spacetime coordinates. By including the temporal coordinate, moving and animated objects can be represented. The description of the light ray and the object utilizes the restriction to a single chart because both are described by a single four-dimensional coordinate system.

Thirdly, the standard ray `Projector` has to be modified. The new ray projector generates a light ray whose initial position in spacetime and initial spatial direction are determined by the position, orientation, and field of view of the observer's camera and by the coordinates of the corresponding pixel on the image plane. The time component of the initial direction is fixed by the null condition (9.2). Therefore, the geodesic equations (9.1) yield an initial value problem for ordinary differential equations. There exist well-known numerical methods for solving this problem, cf., for example, Numerical Recipes[130].

The projector communicates with the solver for the geodesic equations (9.1) and, in this way, provides the interface to the physics of spacetime. The system of ordinary differential equations can be solved by numerical integration. The standard technique in the implemented system is the Runge-Kutta method of fourth order with adaptive step size control, but other methods which are better suited for the numerics of a specific problem can be included as well. The integrator solves the geodesic equation both for the position in spacetime,  $x^\mu$ , and the tangent vector,  $k^\mu$ . In this way, the path of the photons is known and can be used to perform calculations of intersection with scene objects.

Fourthly, the transported wave vector can be used to model frequency and intensity changes in the registered light. Equation (9.6) provides the frequency shift, Eq. (9.7) yields the transformed specific intensity. The `Projector` takes into account the modified power spectrum of the incoming light in order to return the transformed tristimulus values to the `Sample Manager`.

The advantage of this modular and object-oriented concept is a freedom of choice of the simulated system, combined with a complete, sophisticated rendering environment and only minimal extra implementation costs. Any physical configuration can be examined by replacing the module which supplies the information about the metric and the Christoffel symbols.

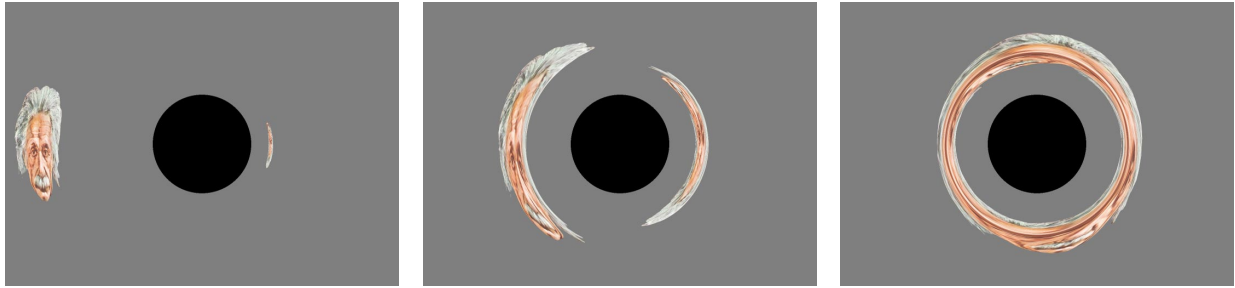


Figure 9.2: Ray tracing in Schwarzschild metric.

### 9.2.1 Implementation and Results

The implementation of general relativistic ray tracing is based on *RayViS*[64], which is an object-oriented and easily extensible ray tracing program written in C++. In *RayViS*, all relevant parts of the visualization system are derived from abstract base classes which allow the extension of the original functionality by subclassing. The current implementation of general relativistic ray tracing does not support changes in color or brightness due to shifts in frequency or specific intensity of photons. Secondary rays and shadow rays are neglected as well. A detailed documentation of the relativistic extension can be found in Appendix C.3.

In standard three-dimensional ray tracing, rendering time is significantly determined by the computation of the intersections between rays and objects. This is not true for general relativistic ray tracing because here the generation of bent light rays by solving the geodesic equations plays a dominant role. In typical applications, general relativistic ray tracing is two to four magnitudes slower than non-relativistic ray tracing.

Therefore, parallelization is an urgent need for general relativistic ray tracing. Fortunately, the computation of the null geodesics and the ray-object intersections for one pixel is independent of those for the other pixels. Hence, parallelization is performed on a per-pixel basis and utilizes a domain decomposition on the image plane. The granularity can be as fine as a single line on the image plane in order to achieve good load balancing. The implementation makes use of MPI[104] and thus is platform-independent. It scales well, even up to a several hundred nodes on a massive-parallel architecture such as the CRAY T3E.

Figure 9.2 shows results of general relativistic ray tracing for the Schwarzschild metric. A portrait of Einstein moves behind the gravitational source to illustrate the bending of light. The Schwarzschild metric is only a test case for general relativistic ray tracing. Physically more interesting and more complex spacetimes are discussed in detail in the following two sections.

## 9.3 Visualization of the Rigidly Rotating Disk of Dust

In this section, the visualization of the general relativistic rigidly rotating disk of dust is investigated. This project is joint work with the group for gravitational theory at the University of Jena, in particular, with Marcus Ansorg. Results can also be found in our paper[173].

The general relativistic gravitational field created by the rigidly rotating disk of dust was first studied numerically in 1971 by Bardeen and Wagoner[11]. The global analytical solution of Einstein's field equations for this object was found in 1995 by Neugebauer and Meinel[115]. Their explicit expressions for the metric coefficients can be used to create a direct numerical implementation of the geodesic equation.

### 9.3.1 The Metric

Einstein's field equations for the rigidly rotating disk of dust can be reduced to a single non-linear complex partial differential equation—the so-called Ernst equation—for which a boundary value problem has to be solved[47, 87]. Neugebauer and Meinel succeeded in solving this problem by means of the inverse scattering method, a technique known from soliton theory.

In Weyl-Papapetrou coordinates  $(\rho, \zeta, \phi, t)$ , the metric  $g_{\mu\nu}$  assumes the form

$$\begin{pmatrix} e^{2(k-U)} & 0 & 0 & 0 \\ 0 & e^{2(k-U)} & 0 & 0 \\ 0 & 0 & -a^2 e^{2U} + \rho^2 e^{-2U} & -a e^{2U} \\ 0 & 0 & -a e^{2U} & -e^{2U} \end{pmatrix},$$

with

$$g_{\mu\nu, \phi} = g_{\mu\nu, t} = 0,$$

and with  $g_{\mu\nu}$  possessing reflection symmetry with respect to the plane  $\zeta = 0$ , i.e.,

$$g_{\mu\nu}(\rho, \zeta) = g_{\mu\nu}(\rho, -\zeta).$$

The resulting field equations turn out to be equivalent to the Ernst equation,

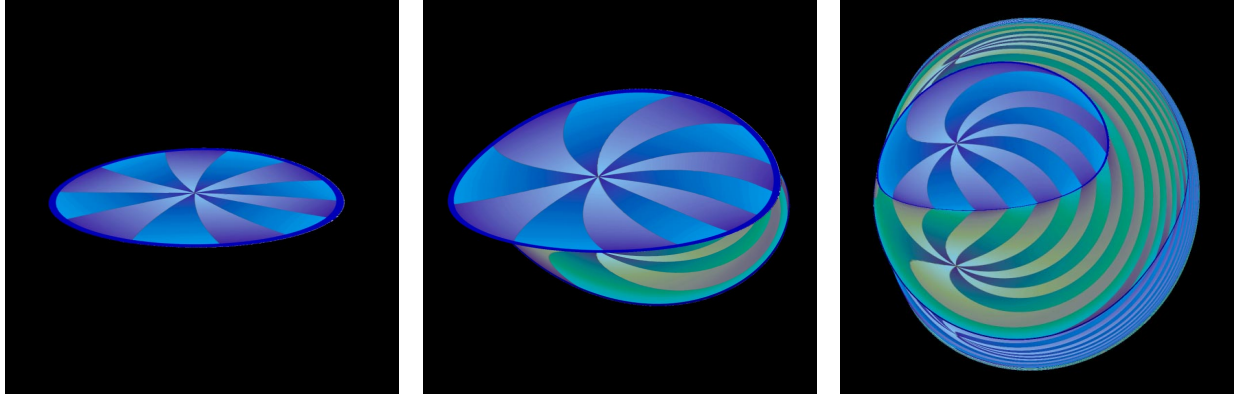
$$\Re f \left( f_{,\rho\rho} + f_{,\zeta\zeta} + \frac{f_{,\rho}}{\rho} \right) = f_{,\rho}^2 + f_{,\zeta}^2,$$

for the Ernst potential  $f$  which is defined by

$$f = e^{2U} + ib \quad \text{with} \quad b_{,\zeta} = \frac{e^{4U}}{\rho} a_{,\rho}, \quad b_{,\rho} = -\frac{e^{4U}}{\rho} a_{,\zeta}.$$

The remaining metric function  $k$  can be calculated from the functions  $U$  and  $a$  by quadratures.

Neugebauer and Meinel found the Ernst potential for the rigidly rotating disk of dust in terms of ultraelliptic



**Figure 9.3:** Visualization of the rigidly rotating disk of dust. The relativistic parameter  $\mu$  is 0.1, 0.7, 3, from left to right.

functions[115]. In their expressions, the Ernst potential depends on a parameter  $\mu$  which is related to the angular velocity  $\Omega$  and the radius  $\rho_0$  of the disk by

$$\mu = 2\Omega^2 \rho_0^2 e^{-2V_0} \quad \text{with} \quad V_0(\mu) = U(\rho = 0, \zeta = 0; \mu).$$

The parameter  $\mu$  runs on the interval  $[0, \mu_0]$  with  $\mu_0 = 4.62966\dots$ . For  $\mu \ll 1$ , the Newtonian limit of the Maclaurin disk is obtained;  $\mu \rightarrow \mu_0$  and  $\rho_0 \rightarrow 0$  yields the extreme Kerr solution. For  $\mu > \mu_e \approx 1.69$ , the disk creates an ergoregion in which the metric function ( $-e^{2U}$ ) is positive.

### 9.3.2 Visualization Results

The properties of the corresponding geometry of spacetime can be studied, for example, by investigating the trajectories of particles, i.e., the timelike geodesics[4]. General relativistic ray tracing serves as another tool for examining solutions to Einstein's field equations. The following fields of application are explored in detail for the example of the rigidly rotating disk of dust, but might be useful for other metrics as well.

First, visualization allows a compact representation of a vast number of null geodesics which are used as another means of probing the gravitational field. Secondly, there is great interest in "seeing" the results of theoretical work in order to gain some intuitive approach to these results. Thirdly, the communication of the theoretical research to the public should be facilitated.

These visualization techniques can be applied to the example of the rigidly rotating disk of dust. Respective results are presented in the following.

#### Outside View

The first and straightforward approach to visualizing a given gravitational field is to adopt an outside position. Figure 9.3 illustrates such an outside view. The three images show the disk with varying parameter  $\mu$ .

The left image presents an almost Newtonian, classical situation with  $\mu = 0.1$ . The top side of the disk is colored blue. An artificial "pie slice" texture is applied in order to visualize rotational distortions. The middle image shows a slightly relativistic case with  $\mu = 0.7$ . Due to gravitational light bending both the top and the bottom faces are visible simultaneously.

The bottom side is colored green and brown. The right image shows a more relativistic situation with  $\mu = 3$ . Here, multiple images of the top and the bottom emerge. Moreover, rotational distortions caused by frame dragging (a general relativistic effect) and by the finite speed of light and the rotation of the disk are prominent.

The outside view gives a first, intuitive approach to the gravitating object. It is most useful for presenting the theoretical research to the public. For example, pictures of the rigidly rotating disk of dust were published in a scientific calendar[40] edited by the DFG (Deutsche Forschungsgemeinschaft).

#### Parameter Study

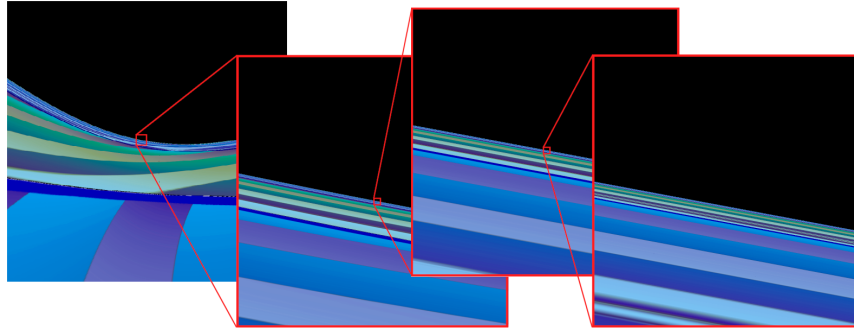
After these first visualization steps a systematic investigation of the properties of the light rays in the metric of the rigidly rotating disk of dust is required in order to obtain reliable scientific results. Therefore, a sampling of the parameter space for the null geodesics has to be considered.

The null geodesics are determined by two types of parameters. Parameters of the first kind describe the gravitating body. The properties of the disk of dust are completely determined by the parameter  $\mu$ . The position of the observer and the direction of the incoming light constitute parameters of the second kind.

The sampling of the direction of the light rays is implemented in the form of a  $4\pi$  sterad camera, i.e., an observer looking in all directions simultaneously. Here, the projection onto a virtual sphere surrounding the observer is used instead of the standard projection onto an image plane. Therefore, the parameter space is completely sampled by generating  $4\pi$  sterad images for various values of  $\mu$  and positions of the observer.

The produced panorama images are viewed with a simple, external, OpenGL-based rendering program which maps these images onto a sphere. The viewpoint is located at the center of the sphere. The viewing direction can be interactively changed by the user.

The parameter study confirms the qualitative results from the previous subsection, i.e., multiple images of the top and the bottom side and rotational distortions. In addition, new interesting results are found for observers that are closer to



**Figure 9.4:** Fractal-like structures and self-similarity in the gravitational field of the rigidly rotating disk of dust with  $\mu = 3$ . The observer is located on the symmetry axis and is zooming in on the image.

the disk of dust. These results are described in the following subsection.

### Fractal-Like Structure

An interesting result of the parameter study is the existence of fractal-like structures created by the gravitational field of the rigidly rotating disk of dust. Figure 9.4 shows a typical example. Here, the position of the observer and the parameter  $\mu = 3$  are fixed. The observer is located on the axis of symmetry and is looking towards the edge of the disk. The leftmost picture shows a snapshot with a wide angle field of view. Parts of the top side of the disk are visible in the lower part of the picture. An image of the bottom side is found directly above this first image of the top side. Further above, alternating images of the top and the bottom faces follow. The pictures to the right document increased zooming in on the original picture, whereas the rightmost image shows a part of the leftmost image which has a size approximately ten orders of magnitude smaller than the original image. This series reveals self-similarity and a fractal-structure.

Normally, Poincaré sections are used to investigate the stability of a physical system, for example, the properties of time-like geodesics[4]. For a description of Poincaré sections, I refer to the numerous textbooks on non-linear dynamics and chaos, for example, Alligood et al.[3] or Ott[122]. Poincaré sections require the investigation of bound orbits. However, bound orbits of photons are hard to find or sometimes do not even exist. Therefore, Poincaré sections are usually not applicable to null geodesics. Conversely, ray tracing can be employed in this setting in order to give indications about the behavior of the system.

General relativity is an excellent example for a non-linear dynamic theory. However, the standard measures of chaos are based on metrical properties of phase space and thus depend on the chosen observer. Therefore, gauge-dependent measures of chaotic behavior such as Lyapunov exponents should be replaced by invariant measures such as fractal dimensions. Fractal methods are applied within general relativity, for example, by Dettmann et al.[39] or Cornish and Levin[30]. The chaotic behavior of geodesics can be investigated by fractal methods. An example is the examination of the geodesics in the Einstein-dilaton-Maxwell two-center spacetimes[29], the relativistic analog of the two fixed-centers problem. The ray-tracing approach could be extended to incorporate techniques for measurements of fractal dimensions.

## 9.4 Visualization of the Warp Metric

The visualization of the warp drive metric is another example of general relativistic ray tracing. The warp metric is a dynamic metric which does not conserve the energy of test particles and photons traveling through spacetime.

Alcubierre's solution[2] of Einstein's field equations allows to travel faster than the speed of light as measured in an outside, flat region of spacetime. Basically, the warp drive constructs a warp bubble which separates two flat parts of spacetime. The warp bubble is able to move faster than the speed of light as measured in an outside, flat region of spacetime. A spaceship which is at rest inside the warp bubble would then travel faster than the speed of light.

The view from inside the warp spaceship was independently investigated by Clark et al.[26]. Their scenario is restricted to an observer inside the warp bubble. They focus on a physical interpretation of the warp metric, whereas the generic ray tracing approach of this thesis rather addresses visualization issues and enables us to adopt any observer position.

### 9.4.1 The Metric

The warp drive spacetime is described by the line element

$$ds^2 = dt^2 - (dx - v_s f(r) dt)^2 - dy^2 - dz^2,$$

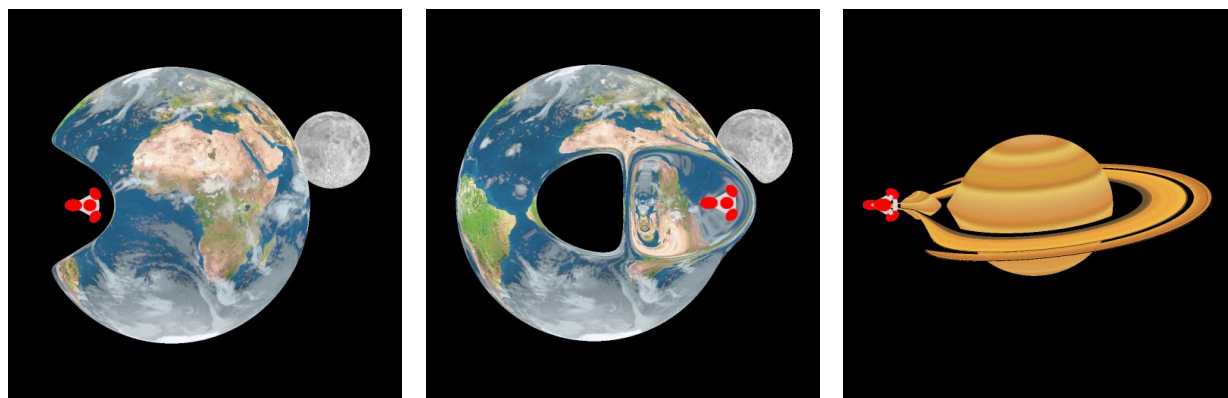
where  $v_s = dx_s/dt$  is the velocity of the warp bubble moving along a trajectory  $x_s(t)$ . A measure for the radial distance from the center of the spacetime distortion is

$$r = \sqrt{(x - x_s)^2 + y^2 + z^2}.$$

The function  $f(r)$  has to be normalized to unit value at the center of the warp bubble. It rapidly falls off at some finite radius  $R$ , i.e., at the boundary of the warp bubble, and asymptotically approaches zero at large distance. The explicit form for  $f(r)$  in Alcubierre's paper is

$$f(r) = \frac{\tanh[\sigma(r+R)] - \tanh[\sigma(r-R)]}{2\tanh(\sigma R)}.$$

This "top hat" function has an almost constant value  $f \approx 1$  around  $r = 0$ . It falls rapidly from  $f \approx 1$  to  $f \approx 0$  around  $r = R$ , where the width of the drop-off region is determined by the constant  $\sigma$ . In this way, the spacetime within the warp



**Figure 9.5:** Visualization of the warp metric. The warp bubble and the spaceship are moving at a speed of  $v = 1.5c$  in the left and middle image, and  $v = 2.5c$  in the right image.

bubble and outside the warp bubble is roughly flat; extreme curvature is limited to the boundary of the bubble.

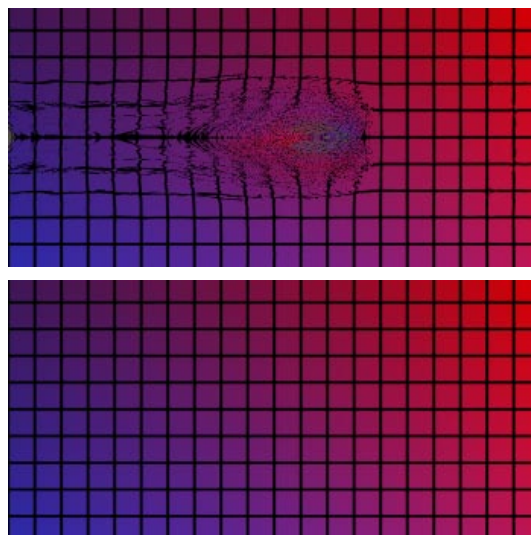
Although the warp metric is in total agreement with general relativity, there exist some issues related to energy conditions and causality. Known forms of classical matter obey the *weak energy condition* (WEC), i.e.,  $T_{\mu\nu}u^\mu u^\nu \geq 0$ , where  $T_{\mu\nu}$  is the stress-energy tensor of matter and  $u^\mu$  an arbitrary time-like vector (four-velocity). By continuity, the WEC holds for null vectors as well. The WEC implies that the energy density measured by any observer has to be non-negative. The warp metric, however, requires “exotic matter”, which violates the WEC. It is known that renormalized stress-energy tensors of quantum fields can violate the WEC. For example, the Casimir effect produces negative energy densities. Therefore, it has not yet been ruled out that “exotic matter” might be generated, but so far it is not clear whether such types of material can exist at all.

Another problem is causality. The warp drive allows to travel faster than light with respect to an outside frame of reference. Hence, a time machine could be constructed, with fundamental problems of determinism and causality. Ford and Roman[52] and Pfenning and Ford[127] discuss some issues related to energy conditions and causality. In the following, these fundamental issues are not considered, but the warp drive is visualized as it is. In fact, it might be worthwhile to study extreme spacetimes such as the warp drive in order to achieve a deeper understanding of the physics of gravitation.

### 9.4.2 Implementation and Results

The warp drive is implemented as a module in the general relativistic ray tracing concept within *RayVis*, cf. Appendix C.3. The steep drop-off at the boundary of the warp bubble causes problems for the numerical integration of the null geodesics. The warp module uses Runge-Kutta with adaptive step size control to achieve an acceptable accuracy.

The overall accuracy of the numerical integration is tested by checking the null condition for the wave vector at the end of the light ray. Additionally, a specific check of the global accuracy can be enabled. Here, the light rays are reversed in time and direction when they leave the computational region. If the numerical integration is satisfactory, the reversed light rays will exactly return to the observer. A textured object can



**Figure 9.6:** Numerical accuracy of ray tracing for the warp metric with  $v = 1.5c$ . The upper image indicates numerical problems, occurring for a relative error  $\epsilon = 10^{-5}$ . The computation of the lower image is based on  $\epsilon = 10^{-8}$ .

be positioned right behind the observer to visualize the quality of the numerical integration. Normal intersection calculations, now based on the original ray and the reversed ray, provide an image of the textured object. An unperturbed image indicates high accuracy, whereas cracks in the image signalize numerical inaccuracies.

Figure 9.5 compares numerical integration for relative errors of  $\epsilon = 10^{-5}$  and  $\epsilon = 10^{-8}$ , respectively. The error criterion  $\epsilon$  determines the maximal relative error in a single step of the numerical integration. The spacetime is described by the warp metric at  $v = 1.5c$ . The upper image reveals numerical problems, especially in the region of the warp bubble. In contrast, the lower image does not display any numerical errors.

Figure 9.5 shows examples of the visualization of the warp metric. Here, the warp spaceship travels in front of the earth

and moon, and Saturn. The light deflection at the warp bubble causes astonishing visual distortions on the background objects. In addition to this outside view, a position inside the warp bubble can be adopted, cf. Sect. 10.3. The accompanying video shows further visualizations with an outside viewing position, cf. Appendix D.1.5.

Images of the warp drive visualization were also shown at “Seven Hills”[13], an exhibition which intends to give an inkling of what the future of mankind may look like in the dawning millennium. A leading-edge topic of physics like the visualization of the warp metric is well suited for such an exhibition and allows to communicate aspects of a complex scientific content to a wide public.

## 9.5 Spacetimes of Non-Trivial Topology

Not only do Einstein’s field equations allow curved spacetimes but also spacetimes of various topologies. In this section, ray tracing of spacetimes of non-trivial topologies is proposed. This technique is specifically applied to the visualization of wormholes, which are a prominent example for spacetimes of complex topology, but it is not restricted to wormholes.

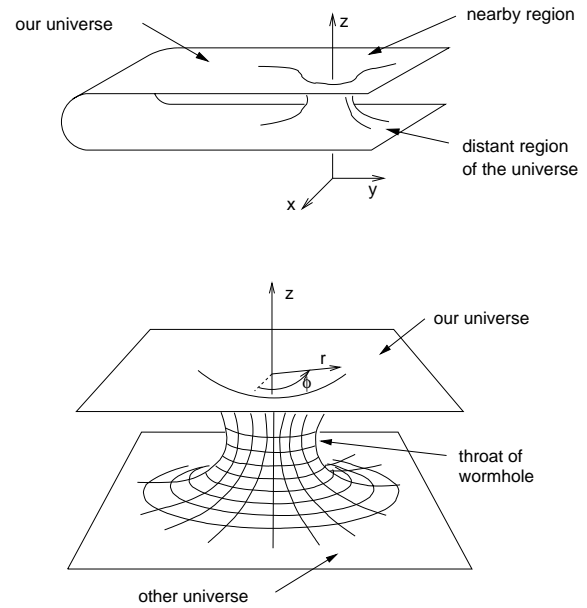
Since spacetimes of non-trivial topology cannot be described by a single coordinate system, a collection of several charts is required. The idea is to extend the general relativistic ray tracing from the previous section to incorporate the differential geometric concept of an atlas with several charts. This enables us to visualize spacetimes of non-trivial topologies. Moreover, it can be useful even for ray tracing of spacetimes with flat topology. Here, the quality of numerical integration of null geodesics can be improved by choosing several coordinate systems which are adapted to the numerics in the respective regions of the spacetime.

Differential geometry and Einstein’s field equations can even handle non-orientable manifolds. Non-orientability implies that a right-handed body which travels through spacetime might return left-handed. It is generally believed that the CP-invariance violations observed in particle experiments forbid a non-orientable universe, cf. the discussion by Zel’dovich and Novikov[185]. Nevertheless, the investigation of non-orientable manifolds might be interesting in order to gain further understanding of the geometry of spacetime. Non-orientable spacetimes have not been implemented yet, but could be considered in future work.

### 9.5.1 Theoretical Description of Wormholes

In general, a wormhole is regarded as a connection between two asymptotically flat parts of spacetime. They might be used to travel to faraway galaxies in acceptable time spans.

Flamm[50] recognized that the spherically symmetric vacuum solutions by Schwarzschild represent a connection between different regions of spacetime. Possible roles of Schwarzschild wormholes and other wormholes were discussed by Weyl[179], Einstein and Rosen[45], and Wheeler [180]. The Schwarzschild wormhole has some disadvantages which rule it out as being a mechanism for space travel. For a reasonably sized wormhole, the tidal forces are too big to be sustained by a vehicle or person. Only one-directional traversal is possible because a horizon and an anti-horizon are con-



**Figure 9.7:** Embedding diagrams for a wormhole. The upper diagram shows a wormhole connecting two regions of a single universe, the lower diagram shows a wormhole connecting two different universes.

nected. Furthermore, a Schwarzschild wormhole is unstable against small perturbations.

Morris and Thorne[111] and Morris et al.[110] propose another metric which avoids the above problems. Their wormhole is traversable, i.e., it possesses no horizons. It exerts only little acceleration and tidal forces onto the traveler. It is stable against small perturbations. Therefore, it is specifically suited for space travel. Figure 9.7 shows embedding diagrams for a wormhole which connects two distant regions of a single universe or two different universes, respectively.

The wormhole metric has the general Schwarzschild form,

$$ds^2 = e^{2\Phi(r)} dt^2 - (1 - b(r)/r)^{-1} dr^2 - r^2 (d\theta^2 + \sin^2 \theta d\phi^2).$$

The spatial shape of the wormhole is determined by  $b(r)$ . The redshift function  $\Phi(r)$  represents a Newtonian gravitational potential which takes effect in addition to the spatial curvature. The radial coordinate  $r$  is not unique because it has a minimal radius  $r_t$  at the throat and increases with the distance to the throat on both sides of the wormhole. By employing the radial proper length,

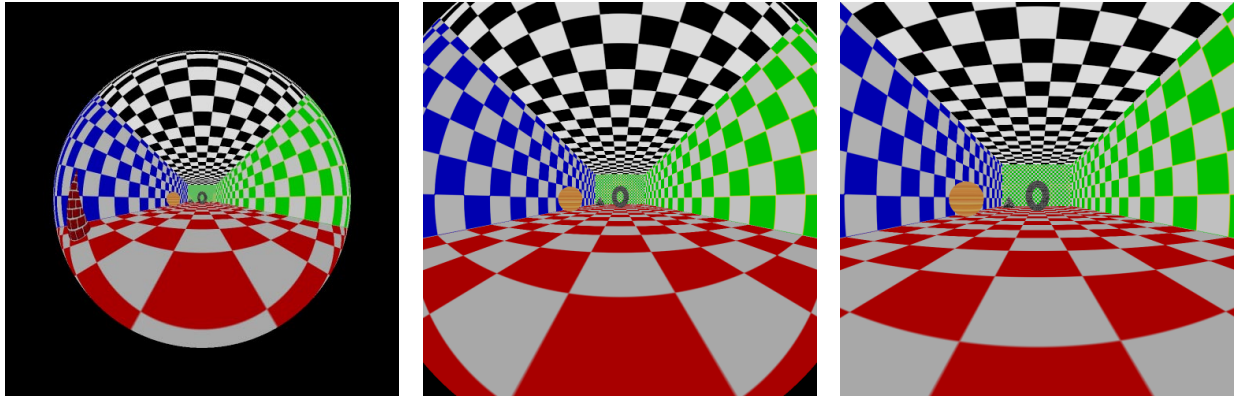
$$l = \int_{r_t}^r \frac{dr'}{\pm \sqrt{1 - b(r')/r'}},$$

one obtains unique coordinates and the following form for the metric:

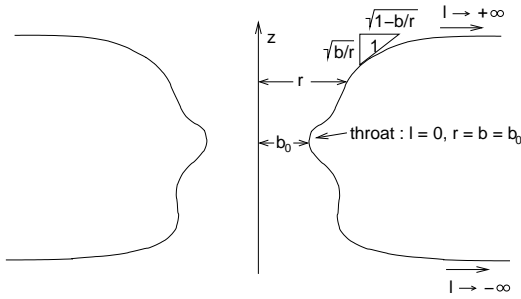
$$ds^2 = e^{2\Phi(r)} dt^2 - dl^2 - r^2 (d\theta^2 + \sin^2 \theta d\phi^2),$$

where  $1 - b(r)/r \geq 0$ .





**Figure 9.9:** Flight through a wormhole connecting two different universes. From left to right, the position of the observer is described by  $r = 5/2r_t$ ,  $5/3r_t$ , and  $4/3r_t$ .



**Figure 9.8:** Embedding of the throat of a wormhole.

The function  $\Phi$  has to be finite everywhere to guarantee the absence of horizons. The spacetime should be asymptotically flat, which implies

$$\frac{b}{r} \rightarrow 0 \quad \text{as } l \rightarrow \pm\infty.$$

Figure 9.8 shows the embedding diagram of the throat of a wormhole for an example choice of  $b(r)$ . Morris and Thorne give three explicit choices for the functions  $\Phi$  and  $b$ . Their first class of wormholes has zero tidal force, i.e.,  $\Phi = 0$  everywhere. The second class has a finite radial cutoff for exotic matter. The third class limits exotic matter to the throat vicinity.

The traversable Morris-Thorne wormhole has some major deficiencies. First, it requires an energy-stress tensor representing exotic matter, i.e., it violates the weak energy condition. Secondly, the wormhole enables us to realize time travel. This initiates deep problems related to causality. In this way, the wormhole has similar fundamental problems as the warp drive. In the following, these issues are not considered, but the wormhole is visualized as it is.

### 9.5.2 Ray Tracing Concept

An atlas can be taken into account by the following conceptual extensions of the single chart ray tracing system.

The root node of the scene graph has to be a special type of node which represents the whole Spacetime. Space-

time contains a list of Charts and Chart Boundaries. The Chart itself is a node in the scene graph. It contains further subnodes with object descriptions with respect to the associated coordinate system. These subnodes are identical to the scene objects in the single chart ray tracer. The Chart Boundary represents the boundary of a coordinate system and provides the functionality for the transition from one coordinate system to the other, i.e., the transformation of points in spacetime and tangent vectors.

The integration of the null geodesics takes into account the information of the Spacetime. Whenever a Chart Boundary is crossed, the current position and tangent vector of the light ray is transformed from the old coordinate system into the new coordinate system and integration is continued in the new coordinates. Hence, the Ray is represented by polygonal lines associated with Charts.

After the generation of the light ray, the intersection calculations are performed. The Projector communicates with the Scene via Ray to find ray-object intersections. The root node of the Scene is a Spacetime node. Spacetime delegates the intersection computation to the respective subnode—the Chart. The Chart forwards the intersection computation to its scene subnodes. The scene subnode performs the intersection calculation in local coordinates, i.e., this calculation is the same process as in the single chart ray tracer.

In intermediate calculations, extensive usage of vectors is made. The concept of a vector allows for the requirements of differential geometry by including the coordinates of the vector itself, the coordinates of the footpoint, and a reference to the associated Chart.

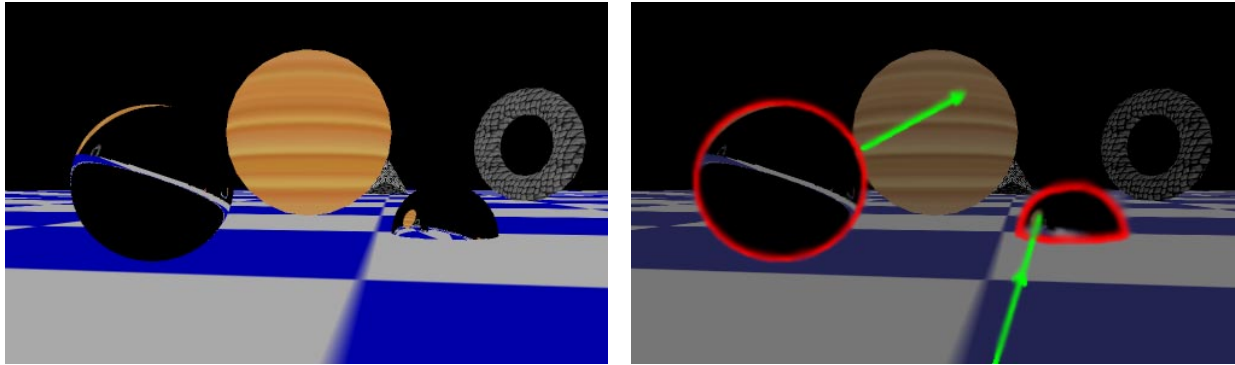
### 9.5.3 Implementation and Results

The current implementation of non-trivial spacetimes within RayViS is limited to a restricted class of static wormholes with

$$l(r) = \pm(r - r_t),$$

where  $r_t$  is the radial coordinate of the throat. Here, two flat spacetimes are connected by a throat of infinitesimal length.

This implementation was developed by Mellinger[102]. A detailed presentation of the implementation and further results can be found in his thesis. The full implementation of an atlas for four-dimensional spacetime is planned in future work.



**Figure 9.10:** Visualization of both ends of the wormhole connecting a single universe. The left image shows the situation as seen by the observer. In the right image, both mouths of the wormhole are marked red. The right mouth intersects the checkered ground and is only partly visible. The green line is a qualitative representation of an exemplary light ray that crosses the wormhole.

Figures 9.9 and 9.10 show the visualization of a wormhole spacetime. In Fig. 9.9, the observer is traveling through a wormhole connecting two different universes. The travel starts in a universe which does not contain any visible objects. An increasing field of view is covered by the second universe when the observer is approaching the throat of the wormhole. The position of the observer is described by the radial coordinate,  $r$ , with respect to the radial coordinate of the throat,  $r_t$ .

In Fig. 9.10, a wormhole connecting a single universe is depicted. The left image shows the situation as seen by the observer located at an outside position. Both ends of the wormhole are visible for the observer. In the right image, the mouths of the wormhole are marked red. The right mouth intersects the checkered ground and is only partly visible. The green line is a qualitative representation of an exemplary light ray which enters the right mouth of the wormhole, traverses the wormhole, leaves the left mouth, and finally hits the surface of the spherical scene object.

## Chapter 10

# Gravitational Lensing

Gravitational fields bend light rays and can thus play the role of a lens—a gravitational lens. Gravitational lensing was early predicted by Einstein. In fact, the light deflection measured during the total eclipse of the sun in 1919 was the first experimental evidence for general relativity. Today gravitational lenses are a hot topic in astronomy and are extensively used for observations in various ways. A comprehensive presentation of gravitational lensing can be found in the book by Schneider et al.[141]. The review article by Fort and Mellier[53] is focused on gravitational lensing by clusters of galaxies.

Figure 10.1 shows the first confirmed case of gravitational lensing. In 1979, the QSOs (quasi-stellar objects) 0957+561 A and B were identified by Walsh et al.[163] as a double image of the same source. QSO 0957+561 has a redshift of  $z = 1.41$ , the lens has a redshift of  $z = 0.36$

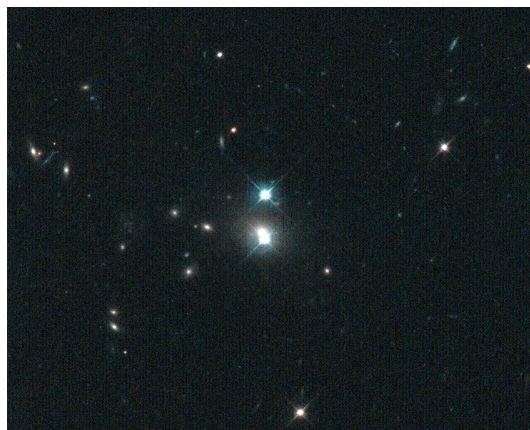
The main difference between optical lenses and gravitational lenses is that the deflection caused by a typical, spherically convex, optical lens increases with the distance of the ray from the optical axis, whereas the deflection caused by a gravitational lens decreases with the impact parameter. A standard optical lens has a focal point onto which rays parallel to the optical axis are focused. In contrast, a gravitational lens has no focal point. However, the qualitative behavior of a gravitational lens can be described by its caustic surfaces. A gravitational lens might bend light rays in a way that the light emitted by the same source can follow different trajectories to reach the observer, i.e., the light source is projected onto multiple points on the observer's image plane. A caustic surface separates regions where the image multiplicity changes.

Lensing theory is based on a couple of assumptions and approximations which are explained in the next section. Based on this footing, astrophysical lenses can be examined to obtain information about the lens, such as its mass distribution, or about the background object, i.e., the observed data is analyzed to reconstruct the properties of the lens or the source.

In this chapter, a different approach is pursued. I target at visualizing a known gravitational field to gain further insight in its characteristics. Therefore, the focus is on image synthesis instead of image analysis.

The outline is as follows. I start with a brief summary of the theoretical background of standard lensing theory. Some examples of visualization of standard lensing are presented. A simple extension to ray tracing is developed to incorporate visualization of standard lensing. Then, two alternative means of visualizing gravitational lenses are proposed. For both techniques, the approximations underlying standard lensing theory are dropped and the geodesic equations are completely solved.

The first technique directly utilizes general relativistic ray



**Figure 10.1:** Double QSO 0957+561 (from Keel[84], University of Alabama).

tracing. Here, objects are placed around the gravitational lens as image sources. The mapping of these background objects reveals distortions due to gravitational light deflection and a possible multiplicity of points in the image plane. In this way, it discloses some information about the structure of the lens and, in particular, its caustics.

The second technique targets at a more systematic analysis of the caustic surfaces. The deficiency of the first method is that the background objects are only two-dimensional and thus cannot visualize the caustic surfaces embedded in three-dimensional space. This problem can be overcome by a three-dimensional volumetric object which determines the image multiplicity for points lying inside.

## 10.1 Theoretical Background

The following summary of facts from standard gravitational lensing theory is based on Schneider et al.[141]. Please refer to their book for more detailed information.

The simplest gravitational lens is a static spherical body of mass  $M$ . This Schwarzschild lens causes a deflection of light rays by the Einstein angle,

$$\hat{\alpha} = \frac{2R_S}{\xi},$$

with the impact parameter  $\xi$ , the Schwarzschild radius  $R_S =$

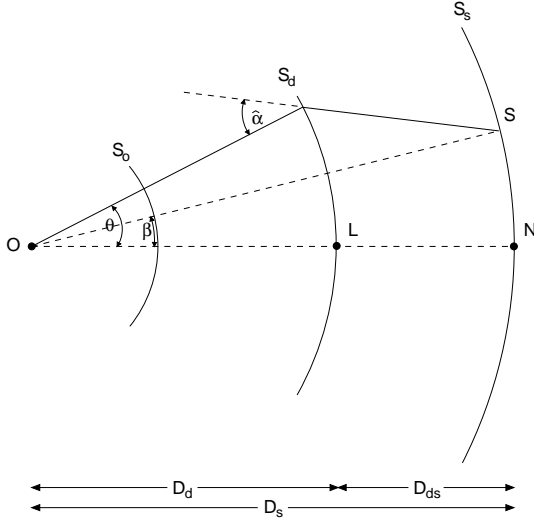


Figure 10.2: General lensing system.

$2GM$ , and the gravitational constant  $G$ . This relationship is based on the linearized Schwarzschild metric and is valid for small angles of deflection, which is equivalent to the condition  $\xi \gg R_s$ .

The general lensing situation is depicted in Fig. 10.2. The lens is located around  $L$ . The line through  $L$  and the observer  $O$  is the optical axis of the system. The source  $S$  would have the angular position  $\beta$  if gravitational light deflection was absent. Actually, the light is deflected by an angle  $\hat{\alpha}$ , so the source is observed at the angle  $\theta$ . The source sphere  $S_s$  has radius  $D_s$  and is centered around the observer. The lens or deflection sphere,  $S_d$ , has radius  $D_d$ . The “sky” of the observer,  $S_o$ , is located in the vicinity of the observer in order to quantify the angular positions of  $\theta$  and  $\beta$ . In a cosmological setting, all geometric terms, such as sphere, line, or radius, refer to the three-dimensional spatial hypersurface of constant curvature of the background Friedmann-Lemaître model and the distances are assumed to be angular-diameter distances.

Standard lensing theory is based on the following assumptions and approximations. The gravitational field has to be weak. Therefore, only small deflection angles have to be considered and the tangent planes to the spheres  $D_s$  and  $D_d$  can be used instead of the spheres themselves. The angular positions  $\vec{\theta}$  and  $\vec{\beta}$  are regarded as two-dimensional vectors in  $S_o$ . Furthermore, the matter distribution has to be nearly stationary, i.e., the velocity of the matter in the lens has to be well below the speed of light. All these conditions are satisfied for most astrophysical situations. However, some phenomena, such as black holes, require more accurate modeling.

Based on the above assumptions, the lens equation describes the relationship between the unlensed position angle  $\vec{\beta}$  and the image position  $\vec{\theta}$ :

$$\vec{\beta} = \vec{\theta} - \frac{D_{ds}}{D_s} \hat{\alpha}(\vec{\xi}), \quad (10.1)$$

where  $\vec{\xi} = D_d \vec{\theta}$  is the impact parameter measured in the lens plane. For a geometrically thin lens, the deflection angle is

determined by

$$\hat{\alpha}(\vec{\xi}) = 4G \int \Sigma(\vec{\xi}') \frac{\vec{\xi} - \vec{\xi}'}{|\vec{\xi} - \vec{\xi}'|^2} d^2\xi', \quad (10.2)$$

where the integration extends over the lens plane,  $d^2\xi'$  is a surface element in the lens plane, and  $\Sigma(\vec{\xi}')$  is the surface mass density.

The lens equation allows to calculate the true position  $\vec{\beta}$  for a given image position  $\vec{\theta}$ , as long as the deflection law  $\hat{\alpha}(\vec{\xi})$  is provided. In most applications, however, the lens equation has to be inverted, since we want to compute the image position from a given source position. Different approaches to this issue are presented in the context of lens visualization in the next section.

As already mentioned in the discussion of general relativistic ray tracing, the term  $I_\nu/v^3$  is constant along a light ray, cf. Eq. (9.7). The redshift is identical for the images of a single source, and so are the respective specific intensities. The radiant flux of an image of an infinitesimal source is the product of the specific intensity and the solid angle  $\Delta\Omega$  covered by the image on the “sky”. The specific intensity is identical for the lensing situation and the unlensed case. Therefore, the ratio of the radiant fluxes for a small image and its corresponding unlensed source is solely determined by the ratio of the respective solid angles. The magnification factor is

$$\mu = \frac{\Delta\Omega}{(\Delta\Omega)_0} = \left| \det \frac{\partial\beta}{\partial\theta} \right|^{-1},$$

where the 0-subscript denotes quantities in the absence of a lens. The magnification is governed by the determinant of the Jacobian matrix of the lens mapping.

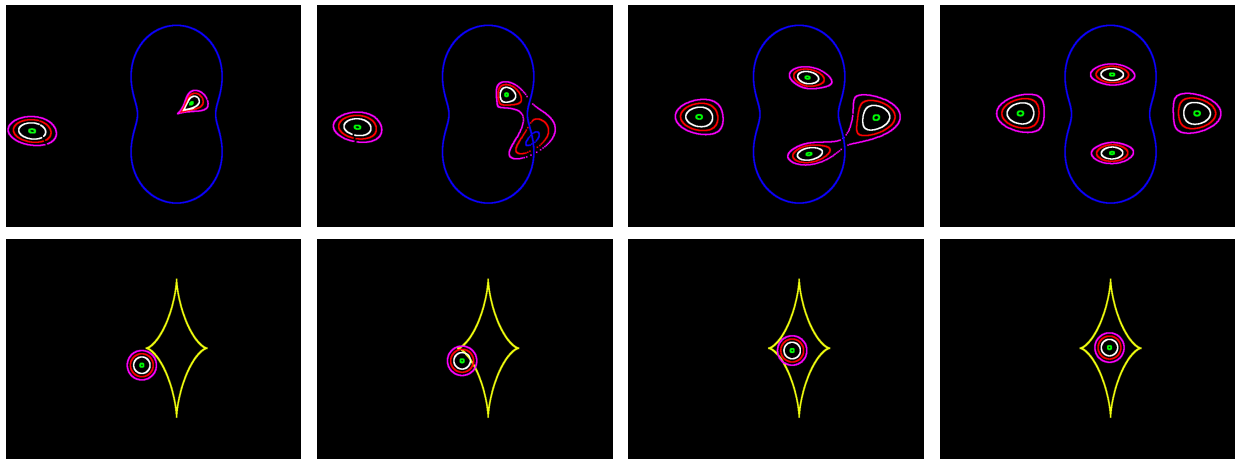
*Critical curves* are defined as regions in the lens plane where the Jacobian vanishes and the magnification factor diverges. This does not imply that the image is infinitely bright because a weighted sum for extended sources has to be computed and effects of wave optics have to be considered for point-like sources. Nevertheless, the image of a source becomes much brighter when it is approaching a critical curve.

The mapping of a critical curve under the lens equation is called *caustic*. The number of images changes by two if, and only if, the source crosses a caustic. Moreover, any transparent mass distribution with finite total mass and with a weak gravitational field produces one single image of a point source if the source is sufficiently misaligned with the lens. In conclusion, such a mass distribution produces an odd number of images for a point-like source. This is not true for lenses which have points of infinite surface density, such as black holes.

## 10.2 Visualization in Standard Lensing Theory

The standard problem is the inversion of the lens equation, i.e., to find all the images for a given position of the source. In some simple cases, this problem can be solved analytically. In general, however, we have to find all the roots of the two-dimensional system of lens equations by means of numerical methods.

Frutos-Alfaro[54] describes an interactive system for visualizing gravitational lenses. If the lens equation reduces to a



**Figure 10.3:** Visualization of gravitational lensing. The lens is a perturbed isothermal sphere. The source consists of colored circles and moves from left to right. The top row shows images of the source and the critical curve (blue) in the image plane. The bottom row illustrates the corresponding positions of the source and the caustics (yellow) in the source plane.

one-dimensional polynomial function, he either computes the roots of the lens equation exactly (for polynomials of degree less than four) or he uses the Bairstow method for iterative root-finding (for polynomials of higher degrees). In all other cases, the image of a source is computed according to *ray shooting*[141]. Ray shooting can be considered as a special case of ray tracing. It maps a uniform grid of points in the lens plane onto the source plane according to the lens equation. The grid points adopt the color of the points which they hit on the source plane. Since each of these mappings can be associated with a light ray, ray shooting can be regarded as a subclass of ray tracing which does not generate secondary rays or shadow rays.

Figure 10.3 shows an example of lens visualization. The images were produced by the interactive system due to Frutos-Alfaro[54]. The deflecting body is an isothermal sphere with a quadrupole perturbation, cf. Appendix B. The source consists of colored circles. The top row shows images of the source in the image plane and the critical curve which is marked blue. The bottom row illustrates the corresponding positions of the source and the caustics in the source plane. From left to right, the source moves across the caustic, increasing the number of images by two.

## 10.3 Direct Application of Ray Tracing

As mentioned in the previous section, ray shooting is a special case of ray tracing. Hence, the visualization of standard lensing can be easily incorporated into a general relativistic ray tracer. Two direct applications of general relativistic ray tracing are investigated.

### 10.3.1 Standard Lensing

The first application implements visualization of standard lensing. This yields the same results as described in the previous section. The software architecture for ray tracing in single chart spacetime is utilized, cf. Sect. 9.2. Again, the idealized

structure of ray tracing, Fig. 9.1, is the basis for the visualization system. The following extensions of non-relativistic Euclidean ray tracing are required.

First, the standard `Ray` class representing a straight light ray is replaced by a class representing a bent light ray. This bent ray is expressed by a polygonal line consisting of two straight parts. The first part reaches from the observer to the lens plane, the second part from the lens plane through the source plane to infinity.

Secondly, the standard ray `Projector` has to be modified. The new ray projector generates the two-piece polygonal line described above. The physics of the lens is supplied by describing the deflection angle. The implemented types of lenses and the underlying physical models are documented in Appendix B.

The implementation of the visualization of standard lensing is based on *RayViS*. Appendix C.3 contains a detailed program documentation.

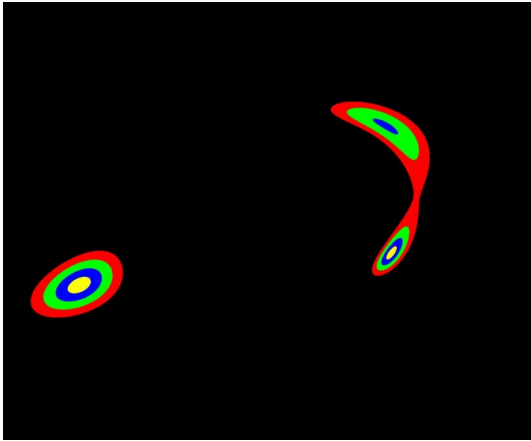
Figure 10.4 shows the visualization of the Chang-Refsdal lens, based on ray tracing. The Chang-Refsdal lens is a superposition of a Schwarzschild lens and a larger-scale matter distribution, such as a galaxy. A description of the Chang-Refsdal lens can be found in Appendix B.

### 10.3.2 Strong Light Deflection

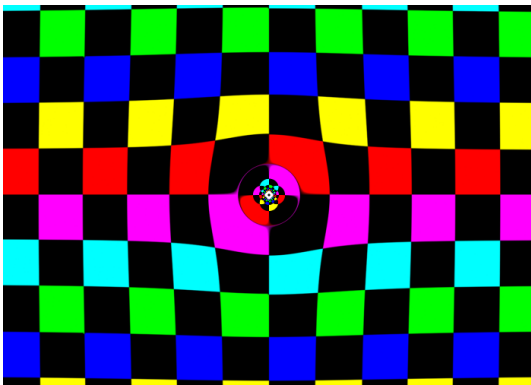
The second application of general relativistic ray tracing implements visualization of strong deflection. This application needs no further extension of the software already developed for general relativistic ray tracing in Chapter 9.

Lensing is visualized via the effects of the lens on the visual appearance of background objects. The main task is to find an appropriate choice for the background objects. For example, a planar surface with a textured structure can be positioned behind the lens, resembling the visualization technique described in Sect. 10.2. The texture enables us to recognize visual distortions induced by the lens.

In other cases, the observer is located close to the lens, for example, close to the horizon of a black hole or even inside



**Figure 10.4:** Visualization of the Chang-Refsdal lens, based on ray tracing.



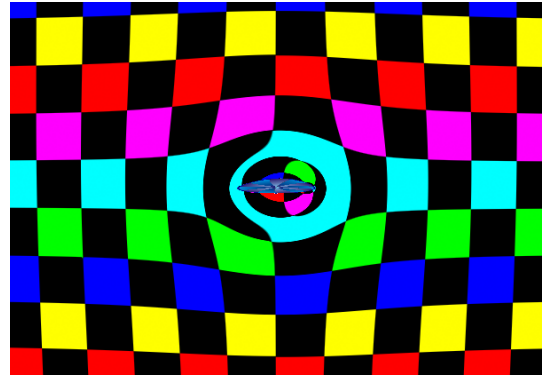
**Figure 10.5:** Strong deflection of light by a Schwarzschild lens.

the region of the gravitating object. Here, a sphere surrounding the observer can be used as a background object. If the sphere reaches into asymptotically flat spacetime and its radius is large enough, the image of the sphere yields the perceived “sky”.

Figure 10.5 is a typical example of the visualization of gravitational lensing via the appearance of a background object. The lens is a Schwarzschild lens, the background object is a checkered plane. The observer, the lens, and the midpoint of the plane lie on a line. The distance from the observer to the lens equals the distance from the lens to the source plane. The Einstein ring is present between the violet and red tiles. Note that here the geodesic equation is completely integrated and no lensing approximation is applied.

Figure 10.6 is another example of the visualization of gravitational lensing. The lens is a rigidly rotating disk of dust with  $\mu = 0.1$ . The observer is close to the symmetry plane of the system. The rotation of the disk induces frame dragging, causing asymmetric distortions on the background texture.

Figure 10.7 documents the view from inside the warp bubble of the Alcubierre metric, as defined in Sect. 9.4. A checkered “sky” surrounds the observer and the bubble. The images show panoramas with a vertical field of view of 120 degree. The horizontal axis covers 360 degree. The center of the im-



**Figure 10.6:** Gravitational lensing by a rigidly rotating disk of dust with  $\mu = 0.1$ .

age corresponds to viewing perpendicular to the direction of motion. The backward view is located at the left part of the image, the forward view at the right part. The velocities in the three images are  $v = 0.9c$ ,  $1.5c$ , and  $4c$ , from top to bottom.

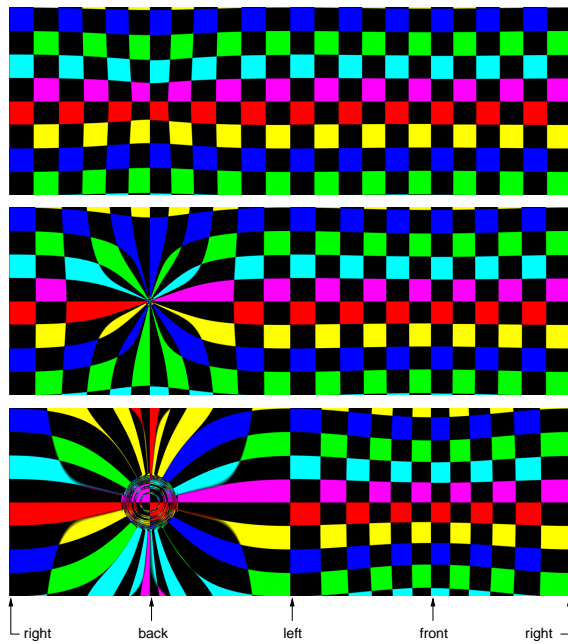
The “sky” is compressed in the forward direction. This behavior is qualitatively similar to the relativistic aberration of light in special relativity. For velocities above the speed of light, a part of the backward “sky” is no longer visible. A conical-shaped region from which no photons can reach the observer is generated behind the warp bubble. The spaceship literally outruns these photons. The middle and lower image in Fig. 10.7 reveal this behavior by contracting parts of the texture in the backward direction to a single point. Artifacts of the numerical integration of the geodesics are noticeable around the backward view for  $v = 4c$ .

## 10.4 Caustic Finder

So far, only the visual distortions of background objects have been used to visualize the structure of the lens. As mentioned before, caustics appropriately describe the qualitative behavior of lenses. A systematic visualization of caustics improves the understanding of lensing.

*Grid search*[141] is a numerical method which enables us to determine caustics. It uses a triangulation of a regular grid in the lens plane. The triangles are mapped onto the source plane via the lensing equation. In the source plane, every point can be associated with the triangles containing this point. The first application allows tracing points on the source plane back to the lens plane. More importantly, the second application counts the number of image multiplicities in the source plane. The number of triangles attached to a point in the source plane is identical to the respective image multiplicities. The caustic lines can be identified by investigating changes of multiplicity in the source plane.

Grid search examines the two-dimensional source plane only and thus is restricted to one-dimensional caustic lines. I adopt some ideas of grid search and alter and extend them to take into account two-dimensional caustic surfaces in three-dimensional space. The respective *caustic finder* for two-dimensional caustic surfaces is presented in the following section.



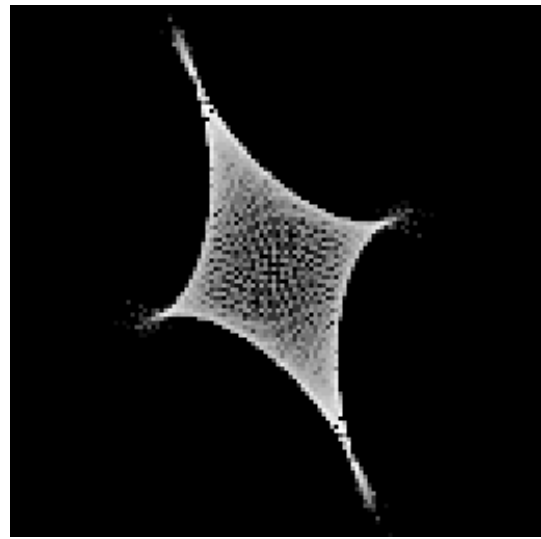
**Figure 10.7:** View from inside the warp bubble. The velocity of the spaceship is  $v = 0.9c$ ,  $1.5c$ , and  $4c$ , from top to bottom.

### 10.4.1 Algorithm

The restriction to a two-dimensional source plane is dropped. A source volume is utilized instead. The ray shooting principle is adopted, cf. Sect. 10.2. Rays are traced from the observer through the image plane into the scene. A clearly defined lens plane is no longer necessary.

A triangle mesh in the image plane cannot be used, since it is problematic to handle the mapping of a triangle in the source volume. In fact, each pixel in the image plane is mapped to the source volume to identify image multiplicities. The source volume itself is partitioned into a three-dimensional grid. Each voxel can be associated with the crossing rays and the corresponding pixels in the image plane. The number of rays attached to a voxel does not directly determine the image multiplicities. The main problem is to decide whether two rays cross the same voxel because of caustics or only because they are nearby already in the image plane. I propose the Euclidean or the angular distance in the image plane as a criterion to distinguish multiplicities. In a typical example with  $500^2$  pixels in the image plane and a  $256^3$  source volume, a standard value for the criterion lies between 10 and 20 pixels difference.

Two light rays crossing the same voxel of the *caustic finder* are counted as multiple images, only if they satisfy this minimal distance in the image plane. This method might introduce some artifacts. On one hand, multiple images might be incorrectly counted if the distance criterion is chosen too small. On the other hand, rays might be identified as single images, even if they represent multiple images. In particular, this can be the case for rays close to a critical curve on the lens plane. However, similar problems might occur in the original grid search method as well. The accuracy of both techniques can be improved by increasing the sampling rates.



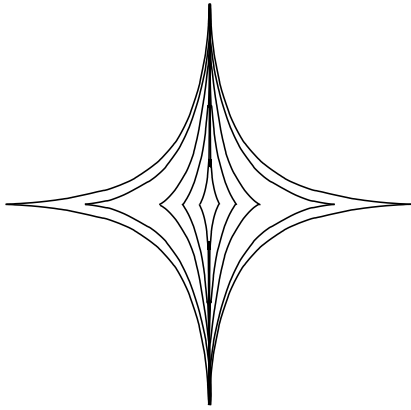
**Figure 10.8:** Caustic structure for the Chang-Refsdal lens. White represents regions of the source with multiple images, black represents regions without image multiplicities.

The actual algorithm for the *caustic finder* is as follows. A specific volume object is included in the scene graph. This volume object stores the number and the initial directions of the light rays crossing a voxel of the object. The ray tracing process is performed in the usual way, i.e., bent light rays are generated according to the metric and ray-object intersections are determined. Whenever a voxel of the *caustic finder* is traversed during intersection tests, a counter is incremented and the position of the pixel on the image plane corresponding to the current ray is attached to this voxel.

In a post processing step, each single voxel is evaluated and the number of rays coming from different areas of the image plane are accumulated. In this way, unintentional counting of nearby rays which cross the same voxel is avoided. The minimal required distance on the image plane is specified by the user. Currently, only a regular grid is implemented as a *caustic finder*, but other mesh structures could be employed as well.

The scalar field containing the image multiplicities is written to file and visualized by an external program. There exist numerous techniques for direct volume visualization, such as ray casting or shear-warp factorization[90]. Shear-warp rendering is mainly used for interactive explorations, whereas the ray casting program provides images of higher quality, for example, for illustrations.

In contrast to general relativistic ray tracing from the previous chapter, the *caustic finder* provides coordinate-dependent data. This might cause interpretation problems in regions of high curvature. Therefore, this visualization technique is best suited for almost flat parts of spacetime, for example, behind the gravitational lens at adequate distance. High computational and memory costs for a fine sampling of the volume grid are another issue. In future work, this problem could be overcome by utilizing an adaptive grid.



**Figure 10.9:** Caustics of the Chang-Refsdal lens for different lens parameters. The caustic lines are based on analytical considerations.

#### 10.4.2 Implementation and Results

The *caustic finder* is implemented as an extension to general relativistic ray tracing in *RayViS*. Both surface and volume objects are supported in *RayViS*. A standard volume object is subclassed to additionally store the image multiplicities. Currently, only a regular grid is implemented as a *caustic finder*, but other mesh structures could be employed as well. A detailed program documentation can be found in Appendix C.3.

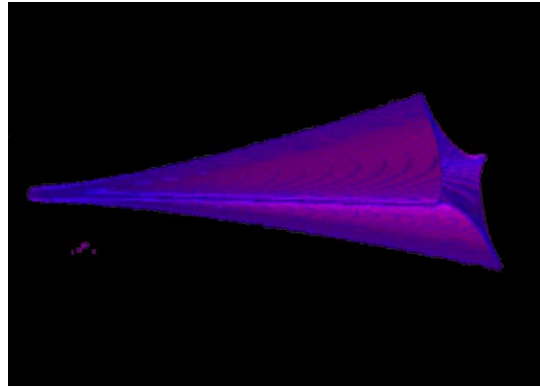
Figure 10.8 shows the structure of the caustics for the Chang-Refsdal lens. Here, the shear perturbation of the Schwarzschild lens contains a rotational part. White represents regions of the source with multiple images, black represents regions without image multiplicities. These two regions are separated by caustic lines. The diamond-shaped caustic is noticeable.

In this application, the *caustic finder* consists of a thin volume which lies in the source plane. The grid has a size of  $128 \times 128 \times 2$  voxels. In this way, the volumetric *caustic finder* emulates standard methods which identify caustic lines. The source plane in Fig. 10.8 is identical to the source plane in Fig. 10.4.

Figure 10.9 shows the caustics for the Chang-Refsdal lens without shear rotation. Here, the caustics originate from analytical calculations. This demonstrates that the numerical result of the *caustic finder* is in agreement with the analytical result.

Figure 10.10 visualizes the structure of the caustic surfaces for the Chang-Refsdal lens. The lens is identical to the one in Fig. 10.8. However, here the *caustic finder* consists of a volume which reaches from the lens plane backwards. The grid has a size of  $128 \times 128 \times 256$  voxels. In contrast to the visualization of caustic lines in Figs 10.8 and 10.9, caustic surfaces are shown.

The scalar field containing the image multiplicities was written to file and visualized by an external volume rendering program based on shear-warp factorization. The figure does not directly display image multiplicities, but utilizes the gradient of the scalar field in order to classify opacities and colors. Hence, the caustics become noticeable because they indicate a change of image multiplicity. Blue denotes a high gradient, red a smaller gradient. The higher gradients in the vicinity of cusps is caused by the *caustic finder* method. Based on the



**Figure 10.10:** Volumetric caustic structure for the Chang-Refsdal lens.

sampling of the image plane, more rays cross a voxel at the cusp than a voxel of the same size which is located at other parts of the caustic. This does not imply that the cusp has a higher image multiplicity.



## Chapter 11

# Visualization of Inner Geometry by Embedding

Since we live in a world of almost flat spacetime, we are not used to curved geometries and have not developed an intuitive understanding of the inner structure of curved four-dimensional manifolds. However, an equivalent surface for a two-dimensional surface can be found in three-dimensional Euclidean space. Equivalent means that the representing surface has the same topology and inner geometry as the original one. The inner geometry is determined by the metric. The equivalent surface is an isometric embedding of the original surface in three-dimensional Euclidean space. We are able to deal with lengths and geometry in three-dimensional Euclidean space and hence have an intuitive comprehension of the geometry of the embedded surface.

Isometric embedding with co-dimension one is a mapping from an  $n$ -dimensional manifold to an  $n$ -dimensional surface in  $(n+1)$ -dimensional space which preserves distances, i.e.,

$$ds_{(n)}^2 = ds_{(n+1)}^2,$$

where  $ds_{(n)}^2$  means the line element in the  $n$ -dimensional manifold and  $ds_{(n+1)}^2$  means the line element on the surface in  $(n+1)$ -dimensional space. In our specific case, the dimension of the surface is  $n=2$  and the three-dimensional space is Euclidean.

Suppose that the embedded two-surface is given by a parametric representation  $\vec{r}(\vec{x})$ . Then, the isometry condition yields the relationship,

$$\begin{aligned} ds_{(2)}^2 &= \langle d\vec{x}, d\vec{x} \rangle_{(2)} = g_{ij(2)} dx^i dx^j \\ &\stackrel{!}{=} \langle d\vec{r}, d\vec{r} \rangle_{(3)} = \langle \vec{r}_{,i}, \vec{r}_{,j} \rangle_{(3)} dx^i dx^j, \end{aligned}$$

which results in a constraint for the partial derivatives of  $\vec{r}(\vec{x})$ :

$$\langle \vec{r}_{,i}, \vec{r}_{,j} \rangle_{(3)} \stackrel{!}{=} g_{ij(2)}. \quad (11.1)$$

The scalar product  $\langle \cdot, \cdot \rangle_{(3)}$  is the standard scalar product in three-dimensional Euclidean space. Within this Chapter, lower case Latin indices are from the set  $\{1, 2\}$ .

This system of partial differential equations is not directly tractable. It is not clear of what type the differential equations are, for example, whether they are hyperbolic or elliptic. The type depends on the actual data for the metric, and a different type might require a different kind of numerical method. Moreover, numerical methods for partial differential equations are usually based on a discretization of the problem domain. The idea is to directly solve the embedding problem on the

mesh of the triangulated surface itself, i.e., the discretization for solving the differential equations and the representation of the surface are identical. This approach was proposed by Schmidt[140] and Nollert and Herold[119].

### 11.1 Embedding of a Triangulated Two-Surface

Any manifold can be partitioned into a set of simplices by triangulation. In the case of a two-dimensional surface, the simplices are triangles. The following construction of the embedding surface is based on a triangulated mesh representation rather than a parameterized representation. In this way, the unknowns are the coordinates of the vertex points  $\vec{r}_i = \vec{r}(\vec{x}_i)$ , not the continuous map  $\vec{r}(\vec{x})$ . A system of algebraic equations has to be solved instead of a system of partial differential equations.

The basic idea is to relate the given metric of the two-dimensional surface to the distance between each single pair of mesh vertices connected by a triangle edge, i.e.,

$$|\vec{r}_a - \vec{r}_b|^2 \approx \left| \int_{t_a}^{t_b} ds \right|^2, \quad (11.2)$$

which is integrated along a path in the embedded surface with the starting point  $\vec{x}(t_a) = \vec{x}_a$  and the ending point  $\vec{x}(t_b) = \vec{x}_b$ .

However, this first approximation induces a systematic error because the Euclidean distance of two vertices is not identical to the actual distance along the path in the surface. The Euclidean distance is always shorter than the length of the connecting path, since the Euclidean distance is the shortest connection in the three-dimensional space, whereas the connecting path has to lie in the two-surface. Furthermore, the evaluation of the integral has to be done numerically and introduces additional computational costs.

### 11.2 Conditions on Triangle Mesh

The systematic error and the numerical problems in Eq. (11.2) can be solved by the following computational scheme.

The right hand side of Eq. (11.2) is changed in a way that allows to compute the Euclidean distance of two vertices, essentially based on the inner geometry of the original surface. Let us consider two vertices connected by a path in two-dimensional coordinate space. A point on the path can be

expressed by a Taylor series expansion around the midpoint of the path  $\vec{r}(t_0)$ , i.e.,

$$\vec{r}(t) = \vec{r}(t_0) + h\dot{\vec{r}}(t_0) + \frac{1}{2}h^2\ddot{\vec{r}}(t_0) + \frac{1}{6}h^3\dddot{\vec{r}}(t_0) + O(h^4),$$

where the path is parameterized by  $t$ ,  $h$  is the distance from the midpoint in parameter space, and a dot means a differentiation with respect to  $t$ . By applying the Taylor series to the distance between two vertices  $\vec{r}_a$  and  $\vec{r}_b$ , we obtain

$$\begin{aligned} |\vec{r}(t_a) - \vec{r}(t_b)|^2 &= 4h^2 \langle \dot{\vec{r}}(t_0), \dot{\vec{r}}(t_0) \rangle \\ &\quad + \frac{4}{3}h^4 \langle \dot{\vec{r}}(t_0), \ddot{\vec{r}}(t_0) \rangle + O(h^6), \end{aligned}$$

with the parameter for the midpoint,  $t_0 = (t_a + t_b)/2$ , and the distance in parameter space,  $h = (t_b - t_a)/2$ . The scalar product is assumed to be the standard scalar product in three-dimensional Euclidean space.

The dot differentiations of  $\vec{r}$  are

$$\begin{aligned} \dot{\vec{r}} &= \vec{r}_{,i}\dot{x}^i, \\ \ddot{\vec{r}} &= \vec{r}_{,ij}\dot{x}^i\dot{x}^j + \vec{r}_{,i}\ddot{x}^i, \\ \dddot{\vec{r}} &= \vec{r}_{,ijk}\dot{x}^i\dot{x}^j\dot{x}^k + 3\vec{r}_{,ij}\dot{x}^i\ddot{x}^j + \vec{r}_{,i}\dddot{x}^i. \end{aligned}$$

With the following relations for the partial derivatives of  $\vec{r}$ ,

$$\begin{aligned} \langle \vec{r}_{,i}, \vec{r}_{,j} \rangle &= g_{ij}, \\ \langle \vec{r}_{,i}, \vec{r}_{,jkl} \rangle &= \Gamma_{ijk}, \\ \langle \vec{r}_{,i}, \vec{r}_{,ijkl} \rangle &= \Gamma_{ijkl} - \langle \vec{r}_{,il}, \vec{r}_{,jk} \rangle, \end{aligned}$$

we finally obtain

$$\begin{aligned} |\vec{r}_a - \vec{r}_b|^2 &= 4h^2 g_{ij}\dot{x}^i\dot{x}^j \\ &\quad + \frac{4}{3}h^4 [(\Gamma_{ijk,l} - \langle \vec{r}_{,il}, \vec{r}_{,jk} \rangle)\dot{x}^i\dot{x}^j\dot{x}^k\dot{x}^l \\ &\quad + 3\Gamma_{ijk}\dot{x}^i\dot{x}^j\dot{x}^k + g_{ij}\dot{x}^i\ddot{x}^j] + O(h^6). \end{aligned} \quad (11.3)$$

The term  $\langle \vec{r}_{,il}, \vec{r}_{,jk} \rangle$  depends on the outer curvature. It has to be calculated by using the embedded surface. In the following section, an iterative scheme which yields intermediate and approximate solutions of the embedding problem is presented. The intermediate solution for the surface can be used to approximately determine the term depending on the outer curvature. The other terms in Eq. (11.3) are based on the inner geometry only.

### 11.2.1 Spherical Topology

So far, the relationship between two vertices connected by a triangle edge has been discussed. How is a system of equations obtained to determine the geometry of a whole embedded mesh?

Let the vector  $\vec{R} = (\vec{r}_1, \vec{r}_2, \dots, \vec{r}_n)$  contain the unknown coordinates of the  $n$  vertices constituting the triangle mesh. The vector  $\vec{F}$  contains the residues of the conditions on an edge, i.e., the difference between left hand side and right hand side of Eq. (11.3). Hence, we have to solve a system of non-linear algebraic equations.

The solution is computed by using the Newton–Raphson method[130], which reduces a system of non-linear algebraic

equations to an iterative solution of linear algebraic equations. Based on the Taylor expansion,

$$\vec{F}(\vec{R} + \delta\vec{R}) = \vec{F}(\vec{R}) + \frac{\partial\vec{F}(\vec{R})}{\partial\vec{R}}\delta\vec{R} + O((\delta\vec{R})^2),$$

we obtain a linear system for the corrections  $\delta\vec{R}$ ,

$$\delta\vec{R} = -\mathbf{J}^{-1}\vec{F}, \quad (11.4)$$

with the Jacobi matrix,

$$\mathbf{J} = \frac{\partial\vec{F}(\vec{R})}{\partial\vec{R}}. \quad (11.5)$$

Equation (11.4) is iterated until the specified accuracy is achieved. The iteration scheme starts with an initial guess for  $\vec{R}$ . The current implementation supports manifolds of spherical topology only. Hence, the initial surface is a sphere.

The inversion of  $\mathbf{J}$  in Eq. (11.4) is only possible if  $\mathbf{J}$  has rank  $3n$ , i.e., if we have  $3n$  independent equations for the  $3n$  unknown coordinates of the  $n$  vertices. In general, this is not the case. For example, let us consider the topology of a closed two-manifold. Euler's formula describes the relationship between the number of vertices  $V$ , edges  $E$ , and faces  $F$ , and genus  $G$  (number of "holes") for a closed two-manifold:

$$V - E + F = 2(1 - G).$$

The sphere is the simplest closed two-manifold; it has genus  $G = 0$ . If the sphere is tessellated by subdividing a tetrahedron, the number of vertices is  $V = n$  and the number of edges  $E = 3n - 6$ . Hence, six equations are missing to allow a solution to Eq. (11.4).

Six additional conditions can be imposed on the triangle mesh by prohibiting translations of the mesh as a whole,

$$\sum_{i=1}^n \delta\vec{r}_i = 0,$$

and rotations of the mesh,

$$\sum_{i=1}^n \vec{r}_i \times \delta\vec{r}_i = 0,$$

where  $\delta\vec{r}_i$  is the correction for vertex  $i$ .

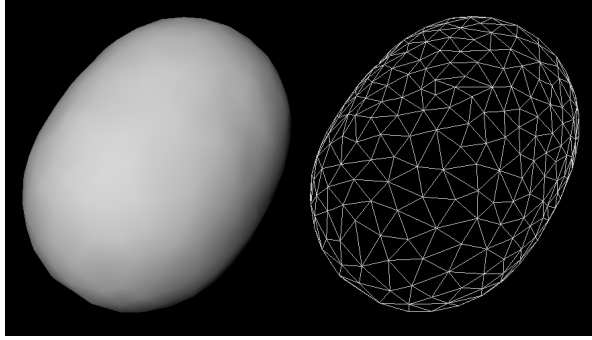
These conditions are directly added to the system of linear equations or the vector  $\vec{F}$ , respectively. Therefore, the embedding problem for two-surfaces of spherical topology can be approached by iteratively solving linear systems.

Surfaces of topologies other than spherical cannot be handled in this way. They require different types of additional conditions, for example, induced by boundary conditions. These conditions could be investigated in future work.

## 11.3 Implementation and Results

The implementation is joint work with Stefan Kulla<sup>1</sup>. It makes use of an object-oriented framework. The program is written

<sup>1</sup>Institute for Astronomy and Astrophysics at the University of Tübingen.



**Figure 11.1:** Embedding surface. The left image shows the visualization of the surface itself, the right image shows the underlying triangle mesh.

in C++. It is based on libraries by Gumhold[65] for handling connectivity in triangle meshes. So far, only the embedding of spherical surfaces has been implemented. A program documentation can be found in Appendix C.4.

Although there exists an implementation of the embedding program by Schmidt[140], we decided to realize a completely new implementation for the following reasons. First, the old program is written in C and does not support the object-oriented programming model, a fact which makes it hard to integrate modern software engineering concepts. Secondly, the previous implementation does not allow for the visualization of sampled, numerically calculated metrics. However, numerical computation of metric data is prevailing in gravitational research. Thirdly, the old implementation does not allow adaptive subdivision of the triangle mesh.<sup>2</sup>

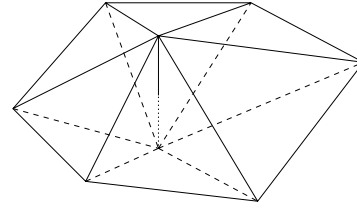
Figure 11.1 shows a result of the developed object-oriented embedding program. The sampled input data is given at equidistant values of  $\theta$  and  $\phi$  and shows the two-surface of the horizon in a numerical relativity simulation.

## 11.4 Discussion

The chosen discretization of the partial differential equations on a triangle mesh can introduce wrong solutions. On the mesh, there exist a concave and a convex solution for the position of a vertex, in contrast to the continuous parameterized representation, which allows only the true solution. Figure 11.2 illustrates that the lengths of the connecting edges are identical in both cases. However, since the iterative scheme starts with a smooth, spherical solution as initial guess, the algorithm tends to converge towards the true, smooth solution. Nevertheless, additional conditions could be investigated in future work in order to prevent the wrong solutions, for example, by including information about the curvature of the mesh in the system of equations.

So far, only surfaces of spherical topology are supported. Surfaces of toroidal and, in particular, flat topology provide very interesting insight in relativistic investigations. A typical example is a flat, meridional cutting surface through the equator of a black hole or neutron star. In these cases, however, the relationship between the number of vertices and

<sup>2</sup>Adaptive subdivision is now supported by the underlying data structures, whereas control of subdivision is not yet implemented in the iterative Newton–Raphson solver.



**Figure 11.2:** Concave and convex solution.

edges is different from that in spherical topology. Different types of boundary conditions are required to achieve a well-conditioned system of equations. These conditions could be investigated in future work.

If the surface has parts of negative Gaussian curvature, a global embedding in Euclidean space might not be feasible, whereas it is guaranteed that a surface (of spherical topology) with positive Gaussian curvature can be embedded. Global embedding of manifolds with negative Gaussian curvature could be realized by two different methods.

The first method partitions the surface in subsections which can be embedded. The problem is to reduce the number of such subsections. Moreover, these pieces have to be connected in a visually pleasant way in order to provide a comprehensible visualization for the user.

The second method rather uses a hyperbolic background space than a Euclidean background space. For a two-dimensional surface which is embedded in a three-dimensional manifold of constant curvature  $K_0$ , the internal curvature  $K_{\text{int}}$  and the external curvature  $K_{\text{ext}}$  are related by

$$K_{\text{int}} = K_{\text{ext}} + K_0,$$

cf. Spivak[150, pages 104 and 136]. The external curvature of an embedded two-surface is the product of the two principal curvatures  $\lambda_1$  and  $\lambda_2$ :

$$K_{\text{ext}} = \lambda_1 \lambda_2.$$

The respective internal curvature depends on the inner geometry only and can be expressed in terms of the Riemann tensor  $R_{ijkl}$  and the metric  $g_{ij}$ , i.e.,

$$K_{\text{int}} = \frac{R_{1212}}{\det(g_{ij})}.$$

Negative internal curvature can be absorbed by a negative curvature of the background space in order to achieve a positive external curvature of the embedded surface. As mentioned before, a positively curved two-surface of spherical topology can always be embedded globally. The main problem is to find a good visualization of hyperbolic space which can be comprehended by the user.



## Chapter 12

# Classical Visualization

In the context of this thesis, *classical visualization* means the visual representation of data from relativistic simulations without taking into account the curvature of spacetime. Techniques known from other fields of scientific visualization can be utilized to illustrate quantities which are specific to relativistic systems.

A prominent difference between relativistic and Newtonian mechanics is abandoning a formal separation of space and time. The mathematical formulation of relativity makes extensive use of covariant descriptions of physical quantities. Therefore, we have to deal with four-dimensional tensors or tensor fields of various orders, especially with Lorentz scalars (tensor of zeroth order), four-vectors (tensors of first order), and second order tensors. The conversion of the abstract tensor concept to a comprehensible visual representation is investigated by Lehle[92] in detail. His extensive discussion of classical visualization shall not be extended to broader range in this chapter.

I rather focus on a specific topic of classical visualization which uses height fields to visualize scalar fields on a two-dimensional domain. As an application, data from numerical simulations of neutron star oscillations by Ruoff[139] are visualized adequately. The data is available as a three-component field on a three-dimensional domain. Each single component is stored in the form of coefficients for spherical harmonics. The components are to be visualized separately because they do not represent a single geometric structure. In this way, the problem is reduced to the visualization of a scalar field in three dimensions.

As proposed by Lehle[92], standard techniques implemented in the commercial visualization system *AVS Express*[8] were tested. Volume rendering and isosurface visualization do not work out very well for this specific application because they show the position of test particles, but fail to visualize the induced perturbations of spacetime. In fact, height fields on hyperplanes cut through the three-dimensional domain prove to show the wave structures in which relativists are most interested. However, the height field created by standard visualization modules shows poor quality because an adaptive generation of the underlying mesh is not supported. Figure 12.3 shows the problem by means of an example mesh.

### 12.1 Triangulation of Height Fields

The areas of poor mesh quality are restricted to regions of high gradients of the scalar field. Hence, image quality can be improved by an adaptive refinement of the mesh in these areas.

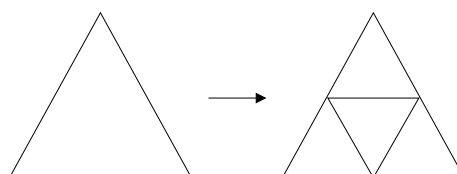


Figure 12.1: Red refinement.

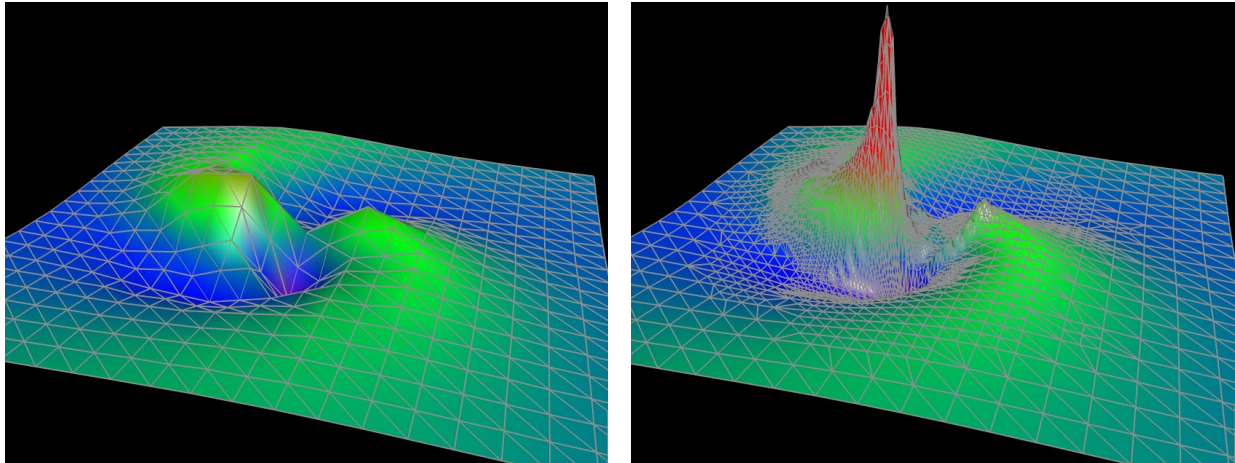
The triangulation of the height field is generated by the following adaptive subdivision scheme. The mesh itself represents a sampling of the true height field, i.e., the true values of the underlying function.

The algorithm starts with a rectilinear grid representing the domain of the height field. A typical number of sample points is  $50^2$ . The rectilinear grid is transformed into a triangle mesh by splitting each rectangle into two triangles. The height profile of the mesh yields the scalar function values. Without loss of generality, the  $x$  and  $y$  coordinates of a vertex hold the coordinates of the triangulation of the domain and the  $z$  coordinate holds the function value—up to scalings along the coordinate axes.

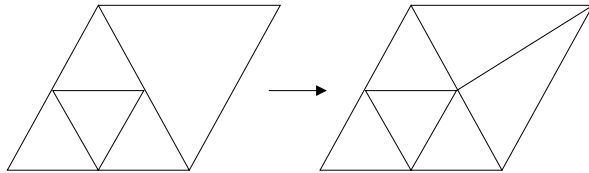
The original mesh is refined by adaptive subdivision. For each triangle, it is checked whether a subdivision is needed or not. This decision is based on an abstract error criterion which is explicitly specified in an actual implementation. A straightforward criterion is the deviation of the true height of an inner point from the height of the triangle at that point. In the current implementation, four points (midpoints of three edges, midpoint of the triangle) are tested for a maximally permitted deviation.

Whenever a triangle violates the error criterion, it is subdivided according to a *red refinement* or a *regular refinement*. Figure 12.1 illustrates red refinement. One triangle is split into four new triangles which arise from the connection between points inserted at the midpoints of the edges of the original triangle. Red refinement preserves the angles of the original triangle and does not create extremely acute or obtuse angles, which cause ill-conditioned linear interpolation within the triangle and thus poor quality of the rendered mesh.

The subdivision scheme is recursively iterated until the desired accuracy or a maximal subdivision level is reached. In a second step, the hanging nodes (t-vertices) are eliminated. Hanging nodes occur when a new vertex is inserted, but connected to the vertices of only one of the two adjacent triangles. Hanging nodes cause cracking of the mesh and thus poor image quality. Hanging nodes are eliminated by a *green refine-*



**Figure 12.3:** Visualization of neutron star data. The left image shows regular sampling, the right image shows adaptive refinement for the same data set.



**Figure 12.2:** Green refinement.

ment or an *irregular refinement*. Figure 12.2 illustrates green refinement. If more than one hanging node is attached to a single edge, all of these nodes are connected to the opposite vertex. This closing scheme is different from the standard closing scheme in red/green refinement, which prevents hanging nodes of order larger than one. Our closing method produces less triangles, but might result in acute or obtuse angles. A more detailed presentation of standard red/green refinement can be found, for example, in work by Bank et al.[9, 10].

## 12.2 Implementation and Results

The implementation of adaptively triangulated height fields makes use of an object-oriented framework. The program is written in C++. It is based on triangle mesh libraries by Gumhold[65]. The libraries manage the geometry and connectivity of the subdivision meshes. The height field is represented by an abstract base class which implements the subdivision scheme and the error criterion on which adaptive refinement is based. Time-dependent data is supported. The functional relationship between the two-dimensional domain and the values of the graph is defined in a derived class. An example is a class representing a height field based on coefficients for spherical harmonics. A detailed program documentation can be found in Appendix C.5.

Figure 12.3 gives an example of height field visualization. The scalar data is visualized by employing both a height field and color coding simultaneously. In this way, the values of the scalar field can be recognized by the user more clearly, in

particular, when the time-dependent evolution of the field is shown.

Figure 12.3 shows the visualization of results from a neutron star simulation[139]. It visualizes a particle which travels around a neutron star and excites the neutron star to oscillations by gravitational interaction and to coupled emission of gravitational waves. The underlying computation is based on a multipole expansion of the particle. Each multipole generates its own oscillation. Summing over all multipoles yields the overall oscillation pattern.

Only the difference between the true metric and the underlying metric of the neutron star is visualized, i.e., only the perturbation of the spacetime which is induced by the particle. One component of the metric perturbations is displayed. The particle travels in the  $x$ - $y$  plane. The cutting surface lies in the  $x$ - $y$  plane as well. The neutron star itself is located at the midpoint of the plane. When the particle is approaching the neutron star, emission of gravitational waves is induced. These waves are superimposed onto the gravitational field of the particle.

Other terms describing the metric perturbation and the time evolution of the system are shown on the accompanying video, cf. Appendix D.1.7.

The produced images demonstrate that an adaptive refinement is indispensable for height fields representing scalar data with high gradients. Nevertheless, the implemented subdivision method might cause some artifacts. The first kind of artifacts is caused by a principle problem in all sampling algorithms. A rapid change in the scalar field between two sampling points might be ignored. This can only be overcome if more information about the behavior of the underlying function is available, such as its derivatives. The second kind of artifacts is induced by the closing scheme, which can produce acute and obtuse triangles for hanging nodes of order larger than one. Isotropic refinement gives rise to the third kind of artifacts. Red refinement generates similar triangles in the projection onto the domain, but elongated triangles for the surface of the height field, especially around peaks. The problem could be overcome by anisotropic refinement, as proposed by Kornhuber and Roitzsch[86] in the context of numerical methods for partial differential equations.

## Chapter 13

# General Relativistic Visualization: Summary and Open Questions

In this second part of the thesis, visualization techniques for curved spacetimes have been discussed. The mathematical structure of curved Lorentzian manifolds is more complex than the structure of the flat spacetime in special relativity. Therefore, a wide variety of visualization approaches has been proposed, reaching from direct representation of visual perception by ray tracing via visualization of inner geometry by embedding to classical visualization of metric quantities. Properties as well as advantages and disadvantages of each approach are discussed in the following sections. Open questions and possible future work are also described.

### 13.1 Ray Tracing and Gravitational Lensing

Ray tracing renders the visual perception of an observer in curved spacetimes. The method is based on geometric optics, where the process of time is reversed. Hence, rays are traced from the eyepoint back into the scene to find intersections with objects, yielding the color of the incoming light for each pixel. Ray tracing employs a *gedanken experiment*. It represents what we could see within a general relativistic setting. Therefore, the ray traced images are observables of an experiment. They have a direct and evident physical meaning. Most importantly, they are independent of the coordinate system which is used to accomplish the computations.

General relativistic ray tracing in a single chart had been used before. The development of this technique is rather advanced and sophisticated; single chart ray tracing is in a consolidation stage. A contribution of this thesis is the description of an open, flexible, and modular ray tracing system. Within an object-oriented framework, single chart general relativistic ray tracing needs only few extensions to a standard three-dimensional Euclidean ray tracer. Previous implementations were specifically designed and developed for general relativistic visualization. Therefore, they lacked the extensive functionality of a complete ray tracing system. Another advantage of the presented implementation is a generic description of scene objects and their motion through spacetime. Moreover, the modular software design allows the simple integration of new metrics and physical systems. Finally, a parallel implementation has been described. Due to high computation costs, parallelization is indispensable, especially for extensive parameter studies and film production.

The flexible design achieves a good usability of ray tracing

as a tool in gravitational research. I have focused on appropriate fields of application and demonstrated them by means of some examples. Ray tracing illustrates the effects of gravitational light bending on scene objects. The properties of spacetime are probed by a vast number of null geodesics, which can even lead to such interesting findings as fractal structures. Ray tracing can be used for investigating null geodesics, where standard tools like Poincaré sections fail. In addition, ray tracing allows to communicate results of gravitational research to colleagues and the public. A disadvantage of ray tracing is the restriction to illustrations from a single viewpoint. However, I have shown that an extensive parameter study can systematically sample the space of viewpoints in order to permit a thorough examination of the properties of spacetime.

Ray tracing has been specifically investigated as a tool for gravitational lensing. It has been shown how models of standard lensing theory can be included in a ray tracing system. Several prominent lenses have been implemented. It has been demonstrated how ray tracing can even be applicable to strong light deflection, which is ignored by standard lensing. Furthermore, a method for finding caustic surfaces has been proposed. It utilizes the infrastructure of the ray tracer to identify regions of different image multiplicity. Caustic surfaces illustrate the qualitative behavior of a gravitational source and give insight from another perspective than ray tracing. A disadvantage is the dependency on a coordinate system.

An extension of ray tracing in a single chart to multiple charts of an atlas has been developed. In this way, non-trivial topologies can be considered. Moreover, the quality of the numerical integration of null geodesics can be improved by choosing several, adequate coordinate systems. The current implementation allows to visualize a simple static wormhole. In future work, a generic implementation of an atlas could be pursued. In this way, spacetimes of complex topologies and even non-orientable manifolds could be investigated.

Another issue of future work is the extension to general relativistic direct volume rendering. The adaptation of standard three-dimensional ray casting should be straightforward. The only fundamental change is the introduction of bent rays instead of straight rays. A much more complex approach would realistically model radiation transport by Monte-Carlo simulations.

Up to now, general relativistic ray tracing is far away from being a real-time application. Due to the ever increasing computational power of processors and the wider availability of massive-parallel computers, however, a chance has been

opened for interactivity. In addition to increasing computer power, rendering performance has to be improved by better algorithms, for example, by exploiting coherence between frames, by pre-computation of some data for null geodesics, and by caching strategies. Moreover, the computational complexity could be reduced by restricting the supported class of spacetimes, in particular, by utilizing possible symmetries.

## 13.2 Embedding

The inner geometry of a two-dimensional spatial hypersurface can be visualized by embedding in flat three-dimensional space. The problem of embedding can be formulated by means of partial differential equations. Partial differential equations have to be discretized to make them tractable for numerical methods. The described embedding algorithm uses the same mesh for discretizing the embedding problem and for the representation of the embedded surface itself.

Embedding provides a visualization that is independent of the underlying coordinate system. A key application is the embedding of hypersurfaces of spacetimes from numerical simulations, for example, the horizons of black holes.

So far, only surfaces of spherical topology are supported. In future work, the embedding of manifolds of different topologies could be pursued, especially, of flat and toroidal topologies. Corresponding boundary conditions have to be formulated to achieve a well-conditioned system of equations.

Another issue is the visualization of surfaces with negative Gaussian curvature. In many cases, these surfaces cannot be embedded globally. This problem could be approached in two ways. In the first way, the negative curvature is “absorbed” by embedding in a hyperbolic background space. The second method partitions the surface into smaller parts which can be embedded. These parts have to be glued together in a second step.

## 13.3 Classical Visualization

Classical visualization means the visual representation of data from relativistic simulations without taking into account the curvature of spacetime. Techniques known from other fields of scientific visualization can be utilized to illustrate quantities specific to relativistic systems. This approach has been evaluated in detail before. Therefore, I have focused on a specific visualization technique which needs improvements within a particular setting.

This visualization method represents scalar data on a two-dimensional domain via height fields. A scheme for an adaptive triangulation and refinement of the mesh representing the height field has been described. Adaptive remeshing generates meshes and final images of good quality, whereas the number of triangles is kept small and interactive frame rates in the rendering of the mesh are possible. Adaptive refinement is required in areas of high gradients; for example, spherical harmonics of typical physical problems can produce fairly high gradients. The actual visualization combines both height fields and color coding. In this way, wave propagation in time-dependent data is clearly noticeable.

In future work, mesh quality could be further improved by applying other subdivision schemes, especially anisotropic refinement.



## Chapter 14

# Conclusion

New and improved methods for the visualization of the flat four-dimensional spacetime of special relativity and the curved four-dimensional spacetimes of general relativity have been presented. Special relativity and general relativity are connected by the fact that a local Lorentz frame can always be found, even in curved spacetime. Therefore, ideas and results from special relativity can be adopted in a local scale within general relativity. Nevertheless, the mathematical structure of curved spacetime is much more complex than the structure of the flat Minkowski spacetime. Hence, general relativistic visualization comprises a wider range of fundamental visualization techniques.

One principal visualization method shows the environment as seen by an observer situated in the simulated system. This conceptually simple approach is a modern type of a *gedanken experiment*. Both general relativistic ray tracing and the described techniques for special relativistic visualization are based on this footing.

Special relativistic radiosity, texture-based special relativistic rendering, and image-based special relativistic rendering have been introduced as novel rendering methods. A four-dimensional extension of special relativistic ray tracing has been developed. It has been shown how relativistic effects on illumination can be incorporated in all known rendering techniques. It has been demonstrated that the Doppler and searchlight effects dominate visual perception. Furthermore, navigation techniques for special relativity, especially the *relativistic-vehicle-control* metaphor, and virtual environments have been proposed. With the contributions of this thesis, special relativistic visualization has achieved a highly developed and refined stage. I think that further improvement is restricted to changes in details. Since all fundamental rendering approaches from standard non-relativistic computer graphics have been adapted to the special relativistic situation, it will be difficult to find totally new special relativistic rendering techniques. However, in my opinion there is one great challenge left: how can we describe accelerating extended bodies on physical grounds, how can animation be included in special relativistic visualization?

Like special relativistic visualization, general relativistic ray tracing directly renders visual perception. Ray tracing of single chart spacetimes also is in a highly refined state. A complete and practical ray tracing system has been developed. By means of two examples, I have demonstrated that ray tracing can be used as an applicable visualization tool for gravitational physics. Moreover, gravitational lensing has been shown to be a field of application for general relativistic ray tracing. The *caustic finder* has been proposed as a numerical method based

on ray tracing. There still remain interesting possible improvements. I think that both the complete implementation of an atlas and real-time rendering could significantly advance general relativistic ray tracing.

General relativistic visualization offers further and more abstract fundamental visualization techniques besides direct rendering of visual perception. The inner geometry of two-dimensional spatial hypersurfaces can be visualized by embedding. The implemented method provides isometric embedding of surfaces with spherical topology into three-dimensional Euclidean space. I regard embedding as a very intuitive and appealing principal visualization approach. The developed algorithms represent only a first example; a much wider class of surfaces representing interesting physical systems should be included, in particular, because embedding is specifically suited for visualizing data from simulations in numerical relativity.

In classical visualization, the connection between relativistic quantities and standard visualization techniques had been investigated before. The development of classical visualization is tightly connected to the development of standard techniques in other fields of visualization. The visualization of two-dimensional and three-dimensional scalar fields as well as the visualization of two-dimensional, second order tensor fields can almost perfectly be accomplished by existing techniques. In this thesis, height field visualization has been enhanced by adaptive subdivision. Principal challenges concern the visualization of tensor fields of higher dimension or higher order. Respective advances would also help other fields of visualization.

In my opinion, visualization presents a most appropriate approach to complex theories such as Einstein's theory of relativity. Visualization allows to extend our perception beyond our everyday experience and thus provides an intuitive understanding of scientific models. I think that intuition is particularly improved by establishing a tight link between the user and the visualization environment. Therefore, real-time applications and interactivity should be employed whenever possible; human-computer interfaces and the inclusion of real world elements play an ever increasing role. In conclusion, scientists benefit from an intuitive and better understanding of research data. Moreover, visualization has proven to be a very successful means of communicating scientific ideas and results to the public.



## Appendix A

# Special Relativistic Transformation of Radiance and Irradiance

### A.1 Derivation of the Transformation of Radiance

The following derivation of the searchlight effect is based on a photon-counting technique. A similar approach can be found in articles by Peebles[124], McKinley[99, 100], and Kraus[89].

Let us consider two inertial frames of reference,  $S$  and  $S'$ , with  $S'$  moving with velocity  $v$  along the  $z$  axis of  $S$ . Suppose the observer  $O$  is at rest relative to  $S$  and the observer  $O'$  is moving with speed  $v$  along the  $z$  axis of  $S$ . The usual Lorentz transformation along the  $z$  axis connects frames  $S$  and  $S'$ .

In reference frame  $S$ , consider a photon with circular frequency  $\omega$ , wavelength  $\lambda$ , energy  $E$ , and wave vector  $\vec{k} = (\omega \sin \theta \cos \phi, \omega \sin \theta \sin \phi, \omega \cos \theta)$  with spherical coordinates  $\theta$  and  $\phi$ .

In frame  $S'$ , the circular frequency is  $\omega'$ , the wavelength is  $\lambda'$ , the energy is  $E'$ , and the wave vector is  $\vec{k}' = (\omega' \sin \theta' \cos \phi', \omega' \sin \theta' \sin \phi', \omega' \cos \theta')$ . The expressions for the Doppler effect and the aberration connect these two representations, cf. Chapter 4:

$$\lambda' = \lambda D, \quad (\text{A.1})$$

$$\omega' = \omega/D, \quad (\text{A.2})$$

$$E' = E/D, \quad (\text{A.3})$$

$$\cos \theta' = \frac{\cos \theta - \beta}{1 - \beta \cos \theta}, \quad (\text{A.4})$$

$$\phi' = \phi, \quad (\text{A.5})$$

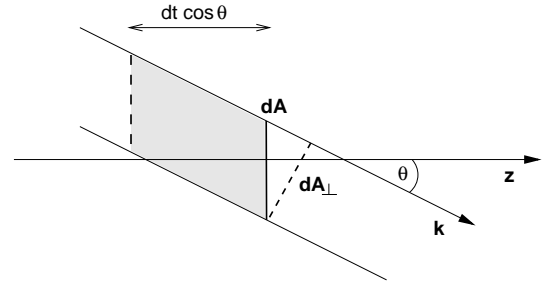
$$D = \frac{1}{\gamma(1 - \beta \cos \theta)}, \quad (\text{A.6})$$

where  $D$  is the Doppler factor,  $\gamma = 1/\sqrt{1 - \beta^2}$  and  $\beta = v$ .

Radiance is the radiant power per unit of foreshortened area emitted into a unit solid angle. A detector at rest in  $S$  will measure the energy-dependent radiance

$$L_E(E, \theta, \phi) = \frac{d\Phi}{dE dA_{\perp} d\Omega},$$

where  $\Phi$  is the radiant power or radiant flux,  $E$  is the energy,  $d\Omega$  is the solid angle, and  $dA_{\perp}$  is the area  $dA$  of the detector projected along the radiation direction  $(\theta, \phi)$ . The radiant flux  $\Phi$  is the radiant energy per unit time. Accordingly, the



**Figure A.1:** Photons with propagation direction along the wave vector  $\vec{k}$ . The area of the detector is denoted  $dA$  and is perpendicular to the  $z$  axis,  $dA_{\perp}$  is the projection of  $dA$  along the radiation direction. The shaded volume  $dV$  contains the photons passing  $dA$  between time  $t_0$  and time  $t_0 + dt$ .

wavelength-dependent radiance is

$$L_{\lambda}(\lambda, \theta, \phi) = \frac{d\Phi}{d\lambda dA_{\perp} d\Omega}, \quad (\text{A.7})$$

with the wavelength  $\lambda$ .

In reference frame  $S$ , consider a group of photons,  $dN$  in number, with energies between  $E$  and  $E + dE$  and propagation directions in the element of solid angle  $d\Omega$  around  $(\theta, \phi)$ . Here, the energy-dependent radiance is

$$L_E(E, \theta, \phi) = \frac{dNE}{dE dA_{\perp} d\Omega dt},$$

or

$$dN = \frac{L_E(E, \theta, \phi)}{E} dE dA_{\perp} d\Omega dt.$$

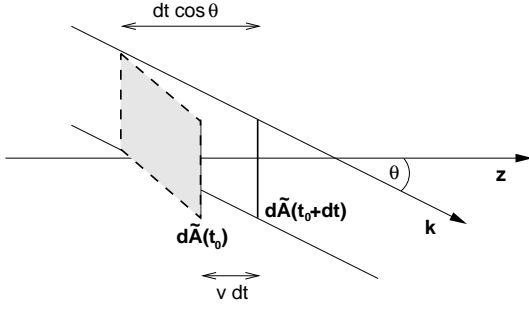
We choose the area  $dA$  to be perpendicular to the  $z$  axis so that

$$dA_{\perp} = dA \cos \theta.$$

The  $z$  component of the velocity of the photons is  $\cos \theta$ . The photons passing  $dA$  between time  $t_0$  and time  $t_0 + dt$  are contained in the shaded volume  $dV$  in Fig. A.1:

$$dV = dA dt \cos \theta.$$

Consider another area  $d\tilde{A}$  having the same size and orientation as  $dA$ . Still in reference frame  $S$ , suppose  $d\tilde{A}$  is moving



**Figure A.2:** Photons with propagation direction along the wave vector  $\vec{k}$ . The area  $d\tilde{A}$  is moving with velocity  $v$  along the  $z$  axis. The shaded volume  $d\tilde{V}$  contains the photons passing  $d\tilde{A}$  between  $t_0$  and  $t_0 + dt$ .

with velocity  $v$  along the  $z$  axis. The photons passing  $d\tilde{A}$  between  $t_0$  and  $t_0 + dt$  are contained in the shaded volume in Fig. A.2:

$$d\tilde{V} = dA dt (\cos \theta - v) = \frac{\cos \theta - \beta}{\cos \theta} dV.$$

The ratio of the number of photons passing  $d\tilde{A}$  in the time interval  $dt$  and the number of photons passing  $dA$  is the same as the ratio of the volume  $d\tilde{V}$  and the volume  $dV$ :

$$d\tilde{N} = \frac{L_E(E, \theta, \phi)}{E} dE d\Omega dt \cos \theta d\tilde{A} \frac{\cos \theta - \beta}{\cos \theta}. \quad (\text{A.8})$$

Now consider the same situation in reference frame  $S'$ . The area  $d\tilde{A}$  is at rest in  $S'$ . The time interval is

$$dt' = dt / \gamma. \quad (\text{A.9})$$

The number of photons counted does not depend on the frame of reference, i.e.,

$$d\tilde{N} = d\tilde{N}' = \frac{L'_E(E', \theta', \phi')}{E'} dE' d\Omega' dt' \cos \theta' d\tilde{A}'. \quad (\text{A.10})$$

From Eqs. (A.8) and (A.10) we obtain

$$\frac{L_E(E, \theta, \phi)}{L'_E(E', \theta', \phi')} = \frac{E}{E'} \frac{dE'}{dE} \frac{d\Omega'}{d\Omega} \frac{dt'}{dt} \frac{\cos \theta'}{\cos \theta - \beta} \frac{d\tilde{A}'}{d\tilde{A}}. \quad (\text{A.11})$$

Since the area  $d\tilde{A}$  is perpendicular to the separating velocity, it is not changed by Lorentz transformations:

$$d\tilde{A}' = d\tilde{A}. \quad (\text{A.12})$$

With Eqs. (A.4) and (A.5), the transformed solid angle is

$$\frac{d\Omega'}{d\Omega} = \frac{\sin \theta'}{\sin \theta} \frac{d\theta'}{d\theta} = \frac{d(\cos \theta')}{d(\cos \theta)} = \frac{1}{\gamma^2 (1 - \beta \cos \theta)^2} = D^2. \quad (\text{A.13})$$

Using Eqs. (A.3), (A.4), (A.9), (A.12), (A.13), and (A.11), we obtain

$$\frac{L_E(E, \theta, \phi)}{L'_E(E', \theta', \phi')} = D^3 = \frac{E^3}{E'^3}.$$

Please observe that this result is the special relativistic case of a generalized property in curved spacetime, Eq. (9.7).

With the relation between energy and wavelength,

$$\lambda = \frac{hc}{E}, \quad d\lambda = -\frac{hc}{E^2} dE,$$

and with

$$L_\lambda(\lambda, \theta, \phi) |d\lambda| = L_E(E, \theta, \phi) |dE|,$$

we get

$$L_\lambda(\lambda, \theta, \phi) = L_E(E, \theta, \phi) \frac{E^2}{hc}.$$

Ultimately, then, the transformation expression for the wavelength-dependent radiance is

$$\frac{L_\lambda(\lambda, \theta, \phi)}{L'_\lambda(\lambda', \theta', \phi')} = D^5. \quad (\text{A.14})$$

From this equation the transformation law for the following integrated quantity is easily obtained. With the use of Eq. (A.1), the transformed radiance is

$$\begin{aligned} L(\theta, \phi) &= \int_0^\infty L_\lambda(\lambda, \theta, \phi) d\lambda \\ &= D^4 \int_0^\infty L'_\lambda(\lambda', \theta', \phi') d\lambda' \\ &= D^4 L'(\theta', \phi'). \end{aligned} \quad (\text{A.15})$$

## A.2 Incident Irradiance

In this section, the derivation of Eqs. (4.10) and (4.12) is presented.

First, consider a finite light source which is at rest in frame  $S$ . With Eq. (A.7), the radiant flux emitted by the light source can be obtained in terms of the wavelength-dependent radiance:

$$d\Phi = L_\lambda(\lambda) dA_\perp^{\text{light}} d\Omega^{\text{obj}} d\lambda, \quad (\text{A.16})$$

where  $dA_\perp^{\text{light}}$  is the area of the projected surface patch of the light source and  $d\Omega^{\text{obj}}$  is the solid angle of the illuminated surface patch of the object as seen from the position of the light source.

Now consider the same situation in frame  $S'$  in which the object is at rest. The radiant flux on the surface patch of the object is

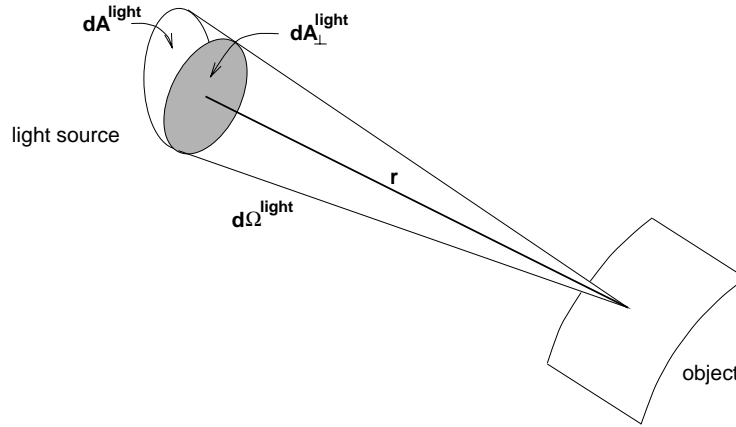
$$d\Phi' = L'_\lambda(\lambda') dA_\perp^{\text{obj}'} d\Omega^{\text{light}'} d\lambda', \quad (\text{A.17})$$

with the projected area  $dA_\perp^{\text{obj}'}$  on the object and the solid angle  $d\Omega^{\text{light}'}$  of the surface patch of the light source as seen from the position of the object. Using Eqs. (A.14) and (A.17), we obtain

$$d\Phi' = \frac{1}{D^5} L_\lambda(\lambda) dA_\perp^{\text{obj}'} d\Omega^{\text{light}' } d\lambda'.$$

With the definition in Eq. (3.1), the incident irradiance emitted from the small solid angle  $d\Omega^{\text{light}'}$  onto the surface patch of the object is

$$dE_\lambda^i(\lambda') = \frac{d\Phi'}{dA^{\text{obj}' } d\lambda'} = \frac{L_\lambda(\lambda) dA_\perp^{\text{obj}'}}{D^5 dA^{\text{obj}'}} d\Omega^{\text{light}'}. \quad (\text{A.18})$$



**Figure A.3:** Geometry of the surface patch of the light source in its rest frame  $S$ . The solid angle is given by  $d\Omega^{\text{light}} = dA_{\perp}^{\text{light}}/r^2$ . The distance between the light source at emission time and the surface patch of the object at absorption time is denoted  $r$ .

The area  $dA^{\text{obj}'}$  of the surface patch is related to the projected area  $dA_{\perp}^{\text{obj}'}$  by

$$dA_{\perp}^{\text{obj}'} = dA^{\text{obj}'} \cos \alpha', \quad (\text{A.19})$$

with the angle  $\alpha'$  between the surface normal and the incident light.

With Eq. (A.13), the solid angle  $d\Omega^{\text{light}'}$  is transformed into the frame  $S$  of the light source. Furthermore,  $d\Omega^{\text{light}'}$  is expressed in terms of the projected area on the light source and of the distance between the light source and the surface patch, as shown in Fig. A.3:

$$d\Omega^{\text{light}'} = D^2 d\Omega^{\text{light}} = D^2 \frac{dA_{\perp}^{\text{light}}}{r^2} = dA_{\perp}^{\text{light}} \left(\frac{D}{r}\right)^2. \quad (\text{A.20})$$

The lightlike connection of the emission event at the light source and the absorption event at the object has the same direction as the wave vector that describes the photons. Therefore, the distance  $r$  is transformed in the same way as the circular frequency  $\omega$ , see Eq. (A.2). By following this reasoning or by explicit Lorentz transformation of the separating vector between the emission event and the absorption event, we get:

$$r' = r/D. \quad (\text{A.21})$$

Using Eqs. (A.18), (A.19), (A.20), and (A.21), we obtain the incident wavelength-dependent irradiance originating from a small area of the light source:

$$dE_{\lambda}^i(\lambda') = \frac{1}{D^5} \frac{\cos \alpha'}{r'^2} L_{\lambda}(\lambda) dA_{\perp}^{\text{light}}.$$

By integrating over the area of the whole light source, we get the wavelength-dependent irradiance produced by this finite light source:

$$E_{\lambda}^i(\lambda') = \int \frac{1}{D^5} \frac{\cos \alpha'}{r'^2} L_{\lambda}(\lambda) dA_{\perp}^{\text{light}}. \quad (\text{A.22})$$

Now, consider a very small, yet finite light source described by its wavelength-dependent intensity  $I_{\lambda}$ . With

Eqs. (3.1) and (A.16), the wavelength-dependent radiance and the wavelength-dependent intensity from the area  $dA_{\perp}^{\text{light}}$  are related by

$$dI_{\lambda}(\lambda) = L_{\lambda}(\lambda) dA_{\perp}^{\text{light}}. \quad (\text{A.23})$$

With Eq. (A.22) and after integrating over the area of the small light source, we find the wavelength-dependent irradiance on the object:

$$\begin{aligned} E_{\lambda}^i(\lambda') &= \int \frac{1}{D^5} \frac{\cos \alpha'}{r'^2} L_{\lambda}(\lambda) dA_{\perp}^{\text{light}} \\ &= \frac{1}{D^5} \frac{\cos \alpha'}{r'^2} \int L_{\lambda}(\lambda) dA_{\perp}^{\text{light}} \\ &= \frac{1}{D^5} \frac{\cos \alpha'}{r'^2} I_{\lambda}(\lambda). \end{aligned}$$

This equation holds even for the limit of an infinitesimal light source. Therefore, we obtain the wavelength dependent irradiance due to a point light source:

$$E_{\lambda}^i(\lambda') = \frac{1}{D^5} \frac{\cos \alpha'}{r'^2} I_{\lambda}(\lambda).$$

Accordingly, the irradiance is

$$E^i = \frac{1}{D^4} \frac{\cos \alpha'}{r'^2} I,$$

where  $I$  is the intensity of the light source.



# Appendix B

## Implemented Gravitational Lenses

In this chapter, the gravitational lenses which have been implemented as visualization modules in *RayViz* are described. I briefly present the underlying models for the distribution of surface mass density and the resulting deflection angles. More detailed information on the lensing models can be found, for example, in Schneider et al.[141] and Frutos-Alfaro[54]. The actual implementation is documented in Sect. C.3.

In the following section, models of axially symmetric lenses are presented. Due to their simplicity, axially symmetric lenses are frequently discussed in the literature, for example, as first, simple models of observed gravitational lensing or in statistical gravitational lensing. In Sect. B.2, quadrupole lenses are considered. They are created by a superposition of the gravitational field of a symmetric lens and a quadrupole perturbation. They describe a combination of a symmetric lens and a larger matter distribution, for example, a star within a larger galaxy.

### B.1 Axially Symmetric Lenses

An axially symmetric lens has a volume density which is invariant with respect to rotations about the optical axis. In particular, any spherical mass distribution is axially symmetric.

For axially symmetric lenses, the lens equation (10.1) reduces to a one-dimensional form because all light rays are contained in the plane defined by the observer, the center of the lens, and the source. By applying the surface mass density  $\Sigma(\xi) = \Sigma(|\vec{\xi}|)$  to the deflection angle for a general lens, Eq. (10.2), and by integrating over the lens plane, we obtain the deflection for an axially symmetric lens,

$$\hat{\alpha}(\xi) = \frac{4G}{\xi} 2\pi \int_0^\xi \xi' \Sigma(\xi') d\xi' = \frac{4GM(\xi)}{\xi}. \quad (\text{B.1})$$

The mass enclosed by a circle of radius  $\xi$  is denoted  $M(\xi)$ . The deflection angle is just the Einstein angle for the enclosed mass  $M(\xi)$ .

#### B.1.1 Point Lens

The Schwarzschild lens represents the deflection by a point mass, cf. Chapter 10. The deflection angle is

$$\hat{\alpha}(\xi) = \frac{4GM}{\xi}.$$

#### B.1.2 Singular Isothermal Sphere

The singular isothermal sphere[15] has the surface mass density

$$\Sigma(\xi) = \frac{\sigma_v^2}{2G} \xi^{-1}.$$

This lens model has infinite mass density at the center and infinite total mass. Nevertheless, the singular isothermal sphere reproduces the matter distribution of galaxies quite well, especially for larger values of  $\xi$ . The parameter  $\sigma_v$  is the line-of-sight velocity dispersion of stars in the galaxy.

With the use of Eq. (B.1), we obtain the deflection angle

$$\hat{\alpha}(\xi) = 4\pi\sigma_v^2.$$

#### B.1.3 Non-Singular Isothermal Sphere

The model of the singular isothermal sphere can be generalized by taking into account the core radius of the galaxy[73]. The singularity in the mass density of the singular isothermal sphere is removed by adding the core radius  $r_c$ . The mass density of the resulting non-singular isothermal sphere is

$$\Sigma(\xi) = \frac{\sigma_v^2}{2G} \frac{1}{\sqrt{\xi^2 + r_c^2}}.$$

The deflection angle is

$$\hat{\alpha}(\xi) = 4\pi\sigma_v^2 \frac{\sqrt{\xi^2 + r_c^2} - r_c}{\xi}.$$

#### B.1.4 Hubble Model

The Hubble model[15] is an empirical model for a galactic lens. Its mass density is

$$\Sigma(\xi) = \frac{\Sigma_0}{(1 + \xi/r_0)^2},$$

where  $r_0$  is the length scale and  $\Sigma_0$  is the mass density at the center of the galaxy. The deflection angle is

$$\hat{\alpha}(\xi) = 8\pi G \Sigma_0 r_0^2 \frac{1}{\xi} \left( \ln \left( \frac{\xi + r_0}{r_0} \right) - \frac{\xi}{\xi + r_0} \right).$$

### B.1.5 King Model

The King model[15] is a semi-empirical model for a galactic lens. Its mass density is

$$\Sigma(\xi) = \frac{\Sigma_0}{1 + \xi^2/r_c^2},$$

where  $r_c$  is the core radius and  $\Sigma_0$  is the mass density at the center of the lens. The deflection angle is

$$\hat{\alpha}(\xi) = 4\pi G \Sigma_0 r_c^2 \frac{1}{\xi} \ln \left( \frac{\xi^2 + r_c^2}{r_c^2} \right).$$

### B.1.6 Spiral Lens

The mass density for a spiral galaxy[80] is given by

$$\Sigma(\xi) = \Sigma_0 e^{-\xi/\xi_0},$$

where  $\xi_0$  is the length scale and  $\Sigma_0$  is the mass density at the center of the galaxy. The deflection angle is

$$\hat{\alpha}(\xi) = 8\pi G \frac{\xi_0}{\xi} \left( -(\xi + \xi_0)e^{-\xi/\xi_0} + \xi_0 \right).$$

## B.2 Quadrupole Lenses

Quadrupole lenses are created by a superposition of the gravitational field of a symmetric lens and a quadrupole perturbation. The symmetric matter distribution is perturbed by a larger-scale gravitational field which is approximated by Taylor terms up to second order.

In general, the deflection angle of a perturbed system can be formulated as

$$\hat{\alpha}(\vec{\xi}) = \hat{\alpha}_A(\vec{\xi}) + \hat{\alpha}_P(\vec{\xi}), \quad (\text{B.2})$$

where  $\hat{\alpha}_A(\vec{\xi})$  is the axially symmetric part and  $\hat{\alpha}_P(\vec{\xi})$  is the perturbed part.

The Taylor expansion of the perturbation up to quadratic terms yields

$$\hat{\alpha}_P(\vec{\xi}) = \hat{\alpha}_P(0) + \mathbf{Q}(\vec{\xi}),$$

if the coordinate system is chosen in a way that its origin coincides with the center of the axially symmetric lens. The matrix  $\mathbf{Q}$  is the quadrupole matrix. If the orientation of the coordinate system is chosen appropriately, the quadrupole matrix takes the diagonal form,

$$\mathbf{Q} = \begin{pmatrix} \kappa_P + \gamma_P & 0 \\ 0 & \kappa_P - \gamma_P \end{pmatrix}, \quad (\text{B.3})$$

where  $\kappa_P$  is a measure for the mass density and  $\gamma_P$  is a measure for the shear.

### B.2.1 Chang-Refsdal Lens

The perturbation of the gravitational field of a point mass is called Chang-Refsdal lens. A prominent example of a perturbed Schwarzschild lens is microlensing, i.e., lensing by a

single star in a galaxy. The Chang-Refsdal lens yields the deflection angle

$$\hat{\alpha}(\vec{\xi}) = 2R_S \frac{\xi_0}{\xi^2} + \mathbf{Q}(\vec{\xi}).$$

The quadrupole matrix is defined in Eq. (B.3).



# Appendix C

## Program Documentation

### C.1 Special Relativistic Polygon Rendering

The implementation of special relativistic polygon rendering is called *Virtual Relativity*. The two main objectives of the *Virtual Relativity* project are high performance for interactive real-time simulations and portability.

*Virtual Relativity* is implemented in C++. The following libraries are used: OpenGL or Mesa; QvLib, ImageMagick, and Glut; TclTk or MFC (Microsoft Foundation Classes). MFC is available for Window 9x and Windows NT only, the other libraries are available for Windows 9x, Windows NT, Linux, and Unix. Rendering is based on OpenGL 1.1 [183] or, alternatively, Mesa[103]. QvLib[155] allows to read three-dimensional scenes in the VRML 1.0 format[12]. VRML 1.0 is based on the OpenInventor[178] file format. ImageMagick[31] provides a transparent interface to all standard two-dimensional image formats. It is used for reading texture information and writing rendered images and movies.

Three different kinds of user front ends to *Virtual Relativity* are available. The first front end uses Glut[60], which provides a simple interface to OpenGL, including window handling, pop-up menus, and support of input devices. The Glut implementation of *Virtual Relativity* mainly serves as a simple experimental cross-platform environment. It lacks a sophisticated design of graphical elements, for example, a menu bar and dialog boxes. The second front end provides a complete platform-independent GUI (graphical user interface). It is based on TclTk[158] for the GUI elements and Togl[123] for the handling of the OpenGL rendering context. The third front end is specific to Windows 9x and Windows NT and is based on MFC.

#### C.1.1 Scene Graph

*Virtual Relativity* supports VRML 1.0 except for the geometric primitives sphere, cone, cylinder, and cube. These have to be replaced by a polygonal representation in the form of indexed face sets. The following types of nodes are added to the VRML format: RELmaterialConstant, RELmaterialPlanck, RELmaterialRGB, and RELmaterialSampled. These nodes describe the spectral dependency of the reflectance or emission properties of a material. They correspond to perfectly Lambertian reflection, blackbody radiation, a sum of Gaussian spectra with a given RGB value, and a sampled spectrum, respectively.

VRML organizes the three-dimensional scene in a scene graph. The scene graph is an acyclic directed graph which

contains geometry, transform, property, and group nodes. The structure of the scene graph is very well suited for the needs of standard non-relativistic graphics.

However, the relativistic transformation does not commute with standard affine transformations. Hence, the scene graph has to be “flattened” by expanding it to a tree and applying all modeling transformations. Shared instancing of scene objects is lifted by adding copies of these objects. (Shared instancing occurs for objects having more than one parent in the scene graph). By using a flattened scene tree, the rendering pipeline is simplified and does not require any modeling transformations. Therefore, the relativistic transformation can be computed before the standard rendering pipeline is applied.

#### C.1.2 Program Structure

The connection and data flow between the major parts of *Virtual Relativity* is illustrated in Fig. C.1.

User-specific parameters are stored in `Parameters`. `Init` evaluates these parameters and sets the respective properties in the main parts of the program.

`RelObserver` represents the relativistic observer and is the central element of relativistic computations. It holds the information about the four-position, four-velocity, proper time, and orientation of the observer. Based on this data, `RelObserver` calculates the transformation of the vertices and the Doppler factors. It computes the trajectory of the observer through spacetime by solving the equation of motion.

The scene graph is described in Sect. C.1.1. The root of the scene graph is a group node `DvGroup`.

Window handling and viewing functionality are implemented in the abstract base class `DvViewer`. `SimpleGlutViewer` is derived from `DvViewer` and implements the Glut interface described above. Accordingly, `ToglViewer` and `MfcViewer` implement the TclTk/Togl and MFC versions, respectively. All these viewers allow the user to steer the relativistic vehicle via keyboard or (space-) mouse.

The handling of rendering is accomplished by `DvDisplayManager`, which also provides the connection to the relativistic observer (`RelObserver`), the scene graph (`DvGroup`), and the viewer (`DvViewer`). A timer function initiates the calculation of new positions of the observer in equidistant time intervals. In this way, the motion of the user through the virtual world is established.

The following standard methods are implemented in `DvDisplayManager` in order to improve image quality. Supersampling provides anti-aliasing on the image plane, cf.,

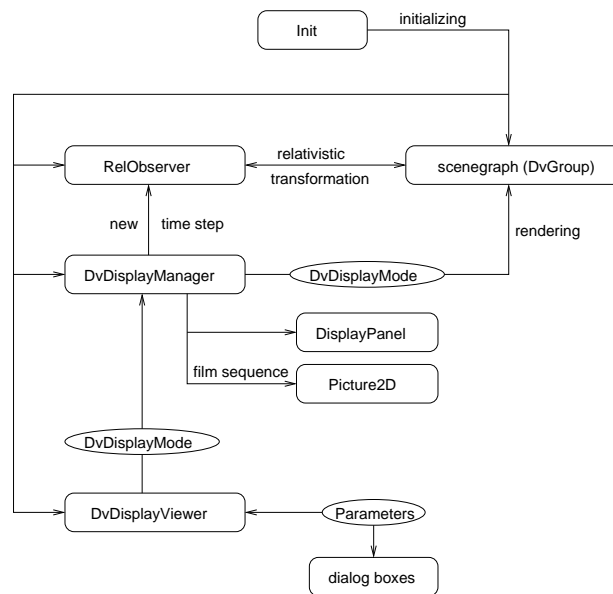


Figure C.1: Structure of *Virtual Relativity*.

for example, Watt[165]. The implementation uses the accumulation buffer of OpenGL, cf. Woo et al.[183]. Textures are prefiltered by MIPmapping, which is directly supported by OpenGL. Effects of a finite time of exposure can be simulated by temporal supersampling (motion blur). Here, intermediate images corresponding to different time steps which all lie inside the exposure time are generated. The intermediate images are combined by using the accumulation buffer. A depth-of-field effect is implemented by supersampling of the viewing frustum. Once again, the accumulation buffer is employed to blend the intermediate images.

`DisplayPanel` displays information about the current position, velocity, and acceleration at the bottom line of the screen. `Picture2D` provides an interface to two-dimensional image files, enabling us to write pictures and movies. It is based on `ImageMagick` and supports all common image file formats.

## C.2 Texture-Based Special Relativistic Rendering

The implementation of texture-based special relativistic rendering is based on the rendering and viewing modules of the previous section. It uses OpenGL, `QvLib`, `ImageMagick`, and `Glut`. `QvLib` allows to read three-dimensional scenes in the VRML 1.0 format. `ImageMagick` provides a transparent interface to standard two-dimensional image formats. The user front end and window handling is based on `Glut`.

Two versions of texture-based rendering are available. The first version supports the visualization of apparent geometry and the Doppler and searchlight effects. It requires OpenGL 1.1 with `pbuffer` (`SGIX_pbuffer`), pixel texture (`glPixelTexGenSGIX`), and color matrix (`SGI_co-`

`lor_matrix`) extensions. It runs on an SGI Octane with Maximum Impact graphics.

The second version provides the visualization of apparent geometry only. This restricted implementation makes use of standard OpenGL 1.1 only and runs on any graphics board with standard OpenGL 1.1.

In both versions, a constant number of six textures is used to cover the whole sphere. The six textures originate from the projection of the six sides of a cube onto the sphere. The texture coordinates are only recalculated if the velocity has changed.

The restricted version is located in the directory `Aberration`. Most of the specific functionality is isolated in the new display manager `AberrationDisplayManager`, which is derived from the standard display manager `DvDisplayManager`. The extended display manager initializes the generation of the non-relativistic images in the back buffer of the frame buffer. It transfers this data from the frame buffer to texture memory via `glCopyTexImage2D`. Texture objects (`glBindTexture`) allow fast access to the stored textures. Finally, the textured sphere elements are drawn in order to generate the relativistic image as seen by the user.

The aberration formula and the corresponding mapping of texture coordinates, Eq. 5.4, is implemented in the class `Aberration`. The sphere onto which the textures are mapped is represented by `Sphere`. `Sphere` allows to refine its triangulation by subdivision and render itself into the frame buffer.

The class `AberrationViewer` is derived from the standard implementation `DvViewer`. It provides usual viewer functionality, such as changes of viewing direction and steering of the relativistic vehicle.

The complete implementation of apparent geometry and the Doppler and searchlight effects is located in the directory `AberrationSearchlight`. Two further, derived classes

of the display manager are included. The class `AberrationDisplayManagerPbuffer` is derived from `AberrationDisplayManager`. It allows to render the intermediate, non-relativistic images to the pbuffer instead of the frame buffer. In this way, the resolution of the textures is independent of the resolution of the frame buffer.

The class `AberrationDisplayManagerIllum` is derived from `AberrationDisplayManagerPbuffer`. It enables the visualization of the Doppler and searchlight effects. It generates the pixel textures which represent the functions  $X_{i,j}(l_j, D)$  defined in Eq. (5.6). Three arbitrary basis functions  $L_j$  are supported. The rendering process is structured as in Fig. 5.13. Essentially, rendering of the Doppler factor map and applying the pixel textures is added to the rendering pipeline. The Doppler factor map and the pixel textures are represented by `Searchlight`.

### C.3 Extensions to RayViS

The implementation of relativistic ray tracing is based on *RayViS*[64]. *RayViS* was originally designed as a visualization tool based on linear three-dimensional ray tracing.

Due to the object-oriented framework of *RayViS* extension for relativistic rendering can be included in the existing system. Therefore, only the parts specific to the theory of relativity have to be added to the system; the rest of the rendering functionality can be adopted without changes, such as texture handling, image file I/O, scene graph management, etc.

#### C.3.1 Parallelization

In standard three-dimensional ray tracing, rendering time is significantly determined by the computation of intersections between rays and objects. Ray tracing has the burden of low rendering speed, since it is not supported by graphics hardware. The issue of rendering performance is even more problematic for general relativistic ray tracing, which is a couple of magnitudes slower than non-relativistic ray tracing. In general relativistic ray tracing, the generation of bent light rays by solving the geodesic equations plays a dominant role. Therefore, parallelization is an urgent need, especially for the production of high-resolution images, movies, and extensive parameter studies.

Fortunately, the computation of the null geodesics and the ray-object intersections for one pixel is independent of those for the other pixels. Hence, parallelization is performed on a per-pixel basis and utilizes domain decomposition on the image plane. The granularity can be as fine as a single line on the image plane in order to achieve good load balancing. The implementation makes use of MPI[104] and thus is platform-independent.

The standard main program for *RayViS* is located in the directory `main/CommandLine`. The implementation `clrvs.cxx` (command line *RayViS*) provides a command line-oriented interface to the ray tracer. It evaluates command line parameters, loads the scene description in the *RayViS* format (`.rvs`), and initializes the sample manager. Then, the whole image plane is sampled and the final image is saved to file. In this form, the program is processed sequentially.

The parallel version of *RayViS* is located in the directory `main/MpiCmd`. The main program `MpiCmd.cxx` replaces `clrvs.cxx`. Nodes (processors) are organized in two

classes: one master node and an arbitrary number of slave nodes. The master node controls the parallelization of the rendering process. It computes the domain decomposition, distributes rendering tasks to the slave nodes, and collects their results. Whenever a task is completed by a slave node, the master node assigns the next unrealized task to this slave node in order to avoid idling and bad load balancing. Finally, the master node combines the intermediate results to form the complete image and write it to file.

The computational domain is the image plane. It is decomposed by the master node in rectangular subimages of equal size. Each subimage has the width of a full line, i.e., the maximum number of tasks is limited by the number of lines. The number of tasks is provided by the user via the command line parameter `-tasks`. It determines the granularity and thus the quality of load balancing.

The maximum speedup  $s$  for a parallel program is limited by Amdahl's law,

$$s = \frac{T_s + T_p}{T_s + T_p/n},$$

where  $n$  is the number of processors,  $T_s$  is the time span in which the program is processed sequentially, and  $T_p$  is the time span in which the program can run in parallel. The speedup describes the ratio between the execution time of the single-processor program and the multi-processor program. This speedup is only achieved if the processors have enough jobs to execute and do not idle, i.e., if optimal load balancing is accomplished. Another goal is to attain a high degree of parallelization, i.e., to minimize the time  $T_s$ .

The current implementation of parallel *RayViS* provides almost optimal parallelization. The time for sequential execution,  $T_s$ , is limited to reading the parameters and the scene data in a preprocessing step and to combining and writing the final image in the postprocessing step. The time for both steps is of the order of one second in most cases. Typical values for general relativistic ray tracing,  $T_p = 10^4$  sec and  $T_s < 10$  sec, allow very good relative speedups of  $s/n > 0.8$  even on massive-parallel machines with several hundred processors. Load balancing is a more important issue, since processors idle at the end of the rendering process, when no further jobs are available but other processors have not finished their tasks yet. In these cases, the granularity has to be reduced by the user.

#### C.3.2 Movie Production

One important application of ray tracing is the production of movies. The following issues have to be taken into account.

First, a camera path has to be represented in order to provide motion through scenes. Secondly, movies have higher demand with respect to image quality. Spatial aliasing on the image plane causes flickering and thus is more disturbing in a movie than a fixed image. Moreover, temporal aliasing artifacts have to be taken into account. In this implementation, spatial aliasing is reduced by supersampling on the image plane. Temporal aliasing is avoided by temporal supersampling (motion blur), which assumes a finite exposure time.

All these measures further increase rendering time; the production of high-quality movies is associated with high computational costs. Therefore, an adequate implementation of multi-processor architectures, clusters, or network of workstations is required.

The standard main program in `main/CmdLine` is replaced by implementations for movie production. The sequential version is located in the directory `main/CmdLineSeq`, the parallelized version in the directory `main/MpiCmdSeq`. Both versions are structured as follows.

The abstract class `RvsMpiTaskManager` manages all rendering tasks, the camera parameters, and the scene graph. The tasks comprise rendering of intermediate image (for motion blur) or intermediate subimages (in the parallel version). The camera parameters are determined by the motion along the camera path. In addition, changing scene objects (animation) can be considered as well.

Classes derived from `RvsMpiTaskManager` actually implement the camera path and the handling of the scene graph. The following classes are provided. `RvsMpiTaskManagerFiles` contains a list of scene files (.rvs) describing camera parameters and scene objects. `RvsMpiTaskManagerRotation` implements a camera rotating on a circle around a fixed point. Here, the scene is static. `RvsMpiTaskManagerSrtLinear` provides a uniformly moving observer for special relativistic visualization. `RvsMpiTaskManager4DTimeLinear` implements a uniformly moving observer or object in four dimensions, for example for general relativistic visualization of dynamic spacetimes and scenes.

### C.3.3 Special Relativistic Rendering

The implementation of three-dimensional special relativistic ray tracing is located in the directory `Dev`. The class `RvsProjectorSRT` provides the ray projector for special relativity. The relativistic aberration of light is implemented by an extension of the ray projector. The direction and the starting point of a light ray are first computed in the reference frame of the camera and then transformed into the frame of the objects. The searchlight effect is taken into account in the pixel sampling function. This function yields the final RGB color for the respective pixels and thus allows the modification of brightness according to the searchlight effect.

The special relativistic ray projector can be included in the scene description according to the following syntax:

```
ProjectorSRT {
    background <real> <real> <real>
    beta <real>
    dir <real> <real> <real>
    pos <real> <real> <real>
    searchlighteffect <real> } .
```

The velocity of the observer is given by `beta`, the direction of motion by `dir`, and the current position by `pos`. The parameter `searchlighteffect` determines the exponent in the equation for the transformation of radiance, Eqs. (4.8) or (4.9). A value of 0 yields the visualization of apparent geometry only. A value of 4 allows the visualization of apparent geometry and searchlight effect, based on the transformation of wavelength-independent radiance, Eq. (4.9). This type of visualization models an observer detecting radiation of all wavelengths with equal efficiency. An example is shown in Fig. 5.10. A value of 5 for `searchlighteffect` reproduces the  $D^{-5}$  dependency in the transformation of wavelength-dependent radiance, Eq. (4.8).

### C.3.4 Four-Dimensional Scene Objects

Although the object-oriented framework of *RayViS* allows the extension for general relativistic ray tracing, most basic types of scene objects had to be implemented in order to have a practical system at hand. All primitive objects had to be made aware of temporal dependency.

In the directory `Obj/Poly`, objects describing various kinds of triangles have been included. The implemented classes are `Rvs4DFlatTriangle`, `Rvs4DPhongTextureTriangle`, `Rvs4DPhongTriangle`, `Rvs4DPlanarSurf`, and `Rvs4DTriangle`. The syntax in the scene description is analogous to the three-dimensional counterparts.

The following solid objects are included in the directory `Obj/SolObj`: `RvsSol4DBackground`, `RvsSol4DCylinder`, `RvsSol4DEllipsoid`, `RvsSol4DRigidDisc`, and `RvsSol4DRing`. Cylinder and ellipsoid are analogous to the three-dimensional examples. `RvsSol4DRing` does not have a three-dimensional analog. It is included for convenience, especially for the design of Saturn rings. The syntax is:

```
Sol4DRing {
    base <real> <real> <real>
    top <real> <real> <real>
    radii <real> <real>
    radii <real> <real> } .
```

The midpoint of the top and the bottom surface and the outer and inner radii have to be supplied. By specifying two different values for the radii, an ellipsoidal shape is achieved.

An object representing a rigidly rotating disk is specified by

```
Sol4DRigidDisc {
    center <real> <real> <real>
    radius <real>
    rmue <real> } .
```

The size and position of the disk is determined by `center` and `radius`. The relativistic parameter  $\mu$  fixes the rotational velocity, cf. Sect. 9.3.

Within general relativity, a scene object is not only determined by its spatial extent, but also by its motion through spacetime. Both the temporal and spatial “behavior” have to be known. The four-dimensional scene objects presented above do not allow direct description of their motion, since their syntax is based on the syntax of the three-dimensional static analogs.

Animation of objects is introduced indirectly. A new state variable is added to the functionality of the scene graph. This state variable holds the current description of motion, similarly to other state variables such as the current texture. A four-dimensional object is associated with the current motion. The intersection calculations, which are computed by a scene object, take into account this motion through spacetime.

The implementation is located in the directory `Obj/St` (`st=spacetime`). `RvsStMotion` is the abstract base class for all motions through spacetime. Only uniform motion is currently implemented. The syntax for the class `RvsStMotionConstVelocity` is:

```
SpacetimeMotionConstVelocity {
    t <real>
    velocity <real> <real> <real> } .
```

The parameter  $t$  determines the time when the origin of the object coordinates coincides with the origin of the global coordinate system. The second parameter is the velocity.

### C.3.5 Implemented Spacetimes

The implementation of the physics of spacetime is located in the directory `Physics`. The modules provide solvers for geodesic equations in the respective spacetimes, i.e., they return a polygonal approximation of the light ray for given initial parameters. The implemented modules comprise Schwarzschild metric, warp metric, and the metric of the rigidly rotating disk of dust. The numerical integration is based on Runge-Kutta method of fourth order with adaptive step size control.

In the directory `Ray`, the interface between the ray projectors and physics is implemented. The classes `RvsProjectorPolRay` and `RvsProjectorPolRay4D` represent ray projectors for bent light rays in three and four dimensions, i.e., for static spacetimes and dynamic spacetimes, respectively. They provide interfaces to the actual ray generators, which communicate with the geodesic solvers. The current implementation of the ray projectors does not support changes in color or brightness due to shifts in frequency or specific intensity of photons. Secondary rays and shadow rays are neglected as well.

Corresponding to the modules in `Physics`, the following ray generators are included in the directory `Ray`.

The Schwarzschild metric is represented by `RvsPolRayGenSchwarzschildMetric` and has the syntax:

```
PolRayGenSchwarzschild {
    maxpoints <real>
    mass <real>
    epsilon <real>
    corner <real> <real> <real>
    corner <real> <real> <real> } .
```

The maximum number of points for a single light ray, the mass of the gravitational source, the allowed integration error  $\epsilon$ , and the computational region are supplied.

`RvsPolRay4DGenRigidDisc` is the interface to the spacetime of the rigidly rotating disk of dust. Its syntax is:

```
PolRay4DGenRigidDisc {
    maxpoints <real>
    rmue <real>
    epsilon <real>
    rmax <real>
    [filename <pathname>
    [generate]] } .
```

The maximum number of points for a single light ray, the parameter  $\mu$ , the allowed integration error  $\epsilon$ , and the radius of the computational region are supplied. The geodesic solver in `Physics` accelerates the evaluation of the metric coefficients, which are needed in the form of Christoffel symbols in the geodesic equation. The metric coefficients are represented by Chebyshev polynomials, whose coefficients are calculated in a preprocessing step. In this way, the time consuming evaluation of integrals and ultraelliptic functions for the metric coefficients can be avoided in the rendering process. The Chebyshev coefficients can be saved to file and reloaded in later calculations, preventing unnecessary computations. If only the

parameter `filename` is supplied, the Chebyshev coefficients are loaded from file. If both `filename` and `generate` are supplied, the Chebyshev coefficients are generated and written to file.

`RvsPolRay4DGenWarpMetric` represents the warp metric. The syntax is:

```
PolRay4DGenWarpMetric {
    maxpoints <real>
    radius <real>
    sigma <real>
    epsilon <real>
    hmin <real>
    corner <real> <real> <real>
    corner <real> <real> <real>
    [accuracycheck]
    <spacetime_motion> } .
```

The maximum number of points for a single light ray, the radius of the warp bubble, and the parameter  $\sigma$  are supplied, cf. Sect. 9.4. The quality of integration is determined by the targeted maximum error, `epsilon`, and by the minimum step size, `hmin`. The computational box is described by its two corner points. The optional parameter `accuracycheck` enables a specific check of the accuracy of the numerical integration. Here, the light rays are reversed when they intersect the computational region. The distance of the reversed ray from the camera position is a measure for the inaccuracy of the numerics. The motion of the warp bubble is described by `<spacetime_motion>`.

### C.3.6 Standard Lensing

Standard lensing theory is based on (almost) static physical systems. Hence, a four-dimensional description is not required. Light rays are represented by a polygonal approximation (`RvsPolRay`) in three spatial dimensions. The lensing models are implemented as modules for the ray generator. They are located in the directory `Physics/GravitationalLens`. The physics of the implemented lensing models is described in Appendix B.

The class `RvsPolRayGenGravitationalLens` is the abstract base class for all lensing modules. It computes the light rays consisting of two straight parts. The first part reaches from the observer to the lens plane, the second part from the lens plane through the source plane to infinity. Derived classes implement the specific laws for the deflection angle, cf. Eq. (10.2).

The implementation of axially symmetric lenses is based on the derived class `RvsPolRayGenGravitationalAxialLens`. It reduces the general two-dimensional deflection law to the one-dimensional deflection law for symmetric lenses, Eq. (B.1). Derived classes actually implement the specific law for the deflection angle. The following axially symmetric lens models are available: Schwarzschild lens (`RvsPolRayGenGravitationalPointLens`), singular isothermal sphere (`RvsPolRayGenGravitationalSingularIsothermalSphere`), non-singular isothermal sphere (`RvsPolRayGenGravitationalNonSingularIsothermalSphere`), Hubble model (`RvsPolRayGenGravitationalHubbleLens`), King model (`RvsPolRayGenGravitationalKingLens`), and spiral lens model (`RvsPolRayGenGravitationalSpiralLens`).

Quadrupole lenses are based on the class `RvsPolRayGenGravitationalQuadrupoleLens`. This class splits the expression for the deflection angle in a symmetric part and a perturbed part, according to Eq. (B.2). The perturbed part is actually implemented in this class. The symmetric part is implemented in derived classes. The perturbed Schwarzschild lens (`RvsPolRayGenGravitationalChangRefsdalLens`) is an example of a quadrupole lens.

### C.3.7 Caustic Finder

The implementation of the three-dimensional *caustic finder* is located in the directory `Caustics`.

The central managing class is `CaRaySpace`, representing the *caustic finder*. It aggregates objects representing the region covered by the *caustic finder*, the computational grid, and the traversal function for the volume (`CaVolTrav`). So far, only box-shaped regions (`CaBoxReg`) and regular grids are supported. `CaVolRegularGrid` only counts the number of rays which touch each of its voxels. The derived class `CaVolDirRegularGrid` recognizes the distance of two rays on the image plane in order to avoid incorrect counting of image multiplicities. Volume traversal can be based on floating point arithmetic (`CaVolDDATrav`) or Bresenham algorithm (`CaVolBresTrav`).

The sample manager `CaSimpleSmplMgr` adds an object of the ray space to the intersection tests with the generated light rays. The ray space traverses the volume grid by means of the traversal object. Finally, the volumetric scalar field containing the image multiplicities is written to file.

The main program `CausticsMain` provides a command line-oriented interface to the ray tracer and the *caustic finder*. It evaluates command line parameters, loads the scene description in the *RayVis* format, and initializes the sample manager. Then, the whole image plane is sampled and the final volume data is saved to file.

## C.4 Isometric Embedding

The implementation of isometric embedding makes use of an object-oriented framework. The program is written in C++. It is based on mesh libraries by Gumhold[65] for handling connectivity in triangle meshes.

Basis data types represent a point (`Vertex`), an edge (`Edge`), and the whole mesh (`EdgeNet`), including connectivity and geometry. The mesh class is due to Gumhold.

The class `EdgeSphere` is derived from `EdgeNet` and represents the triangulation of a sphere. The sphere is the starting point for the Newton method which solves the embedding problem.

The class `EmEdgeNet` is derived from `EdgeSphere`. It implements the main functionality of the embedding algorithm. It fills the elements of the Jacobi matrix, Eq. (11.5), and iterates the Newton method, Eq. (11.4). The matrix is solved by singular value decomposition, as presented in Numerical Recipes[130].

The abstract class `Metric` represents the metric of the two-dimensional surface to be embedded. The classes `MetricEllipsoid` and `MetricSampled` implement the metric of an ellipsoid and a sampled metric, respectively. The sampled metric calculates intermediate values by linear interpolation.

Meshes resulting from intermediate steps of the Newton method or the final mesh can be written to a VRML 1.0 file by means of the output class `OutputVRML`.

## C.5 Adaptive Subdivision of Height Fields

The implementation of adaptively triangulated height fields is written in C++. It is based on a triangle mesh library by Gumhold[65]. This library provides a class `SubdivisionTriangleMesh`, which represents a triangle mesh. This mesh can be subdivided and hanging nodes can be eliminated.

The height field is represented by the abstract class `GraphSubdivisionMesh`. This class implements the subdivision scheme, the error criterion on which adaptive refinement is based, and additional color coding of the scalar values. The functional relationship between the two-dimensional domain and the values of the graph is defined in a derived class.

`SphericalHarmonicsMesh` is such a derived class. It represents a height field based on coefficients for spherical harmonics. Time-dependent data is supported. The coefficients are managed by the class `Field`. This class reads the coefficients from file and constructs data structures for fast access to the coefficients. Since the coefficients are given at discrete radii only, `Field` obtains the values at intermediate radii by linear interpolation. `SphericalHarmonics` implements the functional definition of spherical harmonics. `SphericalHarmonicsMesh` evaluates the values of the graph by looking up the corresponding coefficients in `Field` and by using the function definitions of the spherical harmonics.

# Appendix D

## Supplementary Video Material

Additional video material accompanies the thesis. The videos are available on tape and on CD-ROM. The films show further examples of relativistic visualization and reveal time-dependent properties. Table D.1 gives an overview of the contents of the video tape. The starting times are stated in order to conveniently locate a specific video.

### D.1 Detailed Description of the Videos

#### D.1.1 An Immersive Virtual Environment for Special Relativity

This video gives a good impression of the immersive virtual environment for special relativity, as presented in Sect. 6.4. The video was recorded during a typical simulation in a CAVE-like environment.

The first sequence shows a simulation with Saturn, see Fig. 6.4. 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. 6.5. 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 and tracking of velocity and position is presented. The last simulation takes place in a virtual gallery, cf. Fig. 6.6.

The rendered images are results of a perspective projection which depends on the position of the user's head. The position of the video camera is different from the user's position, which causes apparent cracks along the screen boundaries. The user,

however, perceives a correct three-dimensional world. The three-dimensional menu is at rest in the CUBE and is subject to the perspective projection onto the screens but not to relativistic transformations. The video was recorded with a shutter speed of only six Hertz. This causes some artifacts, especially when the rendered images are rapidly changing. This video was shown at the WSCG 2000 conference[171].

#### D.1.2 Speed Limit 1 079 252 849 km/h

This video shows special relativistic flights through various scenes. Only the apparent geometry is visualized, the searchlight and Doppler effects are neglected. The introductory sequence displays a traffic sign we pass by at high velocity. The sign reads a speed limit of 1 079 252 849 km/h—the speed of light. The next sequences contain high-speed flights through simplified models of the Brandenburg Gate and an NaCl lattice. Relativistic illustrations of the Eiffel Tower, the earth, Saturn, and a spaceship follow. This video was shown at the Computer Animation Festival of the GI Jahrestagung '98[168].

#### D.1.3 Image-Based Special Relativistic Rendering

This video presents examples of image-based special relativistic visualization, cf. Sect. 5.5. Here, only the relativistic effects on apparent geometry are taken into account.

The first part of the video shows a relativistic trip across the Deutzer bridge in Cologne. This movie was produced with the use of *Off-Terdingen*. The recordings of three cameras were stitched together to form the final movie. The film was shown on TV as part of a broadcast on Einstein and special relativity[132]. The first sequence shows the non-relativistic situation. The second sequence presents the relativistic case. The observer accelerates from non-relativistic speed to a maximum velocity of  $\beta = 0.9$ . The black stripes at the top and bottom sides are caused by missing data in the original recording. In the third sequence, the relativistic and the non-relativistic views are compared.

The second part of the video illustrates the interactive relativistic panorama viewer *Imagine*. It was recorded from the computer screen during a simulation on an SGI Onyx2 with InfiniteReality graphics board. First, the non-relativistic viewer is shown. Then, the velocity is increased and snapshots of a relativistic flight can be seen. Finally, the camera leaves its physically correct position and an outside view onto the virtual sphere surrounding the observer is adopted.

Table D.1: Contents of the video.

Time	Title
00:10	Introduction
01:26	An Immersive Virtual Environment for Special Relativity
06:04	Speed Limit 1 079 252 849 km/h
10:00	Image-Based Special Relativistic Rendering
12:37	Visualization of the Rigidly Rotating Disk of Dust
15:00	Visualization of the Warp Drive
16:15	Visualization of the Chang-Refsdal Lens
17:27	Neutron Star Oscillations

#### D.1.4 Visualization of the Rigidly Rotating Disk of Dust

In this video, the visualization of the rigidly rotating disk of dust is presented, cf. Sect. 9.3. The visual appearance of the disk as seen by a realistic observer is calculated by using general relativistic ray tracing of the photons emitted by the disk and registered by the observer.

The first part of the video shows the image of the disk of dust as seen by an observer. The observer is looking at the disk from an outside position which lies in almost flat spacetime while the relativistic parameter  $\mu$  is increased. For sufficiently relativistic disks, multiple and distorted images appear. Due to gravitational light bending both the top and the bottom faces are visible simultaneously. An artificial “pie slice” texture is applied both to the top and the bottom side of the disk in order to visualize rotational distortions. In this way, the effects of frame dragging become noticeable.

In the second part of the video, the position of the observer and the parameter  $\mu = 3$  are fixed. The observer is located on the axis of symmetry and is looking towards the edge of the disk. The sequence starts with a wide angle field of view. Parts of the top side of the disk are visible in the lower part of the picture. An image of the bottom side is found directly above this first image of the top side. Further above, alternating images of the top and the bottom faces follow. Further pictures document increasing zooming in on the original picture. This series reveals self-similarity and a fractal-like structure in parts of the ray-traced images.

This video was shown at Journées Relativistes '99[173].

#### D.1.5 Visualization of the Warp Drive

This video visualizes a spaceship equipped with a warp drive according to Alcubierre[2], cf. Sect. 9.4. The warp drive constructs a warp bubble which separates two flat parts of spacetime. The warp bubble is able to move faster than the speed of light as measured in an outside flat region of spacetime. A spaceship which is at rest inside the warp bubble would then travel faster than the speed of light.

The visual appearance of the spaceship and the background objects is calculated by using general relativistic ray tracing. In this video, the spaceship travels in front of the earth and moon, Mars, Jupiter, and Saturn, at velocities of  $0.8c$ ,  $1.5c$ , or  $2.5c$ . The light deflection at the warp bubble causes astonishing visual distortions on these background objects.

The visualizations of this video were also shown at the exhibition “Seven Hills”[13].

#### D.1.6 Visualization of the Chang-Refsdal Lens

This video presents the visualization of gravitational lensing, cf. Chapter 10. The deflecting body is a Chang-Refsdal lens. The Chang-Refsdal lens is a Schwarzschild lens with a quadrupole distortion, cf. Sect. B.2.

The first sequence shows the image of a moving source under the influence of the gravitational lens. The second sequence shows the same situation in the lens plane. Both the source and the caustic structure are displayed. In the third sequence, the image and the source plane are visible simultaneously. The last sequence shows the two-dimensional caustic structure. The caustic was numerically generated by means of

the *caustic finder*. Rendering is based on shear-warp factorization.

#### D.1.7 Neutron Star Oscillations

This video visualizes a particle which travels around a neutron star on a hyperbolic path. The particle excites the star to oscillations by gravitational interaction and to coupled emission of gravitational waves. The underlying computation is based on a multipole expansion of the particle[139]. Each multipole generates its own oscillation. Summing over all multipoles yields the overall oscillation pattern. The perturbation of the metric can be decomposed into two terms. The first term is a conformal factor. The second term describes the deviation from the conformally flat metric.

The data is visualized by using both a height field and color coding, as presented in Chapter 12. Only the difference of the true metric and the underlying metric of the neutron star is visualized, i.e., only the perturbation of the spacetime induced by the particle. The quality of the triangle mesh representing the height field is increased by adaptive refinement, as opposed to regular sampling.

The first part of the film shows the time evolution of the conformal factor (metric component “ $t$ ”) in the  $x$ - $y$  plane. The first sequence specifically visualizes the triangulation by marking the triangle edges with a wireframe. The second sequence shows the surfaces by smooth shading. In both sequences, the particle travels in the  $x$ - $y$  plane. In the beginning, the gravitational field of the particle is visible. The particle is smeared out because only ten orders of the multipole expansion are taken into account. The neutron star itself is located at the midpoint of the plane. When the particle is approaching the neutron star, emission of gravitational waves is induced. These waves are superimposed onto the gravitational field of the particle. The neutron star keeps oscillating and emitting characteristic gravitational waves, mainly consisting of  $f$ -mode and some  $p$ -modes, after the particle left the neutron star.

The second part of the film shows the time evolution of the second term (metric component “ $s$ ”). Again, both wireframe rendering and smooth shading are included. The spiral emission of gravitational waves, while the particle is orbiting the neutron star, and the subsequent oscillations of the neutron star are most prominent.



# Appendix E

## Conventions and Notation

Throughout this thesis, units are used in which the speed of light  $c = 1$  and the gravitational constant  $G = 1$ . In some cases,  $c$  and  $G$  are explicitly included for clarity.

The Einsteinian sum convention applies to indices contained twice in an expression. In the context of general relativity, lower case Greek letters cover an index range of  $0 \dots 3$ , whereas lower case Latin letters cover an index range of  $1 \dots 3$ . Partial derivatives might be marked by commas, for example,  $f_{,\mu}$ . Covariant derivatives are represented by semicolons, for example,  $f_{;\mu}$ .

The indices  $\nu$ ,  $E$ , and  $\lambda$  in frequency-dependent radiance  $L_\nu$ , energy-dependent radiance  $L_E$ , and wavelength-dependent radiance  $L_\lambda$  just indicate that radiance depends on the respective quantity. The respective parameters have to be provided separately, for example,  $L_\lambda(\lambda')$  is the wavelength-dependent radiance for the wavelength  $\lambda'$ .

References to program code, classes, variables, file and directory names, and syntax descriptions are set in `typeewriter` font.

### E.1 Variable Names

The following variable and function names are connotated with specific meaning.

Variable	Meaning
$A$	area
$\vec{a}$	acceleration
$a^\mu$	four-acceleration
$b$	chromaticity for blue
$\vec{b}$	color-matching function for blue
$B$	tristimulus-value for blue, radiosity
$\vec{B}$	blue primary
$\beta$	velocity, unlensed position of the source
$c$	speed of light
$c_i$	chromaticity for a color
$\vec{C}$	color
$C_i$	tristimulus value for a color
$D$	Doppler factor
$D_d$	distance of the deflector (gravitational lens)
$D_s$	distance of the source in gravitational lensing

Variable	Meaning
$D_{ds}$	distance between gravitational lens and source
$\vec{f}_i$	color-matching function
$F_{\mu\nu}$	electromagnetic field tensor
$g$	chromaticity for green
$\vec{g}$	color-matching function for green
$G$	gravitational constant, tristimulus-value for green
$\vec{G}$	green primary
$E$	energy
$E^i$	irradiance
$E_\lambda^i$	wavelength-dependent irradiance
$\eta_{\mu\nu}$	Minkowski metric
$\gamma$	velocity parameter
$\Gamma^{\mu}_{\nu\rho}$	Christoffel symbol
$h$	Planck constant
$I$	radiant intensity
$I_\nu$	frequency-dependent radiant intensity, specific intensity
$I_\lambda$	wavelength-dependent radiant intensity
$k$	Boltzmann constant
$\vec{k}$	wave vector
$k^\mu$	four-wave vector
$L$	radiance
$L_\nu$	frequency-dependent radiance
$L_\lambda$	wavelength-dependent radiance
$L_E$	energy-dependent radiance
$\lambda$	wavelength
$\Lambda^\mu_\nu$	matrix for Lorentz transformation
$M$	radiant excitation
$\mu$	magnification factor, relativistic parameter for rigidly rotating disk of dust
$N$	number of particles
$\bar{N}$	number density in phase space
$\nu$	frequency
$\omega$	circular frequency
$\Omega$	solid angle

*to be continued...*

<i>...continued</i>	
Variable	Meaning
$\vec{P}_i$	primary color
$\phi$	spherical coordinate
$\Phi$	radiant power or radiant flux
$Q$	radiant energy
$r$	radial distance,
	chromaticity for red
$\bar{r}$	color-matching function for red
$R$	tristimulus-value for red
$R_S$	Schwarzschild radius
$\vec{R}$	red primary
$S$	phase
$\vec{S}$	Poynting vector
$t$	time
$\theta$	spherical coordinate,
	position of the image in gravitational
	lensing
$\tau$	proper time
$v$	velocity
$u^\mu$	four-velocity
$V$	volume in phase space
$x$	chromaticity for X in the CIE system
$\bar{x}$	color-matching function for X in the
	CIE system
$X$	tristimulus-value for X in the CIE
	system
$\vec{X}$	X primary in the CIE system
$\xi$	impact parameter in gravitational
	lensing
$y$	chromaticity for Y in the CIE system
$\bar{y}$	color-matching function for Y in the
	CIE system
$Y$	tristimulus-value for Y in the CIE
	system
$\vec{Y}$	Y primary in the CIE system
$z$	chromaticity for Z in the CIE system
$\bar{z}$	color-matching function for Z in the
	CIE system
$Z$	tristimulus-value for Z in the CIE
	system
$\vec{Z}$	Z primary in the CIE system

# Bibliography

- [1] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. In M. Landy and J. A. Movshon, editors, *Computational Models of Visual Processing*, pages 3–20, Cambridge, 1991. MIT Press.
- [2] M. Alcubierre. The warp drive: hyper-fast travel within general relativity. *Classical and Quantum Gravity*, 11:L73–L77, 1994.
- [3] K. T. Alligood, T. D. Sauer, and J. A. Yorke. *Chaos: An Introduction to Dynamical Systems*. Springer, New York, 1996.
- [4] M. Ansorg. Timelike geodesic motions within the general relativistic gravitational field of the rigidly rotating disk of dust. *Journal of Mathematical Physics*, 39(11):5984–6000, 1998.
- [5] A. Appel. Some techniques for shading machine renderings of solids. In *AFIPS 1968 Spring Joint Computer Conf.*, volume 32, pages 37–45, 1968.
- [6] A. Ashtekar, A. Corichi, and M. Pierri. Geometry in color perception. In B. R. Iyer, editor, *Black Holes, Gravitational Radiation and the Universe*, pages 535–550. Kluwer, Dordrecht, 1999.
- [7] P. Atherton, K. Weiler, and D. Greenberg. Polygon shadow generation. In *SIGGRAPH 78 Conference Proceedings*, pages 275–281, Aug. 1978.
- [8] *AVS Web Site*. <http://www.avs.com>, 2000.
- [9] R. E. Bank. *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations. Users' Guide 6.0*. SIAM, Philadelphia, 1990.
- [10] R. E. Bank, A. H. Sherman, and A. Weiser. Refinement algorithms and data structures for regular local mesh refinement. In R. Stepleman, M. Carver, R. Piskin, W. F. Ames, and R. Vichnevetsky, editors, *Scientific Computing, Volume 1 (IMACS Transactions on Scientific Computation)*, pages 3–17. North-Holland, New York, 1983.
- [11] J. M. Bardeen and R. V. Wagoner. Relativistic disks. I. Uniform rotation. *Astrophysical Journal*, 167:359–423, 1971.
- [12] G. Bell, A. Parisi, and M. Pesce. *VRML: Version 1.0 specification*. <http://webSPACE.sgi.com/Archive/Spec1.0/index.html>, 1995.
- [13] Berliner Festspiele GmbH. *Seven Hills. Images and Signs of the 21st Century*. Exhibition, 14th May to 29th October 2000, Berlin. <http://www.berlinerfestspiele.de/berlinzweitausend/seven.html>.
- [14] C. Betts. Fast rendering of relativistic objects. *The Journal of Visualization and Computer Animation*, 9(1):17–31, 1998.
- [15] J. Binney and S. Tremaine. *Galactic Dynamics*. Princeton University Press, Princeton, 1987.
- [16] J. F. Blinn. Models of light reflection for computer synthesized pictures. In *SIGGRAPH 77 Conference Proceedings*, pages 192–198, July 1977.
- [17] M. L. Boas. Apparent shape of large objects at relativistic speeds. *American Journal of Physics*, 29(5):283–286, May 1961.
- [18] W. J. Bouknight and K. C. Kelly. An algorithm for producing half-tone computer graphics presentations with shadows and movable light sources. *Proc. AFIPS JSCC*, 36:1–10, 1970.
- [19] R. Brutscher. In *GEO Wissen: Sinne und Wahrnehmung*, pages 114–119. Sept. 1997. In German.
- [20] R. Brutscher. Einsteins Temporaus. *Star Observer*, pages 12–25, Nov. 1998. In German.
- [21] S. Bryson. Virtual spacetime: An environment for the visualization of curved spacetimes via geodesic flows. In *Proceedings of the IEEE Visualization '92 Conference*, pages 291–298, 1992.
- [22] J. Bublath. *Geheimnisse unseres Universum*. Droemer, München, 1999. In German.
- [23] 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.
- [24] S. E. Chen. QuickTime VR – An image-based approach to virtual environment navigation. In *SIGGRAPH 95 Conference Proceedings*, pages 29–38, Aug. 1995.
- [25] S. E. Chen and L. Williams. View interpolation for image synthesis. In *SIGGRAPH 93 Conference Proceedings*, pages 279–288, Aug. 1993.
- [26] C. Clark, W. A. Hiscock, and S. L. Larson. Null geodesics in the Alcubierre warp-drive: The view from the bridge. *Classical and Quantum Gravity*, 16:3965–3972, 1999.

- [27] M. F. Cohen and J. R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, Boston, 1993.
- [28] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1(1):7–24, Jan. 1982.
- [29] N. J. Cornish and G. Gibbons. The tale of two centres. *Classical and Quantum Gravity*, 14:1865–1881, 1997.
- [30] N. J. Cornish and J. J. Levin. The mixmaster universe: A chaotic fary tale. *Physical Review D*, 55:7489–7510, 1997.
- [31] J. Cristy. *Image Magick*. <http://www.wizards.dupont.com/cristy>, 2000.
- [32] F. C. Crow. Shadow algorithms for computer graphics. In *SIGGRAPH 77 Conference Proceedings*, pages 242–248, July 1977.
- [33] F. C. Crow. Summed-area tables for texture mapping. In *SIGGRAPH 84 Conference Proceedings*, pages 207–212, July 1984.
- [34] 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.
- [35] R. A. d’Inverno. *Introducing Einstein’s Relativity*. Clarendon Press, Oxford, 1992.
- [36] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry and image-based approach. In *SIGGRAPH 96 Conference Proceedings*, pages 11–20, Aug. 1996.
- [37] B. I. des Poids et Mesures. *Le Système International d’Unités (SI), The International System of Units (SI)*. Organisation Intergouvernementale de la Convention du Mètre, Sèvres, France, seventh edition, 1998.
- [38] DESY. *DESY Web Site*. <http://www.desy.de>.
- [39] C. P. Dettmann, N. E. Frankel, and N. J. Cornish. Fractal basins and chaotic trajectories in multi-black hole space-times. *Physical Review D*, 50:618–621, 1994.
- [40] Deutsche Forschungsgemeinschaft (DFG). *Kalender 2000 (calendar 2000)*, Bonn, 2000.
- [41] R. Earnshaw, M. Gigante, and H. Jones, editors. *Virtual Reality Systems*. Academic Press, London, 1993.
- [42] G. Eckel and S. Hiatt. *IRIS Performer Programmer’s Guide*. Silicon Graphics Document Number 007-1680-040, 1998.
- [43] J. Ehlers. General relativity and kinetic theory. In R. K. Sachs, editor, *General Relativity and Cosmology*. Academic Press, 1971.
- [44] A. Einstein. Zur Elektrodynamik bewegter Körper. *Annalen der Physik*, 17:891–921, 1905. In German.
- [45] A. Einstein and N. Rosen. The particle problem in the general theory of relativity. *Physical Review*, 48:73–77, 1935.
- [46] J. Encarnação, W. Straßer, and R. Klein. *Graphische Datenverarbeitung*, volume 1. Oldenbourg, München, fourth edition, 1996. In German.
- [47] F. J. Ernst. New formulation of the axially symmetric gravitational field problem. *Physical Review*, 167:1175–1178, 1968.
- [48] T. Ertl, F. Geyer, H. Herold, U. Kraus, R. Niemeier, H.-P. Nollert, A. Rebetzky, H. Ruder, and G. Zeller. Visualization in astrophysics. In *Eurographics ’89 Proceedings*, pages 149–158, 1989.
- [49] O. Faugeras. *Three-Dimensional Computer Vision. A Geometric Viewpoint*. MIT Press, Cambridge, MA, 1993.
- [50] L. Flamm. Beitrag zur Einsteinschen Gravitationstheorie. *Physikalische Zeitschrift*, 17:448–454, 1916. In German.
- [51] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, Massachusetts, 1990.
- [52] L. H. Ford and T. A. Roman. Negative energy, wormholes and warp drive. *Scientific American*, pages 30–37, Jan. 2000.
- [53] B. Fort and Y. Mellier. Arc(let)s in clusters of galaxies. *The Astronomy and Astrophysics Review*, 5:239–292, 1994.
- [54] F. Frutos-Alfaro. *Die interaktive Visualisierung von Gravitationslinsen*. Dissertation, University of Tübingen, Germany, 1998. In German.
- [55] G. Gamow. *Mr Tompkins in Wonderland*. University Press, Cambridge, 1939.
- [56] G. Gamow. *Mr Tompkins in Paperback*. University Press, Cambridge, 1993.
- [57] W. Gekelman, J. Maggs, and L. Xu. Real-time relativity. *Computers in Physics*, 5(4):372–385, July/Aug. 1991.
- [58] A. S. Glassner. How to derive a spectrum from an RGB triplet. *IEEE Computer Graphics and Applications*, 9(4):95–99, July 1989.
- [59] A. S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann, San Francisco, 1995.
- [60] *GLUT: The OpenGL Utility Toolkit*. <http://reality.sgi.com/mjk/glut3/glut3.html>, 2000.
- [61] C. M. Goral, K. K. Torrance, D. P. Greenberg, and B. Battaile. Modelling the interaction of light between diffuse surfaces. *Computer Graphics*, 18(3):213–222, July 1984.

- [62] S. J. Gortler, R. Grzeszczuk, and R. S. M. F. Cohen. The lumigraph. In *SIGGRAPH 96 Conference Proceedings*, pages 43–54, Aug. 1996.
- [63] E. Gröller. Nonlinear ray tracing: Visualizing strange worlds. *The Visual Computer*, 11(5):263–276, 1995.
- [64] A. Gröne. *Entwurf eines objektorientierten Visualisierungssystems auf der Basis von Raytracing*. Dissertation, University of Tübingen, Germany, 1996. In German.
- [65] S. Gumhold. C++ mesh class library. Private communications, S. Gumhold is with WSI/GRIS (Graphical-Interactive Systems) at the University of Tübingen, Germany, 2000.
- [66] R. Hall. *Illumination and Color in Computer Generated Imagery*. Springer, New York, 1988.
- [67] R. Hall and D. P. Greenberg. A testbed for realistic image synthesis. *IEEE Computer Graphics and Applications*, 3(8):10–20, Nov. 1983.
- [68] C. Hand. Survey of 3D interaction techniques. *Computer Graphics Forum*, 16(5):269–281, Dec. 1997.
- [69] R. M. Haralick and L. D. Shapiro. *Computer and Robot Vision*, volume 1 & 2. Addison-Wesley, Reading, MA, 1992 & 1993.
- [70] H. Helmholtz. *Handbuch der physiologischen Optik*. Voss, Hamburg, second edition, 1896. In German.
- [71] *Virtual Environments Lab*. In High-Performance Computing-Center Stuttgart Web Site: <http://www.hlrs.de/structure/organisation/vis/velab>.
- [72] 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.
- [73] G. Hinshaw and L. Krauss. Gravitational lensing by isothermal spheres with finite core radii. *Astrophysical Journal*, 320:468–476, 1987.
- [74] 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.
- [75] B. K. P. Horn. *Robot Vision*. MIT Press, Cambridge, MA, 1986.
- [76] P.-K. Hsiung and R. H. P. Dunn. Visualizing relativistic effects in spacetime. In *Proceedings of Supercomputing '89 Conference*, pages 597–606, 1989.
- [77] 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.
- [78] 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 IEEE Visualization '90 Conference*, pages 83–92, Oct. 1990.
- [79] P.-K. Hsiung, R. H. Thibadeau, and M. Wu. T-buffer: Fast visualization of relativistic effects in spacetime. *Computer Graphics*, 24(2):83–88, Mar. 1990.
- [80] J. Huchra. Galactic structure and evolution. In R. A. Meyers, editor, *Encyclopedia of Astronomy and Astrophysics*, pages 203–219. Academic Press, San Diego, 1989.
- [81] T. Hüttner and W. Straßer. Fast footprint MIPmapping. In *Proceedings of the Eurographics/SIGGRAPH Workshop on Graphics Hardware 1999*, pages 35–44, 1999.
- [82] 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.
- [83] D. B. Judd and G. Wyszecki. *Color in Business, Science, and Industry*. John Wiley & Sons, New York, third edition, 1975.
- [84] W. C. Keel. *Quasars and Active Galactic Nuclei*. <http://www.astr.ua.edu/keel/agn>, 2000.
- [85] D. Kobras. Bildbasierte speziellrelativistische Visualisierung. Diplomarbeit, Physics Department, University of Tübingen, Germany, 2000. In German.
- [86] R. Kornhuber and R. Roitzsch. On adaptive grid refinement in the presence of internal or boundary layers. Report SC 89–5, Konrad-Zuse-Institute, Berlin, Germany, Nov. 1989.
- [87] D. Kramer and G. Neugebauer. Algebraisch spezielle Einstein-Räume mit einer Bewegungsgruppe. *Communications in Mathematical Physics*, 7:173–180, 1968. In German.
- [88] U. Kraus. Light deflection near neutron stars. In H. Riffert, H. Ruder, H.-P. Nollert, and F. W. Hehl, editors, *Relativistic Astrophysics*, pages 66–81. Vieweg, 1998.
- [89] U. Kraus. Brightness and colour of rapidly moving objects: The visual appearance of a large sphere revisited. *American Journal of Physics*, 68(1):56–60, 2000.
- [90] P. Lacroute and M. Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *SIGGRAPH 94 Conference Proceedings*, pages 451–458, July 1994.
- [91] A. Lampa. Wie erscheint nach der Relativitätstheorie ein bewegter Stab einem ruhenden Beobachter? *Zeitschrift für Physik*, 27:138–148, 1924. In German.
- [92] B. Lehle. *Visualisierung in der Relativitätstheorie*. Dissertation, University of Tübingen, Germany, 1997.

- [93] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH 96 Conference Proceedings*, pages 31–42, Aug. 1996.
- [94] *Live Picture*. Web site: <http://www.livepicture.com>.
- [95] J. D. Mackinlay, S. K. Card, and G. G. Robertson. Rapid controlled movement through a virtual 3D workspace. In *SIGGRAPH 90 Conference Proceedings*, pages 171–176, Aug. 1990.
- [96] H. A. Mallot. *Sehen und die Verarbeitung visueller Information*. Vieweg, Braunschweig, Wiesbaden, 1998. In German.
- [97] M. Marelli. Turisti alla velocità della luce. *Newton (Italy)*, pages 62–70, Sept. 2000. In Italian.
- [98] D. Marr. *Vision*. Freeman, San Francisco, 1982.
- [99] J. M. McKinley. Relativistic transformations of light power. *American Journal of Physics*, 47(7):602–605, July 1979.
- [100] J. M. McKinley. Relativistic transformation of solid angle. *American Journal of Physics*, 48(8):612–614, Aug. 1980.
- [101] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. In *SIGGRAPH 95 Conference Proceedings*, pages 39–46, Aug. 1995.
- [102] J. Mellinger. Visualisierung statischer Wurmlöcher. Zulassungsarbeit, Physics Department, University of Tübingen, Germany, 1999. In German.
- [103] *The Mesa 3D Graphics Library*. <http://www.mesa3d.org>, 2000.
- [104] *Message Passing Interface Forum*. <http://www.mpi-forum.org>.
- [105] A. I. Miller. *Albert Einstein's Special Theory of Relativity*. Addison-Wesley, Reading, MA, 1981.
- [106] M. R. Mine. Virtual environment interaction techniques. Technical Report TR95-018, Department of Computer Science, University of North Carolina at Chapel Hill, May 1995.
- [107] C. W. Misner, K. S. Thorne, and J. A. Wheeler. *Gravitation*. Freeman, New York, 1973.
- [108] C. Møller. *The Theory of Relativity*. Clarendon Press, Oxford, second edition, 1972.
- [109] J. S. Montrym, D. R. Baum, D. L. Dignam, and C. J. Migdal. InfiniteReality: A real-time graphics system. In *SIGGRAPH 97 Conference Proceedings*, pages 293–302, Aug. 1997.
- [110] M. S. Morris and K. S. Thorne. Wormholes in spacetime and their use for interstellar travel: A tool for teaching general relativity. *American Journal of Physics*, 56(5):395–412, 1988.
- [111] M. S. Morris, K. S. Thorne, and U. Yurtsever. Wormholes, time machines, and the weak energy condition. *Physical Review Letters*, 61:1446–1449, 1988.
- [112] B. Müller. Irrte Einstein? *Bild der Wissenschaft*, pages 42–46, Mar. 1998. In German.
- [113] D. Münter. Süddeutsche Zeitung, 5th March 1998. In German.
- [114] R. J. Nemiroff. Visual distortions near a neutron star and black hole. *American Journal of Physics*, 61(7):619–632, July 1993.
- [115] G. Neugebauer and R. Meinel. General relativistic gravitational field of a rigidly rotating disk of dust: Solution in terms of ultraelliptic functions. *Physical Review Letters*, 75:3046, 1995.
- [116] *Newton (Japan)*, pages 98–105, Dec. 1998. In Japanese.
- [117] G. M. Nielson, H. Hagen, and H. Müller. *Scientific Visualization: Overviews, Methodologies, and Techniques*. IEEE Computer Society Press, Silver Spring, MD, 1997.
- [118] T. Nishita and E. Nakamae. Continuous tone representation of 3-D objects taking account of shadows and interreflection. In *SIGGRAPH 85 Conference Proceedings*, pages 23–30, July 1985.
- [119] H.-P. Nollert and H. Herold. Visualization in curved spacetimes. II. Visualization of surfaces via embedding. In F. W. Hehl, R. A. Puntigam, and H. Ruder, editors, *Relativity and Scientific Computing*, pages 330–351. Springer, 1996.
- [120] H.-P. Nollert, U. Kraus, and H. Ruder. Visualization in curved spacetimes. I. Visualization of objects via four-dimensional ray-tracing. In F. W. Hehl, R. A. Puntigam, and H. Ruder, editors, *Relativity and Scientific Computing*, pages 314–329. Springer, 1996.
- [121] H.-P. Nollert, H. Ruder, H. Herold, and U. Kraus. The relativistic “looks” of a neutron star. *Astronomy and Astrophysics*, 208:153, 1989.
- [122] E. Ott. *Chaos in Dynamical Systems*. Cambridge University Press, Cambridge, 1993.
- [123] B. Paul and B. Bederson. *Togl - a Tk OpenGL widget*. <http://Togl.sourceforge.net>, 2000.
- [124] P. J. E. Peebles and D. T. Wilkinson. Comment on the anisotropy of the primeval fireball. *Physical Review*, 174(5):2168, Oct. 1968.
- [125] M. S. Peercy. Linear color representations for full spectral sampling. In *SIGGRAPH 93 Conference Proceedings*, pages 191–198, Aug. 1993.
- [126] R. Penrose. The apparent shape of a relativistically moving sphere. *Proceedings of the Cambridge Philosophical Society*, 55:137–139, 1959.

- [127] M. J. Pfenning and L. H. Ford. The unphysical nature of “warp drive”. *Classical and Quantum Gravity*, 14(7):1743–1751, 1997.
- [128] B.-T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, June 1975.
- [129] A. Pompos and K. Riesselmann. Santa’s world revisited. *Ferminews (Fermilab, Illinois)*, 23(9):10–11, May 2000.
- [130] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, second edition, 1994.
- [131] *QGLViewer*. Web site: <http://www.qglviewer.de>.
- [132] Quarks & Co. Die Relativitätstheorie—Einfach erklärt. Broadcast on 9th November 1999 by German TV station WDR.
- [133] D. Rantzau and U. Lang. A scalable virtual environment for large scale scientific data analysis. *Future Generation Computer Systems*, 14:215–222, 1998.
- [134] 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 Verlag, Heidelberg, 1998.
- [135] W. Rindler. *Introduction to Special Relativity*. Clarendon Press, Oxford, second edition, 1991.
- [136] 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.
- [137] 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, July 1994.
- [138] H. Ruder, T. Ertl, K. Gruber, F. Mosbach, S. Subke, and K. Widmayer. Kinematics of the special theory of relativity. In *Eurographics ’91, Tutorial No. 12*, Vienna, 1991.
- [139] J. Ruoff. *The Numerical Evolution of Neutron Star Oscillations*. Dissertation, University of Tübingen, Germany, 2000.
- [140] T. Schmidt. Einbettungen von Metriken mit sphärischer Topologie in den euklidischen  $R^3$  im Rahmen der Allgemeinen Relativitätstheorie. Diplomarbeit, Physics Department, University of Tübingen, Germany, 1995. In German.
- [141] P. Schneider, J. Ehlers, and E. E. Falco. *Gravitational Lenses*. Springer, Berlin, 1992.
- [142] E. Schrödinger. Grundlinien einer Theorie der Farbenmetrik im Tagessehen. *Annalen der Physik*, 62:603, 1920.
- [143] G. D. Scott and H. J. van Driel. Geometrical appearances at relativistic speeds. *American Journal of Physics*, 38(8):971–977, Aug. 1970.
- [144] G. D. Scott and R. R. Viner. The geometrical appearance of large objects moving at relativistic speeds. *American Journal of Physics*, 33(7):534–536, July 1965.
- [145] S. M. Seitz and C. R. Dyer. View morphing. In *SIGGRAPH 96 Conference Proceedings*, pages 21–30, Aug. 1996.
- [146] H.-Y. Shum and L.-W. He. Rendering with concentric mosaics. In *SIGGRAPH 93 Conference Proceedings*, pages 299–306, Aug. 1999.
- [147] L. Silberstein. The complete three-dimensional colour domain and its metrics. *Philosophical Magazine (VII)*, 37:126, 1946.
- [148] R. Sonntag. *RadioLab: An object-oriented global illumination system*. Dissertation, University of Tübingen, Germany. Forthcoming.
- [149] L. Spillmann and J. S. Werner, editors. *Visual Perception. The Neurophysiological Foundations*. Academic Press, San Diego, 1990.
- [150] M. D. Spivak. *A Comprehensive Introduction to Differential Geometry*, volume 4. Publish or Perish, Berkeley, 1979.
- [151] W. S. Stiles. A modified Helmholtz line element in brightness-colour space. *Proceedings of the Physical Society of London*, 58:41, 1946.
- [152] 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.
- [153] A. Stockman, D. I. A. MacLeod, and N. E. Johnson. Spectral sensitivities of the human cones. *Journal of the Optical Society of America A*, 10(12):2491–2521, Dec. 1993.
- [154] M. C. Stone, W. B. Cowan, and J. C. Beatty. Color gamut mapping and the printing of digital color images. *ACM Transactions on Graphics*, 7(4):249–292, Oct. 1988.
- [155] P. S. Strauss, G. Bell, and M. Pesce. *QvLib: The VRML Programming Library Version 1.0*. <http://www.vrml.org/www-vrml/vrml.tech/qv.html>, 1995.
- [156] Y. Sun, F. D. Fracchia, T. W. Calvert, and M. S. Drew. Deriving spectra from colors and rendering light interference. *IEEE Computer Graphics and Applications*, 19(4):61–67, July/Aug. 1999.
- [157] R. Szeliski and H.-Y. Shum. Creating full view panoramic mosaics and environment maps. In *SIGGRAPH 97 Conference Proceedings*, pages 251–258, Aug. 1997.

- [158] *Tcl/Tk*. <http://www.scriptics.com>, 2000.
- [159] J. Terrell. Invisibility of the Lorentz contraction. *Physical Review*, 116(4):1041–1045, Nov. 1959.
- [160] H. Teufel. *Psychophysical Investigation of Chromatic Induction*. Dissertation, University of Tübingen, Germany, 2000.
- [161] R. Turner, F. Balaguer, E. Gobetti, 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.
- [162] R. Vass. Der Mord am eigenen Ahnen. *Bild der Wissenschaft*, pages 72–75, July 1998. In German.
- [163] D. Walsh, R. F. Carswell, and R. J. Weymann. 0957 + 561 A, B: Twin quasistellar objects or gravitational lens? *Nature*, 279:381–384, 1979.
- [164] 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.
- [165] A. Watt. *3D Computer Graphics*. Addison-Wesley, Reading, second edition, 1993.
- [166] S. Weinberg. *Gravitation and Cosmology: Principles and Applications of the General Theory of Relativity*. John Wiley & Sons, New York, 1972.
- [167] D. Weiskopf. Spezielle Relativitätstheorie: Visualisierung und Virtuelle Realität. Diplomarbeit, Physics Department, University of Tübingen, Germany, 1997. In German.
- [168] D. Weiskopf. Speed limit 1 079 252 849 km/h. Video shown at the Computer Animation Festival '98, GI Jahrestagung, Magdeburg, 1998.
- [169] D. Weiskopf. A texture mapping approach for the visualization of special relativity. In *IEEE Visualization '99 Late Breaking Hot Topics Proceedings*, pages 41–44, 1999.
- [170] D. Weiskopf. Fast visualization of special relativistic effects on geometry and illumination. In *Proceedings of EG/IEEE TCVG Symposium on Visualization 2000*, pages 219–228, 2000.
- [171] D. Weiskopf. An immersive virtual environment for special relativity. In *Proceedings WSCG 2000 Conference*, pages 337–344, 2000.
- [172] D. Weiskopf. Non-linear ray tracing as a visualization tool for gravitational physics. In *Proceedings of the IEEE Visualization 2000 Conference*, pages 445–448, 2000.
- [173] D. Weiskopf and M. Ansorg. Visualization of the general relativistic rigidly rotating disk of dust. *Annalen der Physik*, 9:SI179–SI185, 2000.
- [174] D. Weiskopf, D. Kobras, and H. Ruder. Real-world relativity: Image-based special relativistic visualization. In *Proceedings of the IEEE Visualization 2000 Conference*, pages 303–310, 2000.
- [175] D. Weiskopf, U. Kraus, and H. Ruder. Searchlight and Doppler effects in the visualization of special relativity: A corrected derivation of the transformation of radiance. *ACM Transactions on Graphics*, 18(3):278–292, July 1999.
- [176] D. Weiskopf, U. Kraus, and H. Ruder. Illumination and acceleration in the visualization of special relativity: A comment on fast rendering of relativistic objects. *The Journal of Visualization and Computer Animation*, 11(4):185–195, 2000.
- [177] V. F. Weisskopf. The visual appearance of rapidly moving objects. *Physics Today*, 13(9):24–27, 1960.
- [178] J. Wernecke. *The Inventor Mentor*. Addison-Wesley, Reading, 1994.
- [179] H. Weyl. *Philosophy of Mathematics and Natural Sciences*. Princeton University Press, Princeton, 1949.
- [180] J. A. Wheeler. *Geometrodynamics*. Academic Press, New York, 1962.
- [181] L. Williams. Casting curved shadows on curved surfaces. In *SIGGRAPH 78 Conference Proceedings*, pages 270–274, Aug. 1978.
- [182] L. Williams. Pyramidal parametrics. In *SIGGRAPH 83 Conference Proceedings*, pages 1–11, July 1983.
- [183] M. Woo, J. Neider, T. Davis, and OpenGL Architecture Review Board. *OpenGL programming guide: the official guide to learning OpenGL, version 1.1*. Addison-Wesley, Reading, MA, USA, 1997.
- [184] G. Wyszecki and W. S. Stiles. *Color Science*. John Wiley & Sons, New York, second edition, 1982.
- [185] Y. B. Zel'dovich and I. D. Novikov. Physical limitations on the topology of the universe. *JEPT Lett.*, 6:236–238, 1967.



# Danksagung

An erster Stelle möchte ich mich bei Professor Dr. Hanns Ruder bedanken, ohne dessen Unterstützung diese Arbeit nicht möglich gewesen wäre. Seine Offenheit auch gegenüber unkonventionellen Ansätzen und sein Enthusiasmus waren eine enorme Hilfe bei der Untersuchung dieses interdisziplinären Forschungsgebiets.

Darüber hinaus möchte ich allen Mitgliedern des Lehrstuhls für Theoretische Astrophysik an der Universität Tübingen für eine sehr angenehme Zusammenarbeit in den letzten Jahren danken – insbesondere Dr. Ute Kraus für hilfreiche Gespräche zur speziellrelativistischen Beleuchtungsrechnung; Jürgen Mellinger für die Zusammenarbeit bei der Visualisierung von Wurmlöchern; Stefan Kulla und Dr. Hans-Peter Nollert für die Kooperation bei der Einbettung von 2D Flächen; Daniel Kobras für die Zusammenarbeit bei der bildbasierten speziellrelativistischen Visualisierung; Dr. Johannes Ruoff bei der Visualisierung von Neutronensternschwingungen; den Systembetreuern, insbesondere Dr. Roland Speith, für schnelle Hilfe bei technischen Problemen; dem C10A02 – Daniel „Belboslav“ Kobras, Christoph „Teemann“ Schäfer und Raphael „Zladdi“ Dannecker – für manche wohlverdiente Tasse Tee oder Pause in der Cafete.

Ich möchte mich bei Professor Dr. Wolfgang Straßer für die freundliche Aufnahme am Lehrstuhl für Graphisch-Interaktive Systeme an der Universität Tübingen bedanken. Insbesondere danke ich Dr. Stefan Gumhold für viele fruchtbare Diskussionen; Ralf Sonntag für die Bereitstellung des Quelltextes zu RadioLab und für die Hilfe bei der Einarbeitung in dieses Programm; Dr. Tobias Hüttner für Tipps und Tricks zu OpenGL; Dr. Michael Meißner für Informationen zur hardwarenahen Grafikprogrammierung, die Bereitstellung von QGL-Viewer und Hilfe bei der Volumenvisualisierung.

Ebenso möchte ich mich bei Dr. Ulrich Lang und allen Kolleginnen und Kollegen der Abteilung Visualisierung am Rechenzentrum der Universität Stuttgart für eine sehr angenehme Zusammenarbeit in den letzten Jahren bedanken. Meinen Zimmerkollegen Dr. Roland Niemeier, Paul Benölken und Jürgen Schulze-Döbold verdanke ich interessante und lockere Gespräche – und viele leckere M&M's. Daniela Rainer, Andreas Werner und Uwe Wössner hatten Tipps und Tricks beim Umgang mit der VR-Umgebung und der Hardware parat. Die zahlreichen Filme konnten erst dank der Mitarbeit von Jutta Sauer, Uwe Zimmat und Hans Krämer entstehen.

Besonders danke ich Daniel Kobras, Bettina A. Salzer, Jürgen Schulze-Döbold und Nikolaus Weiskopf für das Korrekturlesen dieser Arbeit.

Zum Schluss möchte ich mich herzlich bei Bettina A. Salzer und bei meinen Eltern Elisabeth und Gerhard Weiskopf für ihre Unterstützung bedanken.