University of Derby

Interactively Visualising the Optical Effects of Special Relativity through Real-time Path Tracing

A thesis submitted in partial fulfilment of the requirements for the degree of BSc (Hons) in Computer Games Programming

Cathal Flynn 100458008

Academic Year: 2020/2021

University of Derby

Interactively Visualising the Optical Effects of Special Relativity through Real-time Path Tracing

A thesis submitted in partial fulfilment of the requirements for the degree of BSc (Hons) in Computer Games Programming

Cathal Flynn 100458008

Academic Year: 2020/2021

Acknowledgements

I would like to thank the faculty of the Computing department at the University of Derby for their support and advice over the past several years. In particular, I would like to extend my gratitude to my lecturers Chris Windmill and Wayne Rippin for helping me to develop my understanding and ability, and to my dissertation supervisor Leonardo Stella for his guidance and feedback through this process of academic writing.

I would like to express my thanks also to my colleagues at MXTreality, who have endeavoured to keep me involved while I complete my education and to whom I look forward to returning.

I extend my gratitude to the Computing Society at the University of Derby and the Interesting Times Gang, groups from which I have made a number of lifelong friends who have supported me in writing this dissertation, as well as weathering the COVID-19 pandemic.

Lastly, I would like to thanks and appreciation to Tom, Brian and Dom for their time spent proofreading, and my family for the plentiful supply of cat pictures.

Abstract

In this paper we evaluate the use of general-purpose GPU (GPGPU) programming and ray tracing to produce interactive, real-time simulations of the optical effects of special relativity. We consider the value of interactivity in relativity visualisation in physics education. We discuss the usefulness of rasterization, ray tracing and image-based technique for interactive relativity visualisations. We examine existing interactive and offline visualisations of relativistic effects. We implement a path tracing renderer using compute shaders in the *Unity* game engine. We consider the implications of recent developments in real-time ray tracing hardware and their impact on relativistic visualisation. We demonstrate an approach to visualise optical effects in real-time by modeling the physical behaviour of light according to Einstein's Special Theory of Relativity. We outline the limitations in our work, and suggest potential directions future work could take in terms of functionality and performance.

Contents

List of Figures

Chapter 1

Introduction

Einstein's theories of Special and General Relativity underpin modern physics at a macro scale, with profound implications for reasearch in astronomy, astrophysics and terrestrial applications such as telecommunications and navigation. Relativity also provides us with a model for the fundamental nature of universe, causality and the concepts of space and time. However, relativity can be a difficult subject to approach in physics education, since it is contradictory of our experience of every day physical phenomena. Relativistic phenomena can be unintuitive and mathematically complex, while Newtonian physics agrees well with our preconceived notions of Euclidean space and universal time.

Colloquially, 'relativity' refers to Einstein's general theory of relativity, which is a refinement of his ideas to include gravity within its framework. Special relativity deals specifically with problems relating to the observation events while travelling at a significant proportion of the speed of light, c. Our research is focused on special relativity.

In physics education, models and demonstrations are used to aid students in developing an intuition for particular physical effects and processes. Newtonian mechanics can be demonstrated easily in the classroom by way of physical experiments which can be performed directly by students. However, relativistic effects are intangible at a human scale, being invisible in reality except in the contexts of astronomy or fast-moving spacecraft. Unfortunately, since our senses are tuned to expect rigid spatial and temporal dimensions, it is often difficult for students to develop an intuition for relativistic concepts. This is in contrast to Newtonian mechanics, which may be easily demonstrated and observed in the classroom (Scherr, Shaffer, and Vokos 2002). As a result, relativistic concepts can remain opaque to the student even after advanced instruction. Physics educators must rely solely on mathematics, thought experiments and visualisations in order to teach the subject (Kaur et al. 2017).

Recently, as graphics processors and real-time rendering algorithms have improved, there has been an interest in the use of real-time visualisations as tools for developing an understanding of special relativistic optics (Sherin et al. 2016) (Savage, Searle, and McCalman 2006). Recent advancements in graphics technology such as NVIDIA's Turing architecture (Burgess 2020) make the use of path-tracing in real-time special relativity visualisations feasible on consumer hardware. This is particularly attractive as it offers the potential to intrinsically model relativistic effects instead of relying on approximations as with rasterisation- or image- based techniques by simulating the propogation of light around a scene over time.

1.1 Motivations

Visualisations are of great value when trying to foster student intuition and understanding of difficult physics concepts (Kaur et al. 2017). This is particularly true of relativity, as the circumstances under which relativistic effects can be observed require great speed or distance, or both - neither of which are available in a conventional physics classroom. Interactive visualisations in particular are of greater interest due to their ability to actively engage the student in exploring the nature of a topic, which has been shown to improve student understanding and learning retention (Annetta 2008) (Cheng et al. 2017).

In addition to an inability to internalise relativistic concepts, there is a difference between the idealised inertial observers on which Einstein's equations are based and a personal observer. In the case of the former, relativistic effects are related by way of relatively straightforward mathematical equations, describing the results of instantaneous, idealised measurement. In the latter, relativistic optics must also account for the time taken for light rays to reach the observer in order to form an image, producing additional distortion. This difference between what is 'measured' and what is 'seen' is a well-documented difficulty of teaching relativistic concepts (Terrell 1959) (Chang, Lai, and Chen 1996) (Hughes and Kersting 2021) (Müller, Grottel, and Weiskopf 2011) and of particular interest to visualisations which attempt to reproduce what an observer would see in particular scenarios.

Prior work on interactive visualisations (Sherin et al. 2016) (Müller, Grottel, and Weiskopf 2011) have noted difficulty with rasterization-based approaches, since the deformation of model vertices quickly results in visual artifacting. Image-based (Acoleyen and Doorsselaere 2018) and particularly path-tracing based (P. Hsiung and Dunn 1989) approaches can minimise such artifacting with pixel-perfect imagery, but are generally either inflexible or too computationally expensive for consideration in 3D real-time applications.

By providing more accurate and interactive visualisations of relativity to students of the topic we can provide a more engaging learning experience. Additionally, by ensuring that produced visualisations are correct we can reduce the long-term proliferation of errors or misconceptions about the underlying physics. Path-tracing techniques in particular offer a route to maximising the physical accuracy of visualisations by directly modelling the behaviour of light as opposed to approximating it with traditional 3D real-time rendering techniques.

1.2 Aims

We aim to demonstrate the use of GPGPU programming to produce a real-time, path-traced interactive visualisation of special relativity. We will use the popular *Unity* game engine [1] to develop our visualisation. By using off-the-shelf software and hardware, we plan to demonstrate that a path-traced approach is viable for use in interactive applications.

Our objectives are:

1. Investigate contemporary literature on creating visualisations of special relativity and acceleration techniques used in real-time path-tracing.
2. Design and implement a process by which rays generated from a GPGPU path-tracing application can be modified to produce the optical effects of special relativity.
3. Compare the results produced by our implementation with prior work in terms of accuracy, performance and features.
4. Critically evaluate and discuss specific elements of our implementation.
5. Identify the limitations of our implementation and propose potential directions to be taken by future work.

[1] Website. Unity Technologies `https://unity.com/`

Chapter 2

Literature Review

In this chapter we review past and current literature on producing visualisations of special relativity. We first provide a description of the physics of special relativity and its implications for the optical effects observed by a camera which is moving at a significant fraction of c. We then explore prior offline and interactive visualisations and a comparison of contemporary rendering techniques used by them. We then further evaluate methods of improving performance of path-traced applications in terms of visual quality.

2.1 Special Relativity

Special relativity describes the deformation of spacetime between inertial reference frames (DiSalle 2020) moving relative to each other at a high proportion of c. The principle of relativity is that the laws of physics and c must remain constant in all reference frames. Since speed is a function of distance and time, we see that space and time must therefore alter depending on the observer and observee in order to preserve a constant c. These effects would be perceived as time dilation and length contraction by an observer. Known together as the Lorentz contraction, this effect is described by Lorentz and Poincaré group transformations (Weiskopf 2010).

The Doppler effect is the apparent shift in the observed wavelength of light as a result of the Lorentz contraction. This effects the perceived colour of a light ray, and may cause rays to shift into or out of the visible spectrum in a process known as blue- or red-shifting. The searchlight effect increases the luminance of light in the direction of travel, a consequence of time dilation, the Doppler effect and relativistic aberration (Weiskopf, Kraus, and Ruder 2000).

In relativity, Minkowski spacetime is a four-dimensional manifold composed of three-dimensional space and an additional dimension, which we generally experience as time. Within this manifold, a four-dimensional point consisting of three spatial coordinates and a temporal coordinate is known as an event. Spacetime events are separated not only by space but also by time, due to the finite speed of light. The use of such a manifold allows us to specify an absolute distance between two events, although the observed space and time between them may differ by the reference frame of the observer.

The properties of four-dimensional spacetime are visible only when observing light which has originated in an inertial reference frame moving at a high fraction of c relative to the observer. Optical distortions appear, which alter the shape, colour and brightness of objects as a function of ϕ, the angle of the observed object from the direction of the observer's travel relative to it. Visualisations of special relativity aim to recreate these effects virtually, since they cannot be achieved by any real observer due to the extreme difficulty of achieving relativistic speeds. The precise nature of the deformation of spacetime according to relativity is described by equations in the Lorentz and Poincaré groups.

2.2 Visualisations of special relativity

For demonstrations of geometric relationships within spacetime constructions such as Minkowski diagrams are traditionally used. They are versatile, often used in textbooks. It is possible to make dynamic physical analogues [1] which can aid learning usefulness. However, since they are presented in Euclidean space as either two- or three- dimensional graphs they cannot fully represent the true four-dimensional geometry of spacetime (Weiskopf 2010).

Academic software such as The Einstein Toolkit (Haas et al. 2020) for example provides a number of distinct core tools for use in scientific visualisation, both standalone and for use in combination with other systems. However, while valuable to the physics community at large the monolithic nature of such a project and its sheer complexity is unlikely to encourage newcomers to the subject.

Instead, we may hope to provide an intuition or spurn further interest in the topic by way of camera-centric visualisation techniques, which attempt to represent what a camera would see in a particular relativistic situation. An interest in such visualisations existed at least as early as 1939 when Gamow (1939) imagined the experiences of a cyclist in a world where c was a comparable speed to that achievable by bicyle. The computational complexity of computing such

[1] Video: *Lorentz Transformations — Special Relativity Ch. 3* `https://www.youtube.com/watch?v=Rh0pYtQG5wI`

an image would not be available for decades until P. Hsiung and Dunn (1989) provided a basis for applying Lorentz transformations to simulated light rays as a means of visualising the view of a relativistic camera.

The film *Interstellar* (2015) had a Hollywood visual effects studio work in tandem with physicist Kip Thorne to produce images of the optical effects in the vicinity of a Kerr black hole (James et al. 2015), where gravitational effects result in the extreme warping of spacetime. Since their work was intended for cinematic application, focus was put on the temporal stability of the image over a sequence of frames rather than the raw images preferred for scientific purposes. It includes reference to the simulation of a physical IMAX camera in addition to the direct optical qualities of the black hole.

A key issue of the visualisation of special relativity is due to the difference between instantaneous measurement of alternate reference frames with the reality of what would be seen by a relativistic observer, for whom vision is formed as a consequence of arriving photons moving at c (Lampa 1924) (Penrose 1959) (Terrell 1959) (Hughes and Kersting 2021). This makes for a number of subtleties in visualisations which mean that computation is required to accurately generate an image incorporating relativistic optical effects.

When shown an interactive visualisation of special relativity, researchers found that first year students (Savage, Searle, and McCalman 2006) appreciated the "concrete and visual" nature of the software in favour of their labatory manual. The researchers found that students would lose faith in the simulation when it behaved unexpectedly, with students describing the controls as "hard to use". Students' lack of understanding of the mechanics at play would also be blamed on a software "bug", though there was none.

OpenRelativity (Sherin et al. 2016) showcases a rasterization-based approach to visualising the effects of special relativity using traditional vertex and pixel shaders. They successfully demonstrate effects of special relativity such as time dilation and length contraction as well as the Doppler and searchlight effects, implemented in the Unity game engine. Notably, they utilise textures to represent non-visible parts of the light spectrum, modelling the way in which ultraviolet and infrared light can become visible when travelling at relativistic speeds towards or away from an emission source.

The authors of OpenRelativity produced *A Slower Speed of Light* (Sherin et al. 2013), a complete game which takes place in a relativistic universe. As the player moves around an "immersive and engaging" world, they are able to interact with objects in order to progress through the game. As they progress, the optical effects of special relativity become more pronounced and impede the player's progress.

The inclusion of interaction with other objects is notable, as it requires integration with an underlying physics engine to compute which objects have been interacted with. This is non-trivial in a relativistic context, as objects are subject to appear to an observer to move along distorted paths relative to their local reference frame. The authors acknowledge this limitation, stating that their physics objects are limited to moving at a constant speed in a single direction. Additionally, they note lack of dynamic lighting and an over-simplified skybox as limitations of their implementation.

2.3 Rendering techniques

In any rendering context there is a tradeoff between accuracy and speed which is paramount to developers of interactive media. In general, rendering techniques and hardware are developed with Euclidean space in mind, so special consideration must be made in order to repurpose them for non-Euclidean contexts.

2.3.1 Rasterization

Rasterization is a rendering technique by which a scene is approximated with triangular meshes and subsequently converted to a raster image to be displayed on a screen. Rasterization has enjoyed superiority in real-time graphics contexts such as computer games due to its relative speed and flexibility. With sufficient quality meshes and use of programmable rendering pipelines near-photorealistic results can be achieved, as seen in contemporary computer games.

However, as with other non-Euclidean rendering contexts [2], relativistic optical effects are likely to significantly deform objects, so rasterization-based techniques must use high-fidelity mesh data to minimise visual artifacts. This reduces the quality of the resulting image as artifacts of this type are impossible to remove completely, and it reduces the efficiency of the rendering pipeline since more video memory is required to store the additional mesh data.

Hybrid approaches between rasterisation and ray tracing have been used (Müller, Grottel, and Weiskopf 2011), affording interactive visualisation framerates using geometry shaders to perform geometric distortion. This approach negated the need for expensive ray-triangle intersection through the use of local image-space ray tracing, in which rays are targeted only at scene fragments visible to the camera. This is effective for achieving simple interactive relativistic visualisations, and presents a technique for correctly shading fragments in accordance with the Doppler and searchlight effects, but is lacking compared to a fully ray-traced based approach as multiple bounces of light around the scene cannot be correctly calculated as in a fully path-based approach.

2.3.2 Image-based

Non-stereoscopic images are formed by pencils of rays (Nichols and Franklin 1903). With image-based techniques, an image can be deformed to visualise some relativistic effects through the use of spherical coordinates (Weiskopf, Kobras, and

[2] Video: *Projecting Space - Hyperbolica Devlog #4* `https://www.youtube.com/watch?v=rqSLu0R3dwY`

Ruder 2000). This technique is attractive as it is computationally cheap, being unlikely to require hardware acceleration as with rasterisation or path tracing. Additionally, the use of standard image formats make it trivial to use a variety of useful images, such as real-life photographs (often panoramic, 360-degree photos) or tools such as a patterned grid in order to measure the deformation of the image. A particular advantage of image-based techniques is that they are compatible with 360° video, applicable in virtual reality contexts, as demonstrated by the VR movie *Captain Einstein* (Acoleyen and Doorsselaere 2018).

Image-based techniques are limited in their ability to represent fully 3D scenes, as a static image would be unable to present changes in perspective and visible objects as a camera moves through the scene. It would also be difficult to represent the different passage of time in different reference frames within the image, although it could potentially be explored by treating video as a three-dimensional 'stack' of frames [3].

2.3.3 Path Tracing

Monte Carlo stochastic ray tracing, or path tracing (Kajiya 1986), produces images by casting individual rays of light through a scene. These rays are tested for intersections with scene objects. Additional rays are recursively formed and traced to simulate the scattering of light around the scene. With path tracing it is conceptually easy to produce realistic lighting effects, such as ambient occlusion and accurate refraction, which can only be approximated with other rendering methods. When an intersection between a ray and a scene object is found, the ray is imbued with the material properties of the surface. The surface normal of the object at the hit point is then used to either reflect or refract the ray according to its bidirectional reflectance distribution function (Nicodemus 1965). The reflected ray is its own ray, which is itself then cast into the scene to repeat the process. This process is repeated for every pixel of the image for some maximum number of bounces. As a result, though path tracing may be versatile and accurate in the images it can produce, it is also extremely computationally expensive, which has historically precluded it from use in real-time contexts. Execution time scales with image resolution, number of simulated bounces per ray and overall scene complexity.

Path tracing is attractive when modelling large numbers of lights, since the illuminance of each ray is calculated each time it hits a surface, as opposed to being a function of the number of lights in the scene. As a result, the runtime of computing the scene illumination from a given number of emissive objects is equal to the cost of computing the scene lighting of the same number of diffuse objects. Emission-based lighting such as this is not available in rasterization-based pipelines, which must instead approximate lighting with point, spot and directional lights.

Exact solutions can be found for ray intersection with simple geometric primitives[4], with solutions given in terms of the distance along an intersecting ray. Such primitives can then be composed into complex scenes. Since they themselves are composed of simple geometric triangles, polygonal meshes can also be included in such scenes. In the context of relativistic visualisation, ray tracing can achieve pixel-perfect imagery without deforming triangulated meshes and so can be immune from artifacts resulting their use. This is a significant advantage when dealing with scene geometry that may be heavily distorted by relativistic optical effects.

Early relativistic ray tracing implementations (P. Hsiung and Dunn 1989) (Howard, Dance, and Kitchen 1995) formed low-resolution static images of relativistic scenarios. This allowed for the direct comparison of the contribution of different relativistic effects to a final image and for the validation of the invisibility of the Lorentz contraction as observed by Penrose (1959).

The REST-frame technique (P. Hsiung and Dunn 1989) describes an approach by which rays can be simulated as having a finite speed, in contrast to contemporary and modern path traced applications. This allows for implicit modelling of the Doppler effect and visually correct 'observations' as opposed to 'measurements' of relativistic effects. This is achieved by considering image formation to take place at a particular spacetime event within the observer's inertial reference frame. This information can then be used to compute not only hit positions as with normal ray tracing, but additionally interweaves temporal information with each ray. Rays are initially formed in the reference frame of the observer but are related to other, moving reference frames by way of Lorentz transformation. In their work they align the observer and observee reference frames along a shared axis. This is inconsequential for simple, non-interactive applications. However, for a complete and interactive visualisation, free movement in all three spatial axes would be preferable.

A key insight into performing relativistic rendering is associating scene depth with time as described by P.-K. Hsiung, Thibadeau, and Wu (1990). While in most rendering contexts c is assumed infinite, the second postulate of Special Relativity is the constancy of c: in a vacuum, c is a constant *finite* value regardless of inertial reference frame (Resnick 1968). Light's finite speed therefore allows us to utilise a *time buffer* to distinguish elements temporally beyond their 3D spatial reprsentation.

The modelling of physical light is useful beyond the intrinsic simulation of relativistic optical distortion. It additionally allows for more sophisticated simulation of the image-forming camera itself (Courtial et al. 2016). Though effects such as chromatic aberration and depth of field may be quickly approximated through image- and depth- based post-processing steps, it may be of value to physically simulate the internal optical mechanisms of a relativistic camera. This has already found application in non-interactive visualisations, notably in the cinematic rendering of a black hole in the film *Interstellar* (James et al. 2015).

[3] Video: *video is 3D. not 2D* `https://www.youtube.com/watch?v=NZFxQXe7LMM`

[4] Inigo Quilez. *distance functions.* Accessed: 2021-04-02. URL: `https://iquilezles.org/www/articles/distfunctions/distfunctions.htm`.

Acceleration Techniques

Path tracing applications with a real-time constraint are subject to high amounts of noise as only a few paths per pixel can be traced within a few milliseconds. This results in extremely noisy per-frame data. Given that Monte Carlo methods can take many thousands of samples to ultimately converge on a noise-free image, this is unlikely to change in the near future. As a result, contemporary real-time path tracing pipelines (Galvan 2020) are designed with single sample per pixel (1spp) images in mind to accommodate the greatest possible fraction of consumer graphics cards. They make use of denoising, filtering and spatiotemporal reprojection techniques to make efficient use of the available processing power each frame.

Deferred shading real-time graphics pipelines make use of auxiliary textures (G-buffer) for use in performing shading operations. These buffers are generally computed in a pre-pass step, before the bulk of the shading work begins. While depth- and stencil-buffers have hardware support, the use of additional textures per-application allows for general usage of view-space acceleration methods. An image buffer may store a wide variety of view-space information, such as per-pixel velocity, surface normals or a history buffer. While not specific to path-tracing pipelines, the use of a G-buffer allows for considerable and relatively cheap improvement to visual quality of an image. For example, for a given viewpoint in a three-dimensional scene, each point is a particular distance, or depth from the observer. This information can be written to a depth texture for use in image synthesis. This buffer is crucial for efficiently rendering scenes, as it allows for quick culling of occluded objects. In a traditional games context it may also be used for effects such as fog.

Consecutive frames of an animated film tend to share most of their data, with only minor differences between them. This property is known as temporal coherence, and can be exploited to produce images of a higher effective sample count than if only data from the current frame were used (Mäkitalo, Kivi, and Jääskeläinen 2020). In temporal anti-aliasing, motion data from the camera and moving scene objects are used in order to calculate per-pixel motion vectors, with which samples can be reprojected from the previous frame. Though this is liable to introduce its own artifacting [5], this is often preferable to a noisy image produced from a single frame of data. Image reconstruction techniques such as spatiotemporal variance-guided filtering (Schied et al. 2017) are also used extensively. Temporally-based techniques reproject image data from previous frames by use of camera and scene motion data in the form of G-buffer textures. Additionally, the use of history buffers allows direction of spatialtemporal algorithms to priority areas as well as guiding the rejection of invalid samples from previous frames, such as those which are newly occluded by scene geometry.

When sampling a scene, a naive Monte Carlo path tracer aims each ray through one pixel of the image plane. However, regular grid sampling is sub-optimal when considering the distribution of image data, resulting in a sparse distribution of samples (Wolfe 2017). Instead the precise location of samples taken can be directed by some additional input to more evenly sample a scene, disconnected from the arrangement of pixels on an arbitrary image plane. Blue noise is a form of low-disrepancy noise which provides a improved distribution of values when compared to white noise (Russell 2019). By pre-generating blue noise textures, this improved distribution can be made use of in real-time with a low computational cost, resulting in a more evenly sampled image which requires fewer samples to converge into a clean image.

NVIDIA's Turing architecture (Burgess 2020) supports path-tracing based rendering pipelines in a number of ways. Harware acceleration of ray-triangle intersection permits the use of standard polygonal meshes in real-time environments, as rays can be efficiently tested against arbitrary geometry. Additionally, acceleration by machine learning-based upscaling such as deep-learned supersampling (DLSS) (Andrew Edelsten 2019) or neural supersampling (Xiao et al. 2020) is made more accessible. This allows a path tracing pass to take place at a lower resolution, reducing the amount of computation needed for a given frame. The image is then quickly and accurately upscaled through use of a pre-trained machine learning algorithm to match the target display.

2.4 Colour Representation

Correctly representing colour is a challenge for relativistic visualisations because of the mismatch between the sRGB model used by computer displays and that of real light, which exists as electromagnetic waves with a range of wavelengths. Standardised colour spaces such as CIE 1931 (Hunt and Pointer 2011) define colour-matching functions that relate real wavelengths of light to colours in the proportional red-green-blue format used for in computer graphics. These form an additional colour space known as XYZ, which can be converted to and from both RGB and wavelength values.

In addition, relativistic visualisations have an additional challenge in that they must model wavelengths of light beyond the visible spectrum. Sherin et al. (2016) define additional colour-matching functions for infrared and ultraviolet wavelengths as ad-hoc extensions to the CIE tristrimulus functions. They make use of additional material textures to represent the colour reponse of surfaces to wavelengths outside of the visible spectrum.

In traditional physically-based rendering contexts the colour of light reflected from a surface is approximated from various surface properties, represented by textures. However, to additionally model surface colour responses to light outside the visible range, a complete spectrum is required for each object material (Müller, Grottel, and Weiskopf 2011). In this way, an incident ray can look up surface reflectance values according to its wavelength, regardless of its relation to visible light specifically.

[5] Video: *Temporal Antialiasing Artifacts - DOOM* `https://www.youtube.com/watch?v=HZJVIOxFB7M`

Chapter 3

Methodology

In a traditional path tracing process rays are cast from the camera to the scene, through the pixel. We implement an additional stage to this process, altering the direction, wavelength and intensity of individual rays to form a relativistically distorted image. We initially validate our method through the creation of geometric models using Unity gizmos, before implementing a proof-of-concept GPGPU rendering step. Our final implementation allows the user to control a camera flying through an infinite 'lattice' structure which deforms in adherence with relativistic aberration, including the Doppler and searchlight effects. The scale of the deformation is driven by the speed of the camera, capped at c. Our implementation is available on Github[1].

3.1 Gizmos

When implementing relativistic effects, we used a bottom-up approach to develop an understanding of the underlying physics. Using Unity's gizmo system we were able to replicate geometric constructs from the literature. These provided insight into the mechanisms of relativity and formed a basis from which a GPGPU implementation could be developed. In particular, we created simple interactive geometric models of Lorentz transformations and of relativistic aberration (Fig 3.1). This helped us to formulate a model of a relativistic path tracing process (Fig 3.2) from which we were then able to implement a GPGPU rendering algorithm.

3.2 Relativistic Rendering

Though relativity considers all reference frames equal, observer-centric visualisations may consider the consider the observer to be at rest in the 'proper' reference frame, using 'proper' time and 'proper' length. Measurements of time and space in the 'proper' reference frame agree with classical physics. Observed time and length in other reference frames can then be described in terms of the local reference frame. Lorentz transformations are used to relate measurements of space and time between reference frames.

Relativistic effects scale with the relative speed of the observer to another reference frame as a proportion of c, as given by

$$\beta = v/c. \tag{3.1}$$

To simplify the mathematics and to keep values to within a reasonable range for single-precision floating point representations many visualisations normalise c to 1, making β and v interchangeable in most cases.

The Lorentz factor γ is given by

$$\gamma = \sqrt{1 - \beta^2}, \tag{3.2}$$

where v is the difference in velocity between two inertial reference frames and c the speed of light. Once obtained, this can be used to calculate measured lengths and times in a separate frame of reference based on a known value for β.

3.2.1 Aberration of light

The aberration of light refers to the apparent deflection of the path of a ray of light when observed from a moving reference frame (Gjurchinovski 2006). In two axis-aligned reference frames moving with respect to each other, a ray with direction ϕ in the origin reference frame has apparent direction ϕ' from the perspective of a moving observer. Classical and relativistic models of light provide different solutions for this transformation.

Under classical physics, the path of light is deflected due to an observer's movement in the same manner that the path of vertically falling raindrops is deflected sideways when observed from a moving vehicle. This behaviour was first observed by classical astronomers on discovering the aberration of starlight due the to movement of the Earth around the Sun (Stewart 1964).

[1] Cat Flynn. *srpt*. Source code repository. Accessed: 2021-05-07. 2021. URL: `https://github.com/ktyldev/srpt`.

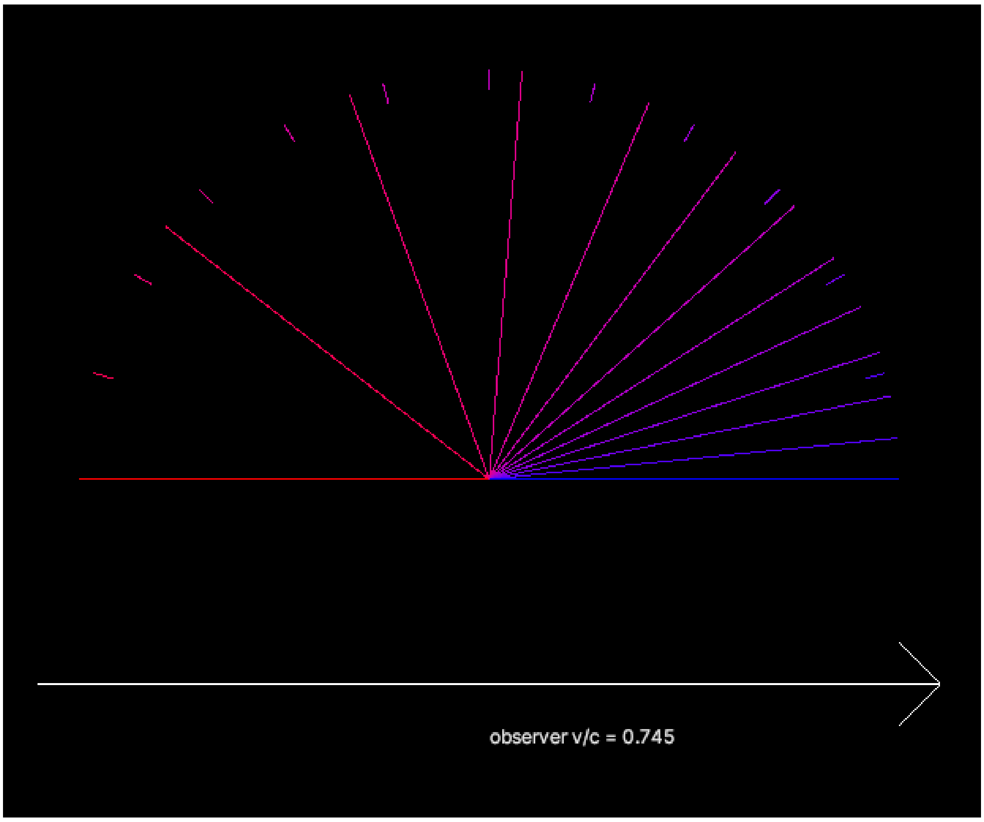

Figure 3.1: A simple 2D representation of relativistic aberration.

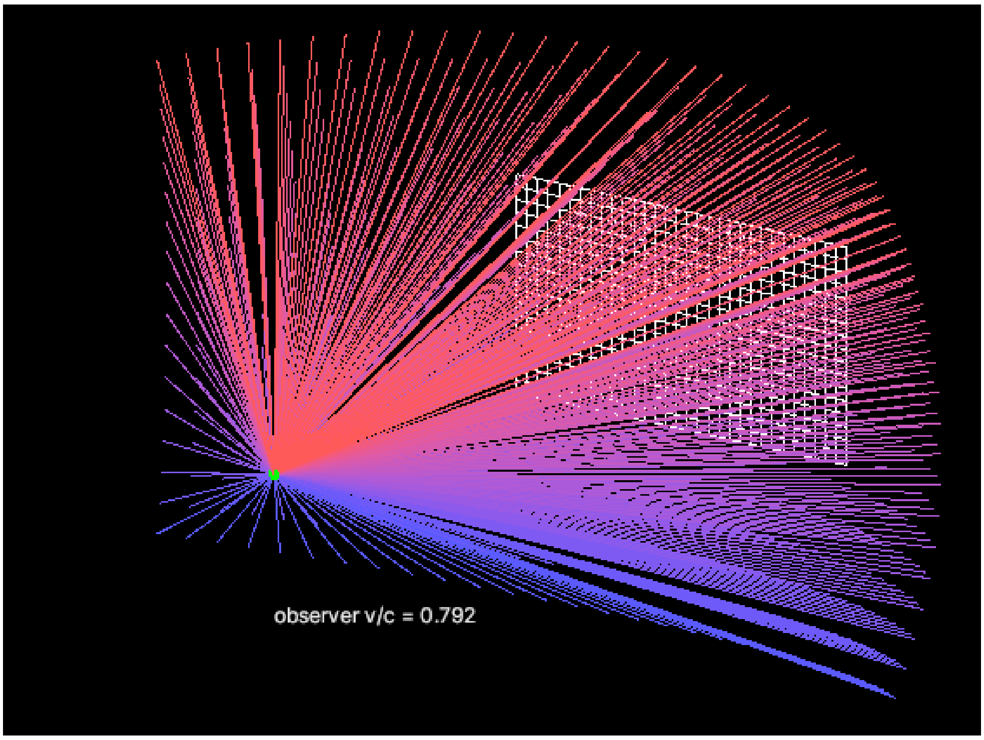

Figure 3.2: An early model of our relativistic aberration ray transformation.

This holds for low values of v/c, such as when v is Earth's velocity around the Sun. However, it implies that aberration greater than 45° is achievable for an observer moving faster than c, which cannot be attained. This also has implications for the measurement of the speed of light based on events observed from inertial reference frames moving with respect to each other, as two observers might measure a light beam to have the same component of velocity orthogonal to the direction of travel, but different components parallel to the direction of travel, resulting in differing values for measurements of c.

The relationship between the apparent direction ϕ of an incoming light beam and the direction from which it arrives into the reference frame ϕ' is given by

$$\cos\phi' = \frac{\cos\phi - \beta}{1 - \beta\cos\phi}. \tag{3.3}$$

In this way, we see that the aberration of light approaches a maximum of 180° when $v = c$. Visually, this corresponds to all observed light emanating from a point in front of the observer in the direction of travel.

3.2.2 The Doppler & searchlight effects

The Doppler effect in relativity is the red- and blue-shifting of light, wherein the observed wavelength changes due to the relative velocities of the observer to the light's source. The perceived colour of light is a function of its wavelength(s) and does not stop at the ends of the visible spectrum. To correctly represent colour in the context of a relativistic visualisation, light must be modelled in terms of wavelength before being converted to traditional sRGB values for rendering (Walker 1996). In addition to requiring a fuller representation of colour to allow for Doppler shifts, luminance information must also be included to visualise the searchlight effect. The Doppler factor D is given by

$$D = \frac{1}{\gamma(1 - \beta\cos\phi)}. \tag{3.4}$$

The searchlight effect is a consequence of the increase in luminosity associated with a decrease in wavelength, causing the image to appear brighter in the direction of travel, given by

$$1/D^3. \tag{3.5}$$

3.3 GPGPU algorithm

We initially implemented a simple GPU-based path tracer with only ray-sphere intersection using Unity's ShaderLab and HLSL (Kuri 2018). We then implemented support for ray-cylinder intersection through the use of an intersection function. From these primitives we were able to construct a simple scene. We found a lattice structure to be best method for communicating the deformation of space in three dimensions, in the vein of Walker (n.d.), P. Hsiung and Dunn (1989) and Courtial et al. (2016).

Our path tracer contains two stages. In both stages, rays cast from the observer are first transformed with respect to the observer's velocity relative to the scene β (Eq 3.1). In this way we are able rotate the direction of camera rays in accordance with the aberration formula (Eq 3.3). The result of this is a path traced into the scene with angle ϕ', though from the camera's perspective it appears to arrive from ϕ. We can additionally calculate the D trivially at this stage (Eq 3.4), which is applied to the ray's wavelength. The direction and wavelength of the ray are not changed after being cast from the observer.

Initially, a single ray is cast for each pixel to determine its depth value, which is written to memory for use by the shading algorithm. Though Unity uses the hardware depth buffer for more generic rendering contexts, we did not make use of it in our method. In the shading stage, rays are additionally re-cast non-relativistically from scene intersections to re-sample the scene and more accurately represent its modelled path of light. The number of additional paths traced by a ray is scaled by the value determined in the earlier depth pass, which improves visual quality by prioritising camera rays which intersect the scene nearby to the observer over those which hit distant geometry or hit nothing at all.

3.4 User Interaction

Our final implementation has simple controls, in which the user can control the orientation of the camera and additionally move forwards and backwards. The camera has an asymptotic maximum speed of c, which is scaled to 1 in the simulation world. User input accelerates the camera either forwards or backwards along its local z-axis. Values for β and γ are computed and supplied to the GPU with the camera position and lattice rotation. The user is able to toggle various elements of the visualisation, such the Doppler and searchlight effects. They are additionally able to toggle the range of wavelengths for which rays are generated.

At any point the user is able to pause their movement while maintaining the latest value of β. In this way the user is able to manipulate the camera to move to a position and speed they are satisfied with before pausing the visualisation to inspect the environment. This allows the user to view effects orthogonal to the direction of travel, as well as to arrange the view such that a clean image can be generated of relativistic effects, since if the camera is at rest the scene experiences no distortion.

Our implementation of relativistic aberration is oriented along the world's z-axis, due the lack of handling of Poincaré transformations. The camera therefore moves only along this axis. The lattice environment is rotated around the camera to provide the illusion of navigating freely around the scene. As the camera moves through the environment, it is repositioned to keep it in the centre 'cell' of the lattice. In this way the environment is effectively made infinite, allowing the user to continue to view the optical distortion of the environment regardless of how far they travel.

Chapter 4

Verification & Validation

In this chapter we analyse the accuracy and performance of our implementation. To evaluate accuracy, we compare our visualisation to prior examples to observe the produced optical distortion. Due to the graphical nature of the topic, our assessments are generally qualitative and achieved through visual comparison of images produced by the visualisations in question, rather than quantitative.

4.1 Accuracy

In implementing our original geometric visualisation we validated our approach by comparison to prior interactive implementations, such as (Walker n.d.) and (Courtial et al. 2016). We were able to validate aberration angles of rays produced by our gizmo visualisation with those produced by other visualisations for similar values of β. Their visualisations feature the ability to visualize individual effects in isolation from each other, which would be useful in pedagogical contexts. This is possible in our implementation through the setting of shader properties.

Rasterisation-based visualisations such as (Sherin et al. 2016) simulate aberration by modifying the position of model vertices in a vertex shader. Our visualisation is immune from artifacts resulting from this vertex modification step since we use rays to form an image on a per-pixel basis. Cylinders can be seen to bend smoothly along their length, which would not be possible for vertex-based distortion. This would remain true in the case that we supported the use of polygonal meshes.

Similarly, we validated our implementation of the Doppler and searchlight effects by visual comparison with sources such as (Sherin et al. 2016) and (Acoleyen and Doorsselaere 2018). We were able to verify the expected colour banding formed as rays in different parts of the image are red- or blue-shifted with changing values of ϕ. Additionally, our implementation correctly brightens the image in the direction of travel and dims the view behind.

Our implementation does not consider the Lorentz contraction nor the finite runtime of light. As a result, it is incapable of representing the altered passage of time in observed reference frames. The omission of the Lorentz contraction, while shown to be relatively inconsequential to the overall distortion (Howard, Dance, and Kitchen 1995), renders the visualisation incomplete. Finally, our camera is an idealised pinhole camera with a focal length of infinity; deeper explorations into what a relativistic camera would see must necessarily simulate the camera's internal function (Courtial et al. 2016).

4.2 Performance

Our path-tracing based implementation of relativistic optics avoids artifacting from the distortion of object vertices, which improves the visual quality at high values of distortion. Additionally, the use of path tracing allows for physically correct shadows, which was noted by Sherin et al. (2016) as a drawback in their method due to their omission of a relativisitic shadow casting step.

However, since our stochastic image-forming process is staggered over several frames, the image is initially noisy and clears only after accumulating enough samples, which is a drawback when compared to rasterisation or local ray tracing. Our implementation does not include any reprojection or filtering methods, which would improve the stability and quality of the image while the camera is moving.

We were able to dramatically improve image quality with the accumulation of samples across multiple frames. As with traditional path-traced image synthesis which accumulates many samples to form a final image, we take advantage of an unchanging scene when the camera is stationary in order to make use of samples from previous frames. Because of the rapid framerate, this allows the image to quickly improve in quality from a noisy 1spp input, though convergence to a clean image is still time-consuming. Additionally, any accumulated samples are immediately discarded once the camera moves or when the scene is changed in any way, resulting in a quick return to a noisy 1spp image.

Our use of a depth buffer generally improves the quality of the image. Using information from the prior relativistic depth pass the shading pass can prioritise nearby areas of the scene, rendering them at a higher fidelity by increasing the

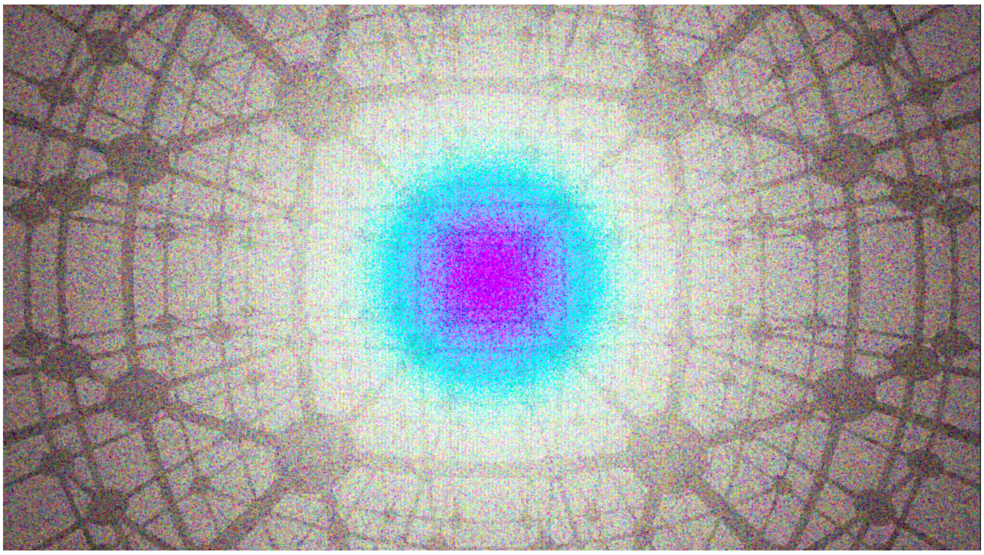

Figure 4.1: A noisy image generated without depth information.

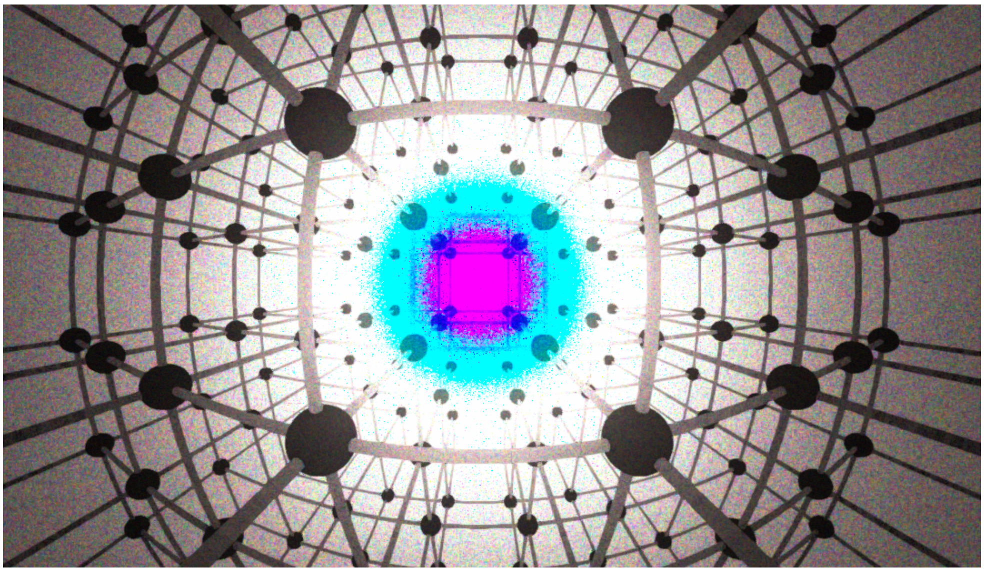

Figure 4.2: Nearby samples are traced preferentially, resulting in a cleaner image.

number of bounces taken by the ray through the scene. We scale the number of bounces taken by cast rays by the value from the depth buffer, such that nearby samples have the maximum number of scene bounces and therefore the most accuracy, while distant samples are traced only once. While the total number of samples taken is similar, by including depth information we are able to spend the most time on more important samples. The graphical improvement is shown in (Fig 4.1) and (Fig 4.2).

We were able to slightly improve the shading runtime by sampling the depth texture to test if any scene geometry was intersected for a particular ray. In the case that there is no scene intersection, the expensive lattice intersection tests may be skipped for that ray. Though still subject to warp divergence, this allows for accelerated rendering of some parts of the image. In general, frequent branching on the GPU hinders the runtime improvements of using a depth buffer. This optimisation is less effective as β, as more of the lattice is visible causing fewer warps to skip scene intersection.

Chapter 5

Discussion

The implementation of lighting and colour are challenging for relativistic visualisations, as contemporary real-time techniques are approximations to the human perception of light in the visible spectrum as sRGB values, as opposed to electromagnetic waves of a given wavelength. Individual rays of light must carry intensity and wavelength information during the shading step, but presented in a format expected by the current graphics context.

5.1 Lighting and colour

Our visualisation models rays of light having wavelengths, as opposed to an sRGB colour. This approach allows us to represent rays of light that may be red- or blue- shifted into the visible spectrum as a consequence of the Doppler shift. Wavelengths can then be approximately converted to sRGB values through the use of standard colour-matching functions (Smith and Guild 1931).

When a ray is created for shading, it is assigned a wavelength between the minimum and maximum possible wavelengths that could potentially be shifted into the visible spectrum. This creates an approximately even spread of wavelengths throughout the image, filling in spots that would otherwise have had no colour due to the shifted wavelength being either infrared or ultraviolet (Fig 5.1). This leads to a more complete and visually appealing image (Fig 5.2), but possibly misrepresents the spread of wavelengths that would be encountered by a physical observer.

In a simple path tracing application light and colour may be adequately represented as sRGB values, considered 'white' before being modified as a result of interactions with the materials of scene objects. The result is that spatially similar samples (e.g. two adjacent pixels) will tend to result in a roughly similar colour. In our prototype each ray is modelled as having a wavelength independent from the last. In our demonstration colours are converged upon by multiple independent samples of different wavelengths, instead of each ray being assigned a complete colour. Our representation of colour is therefore significantly less efficient in terms of colour information per-frame than a purely sRGB-based approach.

As shown by Acoleyen and Doorsselaere (2018) and Sherin et al. (2016), image-space techniques can be effectively used to model the Doppler and searchlight effects. A hybrid approach can be imagined wherein depth and object surface information is computed by path tracing, with the final transformation to colour overlaid on top via image-space techniques. This would produce cleaner images and allow them to converge more quickly, but the potential for intrinsic modelling of the properties of light such as shadows would be lost.

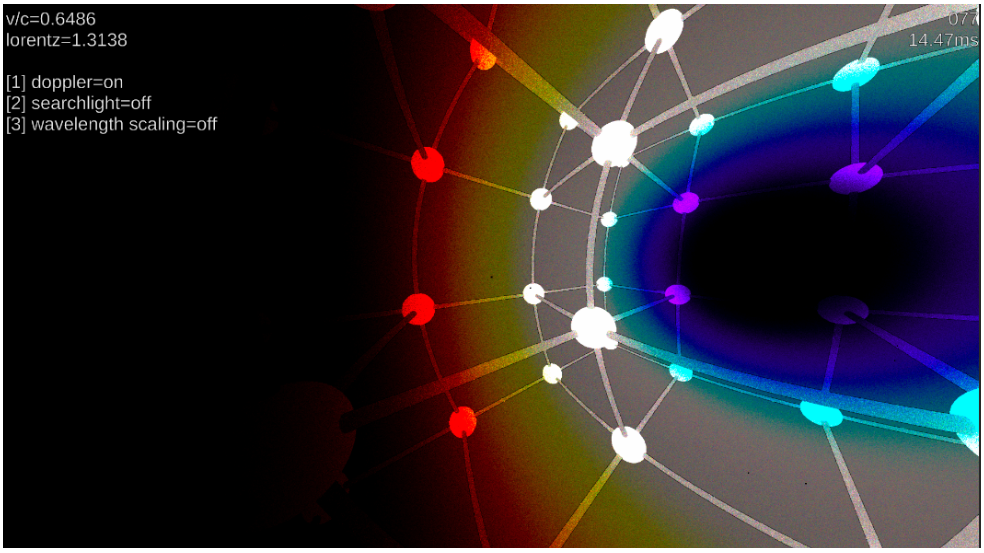

Figure 5.1: All rays are created in the visible spectrum. Dark regions have been blue- or red- shifted out of the visible spectrum.

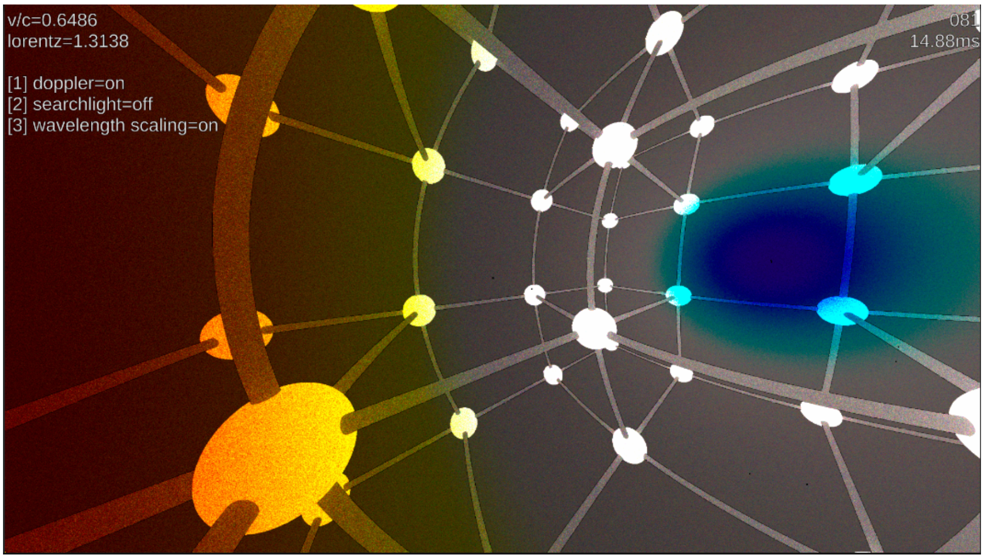

Figure 5.2: Rays are generated across the full range of now-visible wavelengths.

Chapter 6

Conclusion

In this chapter we review our findings and evaluate our implementation. We assess the limitations of our work, with respect to its accuracy in terms of the physics it represents, the visual clarity of communicating said physics and the usefulness of the produced visualisation. We finally suggest routes future work could take, based on our implementation.

6.1 Limitations

Our model of special relativistic optics is incomplete, due to the omission of the Lorentz contraction and the finite runtime of light. As a result, though the modulation of light ray properties with respect to direction of travel is correct, the distortion of visible scene objects is flawed. The general behaviour of the distortion is correct, as scene objects appear to move in front of the camera as it accelerates, but are not accurate. Temporal effects cannot be observed without an implementation of finite-speed light propogation.

Only spheres and cylinders are supported by our path-tracing implementation. Further, the prototype supports only a simple lattice structure which is generated by code. As such it is not capable of supporting generalised scenes.

The camera is limited to motion along its own forward axis with the lattice rotation around the camera to provide the illusion of the camera turning. This is because our implementation of relativistic aberration, which is aligned along world's global z-axis.

Our colour representation is incomplete. Individual rays are modelled as having particular wavelengths, which results in correct blue- or red-shifting as a consequence of the doppler effect. However, we have no represntation of the spectrum of scene objects. Instead, we use a naive approximate of reflectance based on the original RGB colour of the object. This results in all frequencies being equally reflected, so non-emissive objects in our visualisation all have a grey appearance.

Our implementation has no material representation. Only a simple diffuse ray-scattering model is implemented, which limits the types of optical effects we are able to simulate. For example, there is no way to observe the behaviour of specular highlights, reflective or refractive surfaces, although the flexibility of path-tracing as an image forming process supports them all.

6.2 Future Work

The incorrect distortion produced by our work could be resolved by correct implementation of the REST-frame technique (P. Hsiung and Dunn 1989) to relate rays between inertial reference frames and accelerated by use of a time buffer (P.-K. Hsiung, Thibadeau, and Wu 1990). A proper relation between simulated reference frames as in (Sherin et al. 2016) would be crucial to the development of a more general solution to relativistic visualisation.

Work could be done to increase the number of available primitives for use in a relativistic scene. Other geometric primitives could be implemented and combined, and support for polygonal meshes could be included with the integration of hardware acceleration such as RTX or DLSS 2.0 on NVIDIA's Turing architecture. In addition to the current depth pre-pass, additional pre-passes could be made to store per-pixel factors such as ϕ, ϕ', D and the value of the searchlight effect. This would reduce the amount of computation required by caching per-frame data, and could additionally be used to drive further image space effects. This would support the development of image-based post-processing effects, which are extensively used in contemporary graphics pipelines to cheaply improve the visual quality of rendered frames.

The image noise could be mitigated with additional path tracing acceleration techniques, while the existing steps could be improved by better integration with Unity's Scriptable Render Pipeline framework. Additionally, the path tracing process could make more efficient use of trace operations by utilising more of the information available in a G-buffer pre-pass. In particular, spatiotemporal reprojection techniques (Mäkitalo, Kivi, and Jääskeläinen 2020) to re-use samples from previous frames based on camera movement and spatiotemporal filtering are proven methods of cheaply improving path tracer performance. This would allow the accumulation of samples to happen while the camera is moving, as opposed to current instant invalidation. Spatiotemporal filtering would also allow for efficient use of previously traced samples.

Though we saw some success with accelerating our shading pass based on the depth buffer, use of additionally view-space information in the G-buffer would allow for further optimisation of the shading pass runtime. For example, the searchlight effect could be used to quickly evaluate low-intensity rays, which would cover the current weakness of the depth-dependent intersection skipping at high values of β. Scene geometry could also be culled to some extent, however omission of geometry may affect the physical accuracy of lighting effects, particularly reflection or shadows from out-of-frame objects.

Our colour and shading model can be improved significantly. Specularity, reflectivity and transparency including refraction could additionally be modelled with the use of a more sophisticated model such as Phong. This would also support a move to true material-based rendering, where the properties of a material are stored separately to the geometry data of the scene objects. Additional material types would allow for the exploration of more intricate optical effects, such as the appearance of reflective or refractive objects travelling at relativistic speeds. This development may be incompatible with particular acceleration methods (P.-K. Hsiung, Thibadeau, and Wu 1990).

Texture support would also improve the visualisation. As spheres still retain a circular silhouette under extreme relativistic distortion, a textured sphere would be more useful in visualising the more subtle distortion across the surface of the object (Penrose 1959). In combination with support for polygonal meshes, this would allow for the construction of more complex and realistic scenes within the visualisation. Additional texture data such as normal and emission maps could further inform the renderer.

The implementation of a material system would allow for the definition of full spectra of objects. With this, the difference in reflectivity of objects across a variety of wavelengths could be observed. Crucially, this would allow for the representation of light reflected in the infrared and ultraviolet bands but not in the visible, which may be of interest for particular scenarios.

Further integration with the Unity scene editor would increase the usefulness of our renderer. Instead of generating scene objects in script, they could be manipulated at edit-time using Unity's own scene editor, making for a more intuitive experience in formulating visualisations for physics educators without a background in graphics (Sherin et al. 2016). Moreover, additional engine integration would make available the possibility of creating path tracing acceleration structures through the use of the built-in physics system. This would represent a challenge in adapting acceleration structures developed for Euclidean space for use with relativistic rendering.

Though the visualisation of relativistic optics does itself have some application in physics education, a more complete simulation would integrate with a physics engine to allow objects besides the observer to move, as well as allow the user to meaningfully interact with the virtual world around them. Though this represents a significant challenge, this would be fundamental to the further development of the visualisation into an educational game, as demonstrated by (Sherin et al. 2013). Such a physics engine is likely to have a variety of constraints on how objects are able to move, which could be explored in further work.

Bibliography

[1] Rachel E. Scherr, Peter S. Shaffer, and Stamatis Vokos. "The challenge of changing deeply held student beliefs about the relativity of simultaneity". In: *American Journal of Physics* 70.12 (2002), pp. 1238–1248. DOI: 10.1119/1.1509420. eprint: https://doi.org/10.1119/1.1509420. URL: https://doi.org/10.1119/1.1509420.

[2] Tejinder Kaur et al. "Teaching Einsteinian physics at schools: part 1, models and analogies for relativity". In: *Physics Education* 52.6 (Sept. 2017), p. 065012. DOI: 10.1088/1361-6552/aa83e4. URL: https://doi.org/10.1088/1361-6552/aa83e4.

[3] Zachary Sherin et al. "Visualizing relativity: The OpenRelativity project". In: *American Journal of Physics* 84 (May 2016), pp. 369–374. DOI: 10.1119/1.4938057.

[4] C. M. Savage, A. Searle, and L. McCalman. "Real Time Relativity: Exploratory learning of special relativity". In: *American Journal of Physics* 74.8 (2006), pp. 790–798. DOI: 9.1119/1.2744048. eprint: https://doi.org/10.1119/1.2744048. URL: https://doi.org/9.1119/1.2744048.

[5] J. Burgess. "RTX on—The NVIDIA Turing GPU." In: *IEEE Micro, Micro, IEEE* 40.2 (2020), pp. 36–44. ISSN: 0272-1732. URL: https://ezproxy.derby.ac.uk/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.8981896&site=eds-live.

[6] Leonard A. Annetta. "Video Games in Education: Why They Should Be Used and How They Are Being Used". In: *Theory Into Practice* 47.3 (2008), pp. 229–239. ISSN: 00405841, 15430421. URL: http://www.jstor.org/stable/40071547.

[7] Meng-Tzu Cheng et al. "Is immersion of any value? Whether, and to what extent, game immersion experience during serious gaming affects science learning". In: *British Journal of Educational Technology* 48.2 (2017), pp. 246–263. DOI: https://doi.org/10.1111/bjet.12386. URL: https://bera-journals.onlinelibrary.wiley.com/doi/abs/10.1111/bjet.12386.

[8] James Terrell. "Invisibility of the Lorentz Contraction". In: *Phys. Rev.* 116 (4 Nov. 1959), pp. 1041–1045. DOI: 10.1103/PhysRev.116.1041. URL: https://link.aps.org/doi/10.1103/PhysRev.116.1041.

[9] Meng-chou Chang, Feipei Lai, and Wei-Chao Chen. "Image Shading Taking into Account Relativistic Effects". In: *ACM Trans. Graph.* 15 (Oct. 1996), pp. 265–300. DOI: 10.1145/234535.234537.

[10] Theo Hughes and Magdalena Kersting. "The invisibility of time dilation". In: *Physics Education* 56.2 (Jan. 2021), p. 025011. DOI: 10.1088/1361-6552/abce02. URL: https://doi.org/10.1088/1361-6552/abce02.

[11] Thomas Müller, Sebastian Grottel, and Daniel Weiskopf. "Special Relativistic Visualization by Local Ray Tracing". In: *IEEE transactions on visualization and computer graphics* 16 (Jan. 2011), pp. 1243–50. DOI: 10.1109/TVCG.2010.196.

[12] Karel Van Acoleyen and Jos Van Doorsselaere. *Captain Einstein: a VR experience of relativity.* 2018. arXiv: 1806.11085 [physics.pop-ph].

[13] P. Hsiung and R. H. P. Dunn. "Visualizing relativistic effects in spacetime". In: *Supercomputing '89:Proceedings of the 1989 ACM/IEEE Conference on Supercomputing.* 1989, pp. 597–606. DOI: 10.1145/76263.76331.

[14] Robert DiSalle. "Space and Time: Inertial Frames". In: *The Stanford Encyclopedia of Philosophy.* Ed. by Edward N. Zalta. Winter 2020. Metaphysics Research Lab, Stanford University, 2020.

[15] Daniel Weiskopf. "A Survey of Visualization Methods for Special Relativity." In: Jan. 2010, pp. 289–302.

[16] Daniel Weiskopf, Ute Kraus, and Hanns Ruder. "Searchlight and Doppler Effects in the Visualization of Special Relativity: A Corrected Derivation of the Transformation of Radiance". In: *ACM Transactions on Graphics* 18 (Dec. 2000). DOI: 10.1145/336414.336459.

[17] Roland Haas et al. *The Einstein Toolkit.* Version The "DeWitt-Morette" release, ET_2020_11. Nov. 2020. DOI: 10.5281/zenodo.4298887. URL: https://doi.org/10.5281/zenodo.4298887.

[18] George Gamow. *Mr Tompkins in Wonderland.* Cambridge University Press, 1939.

[19] Oliver James et al. "Gravitational lensing by spinning black holes in astrophysics, and in the movie Interstellar". In: *Classical and Quantum Gravity* 32.6 (Feb. 2015), p. 065001. ISSN: 1361-6382. DOI: 10.1088/0264-9381/32/6/065001. URL: http://dx.doi.org/10.1088/0264-9381/32/6/065001.

[20] Anton Lampa. “Wie erscheint nach der Relativitätstheorie ein bewegter Stab einem ruhenden Beobachter?” In: *Zeitschrift fur Physik* 27.1 (Dec. 1924), pp. 138–148. DOI: 10.1007/BF01328021.

[21] R. Penrose. “The apparent shape of a relativistically moving sphere”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 55.1 (1959), pp. 137–139. DOI: 10.1017/S0305004100033776.

[22] Zachary Sherin et al. “A Slower Speed of Light: Developing Intuition about Special Relativity with Games”. In: *MIT Game Lab* (2013).

[23] E.L. Nichols and W.S. Franklin. *The Elements of Physics: A College Text-book*. The Elements of Physics: A College Text-book v. 3. Macmillan, 1903. URL: https://books.google.co.uk/books?id=WK0IAAAAIAAJ.

[24] Daniel Weiskopf, Daniel Kobras, and Hanns Ruder. “Real-world relativity: Image-based special relativistic visualization”. In: Oct. 2000, pp. 303–310. DOI: 10.1109/VISUAL.2000.885709.

[25] James T. Kajiya. “The rendering equation”. In: *Computer Graphics*. 1986, pp. 143–150.

[26] Fred E. Nicodemus. “Directional Reflectance and Emissivity of an Opaque Surface”. In: *Appl. Opt.* 4.7 (July 1965), pp. 767–775. DOI: 10.1364/AO.4.000767. URL: http://ao.osa.org/abstract.cfm?URI=ao-4-7-767.

[27] Inigo Quilez. *distance functions*. Accessed: 2021-04-02. URL: https://iquilezles.org/www/articles/distfunctions/distfunctions.htm.

[28] Andrew Howard, Sandy Dance, and Les Kitchen. “Relativistic Ray-Tracing: Simulating the Visual Appearance of Rapidly Moving Objects”. In: (Oct. 1995).

[29] Ping-Kang Hsiung, Robert H. Thibadeau, and Michael Wu. “T-Buffer: Fast Visualization of Relativistic Effects in Space-Time”. In: *Proceedings of the 1990 Symposium on Interactive 3D Graphics*. I3D ’90. Snowbird, Utah, USA: Association for Computing Machinery, 1990, pp. 83–88. ISBN: 0897913515. DOI: 10.1145/91385.91423. URL: https://doi.org/10.1145/91385.91423.

[30] R. Resnick. *Introduction to Special Relativity*. Wiley, 1968. ISBN: 9780471717249. URL: https://books.google.co.uk/books?id=fsIRAQAAIAAJ.

[31] Johannes Courtial et al. “Photography at relativistic speeds”. In: Sept. 2016, p. 994808. DOI: 10.1117/12.2237850.

[32] Alain Galvan. *Frame Analysis - Minecraft RTX Beta*. Accessed: 2021-04-13. May 2020. URL: https://alain.xyz/blog/frame-analysis-minecraftrtx.

[33] Markku Mäkitalo, Petrus Kivi, and Pekka Jääskeläinen. “Systematic Evaluation of the Quality Benefits of Spatiotemporal Sample Reprojection in Real-Time Stereoscopic Path Tracing”. In: *IEEE Access* PP (July 2020), pp. 1–1. DOI: 10.1109/ACCESS.2020.3010452.

[34] Christoph Schied et al. “Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination”. In: *Proceedings of High Performance Graphics*. New York, NY, USA: Association for Computing Machinery, 2017. ISBN: 9781450351010. DOI: 10.1145/3105762.3105770. URL: https://doi.org/10.1145/3105762.3105770.

[35] Alan Wolfe. *Generating Blue Noise Sample Points With Mitchell’s Best Candidate Algorithm*. Accessed: 2021-04-30. 2017. URL: https://blog.demofox.org/2017/10/20/generating-blue-noise-sample-points-with-mitchells-best-candidate-algorithm/.

[36] Declan Russell. “An exploration of improving sampling within Monte Carlo ray tracing using adaptive blue noise.” Mar. 2019. URL: http://eprints.bournemouth.ac.uk/32104/.

[37] Anjul Patney Andrew Edelsten Paula Jukarainen. “Truly Next-Gen: Adding Deep Learning to Games & Graphics (Presented by NVIDIA)”. GDC. 2019. URL: https://www.gdcvault.com/play/1026184/Truly-Next-Gen-Adding-Deep.

[38] Lei Xiao et al. “Neural Supersampling for Real-Time Rendering”. In: *ACM Trans. Graph.* 39.4 (July 2020). ISSN: 0730-0301. DOI: 10.1145/3386569.3392376. URL: https://doi.org/10.1145/3386569.3392376.

[39] R.W.G. Hunt and M. Pointer. “Measuring Colour: Fourth Edition”. In: *Measuring Colour: Fourth Edition* (July 2011). DOI: 10.1002/9781119975595.

[40] Cat Flynn. *srpt*. Source code repository. Accessed: 2021-05-07. 2021. URL: https://github.com/ktyldev/srpt.

[41] Aleksandar Gjurchinovski. “Relativistic aberration of light as a corollary of the relativity of simultaneity”. In: *European Journal of Physics* 27.4 (May 2006), pp. 703–708. ISSN: 1361-6404. DOI: 10.1088/0143-0807/27/4/002. URL: http://dx.doi.org/10.1088/0143-0807/27/4/002.

[42] Albert B. Stewart. “The Discovery of Stellar Aberration”. In: *Scientific American* 210.3 (Mar. 1964), pp. 100–108. DOI: 10.1038/scientificamerican0364-100.

[43] John Walker. *Colour Rendering of Spectra*. Accessed: 2021-04-04. Apr. 1996. URL: http://www.fourmilab.ch/documents/specrend/.

[44] David Kuri. *GPU Ray Tracing in Unity - Part 1*. Accessed: 2021-04-05. May 2018. URL: http://three-eyed-games.com/2018/05/03/gpu-ray-tracing-in-unity-part-1/.

[45] John Walker. *C-ship: The Aberration of Light*. Website. Accessed: 2021-04-26. URL: https://www.fourmilab.ch/cship/aberration.html.

[46] T Smith and J Guild. “The C.I.E. colorimetric standards and their use”. In: *Transactions of the Optical Society* 33.3 (Jan. 1931), pp. 73–134. DOI: 10.1088/1475-4878/33/3/301. URL: https://doi.org/10.1088/1475-4878/33/3/301.