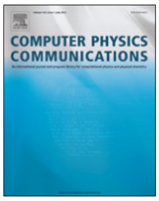
Computer Physics Communications 185 (2014) 1027–1037

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc



Dr TIM: Ray-tracer TIM, with additional specialist scientific capabilities[☆]

Stephen Oxburgh[a], Tomáš Tyc[b], Johannes Courtial[a,*]

[a] SUPA, School of Physics & Astronomy, University of Glasgow, Glasgow G12 8QQ, United Kingdom
[b] Institute of Theoretical Physics and Astrophysics, Masaryk University, Kotlarska 2, 61137 Brno, Czech Republic

ARTICLE INFO

Article history:
Received 7 June 2013
Received in revised form
21 October 2013
Accepted 28 October 2013
Available online 6 November 2013

Keywords:
Ray tracing
Lorentz transform
Geometrical optics
Perfect imaging
METATOYs

ABSTRACT

We describe several extensions to TIM, a raytracing program for ray-optics research. These include relativistic raytracing; simulation of the external appearance of Eaton lenses, Luneburg lenses and generalised focusing gradient-index lens (GGRIN) lenses, which are types of perfect imaging devices; raytracing through interfaces between spaces with different optical metrics; and refraction with generalised confocal lenslet arrays, which are particularly versatile METATOYs.

Program summary

Program title: TIM

Catalogue identifier: AEKY_v2_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEKY_v2_0.html

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licencing provisions: GNU General Public License

No. of lines in distributed program, including test data, etc.: 106905

No. of bytes in distributed program, including test data, etc.: 6327715

Distribution format: tar.gz

Programming language: Java.

Computer: Any computer capable of running the Java Virtual Machine (JVM) 1.6.

Operating system: Any, developed under Mac OS X Version 10.6 and 10.8.3.

RAM: Typically 130 MB (interactive version running under Mac OS X Version 10.8.3)

Classification: 14, 18.

Catalogue identifier of previous version: AEKY_v1_0

Journal reference of previous version: Comput. Phys. Comm. 183(2012)711

External routines: JAMA [1] (source code included)

Does the new version supersede the previous version?: Yes

Nature of problem:

Visualisation of scenes that include scene objects that create wave-optically forbidden light-ray fields.

Solution method: Ray tracing.

Reasons for new version: Significant extension of the capabilities (see Summary of revisions), as demanded by our research.

Summary of revisions:

Added capabilities include the simulation of different types of camera moving at relativistic speeds relative to the scene; visualisation of the external appearance of generalised focusing gradient-index (GGRIN)

[☆] This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

[*] Corresponding author. Tel.: +44 141 330 6429.
E-mail address: johannes.courtial@glasgow.ac.uk (J. Courtial).

0010-4655/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cpc.2013.10.031

lenses, including Maxwell fisheye, Eaton and Luneburg lenses; calculation of refraction at the interface between spaces with different optical metrics; and handling of generalised confocal lenslet arrays (gCLAs), a new type of METATOY.

Unusual features:

Specifically designed to visualise wave-optically forbidden light-ray fields; can visualise ray trajectories and geometric optic transformations; can simulate photos taken with different types of camera moving at relativistic speeds, interfaces between spaces with different optical metrics, the view through METATOYs and generalised focusing gradient-index lenses; can create anaglyphs (for viewing with coloured "3D glasses"), HDMI-1.4a standard 3D images, and random-dot autostereograms of the scene; integrable into web pages.

Running time:

Problem-dependent; typically seconds for a simple scene.

References:

[1] JAMA: A Java Matrix Package, http://math.nist.gov/javanumerics/jama/

1. Introduction

TIM [1] is a ray tracer we developed as a tool for ray-optics research, specifically our work on "metamaterials for rays" (METATOYs), which are surfaces covered with miniaturised optical components/instruments such that the surface appears to change the direction of transmitted light rays according to very unusual generalised laws of refraction [2].

The images TIM creates try to be somewhat photo-realistic, but perfect photo-realism is less important to us than simplicity. Additionally, there is often no obvious unique way in which light behaves at many of TIM's optical elements. For example, most of TIM's generalised laws of refraction cannot be performed for at least some incident light-ray fields without compromising the integrity of the phase fronts in some way [3], usually by introducing arrays of optical vortices into the field [4]. The optical vortices could, in principle, be added at different positions in the field, and this lack of a unique realisation of these generalised laws of refraction means that there is no obvious way to derive the equivalent of Fresnel coefficients for these laws of refraction. In the interest of simplicity, we have chosen not to model Fresnel reflections at *any* surfaces, including those where the Fresnel coefficients could be calculated uniquely.

In addition to being a research tool, TIM is also an outreach tool: TIM can be compiled into an interactive Java applet (which can be embedded into web pages[1]) or Java application (which, after downloading, can be run on almost any computer system). Much of TIM's functionality, including all the extensions described here, can be accessed interactively. We have tried to design TIM such that it is fun to use and invites playful exploration.

We keep adding capabilities to TIM as demanded by our current research interests (which include the identification of generalised confocal lenslet arrays (gCLAs) [5] as approximations to perfectly imaging homogeneous sheets [6] and their application to realising transformation-optics devices [7]), or occasionally just because we think that a feature would increase TIM's appeal as an outreach tool. Here we describe a few of the extensions that we added since we first wrote about TIM [1]. Working in a University, it is only natural to us to recognise TIM's new specialist scientific capabilities by giving the new version the full title "Dr TIM". The remainder of this paper is about this new version of TIM.

This paper is structured as follows. In Section 2 we describe TIM's capabilities to model the effect of the scene moving relative to the camera with relativistic speed. Section 3 describes TIM's simplified raytracing through Maxwell-fisheye, Eaton, Luneburg and generalised focusing gradient-index lens (GGRIN) lenses. Section 4 outlines TIM's raytracing through the interfaces between spaces with different optical metrics. In Section 5 we explain TIM's raytracing through generalised confocal lenslet arrays (gCLAs), a type of METATOY that is becoming increasingly important in our research. Finally, in Section 6, we discuss a number of minor extensions to TIM before we conclude (Section 7). In a number of places throughout the paper we outline details, aimed at readers with some familiarity of the Java programming language, of the way in which particular tasks have been implemented in the Java code.

2. Relativistic ray tracing

There are a number of computer programs that visualise the effect of moving at relativistic speed (see, for example, Refs. [8–10]). One of these, *Real Time Relativity* [9], stands out as it allows the user to move, with relativistic speeds and interactively (in a game-like environment), through complex scenes, and as it can be freely downloaded and run on the most common computer systems.

We have added to TIM the capability to simulate a snapshot taken with a camera that is moving, with relativistic speed, through a scene of stationary objects. The moment in simulated time when the snapshot is taken can be varied. This new capability can be combined with several of TIM's other capabilities, such as simulating a camera with a finite-size aperture and creating anaglyphs for viewing with red–blue 3D goggles.

We simulate a camera moving, with constant velocity $\boldsymbol{v}$, through a scene of stationary objects. We only consider the change in the position in which objects are seen, an effect known as relativistic aberration; we neglect all other effects, specifically the Doppler effect (which alters colour non-isotropically) and the headlight effect (which alters the brightness non-isotropically) [9].

2.1. Calculation of relativistic aberration

We calculate the relativistic aberration by broadly following the approach taken in Ref. [8].

When a camera takes a photo, it briefly opens its shutter, allowing light rays from the scene to enter the camera body and to hit the detector. Raytracing traces these rays backwards, starting from the camera and into the scene. If the camera moves relative to the scene, then each backwards-traced light ray needs to be transformed correspondingly after it leaves the camera and enters the scene. Therefore there are two relevant frames of reference: the *camera frame*, in which the camera is at rest, and the *scene frame*

[1] See, for example, http://tinyurl.com/timray.

in which everything else is at rest. In the scene frame, the camera frame is moving with velocity $\boldsymbol{v}$. We denote space–time coordinates in the camera frame as unprimed and those in the scene frame as primed. At time $t = 0$, the origins of both frames coincide. The simplest way to understand the transformation of light rays between the frames is to consider, for each light ray, two events consisting of the light ray passing through two positions on its trajectory and to transform these between the different frames. As both frames are inertial, light rays travel in straight lines through those positions in both frames.

Because we are tracing rays backwards, from the camera into the scene, we start in the camera frame. We do not actually explicitly transform two events in the ray's life. Instead, we transform one event, and we transform the direction of the backwards-traced ray (which points in the opposite direction of the physical light-ray direction).

In the camera frame, the event happens at position $\boldsymbol{x}$ and at time t. If the camera is moving with velocity $\boldsymbol{v}$ in the scene frame, we can calculate the event's space–time coordinates in the scene frame by applying the Lorentz transformation [8]

$$\boldsymbol{x}' = \boldsymbol{x} + (\gamma - 1)\frac{(\boldsymbol{\beta}\cdot\boldsymbol{x})\boldsymbol{\beta}}{\beta^2} + \gamma\boldsymbol{\beta}ct, \qquad ct' = \gamma ct + \gamma(\boldsymbol{\beta}\cdot\boldsymbol{x}), \quad (1)$$

where $\boldsymbol{\beta} = \boldsymbol{v}/c$ and $\gamma = 1/\sqrt{1-\beta^2}$ is the Lorentz factor. Note that we do not need to calculate the time of the event in the scene frame, as we assume that the scene is not changing. We can transform the direction $\boldsymbol{d}$ of the backwards-traced ray by considering two events, namely the positions $\boldsymbol{x}_1$ and $\boldsymbol{x}_2 = \boldsymbol{x}_1 - \boldsymbol{d}$ on its trajectory and the times t_1 and $t_2 = t_1 + \|d\|/c$ the ray passes through these (note the different signs in the expressions for $\boldsymbol{x}_2$ and t_2). Lorentz transforming the positions of these events (according to Eq. (1)) gives

$$\boldsymbol{x}_1' = \boldsymbol{x}_1 + (\gamma - 1)\frac{(\boldsymbol{\beta}\cdot\boldsymbol{x}_1)\boldsymbol{\beta}}{\beta^2} + \gamma\boldsymbol{\beta}ct_1, \quad (2)$$

$$\boldsymbol{x}_2' = \boldsymbol{x}_2 + (\gamma - 1)\frac{(\boldsymbol{\beta}\cdot\boldsymbol{x}_2)\boldsymbol{\beta}}{\beta^2} + \gamma\boldsymbol{\beta}ct_2 = \boldsymbol{x}_1' - \boldsymbol{d}', \quad (3)$$

where

$$\boldsymbol{d}' = \boldsymbol{d} + (\gamma - 1)\frac{(\boldsymbol{\beta}\cdot\boldsymbol{d})\boldsymbol{\beta}}{\beta^2} - \gamma\boldsymbol{\beta}\|d\| \quad (4)$$

is the direction of the backwards-traced ray in the scene frame.

We choose as the event the light ray leaving the last object it encounters that is stationary in the camera frame. We call the collection of such objects the *camera-frame scene*[2]. Allowing some objects to be stationary in the camera frame allows us to simulate, for example, the view through windows mounted to the front of the moving camera. TIM's backwards raytracing proceeds by each ray leaving the entrance-pupil disk and interacting with any objects in the camera-frame scene, before being transformed into the scene frame and interacting with any objects in the scene. The underlying assumption is that the objects in the camera frame are closer to the camera than any objects in the scene frame, and so the rendering only works correctly if the camera-frame scene is populated accordingly and the shutter timing is selected appropriately (as the shutter time determines the position of the shutter surface at the time the shutter opens).

It is clear from Eq. (1) that the time of the ray leaving the camera frame has an effect on the position of the same event in the scene frame, and this time is determined by the shutter model used. Each camera has a shutter somewhere within its body, in modern SLRs usually immediately in front of the film plane ("focal-plane shutter"). TIM's shutter model allows a choice of three *shutter surfaces* in which the shutter can be placed (Fig. 1):

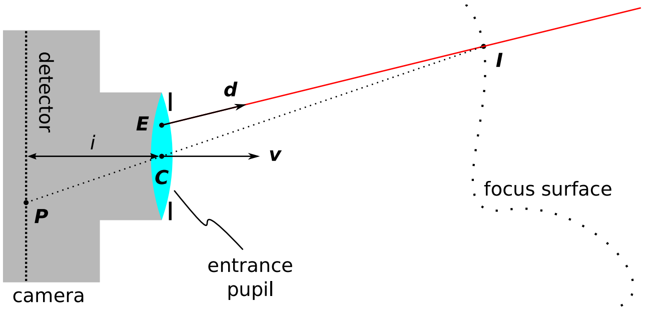


Fig. 1. Camera and shutter model in the camera frame. For each detector pixel, the camera traces one or more rays from different points on the disk of the entrance pupil to the image of that detector pixel, $\boldsymbol{I}$. If the light ray passes through the entrance-pupil disk at the point $\boldsymbol{E}$, then the direction of the backwards-traced ray is $\boldsymbol{d} \propto \boldsymbol{I} - \boldsymbol{E}$. The camera's entrance-pupil disk is defined by its centre point $\boldsymbol{C}$, the view direction $\boldsymbol{v}$, which is perpendicular to the disk, and the disk radius (not shown).

1. the *detector plane*, which is perpendicular to the view direction $\boldsymbol{v}$ and positioned a distance i behind the entrance pupil;
2. the *entrance-pupil disk*, positioned immediately in front of the lens (or generalised focusing element; note that TIM can focus on non-planar surfaces [1]);
3. the *focus surface*, the surface on which the images (formed by the focusing element) of the detector pixels are located.

The shutter model assumes that the shutter opens for an instant, at time t_s, simultaneously (in the camera frame) across the entire shutter surface.

Irrespective of which shutter model is selected, each backwards-traced ray that contributes towards a particular detector pixel originates from a point $\boldsymbol{E}$ on the entrance-pupil disk in the direction of the position $\boldsymbol{I}$ of the image of the detector pixel (see Fig. 1). In the entrance-pupil shutter model, the ray originates from $\boldsymbol{E}$ at time t_s. In the focus-surface shutter model, the ray originates from $\boldsymbol{E}$ at a time $t_s - \|\boldsymbol{I} - \boldsymbol{E}\|/c$ so that it passes through the position $\boldsymbol{I}$ – which lies on the focus surface – at time t_s.

The detector-plane shutter model is most complicated. It makes the following assumptions about the imaging element.

1. Any light ray through the centre of the entrance pupil, $\boldsymbol{C}$, is undeviated.
2. The optical path length from the position of a pixel, $\boldsymbol{P}$, to the position of its image, $\boldsymbol{I}$, is the same irrespective of the point on the entrance pupil the light ray passes through.
3. The optical path length from $\boldsymbol{P}$ to $\boldsymbol{I}$ equals $\|\boldsymbol{I} - \boldsymbol{P}\|$.

The first two assumptions are generalisations of the properties of an idealised lens; the third assumption is that the lens's central thickness is zero, which is unrealistic, but amongst the available arbitrary choices for the lens's central thickness arguably the simplest. The ray originates from $\boldsymbol{E}$ at such a time that it reaches $\boldsymbol{I}$ at the same time as the (undeviated) ray that leaves the pixel position $\boldsymbol{P}$ at time t_s and that reaches $\boldsymbol{I}$ via the aperture centre $\boldsymbol{C}$, which reaches $\boldsymbol{I}$ at $t_s + \|\boldsymbol{I} - \boldsymbol{P}\|/c$. As the ray from $\boldsymbol{E}$ takes a time $\|\boldsymbol{I} - \boldsymbol{E}\|/c$ to travel to $\boldsymbol{I}$, it has to leave $\boldsymbol{E}$ at time $t_s + (\|\boldsymbol{I} - \boldsymbol{P}\| - \|\boldsymbol{I} - \boldsymbol{E}\|)/c$. In line with most other implementations of relativistic ray tracing, TIM uses a value for the speed of light of $c = 1$.

2.2. Examples and discussion

Fig. 2 shows simulated photos taken with a rapidly moving pinhole camera, created with TIM. A number of well-known effects

[2] Note that the camera-frame scene is completely separate from the main, "scene-frame", scene and is also edited separately in TIM's interactive version. The camera-frame scene can be edited by selecting either the "Eye view" or "Anaglyph 3D" tab, clicking the "Edit view" button, selecting the "Relativistic effects" tab, and clicking the "Edit camera-frame scene" button.

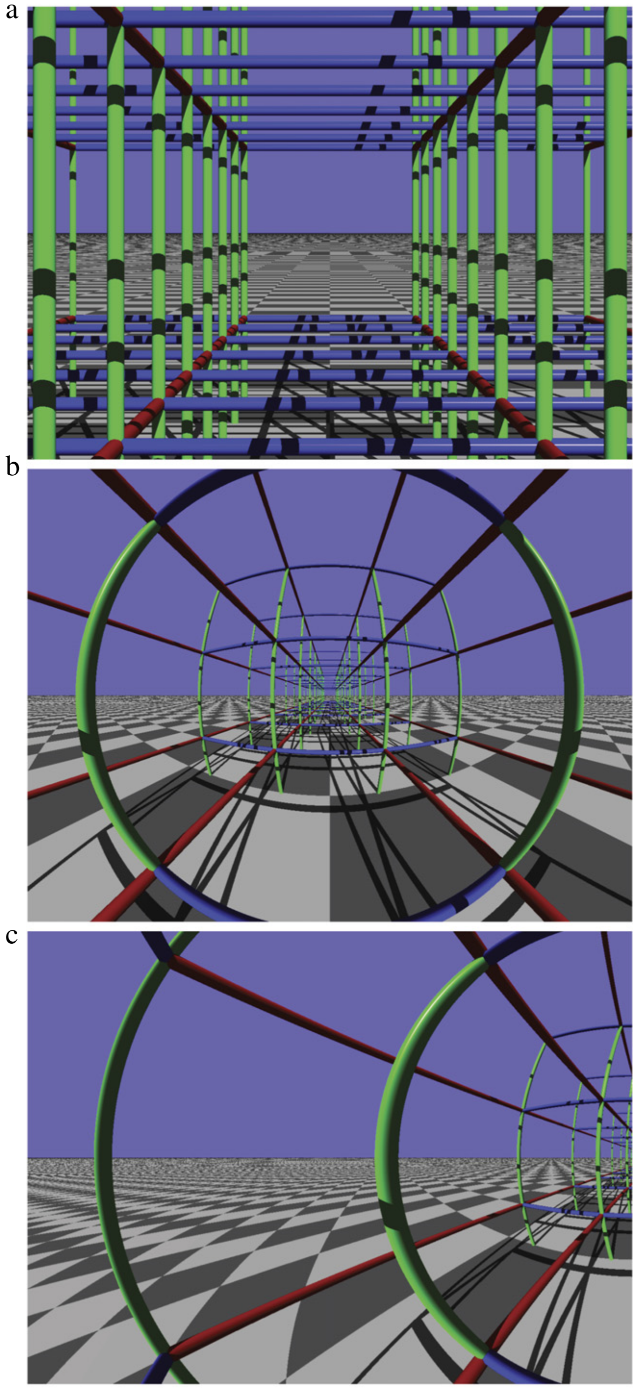


Fig. 2. Raytracing simulation of a photo of a cylinder lattice taken with a pinhole camera moving with velocity $\boldsymbol{v}$ relative to the scene. (a) $\boldsymbol{v} = (0, 0, 0)$; (b) $\boldsymbol{v} = (0, 0, 0.99c)$; (c) $\boldsymbol{v} = (0.2c, 0, 0.97c)$. The camera is pointing in the z direction.

of relativistic aberration are visible, such as Penrose's result that straight lines in one frame are seen as straight lines or circular arcs in the other [11].

The calculation of relativistic aberration applies not only to pinhole cameras, but also to other cameras implemented in TIM, and we have extended TIM accordingly. The resulting combination of capabilities is, to the best of our knowledge, new. First, TIM can simulate a camera with a finite-size circular aperture. TIM's simulated camera can focus on a transverse plane in front of the camera, like a standard camera, but it can also focus on almost arbitrary surfaces [1]. In the new, relativistic, TIM, this surface is interpreted to be in the camera frame, which is a natural generalisation of the fact that the focusing distance of a standard camera would also refer to the focusing distance in the camera frame. Fig. 3 shows an example of a simulated photo taken with such a camera. Second, TIM can now create analyphs of the scene as seen by a binocular observer moving with a relativistic speed. An example is shown in Fig. 4.

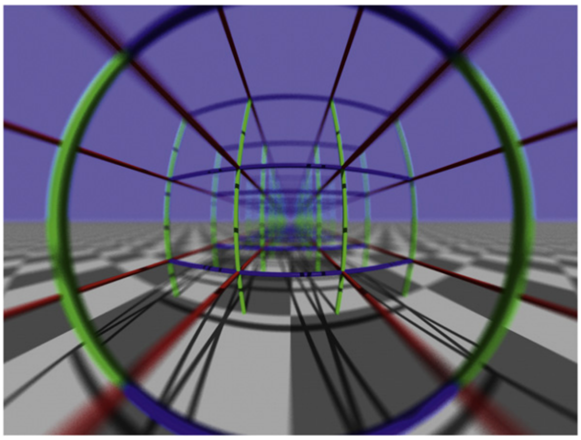


Fig. 3. Raytracing simulation of a photo of a cylinder lattice, taken with a camera moving with velocity $\boldsymbol{v} = (0, 0, 0.99c)$ relative to the scene and with a finite-size aperture. The camera is focused, in the camera frame, on a plane a distance 15 floor-tile lengths in front of the camera. The entrance-pupil shutter opened at time $t = 0$.

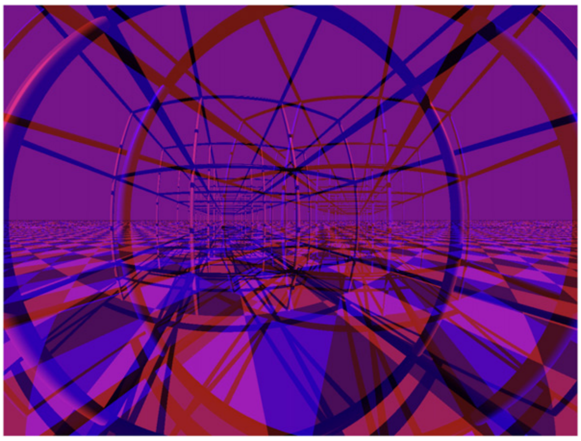


Fig. 4. Relativistic analyph. The image shows a raytracing simulation of a photo of a cylinder lattice taken with two cameras moving with velocity $\boldsymbol{v} = (0, 0, 0.99c)$ relative to the scene. The image recorded by one camera is shown in red, that recorded by the other camera is shown in blue, which makes the image suitable for viewing with red–blue analyph glasses. The image is simulated for an eye separation is 0.4 floor-tile lengths in the x direction (in the camera frame). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2.3. Implementation details

The Java classes describing the cameras outlined above are `RelativisticAnyFocusSurfaceCamera`, which is a subclass of `AnyFocusSurfaceCamera`, and `RelativisticAnaglyphCamera`, which is a modification of the (no longer existing) class `AnaglyphCamera`. All are part of the package `optics.raytrace.cameras`. For using in TIM's interactive version, there are also editable versions of these cameras, called `EditableRelativisticAnyFocusSurfaceCamera` and `EditableRelativisticAnaglyphCamera`, which are part of the `optics.raytrace.GUI.cameras` package.

Lorentz transforms are performed by the new class `LorentzTransform`, which is part of the `math` package.

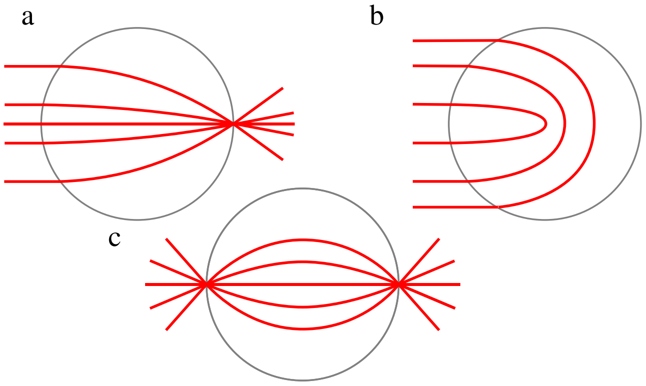


Fig. 5. Ray trajectories through a Luneburg lens (a), Eaton lens (b), and Maxwell-fisheye lens (c). The surface of each lens is shown as a circle. (After Ref. [12]).

In order to allow backwards raytracing through a camera-frame scene before Lorentz-transforming, it was necessary to keep track of time when tracing light rays. This required extensive changes throughout the code. Occasionally, when there was no unique (and easy to implement) way to calculate the time taken to traverse a specific optical component, TIM simply uses the time taken to traverse the equivalent length of vacuum.

3. Maxwell fisheye, Eaton, Luneburg, and generalised focusing gradient-index (GGRIN) lenses

Recent work on perfect imaging has revived interest in refractive-index distributions with ray-optically perfect imaging properties. The Luneburg lens [13] is a sphere whose inside has a spherically symmetric refractive-index distribution which images any parallel ray bundle incident on the sphere onto a point on the opposite side (Fig. 5(a)). A slightly different, but still spherically symmetric, refractive-index distribution results in the Eaton lens [14], which is a perfect retro-reflector (Fig. 5(b)). A Maxwell fisheye [15] is, in principle, a spherically symmetric refractive-index distribution of infinite radius. It is famous not only for being the first of these lenses, but also for perfectly imaging not just ray-optically, but also wave-optically [16]. Following Ref. [12], we consider here only a spherical central part of the Maxwell fisheye with its radius chosen such that any point on the outside of the sphere is imaged to the opposite side of the sphere (Fig. 5(c)). We refer to such a device as a Maxwell-fisheye lens. All of these, and other, "lenses", were recently combined into a more general class by Šarbort and Tyc [12]. We call those general lenses *generalised focusing gradient-index (GGRIN) lenses*.

We have extended TIM to be able to simulate the appearance of such lenses (Fig. 6). TIM does not perform detailed raytracing through the lenses. Instead, TIM uses equations for the position $\boldsymbol{P}'$ on the sphere (of radius R and centred at position $\boldsymbol{C}$) and direction $\boldsymbol{d}'$ after transmission through the lens of a light ray incident at position $\boldsymbol{P}$ with direction $\boldsymbol{d}$.

For the Luneburg lens, $\boldsymbol{P}'$ and $\boldsymbol{d}'$ can be calculated as follows. From Fig. 7(a) it can be seen that the outgoing ray leaves the lens from a position

$$\boldsymbol{P}' = \boldsymbol{C} + R\hat{\boldsymbol{d}} \tag{5}$$

in the (unnormalised) direction

$$\boldsymbol{d}' = \boldsymbol{R} = \boldsymbol{C} - \boldsymbol{P}. \tag{6}$$

For the calculation of the parameters of the ray leaving the Eaton lens, refer to Fig. 7(b). We first calculate the vector $\boldsymbol{R} = \boldsymbol{C} - \boldsymbol{P}$ again, and its component perpendicular to $\boldsymbol{d}$, $\boldsymbol{R} - \boldsymbol{R} \cdot \hat{\boldsymbol{d}}$, where $\hat{\boldsymbol{d}} = \boldsymbol{d}/\|\boldsymbol{d}\|$ is the normalised incident light-ray direction. Then the position $\boldsymbol{P}'$ where the outgoing ray leaves the lens is simply

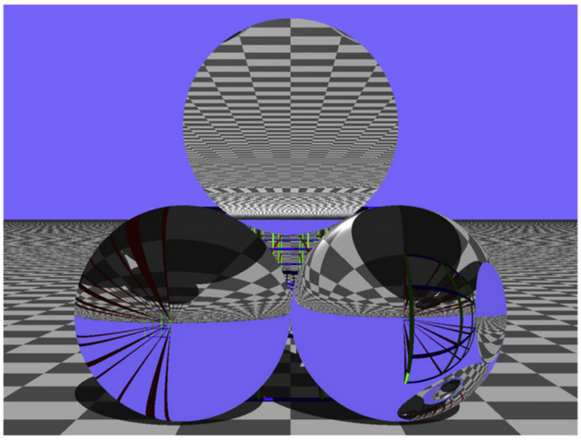


Fig. 6. Simulated view of a Luneburg lens (bottom left), Eaton lens (top), and Maxwell-fisheye lens (bottom right) in front of TIM's standard cylinder lattice. The lattice cannot be seen through the Eaton lens, as it is a retro-reflector.

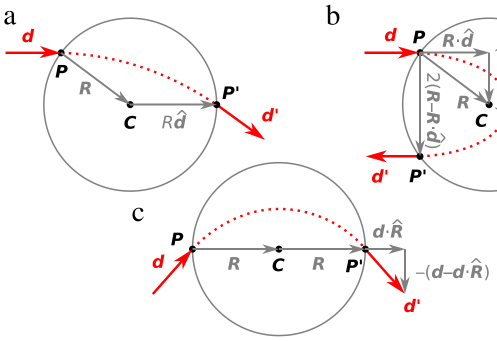


Fig. 7. Redirection of a light ray by a Luneburg lens (a), an Eaton lens (b), and a Maxwell-fisheye lens (c). A light ray hits the lens at $\boldsymbol{P}$ with direction $\boldsymbol{d}$, and leaves it from position $\boldsymbol{P}'$ with direction $\boldsymbol{d}'$.

$$\boldsymbol{P}' = \boldsymbol{P} + 2\left(\boldsymbol{R} - \boldsymbol{R} \cdot \hat{\boldsymbol{d}}\right), \tag{7}$$

and the direction of the outgoing ray is

$$\boldsymbol{d}' = -\boldsymbol{d}. \tag{8}$$

For the Maxwell-fisheye lens (see Fig. 7(c)), we calculate the position from which the ray leaves the lens as

$$\boldsymbol{P}' = \boldsymbol{C} + \boldsymbol{R}, \tag{9}$$

where $\boldsymbol{R} = \boldsymbol{C} - \boldsymbol{P}$ as before, and its direction as the incident light-ray direction, but with the component perpendicular to $\boldsymbol{R}$ reversed:

$$\boldsymbol{d}' = \boldsymbol{d} \cdot \hat{\boldsymbol{R}} - \left(\boldsymbol{d} - \boldsymbol{d} \cdot \hat{\boldsymbol{R}}\right) = 2\boldsymbol{d} \cdot \hat{\boldsymbol{R}} - \boldsymbol{d}. \tag{10}$$

3.1. GGRIN lens

All the lenses discussed above, and more, can be viewed as special cases of a more general GGRIN lens [12]. The GGRIN lens is spherically symmetric, and so any ray incident on the lens traverses it along a trajectory that lies on the ray's plane of incidence. Fig. 8 shows a schematic diagram of an arbitrary ray in its plane of incidence. If the lens is centred at $\boldsymbol{C}$, and the ray hits the lens at position $\boldsymbol{P}$ with direction $\boldsymbol{d}$, the plane of incidence includes the positions $\boldsymbol{C}$ and $\boldsymbol{P}$ and the direction vector $\boldsymbol{d}$. The lens can then be characterised by two distances, r and r', and an angle, φ, as follows. Before hitting

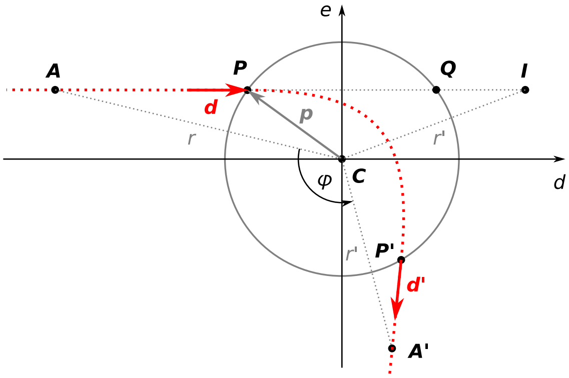


Fig. 8. Geometry of the effect of the GGRIN lens [12]. The entire ray trajectory lies within the ray's plane of incidence.

Table 1
Parameter combinations for which a GGRIN lens is equivalent to other well-known lenses. R is the radius of the lens.

Lens	r	r'	φ
Luneburg	∞	R	180°
Eaton	∞	∞	0°
Maxwell fisheye	R	R	180°

the lens, the (straight-line) ray trajectory passes through a position $\boldsymbol{A}$ a distance r from the lens centre, $\boldsymbol{C}$. After transmission through the lens, the ray will pass through a position $\boldsymbol{A}'$ a distance r' from $\boldsymbol{C}$ such that the distance between the ray's straight-line continuation and the lens centre, $\boldsymbol{C}$, remains unchanged (angular-momentum conservation). $\boldsymbol{A}'$ is positioned such that the straight lines $\boldsymbol{AC}$ and $\boldsymbol{CA}'$ intersect at an angle φ. Table 1 lists the values of the parameters r, r' and φ for which the GGRIN lens is equivalent to the lenses discussed above. (Note that the lens with $r = r' = \infty$ and values of φ other than 0 and 180° has an effect similar to that of an axicon [17], which can be used to turn plane waves into Bessel-like light beams [18]. Unlike axicons, the GGRIN lens should work for plane waves incident from any direction.)

We calculate the position $\boldsymbol{P}'$ where the transmitted ray leaves the lens and its direction $\boldsymbol{d}'$ as follows. First, we define orthogonal unit vectors that span the ray's plane of incidence. We take the normalised incident light-ray direction, $\hat{\boldsymbol{d}} = \boldsymbol{d}/\|\boldsymbol{d}\|$, as the first unit vector. Then the component of the vector $\boldsymbol{p} = \boldsymbol{P} - \boldsymbol{C}$ in the direction of $\hat{\boldsymbol{d}}$, i.e. the d component of $\boldsymbol{p}$, is $p_d = \hat{\boldsymbol{d}} \cdot \boldsymbol{p}$. We define the second unit vector, $\hat{\boldsymbol{e}}$, such that the e component of $\boldsymbol{p}$ is positive and $\hat{\boldsymbol{d}} \cdot \hat{\boldsymbol{e}} = 0$. The part of $\boldsymbol{p}$ in the e direction is $\boldsymbol{p} - \hat{\boldsymbol{d}} p_d$, and so we define

$$\hat{\boldsymbol{e}} = \frac{\boldsymbol{p} - \hat{\boldsymbol{d}} p_d}{|\boldsymbol{p} - \hat{\boldsymbol{d}} p_d|}. \qquad (11)$$

The e component of $\boldsymbol{p}$ is then simply $p_e = \hat{\boldsymbol{e}} \cdot \boldsymbol{p}$.

We can now calculate the d and e components of the vector $\boldsymbol{a} = \boldsymbol{A} - \boldsymbol{C}$. As $\boldsymbol{A}$ lies on the trajectory of the incident ray, $a_e = p_e$. The d component can be calculated from the distance requirement, i.e. $a_d^2 + a_e^2 = r^2$, which gives $a_d = -\sqrt{r^2 - a_e^2}$ (the negative sign is chosen so that $\boldsymbol{A}$ actually lies on the incident light-ray trajectory).

We now consider a hypothetical ray that passes through the lens undeviated. We define $\boldsymbol{Q}$ to be the position on the surface of the lens where this ray would re-emerge, and we define $\boldsymbol{I}$ to be the position where the ray's distance from $\boldsymbol{C}$ is r'. The ray has most of the properties of the refracted ray: the ray passes through a position, $\boldsymbol{I}$, a distance r' from $\boldsymbol{C}$, and its impact parameter (the distance between its straight-line continuation and $\boldsymbol{C}$) is the same as that of the incident ray (because they share the same straight-line continuation and therefore impact parameter); all that is wrong is that, in general, the straight lines $\boldsymbol{AC}$ and $\boldsymbol{CI}$ do not intersect at the desired angle φ. We define $\boldsymbol{q} = \boldsymbol{Q} - \boldsymbol{C}$ and $\boldsymbol{i} = \boldsymbol{I} - \boldsymbol{C}$, whose components are

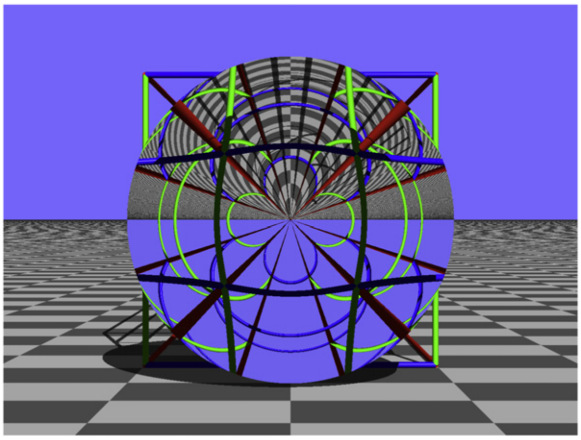


Fig. 9. Simulated view of TIM's standard cylinder lattice seen through an example of a GGRIN lens. The lens parameters are $r = r' = 100{,}000$ ($\approx\infty$), $\varphi = 170°$.

$$q_e = i_e = p_e, \quad q_d = -p_d, \quad i_d = \sqrt{r'^2 - i_e^2}. \qquad (12)$$

The point $\boldsymbol{I}$ has all the properties of the point $\boldsymbol{A}'$, apart from its direction with respect to $\boldsymbol{C}$. To construct $\boldsymbol{A}'$, we need to rotate $\boldsymbol{I}$ in the plane around $\boldsymbol{C}$ by an angle

$$\beta = \arctan(a_d, a_e) - \arctan(i_d, i_e) + \varphi \qquad (13)$$

(where we have used the 2-argument arctan function, sometimes called atan2, that avoids quadrant ambiguity).

Similarly rotating $\boldsymbol{Q}$ gives the point $\boldsymbol{P}'$ where the transmitted ray leaves the lens, and rotating $\hat{\boldsymbol{d}}$ results in the transmitted ray's direction, $\hat{\boldsymbol{d}}'$. We do this by calculating the rotated basis vectors

$$\hat{\boldsymbol{d}}' = \hat{\boldsymbol{d}} \cos\beta + \hat{\boldsymbol{e}} \sin\beta, \quad \hat{\boldsymbol{e}}' = -\hat{\boldsymbol{d}} \sin\beta + \hat{\boldsymbol{e}} \cos\beta, \qquad (14)$$

and calculating the transmitted ray's start point according to

$$\boldsymbol{P}' = \boldsymbol{C} + q_d \hat{\boldsymbol{d}}' + q_e \hat{\boldsymbol{e}}'; \qquad (15)$$

the transmitted ray's direction is simply $\hat{\boldsymbol{d}}'$. Fig. 9 shows an example of the view through a GGRIN lens.

3.2. Combinations of Eaton and Luneburg lenses with an invisible sphere

In a further generalisation [19], Šarbort and Tyc considered multifocal versions of these lenses, formed by replacing the central part with another refractive-index distribution. We have also implemented here the ability to render the view through multifocal lenses formed by combining any of the lenses discussed above with an invisible sphere [20].

Fig. 10 shows a few ray trajectories through two of these multifocal lenses. What they have in common is that rays whose straight-line continuation gets closer to the lens centre than R_t emerge from the other side of the lens as if they have passed straight through. When seen from any direction, the lens appears to have an empty central tunnel of radius R_t in the current view direction. Fig. 11 shows an example.

3.3. Refractive-index ratio at the edge

All the above lenses can be designed such that the refractive index on the surface of the lens is the same as that of the surrounding medium. We have also added to TIM the possibility of simulating the view when this is not the case.

All this requires is a refraction step, from the surrounding medium into a material with the refractive index of the lens at the

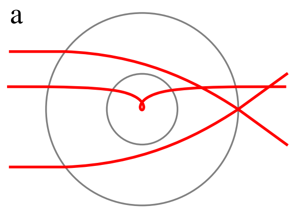


Fig. 10. Ray trajectories through combinations of a Luneburg lens with an invisible sphere (a) and an Eaton lens with an invisible sphere (b). The outer circle is the surface of each lens. The inner circle shows the boundary between the Luneburg or Eaton part of the lens (outside) and the invisible-sphere part (inside). (After Ref. [19]).

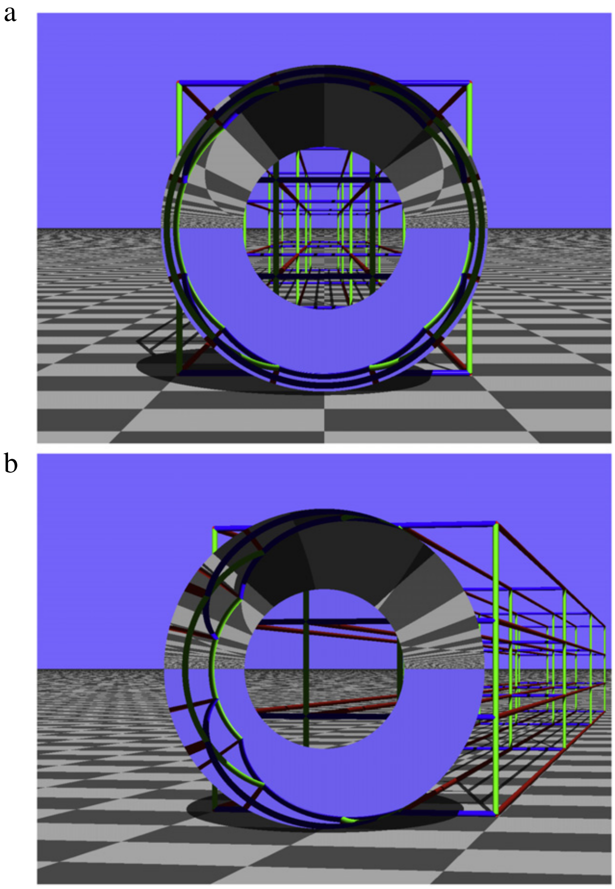


Fig. 11. View of a multifocal lens in front of TIM's standard cylinder lattice, seen from two view positions. The multifocal lens is a combination of a GGRIN lens (parameters $r = r' = 100{,}000\ (\approx\infty)$, $\varphi = 90^\circ$) and an invisible sphere. The appearance is that of a central empty tunnel in the view direction, irrespective of view direction. (a) Standard camera position, (0, 0, 0), and view direction, (0, 0, 1); (b) view position (2, 0, 0), view direction (−0.2, 0, 1).

surface, before the calculations outlined above and afterwards another refraction step, this time from a material with the lens's surface refractive index into the surrounding medium. As a respectable ray tracer, TIM is already equipped with refraction functionality, and so this is simply a matter of invoking the relevant method. Note that the actual values of the refractive indices at the surface and of the surrounding medium do not matter; all that matters is the value of their ratio, and so this is the parameter that can be set for any of the above lenses.

Fig. 12 shows an example of the view through a lens in which the surface refractive index does not match that of the surrounding medium.

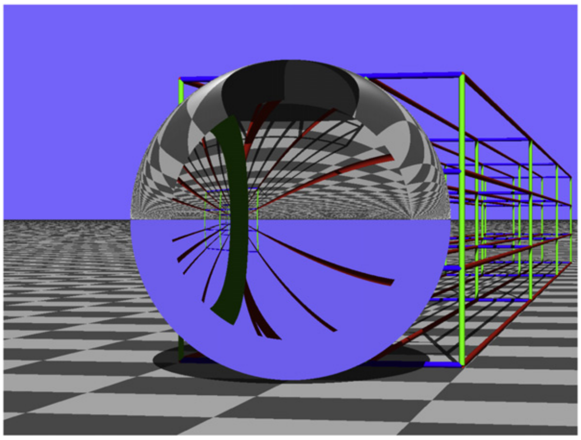


Fig. 12. View through the lens shown in Fig. 11(b), but with the ratio between the refractive indices at the lens's surface and of the surrounding medium being $n_{\text{surface}}/n_{\text{surrounding}} = 1.5$ (instead of 1). Note that the central part of the lens no longer looks like an empty tunnel.

3.4. Implementation details

The lenses described above are implemented as classes that extend the `Sphere` class of scene objects, specifically the classes `EatonLens`, `LuneburgLens`, `MaxwellFisheyeLens`, and `GGRINLens`.[3] The `EatonLens`, `LuneburgLens`, and `MaxwellFisheyeLens` classes calculate the direction of the refracted light ray and the position where it leaves the lens directly from Eqs. (5) to (10); they are not subclasses of the `GGRINLens` class. All these classes are a part of the package `optics.raytrace.sceneObjects`.

For use in TIM's interactive version, the above classes have been extended to be editable. The editable versions of the above classes are called `EditableEatonLens`, `EditableLuneburgLens`, `EditableMaxwellFisheyeLens`, and `EditableGGRINLens`, and all are part of the package `optics.raytrace.GUI.sceneObjects`.

The calculation of the refraction actually happens in classes that describe surface properties, specifically extensions of the class `SurfacePropertyPrimitive`. The surface-property classes that describe the above lenses are called `EatonLensSurface`, `LuneburgLensSurface`, `MaxwellFisheyeLensSurface`, and `GGRINLensSurface`. All are part of the `optics.raytrace.surfaces` package.

4. Refraction at metric interfaces

4.1. Metrics, Fermat's principle, and refraction at metric interfaces

In transformation optics [21,22], the (optical) metric of a material measures the optical path length ds corresponding to an infinitesimal geometrical path of length dx, dy and dz in the x, y and z direction. Fermat's principle demands that the optical path length along light-ray trajectories is stationary and so creating a spatially-varying metric changes light-ray paths. This idea has been used to design invisibility cloaks [21,22].

In vacuum, the square of the optical path length is given by the Euclidean distance

$$ds^2 = dx^2 + dy^2 + dz^2. \tag{16}$$

[3] Other gradient-index spherical objects for which the relationship between incident and refracted light rays is known can easily be implemented similarly.

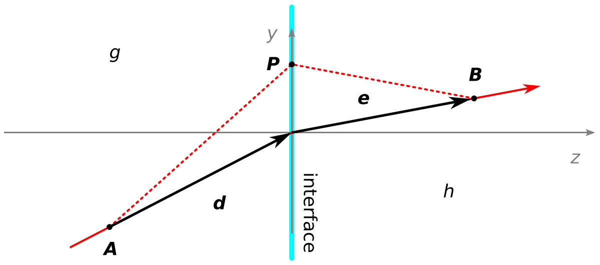


Fig. 13. Geometry of refraction at a metric-tensor interface. The coordinate system is chosen such that the interface is in the $z = 0$ plane and the incident light ray intersects the interface at the origin. The diagram is drawn for positive refraction.

In transformation-optics materials, the formula for the square of the optical path length takes the more general form

$$\begin{aligned} \mathrm{d}s^2 &= g_{11}\mathrm{d}x^2 + g_{22}\mathrm{d}y^2 + g_{33}\mathrm{d}z^2 \\ &\quad + 2g_{12}\mathrm{d}x\mathrm{d}y + 2g_{13}\mathrm{d}x\mathrm{d}z + 2g_{23}\mathrm{d}y\mathrm{d}z \\ &= \mathbf{dr}^T \cdot g \cdot \mathbf{dr}, \end{aligned} \tag{17}$$

where $\mathbf{dr} = (\mathrm{d}x, \mathrm{d}y, \mathrm{d}z)^T$ and g is the (symmetric) metric tensor

$$g = \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{12} & g_{22} & g_{23} \\ g_{13} & g_{23} & g_{33} \end{pmatrix}. \tag{18}$$

We stress that Eq. (17) specifies the *square* of the optical path length. To specify the optical path length itself therefore additionally requires knowledge of the sign of the optical path length; in a material with a negative refractive index, for example, it is negative [23].

We have added to TIM the ability to calculate the direction of a light ray after transmission through the interface between materials with different metrics. There are well-established methods to calculate this light-ray direction change [24]. TIM calculates the refracted light-ray direction directly from Fermat's principle. The coordinate system is placed such that the interface lies in the $z = 0$ plane and the incident light ray intersects the interface at the origin (Fig. 13). The half-space with $z < 0$ is described by the metric tensor g (with elements g_{ij}), the metric tensor for $z > 0$ is h (elements h_{ij}), and the incident light-ray direction is $\boldsymbol{d} = (d_x, d_y, d_z)^T$. The aim is to calculate the outgoing light-ray direction, which we write in the form $\boldsymbol{e} = (e_x, e_y, e_z)^T$.

We pick the two points $\boldsymbol{A} = -\boldsymbol{d}$ and $\boldsymbol{B} = \boldsymbol{e}$, which lie on either side of the interface. The optical path length from $\boldsymbol{A}$ to $\boldsymbol{B}$ via a point $\boldsymbol{P} = (x, y, 0)^T$ on the interface is then $s = s_1 + s_2$, where $s_1 = \pm\sqrt{(\boldsymbol{AP})^T \cdot g \cdot \boldsymbol{AP}}$ (where $\boldsymbol{AP} = \boldsymbol{P} - \boldsymbol{A}$) is the optical path length from $\boldsymbol{A}$ to $\boldsymbol{P}$, $s_2 = \pm\sqrt{(\boldsymbol{PB})^T \cdot h \cdot \boldsymbol{PB}}$ (where $\boldsymbol{PB} = \boldsymbol{B} - \boldsymbol{P}$) is the optical path length from $\boldsymbol{P}$ to $\boldsymbol{B}$. The signs of s_1 and s_2 indicate whether or not these optical path lengths are positive or negative, of course. Fermat's principle states that, at $(x, y) = (0, 0)$ (i.e. at the origin, where the actual light ray intersects the interface),

$$\frac{\partial\sqrt{(\boldsymbol{AP})^T \cdot g \cdot \boldsymbol{AP}}}{\partial x} \pm \frac{\partial\sqrt{(\boldsymbol{PB})^T \cdot h \cdot \boldsymbol{PB}}}{\partial x} = 0, \tag{19}$$

$$\frac{\partial\sqrt{(\boldsymbol{AP})^T \cdot g \cdot \boldsymbol{AP}}}{\partial y} \pm \frac{\partial\sqrt{(\boldsymbol{PB})^T \cdot h \cdot \boldsymbol{PB}}}{\partial y} = 0, \tag{20}$$

where the '+' signs must be chosen if s_1 and s_2 have the same signs (positive refraction), and the '−' signs must be chosen if s_1 and s_2 have the opposite sign (negative refraction). Allowing negative refraction enables raytracing through many interesting interfaces, for example those representing interfaces between media with different signs of the refractive index such as a Veselago lens [25,23]. To characterise a metric interface fully, we therefore state the metric tensors on both sides and whether the refraction is positive or negative. Evaluating the first term of Eq. (19) gives

$$\frac{\partial\sqrt{(\boldsymbol{AP})^T \cdot g \cdot \boldsymbol{AP}}}{\partial x} = \frac{d_x g_{11} + d_y g_{12} + d_z g_{13}}{\sqrt{\boldsymbol{d}^T \cdot g \cdot \boldsymbol{d}}}. \tag{21}$$

We define the number

$$c_x = \pm\frac{d_x g_{11} + d_y g_{12} + d_z g_{13}}{\sqrt{|\boldsymbol{d}^T \cdot g \cdot \boldsymbol{d}|}}, \tag{22}$$

with the '+' sign in the case of positive refraction and the '−' sign for negative refraction. Assuming, for the moment, that the term under the square root on the right-hand side of Eq. (21) is positive, c_x represents the first term of Eq. (19). Note that c_x depends only on the direction of the incident light ray and the material in which it travels. In terms of c_x, Eq. (19) becomes

$$c_x = \frac{\partial\sqrt{(\boldsymbol{PB})^T \cdot h \cdot \boldsymbol{PB}}}{\partial x} = \frac{e_x h_{11} + e_y h_{12} + e_z h_{13}}{\sqrt{\boldsymbol{e}^T \cdot h \cdot \boldsymbol{e}}}. \tag{23}$$

Next, we normalise the vector $\boldsymbol{e}$ with respect to the metric h such that

$$\boldsymbol{e}^T \cdot h \cdot \boldsymbol{e} = 1, \tag{24}$$

so that, on the right-hand side of Eq. (23), the denominator becomes one (thereby also ensuring that the expression under the square root is positive). Eq. (23) then simplifies to

$$c_x = e_x h_{11} + e_y h_{12} + e_z h_{13}. \tag{25}$$

Similarly we get from Eq. (20)

$$c_y = e_x h_{12} + e_y h_{22} + e_z h_{23}. \tag{26}$$

Together, Eqs. (24)–(26) determine the direction $\boldsymbol{e}$ of the refracted light ray.

We solve these equations as follows. First we solve Eqs. (25) and (26) for e_x and e_y to find

$$e_x = \frac{c_x h_{22} - c_y h_{12} - e_z h_{13} h_{22} + e_z h_{12} h_{23}}{h_{11} h_{22} - h_{12}^2}, \tag{27}$$

$$e_y = \frac{c_y h_{11} - c_x h_{12} - e_z h_{23} h_{11} + e_z h_{12} h_{13}}{h_{11} h_{22} - h_{12}^2}. \tag{28}$$

Substitution into Eq. (24) gives, after some manipulation (which assumes that $h_{12}^2 - h_{11} h_{22} \neq 0$),

$$a e_z^2 + c = 0 \tag{29}$$

where

$$a = h_{13}^2 h_{22} - 2h_{12} h_{13} h_{23} + h_{12}^2 h_{33} + h_{11} h_{23}^2 - h_{11} h_{22} h_{33}, \tag{30}$$

$$c = -c_y^2 h_{11} + 2c_x c_y h_{12} - h_{12}^2 - c_x^2 h_{22} + h_{11} h_{22}. \tag{31}$$

The real solutions, which exist provided that $-c/a \geq 0$, are

$$e_z = \pm\sqrt{-\frac{c}{a}}. \tag{32}$$

The correct solution is that which has the same sign as d_z, such that the incident ray hits one side of the interface and the refracted ray leaves from the other side.

If $-c/a < 0$, no real solution exists. This is a generalisation of the situation when total internal reflection (TIR) occurs at refractive-index interfaces, and so in this case TIM assumes that TIR occurs.

4.2. Imaginary optical path lengths

If the expression under the square root in Eq. (22), $\boldsymbol{d}^T \cdot g \cdot \boldsymbol{d}$, is negative (which corresponds to the metric tensor not being positive-definite, like the Lorentzian metric), it is sometimes still possible to find values for e_x, e_y and e_z which are all real-valued and which solve Eq. (19). However, this solution corresponds to imaginary optical path lengths. It is not clear, what (if anything) it represents. We have chosen to program this solution into TIM, but give the user the choice whether or not to allow it. If the user does not allow it, and $\boldsymbol{d}^T \cdot g \cdot \boldsymbol{d} < 0$, TIM again assumes that TIR occurs.

After evaluating the derivatives (as above), Eq. (19) becomes

$$\pm \frac{d_x g_{11} + d_y g_{12} + d_z g_{13}}{\sqrt{\boldsymbol{d}^T \cdot g \cdot \boldsymbol{d}}} = \frac{e_x h_{11} + e_y h_{12} + e_z h_{13}}{\sqrt{\boldsymbol{e}^T \cdot h \cdot \boldsymbol{e}}}. \tag{33}$$

The expression under the square root on the left-hand side is negative, but if the expression under the square root on the right-hand side, $\boldsymbol{e}^T \cdot h \cdot \boldsymbol{e}$, is also negative, a common factor i can be cancelled and Eq. (23) becomes

$$\pm \frac{d_x g_{11} + d_y g_{12} + d_z g_{13}}{\sqrt{-(\boldsymbol{d}^T \cdot g \cdot \boldsymbol{d})}} = \frac{e_x h_{11} + e_y h_{12} + e_z h_{13}}{\sqrt{-(\boldsymbol{e}^T \cdot h \cdot \boldsymbol{e})}}. \tag{34}$$

Thanks to the absolute value under the square root in the denominator we introduced in the definition of c_x, Eq. (22), the left-hand side of Eq. (34), which depends only on the incident ray, equals c_x again. We again normalise $\boldsymbol{e}$ such that the term in the square root of the denominator on the side that depends only on the outgoing ray direction becomes one, which now requires

$$\boldsymbol{e}^T \cdot h \cdot \boldsymbol{e} = -1. \tag{35}$$

Eqs. (25) and (26) then still hold, and so e_x and e_y can still be calculated from Eqs. (27) and (28), and the normalisation equation can again be written in the form of Eq. (32), but with a slightly different coefficient c in which two terms have changed sign (compared to Eq. (31)):

$$c = -c_y^2 h_{11} + 2 c_x c_y h_{12} + h_{12}^2 - c_x^2 h_{22} - h_{11} h_{22}. \tag{36}$$

4.3. General coordinates

So far in this section we have assumed that the interface is in the $z = 0$ plane. This assumption greatly simplifies the calculation of the light-ray direction behind a metric interface. We now generalise these results to an arbitrary interface. We achieve this by defining a local surface coordinate system in which the tangent plane to the surface at the incident ray's intersection point is the equivalent of the $z = 0$ plane; calculating the incident light-ray direction and the metric tensors on both sides of the interface in this surface coordinate system; calculating the refracted light-ray direction in the surface coordinate system (according to the calculation outlined in Sections 4.1 and 4.2); and finally transforming the refracted light-ray direction into the global (x, y, z) coordinate system.

We call the unit base vectors of the local coordinate system $\hat{\boldsymbol{u}}$, $\hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{w}}$, and choose $\hat{\boldsymbol{w}}$ to be perpendicular to the surface at the intersection point. The local tangent plane is therefore the $w = 0$ plane, which from now on takes over the role played by the $z = 0$ plane in Sections 4.1 and 4.2. In TIM, such a coordinate system can easily be calculated for each point on a surface (see Fig. 7 in Ref. [1]).

The calculation of light-ray-direction vectors between the bases is standard vector maths. In terms of the metric tensor g in the (x, y, z) basis, the metric tensor in the (u, v, w) basis, g', is

$$g' = \Lambda^T g \Lambda, \tag{37}$$

where

$$\Lambda = \begin{pmatrix} \hat{\boldsymbol{u}} & \hat{\boldsymbol{v}} & \hat{\boldsymbol{w}} \end{pmatrix} = \begin{pmatrix} \partial x/\partial u & \partial x/\partial v & \partial x/\partial w \\ \partial y/\partial u & \partial y/\partial v & \partial y/\partial w \\ \partial z/\partial u & \partial z/\partial v & \partial z/\partial w \end{pmatrix} \tag{38}$$

is the Jacobian matrix.

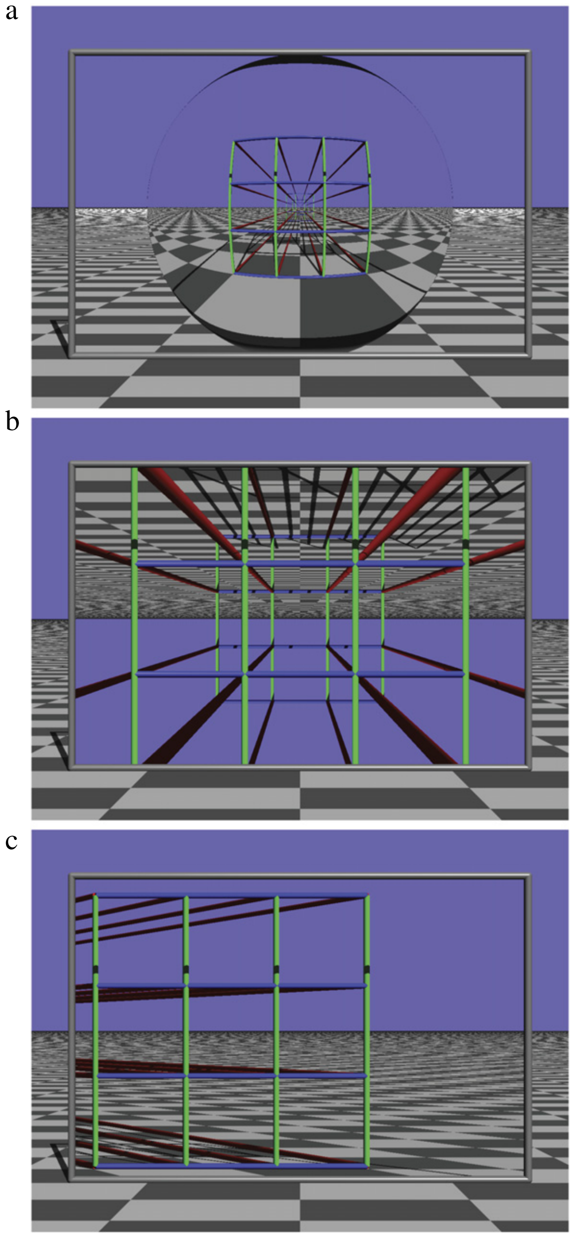


Fig. 14. Simulated view through different metric interfaces. (a) Metric interface equivalent to refractive index $n = 10$ in front of the interface and $n = 1$ behind it. Total internal reflection (TIR) is clearly visible outside the central disc. (b) Metric interface representing a Veselago lens [25], a refractive-index interface between $n = -1$ in front and $n = +1$. (c) Interface between a space sheared by 45° in the (x, z) projection in front and a Euclidean space behind.

4.4. Implementation in TIM

In TIM, metric-tensor interfaces are represented by an object of class `MetricTensorInterface`. The calculation of the refracted light-ray direction $\boldsymbol{e}$ happens in the `getRefractedRayDirection` method. Note that the user can choose whether or not the metric tensors on both sides of the interface are given in the global coordinate system or in the surface coordinate system. The latter is useful for defining homogeneous, but non-planar, surfaces.

Fig. 14 shows a few examples of the simulated view through various metric-tensor interfaces. The procedure to create these (and similar) views is as follows. Start up TIM and click on the "Edit scene" button. In the "Initialise scene to..." drop-down menu select "METATOY science (lattice behind METATOY window)". The

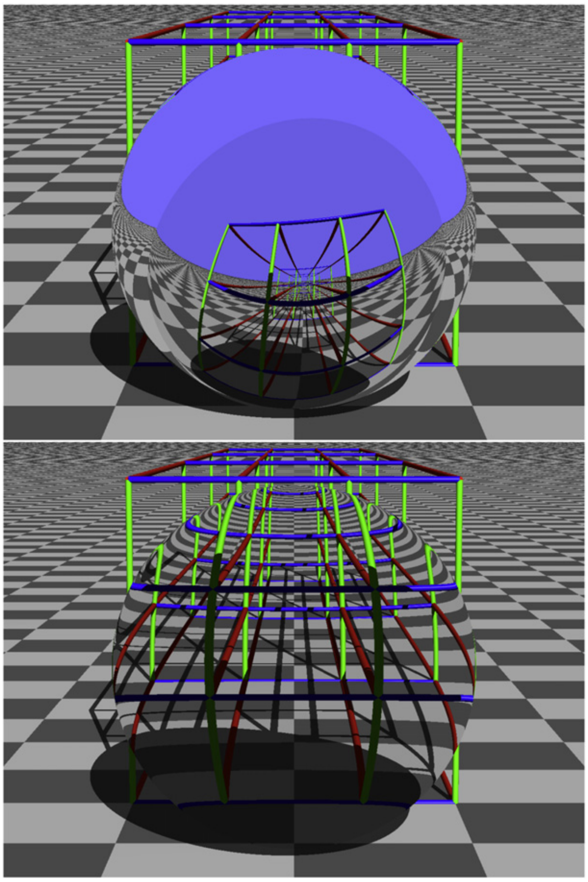


Fig. 15. View of TIM's standard cylinder lattice through "bubbles" of space with a non-Euclidean metric. (Top) The elements of the metric tensor inside the sphere, expressed in the global coordinate basis, are $g_{11} = 1, g_{12} = g_{21} = 0.2, g_{13} = g_{31} = 0, g_{22} = 1.2, g_{23} = g_{32} = -0.5, g_{33} = 0.5$. TIR is visible near the edge of the sphere. (Bottom) The inside of the sphere is described by the Minkowski-like metric diag$(-1, 1, 1)$.

list of scene objects now contains an object called "METATOY window"; double-click on this object to edit its parameters. The window is placed such that it fills a good portion of the default camera. By default, the window's surface type is "Rotating" (which simulates light-ray rotation around the window normal [26]); click on the surface-type drop-down menu and select "Metric-tensor interface". The elements of the metric tensor "inside" and "outside" the window[4] can then be inspected and edited in the tabbed pane.

Fig. 15 shows examples of scenes that involve non-planar metric interfaces.

5. Refraction with generalised confocal lenslet arrays

Another extension we recently added to TIM is the capability to simulate the transmission of light rays through generalised confocal lenslet arrays (gCLAs) [5]. This extension is discussed elsewhere [27], and so, after a brief summary of gCLAs and their effect on light rays, we concentrate on details not already discussed.

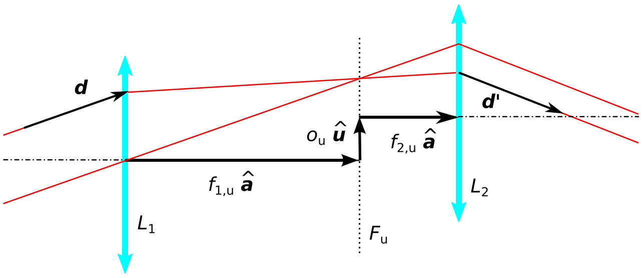

$\boldsymbol{d}$
$\boldsymbol{d}'$
$o_\mathrm{u}\ \hat{\boldsymbol{u}}$
$f_{2,\mathrm{u}}\ \hat{\boldsymbol{a}}$
$f_{1,\mathrm{u}}\ \hat{\boldsymbol{a}}$
L_2
L_1
F_u

Fig. 16. Ray propagation through one of the telescopelets in generalised confocal lenslet arrays (gCLAs). The diagram is an orthographic projection into the plane spanned by the vectors $\hat{\boldsymbol{a}}$ and $\hat{\boldsymbol{u}}$, a unit vector in the u direction, one of the two directions perpendicular to $\hat{\boldsymbol{a}}$ (the other being the v direction). The telescopelet is formed by two lenses, L_1 and L_2, whose optical axes (dash-dotted horizontal lines) are parallel (the unit vector $\hat{\boldsymbol{a}}$ points in the direction of the optical axes), but offset relative to each other, by a distance o_u in the u direction and a distance o_v in the v direction. In general, the lenses have different focal lengths in the u and v directions, but the focal lengths of the two lenses in the u direction add up to the separation between the lenses, as do the focal lengths in the v direction. F_u is the common focal plane in the (u, a) projection. The solid red lines show two light-ray trajectories, both incident with direction $\boldsymbol{d}$ and leaving with direction $\boldsymbol{d}'$.

5.1. Generalised confocal lenslet arrays (gCLAs)

Telescopes have the property that light rays that are parallel when incident are parallel again when leaving. In other words, the direction change does not depend on the precise point where a ray hits the telescope. There is a ray offset, but it is small when the telescopes themselves are small, and so the effect on light rays of a planar array of small telescopes ("telescopelets") with their optical axes perpendicular to the plane is essentially generalised refraction, and so such components are examples of METATOYs [2]. Perhaps the easiest way to build such arrays of telescopelets is from two arrays of lenslets that share a common focal plane, or confocal lenslet arrays (CLAs) [28]. CLAs have interesting imaging properties [28,29], and they have been demonstrated experimentally [30].

The telescopelets in CLAs can be generalised in a number of ways while retaining their basic METATOY character (of introducing a light-ray-direction change that is independent of the precise position where a light ray enters the METATOY while introducing a small offset) [5]. The generalisations are as follows (see Fig. 16).

1. Each lens can be replaced by a combination of cylindrical lenses with their cylinder axes in two transverse directions, u and v, and so each lens has two focal lengths. The focal lengths in the u direction of both lenslets have to add up to the separation between the lenslets, and so do the focal lengths in the v direction.
2. The lenses' optical axes can be offset relative to each other.
3. The entire telescope can be rotated.

In Ref. [5], the telescopes were considered to be aligned initially such that their optical axis points in the z direction, and so that the cylinder axes are aligned in the x and y directions. The parameters necessary to describe a general telescopelet aligned in this way are η_x, the ratio of the focal lengths in the x direction of the second and first lens, multiplied by (-1); η_y, the ratio of the focal lengths in the y direction of the second and 1st lens, again multiplied by (-1); δ_x, the x offset of the optical axis of the second lens relative to that of the first, divided by the first lens's focal length in the x direction; and δ_y, the y offset of the optical axis of the second lens relative to that of the first, divided by the first lens's focal length in the y direction. A general rotation, described by Euler angles ϕ, θ and ψ, makes such a telescopelet completely general. The seven parameters described above are, of course, the seven degrees of freedom that make gCLAs such versatile components.

[4] In TIM, every surface has an inside and an outside, defined by the surface normal which, by definition, points outwards. From the default camera position, the default window is seen from the inside. The direction of the surface normal can be visualised by right-clicking on a surface in the rendered image and selecting, in the pop-up menu that appears, "Add surface-coordinate axes". Rendering the scene again now also shows three arrows pointing along the three coordinate axes defining the surface coordinate system; the outwards surface normal is the blue arrow.

In Ref. [27], gCLAs were described slightly differently, namely by $\hat{\boldsymbol{a}}$, a unit vector in the direction of the optical axis *after* rotation; $\hat{\boldsymbol{u}}$, a unit vector in the direction of the x direction after rotation; η_u, the focal-length ratio in the u direction of the second and first lenses, multiplied by (-1), and η_v, the equivalent ratio for the v direction, which is perpendicular to both the a and u directions (a unit vector in the v direction is defined as $\hat{\boldsymbol{v}} = \hat{\boldsymbol{a}} \times \hat{\boldsymbol{u}}$); and by δ_u, the u offset of the second optical axis relative to the first, divided by the focal length in the u direction of the first lens, and δ_v, the equivalent ratio for the v direction. We use this latter description, for which the law of refraction is [27]

$$\boldsymbol{d}' = \frac{(\boldsymbol{d}\cdot\hat{\boldsymbol{u}})/(\boldsymbol{d}\cdot\hat{\boldsymbol{a}}) - \delta_u}{\eta_u}\hat{\boldsymbol{u}} + \frac{(\boldsymbol{d}\cdot\hat{\boldsymbol{v}})/(\boldsymbol{d}\cdot\hat{\boldsymbol{a}}) - \delta_v}{\eta_v}\hat{\boldsymbol{v}} + \hat{\boldsymbol{a}}. \qquad (39)$$

5.2. Implementation details

In TIM, generalised confocal lenslet arrays are represented by the class `ConfocalLensletArrays`, which is a non-abstract implementation of the abstract `SurfaceProperty` class. The code calculating the refraction is part of the `getColour` method, which handles passage of an incident light ray through the surface. The calculation of the direction of the refracted ray according to Eq. (39) is mostly straightforward. The few subtleties are perhaps worth mentioning.

First, TIM does not ask the user to enter the normalised, and perpendicular, vectors $\hat{\boldsymbol{a}}$ and $\hat{\boldsymbol{u}}$. Instead, TIM asks for vectors $\boldsymbol{a}$ and $\boldsymbol{u}$, which are related to $\hat{\boldsymbol{a}}$ and $\hat{\boldsymbol{u}}$ as follows. The unit vector $\hat{\boldsymbol{a}}$ is simply the vector $\boldsymbol{a}$, normalised. In contrast, $\hat{\boldsymbol{u}}$ is not generally a normalised version of $\boldsymbol{u}$: TIM calculates $\hat{\boldsymbol{u}}$ by projecting $\boldsymbol{u}$ into a plane perpendicular to $\hat{\boldsymbol{a}}$, and then normalising it.

Second, TIM gives the user the choice to define the vectors $\boldsymbol{a}$ and $\boldsymbol{u}$ in different bases, namely either in terms of the "global" (x, y, z) coordinate system or in terms of the "local" surface coordinates (two directions tangential to the surface and a surface normal; see Fig. 7 in Ref. [1]). The latter allows modelling of a non-planar, but homogeneous, gCLA surface.

Finally, it is important in which order a ray encounters the two lenslet arrays that make up the gCLAs. Unlike other surfaces, this order is not decided by whether or not the light ray reaches the surfaces from the inside or the outside, but instead by the sense of the vector $\boldsymbol{a}$, which TIM interprets as pointing from lenslet array 1 to lenslet array 2 (see Fig. 16). TIM establishes whether or not $\boldsymbol{a}$'s and $\boldsymbol{d}$'s components perpendicular to the surface have the same sign; if they do, the ray first intersects lenslet array 1, otherwise lenslet array 2. It can be shown that, in the latter case, the law of refraction is still given by Eq. (39), but with $\hat{\boldsymbol{a}}$ replaced by $-\hat{\boldsymbol{a}}$, η_u replaced by $1/\eta_u$ (and η_v by $1/\eta_v$), δ_u replaced by δ_u/η_u (and δ_v by δ_v/η_v).

6. Other extensions

We have not discussed a number of other changes we made to TIM since the publication of Ref. [1], including a number of extensions (such as the capability to model arrays of objects; phase-conjugating surfaces; surfaces that change the direction of light rays like a Lorentz transform; surfaces which, when spherical, look from the outside like an Eaton lens or a Luneburg lens; imperfect teleporting surfaces; surfaces that refract like phase holograms; new initialisation scenes, including one that simulates an autostereogram resonator [31]; the capability to create 3D images in various HDMI 1.4a standard frame-packing formats (i.e. a single image file simply consisting of the left-eye and right-eye images placed next to each other or on top of each other) compatible with 3D TVs; the capability to move the camera, and vary its field of view, in the interactive version also; and to switch shadow throwing on or off). Most of these changes are minor; others are important, and we intend to describe them in detail elsewhere.

7. Conclusions

TIM is turning out to be a very useful tool for our research. For example, we are currently using the new relativistic-ray-tracing capabilities to investigate the effect of different shutter models on image blur; using the new capability to simulate gCLA refraction, we have recently studied (near-)perfect imaging with planar gCLAs [6]; and we are using the capabilities for simulating refraction at metric interfaces and through gCLAs to test ideas that allow us to realise transformation-optics devices with gCLAs [7].

We are greatly enjoying extending TIM, and we are already planning further extensions. For example, we are currently adding functionality for visualising a few transformation-optics devices, and we are planning to add more. TIM was conceived as a ray-tracing program that allows us to simulate the view through idealised, perfect, METATOYs, ultimately as a tool for finding applications for METATOYs, but now that we have found such applications, the time might have come to add to TIM the ability to simulate some of the imperfections of specific METATOYs.

As before, we aim to encourage others to use and extend TIM. We hope that TIM's new capabilities will help us to achieve this aim.

Acknowledgements

SO acknowledges a studentship from the UK's Engineering & Physical Sciences Research Council (EPSRC). TT acknowledges the support of the grant P201/12/G028 of the Grant agency of the Czech Republic and of the QUEST programme grant of the EPSRC.

References

[1] D. Lambert, et al., Comp. Phys. Commun. 183 (2012) 711.
[2] A.C. Hamilton, J. Courtial, New J. Phys. 11 (2009) 013042.
[3] J. Courtial, T. Tyc, J. Opt. Soc. Am. A 29 (2012) 1407.
[4] J. Courtial, T. Tyc, J. Opt. 13 (2011) 115704.
[5] A.C. Hamilton, J. Courtial, J. Opt. A: Pure Appl. Opt. 11 (2009) 065502.
[6] S. Oxburgh, J. Courtial, Perfect imaging with planar interfaces, J. Opt. Soc. Am. A 30 (2013) 2334.
[7] S. Oxburgh, C.D. White, G. Antoniou, E. Orife, J. Courtial, Transformation optics with windows, in preparation.
[8] A. Howard, S. Dance, L. Kitchen, Relativistic Tay-tracing: Simulating the Visual Appearance of Rapidly Moving Objects, Technical Report, 1995.
[9] C.M. Savage, A. Searle, L. McCalman, Amer. J. Phys. 75 (2007) 791.
[10] T. Müller, S. Grottel, D. Weiskopf, IEEE Trans. Visual. Comput. Graphics 16 (2010) 1243.
[11] R. Penrose, Proc. Cambridge Philos. Soc. 55 (1959) 137.
[12] M. Šarbort, T. Tyc, J. Opt. 14 (2012) 075705.
[13] R.K. Luneburg, Mathematical Theory of Optics, Brown University, Providence, Rhode Island, 1944, pp. 189–213.
[14] J.E. Eaton, Trans. IRE Antennas Propag. 4 (1952) 66.
[15] J.C. Maxwell, Cambridge Dublin Math. J. 9 (1854) 9.
[16] U. Leonhardt, New J. Phys. 11 (2009) 093040.
[17] J.H. McLeod, J. Opt. Soc. Amer. 44 (1954) 592.
[18] R.M. Herman, T.A. Wiggins, J. Opt. Soc. Am. A 6 (1991) 932.
[19] M. Šarbort, T. Tyc, Multi-focal spherical media and geodesic lenses in geometrical optics, J. Opt. 15 (2013) 125716.
[20] J. Perczel, T. Tyc, U. Leonhardt, New J. Phys. 13 (2011) 083007.
[21] U. Leonhardt, Science 312 (2006) 1777.
[22] J.B. Pendry, D. Schurig, D.R. Smith, Science 312 (2006) 1780.
[23] J.B. Pendry, Phys. Rev. Lett. 85 (2000) 3966.
[24] D. Schurig, J.B. Pendry, D.R. Smith, Opt. Express 14 (2006) 9794.
[25] V.G. Veselago, Sov. Phys. Uspekhi 10 (1968) 509.
[26] A.C. Hamilton, B. Sundar, J. Nelson, J. Courtial, J. Opt. A: Pure Appl. Opt. 11 (2009) 085705.
[27] S. Oxburgh, C.D. White, G. Antoniou, J. Courtial, Law of refraction for generalised confocal lenslet arrays, Opt. Commun. 313 (2014) 119.
[28] J. Courtial, New J. Phys. 10 (2008) 083033.
[29] J. Courtial, Opt. Commun. 282 (2009) 2480.
[30] J. Courtial, B.C. Kirkpatrick, E. Logean, T. Scharf, Opt. Lett. 35 (2010) 4060.
[31] S. Leavey, K. Rae, A. Murray, J. Courtial, Opt. Commun. 285 (2012) 3971.