Real-Time Rendering of Eclipses without Incorporation of Atmospheric Effects

S. Schneegans[1] , J. Gilg[1], V. Ahlers[2] and A. Gerndt[1,3]

[1]German Aerospace Center (DLR)
[2]University of Applied Sciences and Arts Hannover, Germany
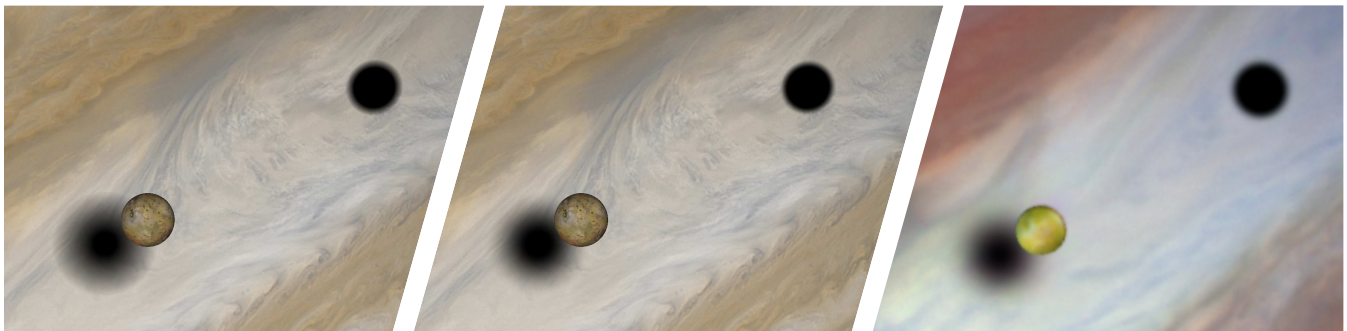[3]University of Bremen, Germany

Figure 1: *All images show the Jovian moon Io and its shadow together with the shadow of Callisto. The left image has been rendered according to the state-of-the-art in real-time rendering using a linear brightness gradient in the shadow. This results in an unrealistically sharp outer edge. The middle image was generated using the approach described in this paper, incorporating not only the occluded fraction of the Sun, but also the effect of solar limb darkening. The right image is a photograph of the same scene, taken by the Hubble space telescope on January, 24th in 2015. The proposed approach results in a smoother and more realistic shadow gradient. The real shadow appears to be even softer which may be due to the fact that Jupiter does not have a hard surface but rather a dense atmosphere.*

Abstract

In this paper, we present a novel approach for real-time rendering of soft eclipse shadows cast by spherical, atmosphereless bodies. While this problem may seem simple at first, it is complicated by several factors. First, the extreme scale differences and huge mutual distances of the involved celestial bodies cause rendering artifacts in practice. Second, the surface of the Sun does not emit light evenly in all directions (an effect which is known as limb darkening). This makes it impossible to model the Sun as a uniform spherical light source. Finally, our intended applications include real-time rendering of solar eclipses in virtual reality, which require very high frame rates. As a solution to these problems, we precompute the amount of shadowing into an eclipse shadow map, which is parametrized so that it is independent of the position and size of the occluder. Hence, a single shadow map can be used for all spherical occluders in the Solar System. We assess the errors introduced by various simplifications and compare multiple approaches in terms of performance and precision. Last but not least, we compare our approaches to the state-of-the-art and to reference images. The implementation has been published under the MIT license.

CCS Concepts
• *Computing methodologies* → *Real-time simulation;*

1. Introduction

During an eclipse, the sunlight reaching an observer is partially or fully blocked by a celestial body. In the Earth-Moon system, two types of eclipses may occur: When the Earth enters the shadow volume of the Moon, a solar eclipse can be observed. Vice-versa, when the Moon enters the shadow volume of the Earth, a lunar eclipse takes place. Such events have always been awe-inspiring and fascinating for humankind. Today, we can visualize the com-
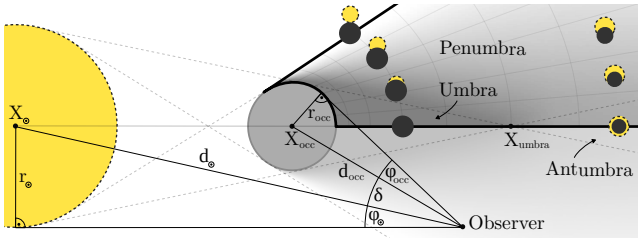
Figure 2: *This graphic introduces terms and measures used in this paper. The Sun (radius $r_\odot$) casts light onto the spherical occluder (radius r_{occ}). The small glyphs depict how Sun and occluder could look like from the respective position in space. The three angles φ_{occ}, δ, and $\varphi_\odot$ are used in* Section 5 *for the parameterization of the eclipse shadow map. This figure is obviously not drawn to scale. See* Figure 3 *for an image of Earth's shadow drawn to scale.*

plex movements of celestial bodies in photorealistic and immersive simulations of our universe. Replicating eclipse shadows in such virtual environments is not only useful for conveying the mechanics of our Solar System, but is also beneficial for use-cases ranging from training for on-orbit servicing missions (as satellites may enter eclipse shadows quite frequently) to interactive planning of space missions. In this paper, we only consider shadows cast by spherical, atmosphereless bodies. Hence, the proposed system is able to simulate the shadow of most large moons of our Solar System. This shadow can be cast onto anything: on planets, on other moons, or even on spacecrafts. It can also be used to simulate shadows cast by planets, however it does not include atmospheric effects which, for instance, lead to the characteristic reddening of the Moon during a lunar eclipse. The proposed approach has only little impact on rendering performance making it suitable for real-time rendering. At the same time, it produces more realistic shadows than the state-of-the-art.

In the following section, we define the geometric relations between the Sun, the occluder, and the occluded observer during an eclipse. Thereafter, in *Section 3*, we analyze how existing interactive visualizations of our Solar System compute eclipse shadows. We show in *Section 4* how this can be improved by exactly computing the occluded fraction of the Sun's surface as well as by the incorporation of limb darkening. As this is impossible to compute in real-time, we propose to use an eclipse shadow map in *Section 5*. In *Section 6*, we compare the results with reference images and evaluate the rendering performance.

2. Physical Background

Eclipses occur when the sunlight reaching an observer is occluded by a celestial body. In this section, we define the geometric properties of such an event, as well as introduce key characteristics of the involved light source, our Sun.

2.1. Geometry of an Eclipse

As shown in *Figure 2*, the eclipse shadow volume can be subdivided into three distinct regions: if the Sun is entirely occluded, the observer witnesses a *total eclipse* in the completely dark *umbra* region of the shadow. In regions where the Sun is only partially occluded, the shadow is called *penumbra*. Finally, in the *antumbra* region, the apparent size of the occluder is too small to fully cover the Sun's disc. Therefore, the Sun appears as a ring around the occluder. In this case, the observer witnesses an *annular eclipse*.

Due to the huge distances in our Solar System, the shadows are extremely thin and long (see *Figure 3*). Their shape depends on the size of the occluding body and its distance to the Sun. While the body size is constant, its distance to the Sun can vary. Since orbits are elliptic, the distance to the Sun changes during the bodies revolution, and thus the shape of the shadow varies over time. If we assume that the Sun and the occluder are spheres, the shadow volume is rotationally symmetric around the Sun-occluder axis.

2.2. Properties of the Sun

While the Sun is neither a perfect sphere nor has a constant radius over time, the variations are very small [RN09, FHLZ08]. For this paper, we use a value of $r_\odot = 696,342 \pm 65\,\text{km}$ for the radius of the Sun [EKBS12].

Another important property of the Sun affecting the appearance of the eclipse shadow is that it does not appear equally bright over its entire area. The brightness rather peaks in the center and decreases towards the outer rim. The effect is called *limb darkening* and there are several wavelength-dependent formulas for modelling it [HM98]. A simple and widely used model is given by the following formula [MJOS17]:

$$L(\varphi) = L(0)\left[1 - u\left(1 - \sqrt{\frac{\varphi_\odot^2 - \varphi^2}{\varphi_\odot^2}}\right)\right]. \qquad (1)$$

Here, $L(0)$ refers to the luminance at the center of the solar disc, $\varphi_\odot$ to the angular radius of the solar disc, and φ to the current angular radial distance to the center. The limb darkening coefficient u is actually wavelength-dependent, however a constant value of $u = 0.6$ is a very good approximation for visible light which has been used for decades [MJOS17]. The right image of *Figure 4* was generated using this value. Also, even if for this paper, we use *Equation 1* with a constant $u = 0.6$, the presented approach can be used with wavelength-dependent limb-darkening models. In such a case, there would be a subtle color shift in the eclipse shadow even if no atmosphere is involved. As we do not incorporate atmospheric effects and use a wavelength-independent model for limb darkening, we eliminate any dependency on the color of the sunlight. Hence, no explicit modelling of the spectrum of the Sun is required for this paper.

Figure 3: *This is Earth's shadow, drawn to scale. Earth is shown on the left hand side (a tiny blue dot), and the orbit of Moon is marked with a red line. The end of the umbra is on the right side of the image. If printed on A4 paper size, the Sun would be about 20 m to the left.*
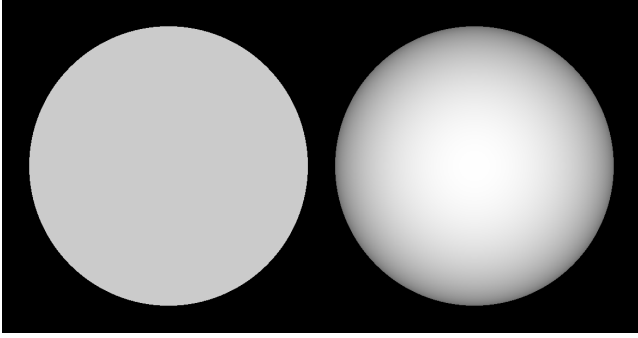
Figure 4: *For some applications, the Sun's surface can be considered to emit light equally in all direction (left image). However, in reality, the luminance decreases from the center to the rim of the Sun's disc (right image). Both images have the same average luminance, the right one has been generated using* Equation 1.

3. Related Work

There are image synthesis tools available, which are able to simulate realistic eclipse shadows using ray tracing, such as the commercial software *SurRender* by Airbus [BLR*18]. However, for this paper, we only evaluate approaches which are designed to generate images in real-time (i.e. at interactive frame rates). Over the decades, vastly different approaches for computing shadows in real-time have been invented [ESAW11]. Their goal is often to offer good enough visuals that make use of approximations and are far from physical correctness. However, to the best of our knowledge, none of them is designed to exploit the rather simple geometric properties of an eclipse. Additionally, while most of them can deal with area or circular light sources, they assume that the light source emits light equally. None of the approaches can deal with light sources that are subject to limb darkening or other nonuniform distributions.

While we did not find related work which attempts to simulate arbitrary eclipses, several papers have been published dedicated to the simulation of lunar eclipses only. Furthermore, there are several open-source 3D visualizations of the Solar System available, which already simulate eclipses to a certain extent.

3.1. Real-Time Simulation of Lunar Eclipses

In 2012, Müller et al. proposed to use a one-dimensional pre-generated lookup table for the color and brightness of light reaching the lunar surface during an eclipse [MED12]. The table only needs one dimension thanks to the rotational symmetry of the shadow. It is generated by matching a model to a series of photographs.

As shown by Müller et al., this approach can produce very realistic lunar eclipses with just one additional texture lookup at run time. However, it is limited to the Earth-Moon system, as there is not enough data available to generate such lookup tables for other moons or planets. Furthermore, the lookup table is only valid for a specific position in Earth's shadow (e.g. at the distance of the Moon). Therefore, it cannot be used to simulate eclipse shadows cast onto other bodies, such as spacecrafts entering Earth's shadow.

A more versatile but also much more complex approach has been presented by Yapo and Cutler in 2009 [YC09]. In their approach, "photons" are generated on the surface of the Sun and traced through Earth's atmosphere on curved paths according to an algorithm presented by Seron et al. [SGGC04]. As soon as a "photon" leaves the atmosphere, its wavelength, intensity, and direction are stored. To compute the final appearance of the Moon at one specific point in time, the paths of all exiting photons are intersected with the lunar sphere and accumulated in an illumination map. Thereafter, this map can be used for real-time rendering of this one specific point in time.

The authors do not consider the impact of limb darkening, although it should be straight-forward to integrate. Yet for our purpose, the approach is not well suited as it requires the recreation of the illumination map if the simulation time changes. Furthermore, raytracing through the atmosphere adds a complexity which is not required for the scope of this paper.

In fact, the rather simple geometric relationship between Sun and occluder makes an analytical solution seem viable. To compute the occluded fraction of the Sun in a mathematically correct manner, the intersection of two spherical caps needs to be computed. Tovchigrechko and Vakser present a formula for the area of intersection A_{caps} between two spherical caps with angular radii φ_a and φ_b and an angular distance of δ between their centers [TV01]:

$$\begin{aligned} A_{caps} = \ & 2\pi - 2\arccos(\cos\delta\,\csc\varphi_a\,\csc\varphi_b - \cot\varphi_a\,\cot\varphi_b) \\ & - 2\arccos(\cos\varphi_b\,\csc\delta\,\csc\varphi_a - \cot\delta\,\cot\varphi_a)\cos\varphi_a \\ & - 2\arccos(\cos\varphi_a\,\csc\delta\,\csc\varphi_b - \cot\delta\,\cot\varphi_b)\cos\varphi_b. \end{aligned}$$
(2)

Oat and Sander used this formula for computing the occluded fraction of the Sun for simulating soft shadows on a terrain [OS07]. To increase rendering performance, they approximated the occluded fraction using a cubic Hermite interpolation:

$$\begin{aligned} A_{smoothstep} = \ & (2\pi - 2\pi\cos(\min(\varphi_a, \varphi_b))) \\ & \cdot \text{smoothstep}\left(1 - \frac{\delta - |\varphi_a - \varphi_b|}{\varphi_a + \varphi_b - |\varphi_a - \varphi_b|}\right) \end{aligned}$$
(3)

where $\quad$ smoothstep$(x) = 3x^2 - 2x^3$.

In *Section 4*, we attempt to compute eclipse shadows using *Equation 2*. Afterwards, in *Section 6*, we evaluate the error introduced by the *Equation 3* as well as the performance gained with this approach.

3.2. Existing Solar System Visualizations

There are several real-time space simulations available, already simulating eclipse shadows throughout the Solar System. In the following, we briefly compare the eclipse implementations of Celestia [Gre17], Cosmographia [Sem18] and OpenSpace [BAC*19]. None of these applications incorporate atmospheric effects or limb darkening into the eclipse shadow computation and hence all of them assume complete darkness in the umbra.

All three applications use a linear brightness gradient from the umbra to the outer edge of the penumbra. This results in two cir-
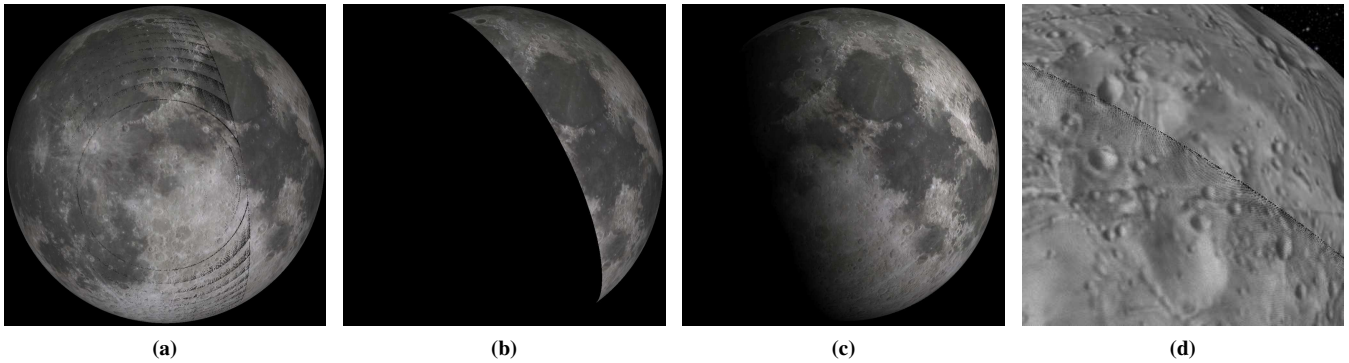
Figure 5: *There are severe artifacts in the transition from complete illumination to the start of the penumbra of the lunar eclipse on 01-21-2019, calculated by the single precision spherical cap intersection method (a). With the same method, the transition from the penumbra to the umbra is a harsh line (b). Using the much simpler calculation of circle intersections results in a smooth transition even with single precision (c). However, also with the single precision circle intersection method some artifacts can be discovered throughout the Solar System (d).*

cular discontinuities in the derivative of the luminance. According to the Mach-band effect, the human visual system emphasizes these discontinuities [Kin14]. This illusion can be seen as an apparent bright ring around the shadows in the first image of *Figure 1*. The last image of *Figure 1* shows that the outer transition is much smoother in reality.

The shadow intensity gradient in the antumbra (beyond the end of the umbra) is handled quite differently by all three applications. Actually, all points in the antumbra with the same distance to the occluder should receive the same amount of sunlight. This is because when seen from those points, the disc of the occluder is fully contained in the disc of the Sun. If limb darkening is ignored, this leads to the same illumination for all points. However, this is only the case in Celestia. In Cosmographia and OpenSpace, there is no area of constant shadow intensity for an annular eclipses in the antumbra. Instead, the shadow intensity increases linearly from the outer edge of the penumbra towards the antumbra and continues to rise towards the Sun-occluder axis. Also, the shadow intensity with increasing distance to the occluder is computed differently. The best approximation is done again by Celestia, which assumes a quadratic intensity falloff beyond the end of the umbra.

Out of the three, only Cosmographia supports shadow volumes for ellipsoidal occluders. It uses a transformation matrix to transform surface points of the occluded body into shadow-relative coordinates. By an asymmetrical scaling of this matrix, they can approximate the shadow of an ellipsoidal occluder. Celestia and OpenSpace approximate Sun and occluder as circles oriented perpendicular to the Sun-occluder axis. This results in a slightly underestimated shadow volume size. For example, at the distance of the Moon, Earth's shadow is about 20 km wider if Sun and Earth are modelled with spheres instead of circles.

In *Section 6*, we compare our approach with the results when using a linear brightness gradient in the penumbra and a quadratic shadow intensity falloff beyond the end of the umbra.

3.3. CosmoScout VR

We implemented the proposed approach into CosmoScout VR, an interactive modular virtual Solar System for the exploration of large planetary datasets in conjunction with simulation data [SZGG22]. The software supports a variety of virtual reality devices, such as head-mounted displays, powerwalls, or CAVEs. Consequently, it is required that the eclipse shadow computation adds as little performance overhead as possible.

The software is based on the ViSTA Virtual Reality Toolkit, initially developed by the RWTH Aachen [AK08] and uses NASA's SPICE library for positioning of celestial objects [Act96]. This allows calculating time-dependent positions of bodies in the Solar System with high precision. Hence, we can use CosmoScout VR to re-enact real eclipses and compare the results with photographs. CosmoScout VR is freely available under the MIT license [SFG].

4. Computation of the Eclipse Shadow

If Sun and occluder are modelled as spheres, calculating the shadow intensity at any point behind the occluder becomes a rather simple geometric problem. First, we present the mathematically correct variant using the intersection area of two spherical caps. Second, we show that this solution can be well approximated by intersecting circles instead. Finally, to increase rendering performance, to reduce rendering artifacts, and to support limb darkening, we describe our proposed parametrization of a pre-generated shadow texture.

4.1. Intersection of spherical caps

In order to correctly calculate the occluded fraction of the solar disc, both, the Sun and the occluding body, are projected onto a unit sphere around the observer. The solid angles subtended by the projected spheres correspond to the surface areas of spherical caps $A_\odot = 2\pi(1 - \cos\varphi_\odot)$ and $A_{occ} = 2\pi(1 - \cos\varphi_{occ})$ respectively. The amount of blocked light is given by their intersection area A_{caps}
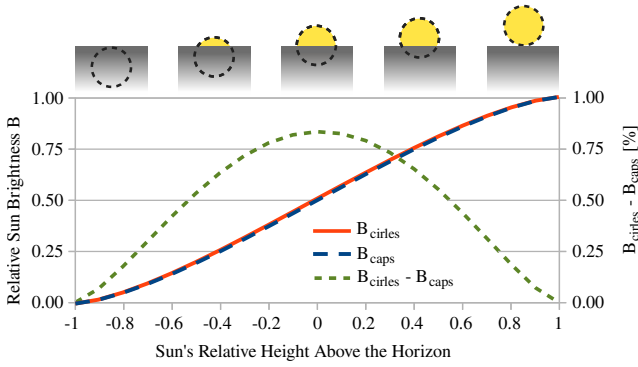
Figure 6: *In this example, the observer is positioned on the surface of a hypothetical body orbiting around the Sun at a tenth of the distance to Mercury. At this distance, the Sun has an angular radius of $\varphi_\odot = 0.1$ radians. The x-axis denotes the relative height of the Sun above the horizon of the occluding body (visualized with the glyphs above the diagram). Even so close to the Sun, B_{caps} and $B_{circles}$ are almost identical. The difference between them in percent is shown with the green curve. At this distance to the Sun, the maximum difference is below 1%.*

Figure 7: *This graph shows the maximum relative Sun brightness difference when computed with circle intersection and the spherical cap intersection method as a function of the distance to the Sun. At the perihelion of Mercury, the value is about 0.081%, at Earth it is at 0.031%. The maximum relative difference of 0.81% of* Figure 6 *is marked with a vertical line. At the Sun's surface (0.0047 AU), the curve would raise up to 10.9%. However, this is just a theoretical number, as an observer would need to be standing on the surface of the Sun* and *on the surface of the occluder at the same time.*

which can be calculated with *Equation 2*. Using $\cot(x) = \frac{\cos(x)}{\sin(x)}$, the original equation can be transformed to the following:

$$
\begin{aligned}
A_{caps} = {} & 2\pi - 2\arccos\left(\frac{\cos\delta - \cos\varphi_a \cos\varphi_b}{\sin\varphi_a \sin\varphi_b}\right) \\
& - 2\arccos\left(\frac{\cos\varphi_b - \cos\delta \cos\varphi_a}{\sin\delta \sin\varphi_a}\right)\cos\varphi_a \qquad (4) \\
& - 2\arccos\left(\frac{\cos\varphi_a - \cos\delta \cos\varphi_b}{\sin\delta \sin\varphi_b}\right)\cos\varphi_b.
\end{aligned}
$$

This "simplified" version reduces the amount of trigonometric functions which need to be evaluated from twelve to nine and allows for efficient use of *fused multiply-add* operations. To compute the final lighting of the occluded body, the luminance value, which is obtained by CosmoScout's rendering pipeline, is multiplied with the visible fraction B_{caps} of the solar disc. In the following, we refer to this value as *relative Sun brightness* (hence the symbol B):

$$
B_{caps} = \frac{A_\odot - A_{caps}}{A_\odot}. \qquad (5)
$$

In practice, floating point operations with 32 bits do not provide enough precision to achieve good results (see *Figure 5* a & b). Implementing the same calculations with double precision in GLSL yields significantly better results. However, as of writing this paper, double precision is not well-supported on graphics cards. Especially trigonometric functions have to be implemented by the user. As demonstrated in *Section 6*, this degrades rendering performance.

4.2. Intersection of circles

In order to increase the performance and to reduce precision issues, we approximate the mathematically correct version using circle intersections. The area enclosed by two intersecting circles $A_{circles}$,
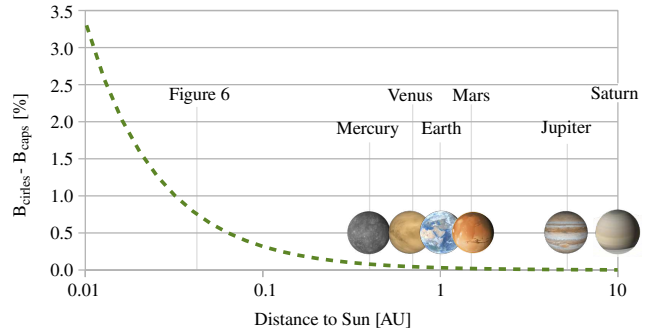
with radii φ_a and φ_b and a distance of δ, is described by the following set of equations. Compared to *Equation 4*, the number of trigonometric functions is much lower.

$$
\begin{aligned}
A_{circles} = {} & \varphi_a^2 \arccos\left(\frac{d_1}{\varphi_a}\right) - d_1\sqrt{\varphi_a^2 - d_1^2} \\
& + \varphi_b^2 \arccos\left(\frac{d_2}{\varphi_b}\right) - d_2\sqrt{\varphi_b^2 - d_2^2} \qquad (6) \\
\text{where} \quad & d_1 = \frac{\varphi_a^2 - \varphi_b^2 + \delta^2}{2\delta} \\
& d_2 = \delta - d_1.
\end{aligned}
$$

$B_{circles}$ is computed analogous to *Equation 5*. In order to assess the error introduced with this approximation, we construct a worst-case scenario. For very small radii, the surface areas of a spherical cap and a circle are almost identical. As their radii increase, also the difference of their surface areas increase. Consequently, the approximation error is the largest when the spherical caps for Sun and occluding body both cover a large portion of the unit sphere around the observer. The observer can be very close to the occluding body, in fact we can put the observer directly onto its surface. This yields the maximum angular radius of the occluder $\varphi_{occ} = \frac{\pi}{2}$ since the observer cannot be placed inside the occluder. The angular radius of the Sun increases when the observer is getting closer to the Sun. *Figure 6* shows the difference between the two methods for a hypothetical body orbiting the Sun at a tenth of the distance to Mercury.

Figure 7 plots the maximum difference between both approaches for varying distances to the Sun (with the observer always located directly on the surface of the occluding body). At the distance of Mercury, which is the inner-most planet of our Solar System with a perihelion of $|X_\odot - X_{occ}| = 0.308$ AU, the maximum relative difference is around 0.1%. This difference directly maps to the resulting shadow-intensity difference. On a typical 8 bit brightness
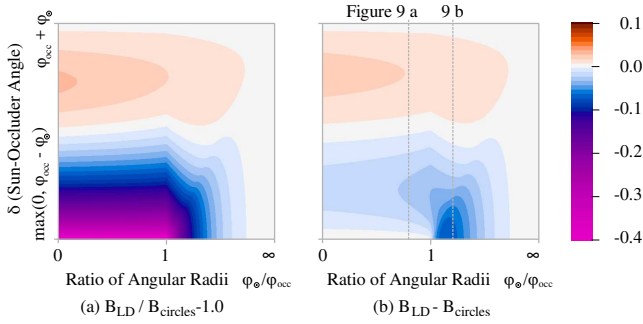
(a) $B_{LD} / B_{circles} - 1.0$

(b) $B_{LD} - B_{circles}$

Figure 8: *Both charts compare the relative Sun brightness at all points in the penumbra and antumbra (see* Figure 10 *a for an explanation of the parameter space) if computed with limb darkening (B_{LD}) and without ($B_{circles}$). The left graph shows that the penumbra is up to 3 % brighter if limb darkening is included (upper left area) while the area close to the umbra appears about 40 % darker (bottom left). The right chart shows the absolute brightness difference. With this metric, the region close to the umbra shows only little difference as not much light is reaching this area anyways. However in the antumbra, the absolute brightness difference is significant (blue area at the bottom). This large absolute difference in the antumbra is also visible on the left-hand side of* Figure 9 *b.*

(a) Total Eclipse
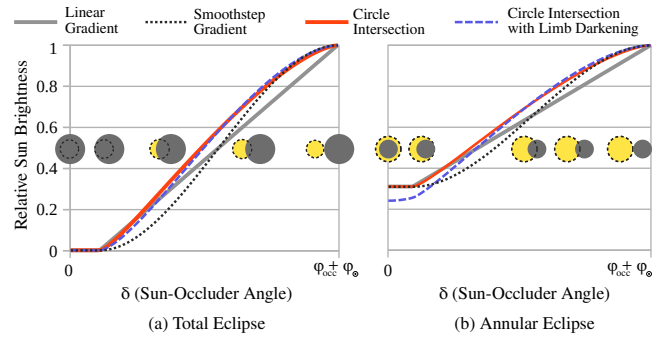
(b) Annular Eclipse

Figure 9: *The graphs show the relative amount of light reaching an observer during a total eclipse (a) and an annular eclipse (b). Compared with a simple linear gradient, computing the visible fraction of the Sun's surface with the circle intersection method results in significantly brighter eclipses. The smoothstep approximation proposed by [OS07] produces too dark eclipse shadows most of the time. The effect of limb darkening is most pronounced during annular eclipses: In such an event, only the limb of the Sun is visible. Due to the limb-darkening effect, this area is darker than the average Sun brightness. Hence, including limb darkening results in darker annular eclipse shadows. The position of both charts is marked in* Figure 8 *with vertical lines.*

scale, this difference is imperceptible. Farther away from the Sun or from the occluding body, the difference becomes even smaller. Therefore, this method seems to be a viable approximation of the spherical-cap-intersection method for computing eclipse shadows.

When implemented in single precision in GLSL, the overall amount of floating point issues is significantly lower (see *Figure 5* c). However, there are still constellations in the Solar System which lead to small artifacts (see *Figure 5* d). Using double precision resolves all rendering artifacts, however it is not a viable solution as it degrades rendering performance significantly. *Section 5* shows how a lookup texture can be used to pre-compute the intersection values with high precision.

4.3. The Effect of Limb Darkening

Until now, we ignored the effect of limb darkening. This allowed us to use the visible fraction of the solar disc to compute the relative Sun brightness (*Equation 5*). However, as discussed in *Section 2.2*, the Sun does not appear equally bright over its entire area. To consider this phenomenon, we integrate *Equation 1* over the solar disc and over the intersection area and call the resulting values $I_\odot$ and $I_{intersection}$ respectively. Then again, $B_{LD} = (I_\odot - I_{intersection})/I_\odot$ is a dimensionless factor in the range $[0,1]$ which can be used to attenuate the Sun-based illuminance in the shadow volume.

As shown in *Section 2.2*, the luminance of the outer rim of the solar disc is actually about 40 % darker than its center. For an observer inside the eclipse shadow, this results in less illuminance close to the umbra (only a portion of the rim of the solar disc is visible from there) and in a greater illuminance in the penumbra (when only a portion of the rim is occluded). *Figure 8* shows the difference in illuminance for all positions in the penumbra and antumbra. *Fig-*

ure 9 shows the effect of limb darkening exemplary for a total and an annular eclipse. In addition, the smoothstep approximation proposed by [OS07] as well as a linear gradient which is used by many state-of-the-art visualization tools (see *Section 3*) is plotted. It can be seen that for total eclipses, the effect of limb darkening is small. However, annular eclipses (especially if close to totality) are significantly darker with limb darkening.

For computing the average luminance of the visible portion of the solar disc during an eclipse, we sample the visible fraction of the solar disc and evaluate *Equation 1* at each point. This is not feasible to do in real-time. This and the increased floating point accuracy, which can be obtained by using double precision on the CPU, encourages the use of a pre-generated shadow lookup texture.

5. The Pre-Computed Shadow Texture

Since double precision and sampling of the limb-darkening function are required for the calculation of artifact-free eclipse shadows, we propose to pre-compute an eclipse shadow lookup texture. For this, we propose a parametrization of the texture which is independent of the occluder's size and its distance to the Sun. Hence, only one such lookup texture is required as it can be used for all spherical bodies without an atmosphere. Furthermore, this texture can also be used for bodies orbiting the Sun on an elliptical path as it is valid at all places in the Solar System.

5.1. Parametrization of the Shadow Texture

We propose to map the horizontal axis of the lookup texture to the ratio of the angular radii of Sun and occluder ($\varphi_\odot / \varphi_{occ}$). This corresponds to a non-linear distance to the occluder. Pixels on the left-
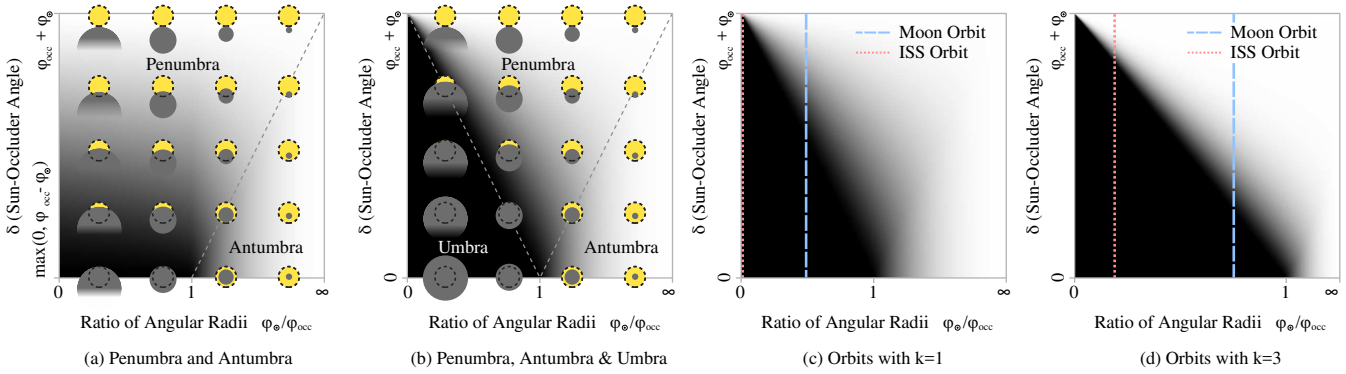
Figure 10: *Graph (a) depicts our parametrization of the penumbra and antumbra based on the apparent angular radii of Sun and occluder ($\varphi_\odot$ and φ_{occ}) and their mutual angular distance δ. The glyphs give an idea how Sun and occluder may look like for the respective point in the parameter space. In variant (b), the mapping of the vertical axis is modified to also include the umbra shadow region. This parametrization contains all points of the area enclosed with a thick black line in* Figure 2. *To give a sense of scale, (c) contains the orbits of the Moon and the International Space Station (ISS) as vertical lines if the eclipse shadow texture represents the shadow of Earth. It can be seen, that the eclipse shadow for objects close to the occluder (such as the ISS) cannot be represented properly by this parametrization. In variant (d), both axes are scaled non-linearly (here with an exponent of k = 3). This provides a higher data resolution in the shadow close to the occluder.*

hand side of the lookup texture map to positions close to the occluder ($\varphi_\odot/\varphi_{occ} = 0$), pixels in the middle correspond to the end of the umbra ($\varphi_\odot/\varphi_{occ} = 1$), and pixels on the right-hand side map to positions very far from the occluder ($\varphi_\odot/\varphi_{occ} = \infty$).

We propose to map the vertical axis to the angular height of the Sun above the occluder's horizon. This means that the Sun is completely hidden behind the horizon for the bottom line of pixels and completely visible above the horizon for the upper line of pixels. If the apparent angular radius of the Sun exceeds the size of the occluder (that is the case in the right half of the texture, beyond the end of the umbra), the minimum angular distance between Sun and occluder is set to zero (*Figure 10* a). This parametrization exactly contains all points in the penumbra and antumbra. The following equations transform the three variables $\varphi_\odot$, φ_{occ}, and δ to relative texture coordinates *x* and *y*:

$$
\begin{aligned}
x &= \frac{1}{\varphi_{occ}/\varphi_\odot + 1} \\
y &= \frac{\delta - \delta_{min}}{\delta_{max} - \delta_{min}} \\
\text{where} \quad \delta_{min} &= \max(0, \varphi_{occ} - \varphi_\odot) \\
\delta_{max} &= \varphi_{occ} + \varphi_\odot.
\end{aligned}
\tag{7}
$$

For this paper, the umbra region does not need to be included in the shadow lookup texture as no light is reaching this area. However, to pave the way for future works which may include atmospheric refraction, we propose an alternative parametrization which also includes the umbra. For this, the vertical axis is mapped to the angular distance between Sun and occluder, always starting from zero at the bottom ($\delta_{min} = 0$) to $\varphi_\odot + \varphi_{occ}$ at the top (*Figure 10* b).

An issue of this alternative parametrization is, that the transition from complete illumination to complete darkness for potentially interesting positions in low orbits are encoded in just a few pixels in the upper left corner of the lookup texture. For instance, this parametrization is not well-suited for visualizing the ISS entering

Earth's shadow (*Figure 10* c). As a solution, both axes of the lookup texture can be scaled with an exponent *k* (*Figure 10* d):

$$
\begin{aligned}
x' &= x^k \\
y' &= 1 - (1-y)^k.
\end{aligned}
\tag{8}
$$

It is crucial to note, that the proposed parametrizations are only possible with the circle-intersection method; they would not work with the spherical-cap intersection. This is due to the following: if Sun and occluder are considered to be spheres, the amount of shadowing at any point in space depends on three variables: the apparent angular radii of Sun and occluder and their angular distance to each other. If this intersection is computed using circles, all three variables can be scaled by a constant and yet the occluded fraction of the Sun-circle stays the same. Hence, only two of the three variables are free and a 2D lookup texture is sufficient. If the intersection is computed on a unit sphere using *Equation 4*, it is not possible to apply a uniform scale to all variables without affecting the occluded fraction. In this case, a 3D lookup texture would be required to properly pre-compute all possible constellations.

In our implementation, we integrate *Equation 1* once over the entire solar disc to obtain an average luminance $I_\odot$ for $L(0) = 1$. Then, using *Equation 7*, the values for φ_{occ} and δ (and an arbitrary $\varphi_\odot$) can be obtained for each pixel position (x, y) in the eclipse shadow map. Thereafter, we compute the average luminance I_{LD} of the corresponding circle-intersection area by integrating *Equation 1* over the area. For a simple limb darkening function, an analytical solution of the integral could be viable. However, to make it possible to use alternative limb-darkening models, we chose to compute the integral by sampling the intersection area with a regular grid of sample points. *Figure 11* shows the required preprocessing time as a function of the number of sample points. In the texture, we store the resulting relative illuminance value $I_{LD}/I_\odot$ which is in the range $[0, 1]$. Since all pixel values can be computed independently, the algorithm can highly benefit from parallel processing.
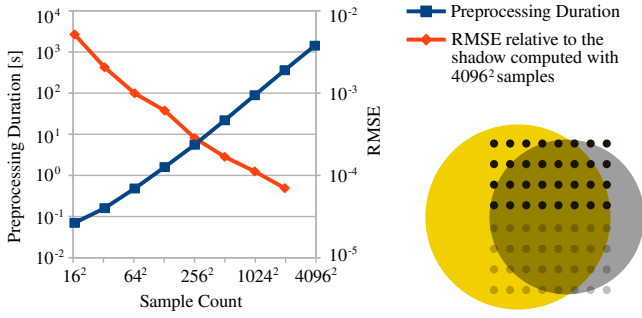
Figure 11: *The blue line plots the time it took to generate a 512^2 shadow texture as a function of the sample count taken in the intersection area of Sun and occluder. The red line shows the root-mean-square error (RMSE) of the respective shadow texture when compared to the shadow texture computed with the maximum amount of samples (4096^2). The figure on the bottom right illustrates the sampling pattern for 8^2 samples. Only half of the samples need to be evaluated due to the vertical symmetry. The shadow textures were computed using CUDA on an NVidia GTX 1650 TI mobile GPU.*

Figure 12: *This figure shows the introduced error in the shadow of various planets if* Equation 10 *is used instead of* Equation 9. *The x-axis encodes the distance to the respective body up to the length of the umbra ($d_{umbra} = |\overrightarrow{X_{occ}X_{umbra}}|$). It can be seen, that the error is large for positions closer to the shadow-casting body as well as for bodies closer to the Sun. However, it decreases quickly with an increasing distance. Even for Mercury, the error already decreased to about 1% at a tenth of the umbra length. The dashed line shows the error if the hybrid approach is used.*

5.2. Using the Shadow Texture

For shading a fragment at runtime, one texture lookup has to be done for each relevant shadow casting body. Using *Equation 7* in the shader, the texture coordinates can be computed with only the angular radii of the Sun, the occluding body, and their angular distance ($\varphi_\odot$, φ_{occ}, and δ). These can be computed from the measures defined in *Figure 2* according to the following formulas (with O being the position of the observer or the to-be-shaded fragment):

$$\varphi_{occ} = \arcsin\left(\frac{r_{occ}}{d_{occ}}\right) \qquad \varphi_\odot = \arcsin\left(\frac{r_\odot}{d_\odot}\right)$$
$$\delta = 2 \cdot \arcsin\left(\frac{d}{2}\right) \quad \text{with} \quad d = \left|\frac{\overrightarrow{OX_{occ}}}{d_{occ}} - \frac{\overrightarrow{OX_\odot}}{d_\odot}\right|. \tag{9}$$

Typically, one would compute $\delta = \arccos\left(\overrightarrow{OX_{occ}}/d_{occ} \cdot \overrightarrow{OX_\odot}/d_\odot\right)$. However, as *arccos* is numerically unstable for values close to one, we use the alternative approach for computing the enclosed angle between two normalized vectors as shown above. Also, in many scenarios $\varphi_\odot$, φ_{occ}, and δ are close to zero. In these cases, we can use the *small-angle approximation* ($\arcsin(\varphi) \approx \varphi$ for $\varphi \to 0$) to simplify the computation. For instance, if the angles are smaller than 10°, the error of this approximation is already below 0.1 %. Thus, for the common case, where both the Sun and the occluding body only cover a small solid angle, the following approximation can accelerate the computation (see *Section 6*):

$$\varphi_{occ} \approx \frac{r_{occ}}{d_{occ}} \qquad \varphi_\odot \approx \frac{r_\odot}{d_\odot} \qquad \delta \approx d. \tag{10}$$

Figure 12 shows the resulting maximum shadow intensity difference introduced by this simplification as a function of the distance to the shadow-casting body. The error is large close to the shadow-casting body but quickly decreases with distance. If the target application only requires to compute shadows cast onto celestial bodies which are sufficiently far away from the shadow caster, this approximation will serve well in practice. However, if also shadowing of
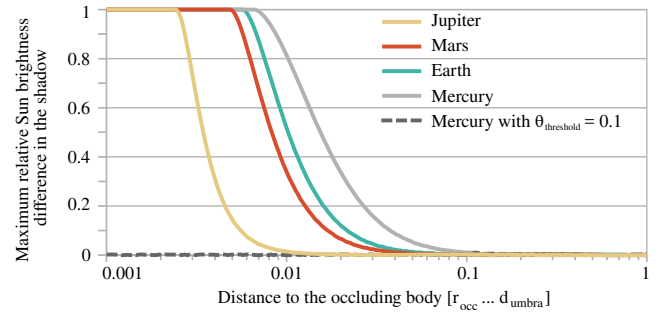
other objects such as spacecrafts orbiting the shadow caster should be simulated, a hybrid approach becomes necessary. For this, the three angles are first computed with *Equation 10* and if either of them exceeds a small threshold value (e.g. $\theta_{threshold} = 0.1$ rad), *Equation 9* is used instead for the respective angle instead. The dashed line in *Figure 12* shows the maximum error in the shadow of mercury for this value of $\theta_{threshold}$. This hybrid approach enables a fast computation in the common case and a high precision in the special case close to the shadow casting body.

Finally, once the texture coordinates are calculated using *Equation 7*, a single texture lookup is required to get the relative Sun brightness at the current location. The retrieved value is then multiplied with the Sun-based illuminance obtained by the rendering pipeline of CosmoScout VR.

6. Evaluation

In this section, we compare the presented approaches in terms of preprocessing time, rendering performance, and visual quality.

6.1. Preprocessing Time

The pre-computation needs to be done only once, as a single lookup texture can be used for all shadow-casting bodies. The time it takes to generate the texture scales linearly with the desired resolution and the amount of samples taken to integrate *Equation 1*. *Figure 11* shows the generation time and the corresponding error compared to the shadow texture computed with a large amount of samples. It can be seen, that 64^2 samples produce already quite good results for a 512^2 shadow-lookup texture. For HDR rendering, more bits of precision may be required to get artifact-free results. However, even with 1024^2 samples, the generation took only about one hundred seconds. Hence, this value can be deliberately increased. All eclipse shadow textures used for images in this paper were generated with 2048^2 samples and a resolution of 512^2 pixels.

Algorithm	Io / Jupiter	Moon / Earth	Phobos / Mars
Linear Gradient	0.06	0.87	0.91
Smoothstep Gradient	0.07	0.94	0.99
Circle Intersection	0.16	1.32	1.52
Circle Intersection (double precision)	30.9	57.1	65.3
Spherical Cap Intersection (double precision)	31.6	58.6	67.2
Proposed Approach (using *Equation 9*)	0.14	1.22	1.47
Proposed Approach (using *Equation 10*)	0.08	0.80	1.03

Table 1: *The scenes of* Figure 14 *were rendered at Full-HD resolution on a NVidia Quadro M6000 GPU. The table shows the increase in render time (in milliseconds) if the eclipse shadows are enabled, averaged over several hundred frames. The solar eclipses on Earth and Mars take longer to compute as the ray marching through the atmosphere requires a computation of the eclipse shadow many times per pixel. On Jupiter, the eclipse shadow is only evaluated once as there is no atmosphere simulated. It can be seen, that using double precision for this operation on the GPU is not feasible.*

(a) $B_{linear} - B_{LD}$ (b) $B_{smoothstep} - B_{LD}$

Figure 13: *This figure illustrates the difference between using a linear gradient in the penumbra (a) and a smoothstep gradient (b) when compared to our approach using circle intersections and limb darkening. Both graphs use the same parametrization of the complete eclipse shadow as* Figure 10 a. *A linear shadow gradient in the penumbra leads to darker outer regions of the eclipse shadow (up to about 10 %) while the antumbra and the regions close to the umbra are too bright (about 7 %). With the smoothstep gradient, the error close to the occluder is smaller (left-hand side of the chart). The vertical lines show the positions of the solar eclipses shown in* Figure 14. *It is also interesting to compare these graphs to* Figure 8 *and* Figure 9.

6.2. Rendering Performance

The performance cost at run-time is more important than the pre-processing time. In order to assess the rendering-performance impact, we have chosen three example scenes (*Figure 14*) and measured the increase in render time if one of the presented approaches is used for computing the eclipse shadows. The scenes have been chosen because they are representative for different regions of the eclipse shadow volume. When watched from the surface of Jupiter, Io appears to be much larger than the Sun, causing a very deep total eclipse. From the surface of Earth, Moon and Sun have nearly the same size. Hence, Earth passes through Moon's shadow approximately at the end of the umbra. Finally, Mars passes through the shadow of Phobos beyond the end of the umbra, causing an annular eclipse. To illustrate this, the positions of the occluded bodies in the respective shadow volumes are marked in *Figure 13*.

The measurement results are shown in *Table 1*. The approaches using a linear and a smoothstep-based gradient in the penumbra can be implemented very efficiently. However, computing the exact circle intersection in the shader, requires three evaluations of *arcsin*. Inverse trigonometric functions are usually not implemented in hardware, resulting in a significant performance impact. Using double precision requires a custom implementation of all trigonometric functions which makes the respective approaches unusable in practice. Using the proposed texture-lookup approach also requires three evaluations of *arcsin* (see *Equation 9*) which results in a similar performance as the direct computation of the circle intersection in the shader. Finally, the hybrid approach allows using *Equation 10* for all three example eclipses. As shown by the results, even the additional branching to test for the small angle case, has very little performance impact compared to the impact of *arcsin*.
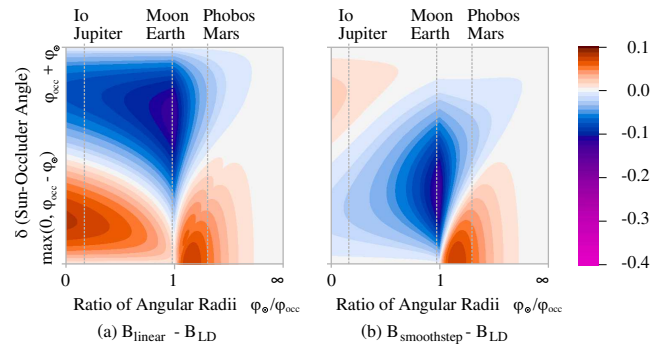
6.3. Visual Quality

To evaluate the visual quality, we compare the rendered images to real photographs. Furthermore, we compare the renderings using the circle-intersection method including limb darkening to images using the linear brightness gradient, as well as to images using the smoothstep-based gradient.

The first column of *Figure 14* shows photographs of the chosen example scenes; the second column shows corresponding renderings by CosmoScout VR using the proposed approach. A quantitative comparison of the brightness gradients was not possible as a deduction of physical luminance values from the reference images is difficult if not impossible. As noted in the image caption, the reference images are composite images which have been generated by combining and processing multiple images. This is required, as most satellites capture images for scientific purposes at very specific wavelengths. As a result, neither of the images really show how the eclipse would be perceived by a human observer. This becomes most apparent in the perceptually linear gradient from the umbra to the outer rim of the penumbra in the reference images. Since humans perceive brightness on a logarithmic scale (Weber-Fechner Law), the center of the shadow gradient would actually appear much brighter for a human observer. To mimic this, CosmoScout VR uses a non-linear tone-mapping operator to transform physical luminance values before displaying them on screen. For the comparison above, this final tone-mapping step in CosmoScout VR has been disabled. The other columns show the eclipse shadow computed with circle intersections but without limb darkening (c), a linear gradient in the penumbra (d), and a smoothstep-based gradient in the penumbra (e). The relative brightness difference is shown with the same color scale as *Figure 13*.
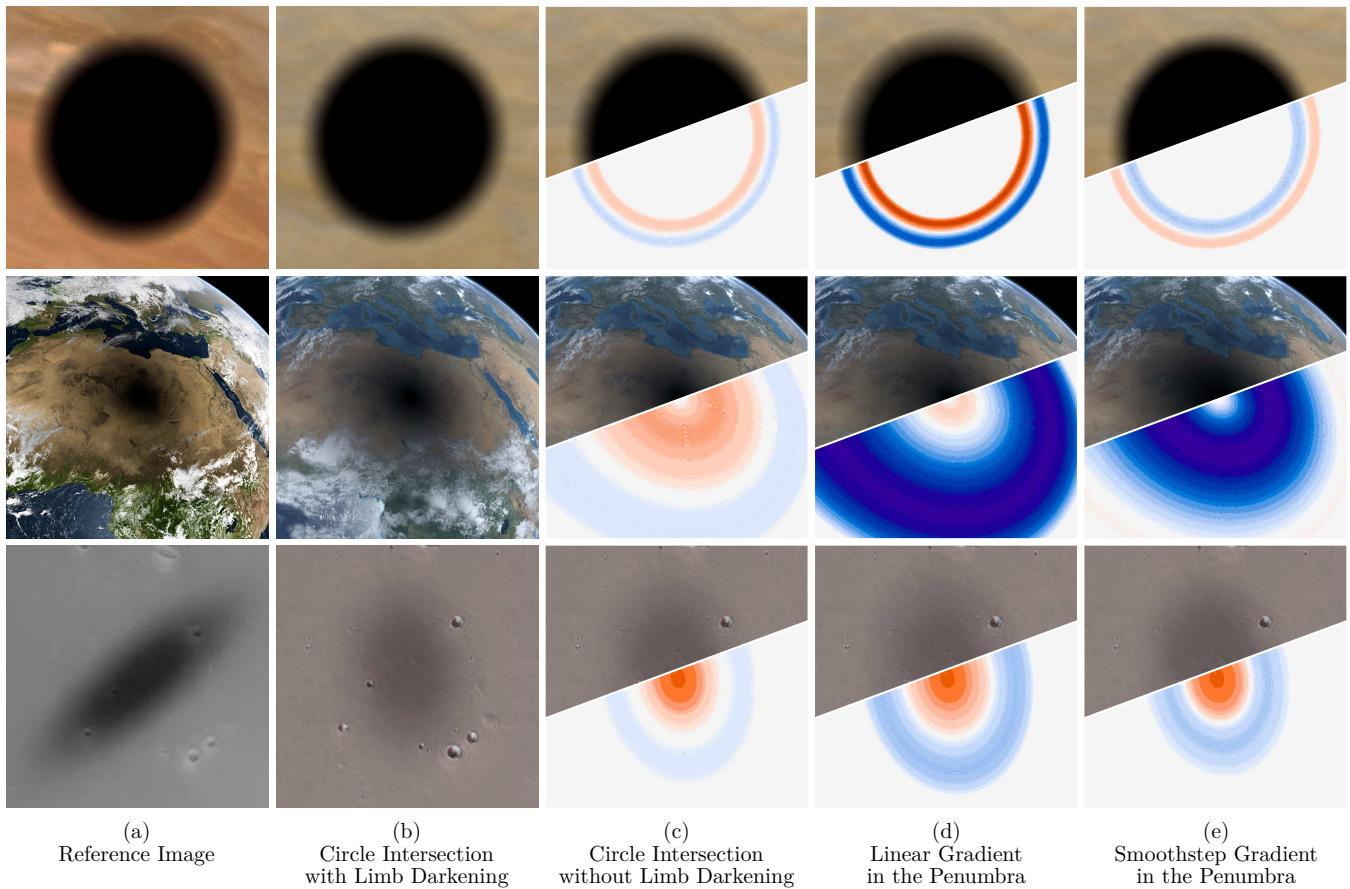
| (a) | (b) | (c) | (d) | (e) |
| Reference Image | Circle Intersection with Limb Darkening | Circle Intersection without Limb Darkening | Linear Gradient in the Penumbra | Smoothstep Gradient in the Penumbra |

Figure 14: *Three real solar eclipses have been chosen for comparison. These are from top to bottom: A total solar eclipse cast by Io onto Jupiter, a total solar eclipse cast by the Moon onto Earth and an annular eclipse cast by Phobos onto Mars. The first column of images shows reference images captured by spacecrafts. The image for the eclipse shadow cast by Io onto Jupiter is a composite of multiple photos captured by the Juno spacecraft on 2019-09-12 (CC-BY, NASA/JPL-Caltech/SwRI/MSSS, Kevin M. Gill). The total solar eclipse shadow over the Sahara occurred on 2006-03-29. The image was generated from MSG satellite images with a technique described in [RP11] by Maximilian Reuter. The annular eclipse cast by Phobos onto Mars has been captured by the HRSC camera onboard the Mars Express satellite on 2005-11-10 (CC BY-SA 3.0, ESA/DLR/FU Berlin, Gerhard Neukum). The shadow of Phobos moved quickly from West to East while the satellite moved from South to North during capturing the image line by line. Hence, the bottom pixels where captured several seconds before the top pixels in this image. This leads to the sheared appearance of the shadow. Also, the contrast in this image has been stretched significantly. The other columns show renderings of the same scenes using the proposed method (b) and the other methods described in this paper (c-e). The bottom halves show the respective shadow-intensity difference to column (b) using the same color map as* Figure 13. *For a detailed analysis of these images, refer to* Section 6.3.

A direct comparison with the reference image is not possible, since only the incoming sunlight is modelled correctly. For a direct comparison much more needs to be simulated. The surface material characteristics and atmospheric effects of the target body, as well as the optics and special characteristics of the sensors onboard the spacecraft have an impact which is not accounted for. A comparison to measurement data, which quantifies the amount of incident sunlight at various positions in the shadow volume of a celestial body, would be better, but to the best of our knowledge, such data is not available.

As described in *Section 4.3*, the impact of limb darkening is most intense for annular eclipses (c, bottom row). Overall, the linear gra-

dient performs worst with large differences for all three types of eclipses. Also, the outer edge of the penumbra is too sharp with a linear gradient (d, upper row). The smoothstep-based gradient produces better results close to the shadow caster (e, upper row), but similar results close to the end of the umbra (e, center row). Most of these findings are also well visible in *Figure 13*.

7. Discussion & Future Work

In this paper, we presented and evaluated a novel approach for simulating eclipse shadows in real-time using an existing open source simulation of our Solar System, CosmoScout VR [SZGG22]. The proposed method produces more realistic shadows than the state-

of-the-art while maintaining the same performance impact. The quality is improved by actually computing the occluded fraction of the solar disc and by incorporating the effect of limb darkening. The mathematically correct approach using the intersection of spherical caps was impossible to implement due to precision issues, but an approximation using circles instead of spherical caps provided artifact-free results. The errors in the approximations are less than 1% for eclipses in our Solar System. Furthermore, we have shown that a lookup texture can be used to include the effect of limb darkening while maintaining a similar performance impact as the state-of-the-art. The parametrization of this texture is chosen to be independent of the size and position of the shadow caster. Thus, only one such texture can be used for all shadow casters in the Solar System.

We have limited the problem to spherical, atmosphereless bodies. However, the shape of many celestial bodies can be better approximated with an ellipsoid. While it would be straight-forward to use an asymmetric scaling of the shadow volume, this would not be a correct solution. It remains future work to quantify the remaining error this approximation would introduce. A method based on Heitz's research on solid angles subtended by ellipsoids [Hei17] might be promising. In addition, we plan to expand the system to also include atmospheric effects. This will enable the proper simulation of lunar eclipses (and potentially eclipses on other moons, such as Phobos), where photons traveling through the atmosphere are refracted and can thus reach objects in the umbra region. The parametrization of the shadow-lookup texture which includes the umbra is specifically designed for this purpose. Also, including a model of wavelength-dependent limb darkening will be interesting. This would lead to subtle color shifts in the eclipse shadow even if the occluding body does not have an atmosphere.

While approximations are often *good enough*, improving them would be hard, since the function needs to take into account the angular radius of the occluder when compared to the angular radius of the Sun and the effects of limb darkening. Ultimately, this would lead to a function which analytically approximates the values of the proposed lookup texture. Overall, the proposed approach for real-time rendering of eclipses allows for a more realistic simulation of our Solar System with a comparable impact on rendering performance when compared to the state-of-the-art.

References

[Act96] ACTON C. H.: Ancillary data services of NASA's Navigation and Ancillary Information Facility. *Planetary and Space Science 44*, 1 (1996), 65–70. doi:https://doi.org/10.1016/0032-0633(95)00107-7. 4

[AK08] ASSENMACHER I., KUHLEN T.: The ViSTA virtual reality toolkit. *Proceedings of the IEEE VR SEARIS* (2008), 23–26. 4

[BAC*19] BOCK A., AXELSSON E., COSTA J., PAYNE G., ACINA-PURA M., TRAKINSKI V., EMMART C., SILVA C., HANSEN C., YN-NERMAN A.: Openspace: A system for astrographics. *IEEE transactions on visualization and computer graphics 26*, 1 (2019), 633–642. doi:10.1109/TVCG.2019.2934259. 3

[BLR*18] BROCHARD R., LEBRETON J., ROBIN C., KANANI K., JON-NIAUX G., MASSON A., DESPRÉ N., BERJAOUI A.: Scientific image rendering for space scenes with the surrender software. *69th International Astronautical Congress (IAC)* (2018). doi:10.48550/arXiv.1810.01423. 3

[EKBS12] EMILIO M., KUHN J. R., BUSH R. I., SCHOLL I. F.: Measuring the Solar Radius from Space during the 2003 and 2006 Mercury Transits. *The Astrophysical Journal 750*, 2 (April 2012), 135. doi:10.1088/0004-637x/750/2/135. 2

[ESAW11] EISEMANN E., SCHWARZ M., ASSARSSON U., WIMMER M.: *Real-time shadows*. CRC Press, 2011. 3

[FHLZ08] FIVIAN M. D., HUDSON H. S., LIN R. P., ZAHID H. J.: A large excess in apparent solar oblateness due to surface magnetism. *Science 322* (2008), 560–562. doi:10.1126/science.1160863. 2

[Gre17] GREGORIO F.: *Celestia User's Guide for Version 1.6.1*, September 2017. URL: http://www.celestiamotherlode.net/catalog/documentation.html. 3

[Hei17] HEITZ E.: Analytical calculation of the solid angle subtended by an arbitrarily positioned ellipsoid to a point source. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 852* (2017), 10–14. doi:https://doi.org/10.1016/j.nima.2017.02.004. 11

[HM98] HESTROFFER D., MAGNAN C.: Wavelength dependency of the solar limb darkening. *Astronomy and Astrophysics 333* (1998), 338–342. 2

[Kin14] KINGDOM F. A. A.: Mach bands explained by response normalization. *Frontiers in human neuroscience 8* (2014), 843. doi:10.3389/fnhum.2014.00843. 4

[MED12] MÜLLER D., ENGEL J., DÖLLNER J.: Single-Pass Rendering of Day and Night Sky Phenomena. In *Vision, Modeling and Visualization* (11 2012), Goesele M., Grosch T., Theisel H., Toennies K., Preim B., (Eds.), The Eurographics Association, pp. 55–62. doi:10.2312/PE/VMV/VMV12/055-062. 3

[MJOS17] MOON B., JEONG D.-G., OH S., SOHN J.: Variation in solar limb darkening coefficient estimated from solar images taken by SOHO and SDO. *Journal of Astronomy and Space Sciences 34*, 2 (2017), 99–103. doi:10.5140/JASS.2017.34.2.99. 2

[OS07] OAT C., SANDER P. V.: Ambient aperture lighting. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2007), I3D '07, ACM, pp. 61–64. doi:10.1145/1230100.1230111. 3, 6

[RN09] ROZELOT J.-P., NEINER C. (Eds.): *What Is Coming: Issues Raised from Observation of the Shape of the Sun*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 15–43. doi:10.1007/978-3-540-87831-5_2. 2

[RP11] REUTER M., PFEIFER S.: Moments from space captured by MSG SEVIRI. *International Journal of Remote Sensing 32*, 14 (2011), 4131–4140. doi:10.1080/01431161.2011.566288. 10

[Sem18] SEMENOV B.: WebGeocalc and cosmographia: Modern tools to access OPS SPICE data. In *2018 SpaceOps Conference* (2018), p. 2366. doi:10.2514/6.2018-2366. 3

[SFG] SCHNEEGANS S., FLATKEN M., GERNDT A.: Source code of CosmoScout VR. URL: https://github.com/cosmoscout/cosmoscout-vr, doi:10.5281/zenodo.3381953. 4

[SGGC04] SERON F., GUTIERREZ D., GUTIERREZ G., CEREZO E.: Visualizing sunsets through inhomogeneous atmospheres. In *Proceedings Computer Graphics International* (2004), pp. 349–356. doi:10.1109/CGI.2004.1309232. 3

[SZGG22] SCHNEEGANS S., ZEUMER M., GILG J., GERNDT A.: CosmoScout VR: A Modular 3D Solar System Based on SPICE. In *2022 IEEE Aerospace Conference (AERO)* (2022), IEEE, pp. 1–13. doi:10.1109/AERO53065.2022.9843488. 4, 10

[TV01] TOVCHIGRECHKO A., VAKSER I. A.: How common is the funnel-like energy landscape in protein-protein interactions? *Protein Science 10*, 8 (8 2001), 1572–1583. doi:10.1110/ps.8701. 3

[YC09] YAPO T. C., CUTLER B.: Rendering lunar eclipses. In *Proceedings of Graphics Interface 2009* (Toronto, Canada, 2009), GI '09, Canadian Information Processing Society, pp. 63–69. 3